

IBM Macros for IBM EMM  
Version 9.0  
11 décembre 2012

*Guide d'utilisation*



**Important**

Avant d'utiliser le présent document et le produit associé, prenez connaissance des informations générales figurant à la section «Remarques», à la page 257.

**Remarque**

Certaines illustrations de ce manuel ne sont pas disponibles en français à la date d'édition.

LE PRESENT DOCUMENT EST LIVRE EN L'ETAT SANS AUCUNE GARANTIE EXPLICITE OU IMPLICITE. IBM DECLINE NOTAMMENT TOUTE RESPONSABILITE RELATIVE A CES INFORMATIONS EN CAS DE CONTREFACON AINSI QU'EN CAS DE DEFAUT D'APTITUDE A L'EXECUTION D'UN TRAVAIL DONNE.

Ce document est mis à jour périodiquement. Chaque nouvelle édition inclut les mises à jour. Les informations qui y sont fournies sont susceptibles d'être modifiées avant que les produits décrits ne deviennent eux-mêmes disponibles. En outre, il peut contenir des informations ou des références concernant certains produits, logiciels ou services non annoncés dans ce pays. Cela ne signifie cependant pas qu'ils y seront annoncés.

Pour plus de détails, pour toute demande d'ordre technique, ou pour obtenir des exemplaires de documents IBM, référez-vous aux documents d'annonce disponibles dans votre pays, ou adressez-vous à votre partenaire commercial.

Vous pouvez également consulter les serveurs Internet suivants :

- <http://www.fr.ibm.com> (serveur IBM en France)
- <http://www.can.ibm.com> (serveur IBM au Canada)
- <http://www.ibm.com> (serveur IBM aux Etats-Unis)

*Compagnie IBM France  
Direction Qualité  
17, avenue de l'Europe  
92275 Bois-Colombes Cedex*

© Copyright IBM Corporation 1998, 2012.

# Table des matières

## Avis aux lecteurs canadiens . . . . . v

## Chapitre 1. Utilisation de macros dans IBM Campaign . . . . . 1

Récapitulatif des fonctions macro . . . . .	1
Fonctions statistiques . . . . .	1
Fonctions mathématiques et trigonométriques . . . . .	2
Fonctions des chaînes de caractères . . . . .	6
Fonctions d'horodatage . . . . .	7
Fonctions de groupement . . . . .	8
Fonctions diverses . . . . .	8
Paramètres des fonctions macro . . . . .	8
Spécifications de format . . . . .	9
Utilisation des constantes . . . . .	9

## Chapitre 2. Utilisation de macros dans IBM Interact . . . . . 11

Récapitulatif des fonctions macro . . . . .	11
Fonctions statistiques . . . . .	11
Fonctions mathématiques et trigonométriques . . . . .	12
Fonctions des chaînes de caractères . . . . .	13
Fonctions d'horodatage . . . . .	14
Fonctions diverses . . . . .	14
Paramètres des fonctions macro . . . . .	14
Spécifications de format . . . . .	14
Utilisation des constantes . . . . .	15

## Chapitre 3. Utilisation de macros dans IBM PredictiveInsight . . . . . 17

Récapitulatif des fonctions macro . . . . .	17
Fonctions statistiques . . . . .	17
Fonctions mathématiques et trigonométriques . . . . .	19
Fonctions d'ingénierie . . . . .	22
Fonctions des chaînes de caractères . . . . .	23
Fonctions d'horodatage . . . . .	24
Fonctions de groupement . . . . .	25
Fonctions diverses . . . . .	25
Remarques sur les pages de références des macros . . . . .	26
Paramètres des fonctions macro . . . . .	27
Spécifications de format . . . . .	27
Utilisation des plages de cibles et de colonnes . . . . .	27
Utilisation des constantes . . . . .	28
Précision de calcul . . . . .	29
Résultats de cible non valides et cibles vides . . . . .	30

## Chapitre 4. Guide de référence des macros IBM . . . . . 33

ABS . . . . .	33
ACOS . . . . .	34
ACOT . . . . .	36
ADD_MONTHS . . . . .	37
ALIGN . . . . .	38
AND . . . . .	39

ASIN . . . . .	40
ATAN . . . . .	42
AVG . . . . .	44
AVG_DEV . . . . .	46
BETWEEN . . . . .	48
BIT_AND . . . . .	48
BIT_NOT . . . . .	50
BIT_OR . . . . .	51
BIT_XOR . . . . .	53
BUFFER . . . . .	55
CEILING . . . . .	56
COLUMN . . . . .	57
CONSTANT . . . . .	59
COS . . . . .	60
COSH . . . . .	62
COT . . . . .	64
COUNT . . . . .	65
COUNT_DIFF . . . . .	66
COV . . . . .	68
CURRENT_DATE . . . . .	69
CURRENT_DAY . . . . .	70
CURRENT_JULIAN . . . . .	71
CURRENT_MONTH . . . . .	71
CURRENT_TIME . . . . .	72
Configuration des dates sur votre application Web . . . . .	72
CURRENT_WEEKDAY . . . . .	73
CURRENT_YEAR . . . . .	74
CV_FOLDS . . . . .	75
DATALINK . . . . .	76
DATE . . . . .	77
DATE_FORMAT . . . . .	80
DATE_JULIAN . . . . .	81
DATE_STRING . . . . .	82
DAY_BETWEEN . . . . .	84
DAY_FROMNOW . . . . .	84
DAY_INTERVAL . . . . .	85
DAYOF . . . . .	86
DDELINK . . . . .	87
DECIMATE . . . . .	88
DELAY . . . . .	89
DERIVATIVE . . . . .	90
DISTANCE . . . . .	92
DISTINCT . . . . .	92
DIV . . . . .	93
EQ . . . . .	95
EXP . . . . .	96
EXTERNALCALLOUT . . . . .	98
EXTRACT . . . . .	98
FACTORIAL . . . . .	100
FLOOR . . . . .	101
FORMAT . . . . .	102
FRACTION . . . . .	105
GAUSS . . . . .	106
GAUSS_AREA . . . . .	108
GE . . . . .	111

GRID . . . . .	112	RANDOM_GAUSS . . . . .	197
GROUPBY . . . . .	114	RANK . . . . .	199
GROUPBY_WHERE . . . . .	117	REPEAT . . . . .	200
GT . . . . .	117	ROTATE_LEFT . . . . .	202
HISTOGRAM . . . . .	119	ROTATE_RIGHT . . . . .	203
IF . . . . .	121	ROUND . . . . .	204
IN . . . . .	122	ROWNUM . . . . .	205
INIT . . . . .	123	RTRIM . . . . .	206
INT . . . . .	125	SAMPLE_RANDOM . . . . .	206
INTEGRAL . . . . .	126	SELECT . . . . .	207
INVERSE . . . . .	128	SIGN . . . . .	209
EST . . . . .	129	SIN . . . . .	210
ISERROR . . . . .	129	SINH . . . . .	212
ISEVEN . . . . .	130	SKEW . . . . .	213
ISMEMBER . . . . .	132	SLIDE_WINDOW . . . . .	215
ISODD . . . . .	133	SORT . . . . .	217
KURTOSIS . . . . .	134	SQRT . . . . .	219
LAG . . . . .	137	STAT . . . . .	220
LE . . . . .	138	STDV ou STDEV . . . . .	222
LIKE . . . . .	140	STRING_CONCAT . . . . .	224
LN ou LOG . . . . .	142	STRING_HEAD . . . . .	226
LOG2 . . . . .	143	STRING_LENGTH . . . . .	227
LOG10 . . . . .	144	STRING_PROPER . . . . .	228
LOWER . . . . .	145	STRING_SEG . . . . .	229
LT . . . . .	146	STRING_TAIL . . . . .	230
LTRIM . . . . .	148	SUBSAMPLE . . . . .	231
MAX . . . . .	148	SUBSTITUTE . . . . .	233
MAXINDEX . . . . .	150	SUBSTR ou SUBSTRING . . . . .	234
MEAN . . . . .	152	SUM . . . . .	235
MERGE . . . . .	154	TAN . . . . .	237
MIN . . . . .	155	TANH . . . . .	238
MINUS . . . . .	157	TO . . . . .	240
MOD . . . . .	158	TOTAL . . . . .	241
MONTHOF . . . . .	160	TRANPOSE . . . . .	242
MULT . . . . .	161	TRUNCATE . . . . .	244
NE . . . . .	162	UPPER . . . . .	245
NORM_MINMAX . . . . .	164	VARIANCE . . . . .	246
NORM_SIGMOID . . . . .	168	WEEKDAY . . . . .	248
NORM_ZSCORE . . . . .	171	WEEKDAYOF . . . . .	249
NOT . . . . .	174	XOR . . . . .	250
NPV . . . . .	176	XTAB . . . . .	251
NUMBER . . . . .	177	YEAROF . . . . .	253
OFFSET . . . . .	185		
OR . . . . .	186	<b>Comment contacter le support</b>	
PCA . . . . .	188	<b>technique IBM . . . . .</b>	<b>255</b>
PCA_FEATURES . . . . .	189		
POSITION . . . . .	191	<b>Remarques . . . . .</b>	<b>257</b>
PLUS . . . . .	193	Marques . . . . .	259
POW . . . . .	194	Règles de confidentialité et conditions d'utilisation	259
RANDOM . . . . .	196		

---

## Avis aux lecteurs canadiens

Le présent document a été traduit en France. Voici les principales différences et particularités dont vous devez tenir compte.

### Illustrations

Les illustrations sont fournies à titre d'exemple. Certaines peuvent contenir des données propres à la France.

### Terminologie

La terminologie des titres IBM peut différer d'un pays à l'autre. Reportez-vous au tableau ci-dessous, au besoin.

IBM France	IBM Canada
ingénieur commercial	représentant
agence commerciale	succursale
ingénieur technico-commercial	informaticien
inspecteur	technicien du matériel

### Claviers

Les lettres sont disposées différemment : le clavier français est de type AZERTY, et le clavier français-canadien de type QWERTY.








### OS/2 et Windows - Paramètres canadiens

Au Canada, on utilise :

- les pages de codes 850 (multilingue) et 863 (français-canadien),
- le code pays 002,
- le code clavier CF.

### Nomenclature

Les touches présentées dans le tableau d'équivalence suivant sont libellées différemment selon qu'il s'agit du clavier de la France, du clavier du Canada ou du clavier des États-Unis. Reportez-vous à ce tableau pour faire correspondre les touches françaises figurant dans le présent document aux touches de votre clavier.

<b>France</b>	<b>Canada</b>	<b>Etats-Unis</b>
 (Pos1)		Home
Fin	Fin	End
 (PgAr)		PgUp
 (PgAv)		PgDn
Inser	Inser	Ins
Suppr	Suppr	Del
Echap	Echap	Esc
Attn	Intrp	Break
Impr écran	ImpEc	PrtSc
Verr num	Num	Num Lock
Arrêt défil	Défil	Scroll Lock
 (Verr maj)	FixMaj	Caps Lock
AltGr	AltCar	Alt (à droite)

### **Brevets**

Il est possible qu'IBM détienne des brevets ou qu'elle ait déposé des demandes de brevets portant sur certains sujets abordés dans ce document. Le fait qu'IBM vous fournisse le présent document ne signifie pas qu'elle vous accorde un permis d'utilisation de ces brevets. Vous pouvez envoyer, par écrit, vos demandes de renseignements relatives aux permis d'utilisation au directeur général des relations commerciales d'IBM, 3600 Steeles Avenue East, Markham, Ontario, L3R 9Z7.

### **Assistance téléphonique**

Si vous avez besoin d'assistance ou si vous voulez commander du matériel, des logiciels et des publications IBM, contactez IBM direct au 1 800 465-1234.

---



# Chapitre 1. Utilisation de macros dans IBM Campaign

Ce chapitre fournit des informations d'utilisation sur les macros pouvant être utilisées dans IBM® Campaign. Tous les utilisateurs IBM Campaign doivent lire ce chapitre avant de consulter le reste du présent document.

---

## Récapitulatif des fonctions macro

Les tableaux de cette section récapitulent les fonctions macro par catégorie.

**Important :** Toutes les macros figurant dans ce document sont disponibles dans IBM Campaign. Les macros disponibles uniquement dans IBM PredictiveInsight sont désignées par cette icône :  Les macros disponibles uniquement dans IBM Interact sont désignées par cette icône : 

Des pages de référence détaillées sont fournies pour chaque macro, dans l'ordre alphabétique, à partir de la rubrique Chapitre 4, «Guide de référence des macros IBM », à la page 33. La rubrique «Paramètres des fonctions macro», à la page 27 fournit des informations sur les paramètres en entrée de la fonction macro.

### Information associée:

«Fonctions statistiques»

«Fonctions mathématiques et trigonométriques», à la page 2

«Fonctions des chaînes de caractères», à la page 6

«Fonctions d'horodatage», à la page 7

«Fonctions de groupement», à la page 8

«Fonctions diverses», à la page 8

## Fonctions statistiques

Nom de la macro	Renvoie	Description
AVG	Valeur unique dans une nouvelle colonne pour le mot clé ALL ; une colonne avec une valeur unique par colonne d'entrée pour le mot clé COL ; une colonne avec une valeur par ligne pour le mot clé ROW.	Calcule la moyenne arithmétique d'une plage de cibles
COUNT	Valeur unique dans une nouvelle colonne.	Compte le nombre de valeurs d'une plage de données spécifiée
MAX	Valeur unique dans une nouvelle colonne pour le mot clé ALL ; une colonne avec une valeur unique par colonne d'entrée pour le mot clé COL ; une colonne avec une valeur par ligne pour le mot clé ROW.	Calcule la valeur maximale d'une plage de cibles

Nom de la macro	Renvoie	Description
MEAN	Valeur unique dans une nouvelle colonne pour le mot clé ALL ; une colonne avec une valeur unique par colonne d'entrée pour le mot clé COL ; une colonne avec une valeur par ligne pour le mot clé ROW.	Calcule la moyenne arithmétique d'une plage de cibles
MIN	Valeur unique dans une nouvelle colonne pour le mot clé ALL ; une colonne avec une valeur unique par colonne d'entrée pour le mot clé COL ; une colonne avec une valeur par ligne pour le mot clé ROW.	Calcule la valeur minimale d'une plage de cibles
STDV ou STDEV	Valeur unique dans une nouvelle colonne pour le mot clé ALL ; une colonne avec une valeur unique par colonne d'entrée pour le mot clé COL ; une colonne avec une valeur par ligne pour le mot clé ROW.	Calcule l'écart type d'une plage de cibles
VARIANCE	Valeur unique dans une nouvelle colonne pour le mot clé ALL ; une colonne avec une valeur unique par colonne d'entrée pour le mot clé COL ; une colonne avec une valeur par ligne pour le mot clé ROW.	Calcule la variance d'une plage de cibles

## Fonctions mathématiques et trigonométriques

Nom de la macro	Renvoie	Description
ABS	Une colonne par colonne d'entrée	Calcule la valeur absolue du contenu de la plage de données spécifiée
ACOS	Une colonne par colonne d'entrée	Calcule le cosinus inverse du contenu de la plage de données spécifiée
ACOT	Une colonne par colonne d'entrée	Calcule l'arc cotangente du contenu de la plage de données spécifiée
ASIN	Une colonne par colonne d'entrée	Calcule l'arc sinus du contenu de la plage de données spécifiée
ATAN	Une colonne par colonne d'entrée	Calcule l'arc tangente du contenu de la plage de données spécifiée



Nom de la macro	Renvoie	Description
AVG	Une colonne par colonne d'entrée	Calcule la moyenne arithmétique des cibles de la plage de données spécifiée
BETWEEN	Une colonne par colonne d'entrée	Compare deux valeurs pour déterminer si la valeur fournie est comprise entre deux autres valeurs
CEILING	Une colonne par colonne d'entrée	Calcule le plafond de chaque valeur de la plage de données spécifiée
COLUMN	Une colonne par colonne d'entrée	Crée des colonnes, en concaténant verticalement les valeurs en entrée dans chaque colonne
COS	Une colonne par colonne d'entrée	Calcule le cosinus du contenu de la plage de données spécifiée
COSH	Une colonne par colonne d'entrée	Calcule le cosinus hyperbolique du contenu de la plage de données spécifiée
COT	Une colonne par colonne d'entrée	Calcule la cotangente du contenu de la plage de données spécifiée
COUNT	Une colonne contenant une valeur unique	Compte le nombre de cibles contenant des valeurs dans la plage de données spécifiée
EXP	Une colonne par colonne d'entrée	Calcule le nombre naturel (e) élevé à la puissance du contenu de chaque cible de la plage de données spécifiée
FACTORIAL	Une colonne par colonne d'entrée	Calcule la factorielle de chaque valeur de la plage de données spécifiée
FLOOR	Une colonne par colonne d'entrée	Calcule la partie entière de chaque valeur de la plage de données spécifiée
FRACTION	Une colonne par colonne d'entrée	Renvoie la partie décimale de chaque valeur de la plage de données spécifiée
INT	Une colonne par colonne d'entrée	Calcule la valeur entière (arrondie) du contenu de la plage de données spécifiée
INVERSE	Une colonne par colonne d'entrée	Calcule la valeur négative du contenu de la plage de données spécifiée
LN	Une colonne par colonne d'entrée	Calcule le logarithme naturel du contenu de la plage de données spécifiée
LOG	Une colonne par colonne d'entrée	Calcule le logarithme naturel du contenu de la plage de données spécifiée

Nom de la macro	Renvoie	Description
LOG2	Une colonne par colonne d'entrée	Calcule le logarithme base 2 du contenu de la plage de données spécifiée
LOG10	Une colonne par colonne d'entrée	Calcule le logarithme base 10 du contenu de la plage de données spécifiée
MAX	Valeur unique dans une nouvelle colonne pour le mot clé ALL ; une colonne avec une valeur unique par colonne d'entrée pour le mot clé COL ; une colonne avec une valeur par ligne pour le mot clé ROW.	Calcule la valeur maximale d'une plage de cibles
MEAN	Valeur unique dans une nouvelle colonne pour le mot clé ALL ; une colonne avec une valeur unique par colonne d'entrée pour le mot clé COL ; une colonne avec une valeur par ligne pour le mot clé ROW.	Calcule la moyenne arithmétique d'une plage de cibles
MIN	Valeur unique dans une nouvelle colonne pour le mot clé ALL ; une colonne avec une valeur unique par colonne d'entrée pour le mot clé COL ; une colonne avec une valeur par ligne pour le mot clé ROW.	Calcule la valeur minimale d'une plage de cibles
RANDOM	Une colonne avec le nombre de valeurs spécifié	Renvoie le nombre spécifié de nombres aléatoires
RANDOM_GAUSS	Une colonne avec le nombre de valeurs spécifié	Renvoie le nombre spécifié de valeurs aléatoires à partir d'une distribution de Gauss
ROUND	Une colonne par colonne d'entrée	Calcule la valeur arrondie du contenu de la plage de données spécifiée
SIGN	Une colonne par colonne d'entrée	Détermine le signe (positif ou négatif) des valeurs dans la plage de données spécifiée
SIN	Une colonne par colonne d'entrée	Calcule le sinus du contenu de la plage de données spécifiée
SINH	Une colonne par colonne d'entrée	Calcule le sinus hyperbolique du contenu de la plage de données spécifiée
SQRT	Une colonne par colonne d'entrée	Calcule la racine carrée du contenu de la plage de données spécifiée

Nom de la macro	Renvoi	Description
STDV ou STDEV	Valeur unique dans une nouvelle colonne pour le mot clé ALL ; une colonne avec une valeur unique par colonne d'entrée pour le mot clé COL ; une colonne avec une valeur par ligne pour le mot clé ROW.	Calcule l'écart type d'une plage de cibles
SUM	Valeur unique dans une nouvelle colonne pour le mot clé ALL ; une colonne avec une valeur unique par colonne d'entrée pour le mot clé COL ; une colonne avec une valeur par ligne pour le mot clé ROW.	Calcule la somme d'une plage de cibles
TAN	Une colonne par colonne d'entrée	Calcule la tangente du contenu de la plage de données spécifiée
TANH	Une colonne par colonne d'entrée	Calcule la tangente hyperbolique du contenu de la plage de données spécifiée
TOTAL	Valeur unique dans une nouvelle colonne pour le mot clé ALL ; une colonne avec une valeur unique par colonne d'entrée pour le mot clé COL ; une colonne avec une valeur par ligne pour le mot clé ROW.	Calcule la somme d'une plage de cibles
TRUNCATE	Une colonne par colonne d'entrée	Renvoie la partie non décimale de chaque valeur de la plage de données spécifiée
VARIANCE	Valeur unique dans une nouvelle colonne pour le mot clé ALL ; une colonne avec une valeur unique par colonne d'entrée pour le mot clé COL ; une colonne avec une valeur par ligne pour le mot clé ROW.	Calcule la variance d'une plage de cibles

## Fonctions des chaînes de caractères

Nom de la macro	Renvoie	Description
FORMAT	Une colonne par colonne d'entrée	Fournit un contrôle de formatage en sortie pour les nombres et les chaînes (tel que la largeur de la sortie, l'alignement, la précision numérique, le symbole du séparateur décimal, le symbole de groupement, etc.). Renvoie la chaîne en sortie formatée.
LIKE	Une colonne par colonne d'entrée	Détermine si une chaîne de texte correspond à un modèle spécifié
LOWER	Une colonne par colonne d'entrée	Convertit une valeur de chaîne en minuscules
LTRIM	Une colonne par colonne d'entrée	Supprime les espaces de début de chaque valeur de chaîne
NUMBER	Une colonne par colonne d'entrée	Convertit les chaînes de texte ASCII des heures et des dates en valeurs numériques
POSITION	Une colonne par colonne d'entrée	Renvoie la position initiale d'un modèle dans une chaîne de texte
RTRIM	Une colonne par colonne d'entrée	Supprime les espaces de fin de chaque valeur de chaîne
STRING_CONCAT	Une colonne avec une valeur pour chaque ligne de la colonne d'entrée la plus courte	Concatène les chaînes de texte des plages de données spécifiées
STRING_HEAD	Une colonne par colonne d'entrée	Renvoie les <i>n</i> premiers caractères de chaque chaîne de la plage de données spécifiée
STRING_LENGTH	Une colonne par colonne d'entrée	Renvoie la longueur de chaque chaîne de la plage de données spécifiée
STRING_PROPER	Une colonne par colonne d'entrée	Convertit chaque valeur de chaîne en mettant en majuscule la première lettre de chaque chaîne ou toute lettre suivant un espace blanc ou un symbole (autre que le trait de soulignement) et transforme en minuscules tous les autres caractères
STRING_SEG	Une colonne par colonne d'entrée	Renvoie le segment de chaîne entre deux index spécifiés

Nom de la macro	Renvoie	Description
STRING_TAIL	Une colonne par colonne d'entrée	Renvoie les $n$ derniers caractères de chaque chaîne de la plage de données spécifiée
SUBSTR ou SUBSTRING	Une colonne par colonne d'entrée	Renvoie les caractères d'une chaîne à partir d'une position initiale
UPPER	Une colonne par colonne d'entrée	Convertit une valeur de chaîne en majuscules

## Fonctions d'horodatage

Nom de la macro	Renvoie	Description
ADD_MONTHS	Une colonne par colonne d'entrée	Renvoie une date après l'ajout du nombre de mois spécifié
CURRENT_DATE	Une colonne par colonne d'entrée	Renvoie la date du jour dans le format format
CURRENT_DAY	Une colonne par colonne d'entrée	Renvoie le jour du mois en cours sous la forme d'un nombre compris entre 1 et 31
CURRENT_JULIAN	Une colonne par colonne d'entrée	Renvoie le nombre au format Julien de la date du jour
CURRENT_MONTH	Une colonne par colonne d'entrée	Renvoie le mois en cours de l'année sous la forme d'un nombre compris entre 1 et 12
CURRENT_TIME	Une colonne par colonne d'entrée	Renvoie l'heure actuelle sous la forme d'une chaîne
CURRENT_WEEKDAY	Une colonne par colonne d'entrée	Renvoie le jour de la semaine en cours du mois sous la forme d'un nombre compris entre 0 et 6
CURRENT_YEAR	Une colonne par colonne d'entrée	Renvoie l'année en cours sous la forme d'un nombre
DATE	Une colonne par colonne d'entrée	Convertit une chaîne de date en date du calendrier Julien
DATE_FORMAT	Une colonne par colonne d'entrée	Convertit les formats de date
DATE_JULIAN	Une colonne par colonne d'entrée	Renvoie la date au format Julien
DATE_STRING	Une colonne par colonne d'entrée	Renvoie la chaîne de date sous la forme d'une date du calendrier Julien
DAY_BETWEEN	Une colonne par colonne d'entrée	Renvoie le nombre de jours entre deux dates
DAY_FROMNOW	Une colonne par colonne d'entrée	Renvoie le nombre de jours entre la date du jour et la date spécifiée
DAY_INTERVAL	Une colonne par colonne d'entrée	Renvoie le nombre de jours entre deux dates

Nom de la macro	Renvoie	Description
DAYOF	Une colonne par colonne d'entrée	Renvoie le jour du mois sous la forme d'un nombre
MONTHOF	Une colonne par colonne d'entrée	Renvoie le mois de l'année sous la forme d'un nombre
WEEKDAY	Une colonne par colonne d'entrée	Convertit les chaînes de date textuelles ASCII en jour de la semaine
WEEKDAYOF	Une colonne par colonne d'entrée	Renvoie le jour de la semaine de la semaine sous forme de nombre
YEAROF	Une colonne par colonne d'entrée	Renvoie l'année sous la forme d'un nombre

## Fonctions de groupement

Nom de la macro	Renvoie	Description
GROUPBY	Une colonne avec une valeur pour chaque ligne	Récapitule plusieurs lignes de données dans un groupe
GROUPBY_WHERE	Une colonne avec une valeur pour chaque ligne	Récapitule plusieurs lignes de données répondant à une condition indiquée et se trouvant dans un groupe

## Fonctions diverses

Nom de la macro	Renvoie	Description
IF	Une colonne avec une valeur pour chaque ligne de la colonne d'entrée la plus courte	Commence une instruction conditionnelle if-then-else
ISERROR	Une colonne avec une valeur pour chaque ligne de la colonne d'entrée la plus courte	Renvoie un un si une valeur de la ligne en entrée contient une cible erronée (???) ; renvoie zéro dans le cas contraire
ISEVEN	Une colonne par colonne d'entrée	Teste si les valeurs en entrée sont paires (à savoir, divisible par deux)
ISODD	Une colonne par colonne d'entrée	Teste si les valeurs en entrée sont impaires (à savoir, non divisible par deux)
ROWNUM	Une colonne par colonne d'entrée	Génère des nombres séquentiels de un à n, où n correspond au nombre d'enregistrements

---

## Paramètres des fonctions macro

Cette section décrit les paramètres et l'utilisation des fonctions macro dans IBM Campaign.

## Spécifications de format

Cette section décrit le format de certains paramètres fréquemment utilisés. Elle s'applique à toutes les références à ces paramètres par les spécifications de fonction macro de ce chapitre.

### **data**

Le paramètre data représente une colonne de données sur laquelle une fonction macro doit agir. Il peut s'agir d'une constante ou d'une zone. Pour plus d'informations, reportez-vous à la fonction macro spécifique.

**Remarque :** IBM Campaign ne prend pas en charge les calculs simultanés sur plusieurs zones ou sur un sous-ensemble de lignes, contrairement à IBM PredictiveInsight.

Certains autres noms de paramètre utilisent également le même format que data. Les descriptions de ces paramètres font référence à cette section et ce format.

### **Mot-clé**

Le paramètre keyword contrôle le comportement de la fonction macro. Il indique qu'un mot clé peut être spécifié (s'il est omis, la valeur par défaut est utilisée). Les options de mot clé sont répertoriées pour chaque fonction macro sous la forme suivante :

**{choice1 | choice2 | choice3}**

Sélectionnez l'option de mot clé offrant le comportement souhaité. L'option par défaut est indiquée en gras. Par exemple, pour les options suivantes :

**{RADIANs | DEGREEs}**

Les fonctions macro suivantes sont toutes deux valides :

`COS(V1, RADIANs) COS(V1, DEGREEs)`

**Remarque :** De nombreuses fonctions macro utilisent les paramètres de mot clé {ALL | COL | ROW}. Ces mots clés ne s'appliquent pas dans IBM Campaign car les données d'entrées correspondent toujours à une colonne ou à une zone. La macro se comporte toujours comme si le mot clé COL a été spécifié. Par conséquent, vous n'avez pas besoin de spécifier ces mots clés lorsque vous utilisez IBM Campaign.

## Utilisation des constantes

La plupart des paramètres de fonction macro acceptent des constantes numériques ou des expressions converties en constante numérique (les fonctions macro utilisant des chaînes acceptent les constantes de chaîne).

Dans les fonctions macro effectuant des opérations enregistrement par enregistrement (par exemple, l'ajout de deux colonnes numériques), l'utilisation d'une constante revient à spécifier une colonne contenant cette valeur de constante dans chaque ligne. En fait, si une constante est fournie comme paramètre d'entrée, cette constante est étendue à la même longueur que l'entrée.

Certaines fonctions macro acceptent des chaînes de texte ASCII et des constantes numériques. Les paramètres acceptant à la fois des constantes numériques et des chaînes de texte ASCII sont indiqués dans la section "Paramètres" de chaque fonction macro.

Des exemples sont fournis dans le tableau ci-après.

Définition de la fonction	Interprétation de la constante
PERCENT_UTILIZ = (CURR_BAL*100)/ CREDIT_LIM	La constante 100 est interprétée comme une colonne contenant le même nombre de lignes que la colonne CURR_BAL, chaque ligne contenant la constante 100. La zone dérivée PERCENT_UTILIZ contiendra chaque valeur de CURR_BAL multipliée par 100 et divisée par chaque valeur de CREDIT_LIM.
NAME = STRING_CONCAT ("Mr. ", LAST_NAME)	La constante "Mr. " est interprétée comme une colonne contenant le même nombre de lignes que la colonne LAST_NAME, chaque ligne contenant la constante "Mr. ". La zone dérivée NAME contiendra chacune des chaînes de texte de LAST_NAME précédée de "Mr. ".



---


## Chapitre 2. Utilisation de macros dans IBM Interact

Cette section fournit des informations d'utilisation sur les macros pouvant être utilisées dans IBM Interact. Tous les utilisateurs IBM Interact doivent lire cette section avant de consulter le reste du présent document.

---

### Récapitulatif des fonctions macro

Les tableaux de cette section récapitulent les fonctions macro en fonction des catégories suivantes :

**Important :** Toutes les macros figurant dans ce document sont disponibles dans IBM Interact. Les macros disponibles uniquement dans IBM PredictiveInsight sont désignées par cette icône : 

Des pages de référence détaillées sont fournies pour chaque macro, dans l'ordre alphabétique, à partir de la rubrique Chapitre 4, «Guide de référence des macros IBM », à la page 33. La rubrique «Paramètres des fonctions macro», à la page 27 fournit des informations sur les paramètres en entrée de la fonction macro.

**Information associée:**

«Fonctions statistiques»

«Fonctions mathématiques et trigonométriques», à la page 12

«Fonctions des chaînes de caractères», à la page 13

«Fonctions d'horodatage», à la page 14

«Fonctions diverses», à la page 14

### Fonctions statistiques

Nom de la macro	Renvoie	Description
AVG	Valeur unique dans une nouvelle colonne pour le mot clé ALL ; une colonne avec une valeur unique par colonne d'entrée pour le mot clé COL ; une colonne avec une valeur par ligne pour le mot clé ROW.	Calcule la moyenne arithmétique d'une plage de cibles
MAX	Valeur unique dans une nouvelle colonne pour le mot clé ALL ; une colonne avec une valeur unique par colonne d'entrée pour le mot clé COL ; une colonne avec une valeur par ligne pour le mot clé ROW.	Calcule la valeur maximale d'une plage de cibles

Nom de la macro	Renvoie	Description
MEAN	Valeur unique dans une nouvelle colonne pour le mot clé ALL ; une colonne avec une valeur unique par colonne d'entrée pour le mot clé COL ; une colonne avec une valeur par ligne pour le mot clé ROW.	Calcule la moyenne arithmétique d'une plage de cibles
MIN	Valeur unique dans une nouvelle colonne pour le mot clé ALL ; une colonne avec une valeur unique par colonne d'entrée pour le mot clé COL ; une colonne avec une valeur par ligne pour le mot clé ROW.	Calcule la valeur minimale d'une plage de cibles
STDV ou STDEV	Valeur unique dans une nouvelle colonne pour le mot clé ALL ; une colonne avec une valeur unique par colonne d'entrée pour le mot clé COL ; une colonne avec une valeur par ligne pour le mot clé ROW.	Calcule l'écart type d'une plage de cibles

## Fonctions mathématiques et trigonométriques

Nom de la macro	Renvoie	Description
AVG	Une colonne par colonne d'entrée	Calcule la moyenne arithmétique des cibles de la plage de données spécifiée
MAX	Valeur unique dans une nouvelle colonne pour le mot clé ALL ; une colonne avec une valeur unique par colonne d'entrée pour le mot clé COL ; une colonne avec une valeur par ligne pour le mot clé ROW.	Calcule la valeur maximale d'une plage de cibles
MEAN	Valeur unique dans une nouvelle colonne pour le mot clé ALL ; une colonne avec une valeur unique par colonne d'entrée pour le mot clé COL ; une colonne avec une valeur par ligne pour le mot clé ROW.	Calcule la moyenne arithmétique d'une plage de cibles
MIN	Valeur unique dans une nouvelle colonne pour le mot clé ALL ; une colonne avec une valeur unique par colonne d'entrée pour le mot clé COL ; une colonne avec une valeur par ligne pour le mot clé ROW.	Calcule la valeur minimale d'une plage de cibles

Nom de la macro	Renvoie	Description
STDV ou STDEV	Valeur unique dans une nouvelle colonne pour le mot clé ALL ; une colonne avec une valeur unique par colonne d'entrée pour le mot clé COL ; une colonne avec une valeur par ligne pour le mot clé ROW.	Calcule l'écart type d'une plage de cibles
SUM	Valeur unique dans une nouvelle colonne pour le mot clé ALL ; une colonne avec une valeur unique par colonne d'entrée pour le mot clé COL ; une colonne avec une valeur par ligne pour le mot clé ROW.	Calcule la somme d'une plage de cibles
TOTAL	Valeur unique dans une nouvelle colonne pour le mot clé ALL ; une colonne avec une valeur unique par colonne d'entrée pour le mot clé COL ; une colonne avec une valeur par ligne pour le mot clé ROW.	Calcule la somme d'une plage de cibles

## Fonctions des chaînes de caractères

Nom de la macro	Renvoie	Description
LIKE	Une colonne par colonne d'entrée	Détermine si une chaîne de texte correspond à un modèle spécifié
LOWER	Une colonne par colonne d'entrée	Convertit une valeur de chaîne en minuscules
LTRIM	Une colonne par colonne d'entrée	Supprime les espaces de début de chaque valeur de chaîne
NUMBER	Une colonne par colonne d'entrée	Convertit les chaînes de texte ASCII des heures et des dates en valeurs numériques
RTRIM	Une colonne par colonne d'entrée	Supprime les espaces de fin de chaque valeur de chaîne
STRING_CONCAT	Une colonne avec une valeur pour chaque ligne de la colonne d'entrée la plus courte	Concatène les chaînes des plages de données spécifiées
SUBSTR ou SUBSTRING	Une colonne par colonne d'entrée	Renvoie les caractères d'une chaîne à partir d'une position initiale
UPPER	Une colonne par colonne d'entrée	Convertit une valeur de chaîne en majuscules

## Fonctions d'horodatage

Nom de la macro	Renvoie	Description
ADD_MONTHS	Une colonne par colonne d'entrée	Renvoie une date après l'ajout du nombre de mois spécifié
CURRENT_DATE	Une colonne par colonne d'entrée	Renvoie la date du jour dans le format format
CURRENT_DAY	Une colonne par colonne d'entrée	Renvoie le jour du mois en cours sous la forme d'un nombre compris entre 1 et 31
CURRENT_MONTH	Une colonne par colonne d'entrée	Renvoie le mois en cours de l'année sous la forme d'un nombre compris entre 1 et 12
CURRENT_WEEKDAY	Une colonne par colonne d'entrée	Renvoie le jour de la semaine en cours du mois sous la forme d'un nombre compris entre 0 et 6
CURRENT_YEAR	Une colonne par colonne d'entrée	Renvoie l'année en cours sous la forme d'un nombre
DATE	Une colonne par colonne d'entrée	Convertit une chaîne de date en date du calendrier Julien
DATE_FORMAT	Une colonne par colonne d'entrée	Convertit les formats de date

## Fonctions diverses

Nom de la macro	Renvoie	Description
EXTERNALCALLOUT	Valeurs telles que définies par l'application personnalisée écrite à l'aide de l'API ExternalCallout	Appelle une application personnalisée écrite à l'aide de l'API ExternalCallout.  Pour plus d'informations, voir le Guide du développeur d' <i>IBM Interact</i> .
IF	Une colonne avec une valeur pour chaque ligne de la colonne d'entrée la plus courte	Commence une instruction conditionnelle if-then-else

---

## Paramètres des fonctions macro

Cette section décrit les paramètres et l'utilisation des fonctions macro dans IBM Interact.

## Spécifications de format

Cette section décrit le format de certains paramètres fréquemment utilisés. Elle s'applique à toutes les références à ces paramètres par les spécifications de fonction macro de cette section.

## data

Le paramètre data représente une colonne de données sur laquelle une fonction macro doit agir. Il peut s'agir d'une constante ou d'une zone. Pour plus d'informations, reportez-vous à la fonction macro spécifique.

**Remarque :** IBM Interact ne prend pas en charge les calculs simultanés sur plusieurs zones ou sur un sous-ensemble de lignes, contrairement à IBM PredictiveInsight.

Certains autres noms de paramètre utilisent également le même format que data. La description de ces paramètres fait référence à cette section et ce format.

## Mot-clé

Le paramètre keyword contrôle le comportement de la fonction macro. Il indique qu'un mot clé peut être spécifié (s'il est omis, la valeur par défaut est utilisée). Les options de mot clé sont répertoriées pour chaque fonction macro sous la forme suivante :

```
{ choice1 | choice2 | choice3 }
```

Sélectionnez l'option de mot clé offrant le comportement souhaité. L'option par défaut est indiquée en gras. Par exemple, pour les options suivantes :

```
{ RADIANS | DEGREES }
```

Les fonctions macro suivantes sont toutes deux valides :

```
COS(V1, RADIANS) COS(V1, DEGREES)
```

**Remarque :** De nombreuses fonctions de macro utilisent les paramètres de mot clé {ALL | COL | ROW}. Ces mots clés ne s'appliquent pas dans IBM Interact car les données d'entrées correspondent toujours à une colonne ou à une zone. La macro se comporte toujours comme si le mot clé COL a été spécifié. Par conséquent, vous n'avez pas besoin de spécifier ces mots clés lorsque vous utilisez IBM Interact.

## Utilisation des constantes

La plupart des paramètres de fonction macro acceptent des constantes numériques ou des expressions converties en constante numérique (les fonctions macro utilisant des chaînes acceptent les constantes de chaîne).

Dans les fonctions macro effectuant des opérations enregistrement par enregistrement (par exemple, l'ajout de deux colonnes numériques), l'utilisation d'une constante revient à spécifier une colonne contenant cette valeur de constante dans chaque ligne. En fait, si une constante est fournie comme paramètre d'entrée, cette constante est étendue à la même longueur que l'entrée.

Certaines fonctions macro acceptent aussi bien des chaînes de texte ASCII que des constantes numériques. Les paramètres acceptant à la fois des constantes numériques et des chaînes de texte ASCII sont indiqués dans la section "Paramètres" de chaque fonction macro.

Des exemples sont fournis dans le tableau ci-après.

Définition de la fonction	Interprétation de la constante
PERCENT_UTILIZ = (CURR_BAL*100)/ CREDIT_LIM	La constante 100 est interprétée comme une colonne contenant le même nombre de lignes que la colonne CURR_BAL, chaque ligne contenant la constante 100. La zone dérivée PERCENT_UTILIZ contiendra chaque valeur de CURR_BAL multipliée par 100 et divisée par chaque valeur de CREDIT_LIM.
NAME = STRING_CONCAT ("Mr.", LAST_NAME)	La constante "Mr." est interprétée comme une colonne contenant le même nombre de lignes que la colonne LAST_NAME, chaque ligne contenant la constante "Mr.". La zone dérivée NAME contiendra chacune des chaînes de texte de LAST_NAME précédée de "Mr."

**Remarque :** Les constantes, telles que DT\_DELIM\_M\_D\_Y, requièrent des apostrophes.

---


## Chapitre 3. Utilisation de macros dans IBM PredictiveInsight

Ce chapitre fournit des informations d'utilisation sur les macros pouvant être utilisées dans IBM PredictiveInsight. Tous les utilisateurs IBM PredictiveInsight doivent lire ce chapitre avant de consulter le reste du présent document.

---

### Récapitulatif des fonctions macro

Les tableaux de cette section récapitulent les fonctions macro en fonction des catégories suivantes :

**Important :** Toutes les macros figurant dans ce document sont disponibles dans IBM PredictiveInsight. Les macros disponibles uniquement dans IBM Interact sont désignées par cette icône : 

Des pages de référence détaillées sont fournies pour chaque macro, dans l'ordre alphabétique, à partir de la rubrique Chapitre 4, «Guide de référence des macros IBM », à la page 33. La rubrique «Paramètres des fonctions macro», à la page 27 fournit des informations sur les paramètres en entrée de la fonction macro.

#### Information associée:

«Fonctions statistiques»

«Fonctions mathématiques et trigonométriques», à la page 19

«Fonctions d'ingénierie», à la page 22

«Fonctions des chaînes de caractères», à la page 23

«Fonctions d'horodatage», à la page 24

«Fonctions de groupement», à la page 8

### Fonctions statistiques

Nom de la macro	Renvoie	Description
AVG	Valeur unique dans une nouvelle colonne pour le mot clé ALL ; une colonne avec une valeur unique par colonne d'entrée pour le mot clé COL ; une colonne avec une valeur par ligne pour le mot clé ROW.	Calcule la moyenne arithmétique d'une plage de cibles
AVG_DEV	Valeur unique dans une nouvelle colonne pour le mot clé ALL ; une colonne avec une valeur unique par colonne d'entrée pour le mot clé COL ; une colonne avec une valeur par ligne pour le mot clé ROW.	Calcule l'écart moyen d'une plage de cibles
HISTOGRAM	Valeur unique dans une nouvelle colonne	Calcule l'histogramme d'une plage de données spécifiée à l'aide des limites d'ensemble fournies

Nom de la macro	Renvoie	Description
KURTOSIS	Valeur unique dans une nouvelle colonne pour le mot clé ALL ; une colonne avec une valeur unique par colonne d'entrée pour le mot clé COL ; une colonne avec une valeur par ligne pour le mot clé ROW.	Calcule le kurtosis d'une plage de cibles
MEAN	Valeur unique dans une nouvelle colonne pour le mot clé ALL ; une colonne avec une valeur unique par colonne d'entrée pour le mot clé COL ; une colonne avec une valeur par ligne pour le mot clé ROW.	Calcule la moyenne arithmétique d'une plage de cibles
SKEW	Valeur unique dans une nouvelle colonne pour le mot clé ALL ; une colonne avec une valeur unique par colonne d'entrée pour le mot clé COL ; une colonne avec une valeur par ligne pour le mot clé ROW.	Calcule l'asymétrie de la distribution d'une plage de cibles
STAT	Nombre variable de colonnes (voir macro)	Calcule les premier, deuxième, troisième et quatrième moments de la plage de données spécifiée
STDV ou STDEV	Valeur unique dans une nouvelle colonne pour le mot clé ALL ; une colonne avec une valeur unique par colonne d'entrée pour le mot clé COL ; une colonne avec une valeur par ligne pour le mot clé ROW.	Calcule l'écart type d'une plage de cibles
VARIANCE	Valeur unique dans une nouvelle colonne pour le mot clé ALL ; une colonne avec une valeur unique par colonne d'entrée pour le mot clé COL ; une colonne avec une valeur par ligne pour le mot clé ROW.	Calcule la variance d'une plage de cibles
XTAB	Une colonne pour chaque valeur distincte dans le deuxième paramètre d'entrée avec un nombre de lignes égal au nombre de valeurs distinctes dans le premier paramètre d'entrée	Calcule les valeurs distinctes dans deux colonnes et renvoie la valeur COUNTOF de chaque combinaison



## Fonctions mathématiques et trigonométriques

Nom de la macro	Renvoi	Description
ABS	Une colonne par colonne d'entrée	Calcule la valeur absolue du contenu de la plage de données spécifiée
ACOS	Une colonne par colonne d'entrée	Calcule le cosinus inverse du contenu de la plage de données spécifiée
ACOT	Une colonne par colonne d'entrée	Calcule l'arc cotangente du contenu de la plage de données spécifiée
ASIN	Une colonne par colonne d'entrée	Calcule l'arc sinus du contenu de la plage de données spécifiée
ATAN	Une colonne par colonne d'entrée	Calcule l'arc tangente du contenu de la plage de données spécifiée
CEILING	Une colonne par colonne d'entrée	Calcule le plafond de chaque valeur de la plage de données spécifiée
COS	Une colonne par colonne d'entrée	Calcule le cosinus du contenu de la plage de données spécifiée
COSH	Une colonne par colonne d'entrée	Calcule le cosinus hyperbolique du contenu de la plage de données spécifiée
COT	Une colonne par colonne d'entrée	Calcule la cotangente du contenu de la plage de données spécifiée
COUNT	Une colonne contenant une valeur unique	Compte le nombre de cibles contenant des valeurs dans la plage de données spécifiée
COV	Valeur unique dans une ou plusieurs colonnes	Calcule la covariance de deux plages en entrée
DERIVATIVE	Une colonne par colonne d'entrée	Calcule la dérivée des valeurs de la plage de données spécifiée
DIV	Une colonne par colonne d'entrée	Divise une plage de données spécifiée par une autre
EQ	Une colonne par colonne d'entrée	Renvoie TRUE si une plage de données est égale à une autre
EXP	Une colonne par colonne d'entrée	Calcule le nombre naturel (e) élevé à la puissance du contenu de chaque cible de la plage de données spécifiée
FACTORIAL	Une colonne par colonne d'entrée	Calcule la factorielle de chaque valeur de la plage de données spécifiée
FLOOR	Une colonne par colonne d'entrée	Calcule la partie entière de chaque valeur de la plage de données spécifiée

Nom de la macro	Renvoie	Description
FRACTION	Une colonne par colonne d'entrée	Renvoie la partie décimale de chaque valeur de la plage de données spécifiée
GAUSS	Une colonne par colonne d'entrée	Calcule la valeur de Gauss des valeurs de la plage de données spécifiée
GAUSS_AREA	Une colonne par colonne d'entrée	Calcule la zone sous la valeur de Gauss des valeurs de la plage de données spécifiée
GE	Une colonne par colonne d'entrée	Renvoie TRUE si une plage de données est supérieure ou égale à une autre
GT	Une colonne par colonne d'entrée	Renvoie TRUE si une plage de données est supérieure à une autre
INT	Une colonne par colonne d'entrée	Calcule la valeur entière (arrondie) du contenu de la plage de données spécifiée
INTEGRAL	Une colonne par colonne d'entrée	Calcule l'intégrale des valeurs de la plage de données spécifiée
INVERSE	Une colonne par colonne d'entrée	Calcule la valeur négative du contenu de la plage de données spécifiée
ISEVEN	Une colonne par colonne d'entrée	Teste si les valeurs en entrée sont paires (à savoir, divisible par deux)
ISODD	Une colonne par colonne d'entrée	Teste si les valeurs en entrée sont impaires (à savoir, non divisible par deux)
LE	Une colonne par colonne d'entrée	Renvoie TRUE si une plage de données est inférieure ou égale à une autre
LN	Une colonne par colonne d'entrée	Calcule le logarithme naturel du contenu de la plage de données spécifiée
LOG	Une colonne par colonne d'entrée	Calcule le logarithme naturel du contenu de la plage de données spécifiée
LOG2	Une colonne par colonne d'entrée	Calcule le logarithme base 2 du contenu de la plage de données spécifiée
LOG10	Une colonne par colonne d'entrée	Calcule le logarithme base 10 du contenu de la plage de données spécifiée
LT	Une colonne par colonne d'entrée	Renvoie TRUE si une plage de données est inférieure à une autre

Nom de la macro	Renvoi	Description
MAX	Valeur unique dans une nouvelle colonne pour le mot clé ALL ; une colonne avec une valeur unique par colonne d'entrée pour le mot clé COL ; une colonne avec une valeur par ligne pour le mot clé ROW.	Calcule la valeur maximale d'une plage de cibles
MIN	Valeur unique dans une nouvelle colonne pour le mot clé ALL ; une colonne avec une valeur unique par colonne d'entrée pour le mot clé COL ; une colonne avec une valeur par ligne pour le mot clé ROW.	Calcule la valeur minimale d'une plage de cibles
MINUS	Une colonne par colonne d'entrée	Soustrait une plage de données spécifiée d'une autre
MOD	Une colonne par colonne d'entrée	Calcule le modulo du contenu de la plage de données spécifiée
MULT	Une colonne par colonne d'entrée	Multiplie le contenu de deux plages de données
NE	Une colonne par colonne d'entrée	Renvoie TRUE si une plage de données n'est pas égale à une autre
PLUS	Une colonne par colonne d'entrée	Ajoute le contenu de deux plages de données
POW	Une colonne par colonne d'entrée	Calcule une valeur de base élevée aux puissances exponentielles spécifiées
RANDOM	Une colonne avec le nombre de valeurs spécifié	Renvoie le nombre spécifié de nombres aléatoires
RANDOM_GAUSS	Une colonne avec le nombre de valeurs spécifié	Renvoie le nombre spécifié de valeurs aléatoires à partir d'une distribution de Gauss
ROUND	Une colonne par colonne d'entrée	Calcule la valeur arrondie du contenu de la plage de données spécifiée
SIGN	Une colonne par colonne d'entrée	Détermine le signe (positif ou négatif) des valeurs dans la plage de données spécifiée
SIN	Une colonne par colonne d'entrée	Calcule le sinus du contenu de la plage de données spécifiée
SINH	Une colonne par colonne d'entrée	Calcule le sinus hyperbolique du contenu de la plage de données spécifiée
SQRT	Une colonne par colonne d'entrée	Calcule la racine carrée du contenu de la plage de données spécifiée

Nom de la macro	Renvoie	Description
SUM	Valeur unique dans une nouvelle colonne pour le mot clé ALL ; une colonne avec une valeur unique par colonne d'entrée pour le mot clé COL ; une colonne avec une valeur par ligne pour le mot clé ROW.	Calcule la somme d'une plage de cibles
TAN	Une colonne par colonne d'entrée	Calcule la tangente du contenu de la plage de données spécifiée
TANH	Une colonne par colonne d'entrée	Calcule la tangente hyperbolique du contenu de la plage de données spécifiée
TOTAL	Valeur unique dans une nouvelle colonne pour le mot clé ALL ; une colonne avec une valeur unique par colonne d'entrée pour le mot clé COL ; une colonne avec une valeur par ligne pour le mot clé ROW.	Calcule la somme d'une plage de cibles
TRUNCATE	Une colonne par colonne d'entrée	Renvoie la partie non décimale de chaque valeur de la plage de données spécifiée

## Fonctions d'ingénierie

Nom de la macro	Renvoie	Fonction
DELAY	Une colonne par colonne d'entrée	Renvoie les valeurs des colonnes en entrée retardées du nombre d'intervalles de temps spécifié
GRID	Une colonne par colonne d'entrée	Renvoie une grille de toutes les combinaisons de valeurs possibles (une par ligne)
LAG	Une colonne par colonne d'entrée	Renvoie les valeurs des colonnes en entrée décalées du nombre d'intervalles de temps spécifié
NORM_MAXMIN	Une colonne par colonne d'entrée	Calcule la normalisation min/max d'une plage de données
NORM_SIGMOID	Une colonne par colonne d'entrée	Calcule la normalisation sigmoïde d'une plage de données
NORM_ZSCORE	Une colonne par colonne d'entrée	Calcule la normalisation z-score d'une plage de données

Nom de la macro	Renvoie	Fonction
PCA	Une colonne par colonne d'entrée, plus un	Calcule les vecteurs propres des composants principaux de la plage de données spécifiée
PCA_FEATURES	Une colonne par colonne d'entrée, plus un	Extrait $n$ fonctions de la plage de données spécifiée
SAMPLE_RANDOM	Une colonne par colonne d'entrée	Renvoie des colonnes de $n$ cibles, chacune contenant un échantillon aléatoire de la plage de données spécifiée
SLIDE_WINDOW	Nombre de colonnes en entrée fois le paramètre width	Crée un modèle à partir d'une fenêtre spécifiée et le fait glisser pour créer le modèle suivant
SORT	Une colonne par colonne d'entrée	Trie les valeurs de la plage de données spécifiée dans l'ordre croissant ou décroissant
SUBSAMPLE	Une colonne par colonne d'entrée	Réduit la quantité de données en renvoyant toutes les $ne$ valeurs de ligne

## Fonctions des chaînes de caractères

Nom de la macro	Renvoie	Fonction
DISTINCT	Une colonne par colonne d'entrée	
FORMAT	Une colonne par colonne d'entrée	Fournit un contrôle de formatage en sortie pour les nombres et les chaînes (par exemple, la largeur de la sortie, l'alignement, la précision numérique, le symbole du séparateur décimal, le symbole de groupement, etc.). Renvoie la chaîne en sortie formatée.
LIKE	Une colonne par colonne d'entrée	Détermine si une chaîne de texte correspond à un modèle spécifié
NUMBER	Une colonne par colonne d'entrée	Convertit les chaînes de texte ASCII des heures et des dates en valeurs numériques
POSITION	Une colonne par colonne d'entrée	Renvoie la position initiale d'un modèle dans une chaîne de texte
STRING_CONCAT	Une colonne avec une valeur pour chaque ligne de la colonne d'entrée la plus courte	Concatène les chaînes de texte des plages de données spécifiées

Nom de la macro	Renvoie	Fonction
STRING_HEAD	Une colonne par colonne d'entrée	Renvoie les <i>n</i> premiers caractères de chaque chaîne de la plage de données spécifiée
STRING_LENGTH	Une colonne par colonne d'entrée	Renvoie la longueur de chaque chaîne de la plage de données spécifiée
STRING_SEG	Une colonne par colonne d'entrée	Renvoie le segment de chaîne entre deux index spécifiés
STRING_TAIL	Une colonne par colonne d'entrée	Renvoie les <i>n</i> derniers caractères de chaque chaîne de la plage de données spécifiée
SUBSTR	Une colonne par colonne d'entrée	Renvoie les caractères d'une chaîne à partir d'une position initiale
SUBSTRING	Une colonne par colonne d'entrée	Renvoie les caractères d'une chaîne à partir d'une position initiale

## Fonctions d'horodatage

Nom de la macro	Renvoie	Description
ADD_MONTHS	Une colonne par colonne d'entrée	Renvoie une date après l'ajout du nombre de mois spécifié
CURRENT_DATE	Une colonne par colonne d'entrée	Renvoie la date du jour dans le format format
CURRENT_DAY	Une colonne par colonne d'entrée	Renvoie le jour du mois en cours sous la forme d'un nombre compris entre 1 et 31
CURRENT_JULIAN	Une colonne par colonne d'entrée	Renvoie le nombre au format Julien de la date du jour
CURRENT_MONTH	Une colonne par colonne d'entrée	Renvoie le mois en cours de l'année sous la forme d'un nombre compris entre 1 et 12
CURRENT_TIME	Une colonne par colonne d'entrée	Renvoie l'heure actuelle sous la forme d'une chaîne
CURRENT_WEEKDAY	Une colonne par colonne d'entrée	Renvoie le jour de la semaine en cours du mois sous la forme d'un nombre compris entre 0 et 6
CURRENT_YEAR	Une colonne par colonne d'entrée	Renvoie l'année en cours sous la forme d'un nombre
DATE	Une colonne par colonne d'entrée	Convertit une chaîne de date en date du calendrier Julien
DATE_FORMAT	Une colonne par colonne d'entrée	Convertit les formats de date

Nom de la macro	Renvoie	Description
MONTHOF	Une colonne par colonne d'entrée	Renvoie le mois de l'année sous la forme d'un nombre
WEEKDAY	Une colonne par colonne d'entrée	Convertit les chaînes de date textuelles ASCII en jour de la semaine
WEEKDAYOF	Une colonne par colonne d'entrée	Renvoie le jour de la semaine de la semaine sous forme de nombre
YEAROF	Une colonne par colonne d'entrée	Renvoie l'année sous la forme d'un nombre

## Fonctions de groupement

Nom de la macro	Renvoie	Description
GROUPBY	Une colonne avec une valeur pour chaque ligne	Récapitule plusieurs lignes de données dans un groupe
GROUPBY_WHERE	Une colonne avec une valeur pour chaque ligne	Récapitule plusieurs lignes de données répondant à une condition indiquée et se trouvant dans un groupe

## Fonctions diverses

Nom de la macro	Renvoie	Fonction
BUFFER	Une colonne par colonne d'entrée	Copie la plage de données d'entrée et la met à jour de manière dynamique
COLUMN	Une colonne par colonne d'entrée	Crée des colonnes, en concaténant verticalement les valeurs en entrée dans chaque colonne
CONSTANT	Une colonne par colonne d'entrée	Copie la plage de données d'entrée une fois (pas de mise à jour dynamique)
COUNT_DIFF	Deux colonnes	Renvoie chaque valeur unique de l'entrée avec le nombre de fois que la valeur est apparue
CV_FOLDS	Une colonne avec une valeur pour chaque ligne de la colonne d'entrée la plus courte	Divise les modèles d'entrée en $n$ plis de données pour la validation croisée
DATALINK	Une colonne par colonne de données liées	Crée un lien interne avec les données d'une feuille de calcul IBM PredictiveInsight
DDELINK	Une colonne par colonne de données liées	Crée un lien externe avec des données d'une autre application Windows

Nom de la macro	Renvoi	Fonction
DECIMATE	MAX_VALUE colonnes (une colonne par valeur en sortie)	Décime une colonne de nombres en plusieurs colonnes dans lesquelles un un indique la valeur d'index
EXTRACT	Une colonne par colonne d'entrée	Extrait les lignes contenant les valeurs d'une colonne de prédicat
IF	Une colonne avec une valeur pour chaque ligne de la colonne d'entrée la plus courte	Commence une instruction conditionnelle if-then-else
INIT	Une colonne	Initialise les valeurs d'intervalle de temps précédentes pour une fonction récursive
ISERROR	Une colonne avec une valeur pour chaque ligne de la colonne d'entrée la plus courte	Renvoie un un si une valeur de la ligne en entrée contient une cible erronée (???) ; renvoie zéro dans le cas contraire
ISMEMBER	Une colonne par colonne d'entrée	Teste une plage en entrée par rapport à une "table" de valeurs et renvoie un si une valeur se trouve dans la table et zéro dans le cas contraire
MAXINDEX	Une colonne avec une valeur pour chaque ligne de la colonne d'entrée la plus courte	Renvoie l'index de colonne de la n <sup>ème</sup> (première, deuxième, troisième, and.) valeur maximale de chaque ligne de la colonne spécifiée
RANK	Une colonne par colonne d'entrée	Divise les données en groupes nbins (10 par défaut), chacun contenant approximativement un nombre égal de valeurs distinctes, et renvoie le groupe auquel chaque ligne de données appartient.
SORT	Une colonne par colonne d'entrée	Trie la plage de données spécifiée dans l'ordre croissant ou décroissant

## Remarques sur les pages de références des macros

Chacune des fonctions macro disponibles est décrite dans la section *Récapitulatif des fonctions macro* de chaque produit. Avant de travailler avec une macro, consultez les conventions de syntaxe dans la préface.

Des exemples simples sont fournis pour chaque fonction macro. Dans ces exemples, les colonnes supplémentaires créées sont nommées VX, VY, VZ, etc. Les noms réels utilisés dans le tableur dépendent de votre situation spécifique.



---

## Paramètres des fonctions macro

Cette section décrit les paramètres et l'utilisation des fonctions macro dans IBM PredictiveInsight.

### Spécifications de format

Cette section décrit le format de certains paramètres fréquemment utilisés. Elle s'applique à toutes les références à ces paramètres par les spécifications de fonction macro de ce chapitre.

#### data

Le paramètre data représente une plage de données sur laquelle une fonction macro doit agir. Il s'agit généralement d'une constante, d'une colonne ou d'une plage de cibles (voir la fonction macro spécifique pour plus de détails). Le format du paramètre data est le suivant :

```
begin_data [: end_data]
```

Où begin\_data peut être une constante (par exemple, 10.2), le nom d'une colonne (par exemple, V1) ou une plage de cibles (par exemple, V1[1:100]). Le paramètre end\_data est facultatif. S'il est fourni, begin\_data est utilisé comme point de départ et doit être une colonne ou une plage de cibles. Le point final est spécifié par end\_data.

**Remarque :** Certains autres noms de paramètre utilisent également le même format que data. La description de ces paramètres fait référence à cette section et ce format.

#### Mot-clé

Le paramètre keyword contrôle le comportement de la fonction macro. Il indique qu'un mot clé peut être spécifié (s'il est omis, la valeur par défaut est utilisée). Les options de mot clé sont répertoriées pour chaque fonction macro sous la forme suivante :

```
{          choice1 | choice2 | choice3}
```

Sélectionnez l'option de mot clé offrant le comportement souhaité. L'option par défaut est indiquée en gras. Par exemple, pour le format suivant :

```
{          ALL | COL | ROW}
```

Les fonctions macro suivantes sont toutes valides :

```
AVG(V1:V5)  AVG(V1:V5, ALL)  AVG(V1:V5, COL)  AVG(V1:V5, ROW)
```

### Utilisation des plages de cibles et de colonnes

Des plages de cibles et de colonnes peuvent être fournies comme entrées pour la plupart des paramètres des fonctions macro. Elles doivent respecter les règles suivantes :

- Le nombre de colonnes doit correspondre
- Les valeurs renvoyées commencent dans la première cible
- Les plages de cibles sont automatiquement remplies de zéros

## Le nombre de colonnes doit correspondre

Si plusieurs plages de données sont fournies en entrée et que des calculs sur des colonnes sont effectués, les deux plages de données doivent contenir le même nombre de colonnes. Sinon, seules les dimensions de la plage de données la plus petite sont utilisées (certaines fonctions macro signaleront une erreur). Si les plages de données contiennent des nombres de lignes différents, la plupart des fonctions macro effectuent les calculs jusqu'à la dernière ligne incluse de la colonne la plus courte.

- Par exemple, avec des plages de colonnes, la définition de macro  $V6 = V1:V3 \text{ AND } V4:V6$  génère trois colonnes en sortie (les deux plages de données contiennent trois colonnes). Le AND de la colonne V1 avec la colonne V4 est calculé ; le AND de la colonne V2 avec la colonne V5 est calculé et le AND de la colonne V3 avec la colonne V6 est calculé. Toutefois,  $V6 = V1:V3 \text{ AND } V4:V5$  ne renvoie que deux colonnes en sortie (la première plage de colonnes contient trois colonnes, la deuxième n'en contient que deux et la plus courte des deux est utilisée). Dans ce cas, le AND de la colonne V1 avec la colonne V4 le AND de la colonne V2 avec la colonne V5 sont calculés. La colonne V3 n'est pas utilisée.
- Avec les plages de cibles, la définition de macro  $V7 = V1[1:5]:V2 \text{ AND } V4[10:50]:V5$  génère deux colonnes en sortie (les deux plages en entrée contiennent deux colonnes). Les colonnes en sortie V7 et V8 contiennent cinq valeurs (AND des cibles 1 à 5 avec les cibles 10 à 14). La définition de macro  $V7 = V1[1:5]:V2 \text{ AND } V4$  ne génère qu'une colonne en sortie car la deuxième plage de données ne contient qu'une colonne.

**Important :** La spécification d'une colonne sans plage de cibles revient à spécifier l'intégralité de la colonne (à savoir, une plage de cibles d'une cible pour toute la longueur de la colonne).

## Les valeurs renvoyées commencent dans la première cible

Toutes les valeurs renvoyées par une fonction macro sont placées dans des cibles consécutives, en commençant par la première cible (par exemple,  $\text{TEMP}[1]$ ). Par exemple,  $V2=\text{SIN}(V1[100:200])$  place 101 valeurs dans les cibles 1 à 100 de la colonne V2.

**Important :** Si vous devez effectuer une opération ligne par ligne sur une plage de cibles et que vous souhaitez conserver les résultats dans les lignes correspondantes (à savoir, si vous agissez sur les cibles  $[10:20]$  et que vous souhaitez que les résultats soient placés dans les cibles 10 à 20 de la colonne résultante), le calcul spécifie la plage de cibles  $[1:20]$  à la place. Des valeurs inutiles sont calculées, mais les résultats sont placés sur les lignes souhaitées.

## Plages de cibles automatiquement remplies avec des zéros

Si vous spécifiez une plage de cibles, les cibles vides de cette plage sont automatiquement remplies de zéros. Par exemple,  $V3 = V1[1:3]*V2$  génère :

V1 V2 V3 1 2 2 3 4 12 [] 6 0

où [] représente une cible vide (la colonne V1 ne contient que deux valeurs de cible). Toutefois, si vous spécifiez  $V3 = V1*V2$ , les valeurs 2 et 12 sont générées (les calculs sont effectués jusqu'à la plus courte des deux colonnes).

## Utilisation des constantes

La plupart des paramètres de fonction macro acceptent des constantes numériques ou des expressions converties en constante numérique (les fonctions macro utilisant des chaînes acceptent les constantes de chaîne). Dans les fonctions macro effectuant

des opérations enregistrement par enregistrement, l'utilisation d'une constante revient à spécifier une colonne contenant cette valeur de constante dans chaque ligne. En fait, si une constante et une plage de cibles ou de colonnes sont fournies comme paramètres en entrée, or column range are provided as input parameters, la constante est étendue aux mêmes dimensions que la plage de cibles ou de colonnes. Toute colonne contenant une cible unique utilisée comme entrée dans une fonction macro est considérée comme une constante.

Certaines fonctions macro acceptent aussi bien des chaînes de texte ASCII que des constantes numériques. Les paramètres acceptant à la fois des constantes numériques et des chaînes de texte ASCII sont indiqués dans la section "Paramètres" de chaque fonction macro.

Des exemples sont fournis dans le tableau ci-après.

Définition de la fonction	Interprétation de la constante
V1=3+5	Chacune des constantes est interprétée comme une colonne unique contenant la valeur unique. La colonne V1 contient la valeur unique 8.
V2=2*V1	La constante 2 est interprétée comme une colonne contenant le même nombre de lignes que la colonne V1, chaque ligne contenant la constante 2. La colonne V2 contient chaque valeur de V1 multipliée par 2.
V2 = STRING_CONCAT(V1, "ing")	La constante "ing" est interprétée comme une colonne contenant le même nombre de lignes que la colonne V1, chaque ligne contenant la constante "ing". La colonne V2 contiendra chacune des chaînes de texte de V1 concaténée avec "ing".
V4=V1:V3/ AVG(V1:V3)	L'expression AVG(V1:V3) est convertie en valeur constante, disons $x$ . La constante $x$ est interprétée comme trois colonnes contenant autant de lignes que la colonne la plus courte entre V1, V2 et V3. Chaque cible contient la constante $x$ . Les colonnes en sortie, V4-V6, contiendront les valeurs des colonnes V1-V3 divisées par $x$ .
V3=V1[10:20]^2	La constante 2 est interprétée comme une plage de cibles avec 11 lignes, chacune contenant la valeur 2.

## Précision de calcul

Tous les calculs dans les feuilles de calcul d'**IBM PredictiveInsight** sont limités à une précision maximale de 32 bits.

### Calculs d'entier

Les fonctions macro effectuant des calculs d'entier (BIT\_AND, BIT\_NOT, BIT\_OR, BIT\_XOR et T0) ne gèrent pas les nombres négatifs. Les valeurs doivent être comprises entre 0 et  $(2^{24} - 1)$  ; sinon, une erreur est renvoyée.

## Résultats de cible non valides et cibles vides

### Cibles contenant des ???

Si une opération de feuille de calcul génère des résultats non valides, la cible contiendra ??? au lieu de la valeur calculée. Par exemple, la racine carrée d'un nombre négatif, calculée à l'aide de la fonction macro SQRT, génère ??? pour chaque valeur négative en entrée.

### Calculs basés sur ???

Si une cible contient une valeur ???, la plupart des calculs utilisant cette cible propagent le résultat ???. Par exemple, la somme d'une colonne contenant une ou plusieurs cibles avec ??? génère ???.

Si le résultat ??? apparaît dans une ou plusieurs cibles, cliquez sur l'une des cibles contenant ??? et lisez le message d'erreur affiché dans la zone de texte **Définition de la fonction**. Reportez-vous à la référence de la fonction macro dans le présent document pour déterminer la raison pour laquelle l'erreur s'est produite. Ensuite, modifiez la définition de fonction de sorte que toutes les valeurs en entrée soient valides. Par exemple, avec l'exemple de la racine carrée, vous pouvez d'abord prendre la valeur absolue des entrées :

$V2 = \text{SQRT}(V1) \text{ --> } V2 = \text{SQRT}(\text{ABS}(V1))$

**Remarque :** Il se peut que vous deviez retracer les valeurs ??? dans des colonnes dépendantes pour rectifier le problème.

**Important :** Les cibles contenant ??? transmises au manager d'expérimentations dans le cadre d'un modèle d'apprentissage ou de test sont transmises comme des zéros.

### Cibles vides et cibles ???

Les cibles vides ne contiennent rien. Elles apparaissent uniquement à la fin d'une colonne ou dans les colonnes vides. Les cibles vides et les cibles ??? sont traitées différemment par les différentes fonctions macro. La plupart des fonctions macro traitent les cibles vides comme des zéros et signalent et propagent les erreurs dans les cibles ??? avec les exceptions suivantes :

- La fonction macro MAXINDEX ignore les cibles vides et les cibles ???.
- La fonction macro OFFSET ignore les cibles vides et les cibles ???.
- La fonction macro DDELINK transmet les cibles vides et les cibles ???.
- Pour les fonctions macro qui ignorent les cibles vides, voir «Utilisation des constantes», à la page 28. Les erreurs dans les cibles ??? sont toujours signalées et propagées.
- Pour les fonctions macro qui ignorent les cibles vides mais utilisent les cibles ??? comme décrit, voir «Fonctions macro qui ignorent les blancs et utilisent des appels ???», à la page 31.
- Pour les fonctions macro qui renvoient ??? pour toutes les cibles si l'une des cibles dans les arguments répertoriés contient ???, voir «Fonctions macro ne pouvant pas traiter les cibles ???», à la page 31.

### Fonctions macro qui ignorent les blancs et propagent les cibles ???

AVG	NORM_MINMAX
DERIVATIVE	NORM_ZSCORE

KURTOSIS	PCA
MAX	PCA_FEATURES
MEAN	SKEW
MIN	SUM
MOMENTS	VARIANCE

### Fonctions macro qui ignorent les blancs et utilisent des appels ???

Fonction macro	Comment les cibles ??? sont utilisées
COUNT	Les cibles ??? sont comptées.
COLUMN	Les cibles ??? sont copiées.
CV_FOLDS	Les points d'interrogation (???) sont traités comme une classe distincte.
DELAY	Les cibles ??? sont copiées.
FREQ	Les points d'interrogation (???) d'une colonne de chaîne sont considérés comme une classe distincte. Dans une colonne numérique, ??? correspond à zéro.
EXTRACT	Les points d'interrogation (???) dans predicate_col sont traités comme des zéros.
LAG	Les cibles ??? sont copiées.
MERGE	Les cibles ??? sont copiées.
SORT	Toutes les cibles ??? sont triées à la fin.
SAMPLE_RANDOM	Les cibles ??? peuvent être échantillonnées.
SELECT	Les cibles ??? sont copiées.
SUBSAMPLE	Les cibles ??? peuvent être échantillonnées.

### Fonctions macro ne pouvant pas traiter les cibles ???

Fonction macro	Argument ne pouvant pas contenir de ???
GRID	col1, col2
HISTOGRAM	data, bin_col
INTEGRAL	data, multiplier
ISERROR	data



---

## Chapitre 4. Guide de référence des macros IBM

Cette section fournit des informations de référence pour les fonctions macro de feuille de calcul disponibles dans la suite de plateforme marketing IBM.

Les macros disponibles sont répertoriées dans les pages qui suivent, par ordre alphabétique. Chaque macro est fournie avec sa syntaxe, les paramètres possibles et des exemples.

**Important :** N'utilisez pas des noms de fonction ou des mots clés du langage des macros IBM pour les en-têtes de colonne des tables utilisateur dans IBM Campaign, que vous effectuiez un mappage à partir d'une base de données ou d'un fichier à plat. Ces mots réservés peuvent provoquer des erreurs s'ils sont utilisés dans les en-têtes de colonne des tables mappées.

---

### ABS

#### Syntaxe

ABS(data)

#### Paramètres

data

Valeurs numériques dont la valeur absolue doit être calculée. Ce paramètre peut correspondre à une valeur constante, une colonne, une plage de cibles ou une expression convertie en l'un de ces types. Pour la définition du format de data, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

ABS calcule la valeur absolue des nombres dans la plage de données spécifiée. La valeur absolue d'un nombre correspond à sa valeur sans son signe (les nombres positifs restent inchangés et les nombres négatifs sont renvoyés sous la forme de nombres positifs). ABS renvoie une nouvelle colonne par colonne d'entrée, chacune contenant la valeur absolue des nombres de la colonne d'entrée correspondante.

#### Exemples

TEMP = ABS(-3) ou TEMP = ABS(3)
Crée une colonne nommée TEMP contenant la valeur trois.
TEMP = ABS(V1)
Crée une colonne nommée TEMP, dans laquelle chaque valeur correspond à la valeur absolue du contenu de la colonne V1.
TEMP = ABS(V1:V3)
Crée trois colonnes nommées TEMP, VX et VY. Les valeurs de la colonne TEMP correspondent aux valeurs absolues du contenu de la colonne V1, les valeurs de la colonne VX correspondent aux valeurs absolues du contenu de la colonne V2 et les valeurs de la colonne VY correspondent aux valeurs absolues du contenu de la colonne V3.

TEMP = ABS(V1[10:20])

Crée une colonne nommée TEMP, dans laquelle les 11 premières cibles contiennent les valeurs absolues des valeurs des lignes 10 à 20 de la colonne V1. Les autres cibles de TEMP sont vides.

TEMP = ABS(V1[1:5]:V2)

Crée deux colonnes nommées TEMP et VX, contenant des valeurs dans les lignes 1 à 5 (les autres cibles sont vides). Les valeurs de la colonne TEMP correspondent aux valeurs absolues des lignes correspondantes de la colonne V1 et les valeurs de la colonne VX correspondent aux valeurs absolues des lignes correspondantes de la colonne V2.

## Fonctions associées

Fonction	Description
SIGN	Détermine le signe (positif ou négatif) des valeurs dans la plage de données spécifiée

---

## ACOS

### Syntaxe

ACOS( data [, units\_keyword ])

### Paramètres

data

Valeurs numériques dont l'arc cosinus doit être calculé. Ce paramètre peut correspondre à une valeur constante, une colonne, une plage de cibles ou une expression convertie en l'un de ces types. Pour la définition du format de data, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

units\_keyword

Ce mot clé facultatif détermine si les valeurs en entrée et les résultats sont interprétés comme des degrés ou des radians. Sélectionnez l'une des valeurs suivantes :

RADIAN - Effectue les calculs en radians (valeur par défaut)

DEGREE - Effectue les calculs en degrés

Si ce paramètre n'est pas spécifié, la valeur par défaut est radians. (Pour convertir des radians en degrés, divisez par PI et multipliez par 180.)

Pour plus de détails sur l'utilisation des mots clés dans IBM Campaign, voir «Spécifications de format», à la page 9.

Pour plus de détails sur l'utilisation des mots clés dans IBM PredictiveInsight, voir «Spécifications de format», à la page 27.



## Description

ACOS calcule l'arc cosinus des valeurs de la plage de données spécifiée. L'arc cosinus est l'angle dont le cosinus est le contenu de chaque cible. ACOS renvoie une nouvelle colonne par colonne d'entrée, chacune contenant l'arc cosinus des nombres de la colonne d'entrée correspondante.

Si le mot clé RADIAN est utilisé, ACOS renvoie les valeurs de la plage 0 à  $\pi$ . Si le mot clé DEGREE est utilisé, ACOS renvoie les valeurs de la plage 0 à 180.

**Remarque :** Le contenu des cibles de chaque colonne spécifiée doit avoir des valeurs comprises entre -1.0 et 1.0 inclus. Dans le cas contraire, une cible vide est renvoyée pour chaque entrée non valide.

## Exemples

TEMP = ACOS(0), TEMP = ACOS(0, 0) ou TEMP = ACOS(0, RADIAN)
Crée une colonne nommée TEMP contenant la valeur 1.571 ( $\pi/2$ radians).
TEMP = ACOS(0, 1) ou TEMP = ACOS(0, DEGREE)
Crée une colonne nommée TEMP contenant la valeur 90 (degrés).
TEMP = ACOS(V1)
Crée une colonne nommée TEMP, où chaque valeur correspond à l'arc cosinus (en radians) du contenu de la colonne V1.
TEMP = ACOS(V1:V3, 1)
Crée trois colonnes nommées TEMP, VX et VY. Les valeurs de la colonne TEMP correspondent aux arc cosinus du contenu de la colonne V1, les valeurs de la colonne VX correspondent aux arc cosinus du contenu de la colonne V2 et les valeurs de la colonne VY correspondent aux arc cosinus du contenu de la colonne V3. Toutes les valeurs sont en degrés.
TEMP = ACOS(V1[10:20])
Crée une colonne nommée TEMP, dans laquelle les 11 premières cibles contiennent les arc cosinus des valeurs des lignes 10 à 20 de la colonne V1 (en radians). Les autres cibles de TEMP sont vides.
TEMP = ACOS(V1[1:5]:V2)
Crée deux colonnes nommées TEMP et VX, contenant des valeurs dans les lignes 1 à 5 (les autres cibles sont vides). Les valeurs de la colonne TEMP correspondent aux arc cosinus des lignes correspondantes de la colonne V1 et les valeurs de la colonne VX correspondent aux arc cosinus des lignes correspondantes de la colonne V2. Toutes les valeurs sont en radians.

## Fonctions associées

Fonction	Description
ACOT	Calcule l'arc cotangente du contenu de la plage de données spécifiée
ASIN	Calcule l'arc sinus du contenu de la plage de données spécifiée
ATAN	Calcule l'arc tangente du contenu de la plage de données spécifiée
COS	Calcule le cosinus du contenu de la plage de données spécifiée

---

## ACOT

### Syntaxe

ACOT(data [, units\_keyword])

### Paramètres

data

Valeurs numériques dont l'arc cotangente doit être calculé. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de data, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

units\_keyword

Ce mot clé facultatif détermine si les valeurs en entrée et les résultats sont interprétés comme des degrés ou des radians. Sélectionnez l'une des valeurs suivantes :

RADIAN - Effectue les calculs en radians (valeur par défaut)

DEGREE - Effectue les calculs en degrés

Si ce paramètre n'est pas spécifié, la valeur par défaut est radians. (Pour convertir des radians en degrés, divisez par PI et multipliez par 180.)

Pour plus de détails sur l'utilisation des mots clés dans IBM Campaign, voir «Spécifications de format», à la page 9.

Pour plus de détails sur l'utilisation des mots clés dans IBM PredictiveInsight, voir «Spécifications de format», à la page 27.

### Description

ACOT renvoie l'angle dont la cotangente correspond au contenu de chaque cible. ACOT renvoie une nouvelle colonne par colonne d'entrée, chacune contenant l'arc cotangente des nombres de la colonne d'entrée correspondante. Des nombres à virgule flottante sur 64 bits sont utilisés.

### Exemples

TEMP = ACOT(0.5), TEMP = ACOT(0.5, 0) ou TEMP = ACOT(0.5, RADIAN)
Crée une colonne nommée TEMP contenant la valeur 2.157 (radians).
TEMP = ACOT(1, 1) ou TEMP = ACOT(1, DEGREE)
Crée une colonne nommée TEMP contenant la valeur 0.022 (1/45) degrés.
TEMP = ACOT(0)
Crée une colonne nommée TEMP contenant la valeur MAX32_Float en radians.

TEMP = ACOT(V1)
Crée une colonne nommée TEMP, où chaque valeur correspond à l'arc cotangente (en radians) du contenu de la colonne V1.
TEMP = ACOT(V1:V3, 1)
Crée trois colonnes nommées TEMP, VX et VY. Les valeurs de la colonne TEMP correspondent aux arc cotangentes du contenu de la colonne V1, les valeurs de la colonne VX correspondent aux arc cotangentes du contenu de la colonne V2 et les valeurs de la colonne VY correspondent aux arc cotangentes du contenu de la colonne V3. Toutes les valeurs sont en degrés.
TEMP = ACOT(V1[10:20])
Crée une colonne nommée TEMP, dans laquelle les 11 premières cibles contiennent les arc cotangentes des valeurs des lignes 10 à 20 de la colonne V1 (en radians). Les autres cibles de TEMP sont vides.
TEMP = ACOT(V1[1:5]:V2)
Crée deux colonnes nommées TEMP et VX, contenant des valeurs dans les lignes 1 à 5 (les autres cibles sont vides). Les valeurs de la colonne TEMP correspondent aux arc cotangentes des lignes correspondantes de la colonne V1 et les valeurs de la colonne VX correspondent aux arc cotangentes des lignes correspondantes de la colonne V2. Toutes les valeurs sont en radians.

## Fonctions associées

Fonction	Description
ACOS	Calcule le cosinus inverse du contenu de la plage de données spécifiée
ASIN	Calcule l'arc sinus du contenu de la plage de données spécifiée
ATAN	Calcule l'arc tangente du contenu de la plage de données spécifiée
COT	Calcule la cotangente du contenu de la plage de données spécifiée

---

## ADD\_MONTHS

### Syntaxe

ADD\_MONTHS(months, date\_string [, input\_format])

### Paramètres

months

Entier représentant un nombre de mois à ajouter à date\_string.

date\_string

Chaîne de texte représentant une date valide, au format DELIM\_M\_D\_Y ou au format spécifié par l'argument input\_format facultatif.

input\_format

Format à utiliser pour la date calculée. Pour une liste des formats de date pris en charge, voir la fonction DATE\_FORMAT. Notez que input\_format détermine le format de la chaîne en entrée, mais aussi de la chaîne en sortie.

## Description

ADD\_MONTHS renvoie une date après l'ajout du nombre de mois spécifié à la chaîne date\_string spécifiée. La date est renvoyée au format par défaut (DELIM\_M\_D\_Y) ou au format spécifié par l'argument input\_format facultatif. Si vous souhaitez un format différent en sortie, utilisez DATE\_FORMAT.

Si l'augmentation du mois par le nombre de mois spécifié génère une date non valide, le résultat est calculé de sorte à correspondre au dernier jour du mois, comme illustré dans le dernier exemple ci-dessous. Si nécessaire, les années bissextiles sont prises en compte. Par exemple, l'ajout d'un mois à 31-Jan-2012 renvoie 29-Feb-2012.

## Exemples

ADD\_MONTHS(12, '06-25-11') ajoute un an (12 mois) à la date spécifiée et renvoie la date 06-25-12.

ADD\_MONTHS(3, '2011-06-25', DT\_DELIM\_Y\_M\_D) ajoute trois mois à la date spécifiée et renvoie la date 2011-09-25.

ADD\_MONTHS(1, '02-28-2011') renvoie la date 03-28-2011.

ADD\_MONTHS(1, '03-31-2012') renvoie la date 04-30-2012.

## Fonctions associées

Fonction	Description
DATE	Convertit une chaîne de date en date du calendrier Julien.
DATE_FORMAT	Convertit une date de input_format en output_format.

---

## ALIGN



### Syntaxe

ALIGN(ref\_series, series, range)

### Paramètres

ref\_series

series

range

## Description

Aligne une plage avec la position identifiée par sa série à la série de référence pour laquelle chaque ligne de <range> possède un numéro correspondant, comme identifié dans <series>. La ligne est alignée avec le décalage de ligne correct en associant <series> au décalage dans <reference\_series>. Si <series> ne contient pas le nombre dans <reference\_series>, des 0 sont ajoutés. Si <reference\_series> ne contient pas le nombre dans <series>, la ligne est ignorée. Si plusieurs lignes sont alignées avec la série de référence, la première est utilisée. Cette permet d'aligner des dates, exprimées par des nombres, associées à des séries commençant à des dates différentes. Ainsi, nous pouvons générer un bloc dans lequel chaque ligne correspond à des données du même jour.

---

## AND

### Syntaxe

```
data1 AND data2 data1 && data2
```

### Paramètres

data1

Nombres dont le AND logique doit être calculé avec les valeurs de data2. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de data, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

data2

Nombres dont le AND logique doit être calculé avec les valeurs de data1. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Le nombre de colonnes dans data2 doit correspondre au nombre de colonnes dans data1, à moins que data2 ne soit une constante. Pour la définition du format de data, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

### Description

AND calcule le AND logique entre les deux plages de données spécifiées. Il renvoie une nouvelle colonne pour chaque colonne d'entrée, chacune contenant la colonne correspondante de data1 dont le AND logique avec la colonne correspondante de data2 a été calculé (à savoir, le AND logique de la première colonne de data1 avec la première colonne de data, celui de la deuxième colonne avec la deuxième colonne et ainsi de suite).

Si data2 est une constante, le AND logique de chaque valeur de data1 avec cette valeur est calculé. Si data2 contient une ou plusieurs colonnes, les calculs sont effectués ligne par ligne entre une colonne de data2 et une colonne de data1. Le AND logique de la première ligne de data1 avec la première ligne de data2 est calculé, la seconde ligne avec la seconde ligne, etc. Ce calcul ligne par ligne génère un résultat pour chaque ligne jusqu'à la dernière valeur de la colonne la plus courte.

**Remarque :** L'opérateur AND peut être abrégé en double perluète ( &&). Utilisez la double perluète pour séparer les deux arguments (par exemple, pour spécifier V1 AND 3, vous pouvez simplement entrer V1&&3).

Cette macro est disponible dans IBM Interact.

## Exemples

TEMP = 1 AND 8 ou TEMP = 1 && 8
Crée une colonne nommée TEMP contenant la valeur un (tout nombre non nul prend la valeur un).
TEMP = V1 && 1
Crée une colonne nommée TEMP avec la valeur un pour chaque valeur de la colonne V1.
TEMP = V1 && V1
Crée une colonne nommée TEMP avec la valeur un pour chaque valeur non nulle de la colonne V1 et la valeur zéro pour chaque zéro de la colonne V1.
TEMP = V1 && V2
Crée une colonne nommée TEMP, dans laquelle chaque valeur est la valeur de ligne de la colonne V1 dont le AND logique est calculé avec la valeur de ligne correspondante de la colonne V2.
TEMP = V1:V3 && V4:V6
Crée trois colonnes nommées TEMP, VX et VY. La colonne TEMP contient les valeurs de V1 dont le AND logique avec les valeurs de ligne correspondantes de la colonne V4 a été calculé. La colonne VX contient les valeurs résultant du AND logique des colonnes V2 et V5. La colonne VY contient les valeurs résultant du AND logique des colonnes V3 et V6.
TEMP = V1[10:20] && V2 ou TEMP = V1[10:20] && V2[1:11]
Crée une colonne nommée TEMP, dans laquelle les 11 premières cibles contiennent le résultat du AND logique des valeurs des lignes 10 à 20 de la colonne V1 avec les valeurs des lignes 1 à 11 de la colonne V2. Les autres cibles de TEMP sont vides.

## Fonctions associées

Fonction	Description
NOT	Calcule le non logique (NON) du contenu de la plage de données spécifiée
OU	Calcule le OR logique entre deux plages de données spécifiées

---

## ASIN

### Syntaxe

ASIN(data [, units\_keyword])

### Paramètres

data

Valeurs numériques dont l'arc sinus doit être calculé. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de data, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

units\_keyword

Ce mot clé facultatif détermine si les valeurs en entrée et les résultats sont interprétés comme des degrés ou des radians. Sélectionnez l'une des valeurs suivantes :

RADIAN - Effectue les calculs en radians (valeur par défaut)

DEGREE - Effectue les calculs en degrés

Si ce paramètre n'est pas spécifié, la valeur par défaut est radians. (Pour convertir des radians en degrés, divisez par PI et multipliez par 180.)

Pour plus de détails sur l'utilisation des mots clés dans IBM Campaign, voir «Spécifications de format», à la page 9.

Pour plus de détails sur l'utilisation des mots clés dans IBM PredictiveInsight, voir «Spécifications de format», à la page 27.

## Description

ASIN calcule l'arc sinus des valeurs de la plage de données spécifiée. L'arc sinus est l'angle dont le sinus est le contenu de chaque cible. ASIN renvoie une nouvelle colonne par colonne d'entrée, chacune contenant l'arc sinus des nombres de la colonne d'entrée correspondante.

Si le mot clé RADIAN est utilisé, ASIN renvoie les valeurs de la plage  $-\pi/2$  à  $\pi/2$ . Si le mot clé DEGREE est utilisé, ASIN renvoie les valeurs de la plage -90 à 90.

**Remarque :** Le contenu des cibles de chaque colonne spécifiée doit avoir des valeurs comprises entre -1.0 et 1.0 inclus. Dans le cas contraire, ??? est renvoyé pour chaque entrée non valide.

## Exemples

TEMP = ASIN(0.5), TEMP = ASIN(0.5, 0) ou TEMP = ASIN(0.5, RADIAN)
Crée une colonne nommée TEMP contenant la valeur 0.524 ( $\pi/6$ radians).
TEMP = ASIN(0.5, 1) ou TEMP = ASIN(0.5, DEGREE)
Crée une colonne nommée TEMP contenant la valeur 30 (degrés).
TEMP = ASIN(V1)
Crée une colonne nommée TEMP, où chaque valeur correspond à l'arc sinus (en radians) du contenu de la colonne V1.

TEMP = ASIN(V1:V3, 1)
Crée trois colonnes nommées TEMP, VX et VY. Les valeurs de la colonne TEMP correspondent aux arc sinus du contenu de la colonne V1, les valeurs de la colonne VX correspondent aux arc sinus du contenu de la colonne V2 et les valeurs de la colonne VY correspondent aux arc sinus du contenu de la colonne V3. Toutes les valeurs sont en degrés.
TEMP = ASIN(V1[10:20])
Crée une colonne nommée TEMP, dans laquelle les 11 premières cibles contiennent les arc sinus des valeurs des lignes 10 à 20 de la colonne V1 (en radians). Les autres cibles de TEMP sont vides.
TEMP = ASIN(V1[1:5]:V2)
Crée deux colonnes nommées TEMP et VX, contenant des valeurs dans les lignes 1 à 5 (les autres cibles sont vides). Les valeurs de la colonne TEMP correspondent aux arc sinus des lignes correspondantes de la colonne V1 et les valeurs de la colonne VX correspondent aux arc sinus des lignes correspondantes de la colonne V2. Toutes les valeurs sont en radians.

## Fonctions associées

Fonction	Description
ACOS	Calcule le cosinus inverse du contenu de la plage de données spécifiée
ACOT	Calcule l'arc cotangente du contenu de la plage de données spécifiée
ATAN	Calcule l'arc tangente du contenu de la plage de données spécifiée
SIN	Calcule le sinus du contenu de la plage de données spécifiée

---

## ATAN

### Syntaxe

ATAN(data [, units\_keyword])

### Paramètres

data

Valeurs numériques dont l'arc tangente doit être calculé. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de data, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

units\_keyword

Ce mot clé facultatif détermine si les valeurs en entrée et les résultats sont interprétés comme des degrés ou des radians. Sélectionnez l'une des valeurs suivantes :

RADIAN - Effectue les calculs en radians (valeur par défaut)



DEGREE - Effectue les calculs en degrés

Si ce paramètre n'est pas spécifié, la valeur par défaut est radians. (Pour convertir des radians en degrés, divisez par PI et multipliez par 180.)

Pour plus de détails sur l'utilisation des mots clés dans IBM Campaign, voir «Spécifications de format», à la page 9.

Pour plus de détails sur l'utilisation des mots clés dans IBM PredictiveInsight, voir «Spécifications de format», à la page 27.

## Description

ATAN calcule l'arc tangente des valeurs de la plage de données spécifiée. L'arc tangente est l'angle dont la tangente est le contenu de chaque cible. ATAN renvoie une nouvelle colonne par colonne d'entrée, chacune contenant l'arc tangente des nombres de la colonne d'entrée correspondante.

Si le mot clé RADIAN est utilisé, ATAN renvoie les valeurs de la plage  $-\pi/2$  à  $\pi/2$ . Si le mot clé DEGREE est utilisé, ATAN renvoie les valeurs de la plage -90 à 90.

## Exemples

TEMP = ATAN(1), TEMP = ATAN(1, 0) ou TEMP = ATAN(1, RADIAN) Crée une colonne nommée TEMP contenant la valeur 0.785 ( $\pi/4$ radians).
TEMP = ATAN(1, 1) ou TEMP = ATAN(1, DEGREE) Crée une colonne nommée TEMP contenant la valeur 45 (degrés).
TEMP = ATAN(V1) Crée une colonne nommée TEMP, où chaque valeur correspond à l'arc tangente (en radians) du contenu de la colonne V1.
TEMP = ATAN(V1:V3, 1) Crée trois colonnes nommées TEMP, VX et VY. Les valeurs de la colonne TEMP correspondent aux arc tangentes du contenu de la colonne V1, les valeurs de la colonne VX correspondent aux arc tangentes du contenu de la colonne V2 et les valeurs de la colonne VY correspondent aux arc tangentes du contenu de la colonne V3. Toutes les valeurs sont en degrés.
TEMP = ATAN(V1[10:20]) Crée une colonne nommée TEMP, dans laquelle les 11 premières cibles contiennent les arc tangentes des valeurs des lignes 10 à 20 de la colonne V1 (en radians). Les autres cibles de TEMP sont vides.
TEMP = ATAN(V1[1:5]:V2) Crée deux colonnes nommées TEMP et VX, contenant des valeurs dans les lignes 1 à 5 (les autres cibles sont vides). Les valeurs de la colonne TEMP correspondent aux arc tangentes des lignes correspondantes de la colonne V1 et les valeurs de la colonne VX correspondent aux arc tangentes des lignes correspondantes de la colonne V2. Toutes les valeurs sont en radians.

## Fonctions associées

Fonction	Description
ACOS	Calcule l'arc cosinus du contenu de la plage de données spécifiée
ASIN	Calcule l'arc sinus du contenu de la plage de données spécifiée
ATAN	Calcule l'arc tangente du contenu de la plage de données spécifiée
TAN	Calcule la tangente du contenu de la plage de données spécifiée

## AVG

### Syntaxe

AVG(data [, keyword])

### Paramètres

data

Valeurs numériques dont la moyenne arithmétique doit être calculée. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de data, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

keyword

Ce mot clé facultatif détermine la manière dont le calcul est effectué sur la plage de données d'entrée. Sélectionnez l'une des valeurs suivantes :

ALL - Effectue le calcul sur toutes les cibles de data (valeur par défaut)

COL - Effectue le calcul séparément pour chaque colonne de data

ROW - Effectue le calcul séparément pour chaque ligne de data

Pour plus de détails sur l'utilisation des mots clés dans IBM Campaign, voir «Spécifications de format», à la page 9.

Pour plus de détails sur l'utilisation des mots clés dans IBM PredictiveInsight, voir «Spécifications de format», à la page 27.

**Remarque :** De nombreuses fonctions de macro utilisent les paramètres de mot clé {ALL | COL | ROW}. Ces mots clés ne s'appliquent pas dans **IBM Campaign** car les données d'entrées correspondent toujours à une colonne ou à une zone. La macro se comporte toujours comme si le mot clé COL a été spécifié. Par conséquent, vous n'avez pas besoin de spécifier ces mots clés lorsque vous utilisez **IBM Campaign**.

### Description

AVG calcule la moyenne arithmétique des cibles de la plage de données spécifiée. Pour calculer la moyenne arithmétique, le contenu de l'ensemble des cibles est

additionné, puis le résultat est divisé par le nombre de cible. Le nombre de colonnes renvoyé par AVG dépend de keyword.

- Si keyword a la valeur ALL, AVG renvoie une nouvelle colonne, contenant une valeur unique (la moyenne de toutes les cibles de data).
- Si keyword a la valeur COL, AVG renvoie une nouvelle colonne pour chaque colonne d'entrée. Chaque nouvelle colonne contient une valeur (moyenne de toutes les cibles de la colonne d'entrée correspondante).
- Si keyword a la valeur ROW, AVG renvoie une nouvelle colonne contenant la moyenne de chaque ligne de data.

**Remarque :** Les cibles vides sont exclues du calcul.

**Remarque :** AVG est identique à la fonction macro MEAN.

Cette macro est disponible dans IBM Interact.

## Exemples

TEMP = AVG(MERGE(3, 4, 5)) ou TEMP = AVG(MERGE(3, 4, 5), ALL) Crée une colonne nommée TEMP contenant la valeur 4.
TEMP = AVG(MERGE(-10, 3, 10))Crée une colonne nommée TEMP contenant la valeur 1.
TEMP = AVG(V1) Crée une colonne nommée TEMP contenant une seule valeur qui représente la moyenne arithmétique du contenu de la colonne V1.
TEMP = AVG(V1:V3) Crée une colonne nommée TEMP contenant une seule valeur qui représente la moyenne arithmétique du contenu des colonnes V1, V2 et V3.
TEMP = AVG(V1[10:20]) Crée une colonne nommée TEMP contenant une seule valeur qui représente la moyenne arithmétique des cibles des lignes 10 à 20 de la colonne V1.
TEMP = AVG(V1[1:5]:V4) Crée une colonne nommée TEMP contenant une seule valeur qui représente la moyenne arithmétique des cibles des lignes 1 à 5 des colonnes V1 à V4.
TEMP = AVG(V1:V3, COL) Crée trois colonnes nommées TEMP, VX et VY. La valeur unique de la colonne TEMP correspond à la moyenne arithmétique du contenu de la colonne V1, la valeur unique de la colonne VX correspond à la moyenne arithmétique du contenu de la colonne V2 et la valeur unique de la colonne VY correspond à la moyenne arithmétique du contenu de la colonne V3.
TEMP = AVG(MERGE(1,4), COL) Crée deux colonnes nommées TEMP et VX. TEMP contient uniquement la valeur un ; VX contient uniquement la valeur quatre.
TEMP = AVG(V1[1:5]:V3, COL) Crée trois colonnes nommées TEMP, VX et VY, contenant chacune une seule valeur. La valeur de la colonne TEMP correspond à la moyenne arithmétique des cibles des lignes 1 à 5 de la colonne V1, la valeur de la colonne VX correspond à la moyenne arithmétique des cibles des lignes 1 à 5 de la colonne V2 et la valeur de la colonne VY correspond à la moyenne arithmétique des cibles des lignes 1 à 5 de la colonne V3.
TEMP = AVG(V1, ROW) Crée une colonne nommée TEMP contenant les mêmes valeurs que la colonne V1 (la moyenne arithmétique de tout nombre est lui-même).
TEMP = AVG(V1:V3, ROW) Crée une colonne nommée TEMP dans laquelle chaque entrée de cible représente la moyenne arithmétique de la ligne correspondante dans les colonnes V1, V2 et V3.
TEMP = AVG(V1[1:5]:V3, ROW) Crée une colonne nommée TEMP, dans laquelle les cibles des lignes 1 à 5 contiennent la moyenne arithmétique de la ligne correspondante dans les colonnes V1 à V3. Les autres cibles de TEMP sont vides.

## Fonctions associées

Fonction	Description
AVG_DEV	Calcule l'écart moyen d'une plage de cibles
SUM ou TOTAL	Calcule la somme d'une plage de cibles

---

## AVG\_DEV



### Syntaxe

AVG\_DEV(data [, keyword])

### Paramètres

data

Valeurs dont l'écart moyen doit être calculé. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de data, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

keyword

Ce mot clé facultatif détermine la manière dont le calcul est effectué sur la plage de données d'entrée. Sélectionnez l'une des valeurs suivantes :

ALL - Effectue le calcul sur toutes les cibles de data (valeur par défaut)

COL - Effectue le calcul séparément pour chaque colonne de data

ROW - Effectue le calcul séparément pour chaque ligne de data

Pour plus de détails sur l'utilisation des mots clés dans IBM Campaign, voir «Spécifications de format», à la page 9.

Pour plus de détails sur l'utilisation des mots clés dans IBM PredictiveInsight, voir «Spécifications de format», à la page 27.

### Description

AVG\_DEV calcule l'écart moyen des valeurs de la plage de données spécifiée à partir de la moyenne. Pour calculer la moyenne arithmétique, le contenu de l'ensemble des cibles est additionné, puis le résultat est divisé par le nombre de cible. L'équation complète de AVG\_DEV se présente comme suit :

$$\sum_{i=1}^n |x_i - \text{mean}|$$

Le nombre de colonnes renvoyé par AVG dépend de keyword.

- Si keyword a la valeur ALL, AVG renvoie une nouvelle colonne, contenant une valeur unique (l'écart moyen de toutes les cibles de data).
- Si keyword a la valeur COL, AVG renvoie une nouvelle colonne pour chaque colonne d'entrée. Chaque nouvelle colonne contient une valeur (écart moyen de toutes les cibles de la colonne d'entrée correspondante).
- Si keyword a la valeur ROW, AVG renvoie une nouvelle colonne contenant l'écart moyen de chaque ligne de data.

**Remarque :** Les cibles vides sont exclues de la moyenne.

## Exemples

<p>TEMP = AVG_DEV(MERGE(3, 4, 5)) ou TEMP = AVG_DEV(MERGE(3, 4, 5), ALL)</p> <p>Crée une colonne nommée TEMP contenant la valeur 0.67.</p>
<p>TEMP = AVG_DEV(COLUMN(-4, 0))</p> <p>Crée une colonne nommée TEMP contenant la valeur 2.</p>
<p>TEMP = AVG_DEV(V1)</p> <p>Crée une colonne nommée TEMP contenant une valeur unique qui correspond à l'écart moyen du contenu de la colonne V1.</p>
<p>TEMP = AVG_DEV(V1:V3)</p> <p>Crée une colonne nommée TEMP contenant une valeur unique qui correspond à l'écart moyen du contenu des colonnes V1, V2 et V3.</p>
<p>TEMP = AVG_DEV(V1[10: 20])</p> <p>Crée une colonne nommée TEMP contenant une valeur unique qui correspond à l'écart moyen des cibles des lignes 10 à 20 de la colonne V1.</p>
<p>TEMP = AVG_DEV(V1[1:5]:V4)</p> <p>Crée une colonne nommée TEMP contenant une valeur unique qui correspond à l'écart moyen des cibles des lignes 1 à 5 des colonnes V1 à V4.</p>
<p>TEMP = AVG_DEV(V1:V3, COL)</p> <p>Crée trois colonnes nommées TEMP, VX et VY. La valeur unique de la colonne TEMP correspond à l'écart moyen du contenu de la colonne V1, la valeur unique de la colonne VX correspond à l'écart moyen du contenu de la colonne V2 et la valeur unique de la colonne VY correspond à l'écart moyen du contenu de la colonne V3.</p>
<p>TEMP = AVG_DEV(MERGE(1,4), COL)</p> <p>Crée deux colonnes nommées TEMP et VX, contenant toutes deux la valeur unique 0.</p>
<p>TEMP = AVG_DEV(V1[1:5]:V3, COL)</p> <p>Crée trois colonnes nommées TEMP, VX et VY contenant chacune une seule valeur. La valeur de la colonne TEMP correspond à l'écart moyen des cibles des lignes 1 à 5 de la colonne V1, la valeur de la colonne VX correspond à l'écart moyen des cibles des lignes 1 à 5 de la colonne V2, et la valeur de la colonne VY correspond à l'écart moyen des cibles des lignes 1 à 5 de la colonne V3.</p>
<p>TEMP = AVG_DEV(V1, ROW)</p> <p>Crée une colonne nommée TEMP, contenant un zéro pour chaque valeur de la colonne V1 (l'écart moyen de tout nombre est zéro).</p>

```
TEMP = AVG_DEV(V1:V3, ROW)
```

Crée une colonne nommée TEMP dans laquelle chaque entrée de cible correspond à l'écart moyen de la ligne correspondante dans les colonnes V1, V2 et V3.

```
TEMP = AVG_DEV(V1[1:5]:V3, ROW)
```

Crée une colonne nommée TEMP, dans laquelle les cibles des lignes 1 à 5 contiennent l'écart moyen de la ligne correspondante dans les colonnes V1 à V3. Les autres cibles de TEMP sont vides.

## Fonctions associées

Fonction	Description
AVG ou MEAN	Calcule la moyenne arithmétique d'une plage de cibles
SUM ou TOTAL	Calcule la somme d'une plage de cibles

---

## BETWEEN

### Syntaxe

```
value1 BETWEEN value2 AND value3
```

### Paramètres

Equivalent to `value1 >= value2 AND < value3`

### Description

BETWEEN est une variante spéciale du prédicat de comparaison. Les détails de ce prédicat sont importants et l'ordre des opérandes a des conséquences inattendues. Voir la section des exemples.

**Remarque :** FROM et FOR utilise une syntaxe identique.

### Exemples

```
SELECT * FROM movie_titles WHERE our_cost BETWEEN 11.00 and 27.50 ;
```

renvoie une liste des vidéos qui peuvent être achetées pour un prix compris entre 11,00 et 27,50 \$.

```
10 BETWEEN 5 AND 15 Is true, but: 10 BETWEEN 15 AND 5 Is false:
```

car la manière équivalente d'exprimer BETWEEN (à l'aide de AND) a un ordre spécifique qui n'a pas d'importance lorsque vous utilisez des littéraux, mais qui en a une si vous spécifiez value2 et value3 à l'aide de variables hôte, de paramètres, voire de sous-requêtes.

---

## BIT\_AND

### Syntaxe

```
data1 BIT_AND data2 data1 & data2
```

## Paramètres

data1

Entiers non négatifs dont le AND bit à bit doit être calculé avec les valeurs de data2. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de data, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

data2

Entiers non négatifs dont le AND bit à bit doit être calculé avec les valeurs de data1. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Le nombre de colonnes dans data2 doit correspondre au nombre de colonnes dans data1, à moins que data2 ne soit une constante. Pour la définition du format de data, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

## Description

BIT calcule le AND bit à bit entre les deux plages de données spécifiées. Il renvoie une nouvelle colonne pour chaque colonne d'entrée, chacune contenant la colonne correspondante de data1 dont le AND bit à bit avec la colonne correspondante de data2 a été calculé (à savoir, le AND bit à bit de la première colonne de data1 avec la première colonne de data, celui de la deuxième colonne avec la deuxième colonne et ainsi de suite).

Si data2 est une constante, le AND bit à bit de chaque valeur de data1 avec cette valeur est calculé. Si data2 contient une ou plusieurs colonnes, les calculs sont effectués ligne par ligne entre une colonne de data1 et une colonne de data2. Le AND bit à bit de la première ligne de data1 avec la première ligne de data2 est calculé, la seconde ligne avec la seconde ligne, etc. Ce calcul ligne par ligne génère un résultat pour chaque ligne jusqu'à la dernière valeur de la colonne la plus courte.

**Remarque :** La précision de cette fonction macro est limitée aux valeurs entières inférieures à  $2^{24}$ . Les valeurs négatives sont interdites.

**Remarque :** L'opérateur BIT\_AND peut être abrégé en perluète (&). Utilisez le perluète pour séparer les deux arguments (par exemple, pour spécifier BIT\_AND(V1, 3), vous pouvez simplement entrer V1&3).

## Exemples

```
TEMP = 3 BIT_AND 7 ou TEMP = 3 & 7
```

Crée une colonne nommée TEMP contenant la valeur trois (le AND bit à bit de 011 et 111 est égal à 011).

```
TEMP = V1 & 8
```

Crée une colonne nommée TEMP, dans laquelle chaque valeur correspond au AND bit à bit du contenu de la colonne V1 avec la valeur binaire 1000.

TEMP = V1 & V1
Crée une colonne nommée TEMP avec le même contenu que la colonne V1 (un AND entre toute valeur et elle-même génère cette même valeur).
TEMP = V1 & V2
Crée une colonne nommée TEMP, dans laquelle chaque valeur est la valeur de ligne de la colonne V1 dont le AND bit à bit est calculé avec la valeur de ligne correspondante de la colonne V2.
TEMP = V1:V3 & V4:V6
Crée trois colonnes nommées TEMP, VX et VY. La colonne TEMP contient les valeurs de V1 dont le AND bit à bit avec les valeurs de ligne correspondantes de la colonne V4 a été calculé. La colonne VX contient les valeurs résultant du AND bit à bit des colonnes V2 et V5. La colonne VY contient les valeurs résultant du AND bit à bit des colonnes V3 et V6.
TEMP = V1[10:20] & V2 ou TEMP = V1[10:20] & V2[1:11]
Crée une colonne nommée TEMP, dans laquelle les 11 premières cibles contiennent le résultat du AND bit à bit des valeurs des lignes 10 à 20 de la colonne V1 avec les valeurs des lignes 1 à 11 de la colonne V2. Les autres cibles de TEMP sont vides.

## Fonctions associées

Fonction	Description
BIT_NOT	Calcule le NOT bit à bit du contenu de la plage de données spécifiée
BIT_OR	Calcule le OR bit à bit entre deux plages de données spécifiées
BIT_XOR ou XOR	Calcule le XOR bit à bit entre deux plages de données spécifiées

---

## BIT\_NOT

### Syntaxe

BIT\_NOT data ~ data

### Paramètres

data

Entiers non négatifs dont le NOT bit à bit doit être calculé. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de data, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

### Description

BIT\_NOT calcule le NOT bit à bit des valeurs de la plage de données spécifiée. Il renvoie une nouvelle colonne par colonne d'entrée, chacune contenant le NOT bit à bit des valeurs des colonnes correspondantes de data.



**Remarque :** La précision de cette fonction macro est limitée aux valeurs entières inférieures à  $2^{24}$ . Les valeurs négative sont interdites.

**Remarque :** L'utilisation d'une colonne contenant le même nombre  $x$  dans chaque ligne que data revient à utiliser la constante  $x$  pour data.

**Remarque :** L'opérateur BIT\_NOT peut être abrégé en tilde (~). Utilisez le tilde avant la valeur (par exemple, pour spécifier BIT\_NOT(V1), vous pouvez simplement saisir ~V1.

## Exemples

TEMP = BIT_NOT 3 ou TEMP = ~3
Crée une colonne nommée TEMP contenant la valeur quatre (le NOT bit à bit de 011 est égal à 100).
TEMP = ~V1
Crée une colonne nommée TEMP, dans laquelle chaque valeur correspond au NOT bit à bit du contenu de la colonne V1.
TEMP = ~V1:V3
Crée trois colonnes nommées TEMP, VX et VY. Les valeurs de la colonne TEMP correspondent au NOT bit à bit du contenu de la colonne V1, les valeurs de la colonne VX correspondent au NOT bit à bit du contenu de la colonne V2 et les valeurs de la colonne VY correspondent au NOT bit à bit du contenu de la colonne V3.
TEMP = ~V1[100:200]
Crée une colonne nommée TEMP, dans laquelle les 101 premières cibles contiennent le NOT bit à bit des valeurs des lignes 1 à 50 de la colonne V1.

## Fonctions associées

Fonction	Description
BIT_AND	Calcule le AND bit à bit entre deux plages de données spécifiées
BIT_OR	Calcule le OR bit à bit entre deux plages de données spécifiées
BIT_XOR ou XOR	Calcule le XOR bit à bit entre deux plages de données spécifiées

---

## BIT\_OR

### Syntaxe

data1 BIT\_OR data2 data1 OR data2 data1 | data2

### Paramètres

data1

Entiers non négatifs dont le OR bit à bit doit être calculé avec les valeurs de data2. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de

data, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

data2

Entiers non négatifs dont le OR bit à bit doit être calculé avec les valeurs de data1. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Le nombre de colonnes dans data2 doit correspondre au nombre de colonnes dans data1, à moins que data2 ne soit une constante. Pour la définition du format de data, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

## Description

BIT calcule le OR bit à bit entre les deux plages de données spécifiées. Il renvoie une nouvelle colonne pour chaque colonne d'entrée, chacune contenant la colonne correspondante de data1 dont le OR bit à bit avec la colonne correspondante de data2 a été calculé (à savoir, le OR bit à bit de la première colonne de data1 avec la première colonne de data, celui de la deuxième colonne avec la deuxième colonne et ainsi de suite).

Si data2 est une constante, le OR bit à bit de chaque valeur de data1 avec cette valeur est calculé. Si data2 contient une ou plusieurs colonnes, les calculs sont effectués ligne par ligne entre une colonne de data2 et une colonne de data2. Le OR bit à bit de la première ligne de data1 avec la première ligne de data2 est calculé, la seconde ligne avec la seconde ligne, etc. Ce calcul ligne par ligne génère un résultat pour chaque ligne jusqu'à la dernière valeur de la colonne la plus courte.

**Remarque :** La précision de cette fonction macro est limitée aux valeurs entières inférieures à  $2^{24}$ . Les valeurs négative sont interdites.

**Remarque :** L'opérateur BIT\_OR peut être abrégé en barre verticale (|). Utilisez la barre verticale pour séparer les deux colonnes (par exemple, pour spécifier BIT\_OR(V1, 3), vous pouvez simplement entrer V1|3. Vous pouvez également utiliser OR.

## Exemples

```
TEMP = 3 BIT_OR 7, TEMP = 3 OR 7 ou TEMP = 3 | 7
```

Crée une colonne nommée TEMP contenant la valeur sept (le OR bit à bit de 011 et 111 est égal à 111).

```
TEMP = V1 | 8
```

Crée une colonne nommée TEMP, dans laquelle chaque valeur correspond au OR bit à bit du contenu de la colonne V1 avec la valeur binaire 1000.

```
TEMP = V1 | V1
```

Crée une colonne nommée TEMP avec le même contenu que la colonne V1 (un OR entre toute valeur et elle-même génère cette même valeur).

TEMP = V1   V2
Crée une colonne nommée TEMP, dans laquelle chaque valeur est la valeur de ligne de la colonne V1 dont le OR bit à bit est calculé avec la valeur de ligne correspondante de la colonne V2.
TEMP = V1:V3   V4:V6
Crée trois colonnes nommées TEMP, VX et VY. La colonne TEMP contient les valeurs de V1 dont le OR logique avec les valeurs de ligne correspondantes de la colonne V4 a été calculé. La colonne VX contient les valeurs résultant du OR logique des colonnes V2 et V5. La colonne VY contient les valeurs résultant du OR logique des colonnes V3 et V6.
TEMP = V1[10:20]   V2 ou TEMP = V1[10:20]   V2[1:11]
Crée une colonne nommée TEMP, dans laquelle les 11 premières cibles contiennent le résultat du OR bit à bit des valeurs des lignes 10 à 20 de la colonne V1 avec les valeurs des lignes 1 à 11 de la colonne V2. Les autres cibles de TEMP sont vides.

## Fonctions associées

Fonction	Description
BIT_AND	Calcule le AND bit à bit entre deux plages de données spécifiées
BIT_NOT	Calcule le NOT bit à bit du contenu de la plage de données spécifiée
BIT_XOR ou XOR	Calcule le XOR bit à bit entre deux plages de données spécifiées

## BIT\_XOR

### Syntaxe

data1 BIT\_XOR data2

### Paramètres

data1

Entiers non négatifs dont le XOR bit à bit doit être calculé avec les valeurs de data2. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de data, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

data2

Entiers non négatifs dont le XOR bit à bit doit être calculé avec les valeurs de data1. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Le nombre de colonnes dans data2 doit correspondre au nombre de colonnes dans data1, à moins que data2 ne soit une constante. Pour la définition du format de data, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

## Description

BIT\_XOR calcule le XOR bit à bit entre les deux plages de données spécifiées. Il renvoie une nouvelle colonne pour chaque colonne d'entrée, chacune contenant la colonne correspondante de data1 dont le XOR bit à bit avec la colonne correspondante de data2 a été calculé (à savoir, le XOR bit à bit de la première colonne de data1 avec la première colonne de data, celui de la deuxième colonne avec la deuxième colonne et ainsi de suite).

Si data2 est une constante, le XOR bit à bit de chaque valeur de data1 avec cette valeur est calculé. Si data2 contient une ou plusieurs colonnes, les calculs sont effectués ligne par ligne entre une colonne de data2 et une colonne de data2. Le XOR bit à bit de la première ligne de data1 avec la première ligne de data2 est calculé, la seconde ligne avec la seconde ligne, etc. Ce calcul ligne par ligne génère un résultat pour chaque ligne jusqu'à la dernière valeur de la colonne la plus courte.

**Remarque :** La précision de cette fonction macro est limitée aux valeurs entières inférieures à  $2^{24}$ . Les valeurs négative sont interdites.

## Exemples

TEMP = 3 BIT_XOR 7
Crée une colonne nommée TEMP contenant la valeur quatre (le XOR bit à bit de 011 et 111 est égal à 100).
TEMP = V1 BIT_XOR 8
Crée une colonne nommée TEMP, dans laquelle chaque valeur correspond au XOR bit à bit du contenu de la colonne V1 avec la valeur binaire 1000.
TEMP = V1 BIT_XOR V1
Crée une colonne nommée TEMP ne contenant que des zéros (un XOR entre toute valeur et elle-même génère zéro).
TEMP = V1 BIT_XOR V2
Crée une colonne nommée TEMP, dans laquelle chaque valeur est la valeur de ligne de la colonne V1 dont le XOR bit à bit est calculé avec la valeur de ligne correspondante de la colonne V2.
TEMP = V1:V3 BIT_XOR V4:V6
Crée trois colonnes nommées TEMP, VX et VY. La colonne TEMP contient les valeurs de V1 dont le XOR bit à bit avec les valeurs de ligne correspondantes de la colonne V4 a été calculé. La colonne VX contient les valeurs résultant du XOR bit à bit des colonnes V2 et V5. La colonne VY contient les valeurs résultant du XOR bit à bit des colonnes V3 et V6.
TEMP = V1[10:20] BIT_XOR V2 ou TEMP = V1[10:20] BIT_XOR V2[1:11]
Crée une colonne nommée TEMP, dans laquelle les 11 premières cibles contiennent le résultat du XOR bit à bit des valeurs des lignes 10 à 20 de la colonne V1 avec les valeurs des lignes 1 à 11 de la colonne V2. Les autres cibles de TEMP sont vides.

## Fonctions associées

Fonction	Description
BIT_AND	Calcule le AND bit à bit entre deux plages de données spécifiées

Fonction	Description
BIT_NOT	Calcule le NOT bit à bit du contenu de la plage de données spécifiée
BIT_OR	Calcule le OR bit à bit entre deux plages de données spécifiées

---

## BUFFER



### Syntaxe

BUFFER(data)

### Paramètres

data

Valeurs à copier comme constantes. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de data, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

### Description

BUFFER effectue une copie dynamique des valeurs de la plage de données spécifiée. Ces valeurs copiées changent si les valeurs correspondantes des colonnes d'entrée sont modifiées. BUFFER renvoie une nouvelle colonne par colonne d'entrée, chacune contenant une copie des valeurs de la colonne d'entrée correspondante.

**Remarque :** L'application de la fonction macro BUFFER peut considérablement améliorer les performances lors des expérimentations. Si les colonnes en entrée ou en sortie d'une expérimentation sont basées sur des calculs complexes (lents) dans la feuille de calcul, ajoutez la fonction macro BUFFER à chacune des colonnes. Les valeurs sont ainsi calculées et stockées. Sinon, chaque fois que l'expérimentation accède à la feuille de calcul pour les données de modèle, les valeurs doivent être recalculées. Si l'une des valeurs en entrée est changée, les données de BUFFER sont mises à jour dynamiquement comme les autres fonctions macro de la feuille de calcul.

**Remarque :** Si les valeurs des données ne changent jamais, utilisez la fonction macro CONSTANT à la place. Elle crée une copie statique de la plage de données.

**Remarque :** Si une fonction utilisateur est créée à partir d'une définition de fonction à l'aide de la fonction macro BUFFER, la portion de la définition de fonction comprise dans la fonction macro BUFFER est considérée comme une constante. Les variables en entrées ne sont pas toutes requises pour appliquer la fonction utilisateur.

## Exemples

<pre>TEMP = BUFFER(4.3)</pre> <p>Crée une colonne nommée TEMP contenant la valeur 4.3.</p>
<pre>TEMP = BUFFER(V1)</pre> <p>Crée une colonne nommée TEMP, dans laquelle chaque valeur correspond à une copie dynamique du contenu de la colonne V1.</p>
<pre>TEMP = BUFFER(V1:V3)</pre> <p>Crée trois colonnes nommées TEMP, VX et VY. Les valeurs de la colonne TEMP sont des copies du contenu de la colonne V1, les valeurs de la colonne VX sont des copies du contenu de la colonne V2 et les valeurs de la colonne VY sont des copies du contenu de la colonne V3.</p>
<pre>TEMP = BUFFER(V1[10:20])</pre> <p>Crée une colonne nommée TEMP, dans laquelle les 11 premières cibles contiennent des copies des valeurs des lignes 10 à 20 de la colonne V1. Les autres cibles de TEMP sont vides.</p>
<pre>TEMP = BUFFER(V1[50:99]:V2)</pre> <p>Crée deux colonnes nommées TEMP et VX, contenant des valeurs dans les lignes 1 à 50 (les autres cibles sont vides). Les valeurs de la colonne TEMP sont des copies des lignes de la colonne V1 et les valeurs de la colonne VX sont des copies des valeurs de la colonne V2.</p>
<pre>TEMP = BUFFER(EXTRACT(!ISERROR(V1:V3), V1:V3))</pre> <p>Crée trois colonnes nommées TEMP, VX et VY. Les trois colonnes contiennent les lignes valides (lignes ne contenant pas de cibles ???) des colonnes V1 à V3. L'encapsulation de la fonction macro EXTRACT dans CONSTANT empêche les colonnes TEMP, VX et VY d'être recalculées si les colonnes V1 à V3 sont changées ; il n'est donc pas nécessaire d'utiliser la fonction macro EXTRACT, qui réclame une puissance de calcul considérable.</p>

## Fonctions associées

Fonction	Description
CONSTANT	Copie la plage de données d'entrée une fois (pas de mise à jour dynamique)

---

## CEILING

### Syntaxe

CEILING(data)

### Paramètres

data

Valeurs numériques dont le plafond doit être calculé. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de data, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

## Description

CEILING calcule le plafond des valeurs de la plage de données spécifiée. Le plafond d'un nombre est l'entier le plus petit *non* inférieur à ce nombre. CEILING renvoie une nouvelle colonne par colonne d'entrée, chacune contenant le plafond des nombres de la colonne d'entrée correspondante.

## Exemples

TEMP = CEILING(4.3)
Crée une colonne nommée TEMP contenant la valeur 5.
TEMP = CEILING(2.9)
Crée une colonne nommée TEMP contenant la valeur -2.
TEMP = CEILING(V1)
Crée une colonne nommée TEMP, dans laquelle chaque valeur correspond au plafond du contenu de la colonne V1.
TEMP = CEILING(V1:V3)
Crée trois colonnes nommées TEMP, VX et VY. Les valeurs de la colonne TEMP correspondent aux plafonds du contenu de la colonne V1, les valeurs de la colonne VX correspondent aux plafonds du contenu de la colonne V2 et les valeurs de la colonne VY correspondent aux plafonds du contenu de la colonne V3.
TEMP = CEILING(V1[10:20])
Crée une colonne nommée TEMP, dans laquelle les 11 premières cibles contiennent les plafonds des valeurs des lignes 10 à 20 de la colonne V1. Les autres cibles de TEMP sont vides.
TEMP = CEILING(V1[50:99]:V2)
Crée deux colonnes nommées TEMP et VX, contenant des valeurs dans les lignes 1 à 50 (les autres cibles sont vides). Les valeurs de la colonne TEMP sont les plafonds des lignes de la colonne V1 et les valeurs de la colonne VX sont les plafonds des valeurs de la colonne V2.

## Fonctions associées

Fonction	Description
FLOOR ou INT	Calcule la partie entière de chaque valeur de la plage de données spécifiée
FRACTION	Renvoie la partie décimale de chaque valeur de la plage de données spécifiée
TRUNCATE	Renvoie la partie non décimale de chaque valeur de la plage de données spécifiée

---

## COLUMN

### Syntaxe

COLUMN(data [, data]...) or (data [, data]...)

### Paramètres

data

Valeur à utiliser lors de la création d'une colonne. Il peut s'agir d'une valeur constante (valeur numérique ou texte ASCII entre guillemets), d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Ce paramètre peut être répété plusieurs fois, mais les paramètres suivants doivent avoir la même dimensionnalité (largeur de colonne) que le premier paramètre. Toutes les valeurs de tous les paramètres data doivent être des valeurs numériques ou du texte ASCII (vous ne pouvez pas mélanger des valeurs numériques et du texte). Si plusieurs paramètres data sont fournis, ils doivent tous comporter le même nombre de colonnes. Pour la définition du format de data, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

## Description

COLUMN concatène verticalement ses entrées dans les colonnes d'un groupe de fonctions. Il renvoie le même nombre de nouvelles colonnes que celui indiqué dans chaque paramètre d'entrée. Un nombre illimité d'arguments peuvent être fournis. Tous les arguments doivent être des valeurs numériques ou des chaînes de texte ASCII (vous ne pouvez pas mélanger des valeurs numériques et du texte).

**Remarque :** La fonction macro COLUMN peut être abrégée en séparant les arguments data par des virgules, entre parenthèses (par exemple, TEMP = MEAN((1,2,3,4), ALL)). En dehors d'une autre fonction macro, la paire de paranthèses n'est pas nécessaire (par exemple, V1=1,2,3 est équivalent à V1=COLUMN(1,2,3)).

## Exemples

TEMP = COLUMN(3, 4, 5), TEMP = (3,4,5) ou TEMP = 3,4,5
Crée une colonne nommée TEMP dont les trois premières cibles contiennent les valeurs 3, 4 et 5.
TEMP = COLUMN("one", "two", "three")
Crée une colonne nommée TEMP dont les trois premières cibles contiennent les valeurs "one", "two " et " three ".
TEMP = AVG(V1), STDV(V1)
Crée une colonne nommée TEMP avec la moyenne de la colonne V1 dans la première cible et l'écart type de la colonne V1 dans la deuxième cible.
TEMP = V1:V2, V3:V4
Crée deux colonnes nommées TEMP et VX où la colonne TEMP contient les valeurs de la colonne V1 suivies des valeurs de la colonne V3. La colonne VX contient les valeurs de la colonne V2 suivies des valeurs de la colonne V4.
TEMP = V1:V2, V3:V4
Crée deux colonnes nommées TEMP et VX où la colonne TEMP contient les valeurs des cibles 1 à 10 de la colonne V1 suivies de toutes les valeurs de la colonne V3. La colonne VX contient les valeurs des cibles 1 à 10 de la colonne V2 suivies de toutes les valeurs de la colonne V4.
TEMP = V1:V2, V3:V4
Crée deux colonnes nommées TEMP et VX, contenant chacune une valeur unique. La colonne TEMP contient la moyenne des colonnes V1 et V2. La colonne VX contient la moyenne des colonnes V3 et V4.



## Fonctions associées

Fonction	Description
MERGE	Crée un groupe de données en concaténant horizontalement les valeurs en entrée
TO	Génère l'opérateur de plage
TRANSPOSE	Transpose une plage de données spécifiée

---

## CONSTANT



### Syntaxe

CONSTANT(data)

### Paramètres

data

Valeurs à copier comme constantes. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de data, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

### Description

CONSTANT effectue une copie statique des valeurs de la plage de données spécifiée. Ces valeurs copiées ne changent *pas* si les valeurs correspondantes des colonnes d'entrée sont modifiées. Les valeurs des données sont copiées une fois au moment où la fonction macro est appliquée. CONSTANT renvoie une nouvelle colonne par colonne d'entrée, chacune contenant une copie statique des valeurs de la colonne d'entrée correspondante.

**Remarque :** L'application de la fonction macro CONSTANT peut considérablement améliorer les performances lors des expérimentations. Si les colonnes en entrée ou en sortie d'une expérimentation sont basées sur des calculs complexes (lents) dans la feuille de calcul, ajoutez la fonction macro CONSTANT à chacune des colonnes. Les valeurs sont ainsi calculées une fois pour toutes et stockées. Sinon, chaque fois que l'expérimentation accède à la feuille de calcul pour les données de modèle, les valeurs doivent être recalculées.

**Remarque :** Si les valeurs des données peuvent être changées, utilisez la fonction macro BUFFER à la place. Cette opération crée une copie dynamique de la plage de données, où les valeurs copiées sont changées si les valeurs en entrée correspondantes sont changées.

**Remarque :** Si une fonction utilisateur est créée à partir d'une définition de fonction à l'aide de la fonction macro CONSTANT, la portion de la définition de fonction comprise dans la fonction macro CONSTANT est considérée comme une constante. Les variables en entrées ne sont pas toutes requises pour appliquer la fonction utilisateur.

## Exemples

<code>TEMP = CONSTANT(4.3)</code> Crée une colonne nommée TEMP contenant la valeur 4.3.
<code>TEMP = CONSTANT(V1)</code> Crée une colonne nommée TEMP, dans laquelle chaque valeur correspond à une copie statique du contenu de la colonne V1.
<code>TEMP = CONSTANT (V1:V3)</code> Crée trois colonnes nommées TEMP, VX et VY. Les valeurs de la colonne TEMP sont des copies du contenu de la colonne V1, les valeurs de la colonne VX sont des copies du contenu de la colonne V2 et les valeurs de la colonne VY sont des copies du contenu de la colonne V3.
<code>TEMP = CONSTANT(V1[10:20])</code> Crée une colonne nommée TEMP, dans laquelle les 11 premières cibles contiennent des copies des valeurs des lignes 10 à 20 de la colonne V1. Les autres cibles de TEMP sont vides.
<code>TEMP = CONSTANT(V1[50:99]:V2)</code> Crée deux colonnes nommées TEMP et VX, contenant des valeurs dans les lignes 1 à 50 (les autres cibles sont vides). Les valeurs de la colonne TEMP sont des copies des lignes de la colonne V1 et les valeurs de la colonne VX sont des copies des valeurs de la colonne V2.
<code>TEMP = CONSTANT(EXTRACT(!ISERROR(V1:V3), V1:V3))</code> Crée trois colonnes nommées TEMP, VX et VY. Les trois colonnes contiennent les lignes valides (lignes ne contenant pas de cibles ???) des colonnes V1 à V3. L'encapsulation de la fonction macro EXTRACT dans CONSTANT empêche les colonnes TEMP, VX et VY d'être recalculées si les colonnes V1 à V3 sont changées ; il n'est donc pas nécessaire d'utiliser la fonction macro EXTRACT, qui réclame une puissance de calcul considérable.

## Fonctions associées

Fonction	Description
BUFFER	Copie la plage de données d'entrée et la met à jour de manière dynamique

---

## COS

### Syntaxe

`COS(data [, units_keyword])`

### Paramètres

data

Valeurs numériques dont le cosinus doit être calculé. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de data, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

units\_keyword

Ce mot clé facultatif détermine si les valeurs en entrée et les résultats sont interprétés comme des degrés ou des radians. Sélectionnez l'une des valeurs suivantes :

RADIAN - Effectue les calculs en radians (valeur par défaut)

DEGREE - Effectue les calculs en degrés

Si ce paramètre n'est pas spécifié, la valeur par défaut est radians. (Pour convertir des radians en degrés, divisez par PI et multipliez par 180.)

Pour plus de détails sur l'utilisation des mots clés dans IBM Campaign, voir «Spécifications de format», à la page 9.

Pour plus de détails sur l'utilisation des mots clés dans IBM PredictiveInsight, voir «Spécifications de format», à la page 27.

## Description

COS calcule le cosinus des valeurs de la plage de données spécifiée. Il renvoie une nouvelle colonne par colonne d'entrée, chacune contenant le cosinus des nombres de la colonne d'entrée correspondante.

## Exemples

<code>TEMP = COS(PI), TEMP = COS(PI, 0) ou TEMP = COS(PI, RADIAN)</code>
Renvoie une colonne nommée TEMP contenant la valeur unique -1.
<code>TEMP = COS(90, 1) ou TEMP = COS(90, DEGREE)</code>
Renvoie une colonne nommée TEMP contenant la valeur unique zéro.
<code>TEMP = COS(V1), TEMP = COS(V1, 0) ou TEMP = COS(V1, RADIAN)</code>
Crée une colonne nommée TEMP, où chaque valeur correspond au cosinus (en radians) du contenu de la colonne V1.
<code>TEMP = COS(V1:V3, 1)</code>
Crée trois colonnes nommées TEMP, VX et VY. Les valeurs de la colonne TEMP correspondent aux cosinus du contenu de la colonne V1, les valeurs de la colonne VX correspondent aux cosinus du contenu de la colonne V2 et les valeurs de la colonne VY correspondent aux cosinus du contenu de la colonne V3. Toutes les valeurs sont en degrés.
<code>TEMP = COS(V1[10:20])</code>
Crée une colonne nommée TEMP, dans laquelle les 11 premières cibles contiennent les cosinus des valeurs des lignes 10 à 20 de la colonne V1 (en radians). Les autres cibles de TEMP sont vides.
<code>TEMP = COS(V1[1:5]:V2)</code>
Crée deux colonnes nommées TEMP et VX, contenant des valeurs dans les lignes 1 à 5 (les autres cibles sont vides). Les valeurs de la colonne TEMP correspondent aux cosinus des lignes correspondantes de la colonne V1 et les valeurs de la colonne VX correspondent aux cosinus des lignes correspondantes de la colonne V2. Toutes les valeurs sont en radians.

## Fonctions associées

Fonction	Description
ACOS	Calcule l'arc cosinus du contenu de la plage de données spécifiée
COSH	Calcule le cosinus hyperbolique du contenu de la plage de données spécifiée
SIN	Calcule le sinus du contenu de la plage de données spécifiée
TAN	Calcule la tangente du contenu de la plage de données spécifiée

---

## COSH

### Syntaxe

`COSH(data [, units_keyword])`

### Paramètres

`data`

Valeurs numériques dont le cosinus hyperbolique doit être calculé. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de `data`, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

`units_keyword`

Ce mot clé facultatif détermine si les valeurs en entrée et les résultats sont interprétés comme des degrés ou des radians. Sélectionnez l'une des valeurs suivantes :

RADIAN - Effectue les calculs en radians (valeur par défaut)

DEGREE - Effectue les calculs en degrés

Si ce paramètre n'est pas spécifié, la valeur par défaut est radians. (Pour convertir des radians en degrés, divisez par PI et multipliez par 180.)

Pour plus de détails sur l'utilisation des mots clés dans IBM Campaign, voir «Spécifications de format», à la page 9.

Pour plus de détails sur l'utilisation des mots clés dans IBM PredictiveInsight, voir «Spécifications de format», à la page 27.

### Description

COSH calcule le cosinus hyperbolique des valeurs de la plage de données spécifiée. Pour  $x$  en radians, le cosinus hyperbolique d'un nombre est :

$$\cosh(x) = \frac{e^x + e^{-x}}{2}$$

où  $e$  représente le nombre naturel, 2.7182818. COSH renvoie une nouvelle colonne par colonne d'entrée, chacune contenant le cosinus hyperbolique des nombres de la colonne d'entrée correspondante.

**Remarque :** Si la valeur  $x$  est trop élevée, une erreur de dépassement est renvoyée. Cela se produit si  $\cosh(x)$  dépasse la valeur maximale à virgule flottante sur 32 bits.

## Exemples

TEMP = COSH(0), TEMP = COSH(0, 0) ou TEMP = COSH(0, RADIAN)
Renvoie une colonne nommée TEMP contenant la valeur un.
TEMP = COSH(V1)
Crée une colonne nommée TEMP, où chaque valeur correspond au cosinus hyperbolique (en radians) du contenu de la colonne V1.
TEMP = COSH(V1:V3, 1) ou TEMP = COSH(V1:V3, DEGREE)
Crée trois colonnes nommées TEMP, VX et VY. Les valeurs de la colonne TEMP correspondent aux cosinus hyperboliques du contenu de la colonne V1, les valeurs de la colonne VX correspondent aux cosinus hyperboliques du contenu de la colonne V2 et les valeurs de la colonne VY correspondent aux cosinus hyperboliques du contenu de la colonne V3. Toutes les valeurs sont en degrés.
TEMP = COSH(V1[10:20])
Crée une colonne nommée TEMP, dans laquelle les 11 premières cibles contiennent les cosinus hyperboliques des valeurs des lignes 10 à 20 de la colonne V1 (en radians). Les autres cibles de TEMP sont vides.
TEMP = COSH(V1[1:5]:V2)
Crée deux colonnes nommées TEMP et VX, contenant des valeurs dans les lignes 1 à 5 (les autres cibles sont vides). Les valeurs de la colonne TEMP correspondent aux cosinus hyperboliques des lignes correspondantes de la colonne V1 et les valeurs de la colonne VX correspondent aux cosinus hyperboliques des lignes correspondantes de la colonne V2. Toutes les valeurs sont en radians.

## Fonctions associées

Fonction	Description
ACOS	Calcule l'arc cosinus du contenu de la plage de données spécifiée
COS	Calcule le cosinus du contenu de la plage de données spécifiée
SINH	Calcule le sinus hyperbolique du contenu de la plage de données spécifiée
TANH	Calcule la tangente hyperbolique du contenu de la plage de données spécifiée

---

# COT

## Syntaxe

COT(data [, units\_keyword])

## Paramètres

data

Valeurs numériques dont la cotangente doit être calculée. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de data, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

units\_keyword

Ce mot clé facultatif détermine si les valeurs en entrée et les résultats sont interprétés comme des degrés ou des radians. Sélectionnez l'une des valeurs suivantes :

RADIAN - Effectue les calculs en radians (valeur par défaut)

DEGREE - Effectue les calculs en degrés

Si ce paramètre n'est pas spécifié, la valeur par défaut est radians. (Pour convertir des radians en degrés, divisez par PI et multipliez par 180.)

Pour plus de détails sur l'utilisation des mots clés dans IBM Campaign, voir «Spécifications de format», à la page 9.

Pour plus de détails sur l'utilisation des mots clés dans IBM PredictiveInsight, voir «Spécifications de format», à la page 27.

## Description

COT calcule la cotangente des valeurs de la plage de données spécifiée. La cotangente est la réciproque de la tangente. COT renvoie une nouvelle colonne par colonne d'entrée, chacune contenant la cotangente des nombres de la colonne d'entrée correspondante.

**Remarque :** Si une cible contient une valeur dont la tangente est zéro, alors l'arc cotangente est infini. Dans ce cas, COT renvoie le plus grand nombre à virgule flottante de 32 bits.

## Exemples

TEMP = COT(90), TEMP = COT(90, 0) ou TEMP = COT(90, RADIAN)
Renvoie une colonne nommée TEMP contenant la valeur -0.5.
TEMP = COT(0)
Renvoie une colonne nommée TEMP contenant la valeur MAX_FLOAT_32.

TEMP = COT(V1, 1) ou TEMP = COT(V1, DEGREE)
Crée une colonne nommée TEMP, où chaque valeur correspond à la cotangente du contenu (en degrés) de la colonne V1.
TEMP = COT(V1:V3, 1)
Crée trois colonnes nommées TEMP, VX et VY. Les valeurs de la colonne TEMP correspondent aux cotangentes du contenu de la colonne V1, les valeurs de la colonne VX correspondent aux cotangentes du contenu de la colonne V2 et les valeurs de la colonne VY correspondent aux cotangentes du contenu de la colonne V3. Toutes les valeurs sont en degrés.
TEMP = COT(V1[10:20])
Crée une colonne nommée TEMP, dans laquelle les 11 premières cibles contiennent les cotangentes des valeurs des lignes 10 à 20 de la colonne V1 (en radians). Les autres cibles de TEMP sont vides.
TEMP = COT(V1[1:5]:V2)
Crée deux colonnes nommées TEMP et VX, contenant des valeurs dans les lignes 1 à 5 (les autres cibles sont vides). Les valeurs de la colonne TEMP correspondent aux cotangentes des lignes correspondantes de la colonne V1 et les valeurs de la colonne VX correspondent aux cotangentes des lignes correspondantes de la colonne V2. Toutes les valeurs sont en radians.

## Fonctions associées

Fonction	Description
ACOT	Calcule l'arc cotangente du contenu de la plage de données spécifiée
COS	Calcule le cosinus du contenu de la plage de données spécifiée
SIN	Calcule le sinus du contenu de la plage de données spécifiée
TAN	Calcule la tangente du contenu de la plage de données spécifiée

---

## COUNT

### Syntaxe

COUNT(data)

### Paramètres

data

Plage de cibles dont le nombre de cibles doit être compté. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de data, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

## Description

COUNT compte le nombre de valeurs dans la plage de données spécifiée. Il renvoie une nouvelle colonne contenant une valeur unique représentant le nombre de cibles contenant des valeurs à l'intérieur de la plage de données spécifiée.

**Remarque :** Le comptage d'une colonne vide renvoie la valeur zéro.

## Exemples

TEMP = COUNT(AVG(V1:V5))
Crée une colonne nommée TEMP contenant une valeur unique égale à un (la fonction AVG renvoie une seule cible dans le mode par défaut).
TEMP = COUNT(V1)
Crée une colonne nommée TEMP contenant une valeur unique indiquant le nombre de cibles contenant des valeurs dans la colonne V1.
TEMP = COUNT(V1:V3)
Crée une colonne nommée TEMP contenant une valeur unique indiquant le nombre de cibles contenant des valeurs dans les colonnes V1, V2 et V3.
TEMP = COUNT(V1[10:20])
Crée une colonne nommée TEMP contenant la valeur 11 (les plages sont inclusives), à condition que toutes les cibles contiennent des valeurs.
TEMP = COUNT(V1[1:5]:V4)
Crée une colonne nommée TEMP contenant la valeur 20 (5 cibles dans chaque colonne fois 4 colonnes = 20 cibles), à condition que toutes les cibles contiennent des valeurs.
TEMP = COUNT(V1[1:10])
Crée une colonne nommée TEMP contenant la valeur 3, à condition que les lignes 1 à 3 de la colonne V1 contiennent des valeurs et que les lignes 4 à 10 sont vides.

## Fonctions associées

Fonction	Description
COUNT_DIFF	Renvoie chaque valeur unique de l'entrée avec le nombre de fois que la valeur est apparue
OFFSET	Renvoie le décalage de chaque valeur de la colonne d'entrée par rapport au début du flux
SUM ou TOTAL	Calcule la somme d'une plage de cibles

---

## COUNT\_DIFF



### Syntaxe

COUNT\_DIFF(data)



## Paramètres

data

Plage de cibles dont les valeurs uniques et la fréquence d'occurrence doivent être comptées. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de data, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

## Description

COUNT\_DIFF recherche le nombre de valeurs différentes dans la plage de données spécifiée et compte le nombre de fois que chaque valeur apparaît. Il renvoie deux nouvelles colonnes. La première colonne contient chaque valeur unique. La deuxième colonne contient le nombre de fois que la valeur correspondante est apparue dans la plage de données. COUNT\_DIFF ignore la casse lors de la comparaison des chaînes de texte. Par exemple, "Aaa", "aAa" et "AAA" sont tous comptés comme une même classe.

**Remarque :** Le calcul effectué par la fonction macro COUNT\_DIFF peut être assez long si data est volumineux. Une barre de progression du calcul est affichée jusqu'à la fin du calcul. Si vous décidez d'annuler le calcul, cliquez sur le "X" dans la barre de progression et supprimez la définition de fonction contenant la fonction macro COUNT\_DIFF.

## Exemples

<code>TEMP = COUNT_DIFF(COLUMN(1,2,3,1))</code>	Crée deux colonnes nommées TEMP et VX. TEMP contient les valeurs 1, 2 et 3. VX contient les résultats 2, 1 et 1.
<code>TEMP = COUNT_DIFF(COLUMN("x", "a", "a", "b"))</code>	Crée deux colonnes nommées TEMP et VX. TEMP contient les valeurs x, a et b. VX contient les résultats 1, 2 et 1.
<code>TEMP = COUNT_DIFF(V1)</code>	Crée deux colonnes nommées TEMP et VX, où TEMP contient toutes les valeurs uniques de la colonne V1 et VX contient un résultat pour chaque ligne correspondante de TEMP.
<code>TEMP = COUNT_DIFF(V1:V3)</code>	Crée deux colonnes nommées TEMP et VX, où TEMP contient toutes les valeurs uniques des colonnes V1 à V3 et VX contient un résultat pour chaque ligne correspondante de TEMP.
<code>TEMP = COUNT_DIFF(V1[10:20])</code>	Crée deux colonnes nommées TEMP et VX, où TEMP contient toutes les valeurs uniques des lignes 10 à 20 de la colonne V1 et VX contient un résultat pour chaque ligne correspondante de TEMP.
<code>TEMP = COUNT_DIFF(V1[1:5]:V4)</code>	Crée deux colonnes nommées TEMP et VX, où TEMP contient toutes les valeurs uniques des lignes 1 à 5 des colonnes V1 à V4 et VX contient un résultat pour chaque ligne correspondante de TEMP.

## Fonctions associées

Fonction	Description
COUNT	Compte le nombre de cibles contenant des valeurs dans la plage de données spécifiée
HISTOGRAM	Calcule l'histogramme d'une plage de données spécifiée à l'aide des limites d'ensemble fournies

---

## COV



### Syntaxe

`COV(data1, data2)`

### Paramètres

`data1`

Premier jeu de données. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de data, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

`data2`

Deuxième jeu de données. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de data, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

### Description

COV calcule la covariance de deux plages de données spécifiées. <sup>1</sup> Il génère autant de colonnes que la dimension minimale des entrées (à la fois en largeur et en hauteur). COV est calculé comme suit :

$$\text{COV}(x, y) = \sum_{i=1}^n \frac{(x_i - \mu_x)(y_i - \mu_y)}{N}$$

où  $x$  et  $y$  correspondent à des colonnes contenant le même nombre de valeurs  $N$ ,  $\mu_x$  est la moyenne de  $x$  et  $\mu_y$  est la moyenne de  $y$ .

### Exemples

```
TEMP = COV(COLUMN(1,2), COLUMN(1,3))
```

Crée une colonne nommée TEMP contenant la valeur 0.5.

```
TEMP = COV(V1, V2)
```

Crée une colonne nommée TEMP contenant la covariance entre les données de la colonne V1 et celles de la colonne V2.

```
TEMP = COV(V1:V2, V3:V4)
```

Crée deux colonnes nommées TEMP et VX. La colonne TEMP contiendra la covariance entre les colonnes V1 et V3. La colonne VX contiendra la covariance entre les colonnes V2 et V4.

```
TEMP = COV(V1[1:50]:V2, V3[100:200]:V4)
```

Crée deux colonnes nommées TEMP et VX. La colonne TEMP contiendra la covariance entre les lignes 1 à 50 de la colonne V1 et les lignes 100 à 200 de la colonne V3. La colonne VX contiendra la covariance entre les lignes 1 à 50 de la colonne V2 et les lignes 100 à 200 de la colonne V4.

---

## CURRENT\_DATE

### Syntaxe

```
CURRENT_DATE([format])
```

### Paramètres

format

Un des mots clés du tableau ci-après spécifiant le format de date de date\_string.

### Description

CURRENT\_DATE renvoie la date du jour dans le format format. La date est déterminée par l'horloge sur le serveur IBM. Si aucun mot clé format n'est spécifié, la valeur par défaut DELIM\_M\_D\_Y est utilisée.

Pour toutes les bases de données recommandées, IBM Campaign tente d'exécuter la macro CURRENT\_DATE dans la base de données à l'aide d'un appel SQL de l'heure actuelle pris en charge par la base de données (par exemple, SYSDATE, GETDATE, DATE ou TODAY). Dans ces cas, tous les paramètres (y compris le format de la date) de cette fonction macro sont ignorés et la sortie inclut tout ce qui est renvoyé par la base de données (par exemple, un composant de temps peuvent être inclus dans la sortie). Si cela se produit et que vous ne souhaitez renvoyer que la date ou la date dans un format différent, vous pouvez écrire votre propre macro personnalisée à l'aide d'un script SQL brut ou utiliser d'autres macros IBM. Par exemple :

```
DATE_STRING(CURRENT_JULIAN( ), ...)
```

Dans certains cas, la macro CURRENT\_DATE() est exécutée sur le serveur IBM Campaign (par exemple, lors d'une exécution sur un fichier à plat, sur une base de données non recommandée sans support SQL équivalent ou si l'expression de la macro Campaign ne peut pas être convertie dans la base de données). Dans ces cas, tous les paramètres sont reconnus et la sortie est renvoyée au format sélectionné.

**Remarque :** Pour plus d'informations sur les formats de date disponibles, voir la section relative aux mots clés de format valide.

Cette macro est disponible dans IBM Interact.

## Exemples

Si la date du jour est le 7 septembre 2000, `CURRENT_DATE()` renvoie "09/07/00".

## Fonctions associées

Fonction	Description
DATE_FORMAT	Convertit les dates d'un format à un autre.
DATE_JULIAN	Renvoie la date du calendrier Julien de l'entrée.
DATE_STRING	Renvoie la chaîne de date sous la forme d'une date du calendrier Julien.
DATE	Convertit une chaîne de date en date du calendrier Julien.

---

## CURRENT\_DAY

### Syntaxe

`CURRENT_DAY()`

### Description

`CURRENT_DAY` renvoie le jour du mois en cours sous la forme d'un nombre compris entre 1 et 31. La date est déterminée par l'horloge système sur le serveur IBM.

Cette macro est disponible dans IBM Interact.

### Exemples

Si la date du jour est le 19 juin, `CURRENT_DAY()` renvoie le nombre 19.

## Fonctions associées

Fonction	Description
CURRENT_JULIAN	Renvoie le nombre au format Julien de la date du jour.
CURRENT_MONTH	Renvoie le mois actuel sous la forme d'un nombre.
CURRENT_TIME	Renvoie l'heure actuelle sous la forme d'une chaîne.
CURRENT_WEEKDAY	Renvoie le jour de la semaine sous la forme d'un nombre.
CURRENT_YEAR	Renvoie l'année en cours sous la forme d'un nombre.

---

## CURRENT\_JULIAN

### Syntaxe

CURRENT\_JULIAN()

### Description

CURRENT\_JULIAN() renvoie le nombre Julien de la date du jour (nombre de jours écoulés depuis le 1er janvier 0000). Equivalent à la macro DATE(CURRENT\_DATE()).

### Exemples

Si la date du jour est le 31 août 2000, CURRENT\_JULIAN() renvoie le nombre 730729.

### Fonctions associées

Fonction	Description
CURRENT_DAY	Renvoie le jour actuel sous la forme d'un nombre.
CURRENT_MONTH	Renvoie le mois actuel sous la forme d'un nombre.
CURRENT_TIME	Renvoie l'heure actuelle sous la forme d'une chaîne.
CURRENT_WEEKDAY	Renvoie le jour de la semaine sous la forme d'un nombre.
CURRENT_YEAR	Renvoie l'année en cours sous la forme d'un nombre.

---

## CURRENT\_MONTH

### Syntaxe

CURRENT\_MONTH()

### Description

CURRENT\_MONTH renvoie le mois en cours de l'année sous la forme d'un nombre compris entre 1 et 12.

Cette macro est disponible dans IBM Interact.

### Exemples

Si la date du jour est le 19 juin, CURRENT\_MONTH() renvoie le nombre 6.

### Fonctions associées

Fonction	Description
CURRENT_DAY	Renvoie le jour actuel sous la forme d'un nombre.
CURRENT_JULIAN	Renvoie le nombre au format Julien actuel.

Fonction	Description
CURRENT_TIME	Renvoie l'heure actuelle sous la forme d'une chaîne.
CURRENT_WEEKDAY	Renvoie le jour de la semaine sous la forme d'un nombre.
CURRENT_YEAR	Renvoie l'année en cours sous la forme d'un nombre.

---

## CURRENT\_TIME

### Syntaxe

CURRENT\_TIME()

### Description

CURRENT\_TIME renvoie l'heure actuelle sous la forme d'une chaîne. L'heure est déterminée par l'horloge système sur le serveur IBM.

## Configuration des dates sur votre application Web

Pour afficher correctement les dates sur votre application Web dans les versions actuelles d'IBM Campaign, le fichier de configuration de votre serveur dorsal doit être correctement configuré. Cela est particulièrement important pour les paramètres `dDateFormat` et `DateOutputFormatString` de la base de données qui contient les tables système. Si ces paramètres ne sont pas configurés correctement, les dates ne seront pas affichées non plus correctement dans Campaign. Vous configurez ces propriétés à l'aide de la plateforme marketing IBM.

### Définition des dates d'une langue spécifique sur votre application Web

**Remarque :** Tous les fichiers référencés sont installés par le programme d'installation de l'application Web, sauf indication contraire.

**Important :** `webapphome` fait référence au répertoire dans lequel l'application Web de Campaign a été installée. `language_code` fait référence aux paramètres de langue que vous choisissez pour votre système.

1. Editez le fichier `webapphome/conf/campaign_config.xml` pour vous assurer que `language_code` est présent dans la liste séparée par des virgules de la balise `<supportedLocales>`, comme illustré ci-dessous :  

```
<supportedLocales>en_US, language_code</supportedLocales>
```
2. Dans le répertoire `webapphome/webapp`, copiez l'intégralité de l'arborescence des répertoires `en_US` to `language_code` (la casse doit être respectée).
3. Dans `webapphome/webapp/WEB-INF/classes/resources`, copiez `StaticMessages_en_US.properties` dans `StaticMessages_language_code.properties`. Copiez également `ErrorMessages_en_US.properties` dans `ErrorMessages_language_code.properties`.
4. Modifiez le fichier `StaticMessages_language_code.properties` : recherchez `DatePattern` et changez-le en `DatePattern=dd/MM/yyyy` (la casse doit être respectée).

**Remarque :** Ce format est défini par Java™. Pour des détails complets sur le format, reportez-vous à la documentation Java de `java.text.SimpleDateFormat`, à l'adresse <http://java.sun.com>. Il n'est pas nécessaire de modifier le fichier `StaticMessages.properties`.

5. Pour WebSphere : recompresssez l'application Web en fichier jar.
6. Pour WebLogic : supprimez le module actuel de l'application Web.
  - a. Ajoutez le nouveau module.
  - b. Redéployez l'application Web.
  - c. Le redémarrage du programme d'écoute de Campaign n'est pas nécessaire.
7. Vérifiez que pour le paramètre de langue du navigateur Web, `language_code` est défini sur la première priorité. Pour plus de détails, voir les sections ci-dessous : Configuration de la langue appropriée pour votre navigateur Web et Configuration de votre ordinateur pour l'affichage d'une langue spécifique.

**Remarque :** Utilisez bien un tiret et non un trait de soulignement dans `language_code`. La configuration de l'application Web est le seul endroit où un tiret est utilisé à la place d'un trait de soulignement.

8. Connectez-vous à Campaign. Dans Campaign, les dates doivent être affichées au format spécifié dans `StaticMessages_language_code.properties`.

Pour plus d'informations sur la manière de configurer l'heure pour IBM Campaign, voir la documentation *IBM Campaign*.

## Exemples

Si l'heure est 10:54 a.m., `CURRENT_TIME()` renvoie la chaîne "10:54:00 AM".

## Fonctions associées

Fonction	Description
<code>CURRENT_DAY</code>	Renvoie le jour actuel sous la forme d'un nombre.
<code>CURRENT_JULIAN</code>	Renvoie le nombre au format Julien actuel.
<code>CURRENT_WEEKDAY</code>	Renvoie le jour de la semaine sous la forme d'un nombre.
<code>CURRENT_YEAR</code>	Renvoie l'année en cours sous la forme d'un nombre.

---

## CURRENT\_WEEKDAY

### Syntaxe

`CURRENT_WEEKDAY()`

### Description

`CURRENT_WEEKDAY` renvoie le jour de la semaine en cours sous la forme d'un nombre compris entre 0 et 6. Dimanche est représenté par 0, lundi par 1, etc.

Cette macro est disponible dans IBM Interact.

## Exemples

Si nous sommes vendredi, `CURRENT_WEEKDAY()` renvoie le nombre 5.

## Fonctions associées

Fonction	Description
<code>CURRENT_DAY</code>	Renvoie le jour actuel sous la forme d'un nombre.
<code>CURRENT_JULIAN</code>	Renvoie le nombre au format Julien actuel.
<code>CURRENT_MONTH</code>	Renvoie le mois actuel sous la forme d'un nombre.
<code>CURRENT_TIME</code>	Renvoie l'heure actuelle sous la forme d'une chaîne.
<code>CURRENT_YEAR</code>	Renvoie l'année en cours sous la forme d'un nombre.

---

## CURRENT\_YEAR

### Syntaxe

`CURRENT_YEAR()`

### Description

`CURRENT_YEAR` renvoie l'année en cours sous la forme d'un nombre.

Cette macro est disponible dans IBM Interact.

### Exemples

Si l'année en cours est 2000, `CURRENT_YEAR()` renvoie le nombre 2000.

## Fonctions associées

Fonction	Description
<code>CURRENT_DAY</code>	Renvoie le jour actuel sous la forme d'un nombre.
<code>CURRENT_JULIAN</code>	Renvoie le nombre au format Julien actuel.
<code>CURRENT_MONTH</code>	Renvoie le mois actuel sous la forme d'un nombre.
<code>CURRENT_TIME</code>	Renvoie l'heure actuelle sous la forme d'une chaîne.
<code>CURRENT_WEEKDAY</code>	Renvoie le jour de la semaine sous la forme d'un nombre.
<code>MONTHOF</code>	Renvoie le mois de l'année sous la forme d'un nombre.
<code>WEEKDAYOF</code>	Renvoie le jour de la semaine de la semaine sous forme de nombre.
<code>YEAROF</code>	Renvoie l'année sous la forme d'un nombre.



---

## CV\_FOLDS



### Syntaxe

```
CV_FOLDS(num_folds, data [, class_data] [seed])
```

### Paramètres

`num_folds`

Nombre de plis à créer pour la validation croisée. Cette valeur doit être un entier positif supérieur à 1. Cette valeur doit être inférieure à 65 536 ou au nombre de lignes de `data`, si ce dernier est inférieur à 65 536.

`data`

Variables en entrée. Il peut s'agir d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de `data`, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

`class_data`

Si cette plage de données facultative est fournie, la fonction macro `CV_FOLDS` crée des plis tout en conservant des probabilités de classe paires. Le contenu de `class_data` est utilisé comme sorties de chaque masque d'entrée correspondant

Si `class_data` est une colonne unique, `CV_FOLDS` suppose que la colonne spécifiée contient des valeurs pour plusieurs classes en sortie (à savoir, chaque valeur distincte est considérée comme une classe distincte). Si `class_data` est une plage de données, chaque colonne en sortie est considérée comme une classe différente. (Avec une plage de données, les valeurs de chaque colonne sont égales à un si un modèle appartient à cette classe et à zéro s'il n'y appartient pas.)

Pour la définition du format de `class_data` (identique à `data`), voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

`seed`

Valeur de départ à utiliser pour le générateur de nombre aléatoire. Il doit s'agir d'un entier.

### Description

`CV_FOLDS` divise équitablement les données d'entrée dans le nombre de plis spécifié. Chaque pli contient le même nombre de modèles d'entrée.<sup>2</sup> Il place chaque ligne de la plage de données d'entrée dans un pli en renvoyant une nouvelle colonne contenant les numéros de pli dont la valeur est comprise entre un et `num_folds`.

Si le paramètre facultatif `class_data` est fourni, les informations des classes en sortie sont utilisées pour créer des plis de validation croisée tels que les probabilités des classes en sortie soient conservées. C'est à dire que dans chaque pli, la probabilité de chaque classe en sortie sera la même.<sup>3</sup>

## Exemples

```
TEMP = CV_FOLDS(3, V1, 0)
```

Crée une colonne nommée TEMP contenant une valeur pour chaque ligne de la colonne V1. La colonne TEMP contient les valeurs 1, 2 et 3 pour les trois plis distincts. Aucune probabilité de classe n'est conservée. La valeur zéro est utilisée comme valeur de départ pour le générateur de nombre aléatoire.

```
TEMP = CV_FOLDS(100, V1:V15)
```

Crée une colonne nommée TEMP contenant une valeur pour chaque ligne de la colonne la plus courte dans V1-V15. La colonne TEMP contiendra les valeurs 1 à 100 pour les 100 différents plis. Aucune probabilité de classe n'est conservée. Une valeur aléatoire de départ est sélectionnée.

```
TEMP = CV_FOLDS(50, V1:V10, V11)
```

Crée une colonne nommée TEMP contenant une valeur pour chaque ligne de la colonne la plus courte dans V1-V10. La colonne TEMP contiendra les valeurs 1 à 50 pour les 50 différents plis. La colonne V11 contient les classes en sortie. Chaque pli possède les mêmes probabilités de classe en sortie. Une valeur aléatoire de départ est sélectionnée.

```
TEMP = CV_FOLDS(10, V1:V10, V11:V15, 96)
```

Crée une colonne nommée TEMP contenant une valeur pour chaque ligne de la colonne la plus courte dans V1-V10. La colonne TEMP contiendra les valeurs 1 à 10 pour les 10 différents plis. Chacune des colonnes en sortie V11-V15 représente une classe en sortie. Chaque pli possède les mêmes probabilités de classe en sortie. La valeur 96 est utilisée comme valeur de départ pour le générateur de nombre aléatoire.

---

## DATALINK



### Syntaxe

```
DATALINK([spreadsheet,] cells)
```

### Paramètres

spreadsheet

Nom de la feuille de calcul dans la session de travail IBM PredictiveInsight en cours vers laquelle un lien doit être créé (par exemple, Sheet1). Si aucune valeur n'est spécifiée, la feuille de calcul en cours est utilisée.

cells

cibles spécifiques de spreadsheet vers lesquelles le lien doit être créé. Le paramètre cells spécifie les colonnes et les lignes vers lesquelles le lien doit être créé et peut être spécifié de l'une des manières suivantes :

```
Cn Cn:Cm RnRn | CmRn RnRn:RNCM | CmRn:CMRN
```

Les caractères C et R spécifient respectivement la colonne et la ligne. Les variables n, m, N et M représentent les numéros de ligne et de colonne.

## Description

DATALINK crée un lien interne avec les données de la session de travail IBM PredictiveInsight en cours. Il renvoie le nombre de colonnes indiqué dans le paramètre `cells`. Les changements des colonnes de données source sont automatiquement reflétés.

## Exemples

<code>TEMP = DATALINK(C1)</code> Crée une colonne nommée TEMP contenant les valeurs de la colonne un de la feuille de calcul actuelle.
<code>TEMP = DATALINK(Sheet2, C1:C3)</code> Crée trois colonnes nommées TEMP, VX et VY, contenant les valeurs des colonnes 1 à 3 de la feuille de calcul <b>Sheet2</b> .
<code>TEMP = DATALINK(Sheet4, C5R10)</code> ou <code>TEMP = DATALINK(Sheet4, R10C5)</code> Crée une colonne nommée TEMP contenant la valeur de cible de la 5e colonne, 10e ligne de la feuille de calcul <b>Sheet4</b> .
<code>TEMP = DATALINK(Sheet1, C1R1:C3R500)</code> ou <code>TEMP = DATALINK(Sheet1, R1C1:R500C3)</code> Crée trois colonnes nommées TEMP, VX et VY, contenant les valeurs des lignes 1 à 500 des colonnes 1 à 3 de la feuille de calcul <b>Sheet2</b> .

## Fonctions associées

Fonction	Description
DDELINK	Crée un lien externe avec des données d'une autre application Windows

---

## DATE

### Syntaxe

`DATE(date_string [, format])`

### Paramètres

`date_string`

Texte représentant une date valide.

`format`

Un des mots clés du tableau sous "Mots clés de format valides", spécifiant le format de date de `date_string`.

### Description

DATE convertit une chaîne de date en date du calendrier Julien (nombre de jours écoulés depuis le 1er janvier 0000). Pratiquement tous les formats de date sont pris

en charge via le mot clé facultatif format, qui spécifie la manière dont la date est représentée. Si aucun mot clé format n'est spécifié, la valeur par défaut DELIM\_M\_D\_Y est utilisée.

Les formats de date sont de longueur fixe (par exemple, la date Février 28, 1970 est représentée sous la forme 02281970 au format MMDDYYYY) ou délimitée (par exemple, Février 28, 1970, 2-28-1970, ou 02/28/1970). Tous les exemples précédents sont des variantes du format DELIM\_M\_D\_Y.

Dans les formats délimités, les délimiteurs sont la barre oblique (/), le tiret (-), l'espace ( ), la virgule (,) et les deux-points (:); les années peuvent être représentées sur 2 ou 4 chiffres et les mois peuvent être indiqués en toutes lettres (par exemple, Février), abrégés (par exemple, Fév) ou sous forme numérique (par exemple, 2 ou 02).

Pour toutes les années spécifiées sur deux chiffres :

- Les années sur deux chiffres antérieures au seuil de l'an 2000 (la valeur par défaut est 20, mais elle peut être définie par l'utilisateur) sont considérées comme étant dans les années 2000.
- Les années sur deux chiffres postérieures ou égales au seuil sont considérées comme étant dans les années 1900.

Cette macro est disponible dans IBM Interact.

## Exemples

DATE("8/31/2000") renvoie le nombre 730729.

Mots clés de format valides

Mot clé	Description	Exemple(s)
MM	Mois sur deux chiffres	01, 02, 03, ..., 12
MMJJ	Mois sur deux chiffres et jour sur deux chiffres	31 mars correspond à 0331
MMJJAA	Mois sur deux chiffres, jour sur deux chiffres et année sur deux chiffres	31 mars 1970 correspond à 033170
MMJJAAAA	Mois sur deux chiffres, jour sur deux chiffres et année sur quatre chiffres	31 mars 1970 correspond à 03311970
DELIM_M_D	Tout mois délimité suivi du jour	31 mars, 3/31 ou 03-31
DELIM_M_D_Y	Toute combinaison de mois, jour et année délimitée	Mars 31, 1970 ou 3/31/70
DELIM_Y_M	Toute année délimitée suivie du mois	Mars 70 ; 3-70 ou 3/1970
DELIM_Y_M_D	Toute combinaison d'année, de mois et de jour délimitée	1970 Mar 31 ou 70/3/31
AAMMM	Année sur 2 chiffres et mois sur 3 lettres	70MAR
AAMMMJJ	Année sur 2 chiffres, mois sur 3 lettres et jour sur 2 chiffres	70MAR31

AA	Année sur 2 chiffres	70
AAMM	Année sur 2 chiffres et mois sur 2 chiffres	7003
AAMMJJ	Année sur 2 chiffres, mois sur 2 chiffres et jour sur 2 chiffres	700331
AAAAMMM	Année sur 4 chiffres et mois sur 3 lettres	1970MAR
AAAAMMJJ	Année sur 4 chiffres, mois sur 3 lettres et jour sur 2 chiffres	1970MAR31
AAAA	Année sur quatre chiffres	1970
AAAAMM	Année sur 4 chiffres et mois sur 2 chiffres	197003
AAAAMMDD	Année sur 4 chiffres, mois sur 2 chiffres et jour sur 2 chiffres	19700331
DELIM_M_Y	Tout mois délimité suivi de l'année	3-70, 3/70, Mar 70, Mars 1970
DELIM_D_M	Tout jour délimité suivi du mois	31-3, 31/3, 31 Mars
DELIM_D_M_Y	Toute combinaison de jour, de mois et d'année délimitée	31-MAR-70, 31/3/1970, 31 03 70
JJ	Jour sur 2 chiffres	31
JJMMM	Jour sur 2 chiffres et mois sur 3 lettres	31MAR
JJMMAA	Jour sur 2 chiffres, mois sur 3 lettres et année sur 2 chiffres	31MAR70
JJMMAAAA	Jour sur 2 chiffres, mois sur 3 lettres et année sur 4 chiffres	31MAR1970
JJMM	Jour sur 2 chiffres et mois sur 2 chiffres	3103
JJMMAA	Jour sur 2 chiffres, mois sur 2 chiffres et année sur 2 chiffres	310370
JJMMAAAA	Jour sur 2 chiffres, mois sur 2 chiffres et année sur 4 chiffres	31031970
MMAA	Mois sur deux chiffres et année sur deux chiffres	0370
MMAAAA	Mois sur deux chiffres et année sur 4 chiffres	031970
MMM	Mois sur 3 lettres	MAR
MMMJJ	Mois sur 3 lettres et jour sur 2 chiffres	MAR31
MMMJJAA	Mois sur 3 lettres, jour sur 2 chiffres et année sur 2 chiffres	MAR3170

MMMJJAAAA	Mois sur 3 lettres, jour sur 2 chiffres et année sur 4 chiffres	MAR311970
MMMAA	Mois sur 3 lettres et année sur 2 chiffres	MAR70
MMMAAAA	Mois sur 3 lettres et année sur 4 chiffres	MAR1970
MONTH	Mois de l'année	Janvier, Février, Mars et ainsi de suite ou Jan, Fév, Mar, etc.
WEEKDAY	Jour de la semaine	Dimanche, Lundi, Mardi, etc. (Dimanche = 0)
WKD	Jour abrégé de la semaine	Dim, Lun, Mar, etc. (Dim = 0)

### Fonctions associées

Fonction	Description
DATE_FORMAT	Convertit les dates d'un format à un autre.
DATE_JULIAN	Renvoie la date du calendrier Julien de l'entrée.
DATE_STRING	Renvoie la chaîne de date sous la forme d'une date du calendrier Julien.
CURRENT_DATE	Renvoie la date du jour dans un format spécifié.

---

## DATE\_FORMAT

### Syntaxe

DATE\_FORMAT(date\_string, input\_format, output\_format)

### Paramètres

date\_string

Texte représentant une date valide.

input\_format

Un des mots clés du tableau ci-après spécifiant le format de date de date\_string.

output\_format

Un des mots clés du tableau ci-après spécifiant le format de date en sortie souhaité.

### Description

DATE\_FORMAT() convertit une date de input\_format dans un autre format output\_format.

Si la date est de largeur fixe, elle doit être définie dans l'un des formats suivants :

- DDMMYY[YY]
- DDMMMYY[YY]
- MMDDYY[YY]
- MMMDDYY[YY]
- YY[YY]MMDD
- YY[YY]MMMDD

MM correspond à un mois sur deux chiffres et MMM représente l'abréviation du mois sur trois caractères

Si la date est délimitée (tout délimiteur et notamment SPACE, DASH ou SLASH peut être utilisé), elle doit être définie dans l'un des formats suivants :

- DELIM\_D\_M\_Y
- DELIM\_M\_D\_Y
- DELIM\_Y\_M\_D

Cette macro est disponible dans IBM Interact.

## Exemples

DATE\_FORMAT("012171", MMDDYY, MMDDYYYY) renvoie la chaîne "01211971".

**Remarque :** Pour des informations supplémentaires sur les formats de date valides, voir «DATE», à la page 77.

Fonctions associées

Fonction	Description
DATE	Convertit une chaîne de date en date du calendrier Julien.
DATE_JULIAN	Renvoie la date du calendrier Julien de l'entrée.
DATE_STRING	Renvoie la chaîne de date sous la forme d'une date du calendrier Julien.

---

## DATE\_JULIAN

### Syntaxe

DATE\_JULIAN(year, month, day)

### Paramètres

year

Numéro d'année valide sur 2 ou 4 chiffres.

month

Numéro de mois valide compris entre 1 et 12.

day

Numéro de jour valide compris entre 1 et 31.

## Description

DATE\_JULIAN envoie la date du calendrier Julien pour l'entrée spécifiée. La date au format julien correspond au nombre de jours écoulés depuis le 1er janvier 0000.

## Exemples

DATE\_JULIAN (2000,08,31) renvoie le nombre 730729.

## Fonctions associées

Fonction	Description
DATE	Convertit une chaîne de date en date du calendrier Julien.
DATE_FORMAT	Convertit les dates d'un format à un autre.
DATE_STRING	Renvoie la chaîne de date sous la forme d'une date du calendrier Julien.

---

## DATE\_STRING

### Syntaxe

```
DATE_STRING(julian_date [, 'output_format'[, max_length]])  
DATE_STRING(julian_date [, 'format_string'[, max_length]])
```

### Paramètres

julian\_date

Nombre représentant une date au format julien ; nombre de jours écoulés depuis le 1er janvier 0000.

output\_format

Chaîne ; format de date valide.

max\_length

format\_string

Chaîne de format incluant éventuellement

Code	Description
%a	Nom abrégé du jour de la semaine
%A	Nom complet du jour de la semaine
%b	Nom abrégé du mois
%B	Nom complet du mois
%c	Représentation de la date et de l'heure conforme aux paramètres régionaux



Code	Description
%d	Jour du mois (01 - 31)
%H	Heure au format 24 heures (00 - 23)
%I	Heure au format 12 heures (01 - 12)
%j	Jour de l'année (001 - 366)
%m	Mois (01 - 12)
%M	Minute (00 - 59)
%p	Indicateur AM/PM des paramètres régionaux actuels pour une horloge au format 12 heures
%S	Secondes (00 - 59)
%U	Semaine de l'année, le dimanche étant considéré comme le premier jour de la semaine (00-51)
%w	Jour de la semaine (0 - 6 ; dimanche correspond à 0)
%W	Semaine de l'année, le lundi étant considéré comme le premier jour de la semaine (00-51)
%x	Représentation de la date pour les paramètres régionaux actuels
%X	Représentation de l'heure pour les paramètres régionaux actuels
%y	Année sur 2 chiffres (00-99)
%Y	Année sur quatre chiffres
%z, %Z	Nom ou abréviation du fuseau horaire ; aucune sortie si le fuseau horaire est inconnu
%%	Signe pourcentage

## Description

DATE\_STRING renvoie la chaîne de date sous la forme d'une date du calendrier Julien. Si output\_format n'est pas fourni, le mot clé par défaut DELIM\_M\_D\_Y est utilisé.

## Exemples

DATE\_STRING(730729) renvoie la chaîne "08/31/00".

**Remarque :** Pour des informations supplémentaires sur les formats de date valides, voir «DATE», à la page 77.

## Fonctions associées

Fonction	Description
DATE	Convertit une chaîne de date en date du calendrier Julien.
DATE_JULIAN	Renvoie la date du calendrier Julien de l'entrée.

Fonction	Description
DATE_FORMAT	Convertit les dates d'un format à un autre.

---

## DAY\_BETWEEN

### Syntaxe

DAY\_BETWEEN(from\_date\_string, to\_date\_string [, input\_format])

### Paramètres

from\_date\_string

Texte représentant une date valide à partir de laquelle le nombre de jours écoulés doit être calculé.

to\_date\_string

Texte représentant une date valide jusqu'à laquelle le nombre de jours écoulés est calculé. Cette date doit être dans le même format que from\_date\_string.

input\_format

Un des mots clés du tableau ci-après, spécifiant le format de date de from\_date\_string et to\_date\_string.

### Description

DAY\_BETWEEN renvoie le nombre de jours compris entre from\_date\_string et to\_date\_string. Si input\_format n'est pas fourni, le mot clé par défaut DELIM\_M\_D\_Y est utilisé.

### Exemples

DAY\_BETWEEN("08/25/00","08/31/00") renvoie le nombre 6.

**Remarque :** Pour des informations supplémentaires sur les formats de date valides, voir «DATE», à la page 77.

### Fonctions associées

Fonction	Description
DAY_FROMNOW	Renvoie le nombre de jours entre le jour actuel et une date spécifiée.
DAY_INTERVAL	Renvoie le nombre de jours entre deux dates spécifiées.

---

## DAY\_FROMNOW

### Syntaxe

DAY\_FROMNOW(to\_year, to\_month, to\_day)

## Paramètres

to\_year

Numéro d'année valide sur 2 ou 4 chiffres.

to\_month

Numéro de mois valide compris entre 1 et 12.

to\_day

Numéro de jour valide compris entre 1 et 31.

## Description

DAY\_FROMNOW renvoie le nombre de jours entre la date actuelle et la date spécifiée par to\_year/to\_month/to\_day.

**Remarque :** Si la date spécifiée est antérieure, la valeur renvoyée est négative.

## Exemples

Si la date du jour est le 31 août 2000, DAY\_FROMNOW(2000,12,31) renvoie le nombre 122.

## Fonctions associées

Fonction	Description
DAY_BETWEEN	Renvoie le nombre de jours entre deux chaînes de date spécifiées.
DAY_INTERVAL	Renvoie le nombre de jours entre deux dates spécifiées.

---

## DAY\_INTERVAL

### Syntaxe

DAY\_INTERVAL(from\_year, from\_month, from\_day, to\_year, to\_month, to\_day)

### Paramètres

from\_year

Numéro d'année valide sur 2 ou 4 chiffres.

from\_month

Numéro de mois valide compris entre 1 et 12.

from\_day

Numéro de jour valide compris entre 1 et 31.

to\_year

Numéro d'année valide sur 2 ou 4 chiffres.

to\_month

Numéro de mois valide compris entre 1 et 12.

to\_day

Numéro de jour valide compris entre 1 et 31.

## Description

DAY\_INTERVAL renvoie le nombre de jours entre la date de début spécifiée (from\_year/from\_month/from\_day) et la date de fin spécifiée (to\_year/to\_month/to\_day).

## Exemples

DAY\_INTERVAL(2000,8,31,2000,12,31) renvoie le nombre 122.

## Fonctions associées

Fonction	Description
DAY_BETWEEN	Renvoie le nombre de jours entre deux chaînes de date spécifiées.
DAY_FROMNOW	Renvoie le nombre de jours entre le jour actuel et une date spécifiée.

---

## DAYOF

### Syntaxe

DAYOF(date\_string [, input\_format])

### Paramètres

date\_string

Texte représentant une date valide.

input\_format

Un des mots clés du tableau ci-après, spécifiant le format de date de date\_string.

### Description

DAYOF renvoie le jour du mois sous la forme d'un nombre pour la date représentée par date\_string. Si input\_format n'est pas fourni, le mot clé par défaut DELIM\_M\_D\_Y est utilisé.

### Exemples

DAYOF("08/31/00") renvoie le nombre 31.

**Remarque :** Pour des informations supplémentaires sur les formats de date valides, voir «DATE», à la page 77.

---

## DDELINK



### Syntaxe

DDELINK(service, topic, items)

### Paramètres

service

Nom du service (par exemple, excel) avec lequel un lien DDE doit être créé.

topic

Rubrique dans service à laquelle le lien doit permettre d'accéder. Pour la plupart des applications, la rubrique est un nom de fichier. Entrez le chemin d'accès complet et le nom de la rubrique souhaitée (par exemple, c:\stock\prices\05jan.xls).

items

Éléments dans topic auxquels le lien doit permettre d'accéder. La syntaxe des éléments dépend du service sélectionné. Par exemple, dans Excel, R1C1:R10C20 sélectionne les lignes 1 à 10 et les colonnes 1 à 20.

### Description

DDELINK crée un lien DDE (Dynamic Data Exchange) avec les données d'une application Windows externe. Il renvoie le nombre de colonnes indiqué dans le paramètre items. Les changements apportés à l'application source sont automatiquement reflétés dans la feuille de calcul IBM PredictiveInsight.

### Exemples

```
TEMP = DDELINK(Excel, c:\excel\data.xls, C1:C2)
```

Crée deux colonnes nommées TEMP et VX, contenant les valeurs des colonnes 1 à 2 de la feuille de calcul c:\excel\data.xls.

```
TEMP = DDELINK(Excel, c:\excel\data.xls, R1:R10)
```

Crée autant de nouvelles colonnes que nécessaire pour accueillir les lignes 1 à 10 de la feuille de calcul c:\excel\data.xls.

```
TEMP = DDELINK(Excel, c:\excel\data.xls, R1C1:R100C3)
```

Crée trois colonnes nommées TEMP, VX et VY contenant les lignes 1 à 100 de la feuille de calcul c:\excel\data.xls.

```
TEMP = DDELINK(123W, c:\lotsuite\sample.wk4, A:A1..A:C8)
```

Crée trois colonnes nommées TEMP, VX et VY contenant les lignes 1 à 8 des colonnes A à C de la feuille de calcul A dans le fichier Lotus sample.wk4.

## Fonctions associées

Fonction	Description
DATALINK	Crée un lien interne avec les données d'une feuille de calcul IBM PredictiveInsight

---

## DECIMATE



### Syntaxe

DECIMATE(column, max\_value)

### Paramètres

column

Colonne de valeurs à décimer. Toutes les valeurs de cette colonne doivent être des entiers positifs inférieurs à max\_value.

max\_value

Nombre de colonnes à renvoyer. Il doit s'agir d'un entier positif supérieur ou égal à la valeur maximale de column.

### Description

DECIMATE convertit un entier positif en un modèle binaire de max\_value colonnes de long. Si la valeur est *n*, la *n*ème colonne contient un un ; toutes les autres colonnes contiennent des zéros. Cette fonction macro renvoie max\_value colonnes.

**Remarque :** DECIMATE est l'opposé de la fonction macro MAXINDEX.

### Exemples

```
TEMP = DECIMATE(COLUMN(1,2,3), 3)
```

Crée des colonnes nommées TEMP, VX et VY avec une ligne pour chaque ligne en entrée correspondante. La première ligne contient 1 0 0, la deuxième contient 0 1 0 et la troisième contient 0 0 1.

```
TEMP = DECIMATE(COLUMN(1,1,2), 3)
```

Crée des colonnes nommées TEMP, VX et VY avec une ligne pour chaque ligne en entrée correspondante. La première ligne contient 1 0 0, la deuxième contient 1 0 0 et la troisième contient 0 1 0.

```
TEMP = DECIMATE(V1, 10)
```

Crée dix colonnes avec une ligne pour chaque ligne en entrée correspondante. Chaque ligne contient un un dans la colonne représentant la valeur en entrée correspondante. Toutes les autres colonnes contiennent des zéros.

## Fonctions associées

Fonction	Description
MAX	Calcule la valeur maximale d'une plage de cibles
MAXINDEX	Renvoie les valeurs des colonnes en entrée décalées du nombre d'intervalles de temps spécifié
MIN	Calcule la valeur minimale d'une plage de cibles

---

## DELAY



### Syntaxe

DELAY(delay, data)

### Paramètres

delay

Nombre d'intervalles de temps à retarder. Cette valeur doit correspondre à un entier positif.

data

Valeurs à retarder. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de data, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

### Description

DELAY renvoie les valeurs de la plage de données d'entrée, retardées du nombre d'intervalles de temps spécifié. Il considère chaque colonne d'entrée comme une série de données dans le temps et renvoie une nouvelle colonne pour chaque colonne d'entrée. Chaque nouvelle colonne contient les valeurs retardées dans le temps (retard de delay nombre d'intervalles de temps) des nombres de la colonne d'entrée correspondante.

**Remarque :** La fonction macro DELAY renvoie une colonne contenant des valeurs telles que la cible  $VY[x] = data[x + delay]$ .

**Remarque :** Cette fonction est utile pour créer des modèles à partir de données de série temporelle. Pour créer plusieurs retards, utilisez la fonction macro SLIDE\_WINDOW.

## Exemples

TEMP = DELAY(1, COLUMN(1,2,3,4))

Crée une colonne nommée TEMP contenant les valeurs 2, 3 et 4 dans les cibles 1 à 3, respectivement.

TEMP = DELAY(2, V1)

Crée une colonne nommée TEMP, dans laquelle chaque valeur correspond au contenu de la colonne V1 retardé de deux intervalles de temps.

TEMP = DELAY(10, V1:V3)

Crée trois colonnes nommées TEMP, VX et VY. Les valeurs de la colonne TEMP correspondent au contenu de la colonne V1 retardé de dix intervalles de temps, les valeurs de la colonne VX correspondent au contenu de la colonne V2 retardé de dix intervalles de temps et les valeurs de la colonne VY correspondent au contenu de la colonne V3 retardé de dix intervalles de temps.

TEMP = DELAY(1, V1[10:20])

Crée une colonne nommée TEMP, dans laquelle les 10 premières cibles contiennent les valeurs de la colonne V1 retardés d'un intervalle de temps (à savoir, les lignes 11 à 20 de la colonne V1). Les autres cibles de TEMP sont vides.

TEMP = DELAY(2, V1[1:5]:V2)

Crée deux colonnes nommées TEMP et VX, contenant des valeurs dans les lignes 1 à 3 (les autres cibles sont vides). Les valeurs de la colonne TEMP sont égales aux lignes correspondantes de la colonne V1 retardées de deux intervalles de temps et les valeurs de la colonne VX sont égales aux lignes correspondantes de la colonne V2 retardées de deux intervalles de temps (à savoir, les lignes 3 à 5 des colonnes V1 et V2).

## Fonctions associées

Fonction	Description
LAG	Renvoie les valeurs des colonnes en entrée décalées du nombre d'intervalles de temps spécifié
SLIDE_WINDOW	Crée un modèle à partir d'une fenêtre spécifiée et le fait glisser pour créer le modèle suivant

---

## DERIVATIVE



### Syntaxe

DERIVATIVE(data [, divisor])

### Paramètres

data

Valeurs numériques dont la dérivée doit être calculée. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie



comme l'une de ces valeurs. Pour la définition du format de data, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

divisor

Valeur par laquelle chaque valeur de data doit être divisée. Il peut s'agir d'une valeur constante ou d'une expression convertie en constante.

## Description

DERIVATIVE calcule la dérivée des valeurs d'une série chronologique. Chaque valeur correspond à la différence entre la valeur actuelle et la valeur de la prochaine étape. Si une valeur est indiquée pour divisor, chaque valeur est divisée par la valeur spécifiée. DERIVATIVE renvoie une nouvelle colonne par colonne d'entrée, chacune contenant la dérivée des valeurs de la colonne d'entrée correspondante.

**Remarque :** La longueur de la colonne renvoyée est d'une unité inférieure à la longueur de la colonne de données source (data).

## Exemples

TEMP = DERIVATIVE(5)
Crée une colonne nommée TEMP ne contenant que des cibles vides (au moins deux valeurs de cible sont requises pour générer un résultat).
TEMP = DERIVATIVE(COLUMN(1, 2, 5))
Crée une colonne nommée TEMP contenant les valeurs 1 et 3.
TEMP = DERIVATIVE(V1)
Crée une colonne nommée TEMP, dans laquelle chaque valeur correspond à la dérivée du contenu de la colonne V1.
TEMP = DERIVATIVE(V1:V3)
Crée trois colonnes nommées TEMP, VX et VY. Les valeurs de la colonne TEMP correspondent aux dérivées du contenu de la colonne V1, les valeurs de la colonne VX correspondent aux dérivées du contenu de la colonne V2 et les valeurs de la colonne VY correspondent aux dérivées du contenu de la colonne V3.
TEMP = DERIVATIVE(V1[10:20])
Crée une colonne nommée TEMP, dans laquelle les cibles des lignes 10 à 20 contiennent les dérivées des lignes correspondantes de la colonne V1. Les autres cibles de TEMP sont vides.
TEMP = DERIVATIVE(V1[1:5]:V2)
Crée deux colonnes nommées TEMP et VX, contenant des valeurs dans les lignes 1 à 5 (les autres cibles sont vides). Les valeurs de la colonne TEMP correspondent aux dérivées des lignes correspondantes de la colonne V1 et les valeurs de la colonne VX correspondent aux dérivées des lignes correspondantes de la colonne V2.

## Fonctions associées

Fonction	Description
INTEGRAL	Calcule l'intégrale des valeurs de la plage de données spécifiée

---

## DISTANCE

### Syntaxe

`DISTANCE(lat1,long1,lat2,long2[,UNITE_DE_MESURE][,PRECISION])`

### Paramètres

`lat1`

Latitude du premier point, sous forme de valeur décimale.

`long1`

Longitude du premier point, sous forme de valeur décimale.

`lat2`

Latitude du deuxième point, sous forme de valeur décimale.

`long2`

Longitude du deuxième point, sous forme de valeur décimale.

`UNITE_DE_MESURE`

Paramètre facultatif indiquant l'unité de mesure de la distance renvoyée. Les valeurs admises sont MILES et KILOMETERS. Si vous omettez ce paramètre, MILES est l'unité par défaut.

`PRECISION`

Paramètre facultatif indiquant le niveau de précision après le séparateur décimal pour la distance renvoyée. Si vous spécifiez une valeur de précision, la valeur de la distance renvoyée est tronquée en fonction du nombre de décimales indiqué. La valeur maximale est 5. Si vous omettez cette valeur, les décimales ne sont pas tronquées.

### Description

DISTANCE calcule la distance entre deux points. La latitude et la longitude doivent être exprimées sous forme d'unités décimales.

### Exemples

`DISTANCE (18.529747,73.839798,18.533511,73.8777995,MILES,2)` renvoie la valeur 2,5 miles.

---

## DISTINCT



### Syntaxe

`DISTINCT(data)`

## Paramètres

data

Plage de données.

## Description

DISTINCT recherche les valeurs uniques de la plage de données spécifiée. Il renvoie cette liste de valeurs dans une colonne unique. DISTINCT ignore la casse lorsqu'il compare les chaînes de texte. Par exemple, \042Aaa\042, \042aAa\042 et \042AAA\042 sont tous considérés comme la même valeur.

Le calcul effectué par la fonction macro DISTINCT peut être assez long si la quantité de données est importante. Une barre de progression \042Computing...\042 est affichée jusqu'à la fin du calcul. Si vous décidez d'annuler le calcul, cliquez sur le \042X\042 dans la barre de progression et supprimez la définition de fonction contenant la fonction macro DISTINCT.

---

## DIV

### Syntaxe

```
data DIV divisor data / divisor
```

### Paramètres

data

Valeurs numérique dans lesquelles la division doit être effectuée. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de data, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

divisor

Valeurs par lesquelles les valeurs de la plage de données spécifiée doivent être divisées. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Le nombre de colonnes dans data2 doit correspondre au nombre de colonnes dans data1, à moins que data2 ne soit une constante. Pour la définition du format de divisor (identique à data), voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

### Description

DIV divise la plage de données spécifiée par la valeur du diviseur. Il renvoie une nouvelle colonne pour chaque colonne d'entrée, chacune contenant la colonne correspondante de data1 divisée par la colonne correspondante de data2 (à savoir, la première colonne de data1 est divisée par la première colonne de data, la deuxième colonne à la deuxième colonne et ainsi de suite).

Si data2 est une constante, chaque valeur de data1 est divisée par cette valeur. Si data2 contient une ou plusieurs colonnes, les calculs sont effectués ligne par ligne entre une colonne de data1 et une colonne de data2. La première ligne de data1

est divisée par la première ligne de data2, la seconde ligne, par la seconde ligne, etc. Ce calcul ligne par ligne génère un résultat pour chaque ligne jusqu'à la dernière valeur de la colonne la plus courte.

**Remarque :** L'utilisation d'une colonne contenant le même nombre  $x$  dans chaque ligne que divisor revient à utiliser la constante  $x$  pour divisor.

**Remarque :** L'opérateur DIV peut être abrégé en barre oblique (/).

Cette macro est disponible dans IBM Interact.

## Exemples

TEMP = 8 DIV 4 ou TEMP = 8/4
Crée une colonne nommée TEMP contenant la valeur deux.
TEMP = V1/8
Crée une colonne nommée TEMP, dans laquelle chaque valeur correspond au contenu de la colonne V1 divisé par huit.
TEMP =V1:V3/2
Crée trois colonnes nommées TEMP, VX et VY. Les valeurs de la colonne TEMP correspondent au contenu de la colonne V1 divisé par deux, les valeurs de la colonne VX correspondent au contenu de la colonne V2 divisé par deux et les valeurs de la colonne VY correspondent au contenu de la colonne V3 divisé par deux.
TEMP = V1/V1
Crée une colonne nommée TEMP ne contenant que des uns (tout nombre divisé par lui-même étant égal à un).
TEMP = V1/V2
Crée une colonne nommée TEMP, dans laquelle chaque valeur est la valeur de ligne de la colonne V1 divisée par la valeur de ligne correspondante de la colonne V2.
TEMP = V1:V3/V4:V6
Crée trois colonnes nommées TEMP, VX et VY. La colonne TEMP contient les valeurs de V1 divisées par les valeurs de ligne correspondantes de la colonne V4. La colonne VX contient la division de la colonne V2 par V5. La colonne VY contient la division de la colonne V3 par V6.
TEMP = V1[10:20] / V2 ou TEMP = V1[10:20] / V2[1:11]
Crée une colonne nommée TEMP, dans laquelle les 11 premières cibles contiennent le résultat de la division des valeurs des lignes 10 à 20 de la colonne V1 par les valeurs des lignes 1 à 11 de la colonne V2. Les autres cibles de TEMP sont vides.

## Fonctions associées

Fonction	Description
MOD	Calcule le modulo du contenu de la plage de données spécifiée
MULT	Multiplie le contenu de deux plages de données
POW	Calcule une valeur de base élevée aux puissances exponentielles spécifiées

## Syntaxe

`data1 EQ data2 data1 == data2 (data1 = data2)`

## Paramètres

`data1`

Plage de cibles à comparer. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de data, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

`data2`

Nombres auxquels toutes les valeurs de la colonne spécifiée doivent être comparées. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Le nombre de colonnes dans data2 doit correspondre au nombre de colonnes dans data1, à moins que data2 ne soit une constante. Pour la définition du format de data, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

## Description

EQ compare les deux plages de données spécifiées, puis renvoie un un si les valeurs sont égales ou un zéro si elles ne le sont pas. Il renvoie une nouvelle colonne pour chaque colonne d'entrée, chacune contenant la colonne correspondante de data1 comparée à la colonne correspondante de data2 (à savoir, la première colonne de data1 est comparée à la première colonne de data, la deuxième colonne à la deuxième colonne et ainsi de suite).

Si data2 est une constante, chaque valeur de data est comparée à cette valeur. Si data2 est une colonne, les calculs sont effectués ligne par ligne. Les valeurs de data1 sont comparées à la valeur de la première ligne de data2, la deuxième ligne à la deuxième ligne et ainsi de suite. Ce calcul ligne par ligne génère un résultat pour chaque ligne jusqu'à la dernière valeur de la colonne la plus courte.

Lors de la comparaison de chaînes, la casse n'est pas importante ("Oui", "OUI", "oui" et "ouI" sont considérées comme étant égales).

**Remarque :** L'opérateur EQ peut être abrégé en double signe égal (==). A l'intérieur des parenthèses, un signe égal unique (=) peut également être utilisé pour la fonction macro EQ (en dehors des parenthèses, e signe égal est interprété comme l'opérateur d'affectation).

Cette macro est disponible dans IBM Interact.

## Exemples

```
TEMP = 3 EQ 4, TEMP = 3==4 ou TEMP = (3=4)
```

Crée une colonne nommée TEMP contenant la valeur zéro (car trois n'est pas égal à quatre).

TEMP = "No" == "NO"
Crée une colonne nommée TEMP contenant la valeur un (la casse n'est pas importante dans les comparaisons de chaînes).
TEMP = V1 == 8
Crée une colonne nommée TEMP, dans laquelle chaque valeur est égale à un si la ligne correspondante de la colonne V1 est égale à huit ; sinon, la valeur est égale à zéro.
TEMP = V1==V1
Crée une colonne nommée TEMP ne contenant que des uns (car tout nombre est égal à lui-même).
TEMP = V1==V2
Crée une colonne nommée TEMP, dans laquelle chaque valeur est la valeur de ligne de la colonne V1 comparée à la valeur de ligne correspondante de la colonne V2.
TEMP = V1:V3 == V4:V6
Crée trois colonnes nommées TEMP, VX et VY. La colonne TEMP contient les valeurs de V1 comparées aux valeurs de ligne correspondantes de la colonne V4. La colonne VX compare les colonnes V2 et V5. La colonne VY compare les colonnes V3 et V6.
TEMP = V1[10:20] == V2 ou TEMP = V1[10:20] == V2[1:11]
Crée une colonne nommée TEMP, dans laquelle les 11 premières cibles contiennent les résultats de la comparaison des valeurs des lignes 10 à 20 de la colonne V1 aux lignes 1 à 11 de la colonne V2. Les autres cibles de TEMP sont vides.

## Fonctions associées

Fonction	Description
EQ	Renvoie TRUE si une plage de données est égale à une autre
GE	Renvoie TRUE si une plage de données est supérieure ou égale à une autre
GT	Renvoie TRUE si une plage de données est supérieure à une autre
LE	Renvoie TRUE si une plage de données est inférieure ou égale à une autre
LT	Renvoie TRUE si une plage de données est inférieure à une autre
NE	Renvoie TRUE si une plage de données n'est pas égale à une autre

---

## EXP

### Syntaxe

EXP(data)

### Paramètres

data

Valeurs numériques utilisées comme exposant du nombre naturel  $e$ . Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de data, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

## Description

EXP élève le nombre naturel  $e$  de chacune des valeurs de la plage de données spécifiée (il calcule  $e^x$ ). La constante  $e$  est égale à 2,7182818. EXP renvoie une nouvelle colonne par colonne d'entrée, chacune contenant le résultat  $e^x$  pour chaque valeur  $x$  des colonnes en entrée correspondantes. EXP est l'inverse de la fonction macro LN.

**Remarque :** Si la valeur  $x$  est trop faible, une erreur de dépassement est renvoyée. Cela se produit si  $e^x$  dépasse la valeur à virgule flottante sur 32 bits maximale ou minimale.

## Exemples

TEMP = EXP(2)	Crée une colonne nommée TEMP contenant la valeur 7.39.
TEMP = EXP(V1)	Crée une colonne nommée TEMP, dans laquelle chaque valeur correspond à l'élévation de $e$ au contenu de la colonne V1.
TEMP = EXP(V1:V3)	Crée trois colonnes nommées TEMP, VX et VY. Les valeurs de la colonne TEMP correspondent à l'élévation de $e$ au contenu de la colonne V1, les valeurs de la colonne VX correspondent à l'élévation de $e$ au contenu de la colonne V2 et les valeurs de la colonne VY correspondent à l'élévation de $e$ au contenu de la colonne V3.
TEMP = EXP(V1[10:20])	Crée une colonne nommée TEMP, dans laquelle les 11 premières cibles contiennent les résultats de l'élévation de $e$ aux valeurs des lignes 10 à 20 de la colonne V1. Les autres cibles de TEMP sont vides.
TEMP = EXP(V1[1:5]:V2)	Crée deux colonnes nommées TEMP et VX, contenant des valeurs dans les lignes 1 à 5 (les autres cibles sont vides). Les valeurs de la colonne TEMP correspondent aux résultats de l'élévation de $e$ aux valeurs de ligne correspondantes de la colonne V1 et les valeurs de la colonne VX correspondent aux résultats de l'élévation de $e$ aux valeurs de ligne correspondantes de la colonne V2.

## Fonctions associées

Fonction	Description
LN ou LOG	Calcule le logarithme naturel du contenu de la plage de données spécifiée
LOG2	Calcule le logarithme base 2 du contenu de la plage de données spécifiée
LOG10	Calcule le logarithme base 10 du contenu de la plage de données spécifiée

Fonction	Description
POW	Puissance exponentielle

---

## EXTERNALCALLOUT



### Syntaxe

```
EXTERNALCALLOUT( calloutName, arg1, ...)
```

### Paramètres

*calloutName*

Nom de l'appel que vous avez créé à l'aide de l'API ExternalCallout. Ce nom doit correspondre au nom de la catégorie d'appels externes que vous avez créée dans la plateforme marketing IBM.

*arg1*

Argument requis par votre appel, si nécessaire.

### Description

EXTERNALCALLOUT permet d'appeler une application externe pour ajouter des données à votre diagramme temps réel. EXTERNALCALLOUT peut renvoyer tout ce pourquoi l'appel a été créé. Vous devez écrire cet appel dans Java à l'aide de l'API ExternalCallout. Pour plus d'informations, voir le Guide du développeur d'IBM Interact.

### Exemples

```
EXTERNALCALLOUT(getStockPrice, UNCA)
```

Appelle l'appel `getStockPrice` en transmettant le nom de l'action, `UNCA`, comme argument. Cet appel défini par l'utilisateur renvoie le cours de l'action tel qu'ils est défini par l'appel.

---

## EXTRACT



### Syntaxe

```
EXTRACT(predicate_col, data)
```

### Paramètres

*predicate\_col*

Colonne de valeurs booléennes ou expression convertie en une colonne de valeurs booléennes. Les valeurs booléennes sont interprétées comme des nulles ou non



nulles. Cette colonne doit contenir au moins autant de lignes que la plage de données de laquelle les données sont extraites. Sinon, `predicate_col` constitue une limite du nombre de lignes traitées par la fonction macro EXTRACT (voir "Description" ci-dessous).

`data`

Valeurs à extraire. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de `data`, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

## Description

EXTRACT renvoie les lignes de la plage de données spécifiée qui possèdent la valeur un dans la ligne correspondante de la colonne de prédicat. Cette fonction macro réduit la quantité de données en "excluant" toutes les lignes dont la valeur correspondante dans `predicate_col` est zéro. EXTRACT renvoie une nouvelle colonne par colonne d'entrée, chacune contenant les valeurs de la colonne d'entrée correspondante pour laquelle la valeur correspondante dans `predicate_col` est un. Les lignes de données extraites occupent les  $n$  premières cibles des colonnes en sortie, où  $n$  représente le nombre de uns dans `predicate_col`.

EXTRACT fonctionnant ligne par ligne, il génère un résultat pour chaque ligne jusqu'à la dernière valeur de la colonne la plus courte (c'est-à-dire la colonne la plus courte entre `predicate_col` et les colonnes de la plage de données `data`). Toutes les valeurs non nulles de `predicate_col` possèdent la valeur 1.

**Remarque :** En général, une colonne de prédicat est créée à l'aide de l'une des fonctions macro de comparaison (par exemple, `==`, `>`, `<`, `ISEVEN`, `ISODD`, etc.). Vous pouvez ensuite extraire les lignes qui vous intéressent d'une plage de données spécifiée à l'aide de la fonction macro EXTRACT. Cela peut être utile pour "éliminer" les points de données incorrects (par exemple, lorsqu'une valeur particulière dépasse la valeur maximale ou minimale d'une variable de données). Cela permet également de consolider tous les exemple d'une classe particulière (par exemple, si la colonne V3 contient des uns et des zéros pour l'une des classes en sortie, utilisez `V4=EXTRACT(V3, V1:V2)` pour extraire les entrées V1 et V2). EXTRACT condensant toutes les lignes extraites comme un bloc de données (à savoir, il remplit la plage de cibles `VX[1:n]:VY`), où  $n$  représente le nombre de lignes extraites, cette fonction est utile pour copier une plage de cibles de leurs emplacements de ligne actuels vers les lignes 1 à  $n$  de la feuille de calcul.

## Exemples

```
TEMP = EXTRACT(1, V1)Crée une colonne nommée TEMP contenant une copie de la colonne V1.
```

```
TEMP = EXTRACT(1, V1[50:100]:V2)
```

Crée deux colonnes nommées TEMP et VX contenant des valeurs dans les 51 premières cibles. Les valeurs de la colonne TEMP correspondent aux cibles 50 à 100 de la colonne V1 et les valeurs de la colonne VX correspondent aux cibles 50 à 100 de la colonne V2.

TEMP = EXTRACT(V3, V1:V2)
Crée deux colonnes nommées TEMP et VX. Pour chaque ligne dont la valeur de la colonne V3 est un, la ligne correspondante dans les colonnes V1 et V2 est extraite dans les colonnes TEMP et VX, respectivement.
TEMP = EXTRACT(V1>V2, V1)
Crée une colonne nommée TEMP contenant toutes les valeurs de la colonne V1 supérieures aux valeurs correspondantes dans la colonne V2.
TEMP = EXTRACT(V3[10:20], V1[10:20]:V2)
Crée deux colonnes nommées TEMP et VX. Pour les lignes 10 à 20 dont la valeur de la colonne V3 est un, la ligne correspondante dans les colonnes V1 et V2 est extraite dans les colonnes TEMP et VX, respectivement.

## Fonctions associées

Fonction	Description
IF	Commence une instruction conditionnelle if-then-else
SELECT	Renvoie les colonnes spécifiées d'une plage de données
SUBSAMPLE	Réduit la quantité de données en renvoyant toutes les n <sup>ème</sup> valeurs de ligne
SUBSTITUTE	Remplace les valeurs d'une colonne par une valeur spécifiée dans une table de conversion

## FACTORIAL

### Syntaxe

FACTORIAL(data)

### Paramètres

data

Valeurs entières dont la factorielle doit être calculée. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs, mais la valeur doit être supérieure ou égale à zéro. Pour la définition du format de data, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

### Description

FACTORIAL calcule la factorielle des valeurs de la plage de données spécifiée. Toutes les entrées doivent être des entiers supérieurs ou égaux à zéro. La factorielle d'un nombre entier inférieur ou égal à un est un. Pour les entiers  $x \geq 2$ , la factorielle  $x!$  =  $x(x-1)(x-2)...(x-(x-1))$ . FACTORIAL renvoie une nouvelle colonne par colonne en entrée, chacune contenant la factorielle des nombres de la colonne d'entrée correspondante.

**Remarque :** Toute valeur supérieure à 34 génère ??? (erreur de dépassement des valeurs à virgule flottante).

## Exemples

<code>TEMP = FACTORIAL(3)</code> Crée une colonne nommée TEMP contenant la valeur 6.
<code>TEMP = FACTORIAL(-2)</code> Génère une erreur 333, indiquant que l'argument doit être supérieur ou égal à 0.
<code>TEMP = FACTORIAL(V1)</code> Crée une colonne nommée TEMP, dans laquelle chaque valeur correspond à la factorielle du contenu de la colonne V1.
<code>TEMP = FACTORIAL(V1:V3)</code> Crée trois colonnes nommées TEMP, VX et VY. Les valeurs de la colonne TEMP correspondent aux factorielles du contenu de la colonne V1, les valeurs de la colonne VX correspondent aux factorielles du contenu de la colonne V2 et les valeurs de la colonne VY correspondent aux factorielles du contenu de la colonne V3.
<code>TEMP = FACTORIAL(V1[10:20])</code> Crée une colonne nommée TEMP, dans laquelle les 11 premières cibles contiennent les factorielles des valeurs des lignes 10 à 20 de la colonne V1. Les autres cibles de TEMP sont vides.
<code>TEMP = FACTORIAL(V1[50:99]:V2)</code> Crée deux colonnes nommées TEMP et VX, contenant des valeurs dans les lignes 1 à 50 (les autres cibles sont vides). Les valeurs de la colonne TEMP sont les factorielles des lignes de la colonne V1 et les valeurs de la colonne VX sont les factorielles des valeurs de la colonne V2.

---

## FLOOR

### Syntaxe

`FLOOR(data)`

### Paramètres

data

Valeurs numériques dont la partie entière doit être calculée. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de data, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

### Description

FLOOR calcule la partie entière des valeurs de la plage de données spécifiée. La partie entière d'un nombre correspond au plus grand entier inférieur à ce nombre. FLOOR renvoie une nouvelle colonne par colonne d'entrée, chacune contenant la partie entière des nombres de la colonne d'entrée correspondante.

**Remarque :** Cela est équivalent à la fonction macro INT.

## Exemples

<code>TEMP = FLOOR(4.3)</code>
Crée une colonne nommée TEMP contenant la valeur 4.
<code>TEMP = FLOOR(2.9)</code>
Crée une colonne nommée TEMP contenant la valeur -3.
<code>TEMP = FLOOR(V1)</code>
Crée une colonne nommée TEMP, dans laquelle chaque valeur correspond à la partie entière du contenu de la colonne V1.
<code>TEMP = FLOOR(V1:V3)</code>
Crée trois colonnes nommées TEMP, VX et VY. Les valeurs de la colonne TEMP correspondent aux parties entières du contenu de la colonne V1, les valeurs de la colonne VX correspondent aux parties entières du contenu de la colonne V2 et les valeurs de la colonne VY correspondent aux parties entières du contenu de la colonne V3.
<code>TEMP = FLOOR(V1[10:20])</code>
Crée une colonne nommée TEMP, dans laquelle les 11 premières cibles contiennent les parties entières des valeurs des lignes 10 à 20 de la colonne V1. Les autres cibles de TEMP sont vides.
<code>TEMP = FLOOR(V1[50:99]:V2)</code>
Crée deux colonnes nommées TEMP et VX, contenant des valeurs dans les lignes 1 à 50 (les autres cibles sont vides). Les valeurs de la colonne TEMP sont les parties entières des lignes de la colonne V1 et les valeurs de la colonne VX sont les parties entières des valeurs de la colonne V2.

## Fonctions associées

Fonction	Description
CEILING	Calcule le plafond de chaque valeur de la plage de données spécifiée
FRACTION	Renvoie la partie décimale de chaque valeur de la plage de données spécifiée
TRUNCATE	Renvoie la partie non décimale de chaque valeur de la plage de données spécifiée

---

## FORMAT

### Syntaxe

Format possède deux formes ; l'une pour les données de type numérique et l'autre pour celles de type texte/caractère.

Pour les données de type numérique :

```
FORMAT(colName, width [, precision [, format_type [, alignment [, padding]]]])
```

Pour les données de type texte/caractère :

```
FORMAT(colName, width [, alignment])
```

## Paramètres

`colName`

La macro examine `colName`, détermine son type de données, puis impose les règles appropriées pour les paramètres suivants en conséquence.

`width`

La largeur doit être assez importante pour contenir le résultat complet ; sinon le résultat est tronqué. Les valeurs admises sont comprises entre 1 et 29 si `colName` est numérique et entre 1 et 255 dans le cas contraire.

`precision`

La précision est le nombre de chiffres après le séparateur décimal. Les valeurs admises sont comprises entre 0 et 15. Si la valeur est égale à zéro, le résultat est un entier. La précision par défaut est de 2.

`format_type`

Les mots clés valides de `format_type` sont les suivants :

PERIOD	Le point (.) est utilisé comme symbole décimal. Aucun symbole de groupement des chiffres n'est utilisé. Il s'agit de la valeur par défaut.
COMMA	La virgule (,) est utilisée comme symbole décimal. Aucun symbole de groupement des chiffres n'est utilisé.
PERIOD_COMMA	Le point est utilisé comme symbole décimal et la virgule comme symbole de groupement des chiffres.
COMMA_PERIOD	Virgule comme symbole décimal et point comme symbole de groupement des chiffres.

`alignment`

Les mots clés valides pour l'alignement sont `LEFT` et `RIGHT`. La valeur par défaut est `RIGHT` pour les données de type numérique et `LEFT` pour les données de type texte/caractère.

`padding`

Les mots clés valides pour le remplissage sont `SPACE` et `ZERO`. La valeur par défaut est `SPACE`. `ZERO` est ignoré (`SPACE` est utilisé à la place) si l'alignement est `LEFT`.

Notez que les chaînes numériques contenues dans des données de type texte/caractère sont traitées comme texte/caractère. Notez également que la forme numérique accepte plusieurs mots clés facultatifs possédant chacun une valeur par défaut. Toutefois, pour remplacer la valeur par défaut du deuxième mot clé facultatif ou des suivants, vous DEVEZ coder les valeurs par défaut des mots clés facultatifs précédents (dans les faits, ils deviennent obligatoires). Par exemple : pour remplacer l'alignement par un alignement à gauche (`LEFT`), vous devez

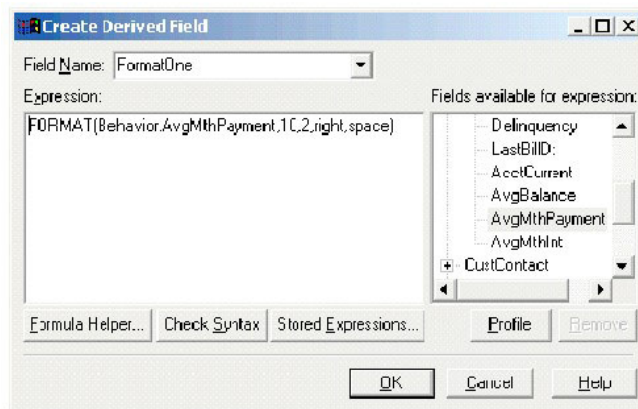
coder : `FORMAT(myNumCo1, 10, 2, PERIOD, LEFT)`.

## Description

FORMAT convertit des données numériques en chaîne avec diverses options de formatage pour contrôler et définir la chaîne en sortie. Cela est particulièrement utile pour créer des fichiers d'instantané avec des formats spécifiques à des fins de publipostage.

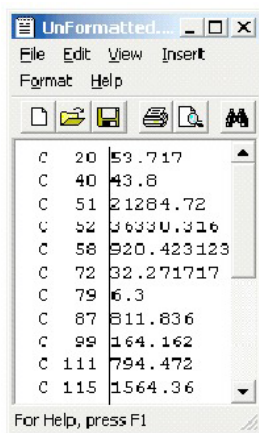
## Exemples

L'exemple ci-après définit une zone dérivée à l'aide de FORMAT.



Les exemples ci-après montrent la même zone, AvgMthPayment, dans trois formats.

Non formaté :



Formaté à l'aide de `FORMAT(Behavior.AvgMthPayment,10,2,right.space)` :

Code	Valeur
C 20	53.72
C 40	43.80
C 51	21284.72
C 52	36330.32
C 58	920.42
C 72	32.27
C 79	6.30
C 87	811.84
C 99	164.16
C 111	794.47
C 115	1564.36

Formaté à l'aide de `FORMAT(Behavior.AvgMthPayment,10,4)` :

Code	Valeur
C 20	53.7170
C 40	43.8000
C 51	21284.7200
C 52	36330.3160
C 58	920.4231
C 72	32.2717
C 79	6.3000
C 87	811.8360
C 99	164.1620
C 111	794.4720
C 115	1564.3600

## FRACTION

### Syntaxe

`FRACTION(data)`

### Paramètres

`data`

Valeurs numériques dont la fraction doit être calculée. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de data, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

### Description

`FRACTION` calcule la partie décimale des valeurs de la plage de données spécifiée. Il renvoie une nouvelle colonne par colonne d'entrée, chacune contenant la partie décimale des nombres de la colonne d'entrée correspondante.

**Remarque :** Les fonctions macro FRACTION et TRUNCATE sont complémentaires car leur somme correspond aux valeurs d'origine.

## Exemples

TEMP = FRACTION(4.3)
Crée une colonne nommée TEMP contenant la valeur 0.3.
TEMP = FRACTION(2.9)
Crée une colonne nommée TEMP contenant la valeur -0.9.
TEMP = FRACTION(V1)
Crée une colonne nommée TEMP, dans laquelle chaque valeur correspond à la partie décimale du contenu de la colonne V1.
TEMP = FRACTION(V1:V3)
Crée trois colonnes nommées TEMP, VX et VY. Les valeurs de la colonne TEMP correspondent aux parties décimales du contenu de la colonne V1, les valeurs de la colonne VX correspondent aux parties décimales du contenu de la colonne V2 et les valeurs de la colonne VY correspondent aux parties décimales du contenu de la colonne V3.
TEMP = FRACTION(V1[10:20])
Crée une colonne nommée TEMP, dans laquelle les 11 premières cibles contiennent les parties décimales des valeurs des lignes 10 à 20 de la colonne V1. Les autres cibles de TEMP sont vides.
TEMP = FRACTION(V1[50:99]:V2)
Crée deux colonnes nommées TEMP et VX, contenant des valeurs dans les lignes 1 à 50 (les autres cibles sont vides). Les valeurs de la colonne TEMP correspondent aux parties décimales des lignes de la colonne V1 et les valeurs de la colonne VX correspondent aux parties décimales des valeurs de la colonne V2.

## Fonctions associées

Fonction	Description
CEILING	Calcule le plafond de chaque valeur de la plage de données spécifiée
FLOOR	Calcule la partie entière de chaque valeur de la plage de données spécifiée
TRUNCATE	Renvoie la partie non décimale de chaque valeur de la plage de données spécifiée

---

## GAUSS



### Syntaxe

GAUSS(data [, mean, std])

### Paramètres

data1



Plage de cibles dont la valeur de Gauss doit être calculée. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de data, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

mean

Moyenne de la valeur de Gauss. Si ce paramètre n'est pas fourni, la valeur par défaut est zéro. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Le nombre de colonnes de mean doit correspondre au nombre de colonnes de data, à moins que mean ne soit une constante ou une colonne unique. Pour la définition du format de mean, (identique à la définition de data) voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

std

Ecart type de la valeur de Gauss. Si ce paramètre n'est pas fourni, la valeur par défaut est un. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Le nombre de colonnes de std doit correspondre au nombre de colonnes de data, à moins que std ne soit une constante ou une colonne unique. Pour la définition du format de std, (identique à la définition de data) voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

### Syntaxe

GAUSS calcule la valeur de Gauss des nombres dans la plage de données spécifiée. Renvoie une nouvelle colonne par colonne d'entrée, chacune contenant la valeur de Gauss de l'entrée correspondante. GAUSS est calculé de la manière suivante :

$$\frac{1}{\sqrt{2\pi}} e^{-\frac{(x - \mu)^2}{2\sigma^2}}$$

Les paramètres de la moyenne et de l'écart type sont traités comme suit :

- Si la moyenne et l'écart type ne sont pas fournis, les valeurs zéro et un sont respectivement utilisées.
- Si mean et std sont des constantes, ces valeurs sont utilisées pour spécifier la valeur de Gauss de toutes les valeurs de data.
- Si mean et std sont des colonnes uniques, les valeurs de ligne correspondantes sont utilisées pour chaque ligne de data.
- Si mean et std correspondent à des plages de colonnes (leur nombre de colonnes doit être identique à data), chaque cible de data utilise sa paire individuelle de cibles correspondantes dans mean et std.

## Exemples

<code>TEMP = GAUSS(0)</code> ou <code>TEMP = GAUSS(0, 0, 1)</code> Crée une colonne nommée TEMP contenant la valeur 0.4.
<code>TEMP = GAUSS(V1)</code> Crée une colonne nommée TEMP, dans laquelle chaque valeur est la valeur de Gauss de la ligne correspondante de la colonne V1, si une moyenne de zéro est une valeur de Gauss unit-variant sont utilisées.
<code>TEMP = GAUSS(V1:V3)</code> Crée trois colonnes nommées TEMP, VX et VY. Les valeurs de la colonne TEMP correspondent aux valeurs de Gauss de la colonne V1, les valeurs de la colonne VX correspondent aux valeurs de Gauss de la colonne V2 et les valeurs de la colonne VY correspondent aux valeurs de Gauss de la colonne V3. La valeur de Gaussian possède une moyenne de zéro et une variance unité.
<code>TEMP = GAUSS(V1[1:50]:V3)</code> Crée trois colonnes nommées TEMP, VX et VY. Les valeurs de la colonne TEMP correspondent aux valeurs de Gauss des lignes 1 à 50 de la colonne V1, les valeurs de la colonne VX correspondent aux valeurs de Gauss des lignes 1 à 50 de la colonne V2 et les valeurs des lignes de la colonne VY correspondent aux valeurs de Gauss de la colonne V3. La valeur de Gaussian possède une moyenne de zéro et une variance unité.
<code>TEMP = GAUSS(V1, 0, 3.5)</code> Crée une colonne nommée TEMP, dans laquelle chaque valeur est la valeur de Gauss de la ligne correspondante de la colonne V1. La valeur de Gauss possède une moyenne de 0 et un écart type de 3.5.
<code>TEMP = GAUSS(V1, V2, V3)</code> Crée une colonne nommée TEMP, dans laquelle chaque valeur représente la valeur de Gauss de la colonne V1, en utilisant la valeur de la ligne correspondante de la colonne V2 comme moyenne et la valeur de la ligne correspondante de la colonne V3 comme écart type.
<code>TEMP = GAUSS(V1:V2, V3:V4, V5:V6)</code> Crée deux colonnes nommées TEMP et VX. La colonne TEMP contient les valeurs de Gauss des valeurs de V1 en utilisant les lignes correspondantes de la colonne V3 comme moyenne et les lignes correspondantes de la colonne V5 comme écart type. La colonne VX contient les valeurs de Gauss des valeurs de V2 en utilisant les lignes correspondantes de la colonne V4 comme moyenne et les lignes correspondantes de la colonne V6 comme écart type.

## Fonctions associées

<code>RANDOM_GAUSS</code>	Renvoie le nombre spécifié de valeurs aléatoires à partir d'une distribution de Gauss
---------------------------	---

---

## GAUSS\_AREA



### Syntaxe

`GAUSS_AREA(data [, mean, std])`

## Paramètres

data1

Plage de cibles dont la zone sous la valeur de Gauss doit être calculée. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de mean, (identique à la définition de data) voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

mean

Moyenne de la valeur de Gauss. Si ce paramètre n'est pas fourni, la valeur par défaut est zéro. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Le nombre de colonnes de mean doit correspondre au nombre de colonnes de data, à moins que mean ne soit une constante ou une colonne unique. Pour la définition du format de data, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

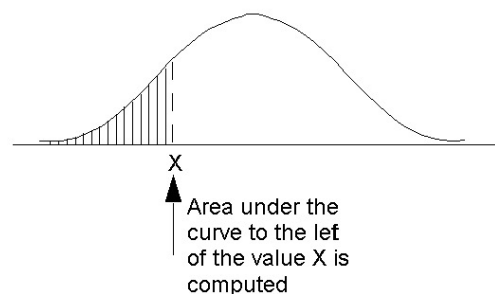
std

Ecart type de la valeur de Gauss. Si ce paramètre n'est pas fourni, la valeur par défaut est un. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Le nombre de colonnes de std doit correspondre au nombre de colonnes de data, à moins que std ne soit une constante ou une colonne unique. Pour la définition du format de std, (identique à data), voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

## Description

GAUSS\_AREA calcule la zone sous la valeur de Gauss (entre  $-\infty$  et la valeur de données  $X$  spécifiée) des nombres de la plage de données spécifiée (voir la figure ci-après). Il renvoie une nouvelle colonne par colonne d'entrée, chacune contenant la zone sous la valeur de Gauss de l'entrée correspondante.

Area Under a Gaussian Computed by GAUSS\_AREA



Les paramètres de la moyenne et de l'écart type sont traités comme suit :

- Si la moyenne et l'écart type ne sont pas fournis, les valeurs zéro et un sont respectivement utilisées.

- Si mean et std sont des constantes, ces valeurs sont utilisées pour spécifier la valeur de Gauss de toutes les valeurs de data.
- Si mean et std sont des colonnes uniques, les valeurs de ligne correspondantes sont utilisées pour chaque ligne de data.
- Si mean et std correspondent à des plages de colonnes (leur nombre de colonnes doit être identique à data), chaque cible de data utilise sa paire individuelle de cibles correspondantes dans mean et std.

**Remarque :** Dans le dernier cas ci-dessus, si mean et std sont des plages de colonnes, la longueur de chaque colonne détermine le nombre de lignes qui seront présentes dans la colonne en sortie correspondante. Si une colonne de mean ou std est une cible unique, cette valeur est utilisée pour toutes les valeurs de ligne de data. Si mean ou std contient plusieurs lignes, les lignes correspondantes sont calculées. Les lignes de data pour lesquelles il n'existe pas de valeurs correspondantes dans mean et std ne sont pas calculées.

## Exemples

<p>TEMP = GAUSS_AREA(0) ou TEMP = GAUSS_AREA(0, 0, 1)</p> <p>Crée une colonne nommée TEMP contenant la valeur 0.5.</p>
<p>TEMP = GAUSS_AREA(V1)</p> <p>Crée une colonne nommée TEMP, dans laquelle chaque valeur est la zone sous la valeur de Gauss de la ligne correspondante de la colonne V1, si une moyenne de zéro est une valeur de Gauss unit-variant sont utilisées.</p>
<p>TEMP = GAUSS_AREA(V1:V3)</p> <p>Crée trois colonnes nommées TEMP, VX et VY. Les valeurs de la colonne TEMP correspondent aux surfaces sous les valeurs de Gauss de la colonne V1, les valeurs de la colonne VX correspondent aux zones sous les valeurs de Gauss de la colonne V2 et les valeurs de la colonne VY correspondent aux zones sous les valeurs de Gauss de la colonne V3. La valeur de Gaussian possède une moyenne de zéro et une variance unité.</p>
<p>TEMP = GAUSS_AREA(V1[1:50]:V3)</p> <p>Crée trois colonnes nommées TEMP, VX et VY. Les valeurs de la colonne TEMP correspondent aux zones sous les valeurs de Gauss des lignes 1 à 50 de la colonne V1, les valeurs de la colonne VX correspondent aux zones sous les valeurs de Gauss des lignes 1 à 50 de la colonne V2 et les valeurs des lignes de la colonne VY correspondent aux zones sous les valeurs de Gauss de la colonne V3. La valeur de Gaussian possède une moyenne de zéro et une variance unité.</p>
<p>TEMP = GAUSS_AREA(V1, 0, 3.5)</p> <p>Crée une colonne nommée TEMP, dans laquelle chaque valeur est la zone sous la valeur de Gauss de la ligne correspondante de la colonne V1. La valeur de Gauss possède une moyenne de 0 et un écart type de 3.5.</p>
<p>TEMP = GAUSS_AREA(V1, V2, V3)</p> <p>Crée une colonne nommée TEMP, dans laquelle chaque valeur représente la zone sous la valeur de Gauss de la colonne V1, en utilisant la valeur de la ligne correspondante de la colonne V2 comme moyenne et la valeur de la ligne correspondante de la colonne V3 comme écart type.</p>

```
TEMP = GAUSS_AREA(V1:V2, V3:V4, V5:V6)
```

Crée deux colonnes nommées TEMP et VX. La colonne TEMP contient les zones sous les valeurs de Gauss des valeurs de V1 en utilisant les lignes correspondantes de la colonne V3 comme moyenne et les lignes correspondantes de la colonne V5 comme écart type. La colonne VX contient les zones sous les valeurs de Gauss des valeurs de V2 en utilisant les lignes correspondantes de la colonne V4 comme moyenne et les lignes correspondantes de la colonne V6 comme écart type.

---

## GE

### Syntaxe

```
data1 GE data2 data1 >= data2
```

### Paramètres

data1

Plage de cibles numériques à comparer. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de data, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

data2

Nombres auxquels toutes les valeurs de la colonne spécifiée doivent être comparées. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Le nombre de colonnes dans data2 doit correspondre au nombre de colonnes dans data1, à moins que data2 ne soit une constante. Pour la définition du format de data, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

### Description

GE compare les deux plages de données spécifiées, puis renvoie un un si les valeurs du premier jeu de données sont supérieures ou égales à celles du deuxième jeu de données ou un zéro dans le cas contraire. Il renvoie une nouvelle colonne pour chaque colonne d'entrée, chacune contenant la colonne correspondante de data1 comparée à la colonne correspondante de data2 (à savoir, la première colonne de data1 est comparée à la première colonne de data, la deuxième colonne à la deuxième colonne et ainsi de suite).

Si data2 est une constante, chaque valeur de data est comparée à cette valeur. Si data2 est une colonne, les calculs sont effectués ligne par ligne. Les valeurs de data1 sont comparées à la valeur de la première ligne de data2, la deuxième ligne à la deuxième ligne et ainsi de suite. Ce calcul ligne par ligne génère un résultat pour chaque ligne jusqu'à la dernière valeur de la colonne la plus courte.

**Remarque :** L'opérateur GE peut être abrégé en signe supérieur suivi d'un signe égal (>=).

Cette macro est disponible dans IBM Interact.

## Exemples

<code>TEMP = 9 GE 4</code> ou <code>TEMP = 9 &gt;= 4</code>
Crée une colonne nommée TEMP contenant la valeur un (car neuf est supérieur à quatre).
<code>TEMP = V1 &gt;= 8</code>
Crée une colonne nommée TEMP, dans laquelle chaque valeur est égale à un si la ligne correspondante de la colonne V1 est supérieure ou égale à huit ; sinon, la valeur est égale à zéro.
<code>TEMP = V1:V3 &gt;= 2</code>
Crée trois colonnes nommées TEMP, VX et VY. Les valeurs de la colonne TEMP correspondent au contenu de la colonne V1 comparé à la valeur deux, les valeurs de la colonne VX correspondent au contenu de la colonne V2 comparé à la valeur deux, et les valeurs de la colonne VY correspondent au contenu de la colonne V3 comparé à la valeur deux.
<code>TEMP = V1 &gt;= V1</code>
Crée une colonne nommée TEMP ne contenant que des uns (car tout nombre est égal à lui-même).
<code>TEMP = V1 &gt;= V2</code>
Crée une colonne nommée TEMP, dans laquelle chaque valeur est la valeur de ligne de la colonne V1 comparée à la valeur de ligne correspondante de la colonne V2.
<code>TEMP = V1:V3 &gt;= V4:V6</code>
Crée trois colonnes nommées TEMP, VX et VY. La colonne TEMP contient les valeurs de V1 comparées aux valeurs de ligne correspondantes de la colonne V4. La colonne VX compare les colonnes V2 et V5. La colonne VY compare les colonnes V3 et V6.
<code>TEMP = V1[10:20] &gt;= V2</code> ou <code>TEMP = V1[10:20] &gt;= V2[1:11]</code>
Crée une colonne nommée TEMP, dans laquelle les 11 premières cibles contiennent les résultats de la comparaison des valeurs des lignes 10 à 20 de la colonne V1 aux valeurs des lignes 1 à 11 de la colonne V2. Les autres cibles de TEMP sont vides.

## Fonctions associées

NE	Renvoie TRUE si une plage de données n'est pas égale à une autre
----	--

---

## GRID



### Syntaxe

`GRID(col1 [, col2]...)`

### Paramètres

`col1`

Première colonne à partir de laquelle la grille doit être générée. Il peut s'agir d'une constante, d'une colonne, d'une plage de cibles à une colonne ou de toute expression convertie comme l'une de ces valeurs.

col2

Une ou plusieurs colonnes supplémentaires à utiliser pour générer la grille. Il peut s'agir d'une constante, d'une colonne, d'une plage de cibles à une colonne ou de toute expression convertie comme l'une de ces valeurs.

## Description

GRID génère une grille de toutes les combinaisons de valeurs possibles, à l'aide des cibles de chacune des colonnes en entrée. Il renvoie une nouvelle colonne pour chaque colonne d'entrée. Si l'une des entrées est une constante, chaque valeur des autres colonnes en entrée est associée à cette valeur constante.

Si la notation V1[1] est utilisée pour la première cible de la colonne V1, V1[2] pour la deuxième cible et ainsi de suite, l'expression GRID(V1, V2) génère :

```
V1[1] V2[1] V1[1] V2[2] V1[1] V2[3] : : V1[2] V2[1] V1[2] V2[2] V1[2]
V2[3] : : : :
```

Toutes les cibles de la colonne V1 sont associées à chaque cible de la colonne V2. GRID se comporte de la même manière pour plus de deux colonnes en entrée. Le nombre de lignes générées est égal au produit de la longueur des colonnes en entrée.

**Remarque :** La longueur concaténée de tous les arguments en entrée ne peut pas dépassé  $(2^{16} - 1)/16$  .

## Exemples

TEMP = GRID(1,2) Crée deux colonnes nommées TEMP et VX, contenant respectivement les valeurs 1 et 2.
TEMP = GRID(COLUMN(1, 2), COLUMN(3, 4)) Crée deux colonnes nommées TEMP et VX. Les lignes sur ces deux colonnes sont : 1 3 1 4 2 3 2 4
TEMP = GRID(V1) Crée une colonne nommée TEMP contenant une copie des valeurs de la colonne V1.
TEMP = GRID(V1, 3) Crée deux colonnes nommées TEMP et VX. La colonne TEMP est une copie de la colonne V1 et la colonne VX contient la valeur 3 pour chaque ligne de la colonne V1.
TEMP = GRID(V1, V2) Crée deux colonnes nommées TEMP et VX, où chaque ligne représente l'une des combinaisons de cibles possibles (voir description).
TEMP = GRID(V1, V3, V7) Crée trois colonnes nommées TEMP, VX et VY, où chaque ligne représente l'une des combinaisons de cibles possibles (voir description).

## Fonctions associées

Fonction	Description
SLIDE_WINDOW	Crée un modèle à partir d'une fenêtre spécifiée et le fait glisser pour créer le modèle suivant

---

## GROUPBY

### Syntaxe

GROUPBY(group\_field, keyword, rolled\_field [,output\_field])

### Paramètres

- group\_field  
Indique la variable sur laquelle les enregistrements sont regroupés (toutes les valeurs identiques de la variable spécifiée sont regroupées ensemble).
- keyword  
Indique la fonction de cumul récapitulatif à effectuer sur rolled-field.
- rolled\_field  
Indique la variable à récapituler ou cumuler.
- output\_field  
Identifie une variable de remplacement à renvoyer pour une ligne d'un groupe et ne pouvant être utilisée qu'avec les mots clés MinOf, MaxOf et MedianOf.

### Description

GROUPBY récapitule plusieurs lignes de données dans un groupe. La sortie de cette fonction est une colonne unique. La sortie est le résultat de l'opération spécifiée par keyword sur la zone rolled\_field du groupe homogène spécifié par group\_field. S'il existe plusieurs réponses correspondant à une condition spécifiée, la première détectée est renvoyée.

Si le paramètre facultatif output\_field n'est pas fourni, la sortie est le résultat de l'opération sur rolled\_field. Si output\_field est fourni, le résultat est la zone output\_field de la ligne dans le groupe.

Si plusieurs lignes d'un groupe répondent à la condition spécifiée (par exemple, il existe des liens pour la valeur max), la zone output-field associé à la première ligne remplissant la condition est renvoyé.

**Remarque :** Pour effectuer un groupement sur plusieurs colonnes, vous pouvez placer entre accolades "{" }" une liste de noms de zone, séparés par des virgules, et utiliser cette liste comme premier paramètre de l'appel de macro GROUPBY.

Les mots clés pris en charge sont les suivants (casse indifférente) :

Mot clé		Description
	Chaîne ?	
	Oui/Non	



CountOf	Oui	Renvoie le nombre d'enregistrements de chaque groupe (rolled_field peut être de type numérique ou chaîne ; la valeur renvoyée est la même quelle que soit la valeur de rolled_field).
MinOf	Oui	Renvoie la valeur minimale de rolled_field dans chaque groupe (rolled_field peut être de type numérique ou chaîne ; si rolled_field est une chaîne, la valeur la plus proche du début de l'alphabet (pour un tri alphabétique) est renvoyée).
MaxOf	Oui	Renvoie la valeur maximale de rolled_field dans chaque groupe (rolled_field peut être de type numérique ou chaîne ; si rolled_field est une chaîne, la valeur la plus proche de la fin de l'alphabet (pour un tri alphabétique) est renvoyée).
DiffOf	Oui	Renvoie le nombre de valeurs distinctes de rolled_field dans chaque groupe (rolled_field peut être de type numérique ou chaîne).
AvgOf	Non	Renvoie la valeur moyenne de rolled_field dans chaque groupe (rolled_field doit être numérique).
ModeOf	Oui	Renvoie la valeur modale (à savoir, la valeur la plus fréquente) de rolled_field dans chaque groupe (rolled_field peut être de type numérique ou chaîne).
MedianOf	Oui	Renvoie la valeur médiane (à savoir, la valeur du milieu lors d'un tri en fonction de rolled_field) de rolled_field dans chaque groupe (rolled_field peut être de type numérique ou chaîne ; si rolled_field est une chaîne, les valeurs sont triées par ordre alphabétique).

OrderOf	Oui	Renvoie l'ordre de rolled_field dans chaque groupe (rolled_field doit être numérique). Si plusieurs enregistrements ont la même valeur, ils reçoivent tous la même valeur.
SumOf	Non	Renvoie la somme de rolled_field dans chaque groupe (rolled_field doit être numérique).
StdevOf	Non	Renvoie l'écart type de rolled_field dans chaque groupe (rolled_field doit être numérique).
IndexOf	Oui	Renvoie l'index de base 1 (trié par rolled_field) de chaque enregistrement (rolled_field peut être de type numérique ou chaîne). L'ordre de tri est croissant.  Remarque : pour les zones numériques, l'ordre de tri de RankOf et IndexOf peut être décroissant, si l'on place un signe moins (-) devant la zone de tri.
RankOf	Oui	Renvoie la catégorie de base 1 (triée par rolled_field) dans laquelle se trouve chaque enregistrement (rolled_field peut être de type numérique ou chaîne). L'ordre de tri est croissant.  Remarque : pour les zones numériques, l'ordre de tri de RankOf et IndexOf peut être décroissant, si l'on place un signe moins (-) devant la zone de tri.

## Exemples

GROUPBY (Household_ID, SumOf, Account_Balance)
Calcule la somme de tous les soldes de compte par ménage.
GROUPBY (Cust_ID, MinOf, Date(Account_Open_Date), Acc_Num)
Renvoie le numéro de compte du premier compte ouvert par le client.

---

## GROUPBY\_WHERE

### Syntaxe

```
GROUPBY_WHERE(group_field, keyword, rolled_field, where_value  
[,output_field])
```

### Paramètres

- `group_field`  
Indique la variable sur laquelle les enregistrements sont regroupés (toutes les valeurs identiques de la variable spécifiée sont regroupées ensemble).
- `keyword`  
Indique la fonction de cumul récapitulatif à effectuer.
- `rolled_field`  
Indique la variable à récapituler ou cumuler.
- `where_value`  
Expression convertie en un ou zéro qui indique les lignes à inclure dans l'opération de cumul.
- `output_field`  
Identifie une variable de remplacement à renvoyer pour une ligne d'un groupe et ne pouvant être utilisée qu'avec les mots clés `MinOf`, `MaxOf` et `MedianOf`

### Description

GROUPBY\_WHERE récapitule des lignes de données spécifiques dans un groupe. La sortie de cette fonction est une colonne unique. La sortie est le résultat de l'opération spécifiée par `keyword` sur la zone `rolled_field` du groupe homogène spécifié par `group_field` et filtré par `where_value`. Seules les lignes pour lesquelles `where_value` contient la valeur un sont incluses dans le calcul.

Si le paramètre facultatif `output_field` n'est pas fourni, la sortie est le résultat de l'opération sur `rolled_field`. Si `output_field` est fourni, le résultat est la zone `output_field` de la ligne dans le groupe.

**Remarque :** Pour plus d'informations sur les valeurs valides de `keyword`, voir «GROUPBY», à la page 114.

### Exemples

```
GROUPBY_WHERE (Household_ID, SumOf, Account_Balance, Account_Balance>0)
```

Calcule la somme de tous les comptes dont les soldes sont positifs pour chaque ménage.

```
GROUPBY_WHERE (Cust_ID, AvgOf, Purchase_Amt,Date(Current_Date) -  
Date(Purchase_Date)<90)
```

Calcule le montant des achats moyen de chaque client pour les achats au cours des 90 derniers jours.

---

## GT

### Syntaxe

```
data1 GT data2 data1 > data2
```

## Paramètres

data1

Plage de cibles numériques à comparer. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de data, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

data2

Nombre auxquels toutes les valeurs de la colonne spécifiée doivent être comparées. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Le nombre de colonnes dans data2 doit correspondre au nombre de colonnes dans data1, à moins que data2 ne soit une constante. Pour la définition du format de data, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

## Description

GT compare les deux plages de données spécifiées, puis renvoie un un si les valeurs du premier jeu de données sont supérieures à celles du deuxième jeu de données ou un zéro dans le cas contraire. Il renvoie une nouvelle colonne pour chaque colonne d'entrée, chacune contenant la colonne correspondante de data1 comparée à la colonne correspondante de data2 (à savoir, la première colonne de data1 est comparée à la première colonne de data, la deuxième colonne à la deuxième colonne et ainsi de suite).

Si data2 est une constante, chaque valeur de data est comparée à cette valeur. Si data2 est une colonne, les calculs sont effectués ligne par ligne. Les valeurs de data1 sont comparées à la valeur de la première ligne de data2, la deuxième ligne à la deuxième ligne et ainsi de suite. Ce calcul ligne par ligne génère un résultat pour chaque ligne jusqu'à la dernière valeur de la colonne la plus courte.

**Remarque :** L'opérateur GT peut être abrégé en signe supérieur (>).

Cette macro est disponible dans IBM Interact.

## Exemples

```
TEMP = 3 GT 4 ou TEMP = 3 > 4
```

Crée une colonne nommée TEMP contenant la valeur zéro (car trois n'est pas supérieur à quatre).

```
TEMP = V1 > 8
```

Crée une colonne nommée TEMP, dans laquelle chaque valeur est égale à un si la ligne correspondante de la colonne V1 est supérieure à huit ; sinon, la valeur est égale à zéro.

```
TEMP = V1:V3 > 2
```

Crée trois colonnes nommées TEMP, VX et VY. Les valeurs de la colonne TEMP correspondent au contenu de la colonne V1 comparé à la valeur deux, les valeurs de la colonne VX correspondent au contenu de la colonne V2 comparé à la valeur deux, et les valeurs de la colonne VY correspondent au contenu de la colonne V3 comparé à la valeur deux.

TEMP = V1 > V1
Crée une colonne nommée TEMP ne contenant que des zéros (car aucun nombre n'est supérieur à lui-même).
TEMP = V1 > V2
Crée une colonne nommée TEMP, dans laquelle chaque valeur est la valeur de ligne de la colonne V1 comparée à la valeur de ligne correspondante de la colonne V2.
TEMP = V1:V3 > V4:V6
Crée trois colonnes nommées TEMP, VX et VY. La colonne TEMP contient les valeurs de V1 comparées aux valeurs de ligne correspondantes de la colonne V4. La colonne VX compare les colonnes V2 et V5. La colonne VY compare les colonnes V3 et V6.
TEMP = V1[10:20] > V2 ou TEMP = V1[10:20] > V2[1:11]
Crée une colonne nommée TEMP, dans laquelle les 11 premières cibles contiennent les résultats de la comparaison des valeurs des lignes 10 à 20 de la colonne V1 aux valeurs des lignes 1 à 11 de la colonne V2. Les autres cibles de TEMP sont vides.

## Fonctions associées

Fonction	Description
EQ	Renvoie TRUE si une plage de données est égale à une autre
GE	Renvoie TRUE si une plage de données est supérieure ou égale à une autre
LE	Renvoie TRUE si une plage de données est inférieure ou égale à une autre
LT	Renvoie TRUE si une plage de données est inférieure à une autre
NE	Renvoie TRUE si une plage de données n'est pas égale à une autre

---

## HISTOGRAM



### Syntaxe

HISTOGRAM(data, bin\_col)

### Paramètres

data

Plage de cibles dont l'histogramme doit être calculé. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Toutes les colonnes de data doit être de même type (à savoir, numérique ou chaîne de texte). Pour la définition du format de data, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

bin\_col

Valeurs des limites de l'ensemble. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles dans une colonne ou d'une expression convertie comme l'une de ces valeurs. Le type de données de `bin_col` doit être le même que celui de `data`. Pour la définition du format de `data`, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

## Description

HISTOGRAM calcule l'histogramme (c'est-à-dire fréquence d'occurrence des valeurs des divers ensembles) des valeurs de la plage de données spécifiée. Il renvoie une colonne unique avec le nombre de valeurs dans `data` comprises dans la plage de l'ensemble correspondant spécifiée par `bin_col`.

Pour les valeurs numériques, deux lignes adjacentes de `bin_col` constituent un "ensemble". Toute valeur de `data` comprise dans un ensemble est accumulée pour cet ensemble. La colonne en sortie contient le nombre final de valeurs de chaque ensemble. La valeur de la première limite est *incluse* dans l'ensemble ; la valeur de la deuxième limite est *exclue*. Par exemple, la paire de valeurs limites 1 et 2 contient le nombre de valeurs de `data` supérieures ou égales à 1 et inférieures à 2. La longueur de la colonne en sortie est un moins la longueur de `bin_col`.

Pour les chaînes de texte, seules les correspondances exactes de la chaîne de texte de `bin_col` sont comptées dans cet ensemble. La longueur de la colonne en sortie correspond à la longueur de `bin_col`. Pour les données numériques, si `bin_col` est scalaire (il contient une seule valeur de cible), le nombre d'éléments de `data` est compté.

**Remarque :** La fonction macro HISTOGRAM place les points de données dans les ensembles de manière différente de l'histogramme d'**IBM PredictiveInsight**. L'histogramme exclut la valeur minimale (excepté pour l'ensemble le plus à gauche) et inclut la valeur maximale de chaque limite d'ensemble.

## Exemples

<pre>TEMP = HISTOGRAM(1..10, COLUMN(1, 3, 10))</pre> <p>Crée une colonne nommée TEMP contenant les valeurs 2 et 7.</p>
<pre>TEMP = HISTOGRAM("a", "b", "a"), COLUMN("a", "b", "c"))</pre> <p>Crée une colonne nommée TEMP contenant les valeurs 2, 1 et 0.</p>
<pre>TEMP = HISTOGRAM(V1, COLUMN(1, 25, 50, 75, 101))</pre> <p>Crée une colonne nommée TEMP contenant quatre valeurs. La première valeur correspond au nombre de valeurs de la colonne V1 supérieures ou égales à 1 et inférieures à 25. La deuxième valeur correspond au nombre de valeurs de la colonne V1 supérieures ou égales à 25 et inférieures à 50. Les troisième et quatrième valeurs contiennent les nombres de valeurs dans les troisième et quatrième quartiles, respectivement.</p>
<pre>TEMP = HISTOGRAM(V1:V3, V4)</pre> <p>Crée une colonne nommée TEMP, dans laquelle chaque valeur correspond au nombre de valeurs des colonnes V1 à V3 comprises dans les limites de l'ensemble spécifiées par la colonne V4.</p>

```
TEMP = HISTOGRAM(V1[50:100]:V5, V6[1:10])
```

Crée une colonne nommée TEMP contenant 10 valeurs. Chaque valeur représente le nombre de valeurs dans les lignes 50 à 100 des colonnes V1 à V5 comprises dans les limites de l'ensemble spécifiées par les lignes 1 à 10 de la colonne V6.

## Fonctions associées

Fonction	Description
COUNT	Compte le nombre de cibles contenant des valeurs dans la plage de données spécifiée

---

## IF

### Syntaxe

```
IF(predicate_col, then_value) IF(predicate_col, then_value, else_value)
```

### Paramètres

predicate\_col

Colonne de valeurs booléennes ou expression convertie en une colonne de valeurs booléennes. Les valeurs booléennes sont interprétées comme des nulles ou non nulles. Cette colonne doit contenir au moins autant de lignes que la plage de données de laquelle les données sont extraites. Sinon, predicate\_col constitue une limite du nombre de lignes traitées par la fonction macro EXTRACT (voir "Description" ci-dessous).

then\_value

Valeurs à renvoyer si la ligne de predicate\_col correspondante contient une valeur non nulle. Il peut s'agir d'une valeur constante, d'une colonne ou d'une expression convertie comme l'une de ces valeurs. Pour la définition de format de then\_value (identique à data), voir «Paramètres des fonctions macro», à la page 8.

else\_value

Si ce paramètre facultatif est fourni, il est renvoyé si la ligne correspondante de predicate\_col contient un zéro. Il peut s'agir d'une valeur constante, d'une colonne ou d'une expression convertie comme l'une de ces valeurs. Si else\_value n'est pas fourni, un zéro est renvoyé chaque fois que predicate\_col a la valeur false. Pour la définition de format de else\_value (identique à data), voir «Paramètres des fonctions macro», à la page 8.

### Description

IF convertit l'expression de predicate\_col et renvoie then\_value si l'expression est true ou else\_value si l'expression est false. Il renvoie le même nombre de colonnes dans then\_value et else\_value. Les nouvelles colonnes contiendront les valeurs de ligne correspondantes de then\_value si la valeur de predicate\_col est non nulle. Si else\_value est fourni, il est renvoyé lorsque la valeur de predicate\_col est zéro. Si else\_value n'est pas fourni, zéro est renvoyé.

IF fonctionnant ligne par ligne, il génère un résultat pour chaque ligne jusqu'à la dernière valeur de la colonne la plus courte (c'est-à-dire la colonne la plus courte entre predicate\_col, then\_value et else\_value).

**Remarque :** En général, une colonne de prédicat est créée à l'aide de l'une des fonctions macro de comparaison (par exemple, ==, >, <, ISEVEN, ISODD, etc.).

Cette macro est disponible dans IBM Interact.

## Exemples

TEMP = IF(1, V1)	Crée une colonne nommée TEMP contenant une copie de la colonne V1.
TEMP = IF(V1, 1, 0)	Crée une colonne nommée TEMP, dans laquelle chaque valeur est égale à un si la valeur correspondante de la colonne V1 est non nulle ; sinon, la valeur est égale à zéro.
TEMP = IF(V3, V1, V2)	Crée une colonne nommée TEMP, dans laquelle chaque valeur est copiée à partir de la colonne V1 si la valeur correspondante de la colonne V3 est non nulle ; sinon; la valeur est copiée à partir de la colonne V2.
TEMP = IF(ABS(V1-AVG(V1)) < STDV(V1), V1)	Crée une colonne nommée TEMP contenant chaque valeur de la colonne V1 qui est inférieure d'un écart type de la moyenne.
TEMP = IF(V3[20:30], V1[30:40], V2)	Crée une colonne nommée TEMP contenant les valeurs des lignes 10 à 20. Chaque valeur est copiée à partir de la colonne V1 (cibles 10 à 20) si la valeur correspondante de la colonne V3 (cibles 30 à 40) est non nulle ; sinon; la valeur est copiée à partir de la colonne V2 (cibles 1 à 11).

## Fonctions associées

Fonction	Description
EXTRACT	Extrait les lignes contenant les valeurs d'une colonne de prédicat
SELECT	Renvoie les colonnes spécifiées d'une plage de données

## IN

### Syntaxe

valuet IN (value1 AND value2 . . . .) or valuet IN subquery

### Paramètres

La première forme autorise l'utilisation d'une liste de valeurs à la place d'une sous-requête.

La seconde forme utilise une sous-requête qui est convertie pour générer un résultat intermédiaire, sur lequel un traitement supplémentaire peut être effectué.



## Description

Le prédicat IN permet d'utiliser une liste de valeurs au lieu d'une sous-requête ou lance une sous-requête.

**Remarque :** IN est différent de ISMEMBER car IN est exécuté sur la base de données (si possible), tandis que ISMEMBER est calculé sur le serveur.

**Remarque :** Le prédicat IN possède une version négative : NOT IN. Son format est identique à celui de IN. NOT IN est vrai uniquement si la valeur fournie est introuvable dans les valeurs renvoyées par la sous-requête.

Cette macro est disponible dans IBM Interact.

**Important :** Si vous utilisez IN dans IBM Interact, vous ne pouvez utiliser que la syntaxe value IN (value1 AND value2 . . . ).

## Exemples

```
TEMP = IN(25, COLUMN(1...10))
```

Renvoie les colonnes spécifiées d'une plage de données

```
TEMP = IN("cat", COLUMN("cat", "dog", "bird"))
```

Crée une colonne nommée TEMP contenant la valeur un.

```
TEMP = IN(V1, V1)
```

Crée une colonne nommée TEMP ne contenant que des uns.

```
TEMP = IN(V1, V2)
```

Crée une colonne nommée TEMP, dans laquelle chaque valeur est égale à un si la ligne correspondante de la colonne V1 contient une valeur dans la colonne V2 et à zéro dans le cas contraire.

## Exemples

```
SELECT title, current_rental_price FROM movie_titles WHERE our_cost IN (14.95, 24.95, 29.95 ) ;
```

signifie que toutes les vidéos coûtant 14,95, 24,95 ou 29,95 \$ seront converties en TRUE pour le prédicat IN et généreront à leur tour une liste de titres, ainsi que le prix de leur location.

```
UPDATE movie_titles SET current_rental_price = (regular_rental_price * .9) WHERE title IN ( SELECT movie_title FROM movie_stars WHERE actor_last_name = 'Stewart' AND actor_first_name = 'James' ) ;
```

Une fois que la sous-requête a généré sa liste de titres de vidéo, cette liste est comparée à la table MOVIE\_TITLES et les prix de location appropriés sont décomptés de *Philadelphia Story*, *It's a Wonderful Life*, etc.



## Syntaxe

INIT(val1 [, val2]...) INIT(column)

## Paramètres

val1

Valeur numérique de la fonction récursive à l'heure ( $t-1$ ).

val2

Valeur numérique de la fonction récursive à l'heure ( $t-n$ ), où  $n$  représente le numéro du paramètre. Ce paramètre peut être répété plusieurs fois pour fournir les valeurs initiales pour un nombre indéfini d'intervalles de temps précédents.

column

Colonne de valeurs numériques. La première cible sera affecté au premier intervalle de temps ( $t-1$ ), la deuxième valeur de cible à ( $t-2$ ), etc.

## Description

INIT indique les valeurs initiales d'une définition de fonction récursive. La première valeur fournie est affectée à l'intervalle de temps ( $t-1$ ), la deuxième à ( $t-2$ ), etc. Si un intervalle de temps n'est pas initialisé à l'aide INIT, sa valeur est considérée comme étant égale à zéro. Soit par exemple l'instruction suivante :

```
V1 = INIT(1, 2, 3)
```

La valeur de l'intervalle de temps ( $t-4$ ) est zéro (comme les valeurs de tous les autres intervalles de temps antérieurs). Une instruction INIT est requise pour pouvoir définir une fonction récursive.

**Remarque :** Pour initialiser toutes les valeurs à zéro, vous pouvez simplement spécifier INIT() sans arguments.

La fonction macro INIT ne renvoie pas de valeurs. Si elle est utilisée seule dans une définition de fonction, elle renvoie une colonne vide.

## Exemples

```
TEMP = INIT() t = 1 to 10 TEMP = 1 + TEMP[t-1]
```

Crée une colonne nommée TEMP contenant les valeurs 1 à 10.

```
TEMP = INIT(1) t = 1 TO 100 TEMP = TEMP[t-1]+TEMP[t-1]
```

Crée une colonne nommée TEMP contenant les valeurs 2, 4, 8, 16, 32, etc. Les 100 premières cibles de TEMP contiennent des valeurs.

```
TEMP = INIT(1, 2, 3, 4, 5) t = 1 to 500 TEMP = TEMP[t-5]
```

Crée une colonne nommée TEMP contenant la séquence de valeurs 5, 4, 3, 2, 1 répétée 100 fois.

```
TEMP = INIT(1, 2, 3) t = 1 to 1000 TEMP = 2*TEMP[t-1] + 4*TEMP[t-2]^2 - TEMP[t-3]
```

Crée une colonne nommée TEMP contenant les valeurs de la fonction cursive.

$$\text{TEMP} = 2 * \text{TEMP}(t - 1) + 4 * \text{TEMP}(t - 2)^2 - \text{TEMP}(t - 2)$$

1000 valeurs de cible sont calculées.

## Fonctions associées

Fonction	Description
T0	Génère l'opérateur de plage

---

## INT

### Syntaxe

INT(data)

### Paramètres

data

Valeurs numériques à arrondir en valeurs entières. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de data, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

### Description

INT calcule l'entier le plus grand inférieur aux valeurs (également appelé partie entière) de la plage de données spécifiée. INT renvoie une nouvelle colonne par colonne d'entrée, chacune contenant la partie entière des nombres de la colonne d'entrée correspondante.

**Remarque :** Cela est équivalent à la fonction macro FLOOR.

### Exemples

```
TEMP = INT(4.7)
```

Crée une colonne nommée TEMP contenant la valeur 4.

```
TEMP = INT(-1.5)
```

Crée une colonne nommée TEMP contenant la valeur -2.

```
TEMP = INT(V1)
```

Crée une colonne nommée TEMP, où chaque valeur correspond à l'entier le plus grand inférieur ou égal au contenu de la colonne V1.

```
TEMP = V1 - INT(V1)
```

Crée une colonne nommée TEMP contenant la partie décimale de chaque valeur de la colonne V1.

<p>TEMP = INT(V1:V3)</p> <p>Crée trois colonnes nommées TEMP, VX et VY. Les valeurs de la colonne TEMP correspondent aux entiers les plus grands inférieurs ou égaux au contenu de la colonne V1, les valeurs de la colonne VX correspondent aux entiers les plus grands inférieurs ou égaux au contenu de la colonne V2 et les valeurs de la colonne VY correspondent aux entiers les plus grands inférieurs ou égaux au contenu de la colonne V3.</p>
<p>TEMP = INT(V1[10:20])</p> <p>Crée une colonne nommée TEMP, dans laquelle les 11 premières cibles contiennent les entiers les plus grands inférieurs ou égaux aux valeurs correspondantes des lignes 10 à 20 de la colonne V1. Les autres cibles de TEMP sont vides.</p>
<p>TEMP = INT(V1[1:5]:V2)</p> <p>Crée deux colonnes nommées TEMP et VX, contenant des valeurs dans les lignes 1 à 5 (les autres cibles sont vides). Les valeurs de la colonne TEMP correspondent aux entiers les plus grands inférieurs ou égaux aux valeurs de ligne correspondantes de la colonne V1 et les valeurs de la colonne VX correspondent aux entiers les plus grands inférieurs ou égaux aux valeurs de ligne correspondantes de la colonne V2.</p>

## Fonctions associées

Fonction	Description
ROUND	Calcule la valeur arrondie du contenu de la plage de données spécifiée
TRUNCATE	Renvoie la partie non décimale de chaque valeur de la plage de données spécifiée

---

## INTEGRAL



### Syntaxe

INTEGRAL(data [, multiplier])

### Paramètres

data

Valeurs numériques dont l'intégrale doit être calculée. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de data, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

multiplier

Valeur par laquelle chaque valeur de data doit être multipliée. Il peut s'agir d'une valeur constante ou d'une expression convertie en constante.

## Description

INTEGRAL calcule l'intégrale des valeurs d'une série chronologique. Chaque valeur correspond à la somme de toutes les valeurs précédentes dans le temps. Si une valeur est indiquée pour multiplier, chaque valeur est multipliée par la valeur spécifiée. INTEGRAL renvoie une nouvelle colonne par colonne d'entrée, chacune contenant l'intégrale des valeurs de la colonne d'entrée correspondante.

## Exemples

TEMP = INTEGRAL(5)
Crée une colonne nommée TEMP contenant la valeur 5.
TEMP = INTEGRAL(COLUMN(1,2,3))
Crée une colonne nommée TEMP contenant les valeurs 1, 3 et 6.
TEMP = INTEGRAL(COLUMN(1,2,3), 2)
Crée une colonne nommée TEMP contenant les valeurs 2, 6 et 12.
TEMP = INTEGRAL(V1)
Crée une colonne nommée TEMP, dans laquelle chaque valeur correspond à la somme de toutes les cibles précédentes de la colonne V1.
TEMP = INTEGRAL(V1, 10)
Crée une colonne nommée TEMP, dans laquelle chaque valeur correspond à la somme de toutes les cibles précédentes de la colonne V1 fois 10.
TEMP = INTEGRAL(V1:V3)
Crée trois colonnes nommées TEMP, VX et VY. Les valeurs de la colonne TEMP correspondent aux intégrales de la colonne V1, les valeurs de la colonne VX correspondent aux intégrales de la colonne V2 et les valeurs de la colonne VY correspondent aux intégrales de la colonne V3.
TEMP = INTEGRAL(V1:V3)
Crée une colonne nommée TEMP, dans laquelle les 11 premières cibles contiennent les intégrales des valeurs correspondantes des lignes 10 à 20 de la colonne V1. Les autres cibles de TEMP sont vides.
TEMP = INTEGRAL(V1:V3)
Crée deux colonnes nommées TEMP et VX, contenant des valeurs dans les lignes 1 à 5 (les autres cibles sont vides). Les valeurs de la colonne TEMP correspondent aux valeurs de ligne de la colonne V1 et les valeurs de la colonne VX correspondent aux valeurs de ligne de la colonne V2.

## Fonctions associées

Fonction	Description
DTEMP = INTEGRAL(V1:V3)	Calcule la dérivée des valeurs de la plage de données spécifiée
SUM ou TOTAL	Calcule la somme d'une plage de cibles

---

# INVERSE

## Syntaxe

INVERSE(data)

## Paramètres

data

Valeurs numériques dont l'inverse doit être calculé. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de data, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

## Description

INVERSE calcule la valeur négative des valeurs de la plage de données spécifiée. Il renvoie  $-x$  (les valeurs négatives sont renvoyées comme valeurs positives et les valeurs positives sont renvoyées comme valeurs négatives). INVERSE renvoie une nouvelle colonne par colonne d'entrée, chacune contenant l'inverse des valeurs de la colonne d'entrée correspondante.

**Remarque :** Pour inverser une valeur ou une colonne, précédez-la d'un signe moins (-). Par exemple,  $V2 = -V1$  est équivalent à  $V2 = \text{INVERSE}(V1)$ .

## Exemples

TEMP = INVERSE(3.2)
Crée une colonne nommée TEMP contenant la valeur -3.2.
TEMP = INVERSE(V1)
Crée une colonne nommée TEMP, dans laquelle chaque valeur correspond à la valeur négative des valeurs de la colonne V1.
TEMP = INVERSE(V1:V3)
Crée trois colonnes nommées TEMP, VX et VY. Les valeurs de la colonne TEMP correspondent aux valeurs négatives des valeurs de la colonne V1, les valeurs de la colonne VX correspondent aux valeurs négatives des valeurs de la colonne V2 et les valeurs de la colonne VY correspondent aux valeurs négatives des valeurs de la colonne V3.
TEMP = INVERSE(V1[10:20])
Crée une colonne nommée TEMP, dans laquelle les 11 premières cibles contiennent les valeurs négatives des valeurs des lignes 10 à 20 de la colonne V1. Les autres cibles de TEMP sont vides.
TEMP = INVERSE(V1[1:5]:V2)
Crée deux colonnes nommées TEMP et VX, contenant des valeurs dans les lignes 1 à 5 (les autres cibles sont vides). Les valeurs de la colonne TEMP correspondent aux valeurs négatives des valeurs des lignes correspondantes de la colonne V1 et les valeurs de la colonne VX correspondent aux valeurs négatives des valeurs des lignes correspondantes de la colonne V2.

## Fonctions associées

Fonction	Description
ABS	Calcule la valeur absolue du contenu de la plage de données spécifiée
NOT	Calcule le non logique (NON) du contenu de la plage de données spécifiée
SIGN	Détermine le signe (positif ou négatif) des valeurs dans la plage de données spécifiée

---

## EST

### Syntaxe

IS <keyword>

### Paramètres

keyword

Condition de recherche, généralement "NULL," "TRUE," "UNKNOWN" et "FALSE."

### Description

IS est utilisé dans les conditions de recherche complexes. Plus la recherche est complexe, plus la condition IS peut s'avérer utile. Les conditions de recherche booléenne de ce type permettent d'améliorer de simples conditions de recherche.

Cette macro est disponible dans IBM Interact.

IS renvoie des résultats différents dans IBM Interact et IBM Campaign. NULL renvoie 1 s'il existe au moins une valeur NULL pour un ID audience. UNKNOWN renvoie 1 pour un ID audience s'il ne possède pas de valeur.

### Exemples

```
SELECT customer FROM customer_table1 WHERE (last_name = "Smith" AND first_name = "John") IS TRUE ;
```

génère une liste de tous les clients s'appelant John Smith.

```
SELECT customer FROM customer_table1 WHERE (last_name = "X" AND first_name = "X") IS UNKNOWN ;
```

demande toute valeur non null.

```
SELECT cost FROM cost_table1 WHERE (current_cost = "200" IS FALSE ;
```

répertorie toutes les valeurs de la table des coûts qui ne sont pas égales à 200 \$.

---

## ISERROR

### Syntaxe

ISERROR(data)

## Paramètres

data

Valeurs à tester pour vérifier si l'une des lignes contient une erreur (à savoir, une cible ???). Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de data, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

## Description

ISERROR vérifie si l'une des cibles de chaque ligne de la plage de données spécifiée contient une erreur (à savoir, une cible ???). Il renvoie une nouvelle colonne, chaque ligne contenant un un si la ligne correspondante de data contient une erreur. Sinon, la ligne contient un zéro. Ce calcul ligne par ligne génère un résultat pour chaque ligne jusqu'à la dernière valeur de la colonne la plus longue.

**Remarque :** Cette fonction est utile pour détecter les erreurs d'une colonne, puis utiliser la fonction macro EXTRACT pour extraire les lignes de données correctes.

## Exemples

TEMP = ISERROR(-3)
Crée une colonne nommée TEMP contenant la valeur zéro.
TEMP = ISERROR(V1)
Crée une colonne nommée TEMP, dans laquelle chaque valeur est égale à un si la ligne correspondante de la colonne V1 contient ??? et à zéro dans le cas contraire.
TEMP = ISERROR(V1:V3)
Crée une colonne nommée TEMP, dans laquelle chaque valeur est égale à un si l'une des cibles des lignes correspondantes des colonnes V1 à V3 contient ??? et à zéro dans le cas contraire.
TEMP = ISERROR(V1[50:100]:V10)
Crée une colonne nommée TEMP, avec les valeurs des lignes 1 à 50. Chaque valeur est égale à un si l'une des cibles des lignes 50 à 100 des colonnes V1 à V10 contient ??? et à zéro dans le cas contraire.

## Fonctions associées

Fonction	Description
EXTRACT	Extrait les lignes contenant les valeurs d'une colonne de prédicat

---

## ISEVEN

### Syntaxe

ISEVEN(data)



## Paramètres

data

Valeurs numériques à tester pour vérifier si elles sont paires. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de data, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

## Description

ISEVEN teste chaque valeur du jeu de données spécifié pour vérifier si elle est paire. Il renvoie une nouvelle colonne pour chaque colonne d'entrée, chacune contenant un un pour toutes les valeurs paires (à savoir, le modulo deux de la valeur est zéro) ou un zéro pour toutes les valeurs non paires (impaires).

**Remarque :** Pour les valeurs autres que des entiers, la fonction macro INT est appliquée au préalable. Par exemple, ISEVEN(2.5) = 1 car 2 est paire.

## Exemples

TEMP = ISEVEN(-3)

Crée une colonne nommée TEMP contenant la valeur zéro.

TEMP = ISEVEN(MERGE(3, 2, 0))

Crée trois colonnes nommées TEMP, VX et VY. TEMP contient la valeur 0, VX contient la valeur 1 et VY contient la valeur 1.

TEMP = ISEVEN(V1)

Crée une colonne nommée TEMP, dans laquelle chaque valeur correspond au résultat du test de valeur paire du contenu de la colonne V1.

TEMP = ISEVEN(V1:V3)

Crée trois colonnes nommées TEMP, VX et VY. Les valeurs de la colonne TEMP correspondent aux résultats du test de valeur paire du contenu de la colonne V1, les valeurs de la colonne VX correspondent aux résultats du test de valeur paire du contenu de la colonne V2 et les valeurs de la colonne VY correspondent aux résultats du test de valeur paire du contenu de la colonne V3.

TEMP = ISEVEN(V1[10:20])

Crée une colonne nommée TEMP, dans laquelle les 11 premières cibles contiennent les résultats du test de valeur paire des valeurs des lignes 10 à 20 de la colonne V1. Les autres cibles de TEMP sont vides.

TEMP = ISEVEN(V1[1:5]:V2)

Crée deux colonnes nommées TEMP et VX, contenant des valeurs dans les lignes 1 à 5 (les autres cibles sont vides). Les valeurs de la colonne TEMP correspondent aux résultats du test de valeur paire des lignes correspondantes de la colonne V1 et les valeurs de la colonne VX correspondent aux résultats du test de valeur paire des lignes correspondantes de la colonne V2.

## Fonctions associées

Fonction	Description
ISODD	Teste si les valeurs en entrée sont impaires (à savoir, non divisible par deux)
ISMEMBER	Teste une plage en entrée par rapport à une "table" de valeurs et renvoie un si une valeur se trouve dans la table et zéro dans le cas contraire

---

## ISMEMBER



### Syntaxe

ISMEMBER(data, table)

### Paramètres

data

Valeurs à tester pour vérifier si elles sont membres d'une table. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Si data contient plusieurs colonnes, toutes les colonnes doivent être de même type (numérique ou chaîne de texte). Pour la définition du format de data, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

table

Valeurs de table par rapport auxquelles les valeurs doivent être comparées. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Si table contient plusieurs colonnes, toutes les colonnes doivent être du même type que data (numérique ou chaîne de texte). Le nombre de valeurs de données dans table ne peut pas dépasser 16 millions. Pour la définition du format de data, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

**Remarque :** ISMEMBER est différent de IN car ISMEMBER est calculé sur le serveur alors que IN est calculé dans la base de données chaque fois que possible.

### Description

ISMEMBER compare les valeurs de données de la plage de données spécifiée par rapport à une table de valeurs de données. Il renvoie une nouvelle colonne par colonne d'entrée dans data, chacune contenant la valeur un si la valeur en entrée correspondante est membre de table et la valeur zéro dans le cas contraire.

### Exemples

```
TEMP = ISMEMBER(25, COLUMN(1...10))
```

Crée une colonne nommée TEMP contenant la valeur zéro.

TEMP = ISMEMBER("cat", COLUMN("cat", "dog", "bird"))
Crée une colonne nommée TEMP contenant la valeur un.
TEMP = ISMEMBER(V1, V1)
Crée une colonne nommée TEMP ne contenant que des uns.
TEMP = ISMEMBER(V1, V2)
Crée une colonne nommée TEMP, dans laquelle chaque valeur est égale à un si la ligne correspondante de la colonne V1 contient une valeur dans la colonne V2 et à zéro dans le cas contraire.
TEMP = ISMEMBER(V1:V2, V5:V10)
Crée deux colonnes nommées TEMP et VX. La colonne TEMP contient un un si la ligne correspondante de la colonne V1 est un membre des colonnes V5 à V10 et un zéro dans le cas contraire. La colonne VX P contient un un si la ligne correspondante de la colonne V2 est un membre des colonnes V5 à V10 et un zéro dans le cas contraire.
TEMP = ISMEMBER(V1[10:15]:V2, V3[1:100]:V6)
Crée deux colonnes nommées TEMP et VX, contenant des valeurs dans les lignes 1 à 6 (les autres cibles sont vides). Les valeurs de la colonne TEMP sont égales à un si le contenu des lignes 10 à 50 de la colonne V1 sont des membres des lignes 1 à 100 des colonnes V3 à V6. Les valeurs de la colonne VX sont égales à un si le contenu des lignes 10 à 50 de la colonne V2 sont des membres des lignes 1 à 100 des colonnes V3 à V6.

## Fonctions associées

Fonction	Description
ISEVEN	Teste si les valeurs en entrée sont paires (à savoir, divisible par deux)
ISODD	Teste si les valeurs en entrée sont impaires (à savoir, non divisible par deux).

---

## ISODD

### Syntaxe

ISODD(data)

### Paramètres

data

Valeurs numériques à tester pour vérifier si elles sont impaires. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de data, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

### Description

ISODD teste chaque valeur du jeu de données spécifié pour vérifier si elle est impaire. Il renvoie une nouvelle colonne pour chaque colonne d'entrée, chacune contenant un un pour toutes les valeurs impaires (à savoir, le modulo deux de la valeur est un) ou un zéro pour toutes les valeurs non impaires (paires).

**Remarque :** Pour les valeurs autres que des entiers, la fonction macro INT est appliquée au préalable. Par exemple, ISODD(2.5) = 0 car 2 n'est pas impaire.

## Exemples

TEMP = ISODD(-3)
Crée une colonne nommée TEMP contenant la valeur un.
TEMP = ISODD(MERGE(1, 4, 0))
Crée trois colonnes nommées TEMP, VX et VY. TEMP contient la valeur 1, VX contient la valeur 0 et VY contient la valeur 0.
TEMP = ISODD(V1)
Crée une colonne nommée TEMP, dans laquelle chaque valeur correspond au résultat du test de valeur impaire du contenu de la colonne V1.
TEMP = ISODD(V1:V3)
Crée trois colonnes nommées TEMP, VX et VY. Les valeurs de la colonne TEMP correspondent aux résultats du test de valeur impaire du contenu de la colonne V1, les valeurs de la colonne VX correspondent aux résultats du test de valeur impaire du contenu de la colonne V2 et les valeurs de la colonne VY correspondent aux résultats du test de valeur impaire du contenu de la colonne V3.
TEMP = ISODD(V1[10:20])
Crée une colonne nommée TEMP, dans laquelle les 11 premières cibles contiennent les résultats du test de valeur impaire des valeurs des lignes 10 à 20 de la colonne V1. Les autres cibles de TEMP sont vides.
TEMP = ISODD(V1[1:5]:V2)
Crée deux colonnes nommées TEMP et VX, contenant des valeurs dans les lignes 1 à 5 (les autres cibles sont vides). Les valeurs de la colonne TEMP correspondent aux résultats du test de valeur impaire des lignes correspondantes de la colonne V1 et les valeurs de la colonne VX correspondent aux résultats du test de valeur impaire des lignes correspondantes de la colonne V2.

## Fonctions associées

Fonction	Description
ISEVEN	Teste si les valeurs en entrée sont paires (à savoir, divisible par deux)
ISMEMBER	Teste une plage en entrée par rapport à une "table" de valeurs et renvoie un si une valeur se trouve dans la table et zéro dans le cas contraire

---

## KURTOSIS



### Syntaxe

KURTOSIS(data [, keyword])

## Paramètres

data

Valeurs numériques dont le kurtosis doit être calculé. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de data, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM. data doit comporter au moins quatre valeurs.

keyword

Ce mot clé facultatif détermine la manière dont le calcul est effectué sur la plage de données d'entrée. Sélectionnez l'une des valeurs suivantes :

ALL

Effectue le calcul sur toutes les cibles de data (valeur par défaut)

COL

Effectue le calcul séparément pour chaque colonne de data

ROW

Effectue le calcul séparément pour chaque ligne de data

Pour plus de détails sur l'utilisation des mots clés dans IBM Campaign, voir «Spécifications de format», à la page 9.

Pour plus de détails sur l'utilisation des mots clés dans IBM PredictiveInsight, voir «Spécifications de format», à la page 27.

## Description

KURTOSIS calcule le kurtosis des valeurs de la plage de données spécifiée. Le kurtosis mesure si une distribution est relativement pointue ou aplatie par rapport à une distribution normale. Plus le kurtosis est négatif, plus la distribution est plate. Plus le kurtosis est positif, plus la distribution est pointue.

Le kurtosis est calculé comme suit :

$$\left\{ \frac{n(n+1)}{(n-1)(n-2)(n-3)} \sum_{j=1}^n \left( \frac{x_j - \text{mean}}{\sigma} \right)^2 \right\} - \frac{3(n-1)^2}{(n-2)(n-3)}$$

où  $n$  représente le nombre d'échantillons dans la distribution,  $\text{mean}$  correspond à la moyenne et  $\sigma$  est l'écart-type de la distribution. Au moins trois valeurs doivent être fournies pour calculer le kurtosis.

## Exemples

<p>TEMP = KURTOSIS(MERGE(3, 4, 5)) ou TEMP = KURTOSIS(MERGE(3, 4, 5), ALL)</p> <p>Crée une colonne nommée TEMP contenant la valeur -1.5.</p>
<p>TEMP = KURTOSIS(V1)</p> <p>Crée une colonne nommée TEMP contenant une valeur unique qui correspond au kurtosis du contenu de la colonne V1.</p>
<p>TEMP = KURTOSIS(V1:V3)</p> <p>Crée une colonne nommée TEMP contenant une valeur unique qui correspond au kurtosis du contenu des colonnes V1, V2 et V3.</p>
<p>TEMP = KURTOSIS(V1[10:20])</p> <p>Crée une colonne nommée TEMP contenant une valeur unique qui correspond au kurtosis des cibles des lignes 10 à 20 de la colonne V1.</p>
<p>TEMP = KURTOSIS(V1[1:5]:V4)</p> <p>Crée une colonne nommée TEMP contenant une valeur unique qui correspond au kurtosis des cibles des lignes 1 à 5 des colonnes V1 à V4.</p>
<p>TEMP = KURTOSIS(V1:V3, COL)</p> <p>Crée trois colonnes nommées TEMP, VX et VY. La valeur unique de la colonne TEMP correspond au kurtosis du contenu de la colonne V1, la valeur unique de la colonne VX correspond au kurtosis du contenu de la colonne V2 et la valeur unique de la colonne VY correspond au kurtosis du contenu de la colonne V3.</p>
<p>TEMP = KURTOSIS(MERGE(1,4), COL)</p> <p>Crée deux colonnes nommées TEMP et VX, contenant chacune la valeur -3.</p>
<p>TEMP = KURTOSIS(V1[1:5]:V3, COL) ou TEMP = KURTOSIS(V1[1:5]:V3[1:5], COL)</p> <p>Crée trois colonnes nommées TEMP, VX et VY contenant chacune une seule valeur. La valeur de la colonne TEMP correspond au kurtosis des cibles des lignes 1 à 5 de la colonne V1, la valeur de la colonne VX correspond au kurtosis des cibles des lignes 1 à 5 de la colonne V2, et la valeur de la colonne VY correspond au kurtosis des cibles des lignes 1 à 5 de la colonne V3.</p>
<p>TEMP = KURTOSIS(V1:V3, ROW)</p> <p>Crée une colonne nommée TEMP dans laquelle chaque entrée de cible correspond au kurtosis de la ligne correspondante dans les colonnes V1, V2 et V3.</p>
<p>TEMP = KURTOSIS(V1[1:5]:V3],ROW) ou TEMP = KURTOSIS(V1[1:5]:V3[1:5], ROW)</p> <p>Crée une colonne nommée TEMP, dans laquelle les cibles des lignes 1 à 5 contiennent le kurtosis de la ligne correspondante dans les colonnes V1 à V3. Les autres cibles de TEMP sont vides.</p>

## Fonctions associées

Fonction	Description
STAT	Calcule les premier, deuxième, troisième et quatrième moments de la plage de données spécifiée
SKEW	Calcule l'asymétrie de la distribution d'une plage de cibles
STDV ou STDEV	Calcule l'écart type d'une plage de cibles

Fonction	Description
VARIANCE	Calcule la variance d'une plage de cibles

## LAG



### Syntaxe

LAG(lag, data)

### Paramètres

lag

Nombre d'intervalles de temps à décaler. Cette valeur doit correspondre à un entier positif.

data

Valeurs à décaler. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de data, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

### Description

LAG renvoie les valeurs de la plage de données d'entrée, décalées du nombre d'intervalles de temps spécifié. Il considère chaque colonne d'entrée comme une série de données dans le temps et renvoie une nouvelle colonne pour chaque colonne d'entrée. Chaque nouvelle colonne contient les valeurs décalées dans le temps (décalage de lag nombre d'intervalles de temps) des nombres de la colonne d'entrée correspondante. Les lag premières valeurs des colonnes renvoyées sont des zéros. Les longueurs des colonnes renvoyées sont les longueurs des colonnes d'entrée correspondantes + lag.

**Remarque :** La fonction macro LAG renvoie une colonne contenant des valeurs telles que la cible  $VY[x] = data[x - lag]$  pour  $x \geq lag$  ; sinon, 0.

### Exemples

```
TEMP = LAG(1, COLUMN(1,2,3,4))
```

Crée une colonne nommée TEMP contenant les valeurs 0, 1, 2, 3 et 4 dans les cibles 1 à 5, respectivement.

```
TEMP = LAG(2, V1)
```

Crée une colonne nommée TEMP, dans laquelle chaque valeur correspond au contenu de la colonne V1 décalé de deux intervalles de temps.

<p>TEMP = LAG(10, V1:V3)</p> <p>Crée trois colonnes nommées TEMP, VX et VY. Les valeurs de la colonne TEMP correspondent au contenu de la colonne V1 décalé de dix intervalles de temps, les valeurs de la colonne VX correspondent au contenu de la colonne V2 décalé de dix intervalles de temps et les valeurs de la colonne VY correspondent au contenu de la colonne V3 décalé de dix intervalles de temps.</p>
<p>TEMP = LAG(5, V1[10:20])</p> <p>Crée une colonne nommée TEMP, dans laquelle les 5 premières cibles contiennent 0, suivi des lignes 10 à 20 de la colonne V1. Les autres cibles de TEMP sont vides.</p>
<p>TEMP = LAG(2, V1[1:5]:V2)</p> <p>Crée deux colonnes nommées TEMP et VX, contenant des valeurs dans les lignes 1 à 7 (les autres cibles sont vides). Les valeurs des lignes 1 à 2 de chaque colonne sont des zéros. Les valeurs restantes de TEMP correspondent aux valeurs des lignes 1 à 5 de la colonne V1. Les valeurs restantes de la colonne VX correspondent aux lignes 1 à 5 de la colonne V2.</p>

## Fonctions associées

Fonction	Description
DELAY	Renvoie les valeurs des colonnes en entrée retardées du nombre d'intervalles de temps spécifié
SLIDE_WINDOW	Crée un modèle à partir d'une fenêtre spécifiée et le fait glisser pour créer le modèle suivant

---

## LE

### Syntaxe

data1 LE data2 data1 <= data2

### Paramètres

data1

Plage de cibles numériques à comparer. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de data, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

data2

Nombres auxquels toutes les valeurs de la colonne spécifiée doivent être comparées. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de data, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.



## Description

LE compare les deux plages de données spécifiées, puis renvoie un un si les valeurs du premier jeu de données sont inférieures ou égales à celles du deuxième jeu de données ou un zéro dans le cas contraire. Il renvoie une nouvelle colonne pour chaque colonne d'entrée, chacune contenant la colonne correspondante de data1 comparée à la colonne correspondante de data2 (à savoir, la première colonne de data1 est comparée à la première colonne de data, la deuxième colonne à la deuxième colonne et ainsi de suite).

Si data2 est une constante, chaque valeur de data est comparée à cette valeur. Si data2 est une colonne, les calculs sont effectués ligne par ligne. Les valeurs de data1 sont comparées à la valeur de la première ligne de data2, la deuxième ligne à la deuxième ligne et ainsi de suite. Ce calcul ligne par ligne génère un résultat pour chaque ligne jusqu'à la dernière valeur de la colonne la plus courte.

**Remarque :** L'opérateur LE peut être abrégé en signe inférieur suivi d'un signe égal (<=).

Cette macro est disponible dans IBM Interact.

## Exemples

TEMP = 4 LE 4 ou TEMP = 4 <= 4
Crée une colonne nommée TEMP contenant la valeur un (car quatre est égal à lui-même).
TEMP = V1 <= 8
Crée une colonne nommée TEMP, dans laquelle chaque valeur est égale à un si la ligne correspondante de la colonne V1 est inférieure ou égale à huit ; sinon, la valeur est égale à zéro.
TEMP = V1:V3 <= 2
Crée trois colonnes nommées TEMP, VX et VY. Les valeurs de la colonne TEMP correspondent au contenu de la colonne V1 comparé à la valeur deux, les valeurs de la colonne VX correspondent au contenu de la colonne V2 comparé à la valeur deux, et les valeurs de la colonne VY correspondent au contenu de la colonne V3 comparé à la valeur deux.
TEMP = V1 <= V1
Crée une colonne nommée TEMP ne contenant que des uns (car tout nombre est égal à lui-même).
TEMP = V1 <= V2
Crée une colonne nommée TEMP, dans laquelle chaque valeur est la valeur de ligne de la colonne V1 comparée à la valeur de ligne correspondante de la colonne V2.
TEMP = V1[10:20] <= V2 ou TEMP = V1[10:20] <= V2[1:11]
Crée une colonne nommée TEMP, dans laquelle les 11 premières cibles contiennent les résultats de la comparaison des valeurs des lignes 10 à 20 de la colonne V1 aux valeurs des lignes 1 à 11 de la colonne V2. Les autres cibles de TEMP sont vides.

## Fonctions associées

Fonction	Description
EQ	Renvoie TRUE si une plage de données est égale à une autre

Fonction	Description
GE	Renvoie TRUE si une plage de données est supérieure ou égale à une autre
GT	Renvoie TRUE si une plage de données est supérieure à une autre
LT	Renvoie TRUE si une plage de données est inférieure à une autre
NE	Renvoie TRUE si une plage de données n'est pas égale à une autre

---

## LIKE

### Syntaxe

data1 [NOT] LIKE data2

### Paramètres

data1

Plage de cibles à comparer. Il peut s'agir d'une chaîne de texte ou d'une expression convertie en chaîne de texte. Pour la définition du format de data, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

data2

Masque textuel auxquels toutes les valeurs de la colonne spécifiée doivent être comparées. Il peut s'agir d'une chaîne de texte ou d'une expression convertie en chaîne de texte. Le nombre de colonnes dans data2 doit correspondre au nombre de colonnes dans data1, à moins que data2 ne soit une constante. Pour la définition du format de data, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

Un trait de soulignement ( \_ ) dans data2 représente un caractère générique correspondant à tout caractère de data1. Un signe pourcentage (%) correspond à zéro, un ou plusieurs caractères de data1.

### Description

LIKE compare les deux plages de données spécifiées, puis renvoie un un si les chaînes correspondent ou un zéro dans le cas contraire. Il renvoie une nouvelle colonne pour chaque colonne d'entrée, chacune contenant la colonne correspondante de data1 comparée à la colonne correspondante de data2 (à savoir, la première colonne de data1 est comparée à la première colonne de data2, la deuxième colonne à la deuxième colonne et ainsi de suite).

Si data2 est une constante de chaîne, chaque chaîne de data1 est comparée à cette chaîne. Si data2 est une colonne, les calculs sont effectués ligne par ligne. La chaîne de la première ligne de data1 est comparée à la chaîne de la première ligne de data2, la deuxième ligne à la deuxième ligne et ainsi de suite. Ce calcul ligne par ligne génère un résultat pour chaque ligne jusqu'à la dernière chaîne de la colonne la plus courte.

Lors de la comparaison de chaînes, la casse n'est pas importante ("Oui", "OUI", "oui" et "ouI" sont considérées comme étant égales).

**Remarque :** La macro LIKE possède une version négative, NOT LIKE. Son format est identique à celui de LIKE. NOT LIKE renvoie un un si la chaîne de data1 ne correspond pas au modèle défini par data2.

Cette macro est disponible dans IBM Interact.

## Exemples

TEMP = "gold" LIKE "gold"
Crée une colonne nommée TEMP contenant la valeur un (car les deux chaînes correspondent).
TEMP = "No" LIKE "NO"
Crée une colonne nommée TEMP contenant la valeur un (la casse n'est pas importante dans les comparaisons de chaînes).
TEMP = V1 LIKE "gold%"
Crée une colonne nommée TEMP, dans laquelle chaque valeur est égale à un si la ligne correspondante de la colonne V1 est égale à la chaîne "gold" suivie d'un ou de plusieurs caractères. Sinon, chaque valeur est égale à zéro.
TEMP = V1 LIKE "g_ld"
Crée une colonne nommée TEMP, dans laquelle chaque valeur est égale à un si la ligne correspondante de la colonne V1 est égale à la chaîne "g" suivie de tout caractère, lui-même suivi de "ld". Sinon, chaque valeur est égale à zéro.
TEMP = V1 LIKE V1
Crée une colonne nommée TEMP ne contenant que des uns (car tout nombre est égal à lui-même).
TEMP = V1 LIKE V2
Crée une colonne nommée TEMP, dans laquelle chaque valeur est la valeur de ligne de la colonne V1 comparée à la valeur de ligne correspondante de la colonne V2.
TEMP = V1:V3 LIKE V4:V6
Crée trois colonnes nommées TEMP, VX et VY. La colonne TEMP contient les chaînes de V1 comparées aux chaînes de ligne correspondantes de la colonne V4. La colonne VX compare les colonnes V2 et V5. La colonne VY compare les colonnes V3 et V6.
TEMP = V1[10:20] LIKE V2 ou TEMP = V1[10:20] LIKE V2[1:11]
Crée une colonne nommée TEMP, dans laquelle les 11 premières cibles contiennent les résultats de la comparaison des chaînes des lignes 10 à 20 de la colonne V1 aux lignes 1 à 11 de la colonne V2. Les autres cibles de TEMP sont vides.

## Fonctions associées

Fonction	Description
EQ	Renvoie TRUE si une plage de données est égale à une autre
GE	Renvoie TRUE si une plage de données est supérieure ou égale à une autre

Fonction	Description
GT	Renvoie TRUE si une plage de données est supérieure à une autre
LE	Renvoie TRUE si une plage de données est inférieure ou égale à une autre
LT	Renvoie TRUE si une plage de données est inférieure à une autre
NE	Renvoie TRUE si une plage de données n'est pas égale à une autre

## LN ou LOG

### Syntaxe

LN(data) ou LOG(data)

### Paramètres

data

Valeurs numériques dont le logarithme naturel doit être calculé. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de data, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

### Description

LN ou LOG calcule le logarithme naturel de chaque valeur de la plage de données spécifiée. Il renvoie une nouvelle colonne par colonne d'entrée, chacune contenant le logarithme naturel des nombres de la colonne d'entrée correspondante. Les logarithmes naturels sont basés sur la constante  $e = 2.7182818$ . LN est l'inverse de la fonction macro EXP.

**Remarque :** Toutes les valeurs de la plage de données spécifiée doivent être supérieures à zéro. Dans le cas contraire, une cible vide est renvoyée pour chaque entrée non valide.

### Exemples

TEMP = LN(3) ou TEMP = LOG(3)
Crée une colonne nommée TEMP contenant la valeur 1.099.
TEMP = LN(V1)
Crée une colonne nommée TEMP, dans laquelle chaque valeur correspond au logarithme naturel du contenu de la colonne V1.
TEMP = LN(V1:V3)
Crée trois colonnes nommées TEMP, VX et VY. Les valeurs de la colonne TEMP correspondent aux logarithmes naturels du contenu de la colonne V1, les valeurs de la colonne VX correspondent aux naturels du contenu de la colonne V2 et les valeurs de la colonne VY correspondent aux logarithmes naturels du contenu de la colonne V3.

TEMP = LN(V1[10:20])

Crée une colonne nommée TEMP, dans laquelle les 11 premières cibles contiennent les logarithmes naturels des valeurs des lignes 10 à 20 de la colonne V1. Les autres cibles de TEMP sont vides.

TEMP = LN(V1[1:5]:V2)

Crée deux colonnes nommées TEMP et VX, contenant des valeurs dans les lignes 1 à 5 (les autres cibles sont vides). Les valeurs de la colonne TEMP correspondent aux logarithmes naturels des lignes correspondantes de la colonne V1 et les valeurs de la colonne VX correspondent aux logarithmes naturels des lignes correspondantes de la colonne V2.

## Fonctions associées

Fonction	Description
EXP	Calcule le nombre naturel (e) élevé à la puissance du contenu de chaque cible de la plage de données spécifiée
LOG2	Calcule le logarithme base 2 du contenu de la plage de données spécifiée
LOG10	Calcule le logarithme base 10 du contenu de la plage de données spécifiée
POW	Calcule une valeur de base élevée aux puissances exponentielles spécifiées

---

## LOG2

### Syntaxe

LOG2(data)

### Paramètres

data

Valeurs numériques dont le logarithme base 2 doit être calculé. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de data, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

### Description

LOG2 calcule le logarithme base 2 des valeurs de la plage de données spécifiée. Il renvoie une nouvelle colonne par colonne d'entrée, chacune contenant le logarithme base 2 des nombres de la colonne d'entrée correspondante.

**Remarque :** Toutes les valeurs de la plage de données spécifiée doivent être supérieures à zéro. Dans le cas contraire, une cible vide est renvoyée pour chaque entrée non valide.

## Exemples

TEMP = LOG2(8)
Crée une colonne nommée TEMP contenant la valeur trois.
TEMP = LOG2(V1)
Crée une colonne nommée TEMP, dans laquelle chaque valeur correspond au logarithme base 2 du contenu de la colonne V1.
TEMP = LOG2(V1:V3)
Crée trois colonnes nommées TEMP, VX et VY. Les valeurs de la colonne TEMP correspondent aux logarithmes base 2 du contenu de la colonne V1, les valeurs de la colonne VX correspondent aux logarithmes base 2 du contenu de la colonne V2 et les valeurs de la colonne VY correspondent aux logarithmes base 2 du contenu de la colonne V3.
TEMP = LOG2(V1[10:20])
Crée une colonne nommée TEMP, dans laquelle les 11 premières cibles contiennent les logarithmes base 2 des valeurs des lignes 10 à 20 de la colonne V1. Les autres cibles de TEMP sont vides.
TEMP = LOG2(V1[1:5]:V2)
Crée deux colonnes nommées TEMP et VX, contenant des valeurs dans les lignes 1 à 5 (les autres cibles sont vides). Les valeurs de la colonne TEMP correspondent aux logarithmes base 2 des lignes correspondantes de la colonne V1 et les valeurs de la colonne VX correspondent aux logarithmes base 2 des lignes correspondantes de la colonne V2.

## Fonctions associées

Fonction	Description
LN ou LOG	Calcule le logarithme naturel du contenu de la plage de données spécifiée
LOG10	Calcule le logarithme base 10 du contenu de la plage de données spécifiée
POW	Puissance exponentielle

---

## LOG10

### Syntaxe

LOG10(data)

### Paramètres

data

Valeurs numériques dont le logarithme base 10 doit être calculé. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de data, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

## Description

LOG10 calcule le logarithme base 10 des valeurs de la plage de données spécifiée. Il renvoie une nouvelle colonne par colonne d'entrée, chacune contenant le logarithme base 10 des nombres de la colonne d'entrée correspondante.

**Remarque :** Toutes les valeurs de la plage de données spécifiée doivent être supérieures à zéro. Dans le cas contraire, une cible vide est renvoyée pour chaque entrée non valide.

## Exemples

TEMP = LOG10(100)
Crée une colonne nommée TEMP contenant la valeur deux.
TEMP = LOG10(V1)
Crée une colonne nommée TEMP, dans laquelle chaque valeur correspond au logarithme base 10 du contenu de la colonne V1.
TEMP = LOG10(V1:V3)
Crée trois colonnes nommées TEMP, VX et VY. Les valeurs de la colonne TEMP correspondent aux logarithmes base 10 du contenu de la colonne V1, les valeurs de la colonne VX correspondent aux logarithmes base 10 du contenu de la colonne V2 et les valeurs de la colonne VY correspondent aux logarithmes base 10 du contenu de la colonne V3.
TEMP = LOG10(V1[10:20])
Crée une colonne nommée TEMP, dans laquelle les 11 premières cibles contiennent les logarithmes base 10 des valeurs des lignes 10 à 20 de la colonne V1. Les autres cibles de TEMP sont vides.
TEMP = LOG10(V1[1:5]:V2)
Crée deux colonnes nommées TEMP et VX, contenant des valeurs dans les lignes 1 à 5 (les autres cibles sont vides). Les valeurs de la colonne TEMP correspondent aux logarithmes base 10 des lignes correspondantes de la colonne V1 et les valeurs de la colonne VX correspondent aux logarithmes base 10 des lignes correspondantes de la colonne V2.

## Fonctions associées

Fonction	Description
LN ou LOG	Calcule le logarithme naturel du contenu de la plage de données spécifiée
LOG2	Calcule le logarithme base 2 du contenu de la plage de données spécifiée
POW	Puissance exponentielle

---

## LOWER

### Syntaxe

LOWER(data)

## Paramètres

data

Valeur de chaîne à convertir en minuscules.

## Description

LOWER convertit chaque valeur de chaîne de la plage de données spécifiée en minuscules. Il renvoie une nouvelle colonne dont chaque cible contient la chaîne en minuscules de la cible source correspondante.

Cette macro est disponible dans IBM Interact.

## Exemples

Temp = LOWER "GOLD"
Crée une colonne nommée Temp contenant "gold".
TEMP = LOWER( "JAN 15, 1997")
Crée une colonne nommée TEMP, qui contient la chaîne de texte ASCII " jan 15, 1997 ".
TEMP = LOWER( "Pressure")
Crée une colonne nommée TEMP, qui contient la chaîne de texte ASCII "pressure".
TEMP = LOWER(V1)
Crée une colonne nommée TEMP contenant les caractères en minuscules de chaque chaîne de la colonne V1.

---

## LT

### Syntaxe

```
data1 LT data2 data1 < data2
```

### Paramètres

data1

Plage de cibles numériques à comparer. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de data, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

data2

Nombres auxquels toutes les valeurs de la colonne spécifiée doivent être comparées. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de data, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.



## Description

LT compare les deux plages de données spécifiées, puis renvoie un un si les valeurs du premier jeu de données sont inférieures à celles du deuxième jeu de données ou un zéro dans le cas contraire. Il renvoie une nouvelle colonne pour chaque colonne d'entrée, chacune contenant la colonne correspondante de data1 comparée à la colonne correspondante de data2 (à savoir, la première colonne de data1 est comparée à la première colonne de data, la deuxième colonne à la deuxième colonne et ainsi de suite).

Si data2 est une constante, chaque valeur de data est comparée à cette valeur. Si data2 est une colonne, les calculs sont effectués ligne par ligne. Les valeurs de data1 sont comparées à la valeur de la première ligne de data2, la deuxième ligne à la deuxième ligne et ainsi de suite. Ce calcul ligne par ligne génère un résultat pour chaque ligne jusqu'à la dernière valeur de la colonne la plus courte.

**Remarque :** L'opérateur LT peut être abrégé en signe inférieur (<).

Cette macro est disponible dans IBM Interact.

## Exemples

TEMP = 3 LT 4 ou TEMP = 3 < 4
Crée une colonne nommée TEMP contenant la valeur un (car trois est inférieur à quatre).
TEMP = V1 < 8
Crée une colonne nommée TEMP, dans laquelle chaque valeur est égale à un si la ligne correspondante de la colonne V1 est inférieure à huit ; sinon, la valeur est égale à zéro.
TEMP = V1:V3 < 2
Crée trois colonnes nommées TEMP, VX et VY. Les valeurs de la colonne TEMP correspondent au contenu de la colonne V1 comparé à la valeur deux, les valeurs de la colonne VX correspondent au contenu de la colonne V2 comparé à la valeur deux, et les valeurs de la colonne VY correspondent au contenu de la colonne V3 comparé à la valeur deux.
TEMP = V1 < V1
Crée une colonne nommée TEMP ne contenant que des zéros (car aucun nombre n'est inférieur à lui-même).
TEMP = V1 < V2
Crée une colonne nommée TEMP, dans laquelle chaque valeur est la valeur de ligne de la colonne V1 comparée à la valeur de ligne correspondante de la colonne V2.
TEMP = V1[10:20] < V2 ou TEMP = V1[10:20] < V2[1:11]
Crée une colonne nommée TEMP, dans laquelle les 11 premières cibles contiennent les résultats de la comparaison des valeurs des lignes 10 à 20 de la colonne V1 aux lignes 1 à 11 de la colonne V2. Les autres cibles de TEMP sont vides.

## Fonctions associées

Fonction	Description
EQ	Renvoie TRUE si une plage de données est égale à une autre
GE	Renvoie TRUE si une plage de données est supérieure ou égale à une autre

Fonction	Description
GT	Renvoie TRUE si une plage de données est supérieure à une autre
LE	Renvoie TRUE si une plage de données est inférieure ou égale à une autre

---

## LTRIM

### Syntaxe

LTRIM(data)

### Paramètres

data

Chaîne de laquelle l'espace à gauche sera supprimé.

### Description

LTRIM supprime les premiers caractères d'espacement de chaque valeur de chaîne contenue dans la plage de données spécifiée, puis renvoie la chaîne convertie. Il renvoie une nouvelle colonne pour chaque colonne d'entrée.

Cette macro est disponible dans IBM Interact.

### Exemples

```
Temp = LTRIM " gold"
```

Crée une chaîne nommée Temp qui contient "gold".

---

## MAX

### Syntaxe

MAX(data [, keyword])

### Paramètres

data

Valeurs numériques dont la valeur maximale doit être calculée. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de data, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

keyword

Ce mot clé facultatif détermine la manière dont le calcul est effectué sur la plage de données d'entrée. Sélectionnez l'une des valeurs suivantes :

ALL - Effectue le calcul sur toutes les cibles de data (valeur par défaut)

COL - Effectue le calcul séparément pour chaque colonne de data

ROW - Effectue le calcul séparément pour chaque ligne de data

Pour plus de détails sur l'utilisation des mots clés dans IBM Campaign, voir «Spécifications de format», à la page 9.

Pour plus de détails sur l'utilisation des mots clés dans IBM PredictiveInsight, voir «Spécifications de format», à la page 27.

**Remarque :** De nombreuses fonctions macro acceptent les paramètres de mot clé {ALL | COL | ROW}. Ces mots clés ne s'appliquent pas dans IBM Campaign car les données d'entrées correspondent toujours à une colonne ou à une zone. La macro se comporte toujours comme si le mot clé COL a été spécifié. Par conséquent, vous n'avez pas besoin de spécifier ces mots clés lorsque vous utilisez IBM Campaign.

## Description

MAX calcule la valeur maximale des valeurs de la plage de données spécifiée. Il renvoie une nouvelle colonne unique contenant la valeur maximale.

Cette macro est disponible dans IBM Interact.

## Exemples

<code>TEMP = MAX(3)</code> ou <code>TEMP = MAX(3, ALL)</code>
Crée une colonne nommée TEMP contenant la valeur trois.
<code>TEMP = MAX(SELECT(COLUMN(1,3,5), V1:V5))</code>
Crée une colonne nommée TEMP contenant une valeur unique qui correspond à la valeur maximale des colonnes V1, V3 et V5.
<code>TEMP = MAX(V1)</code>
Crée une colonne nommée TEMP contenant une valeur unique qui correspond à la valeur maximale du contenu de la colonne V1.
<code>TEMP = MAX(V1:V3)</code>
Crée une colonne nommée TEMP contenant une valeur unique qui correspond à la valeur maximale des colonnes V1, V2 et V3.
<code>TEMP = MAX(V1[10:20])</code>
Crée une colonne nommée TEMP contenant une valeur unique qui correspond à la valeur maximale des cibles des lignes 10 à 20 de la colonne V1.
<code>TEMP = MAX(V1[1:5]:V4)</code>
Crée une colonne nommée TEMP contenant une valeur unique qui correspond à la valeur maximale des cibles des lignes 1 à 5 des colonnes V1 à V4.
<code>TEMP = MAX(V1:V3, COL)</code>
Crée trois colonnes nommées TEMP, VX et VY. La valeur unique de la colonne TEMP correspond à la valeur maximale du contenu de la colonne V1, la valeur unique de la colonne VX correspond à la valeur maximale du contenu de la colonne V2 et la valeur unique de la colonne VY correspond à la valeur maximale du contenu de la colonne V3.

TEMP = MAX(V1[1:5]:V3, COL)
Crée trois colonnes nommées TEMP, VX et VY contenant chacune une seule valeur. La valeur de la colonne TEMP correspond à la valeur maximale des cibles des lignes 1 à 5 de la colonne V1, la valeur de la colonne VX correspond à la valeur maximale des cibles des lignes 1 à 5 de la colonne V2, et la valeur de la colonne VY correspond à la valeur maximale des cibles des lignes 1 à 5 de la colonne V3.
TEMP = MAX(V1:V3, ROW)
Crée une colonne nommée TEMP dans laquelle chaque entrée de cible correspond à la valeur maximale de la ligne correspondante dans les colonnes V1, V2 et V3.
TEMP = MAX(V1[10:20]:V3, ROW)
Crée une colonne nommée TEMP, dans laquelle les 11 premières cibles contiennent la valeur maximale des valeurs des lignes 10 à 20 dans les colonnes V1 à V3. Les autres cibles de TEMP sont vides.

## Fonctions associées

Fonction	Description
DECIMATE	Décime une colonne de nombres en plusieurs colonnes dans lesquelles un "un" indique la valeur d'index.
MAXINDEX	Renvoie l'index de colonne de la n <sup>ème</sup> (première, deuxième, troisième, etc.) valeur maximale de chaque ligne de la colonne spécifiée
MIN	Calcule la valeur minimale d'une plage de cibles

## MAXINDEX



### Syntaxe

MAXINDEX(data [, n])

### Paramètres

data

Début d'une plage de données pour laquelle l'index de la n<sup>ème</sup> valeur maximale de chaque ligne doit être calculé. Il peut s'agir d'une colonne ou d'une expression convertie en colonne. Pour la définition du format de data, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

n

Nombre supérieur à zéro. La valeur par défaut, un, renvoie l'index de la valeur la plus élevée de chaque ligne. (Deux renvoie l'index de la deuxième valeur la plus élevée, trois, l'index de la troisième valeur la plus élevée, etc.)

## Description

MAXINDEX recherche la n<sup>ème</sup> valeur maximale de chaque ligne de la plage de données spécifiée et renvoie un index de colonne pour son emplacement. Il renvoie une nouvelle colonne contenant une valeur unique représentant l'index de la n<sup>ème</sup> valeur maximale de chaque ligne. La valeur un indique la première cible de la première colonne. S'il existe plusieurs valeurs maximales, la première colonne contenant la n<sup>ème</sup> valeur maximale est renvoyée.

**Remarque :** Si vous avez plusieurs colonnes, chacune d'elles représentant une classe de sortie distincte, vous pouvez utiliser MAXINDEX pour sélectionner la "classe gagnante". Vous pouvez ensuite apprendre avec une seule colonne en sortie au lieu de plusieurs colonnes en sortie. Par exemple, V4 = MAXINDEX(V1:V3) génère :

V1 V2 V3 V4 0 1 0 2 1 0 0 1 0 0 1 3

## Exemples

TEMP = MAXINDEX(MERGE(3,5,-2))
Crée une colonne nommée TEMP contenant la valeur deux car la valeur maximale est cinq (deuxième colonne).
TEMP = MAXINDEX(V1)
Crée une colonne nommée TEMP contenant un un pour chaque ligne de la colonne V1.
TEMP=MAXINDEX (V6:V8,3)
Crée une colonne nommée TEMP dont chaque valeur représente l'index de la valeur minimale (dans le cas présent, la troisième la plus élevée au dessus de 3) de la ligne correspondante dans les colonnes V6, V7 et V8. Un un est renvoyé si la valeur minimale se trouve dans la colonne V6, un deux est renvoyé si la valeur minimale se trouve dans la colonne V7 et un trois est renvoyé si la valeur minimale se trouve dans la colonne V8
TEMP = MAXINDEX(V6:V8)
Crée une colonne nommée TEMP dont chaque valeur représente l'index de la valeur maximale de la ligne correspondante dans les colonnes V6, V7 et V8. Un un est renvoyé si la valeur maximale se trouve dans la colonne V6, un deux est renvoyé si la valeur maximale se trouve dans la colonne V7 et un trois est renvoyé si la valeur maximale se trouve dans la colonne V8.
TEMP = MAXINDEX(V1[1:5]:V3)
Crée une colonne nommée TEMP, dans laquelle les cibles des lignes 1 à 5 contiennent une valeur représentant l'index de la valeur maximale de la ligne correspondante dans les colonnes V1 à V3. Les autres cibles de TEMP sont vides.

## Fonctions associées

Fonction	Description
DECIMATE	Décime une colonne de nombres en plusieurs colonnes dans lesquelles un un indique la valeur d'index
MAX	Calcule la valeur maximale d'une plage de cibles
MIN	Calcule la valeur minimale d'une plage de cibles

---

# MEAN

## Syntaxe

MEAN(data [, keyword])

## Paramètres

data

Valeurs numériques dont la moyenne arithmétique doit être calculée. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de data, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

keyword

Ce mot clé facultatif détermine la manière dont le calcul est effectué sur la plage de données d'entrée. Sélectionnez l'une des valeurs suivantes :

ALL - Effectue le calcul sur toutes les cibles de data (valeur par défaut)

COL - Effectue le calcul séparément pour chaque colonne de data

ROW - Effectue le calcul séparément pour chaque ligne de data

Pour plus de détails sur l'utilisation des mots clé, voir «DATE», à la page 77.

**Remarque :** De nombreuses fonctions de macro utilisent les paramètres de mot clé {ALL | COL | ROW}. Ces mots clés ne s'appliquent pas dans **IBM Campaign** car les données d'entrées correspondent toujours à une colonne ou à une zone. La macro se comporte toujours comme si le mot clé COL a été spécifié. Par conséquent, vous n'avez pas besoin de spécifier ces mots clés lorsque vous utilisez **IBM Campaign**.

## Description

MEAN calcule la moyenne arithmétique des cibles de la plage de données spécifiée. Pour calculer la moyenne arithmétique, le contenu de l'ensemble des cibles est additionné, puis le résultat est divisé par le nombre de cible. Le nombre de colonnes renvoyé par MEAN dépend de keyword.

- Si keyword a la valeur ALL, MEAN renvoie une nouvelle colonne, contenant une valeur unique (la moyenne de toutes les cibles de data).
- Si keyword a la valeur COL, MEAN renvoie une nouvelle colonne pour chaque colonne d'entrée. Chaque nouvelle colonne contient une valeur (moyenne de toutes les cibles de la colonne d'entrée correspondante).
- Si keyword a la valeur ROW, MEAN renvoie une nouvelle colonne contenant la moyenne de chaque ligne de data.

**Remarque :** Les cibles vides sont exclues de la moyenne.

**Remarque :** MEAN est identique à la fonction macro AVG.

Cette macro est disponible dans IBM Interact.

## Exemples

<p>TEMP = MEAN(MERGE(3, 4, 5)) ou TEMP = MEAN(MERGE(3, 4, 5), ALL)</p> <p>Crée une colonne nommée TEMP contenant la valeur 4.</p>
<p>TEMP = MEAN(MERGE(-10, 6, 10))</p> <p>Crée une colonne nommée TEMP contenant la valeur 2.</p>
<p>TEMP = MEAN(V1)</p> <p>Crée une colonne nommée TEMP contenant une valeur unique qui correspond à la moyenne arithmétique du contenu de la colonne V1.</p>
<p>TEMP = MEAN(V1:V3)</p> <p>Crée une colonne nommée TEMP contenant une valeur unique qui correspond à la moyenne arithmétique du contenu des colonnes V1, V2 et V3.</p>
<p>TEMP = MEAN(V1[10:20])</p> <p>Crée une colonne nommée TEMP contenant une valeur unique qui correspond à la moyenne arithmétique des cibles des lignes 10 à 20 de la colonne V1.</p>
<p>TEMP = MEAN(V1[1:5]:V4)</p> <p>Crée une colonne nommée TEMP contenant une valeur unique qui correspond à la moyenne arithmétique des cibles des lignes 1 à 5 des colonnes V1 à V4.</p>
<p>TEMP = MEAN(V1:V3, COL)</p> <p>Crée trois colonnes nommées TEMP, VX et VY. La valeur unique de la colonne TEMP correspond à la moyenne arithmétique du contenu de la colonne V1, la valeur unique de la colonne VX correspond à la moyenne arithmétique du contenu de la colonne V2 et la valeur unique de la colonne VY correspond à la moyenne arithmétique du contenu de la colonne V3.</p>
<p>TEMP = MEAN(MERGE(1,4),COL)</p> <p>Crée deux colonnes nommées TEMP et VX. TEMP contient uniquement la valeur un ; VX contient uniquement la valeur quatre.</p>
<p>TEMP = MEAN(V1[10:20]:V3, COL)</p> <p>Crée trois colonnes nommées TEMP, VX et VY contenant chacune une seule valeur. La valeur de la colonne TEMP correspond à la moyenne arithmétique des cibles des lignes 10 à 20 de la colonne V1, la valeur de la colonne VX correspond à la moyenne arithmétique des cibles des lignes 10 à 20 de la colonne V2, et la valeur de la colonne VY correspond à la moyenne arithmétique des cibles des lignes 10 à 20 de la colonne V3.</p>
<p>TEMP = MEAN(V1:V3, ROW)</p> <p>Crée une colonne nommée TEMP dans laquelle chaque entrée de cible correspond à la moyenne arithmétique de la ligne correspondante dans les colonnes V1, V2 et V3.</p>
<p>TEMP = MEAN(V1[1:5]:V3,ROW)</p> <p>Crée une colonne nommée TEMP, dans laquelle les cibles des lignes 1 à 5 contiennent la moyenne arithmétique de la ligne correspondante dans les colonnes V1 à V3. Les autres cibles de TEMP sont vides.</p>

## Fonctions associées

Fonction	Description
AVG_DEV	Calcule l'écart moyen d'une plage de cibles

Fonction	Description
SUM ou TOTAL	Calcule la somme d'une plage de cibles

## MERGE



### Syntaxe

MERGE(data [, data]...) {data [, data]...}

### Paramètres

data

Nom d'une colonne à combiner dans une plage de données. Il peut s'agir d'une valeur constante (valeur numérique ou texte ASCII entre guillemets), d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Ce paramètre peut être répété plusieurs fois. Pour la définition du format de data, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

### Description

MERGE concatène horizontalement son entrée dans un nouveau groupe. Il renvoie le nombre de nouvelles colonnes fournies en entrée. Un nombre illimité d'arguments peuvent être fournis.

**Remarque :** La fonction macro MERGE peut être spécifiée à l'aide d'accolades ( {} ). Insérez simplement les arguments dans des accolades en les séparant par des virgules (par exemple, TEMP = {1,2,3} est équivalent à TEMP = MERGE(1,2,3)).

### Exemples

TEMP = MERGE(3, 4, "five") ou TEMP = {3, 4, "five"}

Crée trois colonnes nommées TEMP, VX et VY contenant respectivement les valeurs 3, 4 et "five".

TEMP = V1:V3 ou TEMP = MERGE(V1:V3)

Crée trois colonnes nommées TEMP, VX et VY, où TEMP est une copie de la colonne V1, VX est une copie de la colonne V2 et VY est une copie de la colonne V3.

TEMP = MERGE(V1, V3, V5:V7)

Crée cinq colonnes nommées TEMP, VW, VX, VY et VZ. TEMP est une copie de la colonne V1 ; VW est une copie de la colonne V3 ; VX à VZ sont des copies des colonnes V5 à V7.

TEMP = AVG(MERGE(V1,V3,V5), ROW)

Crée une colonne nommée TEMP, dans laquelle chaque ligne correspond à la moyenne de la ligne correspondante dans les colonnes V1, V3 et V5. Une moyenne est calculée uniquement jusqu'à la dernière ligne de la colonne d'entrée la plus courte.



```
TEMP = MERGE(V1[10:50],V3, V5:V7[1:30])
```

Crée cinq colonnes nommées TEMP, VW, VX, VY et VZ. TEMP est une copie des valeurs des lignes 10 à 50 de la colonne V1 ; VW est une copie des valeurs de la colonne V3 ; VX à VZ sont des copies des valeurs des lignes 1 à 30 des colonnes V5 à V7.

```
TEMP = AVG(MERGE(V1, V5:V6)) ou TEMP = AVG({V1, V5:V6})
```

Crée une colonne nommée TEMP contenant la moyenne de toutes les cibles des colonnes V1, V5 et V6.

## Fonctions associées

Fonction	Description
COLUMN	Crée des colonnes, en concaténant verticalement les valeurs en entrée dans chaque colonne
SELECT	Renvoie les colonnes spécifiées d'une plage de données
TRANSPOSE	Transpose une plage de données spécifiée

---

## MIN

### Syntaxe

```
MIN(data [, keyword])
```

### Paramètres

data

Valeurs numériques dont la valeur minimale doit être calculée. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de data, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

keyword

Ce mot clé facultatif détermine la manière dont le calcul est effectué sur la plage de données d'entrée. Sélectionnez l'une des valeurs suivantes :

ALL - Effectue le calcul sur toutes les cibles de data (valeur par défaut)

COL - Effectue le calcul séparément pour chaque colonne de data

ROW - Effectue le calcul séparément pour chaque ligne de data

Pour plus de détails sur l'utilisation des mots clé, voir «DATE», à la page 77.

**Remarque :** De nombreuses fonctions de macro utilisent les paramètres de mot clé {ALL | COL | ROW}. Ces mots clés ne s'appliquent pas dans **IBM Campaign** car les données d'entrées correspondent toujours à une colonne ou à une zone. La macro se comporte toujours comme si le mot clé COL a été spécifié. Par conséquent, vous n'avez pas besoin de spécifier ces mots clés lorsque vous utilisez **IBM Campaign**.

## Description

MIN calcule la valeur minimale de toutes les cibles de la plage de données spécifiée. Il renvoie une colonne unique contenant la valeur minimale.

Cette macro est disponible dans IBM Interact.

## Exemples

<code>TEMP = MIN(MERGE(1,10,-2))</code>	Crée une colonne nommée TEMP contenant la valeur -2.
<code>TEMP = MIN(V1)</code>	Crée une colonne nommée TEMP contenant une valeur unique qui correspond à la valeur minimale de la colonne V1.
<code>TEMP = MIN(V1:V3)</code>	Crée une colonne nommée TEMP contenant une valeur unique qui correspond à la valeur minimale des colonnes V1, V2 et V3.
<code>TEMP = MIN(V1[10:20])</code>	Crée une colonne nommée TEMP contenant une valeur unique qui correspond à la valeur minimale des cibles des lignes 10 à 20 de la colonne V1.
<code>TEMP = MIN(V1[1:5]:V4)</code>	Crée une colonne nommée TEMP contenant une valeur unique qui correspond à la valeur minimale des cibles des lignes 1 à 5 des colonnes V1 à V4.
<code>TEMP = MIN(V1:V3, COL)</code>	Crée trois colonnes nommées TEMP, VX et VY. La valeur unique de la colonne TEMP correspond à la valeur minimale de la colonne V1, la valeur unique de la colonne VX correspond à la valeur minimale de la colonne V2 et la valeur unique de la colonne VY correspond à la valeur minimale de la colonne V3.
<code>TEMP = MIN(V1[1:5]:V3, COL)</code>	Crée trois colonnes nommées TEMP, VX et VY contenant chacune une seule valeur. La valeur de la colonne TEMP correspond à la valeur minimale des cibles des lignes 1 à 5 de la colonne V1, la valeur de la colonne VX correspond à la valeur minimale des cibles des lignes 1 à 5 de la colonne V2, et la valeur de la colonne VY correspond à la valeur minimale des cibles des lignes 1 à 5 de la colonne V3.
<code>TEMP = MIN(V1:V3, ROW)</code>	Crée une colonne nommée TEMP dans laquelle chaque entrée de cible correspond à la valeur minimale de la ligne correspondante dans les colonnes V1, V2 et V3.
<code>TEMP = MIN(V1[10:20]:V3, ROW)</code>	Crée une colonne nommée TEMP, dans laquelle les 11 premières cibles contiennent la valeur minimale des valeurs des lignes 1 à 5 dans les colonnes V1 à V3. Les autres cibles de TEMP sont vides.

## Fonctions associées

Fonction	Description
DECIMATE	Décime une colonne de nombres en plusieurs colonnes dans lesquelles un un indique la valeur d'index

Fonction	Description
MAX	Calcule la valeur maximale d'une plage de cibles
MAX_TO_INDEX	Renvoie l'index de colonne de la valeur maximale pour chaque ligne de la colonne spécifiée

---

## MINUS

### Syntaxe

data MINUS subtrahend data - subtrahend

### Paramètres

data

Plage de cibles contenant les nombres desquels une valeur doit être soustraite. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de data, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

subtrahend

Nombres à soustraire de toutes les valeurs de la colonne spécifiée. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Le nombre de colonnes dans subtrahend doit correspondre au nombre de colonnes dans data, à moins que subtrahend ne soit une constante. Pour la définition du format de subtrahend (identique à data), voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

### Description

MINUS soustrait subtrahend de la plage de données spécifiée data. Il renvoie une nouvelle colonne pour chaque colonne d'entrée, chacune contenant la colonne correspondante de subtrahend (à savoir, la première colonne de data soustrait la première colonne de subtrahend, la deuxième colonne soustrait la deuxième colonne et ainsi de suite).

Si subtrahend est une constante, cette valeur est soustraite de chaque valeur de data. Si subtrahend contient une ou plusieurs colonnes, les calculs sont effectués ligne par ligne entre une colonne de data et une colonne de subtrahend. La première ligne de data soustrait la valeur de la première ligne de subtrahend, la seconde ligne soustrait la valeur de la deuxième ligne, etc. Ce calcul ligne par ligne génère un résultat pour chaque ligne jusqu'à la dernière valeur de la colonne la plus courte.

**Remarque :** L'opérateur MINUS peut être abrégé en signe moins ou tiret (-).

Cette macro est disponible dans IBM Interact.

## Exemples

TEMP = 7 MINUS 4 ou TEMP = 7 - 4 Crée une colonne nommée TEMP contenant la valeur trois.
TEMP = V1 - 8 Crée une colonne nommée TEMP, dans laquelle chaque valeur correspond au contenu de la colonne V1 moins huit.
TEMP = V1:V3 - 2 Crée trois colonnes nommées TEMP, VX et VY. Les valeurs de la colonne TEMP correspondent au contenu de la colonne V1 moins deux, les valeurs de la colonne VX correspondent au contenu de la colonne V2 moins deux et les valeurs de la colonne VY correspondent au contenu de la colonne V3 moins deux.
TEMP = V1 - V1 Crée une colonne nommée TEMP ne contenant que des zéros (car tout nombre moins lui-même est égal à zéro).
TEMP = V1 - V2 Crée une colonne nommée TEMP, dans laquelle chaque valeur est la valeur de ligne de la colonne V1 moins la valeur de ligne correspondante de la colonne V2.
TEMP = V1:V3 -V4:V6 Crée trois colonnes nommées TEMP, VX et VY. La colonne TEMP contient les valeurs de V1 moins les valeurs de ligne correspondantes de la colonne V4. La colonne VX soustrait la colonne V5 de V2. La colonne VY soustrait la colonne V6 de V3.
TEMP = V1[10:20] - V2 ou TEMP = V1[10:20] - V2[1:11] Crée une colonne nommée TEMP, dans laquelle les 11 premières cibles contiennent les valeurs des lignes 10 à 20 de la colonne V1 moins les valeurs des lignes 1 à 11 de la colonne V2. Les autres cibles de TEMP sont vides.

## Fonctions associées

Fonction	Description
PLUS	Ajoute le contenu de deux plages de données
SUM ou TOTAL	Calcule la somme d'une plage de cibles

---

## MOD

### Syntaxe

data MOD divisor data % divisor

### Paramètres

data

Valeurs entières dont le modulo doit être calculé. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie

comme l'une de ces valeurs. Pour la définition du format de data, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

divisor

Entier de base non nulle par rapport auquel le modulo doit être calculé. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Le nombre de colonnes dans divisor doit correspondre au nombre de colonnes dans data, à moins que divisor ne soit une constante. Pour la définition du format de divisor (identique à data), voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

## Description

MOD calcule le reste de la division de la plage de données spécifiée par une valeur spécifiée. Ce reste est calculé en divisant par divisor chaque valeur et en renvoyant le reste. Il renvoie une nouvelle colonne par colonne d'entrée, chacune contenant les nombres de data modulo divisor. Le reste possède le même signe (positif ou négatif) que data.

Si divisor est une constante, chaque valeur de la colonne spécifiée est calculée modulo cette valeur. Si divisor est une colonne, les calculs sont effectués ligne par ligne. Les valeurs de data sont calculées modulo la valeur de la première ligne de divisor et le même calcul est effectué entre les deuxièmes lignes, et ainsi de suite. Ce calcul ligne par ligne génère un résultat pour chaque ligne jusqu'à la dernière valeur de la colonne la plus courte.

**Remarque :** Si divisor est égal à zéro, une erreur de division par zéro est renvoyée.

**Remarque :** L'opérateur MOD peut être abrégé en signe pourcentage (%). Par exemple,  $TEMP = 5 \% 3$  est équivalent à  $TEMP = 5 \text{ MOD } 3$ .

Cette macro est disponible dans IBM Interact.

## Exemples

$TEMP = 10 \text{ MOD } 8$ ou $TEMP = 10 \% 8$ Crée une colonne nommée TEMP contenant la valeur 2.
$TEMP = -10 \% 8$ Crée une colonne nommée TEMP contenant la valeur -2.
$TEMP = V1 \% 8$ Crée une colonne nommée TEMP, dans laquelle chaque valeur correspond au contenu de la colonne V1 modulo huit.
$TEMP = V1:V3 \% 2$ Crée trois colonnes nommées TEMP, VX et VY. Les valeurs de la colonne TEMP correspondent aux valeurs modulo deux du contenu de la colonne V1, les valeurs de la colonne VX correspondent aux valeurs modulo deux du contenu de la colonne V2 et les valeurs de la colonne VY correspondent aux valeurs modulo deux du contenu de la colonne V3.

TEMP = V1 % V1
Crée une colonne nommée TEMP contenant un zéro pour chaque entrée de la colonne V1. En effet, tout nombre modulo lui-même renvoie la valeur zéro.
TEMP = V1 % V2
Crée une colonne nommée TEMP, dans laquelle chaque valeur est la valeur de ligne de la colonne V1 modulo la valeur de ligne correspondante de la colonne V2. Notez que si V2=V1, des zéros sont renvoyés, comme dans l'exemple précédent.
TEMP = V1:V3 % V4:V6
Crée trois colonnes nommées TEMP, VX et VY. La colonne TEMP contient les valeurs de V1 modulo les valeurs de ligne correspondantes de la colonne V4. La colonne VX contient les résultats de la colonne V2 modulo V5. La colonne VY contient les résultats de la colonne V3 modulo V6.
TEMP = V1[10:20] % V2 ou TEMP = V1[10:20] % V2[1:11]
Crée une colonne nommée TEMP, dans laquelle les 11 premières cibles correspondent aux valeurs des lignes 10 à 20 de la colonne V1 modulo les valeurs des lignes 1 à 11 de la colonne V2. Les autres cibles de TEMP sont vides.

## Fonctions associées

Fonction	Description
DIV	Divise une plage de données spécifiée par une autre
MOD	Calcule le modulo du contenu de la plage de données spécifiée

---

## MONTHOF

### Syntaxe

MONTHOF(date\_string [, input\_format])

### Paramètres

date\_string

Texte représentant une date valide.

input\_format

Un des mots clés du tableau ci-après, spécifiant le format de date de date\_string.

### Description

MONTHOF renvoie le mois sous la forme d'un nombre pour la date indiquée par date\_string. Si input\_format n'est pas fourni, le mot clé par défaut DELIM\_M\_D\_Y est utilisé.

### Exemples

MONTHOF("012171",MMDDYY) renvoie le nombre 1.

**Remarque :** Pour des informations supplémentaires sur les formats de date valides, voir «DATE», à la page 77.

## Fonctions associées

Fonction	Description
DAYOF	Renvoie le jour de la semaine sous la forme d'un nombre.
WEEKDAYOF	Renvoie le jour de la semaine de la semaine sous forme de nombre.
YEAROF	Renvoie l'année sous la forme d'un nombre.

---

## MULT

### Syntaxe

```
data MULT multiplier data * multiplier
```

### Paramètres

data

Valeurs numériques à multiplier. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de data, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

multiplier

Nombre par lequel toutes les valeurs de la colonne spécifiée doivent être multipliées. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Le nombre de colonnes dans multiplier doit correspondre au nombre de colonnes dans data, à moins que multiplier ne soit une constante. Pour la définition du format de multiplier (identique à data), voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

### Description

MULT multiplie les valeurs des deux plages de données spécifiées. Il renvoie une nouvelle colonne par colonne d'entrée, chacune contenant les nombres de data multipliés par multiplier. Si multiplier est une constante, chaque valeur de data est multipliée par cette valeur. Si multiplier est une colonne, les calculs sont effectués ligne par ligne. Les valeurs de data sont multipliées par la valeur de la première ligne de multiplier, la seconde ligne, par la seconde ligne, etc. Ce calcul ligne par ligne génère un résultat pour chaque ligne jusqu'à la dernière valeur de la colonne la plus courte.

**Remarque :** L'utilisation d'une colonne contenant le même nombre  $x$  dans chaque ligne que multiplier revient à utiliser la constante  $x$  pour multiplier.

**Remarque :** L'opérateur MULT peut être abrégé en astérisque (\*).

Cette macro est disponible dans IBM Interact.

## Exemples

TEMP = 8 MULT 4 ou TEMP = 8 * 4
Crée une colonne nommée TEMP contenant la valeur 32.
TEMP = V1 * 8
Crée une colonne nommée TEMP, dans laquelle chaque valeur correspond au contenu de la colonne V1 multiplié par huit.
TEMP = V1:V3 * 2
Crée trois colonnes nommées TEMP, VX et VY. Les valeurs de la colonne TEMP correspondent à deux fois le contenu de la colonne V1, les valeurs de la colonne VX correspondent à deux fois le contenu de la colonne V2 et les valeurs de la colonne VY correspondent à deux fois le contenu de la colonne V3.
TEMP = V1 * V1
Crée une colonne nommée TEMP contenant le carré de chaque valeur de la colonne V1.
TEMP = V1 * V2
Crée une colonne nommée TEMP, dans laquelle chaque valeur est la valeur de ligne de la colonne V1 multipliée par la valeur de ligne correspondante de la colonne V2.
TEMP = V1:V3 * V4:V6
Crée trois colonnes nommées TEMP, VX et VY. La colonne TEMP contient les valeurs de V1 multipliées par les valeurs de ligne correspondantes de la colonne V4. La colonne VX multiplie la colonne V2 par V5. La colonne VY multiplie la colonne V3 par V6.
TEMP = V1[10:20] * V2 ou TEMP = V1[10:20] * V2[1:11]
Crée une colonne nommée TEMP, dans laquelle les 11 premières cibles contiennent les valeurs des lignes 10 à 20 de la colonne V1 fois les valeurs des lignes 1 à 11 de la colonne V2. Les autres cibles de TEMP sont vides.

## Fonctions associées

Fonction	Description
DIV	Divise une plage de données spécifiée par une autre
EXP	Calcule le nombre naturel (e) élevé à la puissance du contenu de chaque cible de la plage de données spécifiée
POW	Calcule une valeur de base élevée aux puissances exponentielles spécifiées

## NE

### Syntaxe

data1 NE data2 data1 != data2 data1 <> data2

### Paramètres

data1



Plage de cibles à comparer. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de data, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

data2

Nombres auxquels toutes les valeurs de la colonne spécifiée doivent être comparées. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Le nombre de colonnes dans data2 doit correspondre au nombre de colonnes dans data1, à moins que data2 ne soit une constante. Pour la définition du format de data, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

## Description

NE compare les deux plages de données spécifiées, puis renvoie un un si les valeurs ne sont pas égales ou un zéro si elles le sont. Il renvoie une nouvelle colonne pour chaque colonne d'entrée, chacune contenant la colonne correspondante de data1 comparée à la colonne correspondante de data2 (à savoir, la première colonne de data1 est comparée à la première colonne de data, la deuxième colonne à la deuxième colonne et ainsi de suite).

Si data2 est une constante, chaque valeur de data1 est comparée à cette valeur. Si data2 est une colonne, les calculs sont effectués ligne par ligne. Les valeurs de la première ligne de data1 sont comparées à la valeur de la première ligne de data2, la deuxième ligne à la deuxième ligne et ainsi de suite. Ce calcul ligne par ligne génère un résultat pour chaque ligne jusqu'à la dernière valeur de la colonne la plus courte.

**Remarque :** L'utilisation d'une colonne contenant le même nombre  $x$  dans chaque ligne que data2 revient à utiliser la constante  $x$  pour data2.

**Remarque :** L'opérateur NE peut être abrégé en point d'exclamation suivi d'un signe égal (!=) ou en signe inférieur suivi d'un signe supérieur (<=).

Cette macro est disponible dans IBM Interact.

## Exemples

```
TEMP = 3 NE 4 ou TEMP = 3 != 4 TEMP = 3 <= 4
```

Crée une colonne nommée TEMP contenant la valeur un (car trois n'est pas égal à quatre).

```
TEMP = V1 != 8
```

Crée une colonne nommée TEMP, dans laquelle chaque valeur est égale à un si la ligne correspondante de la colonne V1 n'est pas égale à huit ; sinon, la valeur est égale à zéro.

```
TEMP = V1:V3 != 2
```

Crée trois colonnes nommées TEMP, VX et VY. Les valeurs de la colonne TEMP correspondent au contenu de la colonne V1 comparé à la valeur deux, les valeurs de la colonne VX correspondent au contenu de la colonne V2 comparé à la valeur deux, et les valeurs de la colonne VY correspondent au contenu de la colonne V3 comparé à la valeur deux.

TEMP = V1 != V1
Crée une colonne nommée TEMP ne contenant que des zéros (car tout nombre est égal à lui-même).
TEMP = V1 != V2
Crée une colonne nommée TEMP, dans laquelle chaque valeur est la valeur de ligne de la colonne V1 comparée à la valeur de ligne correspondante de la colonne V2.
TEMP = V1:V3 != V4:V6
Crée trois colonnes nommées TEMP, VX et VY. La colonne TEMP contient les valeurs de V1 comparées aux valeurs de ligne correspondantes de la colonne V4. La colonne VX compare les colonnes V2 et V5. La colonne VY compare les colonnes V3 et V6.
TEMP = V1[10:20] != V2 ou TEMP = V1[10:20] != V2[1:11]
Crée une colonne nommée TEMP, dans laquelle les 11 premières cibles contiennent les résultats de la comparaison des valeurs des lignes 10 à 20 de la colonne V1 aux lignes 1 à 11 de la colonne V2. Les autres cibles de TEMP sont vides.

## Fonctions associées

Fonction	Description
EQ	Renvoie TRUE si une plage de données est égale à une autre
GE	Renvoie TRUE si une plage de données est supérieure ou égale à une autre
GT	Renvoie TRUE si une plage de données est supérieure à une autre
LE	Renvoie TRUE si une plage de données est inférieure ou égale à une autre
LT	Renvoie TRUE si une plage de données est inférieure à une autre

---

## NORM\_MINMAX



### Syntaxe

NORM\_MINMAX(data [, keyword]) NORM\_MINMAX(data, min, max [, keyword])  
 NORM\_MINMAX(data, base\_data [, keyword])

### Paramètres

data

Valeurs numériques à normaliser. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de data, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

min, max

Ces deux paramètres indiquent les valeurs minimale et maximale à utiliser pour la normalisation. Il doit s'agir de constantes, excepté avec le mot clé ROW, où il peut s'agir de constantes ou de colonnes.

`base_data`

Ce paramètre indique une plage de données à utiliser pour le calcul des valeurs maximale et minimale à utiliser pour la normalisation. Le nombre de colonnes fournies dans `base_data` doit correspondre au nombre de colonnes indiqué par `data`. Pour la définition du format de `base_data` (identique à `data`), voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

`keyword`

Ce mot clé facultatif détermine la manière dont le calcul est effectué sur la plage de données d'entrée. Sélectionnez l'une des valeurs suivantes :

ALL - Effectue le calcul sur toutes les cibles de data (valeur par défaut)

COL - Effectue le calcul séparément pour chaque colonne de data

ROW - Effectue le calcul séparément pour chaque ligne de data

Pour plus de détails sur l'utilisation des mots clés dans IBM Campaign, voir «Spécifications de format», à la page 9.

Pour plus de détails sur l'utilisation des mots clés dans IBM PredictiveInsight, voir «Spécifications de format», à la page 27.

## Description

NORM\_MINMAX calcule les valeurs normalisées de la plage de données spécifiée. Toutes les valeurs renvoyées sont comprises entre zéro et un inclus. La normalisation min/max est effectuée comme suit :

$$VX[y] = \frac{VX[y] - \min}{\max - \min}$$

où *min* et *max* sont déterminés de la manière suivante :

- Si les valeurs *min* et *max* sont fournies, elles sont respectivement utilisées pour le minimum et le maximum. Si ces paramètres sont fournis avec le mot clé ROW, les valeurs *min* et *max* peuvent correspondre à des colonnes et spécifier les valeurs *min* et *max* de chaque ligne de data. Si *min* et *max* sont des colonnes, ces colonnes doivent être de même longueur que `data` ou être scalaires (dans ce cas, elles contiennent une seule valeur qui est utilisée comme constante appliquée à toutes les valeurs de la colonne correspondante de `data`).
- Si `base_data` est fourni, les valeurs minimale et maximale de cette plage de données sont calculées et utilisées pour normaliser `data`. Les colonnes de `base_data` doivent contenir plusieurs valeurs de cible.
- Si aucune des options mutuellement exclusives ci-dessus n'est fournie, les valeurs minimale et maximale sont automatiquement calculées à partir de `data`.

**Remarque :** Comme toutes les valeurs renvoyées sont comprises entre 0,0 et 1,0, les valeurs calculées à l'aide de l'équation (ci-dessus) inférieures à 0,0 sont renvoyées sous la forme 0,0. De même, les valeurs supérieures à 1,0 sont renvoyées sous la forme 1,0. NORM\_MINMAX renvoie toujours une plage de données avec les mêmes dimensions que la plage de données d'entrée. Le mot clé ALL spécifie le calcul de la moyenne et de l'écart type sur l'intégralité de la plage de données d'entrée. Le mot clé COL spécifie le calcul d'une moyenne et d'un écart type pour chaque colonne d'entrée et l'utilisation de ces valeurs pour normaliser cette colonne. Le mot clé ROW spécifie le calcul d'une moyenne et d'un écart type pour chaque ligne de la plage de données spécifiée et l'utilisation de ces valeurs pour normaliser cette ligne.

**Remarque :** Si les valeurs minimale et maximale sont égales, des zéros sont renvoyés.

## Exemples

<p>TEMP = NORM_MINMAX(3)</p> <p>Crée une colonne nommée TEMP contenant la valeur zéro.</p>
<p>TEMP = NORM_MINMAX(COLUMN(3, 4, 5))</p> <p>Crée une colonne nommée TEMP contenant les valeurs 0, 0.5 et 1. (Les valeurs minimale et maximale [3 et 5] sont calculées automatiquement à partir de la plage de données.)</p>
<p>TEMP = NORM_MINMAX(COLUMN(3, 4, 5), 0, 10)</p> <p>Crée une colonne nommée TEMP contenant les valeurs 0.3, 0.4 et 0.5. (Cette fois les valeurs minimale et maximale [0 et 10] sont fournies comme arguments.)</p>
<p>TEMP = NORM_MINMAX(V1)</p> <p>Crée une colonne nommée TEMP contenant les valeurs normalisées du contenu de la colonne V1. Les valeurs minimale et maximale utilisées pour la normalisation sont calculées sur la colonne V1.</p>
<p>TEMP = NORM_MINMAX(V1:V3)</p> <p>Crée trois colonnes nommées TEMP, VX et VY. Chacune contient respectivement les valeurs normalisées du contenu des colonnes V1, V2 et V3. Les valeurs minimale et maximale utilisées pour la normalisation sont calculées sur les colonnes V1, V2 et V3.</p>
<p>TEMP = NORM_MINMAX(V1[1:5]:V3)</p> <p>Crée trois colonnes nommées TEMP, VX et VY contenant chacune des valeurs dans les lignes 1 à 5. Le contenu de la colonne TEMP correspond aux valeurs normalisées des lignes correspondantes de la colonne V1, le contenu de la colonne VX correspond aux valeurs normalisées des lignes correspondantes de la colonne V2 et le contenu de la colonne VY correspond aux valeurs normalisées des lignes correspondantes de la colonne V3. Les valeurs minimale et maximale utilisées à des fins de normalisation sont calculées sur les lignes 1 à 5 des colonnes V1-V3.</p>
<p>TEMP = NORM_MINMAX(V1:V3, V4:V6)</p> <p>Crée trois colonnes nommées TEMP, VX et VY. Chacune contient respectivement les valeurs normalisées du contenu des colonnes V1, V2 et V3. Les valeurs minimale et maximale utilisées pour la normalisation sont calculées sur les colonnes V4, V5 et V6.</p>

<p>TEMP = NORM_MINMAX(V1:V3, COL)</p> <p>Crée trois colonnes nommées TEMP, VX et VY. Chacune contient respectivement les valeurs normalisées du contenu des colonnes V1, V2 et V3. Les valeurs minimale et maximale utilisée pour la normalisation sont calculées indépendamment pour chaque colonne (une valeur min/max est calculée pour la colonne V1, une valeur min/max distincte est calculée pour la colonne V2, etc.).</p>
<p>TEMP = NORM_MINMAX(V1[10:50]:V3, COL)</p> <p>Crée trois colonnes nommées TEMP, VX et VY contenant chacune des valeurs dans les 41 premières lignes. Le contenu de la colonne TEMP correspond aux valeurs normalisées des lignes 10 à 50 de la colonne V1, le contenu de la colonne VX correspond aux valeurs normalisées des lignes 10 à 50 de la colonne V2 et le contenu de la colonne VY correspond aux valeurs normalisées des lignes 10 à 50 de la colonne V3. Les valeurs minimale et maximale utilisées à des fins de normalisation sont calculées indépendamment sur les lignes 10 à 50 de chaque colonne.</p>
<p>TEMP = NORM_MINMAX(V1:V3, V4:V6, COL)</p> <p>Crée trois colonnes nommées TEMP, VX et VY. Chacune contient respectivement les valeurs normalisées du contenu des colonnes V1, V2 et V3. Les valeurs minimale et maximale utilisées pour la normalisation sont calculées indépendamment pour chaque colonne, à l'aide des colonnes V4-V6 (une valeur min/max est calculée sur la colonne V4 pour normaliser la colonne V1, une valeur min/max distincte est calculée sur la colonne V5 pour normaliser la colonne V2, etc.).</p>
<p>TEMP = NORM_MINMAX(V1:V3, ROW)</p> <p>Crée trois colonnes nommées TEMP, VX et VY. Chacune contient respectivement les valeurs normalisées du contenu des colonnes V1, V2 et V3. Les valeurs minimale et maximale utilisées pour la normalisation sont calculées de manière indépendante sur chaque ligne des colonnes V1, V2 et V3.</p>
<p>TEMP = NORM_MINMAX(V1[10:20]:V3, ROW)</p> <p>Crée trois colonnes nommées TEMP, VX et VY contenant chacune des valeurs dans les 11 premières lignes. Le contenu de la colonne TEMP correspond aux valeurs normalisées des lignes 10 à 20 de la colonne V1, le contenu de la colonne VX correspond aux valeurs normalisées des lignes 10 à 20 de la colonne V2 et le contenu de la colonne VY correspond aux valeurs normalisées des lignes 10 à 20 de la colonne V3. Les valeurs minimale et maximale utilisées à des fins de normalisation sont calculées sur chacune des lignes 10 à 20 des colonnes V1-V3.</p>
<p>TEMP = NORM_MINMAX(V1:V3, V8:V10, ROW)</p> <p>Crée trois colonnes nommées TEMP, VX et VY. Chacune contient respectivement les valeurs normalisées du contenu des colonnes V1, V2 et V3. Les valeurs minimale et maximale utilisées pour la normalisation sont calculées indépendamment pour chaque ligne sur les colonnes V8-V10.</p>

### Fonctions associées

Fonction	Description
NORM_SIGMOID	Calcule la normalisation sigmoïde d'une plage de données
NORM_ZSCORE	Calcule la normalisation z-score d'une plage de données

---

## NORM\_SIGMOID



### Syntaxe

```
NORM_SIGMOID(data [, keyword]) NORM_SIGMOID(data, mean, std [, keyword])  
NORM_SIGMOID(data, base_data [, keyword])
```

### Paramètres

data

Valeurs à normaliser. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de data, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

mean, std

Ces deux paramètres fournissent la moyenne et l'écart type à utiliser pour la normalisation. Il doit s'agir de constantes, excepté avec le mot clé ROW, où il peut s'agir de constantes ou de colonnes.

base\_data

Ce paramètre indique une plage de données à utiliser pour le calcul de la moyenne et de l'écart type à utiliser pour la normalisation de data.

keyword

Ce mot clé facultatif détermine la manière dont le calcul est effectué sur la plage de données d'entrée. Sélectionnez l'une des valeurs suivantes :

ALL - Effectue le calcul sur toutes les cibles de data (valeur par défaut)

COL - Effectue le calcul séparément pour chaque colonne de data

ROW - Effectue le calcul séparément pour chaque ligne de data

Pour plus de détails sur l'utilisation des mots clés dans IBM Campaign, voir «Spécifications de format», à la page 9.

Pour plus de détails sur l'utilisation des mots clés dans IBM PredictiveInsight, voir «Spécifications de format», à la page 27.

### Description

NORM\_SIGMOID calcule les valeurs normalisées de la plage de données spécifiée. Une normalisation sigmoïde redistribue les données le long d'une courbe sigmoïde et renvoie des valeurs comprises entre -1.0 et +1.0, inclus. En fait, toutes les données d'un écart type de la moyenne sont distribuées de manière linéaire dans la plage intermédiaire de la sigmoïde. Les observations aberrantes sont représentées le long de la queue de la sigmoïde. Cela vous permet de conserver les points de données très aberrants sans sacrifier les possibilités de discrimination parmi les points proches de la moyenne.

La normalisation sigmoïde est effectuée comme suit :

$$VX[y] = \frac{1 - e^{-\alpha}}{1 + e^{-\alpha}}$$

où

$$\alpha = \frac{VX[y] - \text{mean}}{\text{std}}$$

et *mean* et *std* sont déterminés de la manière suivante :

- Si les valeurs *mean* et *std* sont fournies, sont utilisées respectivement pour la moyenne et l'écart type. Si ces paramètres sont fournis avec le mot clé *ROW*, les valeurs *mean* et *std* peuvent correspondre à des colonnes et spécifier la moyenne et l'écart type de chaque ligne de data. Si *min* et *max* sont des colonnes, ces colonnes doivent être de même longueur que *data* ou être scalaires (dans ce cas, elles contiennent une seule valeur qui est utilisée comme constante appliquée à toutes les valeurs de la colonne correspondante de data).
- Si *base\_data* est fourni, la moyenne et l'écart type de cette plage de données sont calculés et utilisés pour normaliser *data*. Les colonnes de *base\_data* doivent contenir plusieurs valeurs de cible.
- Si aucune des options mutuellement exclusives ci-dessus n'est fournie, la moyenne et l'écart standard sont automatiquement calculés à partir de *data*.

*NORM\_SIGMOID* renvoie toujours une plage de données avec les mêmes dimensions que la plage de données d'entrée. Le mot clé *ALL* spécifie le calcul de la moyenne et de l'écart type sur l'intégralité de la plage de données d'entrée. Le mot clé *COL* spécifie le calcul d'une moyenne et d'un écart type pour chaque colonne d'entrée et l'utilisation de ces valeurs pour normaliser cette colonne. Le mot clé *ROW* spécifie le calcul d'une moyenne et d'un écart type pour chaque ligne de la plage de données spécifiée et l'utilisation de ces valeurs pour normaliser cette ligne.

**Remarque :** Si l'écart type est égal à zéro, des zéros sont renvoyés.

**Remarque :** Pour normaliser les données à l'aide de la même plage *base\_data* (par exemple, dans les fonctions utilisateur encapsulées), transformez *mean* et *std* en constantes (pour cela, vous pouvez utiliser la fonction macro *CONSTANT*).

## Exemples

```
TEMP = NORM_SIGMOID(COLUMN(3, 4, 5))
```

Crée une colonne nommée *TEMP* contenant les valeurs -0.55, 0 et 0.55. (La moyenne et l'écart type [4 et 0.816] sont calculés automatiquement à partir de la plage de données.)

```
TEMP = NORM_SIGMOID(COLUMN(3, 4, 5), 3.5, 1.2)
```

Crée une colonne nommée *TEMP* contenant les valeurs -0.21, 0.21 et 0.55. (Cette fois la moyenne et l'écart type [3.5 et 1.2] sont fournis comme arguments.)

<p>TEMP = NORM_SIGMOID(V1) ou TEMP = NORM_SIGMOID(V1,ALL)</p> <p>Crée une colonne nommée TEMP contenant les valeurs normalisées du contenu de la colonne V1. La moyenne et l'écart type utilisés pour la normalisation sont calculés sur la colonne V1.</p>
<p>TEMP = NORM_SIGMOID(V1:V3)</p> <p>Crée trois colonnes nommées TEMP, VX et VY. Chacune contient respectivement les valeurs normalisées du contenu des colonnes V1, V2 et V3. La moyenne et l'écart type utilisés pour la normalisation sont calculés sur les colonnes V1, V2 et V3.</p>
<p>TEMP = NORM_SIGMOID(V1[10:50]:V3)</p> <p>Crée trois colonnes nommées TEMP, VX et VY contenant chacune des valeurs dans les lignes 1 à 41. Le contenu de la colonne TEMP correspond aux valeurs normalisées des lignes 10 à 50 de la colonne V1, le contenu de la colonne VX correspond aux valeurs normalisées des lignes 10 à 50 de la colonne V2 et le contenu de la colonne VY correspond aux valeurs normalisées des lignes 10 à 50 de la colonne V3. La moyenne et l'écart type utilisés à des fins de normalisation sont calculés sur les lignes 10 à 50 des colonnes V1-V3.</p>
<p>TEMP = NORM_SIGMOID(V1:V3, V4)</p> <p>Crée trois colonnes nommées TEMP, VX et VY. Chacune contient respectivement les valeurs normalisées du contenu des colonnes V1, V2 et V3. La moyenne et l'écart type utilisés pour la normalisation sont calculés sur la colonne V4.</p>
<p>TEMP = NORM_SIGMOID(V1:V3, V4:V8)</p> <p>Crée trois colonnes nommées TEMP, VX et VY. Chacune contient respectivement les valeurs normalisées du contenu des colonnes V1, V2 et V3. La moyenne et l'écart type utilisés pour la normalisation sont calculés sur les colonnes V4-V8.</p>
<p>TEMP = NORM_SIGMOID(V1:V3, COL)</p> <p>Crée trois colonnes nommées TEMP, VX et VY. Chacune contient respectivement les valeurs normalisées du contenu des colonnes V1, V2 et V3. La moyenne et l'écart type utilisés pour la normalisation sont calculés indépendamment sur chaque colonne (une moyenne et un écart type sont calculés pour la colonne V1, une moyenne et un écart type distincts sont calculés pour la colonne V2, etc.).</p>
<p>TEMP = NORM_SIGMOID(V1[10:50]:V3, COL)</p> <p>Crée trois colonnes nommées TEMP, VX et VY contenant chacune des valeurs dans les lignes 1 à 41. Le contenu de la colonne TEMP correspond aux valeurs normalisées des lignes 10 à 50 de la colonne V1, le contenu de la colonne VX correspond aux valeurs normalisées des lignes 10 à 50 de la colonne V2 et le contenu de la colonne VY correspond aux valeurs normalisées des lignes 10 à 50 de la colonne V3. La moyenne et l'écart type utilisés à des fins de normalisation sont calculés sur les lignes 10 à 50 des colonnes V1-V3. La moyenne et l'écart type utilisés à des fins de normalisation sont calculés indépendamment pour chaque colonne.</p>
<p>TEMP = NORM_SIGMOID(V1:V3, V4:V6, COL)</p> <p>Crée trois colonnes nommées TEMP, VX et VY. Chacune contient respectivement les valeurs normalisées du contenu des colonnes V1, V2 et V3. La moyenne et l'écart type utilisés pour la normalisation sont calculés indépendamment sur chaque colonne à l'aide des colonnes V4-V6 (une moyenne et un écart type sont calculés pour la colonne V4 pour normaliser la colonne V1, une moyenne et un écart type distincts sont calculés pour la colonne V5 pour normaliser la colonne V2, etc.).</p>



TEMP = NORM\_SIGMOID(V1:V3, ROW)

Crée trois colonnes nommées TEMP, VX et VY. Chacune contient respectivement les valeurs normalisées du contenu des colonnes V1, V2 et V3. La moyenne et l'écart type utilisés pour la normalisation sont calculés indépendamment sur chaque ligne pour les colonnes V1, V2 et V3.

TEMP = NORM\_SIGMOID(V1[10:50]:V3, ROW)

Crée trois colonnes nommées TEMP, VX et VY contenant chacune des valeurs dans les lignes 1 à 41. Le contenu de la colonne TEMP correspond aux valeurs normalisées des lignes 10 à 50 de la colonne V1, le contenu de la colonne VX correspond aux valeurs normalisées des lignes 10 à 50 de la colonne V2 et le contenu de la colonne VY correspond aux valeurs normalisées des lignes 10 à 50 de la colonne V3. La moyenne et l'écart type utilisés à des fins de normalisation sont calculés sur les lignes 10 à 50 des colonnes V1-V3. La moyenne et l'écart type utilisés à des fins de normalisation sont calculés sur chaque ligne des colonnes V1-V3.

TEMP = NORM\_SIGMOID(V1:V3, V4:V10, ROW)

Crée trois colonnes nommées TEMP, VX et VY. Chacune contient respectivement les valeurs normalisées du contenu des colonnes V1, V2 et V3. La moyenne et l'écart type utilisés pour la normalisation sont calculés indépendamment pour chaque ligne sur les colonnes V4-V10.

## Fonctions associées

Fonction	Description
NORM_MINMAX	Calcule la normalisation min/max d'une plage de données
NORM_ZSCORE	Calcule la normalisation z-score d'une plage de données

---

## NORM\_ZSCORE



### Syntaxe

NORM\_ZSCORE(data [, keyword]) NORM\_ZSCORE(data, mean, std [, keyword])  
NORM\_ZSCORE(data, base\_data [, keyword])

### Paramètres

data

Valeurs numériques à normaliser. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de data, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

mean, std

Ces deux paramètres fournissent la moyenne et l'écart type à utiliser pour la normalisation. Il doit s'agir de constantes, excepté avec le mot clé ROW, où il peut s'agir de constantes ou de colonnes.

base\_data

Ce paramètre indique une plage de données à utiliser pour le calcul de la moyenne et de l'écart type à utiliser pour la normalisation de data.

keyword

Ce mot clé facultatif détermine la manière dont le calcul est effectué sur la plage de données d'entrée. Sélectionnez l'une des valeurs suivantes :

ALL - Effectue le calcul sur toutes les cibles de data (valeur par défaut)

COL - Effectue le calcul séparément pour chaque colonne de data

ROW - Effectue le calcul séparément pour chaque ligne de data

Pour plus de détails sur l'utilisation des mots clés dans IBM Campaign, voir «Spécifications de format», à la page 9.

Pour plus de détails sur l'utilisation des mots clés dans IBM PredictiveInsight, voir «Spécifications de format», à la page 27.

## Description

NORM\_ZSCORE calcule les valeurs normalisées de la plage de données spécifiée. La normalisation z-score est effectuée comme suit :

$$VX[y] = \frac{VX[y] - \text{mean}}{\text{std}}$$

où *mean* et *std* sont déterminés de la manière suivante :

- Si les valeurs *mean* et *std* sont fournies, sont utilisées respectivement pour la moyenne et l'écart type. Si ces paramètres sont fournis avec le mot clé ROW, les valeurs *mean* et *std* peuvent correspondre à des colonnes et spécifier la moyenne et l'écart type de chaque ligne de data. Si *min* et *max* sont des colonnes, ces colonnes doivent être de même longueur que *data* ou être scalaires (dans ce cas, elles contiennent une seule valeur qui est utilisée comme constante appliquée à toutes les valeurs de la colonne correspondante de data).
- Si *base\_data* est fourni, la moyenne et l'écart type de cette plage de données sont calculés et utilisés pour normaliser *data*. Les colonnes de *base\_data* doivent contenir plusieurs valeurs de cible.
- Si aucune des options mutuellement exclusives ci-dessus n'est fournie, la moyenne et l'écart standard sont automatiquement calculés à partir de *data*.

NORM\_ZSCORE renvoie toujours une plage de données avec les mêmes dimensions que la plage de données d'entrée. Il calcule une moyenne et un écart type pour chaque colonne d'entrée et utilise ces valeurs pour normaliser cette colonne.

**Remarque :** Si l'écart type est égal à zéro, des zéros sont renvoyés.

## Exemples

```
TEMP = NORM_ZSCORE(COLUMN(3, 4, 5))
```

Crée une colonne nommée TEMP contenant les valeurs -1.22, 0 et 1.22. (La moyenne et l'écart type [4 et 0.816] sont calculés automatiquement à partir de la plage de données.)

```
TEMP = NORM_ZSCORE(COLUMN(3, 4, 5), 3.5, 1.2)
```

Crée une colonne nommée TEMP contenant les valeurs -0.42, 0.42 et 1.25. (Cette fois la moyenne et l'écart type [3.5 et 1.2] sont fournis comme arguments.)

```
TEMP = NORM_ZSCORE(V1)
```

Crée une colonne nommée TEMP contenant les valeurs normalisées du contenu de la colonne V1. La moyenne et l'écart type utilisés pour la normalisation sont calculés sur la colonne V1.

```
TEMP = NORM_ZSCORE(V1:V3)
```

Crée trois colonnes nommées TEMP, VX et VY. Chacune contient respectivement les valeurs normalisées du contenu des colonnes V1, V2 et V3. La moyenne et l'écart type utilisés pour la normalisation sont calculés indépendamment sur chaque colonne (une moyenne et un écart type sont calculés pour la colonne V1, une moyenne et un écart type distincts sont calculés pour la colonne V2, etc.).

```
TEMP = NORM_ZSCORE(V1[10:50]:V3)
```

Crée trois colonnes nommées TEMP, VX et VY contenant chacune des valeurs dans les lignes 1 à 41. Le contenu de la colonne TEMP correspond aux valeurs normalisées des lignes 10 à 50 de la colonne V1, le contenu de la colonne VX correspond aux valeurs normalisées des lignes 10 à 50 de la colonne V2 et le contenu de la colonne VY correspond aux valeurs normalisées des lignes 10 à 50 de la colonne V3. La moyenne et l'écart type utilisés à des fins de normalisation sont calculés sur les lignes 10 à 50 des colonnes V1-V3. La moyenne et l'écart type utilisés à des fins de normalisation sont calculés indépendamment pour chaque colonne.

```
TEMP = NORM_ZSCORE(V1:V3, V4:V6)
```

Crée trois colonnes nommées TEMP, VX et VY. Chacune contient respectivement les valeurs normalisées du contenu des colonnes V1, V2 et V3. La moyenne et l'écart type utilisés pour la normalisation sont calculés indépendamment sur chaque colonne à l'aide des colonnes V4-V6 (une moyenne et un écart type sont calculés pour la colonne V4 pour normaliser la colonne V1, une moyenne et un écart type distincts sont calculés pour la colonne V5 pour normaliser la colonne V2, etc.).

```
TEMP = NORM_ZSCORE(V1:V3, COL)
```

Crée trois colonnes nommées TEMP, VX et VY. Chacune contient respectivement les valeurs normalisées du contenu des colonnes V1, V2 et V3. La moyenne et l'écart type utilisés pour la normalisation sont calculés indépendamment sur chaque colonne (une moyenne et un écart type sont calculés pour la colonne V1, une moyenne et un écart type distincts sont calculés pour la colonne V2, etc.).

```
TEMP = NORM_ZSCORE(V1[10:50]:V3, COL)
```

Crée trois colonnes nommées TEMP, VX et VY contenant chacune des valeurs dans les lignes 1 à 41. Le contenu de la colonne TEMP correspond aux valeurs normalisées des lignes 10 à 50 de la colonne V1, le contenu de la colonne VX correspond aux valeurs normalisées des lignes 10 à 50 de la colonne V2 et le contenu de la colonne VY correspond aux valeurs normalisées des lignes 10 à 50 de la colonne V3. La moyenne et l'écart type utilisés à des fins de normalisation sont calculés sur les lignes 10 à 50 des colonnes V1-V3. La moyenne et l'écart type utilisés à des fins de normalisation sont calculés indépendamment pour chaque colonne.

<p>TEMP = NORM_ZSCORE(V1[10:50]:V3, COL)</p> <p>Crée trois colonnes nommées TEMP, VX et VY. Chacune contient respectivement les valeurs normalisées du contenu des colonnes V1, V2 et V3. La moyenne et l'écart type utilisés pour la normalisation sont calculés indépendamment sur chaque colonne à l'aide des colonnes V4-V6 (une moyenne et un écart type sont calculés pour la colonne V4 pour normaliser la colonne V1, une moyenne et un écart type distincts sont calculés pour la colonne V5 pour normaliser la colonne V2, etc.).</p>
<p>TEMP = NORM_ZSCORE (V1:V3, ROW)</p> <p>Crée trois colonnes nommées TEMP, VX et VY. Chacune contient respectivement les valeurs normalisées du contenu des colonnes V1, V2 et V3. La moyenne et l'écart type utilisés pour la normalisation sont calculés indépendamment sur chaque ligne pour les colonnes V1, V2 et V3.</p>
<p>TEMP = NORM_ZSCORE(V1[10:50]:V3, ROW)</p> <p>Crée trois colonnes nommées TEMP, VX et VY contenant chacune des valeurs dans les lignes 1 à 41. Le contenu de la colonne TEMP correspond aux valeurs normalisées des lignes 10 à 50 de la colonne V1, le contenu de la colonne VX correspond aux valeurs normalisées des lignes 10 à 50 de la colonne V2 et le contenu de la colonne VY correspond aux valeurs normalisées des lignes 10 à 50 de la colonne V3. La moyenne et l'écart type utilisés à des fins de normalisation sont calculés sur les lignes 10 à 50 des colonnes V1-V3. La moyenne et l'écart type utilisés à des fins de normalisation sont calculés sur chaque ligne des colonnes V1-V3.</p>
<p>TEMP = NORM_ZSCORE(V1:V3, V4:V10, ROW)</p> <p>Crée trois colonnes nommées TEMP, VX et VY. Chacune contient respectivement les valeurs normalisées du contenu des colonnes V1, V2 et V3. La moyenne et l'écart type utilisés pour la normalisation sont calculés indépendamment pour chaque ligne sur les colonnes V4-V10.</p>

## Fonctions associées

Fonction	Description
NORM_MINMAX	Calcule la normalisation min/max d'une plage de données
NORM_SIGMOID	Calcule la normalisation sigmoïde d'une plage de données

## NOT

### Syntaxe

NOT(data) ! data

### Paramètres

data

Valeurs numériques dont le NOT logique doit être calculé. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de data, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

## Description

NOT renvoie le NOT logique des valeurs de la plage de données spécifiée. Il renvoie une nouvelle colonne par colonne d'entrée, chacune contenant le NOT logique des valeurs de la colonne d'entrée correspondante. Cette fonction renvoie la valeur zéro pour les valeurs non nulles et la valeur un pour les valeurs nulles.

**Remarque :** L'opérateur NOT peut être abrégé en point d'exclamation (!). Utilisez le point d'exclamation avant la valeur (par exemple, pour spécifier NOT(V1), vous pouvez simplement saisir !V1).

Cette macro est disponible dans IBM Interact.

## Exemples

TEMP = NOT(3.2) ou TEMP = !1
Crée une colonne nommée TEMP contenant la valeur zéro.
TEMP = !0 ou TEMP = !(2+2=3)
Crée une colonne nommée TEMP contenant la valeur un.
TEMP = !V1
Crée une colonne nommée TEMP, dans laquelle chaque valeur correspond au NOT logique des valeurs de la colonne V1.
TEMP = !V1:V3
Crée trois colonnes nommées TEMP, VX et VY. Les valeurs de la colonne TEMP correspondent aux valeurs NOT logiques des valeurs de la colonne V1, les valeurs de la colonne VX correspondent aux valeurs NOT logiques des valeurs de la colonne V2 et les valeurs de la colonne VY correspondent aux valeurs NOT logiques des valeurs de la colonne V3.
TEMP = !V1[10:20]
Crée une colonne nommée TEMP, dans laquelle les 11 premières cibles contiennent les valeurs NOT logiques des valeurs des lignes 10 à 20 de la colonne V1. Les autres cibles de TEMP sont vides.
TEMP = !V1[1:5]:V2
Crée deux colonnes nommées TEMP et VX, contenant des valeurs dans les lignes 1 à 5 (les autres cibles sont vides). Les valeurs de la colonne TEMP correspondent aux valeurs NOT logiques des valeurs des lignes correspondantes de la colonne V1 et les valeurs de la colonne VX correspondent aux valeurs NOT logiques des valeurs des lignes correspondantes de la colonne V2.

## Fonctions associées

Fonction	Description
AND	Calcule le et logique (AND) entre deux plages de données spécifiées
INVERSE	Calcule la valeur négative du contenu de la plage de données spécifiée
OR	Calcule le OR logique entre deux plages de données spécifiées
SIGN	Détermine le signe (positif ou négatif) des valeurs dans la plage de données spécifiée

---

## NPV



### Syntaxe

NPV(data, rate [, keyword])

### Paramètres

data

Valeurs numériques représentant le flux de trésorerie net attendu utilisé pour le calcul de la valeur nette actualisée. Il peut s'agir d'une ligne, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs.

rate

Valeur numérique représentant le taux d'actualisation pour une période.

keyword

Ce mot clé facultatif détermine la manière dont le calcul est effectué sur la plage de données d'entrée. Si aucun mot clé n'est fourni, ROW est utilisé comme valeur par défaut. Sélectionnez l'une des valeurs suivantes :

COL - Effectue le calcul séparément pour chaque colonne de data

ROW - Effectue le calcul séparément pour chaque ligne de data

### Description

NPV calcule la valeur nette actualisée d'un investissement en fonction d'une série de flux de trésorerie périodiques et d'un taux d'actualisation. La valeur nette actualisée d'un investissement correspond à la valeur actuelle d'une série de versements futurs (valeurs négatives) et de revenus (valeurs positives). NPV est calculé à l'aide de la formule suivante :

$$NPV = \sum_{t=1}^n \frac{\text{data}_t}{(1 + \text{rate})^t}$$

où  $n$  représente le nombre de flux de trésorerie (valeurs de données).

L'ordre des valeurs de données est utilisé pour interpréter l'ordre des flux de trésorerie. L'investissement de la valeur nette actualisée commence une période avant la date de la première valeur de flux de trésorerie et se termine au dernier flux de trésorerie de la liste.

**Remarque :** Si votre premier flux de trésorerie a lieu au début de la première période, la première valeur doit être ajoutée au résultat NPV, non inclus dans les valeurs de données.

Le nombre de colonnes renvoyé par NPV dépend de keyword.

- Si keyword a la valeur COL, NPV renvoie une nouvelle colonne pour chaque colonne d'entrée. Chaque nouvelle colonne contient une valeur (valeur nette actualisée de toutes les cibles de la colonne d'entrée correspondante).
- Si keyword a la valeur ROW, NPV renvoie une nouvelle colonne contenant la valeur nette actualisée de chaque ligne de data.

**Remarque :** Les cibles vides sont exclues du calcul de NPV.

## Exemples

TEMP = NPV(V1:V3,.10) ou TEMP = NPV(V1:V3,.10, ROW)

Crée une colonne nommée TEMP contenant une valeur unique qui correspond à la valeur nette actualisée des valeurs des colonnes V1, V2 et V3. Le taux d'actualisation utilisé est de 10 %.

TEMP = NPV(V1, .10, COL)

Crée une colonne nommée TEMP contenant une valeur unique qui correspond à la valeur nette actualisée du contenu de la colonne V1, pour un taux d'actualisation de 10 %.

TEMP = NPV(V1:V3, .10) - 1000

Crée une colonne nommée TEMP contenant une valeur unique qui correspond à la valeur nette actualisée du contenu des colonnes V1, V2 et V3, pour un versement initial de 1000. Le taux d'actualisation est de 10 %.

TEMP = NPV(V1[10:20], .10L, COL)

Crée une colonne nommée TEMP contenant une valeur unique qui correspond à la valeur nette actualisée des cibles des lignes 10 à 20 de la colonne V1. Le taux d'actualisation est de 10 %.

TEMP = NPV(V1[1:5]:V4, .10)

Crée une colonne nommée TEMP contenant une valeur unique qui correspond à la valeur nette actualisée des cibles des lignes 1 à 5 des colonnes V1 à V4.

---

## NUMBER

### Syntaxe

NUMBER(data [, conversion\_keyword])

### Paramètres

data

Données de texte ASCII à convertir en valeurs numériques. Il peut s'agir de texte ASCII entre guillemets, d'une colonne de texte, d'une plage de cibles contenant du texte ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de data, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

conversion\_keyword

Ce mot clé facultatif indique comment interpréter les formats de texte des dates et des heures. Sélectionnez l'un des mots clés du tableau ci-après.

**Remarque :** Si ce paramètre n'est pas spécifié, la valeur par défaut est 1.

Mot clé de conversion	Format	Description
Classe 0	#####	Convertit les 5 premiers caractères de chaque chaîne de texte en un numéro unique
1	\$ (valeur par défaut)	Convertit les valeurs en dollars en valeurs numériques (par exemple, "\$123.45" en 123.45)
2	%	Convertit un pourcentage en valeur numérique (par exemple, "50%" en 0,5)
3	mm/dd/yy hh:mm	Convertit une date et une heure en nombre de jours écoulés depuis le 1 janvier 0000 (1900 est automatiquement ajouté à l'année yy)
4	dd-mmm-yy	Convertit une date en nombre de jours écoulés depuis le 1 janvier 0000 (1900 est automatiquement ajouté à l'année yy)
5	mm/dd/yy	Convertit une date en nombre de jours écoulés depuis le 1 janvier 0000 (1900 est automatiquement ajouté à l'année yy)
6	mmm-yy	Convertit une date en nombre de jours écoulés entre le premier jour du mois spécifié et le 1er janvier 0000 (1900 est automatiquement ajouté à l'année yy)
7	dd-mmm	Convertit une date en nombre de jours écoulés depuis le début de l'année (par exemple, "01-FEB" en 32)
8	mmm	Convertit une abréviation de mois en trois lettres en une valeur comprise entre 1 et 12 (par exemple, "DEC" en 12)
9	{January   February   March ... }	Convertit un nom de mois en toutes lettres en valeur comprise entre 1 et 12 (par exemple, "March" en 3)



10	{Sun   Mon   Tue ... }	Convertit une abréviation de jour de la semaine sur trois caractères en valeur comprise entre 0 et 6, dimanche correspondant au début de la semaine (par exemple, "Sun" en 0)
11	{Sunday   Monday   Tuesday ... }	Convertit un nom de jour de la semaine en toutes lettres en valeur comprise entre 0 et 6, dimanche correspondant au début de la semaine (par exemple, "Monday" en 1)
12	hh:mm:ss {AM   PM}	Convertit l'heure en nombre de secondes écoulées depuis 00:00:00 AM (minuit) (par exemple, "01:00:00 AM" en 3600)
13	hh:mm:ss	Convertit l'heure en nombre de secondes écoulées depuis 00:00:00 AM (minuit) (par exemple, "01:00:00" en 3600)
14	hh:mm {AM   PM}	Convertit l'heure en nombre de minutes écoulées depuis 00:00:00 AM (minuit) (par exemple, "01:00 AM" en 60)
15	hh:mm	Convertit l'heure en nombre de minutes écoulées depuis 00:00:00 AM (minuit) (par exemple, "01:00" en 60)
16	mm:ss	Convertit l'heure en nombre de secondes écoulées depuis 00:00:00 AM (minuit) (par exemple, "30:00" en 1800)
17	ddmm	Convertit une date en nombre de jours écoulés depuis le début de l'année (par exemple, "3101" en 31)
18	ddmmm	Convertit une date en nombre de jours écoulés depuis le début de l'année (par exemple, "31JAN" en 31)
19	ddmmyy	Convertit une date en nombre de jours écoulés depuis le 1 janvier 0000 (1900 est automatiquement ajouté à l'année si yy est inférieur ou égal à 20 ; sinon, 2000 est ajouté)
20	ddmmyyyy	Convertit une date en nombre de jours écoulés depuis le 1er janvier 0000 (par exemple, "31JAN0000" en 31)

21	ddmmyy	Convertit une date en nombre de jours écoulés depuis le 1 janvier 0000 (1900 est automatiquement ajouté à l'année si yy est inférieur ou égal à 20 ; sinon, 2000 est ajouté)
22	ddmmyyyy	Convertit une date en nombre de jours écoulés depuis le 1er janvier 0000 (par exemple, "31010000" en 31)
23	mmdd	Convertit une date en nombre de jours écoulés depuis le début de l'année (par exemple, "0131" en 31)
24	mmddy	Convertit une date en nombre de jours écoulés depuis le 1 janvier 0000 (1900 est automatiquement ajouté à l'année si yy est inférieur ou égal à 20 ; sinon, 2000 est ajouté)
25	mmddyyyy	Convertit une date en nombre de jours écoulés depuis le 1er janvier 0000 (par exemple, "01010001" en 366)
26	mmm	Convertit une abréviation de mois en trois lettres en une valeur comprise entre 1 et 12 (par exemple, "MAR" en 3) [Remarque : identique au mot clé de conversion 8]
27	mmdd	Convertit une date en nombre de jours écoulés depuis le début de l'année (par exemple, "JAN31" en 31)
28	mmddy	Convertit une date en nombre de jours écoulés depuis le 1 janvier 0000 (1900 est automatiquement ajouté à l'année si yy est inférieur ou égal à 20 ; sinon, 2000 est ajouté)
29	mmddyyyy	Convertit une date en nombre de jours écoulés depuis le 1er janvier 0000 (par exemple, "FEB010001" en 32)

30	mmyy	Convertit une date en nombre de jours écoulés entre le 1er du mois spécifié et le 1er janvier 0000 (1900 est automatiquement ajouté à l'année si yy est inférieur ou égal à 20 ; sinon, 2000 est ajouté)
31	mmyyyy	Convertit une date en nombre de jours écoulés entre le 1er du mois spécifié et le 1er janvier 0000 (par exemple, "FEB0001" en 32)
32	mmyy	Convertit une date en nombre de jours écoulés entre le 1er du mois spécifié et le 1er janvier 0000 (1900 est automatiquement ajouté à l'année si yy est inférieur ou égal à 20 ; sinon, 2000 est ajouté)
33	mmyyyy	Convertit une date en nombre de jours écoulés entre le 1er du mois spécifié et le 1er janvier 0000 (par exemple, "020001" en 32)
34	yymm	Convertit une date en nombre de jours écoulés entre le 1er du mois spécifié et le 1er janvier 0000 (1900 est automatiquement ajouté à l'année si yy est inférieur ou égal à 20 ; sinon, 2000 est ajouté)
35	yymmdd	Convertit une date en nombre de jours écoulés depuis le 1 janvier 0000 (1900 est automatiquement ajouté à l'année si yy est inférieur ou égal à 20 ; sinon, 2000 est ajouté)
36	yymmm	Convertit une date en nombre de jours écoulés entre le 1er du mois spécifié et le 1er janvier 0000 (1900 est automatiquement ajouté à l'année si yy est inférieur ou égal à 20 ; sinon, 2000 est ajouté)
37	yymmmdd	Convertit une date en nombre de jours écoulés depuis le 1 janvier 0000 (1900 est automatiquement ajouté à l'année si yy est inférieur ou égal à 20 ; sinon, 2000 est ajouté)

38	yyyy	Convertit le nombre d'années écoulées depuis l'an 0000 (par exemple, "1998" en 1998)
39	yyyymm	Convertit une date en nombre de jours écoulés entre le 1er du mois spécifié et le 1er janvier 0000 (par exemple, "000102" en 32)
40	yyyymmdd	Convertit une date en nombre de jours écoulés depuis le 1er janvier 0000 (par exemple, "00010201" en 32)
41	yyyymm	Convertit une date en nombre de jours écoulés entre le 1er du mois spécifié et le 1er janvier 0000 (par exemple, "000102" en 32)
42	yyyymmdd	Convertit une date en nombre de jours écoulés depuis le 1er janvier 0000 (par exemple, "0001FEB01" en 32)
43	<day>* <month>	Convertit toute date délimitée avec le jour suivi du mois en nombre de jours écoulés depuis le début de l'année (par exemple, "15-JAN" en 15)
44	<day>* <month>* <year>	Convertit toute date délimitée avec le jour précédant le mois et suivi de l'année en nombre de jours écoulés depuis le 1er janvier 0000 (par exemple, "1/1/0001" en 366)
45	<month>* <day>	Convertit toute date délimitée avec le mois suivi du jour en nombre de jours écoulés depuis le début de l'année (par exemple, "JAN 31" en 31)
46	<month>* <day>* <year>	Convertit toute date délimitée avec le mois suivi du jour, suivi de l'année en nombre de jours écoulés depuis le 1er janvier 0000 (par exemple, "JAN 1, 0001" en 366)
47	<month>* <year>	Convertit toute date délimitée avec le mois suivi de l'année en nombre de jours écoulés entre le premier jour du mois spécifié et le 1er janvier 0000

48	<year>* <month>	Convertit toute date délimitée avec l'année suivi du mois en nombre de jours écoulés entre le premier jour du mois spécifié et le 1er janvier 0000
49	<year>* <month>* <day>	Convertit toute date délimitée avec le mois suivi du jour, suivi de l'année en nombre de jours écoulés depuis le 1er janvier 0000 (par exemple, "0001/01/01" en 366)
50	yy	Convertit l'année en nombre d'années écoulées depuis l'an 0000 (par exemple, "97" en 97)
51	mm	Convertit le mois en valeur comprise entre 1 et 12 (par exemple, "SEP" en 9)
52	dd	Convertit le jour en valeur comprise entre 1 et 31 (par exemple, "28" en 28)
53	{January   February   March ... }	Convertit un nom de mois en toutes lettres en valeur comprise entre 1 et 12 (par exemple, "March" en 3) [Remarque : identique au mot clé de conversion 9]
54	{Sunday   Monday   Tuesday ... }	Convertit un nom de jour de la semaine en toutes lettres en valeur comprise entre 1 et 7, dimanche correspondant au début de la semaine (par exemple, "Sunday" en 1)
55	{Sun   Mon   Tue ... }	Convertit une abréviation de jour de la semaine sur trois caractères en valeur comprise entre 1 et 7, dimanche correspondant au début de la semaine (par exemple, "Sun" en 1)

## Description

NUMBER convertit les valeurs de texte dans la plage de données spécifiée en valeurs numériques, à l'aide du format spécifié pour convertir les dates et les heures. Si une chaîne de texte ne peut pas être analysée à l'aide du mot clé `conversion_keyword` spécifié, NUMBER génère une erreur. Le format 0 convertit les cinq premiers caractères de chaque chaîne de texte en nombre différent pour chaque chaîne de texte unique. C'est un moyen simple de changer une colonne de texte en classes uniques pour les sorties dans un discriminant.

Les formats délimités (mots clés de conversion 43 à 49) acceptent les caractères suivants comme délimiteurs :

- / (barre oblique)
- - (tiret)
- , (virgule)
- " " (espace)
- : (deux-points)

Les mois peuvent être représentés sous la forme mm ou mmm ; les jours peuvent être représentés sous la forme d ou dd ; les années peuvent être représentées sous la forme yy ou yyyy.

**Remarque :** Dans le cadre de la conformité à l'an 2000, toutes les années des dates peuvent être désignées sous la forme yyyy au lieu de yy. Pour la compatibilité amont, dans les mots clés de conversion 1 à 16, 1900 est automatiquement ajouté à yy (années sur deux chiffres). Pour les mots clés de conversion 17 à 55, 2000 est automatiquement ajouté à yy < threshold ; 1900 est automatiquement ajouté à yy ≥ threshold.

**Remarque :** La valeur threshold de l'an 2000 est définie dans l'onglet **Nettoyage des données** de la fenêtre **Paramètres avancés**(appelez-la en sélectionnant **Options > Paramètres > Paramètres avancés**).

**Remarque :** Si vous changez la valeur de seuil an 2000, vous devez mettre à jour toutes les fonctions macro à l'aide de la fonction macro NUMBER pour manipuler les valeurs de date avec des années sur 2 chiffres. Pour forcer une mise à jour d'une fonction macro, vous pouvez effectuer toute changement (par exemple, ajouter un espace et le supprimer) et cliquer sur l'icône en forme de coche pour accepter la changement.

**Remarque :** Lorsque vous utilisez le format 0, seuls les cinq premiers caractères de chaque chaîne de texte sont utilisés pour générer un nombre unique. Toutes les chaînes contenant ces cinq premiers caractères seront converties dans la même valeur numérique. La même chaîne de texte générera la même valeur numérique à chaque fois, même entre des feuilles de calcul différentes. Si nécessaire, utilisez des macros de chaîne pour manipuler les chaînes de sorte que les cinq premiers caractères définissent une classe de manière unique. Notez que les valeurs numériques résultantes peuvent être très petites. Utilisez la fenêtre **Formats d'affichage** pour augmenter le nombre de décimales à afficher ou changer le format en mode exponentiel ( 00E+00).

Cette macro est disponible dans IBM Interact.

## Exemples

TEMP = NUMBER("\$1.23") ou TEMP = NUMBER("123%", 2)
---

Crée une colonne nommée TEMP contenant le nombre 1.23.

TEMP = NUMBER(column("Jan", "Mar", "Dec", 8)
--

Crée une colonne nommée TEMP contenant les nombres 1, 3 et 12.

TEMP = NUMBER("1:52 PM", 14)
------------------------------

Crée une colonne nommée TEMP contenant le nombre 832.

TEMP = NUMBER("1/1/95", 5)
----------------------------

Crée une colonne nommée TEMP contenant le nombre 728660.

<p>TEMP = NUMBER(V1)</p> <p>Crée une colonne nommée TEMP contenant les valeurs numériques des chaînes de texte de la colonne V1. Les valeurs en dollars sont correctement converties en valeurs numériques. ??? est renvoyé pour les chaînes de texte qui ne peuvent pas être analysées à l'aide du format \$.</p>
<p>TEMP = NUMBER(V1:V3, 4)</p> <p>Crée trois colonnes nommées TEMP, VX et VY. La colonne TEMP contient les valeurs numériques des chaînes de texte de la colonne V1. La colonne VX contient les valeurs numériques des chaînes de texte de la colonne V2. La colonne VY contient les valeurs numériques des chaînes de texte de la colonne V3. Les dates au format dd-mmm-yy sont converties en nombre de jours de décalage à partir du 1er janvier 0000. ??? est renvoyé pour les chaînes de texte qui ne peuvent pas être analysées à l'aide du format \$.</p>
<p>TEMP = NUMBER(V1[10:20]:V2, 10)</p> <p>Crée deux colonnes nommées TEMP et VX. La colonne TEMP contient les valeurs numériques des chaînes de texte des lignes 10 à 20 de la colonne V1. La colonne VX contient les valeurs numériques des chaînes de texte des lignes 10 à 20 de la colonne V2. Toutes les représentations standard de jour de la semaine à trois caractères sont converties en nombres de 0 à 6 (0 = dimanche, 6 = samedi). En l'absence de correspondance pour un nom de jour de la semaine, ??? est renvoyé.</p>
<p>TEMP = NUMBER(V1, 0)</p> <p>Si la colonne V1 ne contient que des chaînes de texte à cinq chiffres, crée une colonne nommée TEMP contenant une valeur numérique différente par chaîne unique.</p>

## Fonctions associées

Fonction	Description
WEEKDAY	Convertit les chaînes de date textuelles ASCII en jour de la semaine

---

## OFFSET



### Syntaxe

OFFSET(data)

### Paramètres

data

Valeurs dont le décalage doit être calculé. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de data, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

### Description

OFFSET renvoie les valeurs du décalage de la plage de données spécifiée par rapport à la première valeur. Il renvoie une nouvelle colonne contenant les valeurs

de décalage, comprises entre un et la longueur de la colonne la plus longue de la plage de données.

## Exemples

TEMP = OFFSET(5) Crée une colonne nommée TEMP contenant la valeur un.
TEMP = OFFSET(V1) Crée une colonne nommée TEMP, dans laquelle chaque valeur correspond au décalage de la colonne V1, compris entre un et la longueur de la colonne V1.
TEMP = OFFSET(V1:V3) Crée une colonne nommée TEMP, dans laquelle chaque valeur correspond au décalage, compris entre un et la colonne la plus longue parmi V1, V2 et V3.

## Fonctions associées

Fonction	Description
COUNT	Compte le nombre de cibles contenant des valeurs dans la plage de données spécifiée
DELAY	Renvoie les valeurs des colonnes d'entrée retardées du nombre d'intervalles de temps spécifié

---

## OR

### Syntaxe

data1 OR data2 data1 || data2

### Paramètres

data1

Nombres dont le OR logique doit être calculé avec les valeurs de data2. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de data, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

data2

Nombres dont le OR logique doit être calculé avec les valeurs de data1. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Le nombre de colonnes dans data2 doit correspondre au nombre de colonnes dans data1, à moins que data2 ne soit une constante. Pour la définition du format de data, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.



## Description

OR calcule le OR logique entre les deux plages de données spécifiées. Il renvoie une nouvelle colonne pour chaque colonne d'entrée, chacune contenant la colonne correspondante de data1 dont le OR logique avec la colonne correspondante de data2 a été calculé (à savoir, le OR logique de la première colonne de data1 avec la première colonne de data, celui de la deuxième colonne avec la deuxième colonne et ainsi de suite).

Si data2 est une constante, le OR logique de chaque valeur de data1 avec cette valeur est calculé. Si data2 contient une ou plusieurs colonnes, les calculs sont effectués ligne par ligne entre une colonne de data1 et une colonne de data2. Le OR logique de la première ligne de data1 avec la première ligne de data2 est calculé, la seconde ligne avec la seconde ligne, etc. Ce calcul ligne par ligne génère un résultat pour chaque ligne jusqu'à la dernière valeur de la colonne la plus courte.

**Remarque :** L'utilisation d'une colonne contenant le même nombre  $x$  dans chaque ligne que data2 revient à utiliser la constante  $x$  pour data2.

**Remarque :** L'opérateur OR peut être abrégé en double barre verticale (||). Utilisez la double barre verticale pour séparer les deux arguments (par exemple, pour spécifier V1 OR 3, vous pouvez simplement entrer V1||3).

Cette macro est disponible dans IBM Interact.

## Exemples

<pre>TEMP = 1 OR 8 ou TEMP = 1    8</pre>
Crée une colonne nommée TEMP contenant la valeur un (tout nombre non nul prend la valeur un).
<pre>TEMP = V1    1</pre>
Crée une colonne nommée TEMP ne contenant que des uns (un OR entre toute valeur et le nombre un génère un).
<pre>TEMP = V1    V2</pre>
Crée une colonne nommée TEMP, dans laquelle chaque valeur est la valeur de ligne de la colonne V1 dont le OR logique est calculé avec la valeur de ligne correspondante de la colonne V2.
<pre>TEMP = V1:V3    V4:V6</pre>
Crée trois colonnes nommées TEMP, VX et VY. La colonne TEMP contient les valeurs de V1 dont le OR logique avec les valeurs de ligne correspondantes de la colonne V4 a été calculé. La colonne VX contient les valeurs résultant du OR logique des colonnes V2 et V5. La colonne VY contient les valeurs résultant du OR logique des colonnes V3 et V6.
<pre>TEMP = V1[10:20]    V2</pre>
Crée une colonne nommée TEMP, dans laquelle les 11 premières cibles contiennent le résultat du OR logique entre les valeurs des lignes 10 à 20 des colonnes V1 et V2. Les autres cibles de TEMP sont vides.

## Fonctions associées

Fonction	Description
AND	Calcule le et logique (AND) entre deux plages de données spécifiées
NOT	Calcule le non logique (NON) du contenu de la plage de données spécifiée

---

## PCA



### Syntaxe

PCA(data)

### Paramètres

data

Valeurs numériques dont les composants principaux doivent être calculés. Il peut s'agir d'une constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs.

### Description

PCA effectue l'analyse des composants principaux de la plage de données spécifiée. Il recherche les vecteurs propres orthogonaux à la plage de données spécifiée par data à l'aide d'une décomposition en valeurs simples. Il renvoie une nouvelle colonne pour chacune des  $n$  colonnes spécifiées en entrée, plus une colonne supplémentaire. Les  $n$  premières colonnes contiennent les vecteurs propres (chaque vecteur propre est lu comme une ligne des  $n$  colonnes). La dernière colonne renvoyée contient les amplitudes correspondantes des valeurs propres. Les vecteurs propres sont classés en fonction de leur valeur propre.

**Remarque :** Les valeurs manquantes (par exemple, les cibles vides et les cibles ???) sont comptées comme des zéros. Les colonnes les plus courtes de data sont complétées par des zéros jusqu'à la longueur de la colonne la plus longue.

Voici les détails du calcul de PCA :

- Chacune des  $k$  lignes de data est un vecteur à  $n$  dimensions  $v_i$  ( $n$  représente le nombre de colonnes dans data). Elles sont utilisées pour calculer la matrice de corrélation  $A$  comme suit :

$$A = \sum_{i=1}^k v_i v_i^T$$

- La matrice de corrélation  $A$   $n$  par  $n$  est décomposée en valeurs simples dans trois matrices :

$$A = U\Sigma U^T$$

Les lignes de  $U$  sont les vecteurs propres de  $A$  et  $\Sigma$  est une matrice diagonale dans laquelle chaque élément diagonal représente l'ampleur des valeurs propres de  $A$ .

La fonction macro PCA renvoie  $U$  dans les  $n$  premières colonnes et les éléments diagonaux de  $\Sigma$  dans la dernière colonne.

### Exemples

TEMP = PCA(5) Crée deux colonnes nommées TEMP et VX, contenant respectivement les valeurs -1 et 0.
TEMP = PCA(V1)  Crée deux colonnes nommées TEMP et VX. La colonne TEMP contient la valeur un et la colonne VX contient la valeur propre correspondante.
TEMP = PCA(V1:V3)  Crée quatre colonnes nommées TEMP, VX, VY et VZ. Les valeurs des trois colonnes contiennent un vecteur propre par ligne pour les données des colonnes V1-V3. La valeur de la colonne VZ contient les valeurs propres correspondantes.

### Fonctions associées

Fonction	Description
PCA_FEATURES	Extrait $n$ fonctions de la plage de données spécifiée

---

## PCA\_FEATURES



### Syntaxe

PCA\_FEATURES(num\_features, data [, PCA(base\_data)])

### Paramètres

num\_features

Nombre de fonctions à extraire de la plage de données spécifiée à l'aide de l'analyse des composants principaux (PCA). Cette valeur doit être un entier positif compris entre un et le nombre de colonnes de la plage de données spécifiée par data.

data

Valeurs numériques dont les fonctions doivent être extraites. Il peut s'agir d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de data, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

PCA(base\_data)

Si ce paramètre facultatif est fourni, l'analyse PCA est effectuée sur cette plage de données base\_data et les vecteurs propres résultants sont utilisés pour extraire les fonctions de la plage de données data. Pour la définition du format de base\_data (identique à data), voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM. Le nombre de colonnes de base\_data doit correspondre au nombre de colonnes de data.

## Description

PCA\_FEATURES extrait les num\_features premières fonctions de la plage de données spécifiée. Il renvoie num\_features colonnes à l'aide des vecteurs propres générés par l'analyse des composants principaux sur la plage de données base\_data, si elle est fournie. Si elle ne l'est pas, il utilise data pour générer les vecteurs propres. Dans ce cas, data est automatiquement normalisé à l'aide de la méthode "moyenne nulle, variance unité" avant l'analyse des composants principaux.

Les fonctions sont calculées comme suit :

- La plage de données data est automatiquement normalisée à l'aide de la méthode "moyenne nulle, variance unité". En d'autres termes,  $PCA\_FEATURES(num\_features, data)$

est équivalent à

$PCA\_FEATURES(num\_features, data, PCA(data, COL))$

Aucune normalisation de data n'est fournie automatiquement. Pour normaliser data à l'aide de NORM\_ZSCORE, vous pouvez spécifier :

$PCA\_FEATURES(num\_features, data, PCA(NORM\_ZSCORE(data, COL)))$

- L'analyse des composants principaux est effectuée sur la plage de données normalisée pour générer ses vecteurs propres (voir les informations détaillées de la fonction macro PCA). Cette opération est automatique pour data si base\_data n'est pas fourni. Elle est effectuée par l'appel explicite de la fonction macro PCA si base\_data est fourni.
- Chaque ligne ( $V_i$ ) de la plage de données (data) est transformée en nouveau système de coordonnées ( $U_i$ ) en fonction des num\_features ( $m$ ) premiers vecteurs propres qui composent  $E_m$  :

$$u_i = E_m V_i = \begin{bmatrix} E_{11} & \dots & E_{1n} \\ \dots & \dots & \dots \\ E_{m1} & \dots & E_{mn} \end{bmatrix} \begin{bmatrix} v_1 \\ \dots \\ v_n \end{bmatrix}$$

- Les  $k$  lignes des données transformées ( $u^1$  à  $u^k$ ) sont renvoyées ( $n$  colonnes).

Si la plage de données base\_data est fournie, elle doit contenir le même nombre de colonnes que la plage de données data ; sinon, une erreur est renvoyée.

**Remarque :** Le calcul de PCA sur une plage de données pouvant s'avérer intensif, il est bien plus efficace d'utiliser la fonction macro BUFFER sur le calcul PCA. Par exemple : PCA\_FEATURES(num\_features, range, BUFFER(PCA(base\_data)))

## Exemples

TEMP = PCA\_FEATURES(5, V1:V7) Crée cinq colonnes nommées TEMP, VW, VX, VY et VZ contenant les cinq premières fonctions de la plage de données V1:V7. La plage de données V1:V7 est utilisée comme base de la transformation.

TEMP = PCA\_FEATURES(3, V1:V4, PCA(V10:V13))

Crée trois colonnes nommées TEMP, VX et VY, contenant les trois premières fonctions de la plage de données V1:V4. La plage de données V10:V13 est utilisée comme base de la transformation.

TEMP = PCA\_FEATURES(3, V1:V4, BUFFER(PCA(V10:V13)))

Crée trois colonnes nommées TEMP, VX et VY, contenant les trois premières fonctions de la plage de données V1:V4. La plage de données V10:V13 est utilisée comme base de la transformation. Une fois que les composants principaux de la plage de données V10:V13 ont été calculés, ces valeurs sont stockées comme constantes. Si les valeurs de données des colonnes V10 à V13 sont changées, ils n'ont aucun impact sur cette définition de fonction.

## Fonctions associées

Fonction	Description
PCA	Calcule les vecteurs propres des composants principaux de la plage de données spécifiée

## POSITION

### Syntaxe

POSITION(colName, pattern [, start [, occurrence]])

## Paramètres

colName

Valeur d'une colonne (doit être de type string).

pattern

Modèle ou chaîne que vous recherchez.

start

Octet à partir duquel la recherche commence.

occurrence

Indiquez une valeur pour n, où vous recherchez la n<sup>ème</sup> occurrence du modèle à renvoyer.

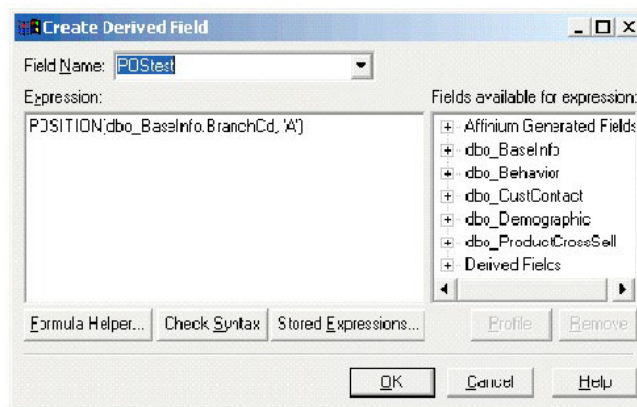
## Description

POSITION renvoie la position de l'octet de départ du modèle ou de la chaîne, dans la valeur d'une colonne (colName) qui doit être de type chaîne. Si le paramètre start est spécifié, la recherche commence à partir de là. L'occurrence correspond à la n<sup>ème</sup> occurrence du modèle à renvoyer.

**Remarque :** La recherche n'est pas sensible à la casse.

## Exemples

Dans l'exemple ci-dessous, nous recherchons le modèle ou la chaîne 'A' dans la valeur de la colonne dbo\_BaseInfo.BranchCd et affectons la valeur renvoyée au fichier dérivé POSTest.



L'exemple suivant illustre quelques lignes de la table contenant les valeurs de dbo\_BaseInfo.BranchCd et POSTest côte-à-côte.



Exemple plus complexe :

```
STRING_SEG(POSITION(CellCode,"X",1,2)+1,
STRING_LENGTH(CellCode),CellCode) = "AAA"
```

Cette commande renvoie les lignes pour lesquelles les valeurs de CellCode se terminent par "AAA" après la deuxième occurrence de "X".

---

## PLUS

### Syntaxe

data PLUS addend data + addend

### Paramètres

data

Plage de cibles contenant les nombres à ajouter. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de data, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

addend

Nombres à ajouter à toutes les valeurs de la colonne spécifiée. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Le nombre de colonnes dans data2 doit correspondre au nombre de colonnes dans data1, à moins que data2 ne soit une constante. Pour la définition du format de addend (identique à data), voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

### Description

PLUS ajoute les valeurs des deux plages de données spécifiées. Il renvoie une nouvelle colonne pour chaque colonne d'entrée, chacune contenant la colonne correspondante de data1 ajoutée à la colonne correspondante de data2 (à savoir, la

première colonne de data1 est ajoutée à la première colonne de data, la deuxième colonne à la deuxième colonne et ainsi de suite).

Si data2 est une constante, chaque valeur de data1 est augmentée de cette valeur. Si data2 contient une ou plusieurs colonnes, les calculs sont effectués ligne par ligne entre une colonne de data2 et une colonne de data2. La première ligne de data1 est ajoutée à la première ligne de data2, la seconde ligne, à la seconde ligne, etc. Ce calcul ligne par ligne génère un résultat pour chaque ligne jusqu'à la dernière valeur de la colonne la plus courte.

**Remarque :** L'opérateur PLUS peut être abrégé en signe plus (+).

Cette macro est disponible dans IBM Interact.

## Exemples

TEMP = 3 PLUS 4 ou TEMP = 3 + 4
Crée une colonne nommée TEMP contenant la valeur sept.
TEMP = V1 + 8
Crée une colonne nommée TEMP, dans laquelle chaque valeur correspond au contenu de la colonne V1 plus huit.
TEMP = V1 + V1
Crée une colonne nommée TEMP contenant deux fois le contenu de la colonne V1.
TEMP = V1 + V2
Crée une colonne nommée TEMP, dans laquelle chaque valeur est la valeur de ligne de la colonne V1 plus la valeur de ligne correspondante de la colonne V2.
TEMP = V1:V3 + V4:V6
Crée trois colonnes nommées TEMP, VX et VY. La colonne TEMP contient les valeurs de V1 plus les valeurs de ligne correspondantes de la colonne V4. La colonne VX ajoute les colonnes V2 et V5. La colonne VY calcule la somme des colonnes V5 et V6.
TEMP = V1[10:20] + V2 ou TEMP = V1[10:20] + V2[1:11]
Crée une colonne nommée TEMP, dans laquelle les 11 premières cibles contiennent les sommes des valeurs des lignes 10 à 20 de la colonne v1 et les valeurs des lignes 1 à 11 de la colonne V2. Les autres cibles de TEMP sont vides.

## Fonctions associées

Fonction	Description
MINUS	Soustrait une plage de données spécifiée d'une autre
SUM ou TOTAL	Calcule la somme d'une plage de cibles

---

## POW

### Syntaxe

base POW exponent base ^ exponent



## Paramètres

base

Valeurs numériques à élevée à une puissance exponentielle. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de base (identique à data), voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

exponent

Nombres exponentiels desquels les valeurs de data doivent être élevées. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Le nombre de colonnes dans exponent doit correspondre au nombre de colonnes dans base, à moins que base ne soit une constante. Pour la définition du format de exponent (identique à data), voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

## Description

POW élève les valeurs de la première plage de données à la puissance spécifiée dans la deuxième plage de données (il calcule  $\text{base}^{\text{exponent}}$ ). Il renvoie une nouvelle colonne pour chaque colonne d'entrée, chacune contenant le résultat de l'élévation de base à la puissance exponent (à savoir, la première colonne de data1 est élevée à la première colonne de data, la deuxième colonne à la deuxième colonne et ainsi de suite).

Si exponent est une constante, chaque valeur de base est élevée par cette valeur. Si exponent contient une ou plusieurs colonnes, les calculs sont effectués ligne par ligne entre une colonne de base et une colonne de exponent. La première ligne de base est élevée à la première ligne de exponent, la seconde ligne, à la seconde ligne, etc. Ce calcul ligne par ligne génère un résultat pour chaque ligne jusqu'à la dernière valeur de la colonne la plus courte.

**Remarque :** L'opérateur POW peut être abrégé en accent circonflexe (^). Par exemple,  $\text{TEMP} = 2^8$  est équivalent à  $\text{TEMP} = 2 \text{ POW } 8$ .

**Remarque :** Si la valeur  $x$  est trop faible, une erreur de dépassement est renvoyée. Cela se produit si  $\text{base}^{\text{exponent}}$  dépasse la valeur à virgule flottante sur 32 bits maximale ou minimale.

## Exemples

$\text{TEMP} = 2 \text{ POW } 3$ ou $\text{TEMP} = 2^3$
Crée une colonne nommée TEMP contenant la valeur huit.
$\text{TEMP} = V1 \wedge 0.5$
Crée une colonne nommée TEMP, dans laquelle chaque valeur correspond à la racine carrée du contenu de la colonne V1 (équivalent à $\text{SQRT}(V1)$ ).
$\text{TEMP} = V1 \wedge V3$
Crée une colonne nommée TEMP, dans laquelle chaque valeur est la valeur de ligne de la colonne V1 élevée à la valeur de ligne correspondante de la colonne V2.

<p>TEMP = V1:V3 ^ V4:V6</p> <p>Crée trois colonnes nommées TEMP, VX et VY. La colonne TEMP contient les valeurs de V1 élevées aux valeurs de ligne correspondantes de la colonne V4. La colonne VX contient le résultat de la colonne V2 élevée aux valeurs correspondantes dans la colonne V5. La colonne VY contient le résultat de la colonne V3 élevée aux valeurs correspondantes de V6.</p>
<p>TEMP = V1[10:20] POW V2 ou TEMP = V1[10:20] POW V2[1:11]</p> <p>Crée une colonne nommée TEMP, dans laquelle les 11 premières cibles contiennent le résultat de l'élevation des valeurs des lignes 10 à 20 de la colonne V1 par les valeurs des lignes 1 à 10 de la colonne V2. Les autres cibles de TEMP sont vides.</p>

## Fonctions associées

Fonction	Description
EXP	Calcule le nombre naturel (e) élevé à la puissance du contenu de chaque cible de la plage de données spécifiée
LN ou LOG	Calcule le logarithme naturel du contenu de la plage de données spécifiée
LN2	Calcule le logarithme base 2 du contenu de la plage de données spécifiée
LN10	Calcule le logarithme base 10 du contenu de la plage de données spécifiée

---

## RANDOM

### Syntaxe

RANDOM(num [, seed]) RANDOM(num, value1, value2 [, seed])

### Paramètres

num

Nombre de nombre aléatoires à générer. Il doit s'agir d'un entier positif supérieur à zéro.

value1

Limite sur les nombres aléatoires à générer. Il peut s'agir de toute valeur constante ou d'une expression convertie en constante. Si ce paramètre n'est pas fourni, la valeur par défaut est zéro.

value2

L'autre limite sur les nombres aléatoires à générer. Il peut s'agir de toute valeur constante ou d'une expression convertie en constante. Si ce paramètre n'est pas fourni, la valeur par défaut est un.

seed

Valeur de départ facultative à utiliser pour la génération de nombres aléatoires. Il doit s'agir d'un entier.

## Description

RANDOM génère une colonne de nombres aléatoires. Il renvoie une nouvelle colonne contenant num nombre aléatoires. Si value1 et value2 sont spécifiés, les nombres aléatoires sont générés entre ces limites (incluses). S'ils ne sont pas spécifiés, les valeurs sont générées par défaut entre zéro et un. Si seed est fourni, il est utilisé comme valeur de départ pour le générateur de nombres aléatoires.

**Remarque :** Si seed est supérieur ou égal à  $2^{32}$ , la valeur est remplacée par  $2^{32} - 1$ . Les valeurs de seed supérieures à  $2^{24}$  sont arrondies (la précision est perdue). Par conséquent, plusieurs valeurs peuvent posséder la même valeur seed.

## Exemples

TEMP = RANDOM()
Crée une colonne nommée TEMP contenant des nombres aléatoires de longueur illimitée.
TEMP = RANDOM(100)
Crée une colonne nommée TEMP contenant 100 nombres aléatoires compris entre 0,0 et 1,0.
TEMP = RANDOM(100, 5943049)
Crée une colonne nommée TEMP contenant 100 nombres aléatoires générés à partir du numéro de valeur de départ 5943049.
TEMP = RANDOM(100, 0, 100)
Crée une colonne nommée TEMP contenant 100 nombres aléatoires compris entre 0 et 100,0.
TEMP = RANDOM(100, 0, 100, 5943049)
Crée une colonne nommée TEMP contenant 100 nombres aléatoires compris entre 0 et 100 générés à partir du numéro de valeur de départ 5943049.

## Fonctions associées

Fonction	Description
RANDOM_GAUSS	Renvoie le nombre spécifié de valeurs aléatoires à partir d'une distribution de Gauss
SAMPLE_RANDOM	Renvoie des colonnes de n cibles, contenant chacune un échantillon aléatoire de la plage de données spécifiée

---

## RANDOM\_GAUSS

### Syntaxe

RANDOM\_GAUSS(num [, seed]) RANDOM\_GAUSS(num, mean, std [, seed])

### Paramètres

num

Nombre de nombre aléatoires à générer. Il doit s'agir d'un entier positif supérieur à zéro.

mean

Moyenne de la valeur de Gauss. Il peut s'agir de toute valeur constante ou d'une expression convertie en constante. Si ce paramètre n'est pas fourni, la valeur par défaut est zéro.

std

Ecart type de la valeur de Gauss. Il peut s'agir de toute valeur constante ou d'une expression convertie en constante. Si ce paramètre n'est pas fourni, la valeur par défaut est un.

seed

Valeur de départ facultative à utiliser pour la génération de nombres aléatoires. Il doit s'agir d'un entier. (Si une valeur non entière est fournie, la partie entière de cette valeur est automatiquement utilisée à la place.)

## Description

RANDOM\_GAUSS génère une colonne de nombres aléatoires basée sur une distribution de Gauss. Il renvoie une nouvelle colonne contenant num nombre aléatoires. Si mean et std sont spécifiés, les nombres aléatoires sont générés à l'aide d'une distribution de Gauss avec la moyenne et l'écart type spécifiés. S'ils ne sont pas spécifiés, la valeur de Gauss par défaut possède une moyenne de zéro et un écart type d'un. Si seed est fourni, il est utilisé comme valeur de départ pour le générateur de nombres aléatoires.

## Exemples

```
TEMP = RANDOM_GAUSS(100)
```

Crée une colonne nommée TEMP contenant 100 valeurs échantillonnées de manière aléatoire à partir d'une distribution de Gauss de moyenne nulle et d'écart type égal à un.

```
TEMP = RANDOM_GAUSS(500, 3)
```

Crée une colonne nommée TEMP contenant 100 valeurs échantillonnées de manière aléatoire à partir d'une distribution de Gauss de moyenne nulle et d'écart type égal à un. Le nombre 3 est utilisé comme valeur de départ pour le générateur de nombre aléatoire.

```
TEMP = RANDOM_GAUSS(5000, 100, 32)
```

Crée une colonne nommée TEMP contenant 5000 valeurs échantillonnées de manière aléatoire à partir d'une distribution de Gauss de moyenne égale à 100 et d'écart type égal à 32.

```
TEMP = RANDOM_GAUSS(500, -1, 2, 3)
```

Crée une colonne nommée TEMP contenant 500 valeurs échantillonnées de manière aléatoire à partir d'une distribution de Gauss de moyenne égale à -1 et d'écart type égal à 2. Le nombre 3 est utilisé comme valeur de départ pour le générateur de nombre aléatoire.

## Fonctions associées

Fonction	Description
RANDOM	Renvoie le nombre spécifié de nombres aléatoires

Fonction	Description
SAMPLE_RANDOM	Renvoie des colonnes de n cibles, contenant chacune un échantillon aléatoire de la plage de données spécifiée

## RANK



### Syntaxe

RANK(data [, nbins])

### Paramètres

data

Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs.

nbins

Nombre d'ensembles dans lequel data sera déterminé. La valeur par défaut est de dix.

### Description

RANK divise les données en groupes nbins (10 par défaut), chacun contenant approximativement un nombre égal de valeurs distinctes, et renvoie le groupe auquel chaque ligne de données appartient. La sortie est comprise entre 1 et nbins ; si le nombre de valeurs distinctes de données est inférieur à nbins, la sortie est comprise entre 1 et le nombre de valeurs distinctes de données.

RANK impose automatiquement une limite supérieure de 1024\*1024 sur la valeur de nbins. Ce même nombre est également utilisé comme nombre maximal de valeurs distinctes à suivre ; les valeurs distinctes suivantes seront incluses dans l'ensemble le plus élevé.

### Exemples

```
TEMP=RANK(V6)
```

Crée une colonne nommée TEMP avec chaque valeur, entre un et dix, représentant l'ensemble de données auquel la ligne appartient. Dans ce cas, le nombre d'ensembles par défaut, 10, est appliqué.

```
TEMP=RANK(V6,15)
```

Crée une colonne nommée TEMP avec chaque valeur, entre un et quinze, représentant l'ensemble de données auquel la ligne appartient.

```
TEMP = REPEAT(3, V1:V3)
```

Crée trois colonnes nommées TEMP, VX et VY. TEMP contient trois copies de la colonne V1, VX contient trois copies de la colonne V2 et VY contient trois copies de la colonne V3. Les colonnes de longueur inégales sont étendues jusqu'à la longueur de la colonne la plus longue parmi les colonnes V1 à V3.

<pre>TEMP = REPEAT(10, V1[10:50]:V2)</pre> <p>Crée deux colonnes nommées TEMP et VX. Les valeurs de la colonne TEMP correspondent à 10 copies des lignes 10 à 50 de la colonne V1 et les valeurs de la colonne VX correspondent à 10 copies des lignes 10 à 50 de la colonne V2.</p>
<pre>TEMP = REPEAT((1,2,3), (10, 20, 30))</pre> <p>Crée une colonne nommée TEMP, contenant les valeurs de cible 10, 20, 20, 30, 30, 30.</p>
<pre>TEMP = REPEAT(V1, V2)</pre> <p>Crée une colonne nommée TEMP. La valeur de la cible V2[1] est répétée V1[1] fois, la valeur de la cible V2[2] est répétée V1[2] fois et ainsi de suite jusqu'à la fin de la colonne V1.</p>
<pre>TEMP = REPEAT(V1, V2:V3)</pre> <p>Crée deux colonnes nommées TEMP et VX. La colonne TEMP contient des copies des cibles de V2 ; la colonne VX contient des copies des cibles de V3. Il existe V1[1] copies de V2[1] et V3[1], V1[2] copies de V2[2] et V3[2]. Cela continue jusqu'à la fin de la colonne V1 ou la fin de la colonne la plus longue de data (si elle est plus courte). Les colonnes les plus courtes de data sont complétées par des zéros.</p>
<pre>TEMP = REPEAT(3, V1, COL)</pre> <p>Crée trois colonnes nommées TEMP, VX et VY, chacune représentant une copie de la colonne V1.</p>
<pre>TEMP = REPEAT(2, V1:V2, COL)</pre> <p>Crée quatre colonnes nommées TEMP, VX, VY et VZ. La colonne TEMP est une copie de la colonne V1 ; la colonne VX est une copie de la colonne V2, la colonne VY est une copie de la colonne V1 ; et la colonne VZ est une copie de la colonne V2.</p>

## REPEAT



### Syntaxe

```
REPEAT(num_times, data [, keyword])
```

### Paramètres

num\_times

Nombre de fois que la plage de données spécifiée doit être répétée. Il peut s'agir d'une constante, d'une colonne ou d'une expression convertie comme l'une de ces valeurs. Toutes les valeurs doivent être des entiers positifs.

data

Valeurs numériques à répéter. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de data, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

keyword

Ce mot clé facultatif détermine la manière dont la plage de données spécifiée est répliquée. Sélectionnez l'une des valeurs suivantes :

ROW - Répète les lignes de data verticalement (valeur par défaut)

COL - Répète les colonnes de data horizontalement

Pour plus de détails sur l'utilisation des mots clés dans IBM Campaign, voir «Spécifications de format», à la page 9.

Pour plus de détails sur l'utilisation des mots clés dans IBM PredictiveInsight, voir «Spécifications de format», à la page 27.

## Description

REPEAT répète la plage de données spécifiée un certain nombre de fois, verticalement (COL) ou horizontalement (ROW).

Avec le mot clé COL, il renvoie une nouvelle colonne par colonne d'entrée contenant num\_times copies de data, concaténées verticalement. Si les colonnes en entrée de data possèdent des longueurs différentes, les colonnes les plus courtes sont automatiquement remplies pour que les tailles de toutes les colonnes soient identiques. Les colonnes numériques sont remplies avec des zéros et les colonnes de chaîne de texte sont complétées par des chaînes vides ("").

Avec le mot clé ROW, le nombre de colonnes renvoyé varie selon que num\_times est une constante ou une colonne. Si num\_times est une constante, il renvoie num\_times fois le nombre de colonnes de data. Si num\_times correspond à une colonne, il renvoie le nombre spécifié de copies de chaque colonne de data, la valeur de la première ligne de num\_times représentant le nombre de répétitions de la première colonne de data ; la valeur de la deuxième ligne représentant le nombre de répétitions de la deuxième colonne de data, etc. Les lignes supérieures au nombre de colonnes de data sont ignorées.

## Exemples

<pre>TEMP = REPEAT(10, 1) ou TEMP = REPEAT(10, 1, ROW)</pre>
Crée une colonne nommée TEMP contenant dix 1.
<pre>TEMP = REPEAT(2, V1)</pre>
Crée une colonne nommée TEMP contenant deux copies de la colonne V1 concaténées verticalement.
<pre>TEMP = REPEAT(3, V1:V3)</pre>
Crée trois colonnes nommées TEMP, VX et VY. TEMP contient trois copies de la colonne V1, VX contient trois copies de la colonne V2 et VY contient trois copies de la colonne V3. Les colonnes de longueur inégales sont étendues jusqu'à la longueur de la colonne la plus longue parmi les colonnes V1 à V3.
<pre>TEMP = REPEAT(10, V1[10:50]:V2)</pre>
Crée deux colonnes nommées TEMP et VX. Les valeurs de la colonne TEMP correspondent à 10 copies des lignes 10 à 50 de la colonne V1 et les valeurs de la colonne VX correspondent à 10 copies des lignes 10 à 50 de la colonne V2.
<pre>TEMP = REPEAT((1,2,3), (10, 20, 30))</pre>
Crée une colonne nommée TEMP, contenant les valeurs de cible 10, 20, 20, 30, 30, 30.

```
TEMP = REPEAT(V1, V2)
```

Crée une colonne nommée TEMP. La valeur de la cible V2[1] est répétée V1[1] fois, la valeur de la cible V2[2] est répétée V1[2] fois et ainsi de suite jusqu'à la fin de la colonne V1.

```
TEMP = REPEAT(V1, V2:V3)
```

Crée deux colonnes nommées TEMP et VX. La colonne TEMP contient des copies des cibles de V2 ; la colonne VX contient des copies des cibles de V3. Il existe V1[1] copies de V2[1] et V3[1], V1[2] copies de V2[2] et V3[2]. Cela continue jusqu'à la fin de la colonne V1 ou la fin de la colonne la plus longue de data (si elle est plus courte). Les colonnes les plus courtes de data sont complétées par des zéros.

```
TEMP = REPEAT(3, V1, COL)
```

Crée trois colonnes nommées TEMP, VX et VY, chacune représentant une copie de la colonne V1.

```
TEMP = REPEAT(2, V1:V2, COL)
```

Crée quatre colonnes nommées TEMP, VX, VY et VZ. La colonne TEMP est une copie de la colonne V1 ; la colonne VX est une copie de la colonne V2, la colonne VY est une copie de la colonne V1 ; et la colonne VZ est une copie de la colonne V2.

---

## ROTATE\_LEFT



### Syntaxe

```
ROTATE_LEFT(num_cols, data)
```

### Paramètres

num\_cols

Nombre de colonnes auxquelles une rotation vers la gauche doit être appliquée. Cette valeur doit correspondre à un entier non négatif. La valeur de zéro copie les lignes sans rotation.

data

Valeurs numériques auxquelles une rotation vers la gauche doit être appliquée. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de data, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

### Description

ROTATE\_LEFT effectue une rotation vers la gauche de la plage de données spécifiée en fonction d'un certain nombre de colonnes. Il renvoie une nouvelle colonne par colonne d'entrée, chacune contenant une copie de la colonne d'entrée correspondante à laquelle une rotation de num\_cols positions vers la gauche a été appliquée. Les colonnes pour lesquelles une rotation à gauche est effectuée sont bouclées à droite.



**Remarque :** ROTATE\_LEFT ne fonctionne qu'avec des données numériques. Aucune des données fournies dans le paramètre data ne peut être du texte ASCII.

## Exemples

<pre>TEMP = ROTATE_LEFT(1, MERGE(1, 2, 3))</pre>
Crée trois colonnes nommées TEMP, VX et VY. TEMP contient la valeur deux, VX contient la valeur trois et VY contient la valeur un.
<pre>TEMP = ROTATE_LEFT(0, V1:V3)</pre>
Crée trois colonnes nommées TEMP, VX et VY. TEMP est une copie de la colonne V1, VX est une copie de la colonne V2 et VY est une copie de la colonne V3.
<pre>TEMP = ROTATE_LEFT(4, V1:V3)</pre>
Crée trois colonnes nommées TEMP, VX et VY. TEMP est une copie de la colonne V2, VX est une copie de la colonne V3 et VY est une copie de la colonne V1.
<pre>TEMP = ROTATE_LEFT(1, V1[10:50]:V2)</pre>
Crée deux colonnes nommées TEMP et VX, contenant des valeurs dans les 41 premières lignes (les autres cibles sont vides). Les valeurs de la colonne TEMP correspondent aux valeurs des lignes 10 à 50 de la colonne V2 et les valeurs de la colonne VX correspondent aux valeurs des lignes 10 à 50 de la colonne V1.

## Fonctions associées

Fonction	Description
ROTATE_RIGHT	Applique aux colonnes de la plage de données spécifiée une rotation vers la droite

---

## ROTATE\_RIGHT



### Syntaxe

```
ROTATE_RIGHT(num_cols, data)
```

### Paramètres

num\_cols

Nombre de colonnes auxquelles une rotation vers la droite doit être appliquée. Cette valeur doit correspondre à un entier non négatif. La valeur de zéro copie les lignes sans rotation.

data

Valeurs numériques auxquelles une rotation vers la droite doit être appliquée. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de data, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

## Description

ROTATE\_RIGHT effectue une rotation vers la droite de la plage de données spécifiée en fonction d'un certain nombre de colonnes. Il renvoie une nouvelle colonne par colonne d'entrée, chacune contenant une copie de la colonne d'entrée correspondante à laquelle une rotation de num\_cols positions vers la droite a été appliquée. Les colonnes pour lesquelles une rotation à droite est effectuée sont bouclées à gauche.

**Remarque :** ROTATE\_RIGHT ne fonctionne qu'avec des données numériques. Aucune des données fournies dans le paramètre data ne peut être du texte ASCII.

## Exemples

TEMP = ROTATE_RIGHT(1, MERGE(1, 2, 3))
Crée trois colonnes nommées TEMP, VX et VY. TEMP contient la valeur trois, VX contient la valeur un et VY contient la valeur deux.
TEMP = ROTATE_RIGHT(0, V1:V3)
Crée trois colonnes nommées TEMP, VX et VY. TEMP est une copie de la colonne V1, VX est une copie de la colonne V2 et VY est une copie de la colonne V3.
TEMP = ROTATE_RIGHT(4, V1:V3)
Crée trois colonnes nommées TEMP, VX et VY. TEMP est une copie de la colonne V3, VX est une copie de la colonne V1 et VY est une copie de la colonne V2.
TEMP = ROTATE_RIGHT(1, V1[10:50]:V2)
Crée deux colonnes nommées TEMP et VX, contenant des valeurs dans les 41 premières lignes (les autres cibles sont vides). Les valeurs de la colonne TEMP correspondent aux valeurs des lignes 10 à 50 de la colonne V2 et les valeurs de la colonne VX correspondent aux valeurs des lignes 10 à 50 de la colonne V1.

## Fonctions associées

Fonction	Description
ROTATE_LEFT	Applique aux colonnes de la plage de données spécifiée une rotation vers la gauche

---

## ROUND

### Syntaxe

ROUND(data)

### Paramètres

data

Valeurs numériques à arrondir. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de data, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

## Description

ROUND arrondit les valeurs de la plage de données spécifiée à l'entier le plus proche. Il renvoie une nouvelle colonne par colonne d'entrée, chacune contenant la valeur arrondie des nombres de la colonne d'entrée correspondante. Les nombres à égale distance de deux entiers sont arrondis vers le haut (par exemple, 2.5 est arrondi à 3.0 et -2.5 est arrondi à -2.0).

## Exemples

TEMP = ROUND(3.2)
Crée une colonne nommée TEMP contenant la valeur trois.
TEMP = ROUND(V1)
Crée une colonne nommée TEMP, dans laquelle chaque valeur correspond à la valeur arrondie du contenu de la colonne V1.
TEMP = ROUND(V1:V3)
Crée trois colonnes nommées TEMP, VX et VY. Les valeurs de la colonne TEMP correspondent aux valeurs arrondies du contenu de la colonne V1, les valeurs de la colonne VX correspondent aux valeurs arrondies du contenu de la colonne V2 et les valeurs de la colonne VY correspondent aux valeurs arrondies du contenu de la colonne V3.
TEMP = ROUND(V1[10:20])
Crée une colonne nommée TEMP, dans laquelle les 11 premières cibles contiennent les valeurs arrondies des lignes 10 à 20 de la colonne V1. Les autres cibles de TEMP sont vides.
TEMP = ROUND(V1[1:5]:V2)
Crée deux colonnes nommées TEMP et VX, contenant des valeurs dans les lignes 1 à 5 (les autres cibles sont vides). Les valeurs de la colonne TEMP correspondent aux valeurs arrondies des lignes correspondantes de la colonne V1 et les valeurs de la colonne VX correspondent aux valeurs arrondies des lignes correspondantes de la colonne V2.

## Fonctions associées

Fonction	Description
INT	Calcule la valeur entière (arrondie) du contenu de la plage de données spécifiée
MOD	Calcule le modulo du contenu de la plage de données spécifiée
TRUNCATE	Renvoie la partie non décimale de chaque valeur de la plage de données spécifiée

---

## ROWNUM

### Syntaxe

ROWNUM()

### Description

ROWNUM génère des nombres séquentiels de un à n, où n correspond au nombre d'enregistrements. Le nombre pour le premier enregistrement est 1, celui du deuxième 2 et ainsi de suite

**Remarque :** Le nombre maximal d'enregistrements que ROWNUM peut gérer est de deux milliards.

---

## RTRIM

### Syntaxe

```
RTRIM(data)
```

### Paramètres

data

### Description

RTRIM supprime les derniers caractères d'espace de chaque valeur de chaîne contenue dans la plage de données spécifiée, puis renvoie la chaîne convertie. Il renvoie une nouvelle colonne pour chaque colonne d'entrée.

Cette macro est disponible dans IBM Interact.

### Exemples

```
Temp = RTRIM "gold "
```

Crée une chaîne nommée Temp qui contient "gold".

---

## SAMPLE\_RANDOM



### Syntaxe

```
SAMPLE_RANDOM(num_samples, data [, seed])
```

### Paramètres

num\_samples

Nombre d'échantillons à prélever de chaque colonne de la plage de données spécifiée.

data

Valeurs à échantillonner de manière aléatoire. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de data, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

seed

Valeur de départ facultative à utiliser pour la génération de nombres aléatoires. Il doit s'agir d'un entier. (Si une valeur non entière est fournie, la partie entière de cette valeur est automatiquement utilisée à la place.)

## Description

SAMPLE\_RANDOM échantillonne de manière aléatoire la plage de données spécifiée. Il renvoie une nouvelle colonne par colonne d'entrée, chacune contenant num\_samples nombres échantillonnés de manière aléatoire à partir de la colonne d'entrée correspondante de data. Les échantillons sont utilisés suivant l'ordre dans lequel ils apparaissent dans chaque colonne (à savoir, les valeurs des données conservent le même ordre les unes par rapport aux autres). Si seed est fourni, il est utilisé comme valeur de départ pour le générateur de nombres aléatoires.

## Exemples

TEMP = SAMPLE_RANDOM(100, 3)
Crée une colonne nommée TEMP avec 100 cibles contenant toutes la valeur 3.
TEMP = SAMPLE_RANDOM(100, V1)
Crée une colonne nommée TEMP, dans laquelle chaque valeur correspond à un échantillon aléatoire du contenu de la colonne V1.
TEMP = SAMPLE_RANDOM(50, V1:V3)
Crée trois colonnes nommées TEMP, VX et VY contenant chacune 50 valeurs. Les valeurs de la colonne TEMP correspondent aux échantillons aléatoires de la colonne V1, es valeurs de la colonne VX correspondent aux échantillons aléatoires de la colonne V2 et les valeurs de la colonne VY correspondent aux échantillons aléatoires de la colonne V3.
TEMP = SAMPLE_RANDOM(100, V1[10:50]:V3)
Crée trois colonnes nommées TEMP, VX et VY contenant chacune 41 valeurs. Les valeurs de la colonne TEMP correspondent aux échantillons aléatoires des cibles 10 à 50 de la colonne V1, les valeurs de la colonne VX correspondent aux échantillons aléatoires des cibles 10 à 50 de la colonne V2 et les valeurs de la colonne VY correspondent aux échantillons aléatoires des cibles 10 à 50 de la colonne V3. Les autres cibles de TEMP sont vides.

## Fonctions associées

Fonction	Description
RANDOM	Renvoie le nombre spécifié de nombres aléatoires
RANDOM_GAUSS	Renvoie le nombre spécifié de valeurs aléatoires à partir d'une distribution de Gauss
SUBSAMPLE	Réduit la quantité de données en renvoyant toutes les n ème valeurs de ligne

---

## SELECT



### Syntaxe

```
SELECT(col_nums, data) SELECT(from_col, data) SELECT(from_col, to_col, data)
```

## Paramètres

`col_nums`

Nom d'une colonne contenant les numéros de colonne à extraire de la plage de données spécifiée (par exemple, une colonne contenant les nombres 1, 3, 4 et 7 extrait les première, troisième, quatrième et septième colonnes de la plage de données spécifiée). Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de `col_nums` (identique à `data`), voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

`from_col`

Position numérique de la colonne à extraire de la plage de données spécifiée. La valeur un extrait la première colonne d'une plage de données spécifiée.

`to_col`

Si ce paramètre est fourni la plage de données spécifiée, `from_col` est utilisé comme point de départ et doit être une colonne ou une plage de cibles. Le point final est spécifié par `to_col`. Cette valeur doit être supérieure à `from_col`.

`data`

Plage de cibles contenant les colonnes à extraire. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de `data`, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

## Description

SELECT renvoie les colonnes spécifiées à partir d'une plage de données. Les colonnes à sélectionner peuvent être spécifiées de différentes manières. Le paramètre `col_nums` contient les numéros de colonne à extraire de la plage de données spécifiée. Pour extraire une seule colonne, utilisez `from_col`. Pour extraire une plage de colonnes consécutives, spécifiez `from_col` comme point de départ (il doit s'agir d'une colonne ou d'une plage de cibles) et `to_col` comme colonne de fin.

Si `data` est une valeur scalaire (à savoir, une constante ou une variable contenant une valeur unique), la sélection de la première colonne renvoie une nouvelle colonne contenant cette constante. La sélection d'une autre colonne à partir d'une constante renvoie une colonne contenant ???.

Cette macro est souvent imbriquée dans des fonctions plus complexes.

**Remarque :** Pour extraire plusieurs colonnes non adjacentes, utilisez la fonction macro COLUMN pour créer une colonne contenant les numéros de colonne à sélectionner. Voir les exemples ci-après.

## Exemples

```
TEMP = SELECT(1, 3)Crée une colonne nommée TEMP contenant le nombre trois.
```

TEMP = SELECT(1, V1) ou TEMP = SELECT(1, V1:V3) Crée une colonne nommée TEMP, qui est une copie de la colonne V1.
TEMP = SELECT(2, 4, V1:V5) Crée trois colonnes nommées TEMP, VX et VY. TEMP est une copie de la colonne V2, VX est une copie de la colonne V3 et VY est une copie de la colonne V4.
TEMP = SELECT(COLUMN(1,4), V6:V10) Crée deux colonnes nommées TEMP et VX. TEMP est une copie de la colonne V6 et VX est une copie de la colonne V9.
TEMP = SELECT(COLUMN(1,4), V6[25:74]:V10) Crée deux colonnes nommées TEMP et VX, contenant chacune 50 valeurs. TEMP est une copie des cibles 25 à 74 de la colonne V6 et VX est une copie des cibles 25 à 74 de la colonne V9. Les autres cibles de TEMP sont vides.

## Fonctions associées

Fonction	Description
EXTRACT	Extrait les lignes contenant les valeurs d'une colonne de prédicat
MERGE	Crée un groupe de données en concaténant horizontalement les valeurs en entrée

---

## SIGN

### Syntaxe

SIGN(data)

Paramètres

data

Valeurs numériques dont la signe doit être déterminé. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de data, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

### Description

SIGN teste le signe des valeurs de la plage de données spécifiée. Il renvoie une nouvelle colonne par colonne d'entrée, chacune contenant le signe des nombres de la colonne d'entrée correspondante. Un un positif est renvoyé pour toutes les valeurs supérieures à zéro ; un un négatif est renvoyé pour toutes les valeurs inférieures à zéro ; zéro est renvoyé pour les valeurs égales à zéro.

### Exemples

TEMP = SIGN(-3) Crée une colonne nommée TEMP contenant la valeur -1.
---

TEMP = SIGN(MERGE(3, 2, 0))
Crée trois colonnes nommées TEMP, VX et VY. TEMP contient la valeur 1, VX contient la valeur -1 et VY contient la valeur 0.
TEMP = SIGN(V1)
Crée une colonne nommée TEMP, dans laquelle chaque valeur correspond au signe du contenu de la colonne V1.
TEMP = SIGN(V1:V3)
Crée trois colonnes nommées TEMP, VX et VY. Les valeurs de la colonne TEMP correspondent aux signes du contenu de la colonne V1, les valeurs de la colonne VX correspondent aux signes du contenu de la colonne V2 et les valeurs de la colonne VY correspondent aux signes du contenu de la colonne V3.
TEMP = SIGN(V1[10:20])
Crée une colonne nommée TEMP, dans laquelle les 11 premières cibles contiennent les signes des valeurs des lignes 10 à 20 de la colonne V1. Les autres cibles de TEMP sont vides.
TEMP = SIGN(V1[10:50]:V2)
Crée deux colonnes nommées TEMP et VX, contenant des valeurs dans les lignes 1 à 41 (les autres cibles sont vides). Les valeurs de la colonne TEMP correspondent aux sinus des valeurs des lignes 10 à 50 de la colonne V1 et les valeurs de la colonne VX correspondent aux signes des valeurs des lignes 10 à 50 de la colonne V2.

## Fonctions associées

Fonction	Description
ABS	Calcule la valeur absolue du contenu de la plage de données spécifiée
INVERSE	Calcule la valeur négative du contenu de la plage de données spécifiée

---

## SIN

### Syntaxe

SIN(data [, units\_keyword])

### Paramètres

data

Valeurs numériques dont le sinus doit être calculé. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de data, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

units\_keyword

Ce mot clé facultatif détermine si les valeurs en entrée et les résultats sont interprétés comme des degrés ou des radians. Sélectionnez l'une des valeurs suivantes :



RADIAN - Effectue les calculs en radians (valeur par défaut)

DEGREE - Effectue les calculs en degrés

Si ce paramètre n'est pas spécifié, la valeur par défaut est radians. (Pour convertir des radians en degrés, divisez par PI et multipliez par 180.)

Pour plus de détails sur l'utilisation des mots clés dans IBM Campaign, voir «Spécifications de format», à la page 9.

Pour plus de détails sur l'utilisation des mots clés dans IBM PredictiveInsight, voir «Spécifications de format», à la page 27.

## Description

SIN calcule le sinus des valeurs de la plage de données spécifiée. Il renvoie une nouvelle colonne par colonne d'entrée, chacune contenant le sinus des nombres de la colonne d'entrée correspondante.

## Exemples

<code>TEMP = SIN(PI/2), TEMP = SIN(PI/2, 0) ou TEMP = SIGN(PI/2, RADIAN)</code>
Crée une colonne nommée TEMP contenant la valeur un.
<code>TEMP = SIN(V1)</code>
Crée une colonne nommée TEMP, où chaque valeur correspond au sinus (en radians) du contenu de la colonne V1.
<code>TEMP = SIN(V1:V3, 1) ou TEMP = SIN(V1:V3, DEGREE)</code>
Crée trois colonnes nommées TEMP, VX et VY. Les valeurs de la colonne TEMP correspondent aux sinus du contenu de la colonne V1, les valeurs de la colonne VX correspondent aux sinus du contenu de la colonne V2 et les valeurs de la colonne VY correspondent aux sinus du contenu de la colonne V3. Toutes les valeurs sont en degrés.
<code>TEMP = SIN(V1[10:50]:V2)</code>
Crée deux colonnes nommées TEMP et VX, contenant des valeurs dans les lignes 1 à 41 (les autres cibles sont vides). Les valeurs de la colonne TEMP correspondent aux sinus des valeurs des lignes 10 à 50 de la colonne V1 et les valeurs de la colonne VX correspondent aux sinus des valeurs des lignes 10 à 50 de la colonne V2. Toutes les valeurs sont en radians.

## Fonctions associées

Fonction	Description
ASIN	Calcule l'arc sinus du contenu de la plage de données spécifiée
COS	Calcule le cosinus du contenu de la plage de données spécifiée
SINH	Calcule le sinus hyperbolique du contenu de la plage de données spécifiée
TAN	Calcule la tangente du contenu de la plage de données spécifiée

## Syntaxe

SINH(data [, units\_keyword])

## Paramètres

data

Valeurs numériques dont le sinus hyperbolique doit être calculé. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de data, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

units\_keyword

Ce mot clé facultatif détermine si les valeurs en entrée et les résultats sont interprétés comme des degrés ou des radians. Sélectionnez l'une des valeurs suivantes :

RADIAN - Effectue les calculs en radians (valeur par défaut)

DEGREE - Effectue les calculs en degrés

Si ce paramètre n'est pas spécifié, la valeur par défaut est radians. (Pour convertir des radians en degrés, divisez par PI et multipliez par 180.)

Pour plus de détails sur l'utilisation des mots clés dans IBM Campaign, voir «Spécifications de format», à la page 9.

Pour plus de détails sur l'utilisation des mots clés dans IBM PredictiveInsight, voir «Spécifications de format», à la page 27.

## Description

SINH calcule le sinus hyperbolique des valeurs de la plage de données spécifiée. Il renvoie une nouvelle colonne par colonne d'entrée, chacune contenant le sinus hyperbolique des nombres de la colonne d'entrée correspondante. Pour  $x$  en radians, le sinus hyperbolique d'un nombre est :

$$\sinh(x) = \frac{e^x - e^{-x}}{2}$$

où  $e$  représente le nombre naturel, 2.7182818.

**Remarque :** Si la valeur  $x$  est trop élevée, une erreur de dépassement est renvoyée. Cela se produit si  $\sinh(x)$  dépasse la valeur maximale à virgule flottante sur 32 bits.

## Exemples

<code>TEMP = SINH(1), TEMP = SINH(1, 0) ou TEMP = SINH(1, RADIAN)</code> Crée une colonne nommée TEMP contenant la valeur 1.18.
<code>TEMP = SINH(V1)</code> Crée une colonne nommée TEMP, où chaque valeur correspond au sinus hyperbolique (en radians) du contenu de la colonne V1.
<code>TEMP = SINH(V1:V3, 1) ou TEMP = SINH(V1:V3, DEGREE)</code> Crée trois colonnes nommées TEMP, VX et VY. Les valeurs de la colonne TEMP correspondent aux sinus hyperboliques du contenu de la colonne V1, les valeurs de la colonne VX correspondent aux sinus hyperboliques du contenu de la colonne V2 et les valeurs de la colonne VY correspondent aux sinus hyperboliques du contenu de la colonne V3. Toutes les valeurs sont en degrés.
<code>TEMP = SINH(V1[10:50]:V2)</code> Crée deux colonnes nommées TEMP et VX, contenant des valeurs dans les lignes 1 à 41 (les autres cibles sont vides). Les valeurs de la colonne TEMP correspondent aux sinus hyperboliques des valeurs des lignes 10 à 50 de la colonne V1 et les valeurs de la colonne VX correspondent aux sinus hyperboliques des valeurs des lignes 10 à 50 de la colonne V2. Toutes les valeurs sont en radians.

## Fonctions associées

Fonction	Description
COSH	Calcule le cosinus hyperbolique du contenu de la plage de données spécifiée
SIN	Calcule le sinus du contenu de la plage de données spécifiée
TANH	Calcule la tangente hyperbolique du contenu de la plage de données spécifiée

---

## SKEW



### Syntaxe

`SKEW(data [, keyword])`

### Paramètres

data

Valeurs numériques dont l'asymétrie doit être calculée. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de data, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM. data doit comporter au moins trois valeurs.

keyword

Ce mot clé facultatif détermine la manière dont le calcul est effectué sur la plage de données d'entrée. Sélectionnez l'une des valeurs suivantes :

ALL - Effectue le calcul sur toutes les cibles de data (valeur par défaut)

COL - Effectue le calcul séparément pour chaque colonne de data

ROW - Effectue le calcul séparément pour chaque ligne de data

Pour plus de détails sur l'utilisation des mots clés dans IBM Campaign, voir «Spécifications de format», à la page 9.

Pour plus de détails sur l'utilisation des mots clés dans IBM PredictiveInsight, voir «Spécifications de format», à la page 27.

## Description

SKEW calcule l'asymétrie de la distribution de toutes les cibles de la plage de données spécifiée. La dissymétrie mesure le degré d'asymétrie d'une distribution par rapport à sa moyenne. Une valeur d'asymétrie positive indique une distribution avec une queue asymétrique penchant vers des valeurs plus positives ; une asymétrie négative indique une distribution avec une queue asymétrique penchant vers des valeurs plus négatives. Une valeur d'asymétrie de zéro indique que la distribution est symétrique par rapport à sa moyenne.

L'asymétrie est calculée comme suit :

$$\frac{n}{(n-1)(n-2)} \sum_1^n \left( \frac{x_i - \text{mean}}{\sigma} \right)^3$$

où  $n$  représente le nombre d'échantillons dans la distribution,  $\text{mean}$  correspond à la moyenne et  $\sigma$  représente l'écart-type de la distribution. Au moins trois valeurs doivent être fournies pour calculer l'asymétrie.

**Remarque :** Si l'écart type  $\sigma = 0$ , SKEW renvoie zéro.

## Exemples

TEMP = SKEW(3) ou TEMP = SKEW(3, ALL)
Crée une colonne nommée TEMP contenant la valeur zéro.
TEMP = SKEW(MERGE(3, 7, 2))
Crée une colonne nommée TEMP contenant la valeur 0.14.
TEMP = SKEW(V1)
Crée une colonne nommée TEMP contenant une valeur unique qui correspond à l'asymétrie du contenu de la colonne V1.

TEMP = SKEW(V1:V3)
Crée une colonne nommée TEMP contenant une valeur unique qui correspond à l'asymétrie du contenu des colonnes V1, V2 et V3.
TEMP = SKEW(V1[10:20])
Crée une colonne nommée TEMP contenant une valeur unique qui correspond à l'asymétrie des cibles des lignes 10 à 20 de la colonne V1.
TEMP = SKEW(V1[1:5]:V4)
Crée une colonne nommée TEMP contenant une valeur unique qui correspond à l'asymétrie des cibles des lignes 1 à 5 des colonnes V1 à V4.
TEMP = SKEW(V1:V3, COL)
Crée trois colonnes nommées TEMP, VX et VY. La valeur unique de la colonne TEMP correspond à l'asymétrie du contenu de la colonne V1, la valeur unique de la colonne VX correspond à l'asymétrie du contenu de la colonne V2 et la valeur unique de la colonne VY correspond à l'asymétrie du contenu de la colonne V3.
TEMP = SKEW(V1[1:5]:V3,COL)
Crée trois colonnes nommées TEMP, VX et VY contenant chacune une seule valeur. La valeur de la colonne TEMP correspond à l'asymétrie des cibles des lignes 1 à 5 de la colonne V1, la valeur de la colonne VX correspond à l'asymétrie des cibles des lignes 1 à 5 de la colonne V2, et la valeur de la colonne VY correspond à l'asymétrie des cibles des lignes 1 à 5 de la colonne V3.
TEMP = SKEW(V1:V3, ROW)
Crée une colonne nommée TEMP dans laquelle chaque entrée de cible correspond à l'asymétrie de la ligne correspondante dans les colonnes V1, V2 et V3.
TEMP = SKEW(V1[10:50]:V3,ROW)
Crée une colonne nommée TEMP, dans laquelle les 41 premières cibles contiennent l'asymétrie des valeurs des lignes 10 à 50 dans les colonnes V1 à V3. Les autres cibles de TEMP sont vides.

## Fonctions associées

Fonction	Description
AVG ou MEAN	Calcule la moyenne arithmétique d'une plage de cibles
KURTOSIS	Calcule le kurtosis d'une plage de cibles
STAT	Calcule les premier, deuxième, troisième et quatrième moments de la plage de données spécifiée
VARIANCE	Calcule la variance d'une plage de cibles

---

## SLIDE\_WINDOW



### Syntaxe

SLIDE\_WINDOW(width, data [, increment])

## Paramètres

width

Taille (nombre de lignes verticales) de la fenêtre dynamique.

data

Plage de cibles à utiliser pour faire glisser une fenêtre pour générer des données. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de data, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

increment

Nombre de lignes à incrémenter chaque fois ; la valeur par défaut est un. Il doit s'agir d'un entier supérieur à zéro (et inférieur à la longueur de data).

## Description

SLIDE\_WINDOW utilise une taille de fenêtre spécifiée et la fait glisser sur la plage de données spécifiée, créant ainsi des modèles à partir des valeurs de données de la fenêtre. Il renvoie (nombre de colonnes en entrée x width) colonnes. La fenêtre dynamique commence au début de data et couvre width lignes. Les valeurs de données de cette fenêtre (lues de gauche à droite et de haut en bas) sont concaténées pour constituer une seule ligne dans la plage de données en sortie. La fenêtre dynamique se déplace vers le bas de increment lignes à chaque fois.

Supposons par exemple que les colonnes V1 et V2 contiennent les données suivantes :

```
10 2 20 3 30 4 40 5 50 ...
```

L'expression V3=SLIDE\_WINDOW(2, V1:V2) génère alors la sortie suivante dans les colonnes V3:V6 :

```
10 2 20 2 20 3 30 3 30 4 40 ...
```

Les deux premières lignes créent la première ligne de sortie. Ensuite, la fenêtre glisse vers le bas d'une ligne pour créer le modèle suivant, etc.

**Remarque :** Cette fonction est utile pour créer des modèles à partir de données de série temporelle.

## Exemples

```
TEMP = SLIDE_WINDOW(1, V1)
```

Crée une colonne nommée TEMP contenant une copie des valeurs de la colonne V1.

```
TEMP = SLIDE_WINDOW(3, V1:V3)
```

Crée neuf colonnes dont chaque ligne contient une fenêtre de données de trois par trois à partir des colonnes V1:V3. Les lignes 1 à 3 de l'entrée constituent la première ligne de la sortie, les lignes 2 à 4 constituent la deuxième ligne, etc.

TEMP = SLIDE_WINDOW(2, V1:V3[10:20])
Crée six colonnes dont chaque ligne contient une fenêtre de données de trois par deux à partir des lignes 10 à 20 des colonnes V1:V3. Les lignes 10 à 11 de l'entrée constituent la première ligne de la sortie, les lignes 11 à 12 constituent la deuxième ligne, etc.
TEMP = SLIDE_WINDOW(2, MERGE(V1, V3, V5))
Crée six colonnes dont chaque ligne contient une fenêtre de données de trois par deux à partir des colonnes V1, V3 et V5. Les lignes 1 à 2 de l'entrée constituent la première ligne de la sortie, les lignes 2 à 3 constituent la deuxième ligne, etc.
TEMP = SLIDE_WINDOW(1, V1:V3, 2)
Crée trois colonnes, où la première ligne contient les données de V1[1]:V3, la deuxième ligne contient les données de V1[3]:V3, la troisième ligne contient les données de V1[5]:V3 et ainsi de suite (une ligne sur deux est ignorée).
TEMP = SLIDE_WINDOW(10, V1, 10)
Crée dix colonnes, où la première ligne contient les données de V1[1:10], la deuxième ligne contient les données de V1[11:20], la troisième ligne contient les données de V1[21:30], et ainsi de suite.
TEMP = SLIDE_WINDOW(3, V1:V2, 5)
Crée six colonnes, où chaque ligne contient une fenêtre de données de deux par trois des colonnes V1:V2. La première ligne contient les données de V1[1:3]:V2 ; la deuxième ligne contient les données de V1[6:8]:V2, la troisième ligne contient les données de V1[11:13]:V2 et ainsi de suite.

## Fonctions associées

Fonction	Description
GRID	Renvoie une grille de toutes les combinaisons de valeurs possibles (une par ligne)

---

## SORT



### Syntaxe

SORT(column [, keyword]) SORT(column, data [, keyword])

### Paramètres

column

Dans le premier format (data non fourni), il s'agit de la colonne de données à trier (numérique ou texte). Il peut s'agir d'une constante, d'une colonne, d'une plage de cibles à une colonne ou d'une expression convertie comme l'une de ces valeurs.

Cette plage de données ne peut pas contenir plus de  $2^{29}$  valeurs.

data

Si ce paramètre est fourni, il correspond aux données à trier en utilisant `column` comme critère de tri (`data` peut contenir des colonnes de données numériques et de texte). Le paramètre `data` peut correspondre à une colonne, une plage de cibles ou une expression convertie comme l'une de ces valeurs. Pour la définition du format de `data`, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM. Cette plage de données ne peut pas contenir plus de 2<sup>29</sup> lignes.

`keyword`

Ce mot clé facultatif détermine si les valeurs sont triées suivant l'ordre croissant (minimum vers maximum) ou décroissant (maximum vers minimum). Sélectionnez l'une des valeurs suivantes :

ASCEND - Trie `data` dans l'ordre croissant (par défaut)

DESCEND - Trie `data` dans l'ordre décroissant

Pour plus de détails sur l'utilisation des mots clés dans IBM Campaign, voir «Spécifications de format», à la page 9.

Pour plus de détails sur l'utilisation des mots clés dans IBM PredictiveInsight, voir «Spécifications de format», à la page 27.

## Description

`SORT` trie les valeurs de la plage de données spécifiée (`column` ou `data` basé sur `column`) en fonction de la valeur `keyword` (ASCEND ou DESCEND). Il renvoie une nouvelle colonne pour chaque colonne d'entrée à trier. Si seul `column` est fourni, les valeurs de `column` sont triées suivant l'ordre croissant ou décroissant spécifié par le paramètre `keyword`. Pour une colonne de texte, l'ordre croissant est l'ordre alphabétique (a-z) et l'ordre décroissant correspond à l'ordre inverse (z-a). Si `data` est également fourni, il est trié en utilisant `column` comme critère de tri.

**Remarque :** Si une plage de cibles à une colonne est fournie pour `column`, pour trier les lignes *correspondantes* de `data`, vous devez indiquer la même plage de cibles pour `data`. Sinon, par défaut, les *n* premières lignes de `data` sont triées. Par exemple, pour trier les lignes correspondantes, spécifiez : `TEMP = SORT(V1[100:200], V2[100:200]:V5)` Sinon, `TEMP = SORT(V1[100:200], V2:V5)` est équivalent à : `TEMP = SORT(V1[100:200], V2[1:101]:V5)`

## Exemples

<pre>TEMP = SORT(COLUMN(5, 3, 2, 4, 1)) ou TEMP = SORT(COLUMN(5, 3, 2, 4, 1), ASCEND)</pre>
---

Crée une colonne nommée TEMP contenant les valeurs 1, 2, 3, 4 et 5.

<pre>TEMP = SORT(COLUMN("b", "c", "a"))</pre>
---

Crée une colonne nommée TEMP contenant les chaînes a, b et c.

<pre>TEMP = SORT(10...15, DESCEND)</pre>
--

Crée une colonne nommée TEMP contenant les valeurs 15, 14, 13, 12, 11 et 10.

<pre>TEMP = SORT(V1)</pre>
----------------------------

Crée une colonne nommée TEMP contenant les valeurs de la colonne V1 triées dans l'ordre croissant.



<p>TEMP = SORT(V1, V1:V3)</p> <p>Crée trois colonnes nommées TEMP, VX et VY. Les valeurs de la colonne TEMP correspondent au contenu trié de la colonne V1, dans l'ordre croissant. Les valeurs de la colonne VX sont égales au contenu correspondant de la colonne V2 et les valeurs de la colonne VY sont égales au contenu correspondant de la colonne V3.</p>
<p>TEMP = SORT(V1[10:20], DESCEND)</p> <p>Crée une colonne nommée TEMP, dans laquelle les 11 premières cibles contiennent les valeurs triées des cibles des lignes 10 à 20 de la colonne V1, dans l'ordre décroissant. Les autres cibles de TEMP sont vides.</p>
<p>TEMP = SORT(V1[5:10], V2) ou TEMP = SORT(V1[5:10], V2[1:6])</p> <p>Crée une colonne nommée TEMP, dans laquelle les 6 premières cibles contiennent les valeurs des lignes 1 à 6 de la colonne V2, triées dans l'ordre décroissant des cibles 5 à 10 de la colonne V1. Les autres cibles de TEMP sont vides.</p>
<p>TEMP = SORT(V1[5:10], V2[5:10])</p> <p>Crée une colonne nommée TEMP, dans laquelle les 6 premières cibles contiennent les valeurs triées des cibles des lignes 5 à 10 de la colonne V2 dans l'ordre décroissant des cibles 5 à 10 de la colonne V1. Les autres cibles de TEMP sont vides.</p>
<p>TEMP = SORT(V1[10:50], V2:V3) ou TEMP = SORT(V1[10:50], V2[1:41]:V3)</p> <p>Crée deux colonnes nommées TEMP et VX, contenant des valeurs dans les lignes 1 à 41 (les autres cibles sont vides). Les valeurs de la colonne TEMP correspondent aux valeurs des lignes 1 à 41 de la colonne V2, triées en fonction des lignes 10 à 50 de la colonne V1. De même, les valeurs de la colonne VX correspondent aux valeurs des lignes 1 à 41 de la colonne V3, triées en fonction des lignes 10 à 50 de la colonne V1. La colonne V1 est triée dans l'ordre croissant.</p>

---

## SQRT

### Syntaxe

SQRT(data)

Paramètres

data

Valeurs numériques dont la racine carrée doit être calculée. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de data, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

### Description

SQRT calcule la racine carrée des valeurs de la plage de données spécifiée. Il renvoie une nouvelle colonne par colonne d'entrée, chacune contenant la racine carrée positive des nombres de la colonne d'entrée correspondante.

**Remarque :** Si l'une des valeurs de la plage de données définie est négative, ??? est renvoyé pour cette cible.

## Exemples

TEMP = SQRT(2)
Crée une colonne nommée TEMP contenant la valeur 1.41.
TEMP = SQRT(V1)
Crée une colonne nommée TEMP, dans laquelle chaque valeur correspond à la racine carrée du contenu de la colonne V1.
TEMP = SQRT(V1:V3)
Crée trois colonnes nommées TEMP, VX et VY. Les valeurs de la colonne TEMP correspondent aux racines carrées du contenu de la colonne V1, les valeurs de la colonne VX correspondent aux racines carrées du contenu de la colonne V2 et les valeurs de la colonne VY correspondent aux racines carrées du contenu de la colonne V3.
TEMP = SQRT(V1[10:20])
Crée une colonne nommée TEMP, dans laquelle les 11 premières cibles contiennent les racines carrées des valeurs des lignes 10 à 20 de la colonne V1. Les autres cibles de TEMP sont vides.
TEMP = SQRT(V1[10:50]:V2)
Crée deux colonnes nommées TEMP et VX, contenant des valeurs dans les lignes 1 à 41 (les autres cibles sont vides). Les valeurs de la colonne TEMP correspondent aux racines carrées des valeurs des lignes 10 à 50 de la colonne V1 et les valeurs de la colonne VX correspondent aux racines carrées des valeurs des lignes 10 à 50 de la colonne V2.

## Fonctions associées

Fonction	Description
DIV	Divise une plage de données spécifiée par une autre
MULT	Multiplie le contenu de deux plages de données
POW	Calcule une valeur de base élevée aux puissances exponentielles spécifiées

---

## STAT



### Syntaxe

STAT(data [, keyword])

### Paramètres

data

Valeurs numériques dont les moments doivent être calculés (moyenne, écart type, asymétrie et kurtosis). Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de data, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM. data doit comporter au moins trois valeurs.

keyword

Ce mot clé facultatif détermine la manière dont le calcul est effectué sur la plage de données d'entrée. Sélectionnez l'une des valeurs suivantes :

ALL - Effectue le calcul sur toutes les cibles de data (valeur par défaut)

COL - Effectue le calcul séparément pour chaque colonne de data

ROW - Effectue le calcul séparément pour chaque ligne de data

Pour plus de détails sur l'utilisation des mots clés dans IBM Campaign, voir «Spécifications de format», à la page 9.

Pour plus de détails sur l'utilisation des mots clés dans IBM PredictiveInsight, voir «Spécifications de format», à la page 27.

## Description

STAT calcule les quatre premiers moments des valeurs de la plage de données spécifiée. Le premier moment est la moyenne. Le deuxième moment est l'écart-type. Le troisième moment est l'asymétrie et le quatrième moment est le kurtosis.

Le nombre de colonnes renvoyé par la fonction macro STAT dépend de keyword et du nombre de colonnes de data.

- Si le mot clé ALL est utilisé (par défaut), les moments sont calculés sur toutes les valeurs de data. Une colonne contenant quatre valeurs est renvoyée.
- Si le mot clé COL est utilisé, les moments sont calculés séparément pour chaque colonne d'entrée. Une colonne est renvoyée par colonne d'entrée, chacune contenant quatre valeurs.
- Si le mot clé ROW est utilisé, les moments sont calculés pour chaque ligne de data. STAT renvoie quatre colonnes. Les moments sont répertoriés sur chaque ligne pour chaque ligne de la plage de données d'entrée.

## Exemples

<code>TEMP = STAT(MERGE(1, 2, 3, 4, 5))</code> ou <code>TEMP = STAT(MERGE(1, 2, 3, 4, 5), ALL)</code>
Crée une colonne nommée TEMP contenant les valeurs 3, 1.58, 0 et -1.2.
<code>TEMP = STAT(V1)</code>
Crée une colonne nommée TEMP contenant les quatre premiers moments de la colonne V1.
<code>TEMP = STAT(V1:V3)</code>
Crée une colonne nommée TEMP contenant les quatre premiers moments des colonnes V1, V2 et V3.
<code>TEMP = STAT(V1[10:20])</code>
Crée une colonne nommée TEMP contenant les quatre premiers moments des cibles des lignes 10 à 20 de la colonne V1.
<code>TEMP = STAT(V1[1:5]:V4)</code>
Crée une colonne nommée TEMP contenant les quatre premiers moments des cibles des lignes 1 à 5 des colonnes V1 à V4.

<p>TEMP = STAT(V1:V3, COL)</p> <p>Crée trois colonnes nommées TEMP, VX et VY. Les quatre valeurs de la colonne TEMP correspondent aux moments de la colonne V1, les quatre valeurs de la colonne VX correspondent aux moments de la colonne V2 et les quatre valeurs de la colonne VY correspondent aux moments de la colonne V3.</p>
<p>TEMP = STAT(V1[1:5]:V3, COL) ou TEMP = STAT(V1[1:5]:V3[1:5], COL)</p> <p>Crée trois colonnes nommées TEMP, VX et VY contenant chacune quatre valeurs. Les valeurs de la colonne TEMP correspondent aux moments des cibles des lignes 1 à 5 de la colonne V1, les valeurs de la colonne VX correspondent aux moments des cibles des lignes 1 à 5 de la colonne V2, et les valeurs de la colonne VY correspondent aux moments des cibles des lignes 1 à 5 de la colonne V3.</p>
<p>TEMP = STAT(V1:V3, ROW)</p> <p>Crée quatre colonnes nommées TEMP, VX, VY et VZ. TEMP contient la moyenne de chaque ligne des colonnes V1, V2 et V3, VX contient l'écart type, VY contient l'asymétrie et VZ contient le kurtosis.</p>
<p>TEMP = STAT(V1[50:100]:V3, ROW) ou TEMP = STAT(V1[50:100]:V3[50:100], ROW)</p> <p>Crée quatre colonnes nommées TEMP, VX, VY et VZ, contenant chacune 51 lignes. TEMP contient la moyenne, VX contient l'écart type, VY contient l'asymétrie et VZ contient le kurtosis. La première ligne correspond à la ligne 50 des colonnes V1, V2 et V3. La seconde ligne correspond à la ligne 51, et ainsi de suite.</p>

## Fonctions associées

Fonction	Description
AVG ou MEAN	Calcule la moyenne arithmétique d'une plage de cibles
KURTOSIS	Calcule le kurtosis d'une plage de cibles
SKEW	Calcule l'asymétrie de la distribution d'une plage de cibles
STDV ou STDEV	Calcule l'écart type d'une plage de cibles
VARIANCE	Calcule la variance d'une plage de cibles

## STDV ou STDEV

### Syntaxe

STDV(data [, keyword]) STDEV(data [, keyword])

### Paramètres

data

Valeurs numériques dont l'écart type doit être calculé. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de data, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

keyword

Ce mot clé facultatif détermine la manière dont le calcul est effectué sur la plage de données d'entrée. Sélectionnez l'une des valeurs suivantes :

ALL - Effectue le calcul sur toutes les cibles de data (valeur par défaut)

COL - Effectue le calcul séparément pour chaque colonne de data

ROW - Effectue le calcul séparément pour chaque ligne de data

Pour plus de détails sur l'utilisation des mots clés dans IBM Campaign, voir «Spécifications de format», à la page 9.

Pour plus de détails sur l'utilisation des mots clés dans IBM PredictiveInsight, voir «Spécifications de format», à la page 27.

**Remarque :** De nombreuses fonctions de macro utilisent les paramètres de mot clé {ALL | COL | ROW}. Ces mots clés ne s'appliquent pas dans **IBM Campaign** car les données d'entrées correspondent toujours à une colonne ou à une zone. La macro se comporte toujours comme si le mot clé COL a été spécifié. Par conséquent, vous n'avez pas besoin de spécifier ces mots clés lorsque vous utilisez **IBM Campaign**.

## Description

STDV calcule l'écart type de toutes les cibles de la plage de données spécifiée. L'écart type d'une distribution est égal à la racine carrée de la variance. L'écart type est calculé comme suit :

$$\sqrt{\frac{1}{n-1} \sum_{j=1}^n (x_j - \text{mean})^2}$$

où les  $x$  représentent les échantillons,  $n$  représente le nombre d'échantillons et  $mean$  est la moyenne de la distribution.

**Remarque :** Si le nombre d'échantillons  $n = 1$ , STDV renvoie une erreur.

Cette macro est disponible dans IBM Interact.

## Exemples

```
TEMP = STDV(MERGE(1, 2, 1, 0)) ou TEMP = STDEV(MERGE(1, 2, 1, 0))
```

Crée une colonne nommée TEMP contenant la valeur 0.71.

```
TEMP = STDV(V1)
```

Crée une colonne nommée TEMP contenant une valeur unique qui correspond à l'écart type du contenu de la colonne V1.

TEMP = STDV(V1:V3)
Crée une colonne nommée TEMP contenant une valeur unique qui correspond à l'écart type du contenu des colonnes V1, V2 et V3.
TEMP = STDV(V1[1:5]:V4)
Crée une colonne nommée TEMP contenant une valeur unique qui correspond à l'écart type des cibles des lignes 1 à 5 des colonnes V1 à V4.
TEMP = STDV(V1:V3, COL)
Crée trois colonnes nommées TEMP, VX et VY. La valeur unique de la colonne TEMP correspond à l'écart type du contenu de la colonne V1, la valeur unique de la colonne VX correspond à l'écart type du contenu de la colonne V2 et la valeur unique de la colonne VY correspond à l'écart type du contenu de la colonne V3.
TEMP = STDV(V1[10:50]:V3,COL)
Crée trois colonnes nommées TEMP, VX et VY contenant chacune une seule valeur. La valeur de la colonne TEMP correspond à l'écart type des cibles des lignes 10 à 50 de la colonne V1, la valeur de la colonne VX correspond à l'écart type des cibles des lignes 10 à 50 de la colonne V2 et la valeur de la colonne VY correspond à l'écart type des cibles des lignes 10 à 50 de la colonne V3.
TEMP = STDV(V1:V3, ROW)
Crée une colonne nommée TEMP dans laquelle chaque entrée de cible correspond à l'écart type de la ligne correspondante dans les colonnes V1, V2 et V3.
TEMP = STDV(V1[1:5]:V3,ROW)
Crée une colonne nommée TEMP, dans laquelle les cibles des lignes 1 à 5 contiennent l'écart type de la ligne correspondante dans les colonnes V1 à V3. Les autres cibles de TEMP sont vides.

## Fonctions associées

Fonction	Description
KURTOSIS	Calcule le kurtosis d'une plage de cibles
SKEW	Calcule l'asymétrie de la distribution d'une plage de cibles
STAT	Calcule les premier, deuxième, troisième et quatrième moments de la plage de données spécifiée
VAR	Calcule la variance d'une plage de cibles

---

## STRING\_CONCAT

### Syntaxe

STRING\_CONCAT(string1, string2, ... stringN)

### Paramètres

string

Chaîne de texte ASCII à concaténer. Il peut s'agir de texte ASCII entre guillemets, d'une colonne de texte, d'une plage de cibles contenant du texte ou d'une

expression convertie comme l'une de ces valeurs. Pour la définition du format de string (identique à data), voir la section *Paramètres des fonctions macro* du chapitre *Utilisation de macros* de votre produit.

## Description

STRING\_CONCAT concatène les valeurs de texte ASCII de la plage de données spécifiée. Il renvoie une nouvelle colonne par colonne d'entrée, chacune contenant les chaînes concaténées des lignes correspondantes de strings. Ce calcul ligne par ligne génère un résultat pour chaque ligne jusqu'à la dernière valeur de la colonne la plus courte.

**Remarque :** La longueur totale de chaque chaîne résultante ne peut pas dépasser 255 caractères.

Cette macro est disponible dans IBM Interact.

IBM Interact prend également en charge la syntaxe suivante :

```
STRING_CONCAT( string1 , string2 , ... stringN )
```

Par exemple, STRING\_CONCAT('a', 'b', 'c', 'd') est valide.

## Exemples

<pre>TEMP = STRING_CONCAT("house", "boat")</pre>
Crée une colonne nommée TEMP, qui contient la chaîne de texte ASCII " houseboat ".
<pre>TEMP = STRING_CONCAT(V1, ".")</pre>
Crée une colonne nommée TEMP, chacune de ses lignes contenant la chaîne de texte ASCII de la ligne correspondante de la colonne V1 à laquelle un point a été ajouté.
<pre>TEMP = STRING_CONCAT(V1, V2)</pre>
Crée une colonne nommée TEMP, chacune de ses lignes contenant la chaîne de texte ASCII de la colonne V1 concaténée avec la chaîne de texte de la colonne V2.
<pre>TEMP = STRING_CONCAT(V1:V3, V4:V6)</pre>
Crée trois colonnes nommées TEMP, VX et VY. Les valeurs de la colonne TEMP correspondent aux chaînes concaténées des lignes correspondantes des colonnes V1 et V4, les valeurs de la colonne VX correspondent aux chaînes concaténées des lignes correspondantes des colonnes V2 et V5 et les valeurs de la colonne VY correspondent aux chaînes concaténées des lignes correspondantes des colonnes V3 et V6.
<pre>TEMP = STRING_CONCAT(V1[5:10]:V2, V3:V4)</pre>
Crée deux colonnes nommées TEMP et VX. Les valeurs de la colonne TEMP correspondent aux chaînes des lignes 5 à 10 de la colonne V1 concaténées avec les lignes 1 à 6 de la colonne V3. Les valeurs de VX correspondent aux chaînes des lignes 5 à 10 de la colonne V2 concaténées avec les lignes 1 à 6 de la colonne V4.
<pre>TEMP = STRING_CONCAT('a', 'b', 'c', 'd')</pre>
Crée une colonne nommée TEMP, qui contient la chaîne de texte ASCII " abcd".

## Fonctions associées

Fonction	Description
STRING_HEAD	Renvoie les n premiers caractères de chaque chaîne de la plage de données spécifiée
STRING_LENGTH	Renvoie la longueur de chaque chaîne de la plage de données spécifiée
STRING_SEG	Renvoie le segment de chaîne entre deux index spécifiés
STRING_TAIL	Renvoie les n derniers caractères de chaque chaîne de la plage de données spécifiée

---

## STRING\_HEAD

### Syntaxe

STRING\_HEAD(num\_chars, data)

### Paramètres

num\_chars

Nombre de caractères à renvoyer à partir du début de chaque chaîne de data. Il doit s'agir d'un entier positif supérieur à zéro.

data

Valeurs de chaîne de texte ASCII. Il peut s'agir de texte ASCII entre guillemets, d'une colonne de texte, d'une plage de cibles contenant du texte ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de data, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

### Description

STRING\_HEAD renvoie les num\_chars premiers caractères de chaque valeur de chaîne de la plage de données spécifiée. Si num\_chars est supérieur au nombre de caractères d'une chaîne de texte, les caractères restants sont remplacés par le caractère null " \0 ".

### Exemples

TEMP = STRING_HEAD(3, "JAN 15, 1997")
Crée une colonne nommée TEMP, qui contient la chaîne de texte ASCII " JAN ".
TEMP = STRING_HEAD(10, "Pressure")
Crée une colonne nommée TEMP, qui contient la chaîne de texte ASCII " Pressure ".
TEMP = STRING_HEAD(5, V1)
Crée une colonne nommée TEMP contenant les cinq premiers caractères de chaque chaîne de la colonne V1.



```
TEMP = STRING_HEAD(1, V1:V3)
```

Crée trois colonnes nommées TEMP, VX et VY. Les valeurs de la colonne TEMP correspondent aux premiers caractères des chaînes des lignes correspondantes de la colonne V1, les valeurs de la colonne VX correspondent aux premiers caractères des chaînes des lignes correspondantes de la colonne V2 et les valeurs de la colonne VY correspondent aux premiers caractères des chaînes des lignes correspondantes de la colonne V3.

```
TEMP = STRING_HEAD(12, V4[1:50]:V6)
```

Crée trois colonnes nommées TEMP, VX et VY. Les valeurs de la colonne TEMP correspondent aux 12 premiers caractères des chaînes des lignes 1 à 50 de la colonne V1, les valeurs de la colonne VX correspondent aux 12 premiers caractères des chaînes des lignes 1 à 50 de la colonne V2 et les valeurs de la colonne VY correspondent aux 12 premiers caractères des chaînes des lignes 1 à 50 de la colonne V3.

## Fonctions associées

Fonction	Description
STRING_CONCAT	Concatène deux chaînes de texte des plages de données spécifiées
STRING_LENGTH	Renvoie la longueur de chaque chaîne de la plage de données spécifiée
STRING_SEG	Renvoie le segment de chaîne entre deux index spécifiés
STRING_TAIL	Renvoie les n derniers caractères de chaque chaîne de la plage de données spécifiée

---

## STRING\_LENGTH

### Syntaxe

```
STRING_LENGTH(data)
```

Paramètres

data

Valeurs de chaîne de texte ASCII dont la longueur doit être calculée. Il peut s'agir de texte ASCII entre guillemets, d'une colonne de texte, d'une plage de cibles contenant du texte ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de data, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

### Description

STRING\_LENGTH renvoie la longueur de chaque valeur de chaîne dans la plage de données spécifiée. Il renvoie une nouvelle colonne par colonne d'entrée, chacune contenant la longueur de la chaîne de texte correspondante.

**Remarque :** Si STRING\_LENGTH est appliqué à des colonnes contenant des données numériques, il renvoie des zéros.

## Exemples

<pre>TEMP = STRING_LENGTH("four")</pre>
Crée une colonne nommée TEMP contenant la valeur 4.
<pre>TEMP = STRING_LENGTH(4)</pre>
Crée une colonne nommée TEMP contenant la valeur 0.
<pre>TEMP = STRING_LENGTH(V1)</pre>
Crée une colonne nommée TEMP, dans laquelle chaque valeur correspond à la longueur de la chaîne dans la ligne correspondante de la colonne V1.
<pre>TEMP = STRING_LENGTH(V1:V3)</pre>
Crée trois colonnes nommées TEMP, VX et VY. Les valeurs de la colonne TEMP correspondent aux longueurs des chaînes des lignes correspondantes de la colonne V1, les valeurs de la colonne VX correspondent aux longueurs des chaînes des lignes correspondantes de la colonne V2 et les valeurs de la colonne VY correspondent aux longueurs des chaînes des lignes correspondantes de la colonne V3.
<pre>TEMP = STRING_LENGTH(V4[1:50]:V6]</pre>
Crée trois colonnes nommées TEMP, VX et VY. Les valeurs de la colonne TEMP correspondent aux longueurs des chaînes des lignes 1 à 50 de la colonne V1, les valeurs de la colonne VX correspondent aux longueurs des chaînes des lignes 1 à 50 de la colonne V2 et les valeurs de la colonne VY correspondent aux longueurs des chaînes des lignes 1 à 50 de la colonne V3.

## Fonctions associées

Fonction	Description
STRING_CONCAT	Concatène deux chaînes de texte de plages de données spécifiées
STRING_HEAD	Renvoie les n premiers caractères de chaque chaîne de la plage de données spécifiée
STRING_SEG	Renvoie le segment de chaîne entre deux index spécifiés
STRING_TAIL	Renvoie les n derniers caractères de chaque chaîne de la plage de données spécifiée

---

## STRING\_PROPER

### Syntaxe

STRING\_PROPER(data)

### Paramètres

data

Valeur de chaîne à convertir.

### Description

STRING\_PROPER convertit chaque valeur de chaîne contenue dans la plage de données spécifiée en mettant en majuscule la première lettre de chaque chaîne ou

toute lettre suivant un espace blanc ou un symbole (autre que le trait de soulignement) et transforme en minuscules tous les autres caractères. Il renvoie une nouvelle colonne par colonne d'entrée, chacune contenant la chaîne convertie dans la colonne d'entrée correspondante.

## Exemples

```
Temp = STRING_PROPER
```

---

## STRING\_SEG

### Syntaxe

```
STRING_SEG(from, to, data)
```

### Paramètres

from

Nombre de caractères depuis le début de la chaîne à partir duquel le segment de chaîne doit commencer à être extrait. Il doit s'agir d'un entier positif supérieur à zéro et inférieur à to ou STRING\_SEG renvoie une chaîne vide.

to

Nombre de caractères depuis le début de la chaîne à partir duquel le segment de chaîne ne doit plus être extrait. Il doit s'agir d'un entier positif supérieur ou égal à from. Si to est égal à from (et to est inférieur ou égal à la longueur de la chaîne), un caractère est renvoyé.

data

Valeurs de chaîne de texte ASCII. Il peut s'agir de texte ASCII entre guillemets, d'une colonne de texte, d'une plage de cibles contenant du texte ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de data, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

### Description

STRING\_SEG renvoie le segment de chaîne entre deux index de chaque valeur de chaîne dans la plage de données spécifiée. Si from est supérieur à la longueur d'une chaîne, aucune valeur n'est renvoyée. Si to est supérieur à la longueur d'une chaîne, tous les caractères de from sont renvoyés.

### Exemples

```
TEMP = STRING_SEG(1, 6, "JAN 15, 1997")
```

Crée une colonne nommée TEMP, qui contient la chaîne de texte ASCII " Jan 15 ".

```
TEMP = STRING_SEG(5, 20, "Pressure")
```

Crée une colonne nommée TEMP, qui contient la chaîne de texte ASCII " sure ".

<pre>TEMP = STRING_SEG(5, 6, V1)</pre> <p>Crée une colonne nommée TEMP contenant les cinquième et sixième caractères de chaque chaîne de la colonne V1.</p>
<pre>TEMP = STRING_SEG(10, 20, V1:V3)</pre> <p>Crée trois colonnes nommées TEMP, VX et VY. Les valeurs de la colonne TEMP correspondent aux caractères 10 à 20 des chaînes des lignes correspondantes de la colonne V1, les valeurs de la colonne VX correspondent aux caractères 10 à 20 des chaînes des lignes correspondantes de la colonne V2 et les valeurs de la colonne VY correspondent aux caractères 10 à 20 des chaînes des lignes correspondantes de la colonne V3.</p>
<pre>TEMP = STRING_SEG(5, 10, V4[1:50]:V6]</pre> <p>Crée trois colonnes nommées TEMP, VX et VY. Les valeurs de la colonne TEMP correspondent aux caractères 5 à 10 des chaînes des lignes 1 à 50 de la colonne V1, les valeurs de la colonne VX correspondent aux caractères 5 à 10 des chaînes des lignes 1 à 50 de la colonne V2 et les valeurs de la colonne VY correspondent aux caractères 5 à 10 des chaînes des lignes 1 à 50 de la colonne V3.</p>

## Fonctions associées

Fonction	Description
STRING_CONCAT	Concatène deux chaînes de texte des plages de données spécifiées
STRING_HEAD	Renvoie les n premiers caractères de chaque chaîne de la plage de données spécifiée
STRING_LENGTH	Renvoie la longueur de chaque chaîne de la plage de données spécifiée
STRING_TAIL	Renvoie les n derniers caractères de chaque chaîne de la plage de données spécifiée

---

## STRING\_TAIL

### Syntaxe

```
STRING_TAIL(num_chars, data)
```

### Paramètres

num\_chars

Nombre de caractères à renvoyer à partir de la fin de chaque chaîne de data. Il doit s'agir d'un entier positif supérieur à zéro.

data

Valeurs de chaîne de texte ASCII. Il peut s'agir de texte ASCII entre guillemets, d'une colonne de texte, d'une plage de cibles contenant du texte ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de data, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

## Description

STRING\_TAIL renvoie les num\_chars derniers caractères de chaque valeur de chaîne de la plage de données spécifiée. Toutes les valeurs de chaîne sont remplies jusqu'à la longueur de la plus longue chaîne avec des caractères nuls " \0 ". Ensuite, les num\_chars derniers caractères sont renvoyés de chaque chaîne. Si num\_chars est supérieur au nombre de caractères d'une chaîne de texte, l'intégralité de la chaîne de texte est renvoyée.

## Exemples

<pre>TEMP = STRING_TAIL(3, "JAN 15, 1997")</pre>
Crée une colonne nommée TEMP, qui contient la chaîne de texte ASCII " 997 ".
<pre>TEMP = STRING_TAIL(10, "Pressure")</pre>
Crée une colonne nommée TEMP, qui contient la chaîne de texte ASCII " Pressure ".
<pre>TEMP = STRING_TAIL(5, V1)</pre>
Crée une colonne nommée TEMP contenant les cinq derniers caractères de chaque chaîne de la colonne V1.
<pre>TEMP = STRING_TAIL(1, V1:V3)</pre>
Crée trois colonnes nommées TEMP, VX et VY. Les valeurs de la colonne TEMP correspondent aux derniers caractères des chaînes des lignes correspondantes de la colonne V1, les valeurs de la colonne VX correspondent aux derniers caractères des chaînes des lignes correspondantes de la colonne V2 et les valeurs de la colonne VY correspondent aux derniers caractères des chaînes des lignes correspondantes de la colonne V3.
<pre>TEMP = STRING_TAIL(12, V4[1:50]:V6]</pre>
Crée trois colonnes nommées TEMP, VX et VY. Les valeurs de la colonne TEMP correspondent aux 12 derniers caractères des chaînes des lignes 1 à 50 de la colonne V1, les valeurs de la colonne VX correspondent aux 12 derniers caractères des chaînes des lignes 1 à 50 de la colonne V2 et les valeurs de la colonne VY correspondent aux 12 derniers caractères des chaînes des lignes 1 à 50 de la colonne V3.

## Fonctions associées

Fonction	Description
STRING_CONCAT	Concatène deux chaînes de texte des plages de données spécifiées
STRING_HEAD	Renvoie les n premiers caractères de chaque chaîne de la plage de données spécifiée
STRING_LENGTH	Renvoie la longueur de chaque chaîne de la plage de données spécifiée
STRING_SEG	Renvoie le segment de chaîne entre deux index spécifiés

---

## SUBSAMPLE



## Syntaxe

SUBSAMPLE(num\_samples, data)

## Paramètres

num\_samples

Nombre d'échantillons à extraire. Il doit s'agir d'un entier positif inférieur au nombre de cibles dans la plage de données spécifiée (la fonction macro SUBSAMPLE ne peut pas être utilisée pour *augmenter* le nombre de points de données par réplication).

data

Valeurs d'où les échantillons doivent provenir. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de data, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

## Description

SUBSAMPLE échantillonne uniformément le nombre de points de données demandé à partir de la plage de données spécifiée. Il renvoie une nouvelle colonne par colonne d'entrée, chacune contenant num\_samples lignes de nombres extraites uniformément de la colonne d'entrée correspondante. La valeur de la première ligne et la valeur de chaque *ne* ligne qui suit sont renvoyées, de sorte qu'au total, num\_samples échantillons sont extraits.

**Remarque :** Cette fonction macro peut être utilisée pour augmenter ou diminuer le nombre d'échantillons.

## Exemples

```
TEMP = SUBSAMPLE(100, V1)
```

Crée une colonne nommée TEMP, qui contient 100 valeurs échantillonnées uniformément de la colonne V1.

```
TEMP = SUBSAMPLE(50, V1:V3)
```

Crée trois colonnes nommées TEMP, VX et VY contenant chacune 50 valeurs. Les valeurs de la colonne TEMP correspondent aux valeurs d'échantillon du contenu de la colonne V1, les valeurs de la colonne VX correspondent aux valeurs d'échantillon du contenu de la colonne V2 et les valeurs de la colonne VY correspondent aux valeurs d'échantillon du contenu de la colonne V3.

```
TEMP = SUBSAMPLE(5, V1[0:100])
```

Crée une colonne nommée TEMP avec les valeurs des cinq premières lignes. Les données sont échantillonnées uniformément à partir des lignes 0 à 100 de la colonne V1.

```
TEMP = SUBSAMPLE(250, V1[1:10]:V2)
```

Crée deux colonnes nommées TEMP et VX, contenant chacune des valeurs dans les 250 premières lignes. Les valeurs de la colonne TEMP sont échantillonnées uniformément à partir des lignes 1 à 400 de la colonne V1 et les valeurs de la colonne VX sont échantillonnées uniformément à partir des lignes 1 à 400 de la colonne V2.

## Fonctions associées

Fonction	Description
EXTRACT	Extrait les lignes contenant les valeurs d'une colonne de prédicat
SAMPLE_RANDOM	Renvoie des colonnes de n cibles, contenant chacune un échantillon aléatoire de la plage de données spécifiée

---

## SUBSTITUTE



### Syntaxe

SUBSTITUTE(data, from\_table, to\_table)

### Paramètres

data

Valeurs numériques ou de chaîne à convertir. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de data, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

from\_table

Colonne contenant les valeurs à convertir. La longueur de la colonne from\_table doit être identique à celle de la colonne to\_table.

to\_table

Colonne contenant les valeurs vers lesquelles la conversion doit être effectuée. La longueur de la colonne to\_table doit être identique à celle de la colonne from\_table.

### Description

SUBSTITUTE convertit les valeurs de data à l'aide des paires de substitution spécifiées dans from\_table et to\_table. Chaque valeur de data détectée dans from\_table est remplacée par la valeur de la ligne correspondante de to\_table.

SUBSTITUTE peut être utilisé pour changer à la fois les valeurs numériques et les valeurs de chaîne. Il renvoie toujours une plage de données avec les mêmes dimensions que data.

**Remarque :** Si vous utilisez SUBSTITUTE pour convertir des chaînes en nombres ou vice versa, *toutes* les valeurs de data doivent être converties. A savoir, toutes les valeurs de data doivent apparaître dans from\_table. Sinon, la ligne résultante contient ???.

## Exemples

<pre>TEMP = SUBSTITUTE(COLUMN(1,5,10),COLUMN(1), COLUMN(7))</pre>
Crée une colonne nommée TEMP, qui contient les valeurs 7, 5, 10 (la valeur 1 est remplacée par 7).
<pre>TEMP = SUBSTITUTE(COLUMN("blue", "red"), COLUMN("blue", "red"), COLUMN(0, 1))</pre>
Crée une colonne nommée TEMP, qui contient les valeurs 0 et 1 (la chaîne "blue" est remplacée par 0 et "red", par 1).
<pre>TEMP = SUBSTITUTE(V1, V2, V3)</pre>
Crée une colonne nommée TEMP contenant les valeurs de la colonne V1, dans laquelle les valeurs de la colonne V2 sont remplacées par la valeur de la ligne correspondante de la colonne V3.
<pre>TEMP = SUBSTITUTE(V1:V2, V4, V5)</pre>
Crée deux colonnes nommées TEMP et VX, contenant chacune les valeurs des colonnes V1 et V2 respectivement, où les valeurs détectées dans la colonne V4 sont remplacées par la valeur de la ligne correspondante de la colonne V5.
<pre>TEMP = SUBSTITUTE(V1[10:20]:V2, V4, V5)</pre>
Crée deux colonnes nommées TEMP et VX, contenant chacune les valeurs des lignes 10 à 20 des colonnes V1 et V2 respectivement, où les valeurs détectées dans la colonne V4 sont remplacées par la valeur de la ligne correspondante de la colonne V5.

## Fonctions associées

Fonction	Description
EXTRACT	Extrait les lignes contenant les valeurs d'une colonne de prédicat
ISMEMBER	Teste une plage en entrée par rapport à une "table" de valeurs et renvoie un si une valeur se trouve dans la table et zéro dans le cas contraire

---

## SUBSTR ou SUBSTRING

### Syntaxe

```
SUBSTR(string_value, start_pos[, nchars]) or SUBSTR(string_value FROM  
start_pos[ FOR nchars]) SUBSTRING(string_value, start_pos[, nchars]) or  
SUBSTRING(string_value FROM start_pos[ FOR nchars])
```

### Paramètres

*string\_value*

Chaîne à partir de laquelle une sous-chaîne est extraite.

*start\_pos*

Caractère de début à partir duquel la sous-chaîne doit être extraite.

*nchars*



Nombre de caractères à extraire (doit être supérieur ou égal à 0). Si cette valeur n'est pas fournie, tous les caractères restants de `string_value` sont extraits.

## Description

SUBSTR ou SUBSTRING extrait `nchars` caractères de la chaîne, à partir de `start_pos`. Si `nchars` est omis, SUBSTR et SUBSTRING extraient les caractères de `start_pos` jusqu'à la fin de la chaîne. Les espaces de fin sont automatiquement tronqués.

Cette macro est disponible dans IBM Interact.

**Important :** IBM Interact ne prend en charge que les formats suivants : SUBSTR(`string_value`, `start_pos`[, `nchars`]) ou SUBSTRING(`string_value`, `start_pos`[, `nchars`])

## Exemples

SUBSTR SUBSTR Renvoie	("abcdef" FROM 1 FOR 2) ("abcdef",1,2) 'ab'
SUBSTR SUBSTR Renvoie	("abcdef" FROM -2 FOR 4) ("abcdef",-2,4) 'a'
SUBSTR SUBSTR Renvoie	("abcdef" FROM 3) ("abcdef",3) 'cdef'

---

## SUM

### Syntaxe

SUM(`data` [, `keyword`])

### Paramètres

`data`

Valeurs numériques dont la somme doit être calculée. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de `data`, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

`keyword`

Ce mot clé facultatif détermine la manière dont le calcul est effectué sur la plage de données d'entrée. Sélectionnez l'une des valeurs suivantes :

ALL - Effectue le calcul sur toutes les cibles de `data` (valeur par défaut)

COL - Effectue le calcul séparément pour chaque colonne de `data`

ROW - Effectue le calcul séparément pour chaque ligne de `data`

Pour plus de détails sur l'utilisation des mots clés dans IBM Campaign, voir «Spécifications de format», à la page 9.

Pour plus de détails sur l'utilisation des mots clés dans IBM PredictiveInsight, voir «Spécifications de format», à la page 27.

**Remarque :** De nombreuses fonctions de macro utilisent les paramètres de mot clé {ALL | COL | ROW}. Ces mots clés ne s'appliquent pas dans **IBM Campaign** car les données d'entrées correspondent toujours à une colonne ou à une zone. La macro se comporte toujours comme si le mot clé COL a été spécifié. Par conséquent, vous n'avez pas besoin de spécifier ces mots clés lorsque vous utilisez **IBM Campaign**.

## Description

SUM calcule la somme de toutes les cibles de la plage de données spécifiée. Il renvoie une colonne unique.

**Remarque :** SUM est identique à la fonction macro TOTAL.

Cette macro est disponible dans IBM Interact.

## Exemples

TEMP = SUM(3)	Crée une colonne nommée TEMP contenant la valeur trois.
TEMP = SUM((COLUMN(3, 5, 1)))	Crée une colonne nommée TEMP contenant la valeur neuf.
TEMP = SUM(V1)	Crée une colonne nommée TEMP contenant une valeur unique qui correspond à la somme du contenu de la colonne V1.
TEMP = SUM(V1:V3)	Crée une colonne nommée TEMP contenant une valeur unique qui correspond à la somme du contenu des colonnes V1, V2 et V3.
TEMP = SUM(V1[1:5]:V4)	Crée une colonne nommée TEMP contenant une valeur unique qui correspond à la somme des cibles des lignes 10 à 20 des colonnes V1 à V4.
TEMP = SUM(V1:V3, COL)	Crée trois colonnes nommées TEMP, VX et VY. La valeur unique de la colonne TEMP correspond à la somme du contenu de la colonne V1, la valeur unique de la colonne VX correspond à la somme du contenu de la colonne V2 et la valeur unique de la colonne VY correspond à la somme du contenu de la colonne V3.
TEMP = SUM(V1[1:5]:V3, COL)	Crée trois colonnes nommées TEMP, VX et VY contenant chacune une seule valeur. La valeur de la colonne TEMP correspond à la somme des cibles des lignes 1 à 5 de la colonne V1, la valeur de la colonne VX correspond à la somme des cibles des lignes 1 à 5 de la colonne V2 et la valeur de la colonne VY correspond à la somme des cibles des lignes 1 à 5 de la colonne V3.
TEMP = SUM(V1:V3, ROW)	Crée une colonne nommée TEMP dans laquelle chaque entrée de cible correspond à la somme de la ligne correspondante dans les colonnes V1, V2 et V3.
TEMP = SUM(V1[1:5]:V3, ROW)	Crée une colonne nommée TEMP, dans laquelle les cibles des lignes 1 à 5 contiennent la somme de la ligne correspondante dans les colonnes V1 à V3. Les autres cibles de TEMP sont vides.

## Fonctions associées

Fonction	Description
AVG ou MEAN	Calcule la moyenne arithmétique d'une plage de cibles
AVG_DEV	Calcule l'écart moyen d'une plage de cibles

---

## TAN

### Syntaxe

TAN(data [, units\_keyword])

### Paramètres

data

Valeurs numériques dont la tangente doit être calculée. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de data, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

units\_keyword

Ce mot clé facultatif détermine si les valeurs en entrée et les résultats sont interprétés comme des degrés ou des radians. Sélectionnez l'une des valeurs suivantes :

RADIAN - Effectue les calculs en radians (valeur par défaut)

DEGREE - Effectue les calculs en degrés

Si ce paramètre n'est pas spécifié, la valeur par défaut est radians. (Pour convertir des radians en degrés, divisez par PI et multipliez par 180.)

Pour plus de détails sur l'utilisation des mots clés dans IBM Campaign, voir «Spécifications de format», à la page 9.

Pour plus de détails sur l'utilisation des mots clés dans IBM PredictiveInsight, voir «Spécifications de format», à la page 27.

### Description

TAN calcule la tangente des valeurs de la plage de données spécifiée. Il renvoie une nouvelle colonne par colonne d'entrée, chacune contenant la tangente des nombres de la colonne d'entrée correspondante.

### Exemples

TEMP = TAN(PI/4), TEMP = TAN(PI/4, 0) ou TEMP = TAN(PI/4, RADIAN)
---

Crée une colonne nommée TEMP contenant la valeur un.
--

TEMP = TAN(V1)
Crée une colonne nommée TEMP, où chaque valeur correspond à la tangente (en radians) du contenu de la colonne V1.
TEMP = TAN(V1:V3, 1) ou TEMP = TAN(V1:V3, DEGREE)
Crée trois colonnes nommées TEMP, VX et VY. Les valeurs de la colonne TEMP correspondent aux tangentes du contenu de la colonne V1, les valeurs de la colonne VX correspondent aux tangentes du contenu de la colonne V2 et les valeurs de la colonne VY correspondent aux tangentes du contenu de la colonne V3. Toutes les valeurs sont en degrés.
TEMP = TAN(V1[1:5]:V2)
Crée deux colonnes nommées TEMP et VX, contenant des valeurs dans les lignes 1 à 5 (les autres cibles sont vides). Les valeurs de la colonne TEMP correspondent aux tangentes des lignes correspondantes de la colonne V1 et les valeurs de la colonne VX correspondent aux tangentes des lignes correspondantes de la colonne V2. Toutes les valeurs sont en radians.

## Fonctions associées

Fonction	Description
ATAN	Calcule l'arc tangente du contenu de la plage de données spécifiée
COS	Calcule le cosinus du contenu de la plage de données spécifiée
COT	Calcule la cotangente du contenu de la plage de données spécifiée
SIN	Calcule le sinus du contenu de la plage de données spécifiée
TANH	Calcule la tangente hyperbolique du contenu de la plage de données spécifiée

---

## TANH

### Syntaxe

TANH(data [, units\_keyword])

### Paramètres

data

Valeurs numériques dont la tangente hyperbolique doit être calculée. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de data, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

units\_keyword

Ce mot clé facultatif détermine si les valeurs en entrée et les résultats sont interprétés comme des degrés ou des radians. Sélectionnez l'une des valeurs suivantes :

RADIAN - Effectue les calculs en radians (valeur par défaut)

DEGREE - Effectue les calculs en degrés

Si ce paramètre n'est pas spécifié, la valeur par défaut est radians. (Pour convertir des radians en degrés, divisez par PI et multipliez par 180.)

Pour plus de détails sur l'utilisation des mots clés dans IBM Campaign, voir «Spécifications de format», à la page 9.

Pour plus de détails sur l'utilisation des mots clés dans IBM PredictiveInsight, voir «Spécifications de format», à la page 27.

## Description

TANH calcule la tangente hyperbolique des valeurs de la plage de données spécifiée. Il renvoie une nouvelle colonne par colonne d'entrée, chacune contenant la tangente hyperbolique des nombres de la colonne d'entrée correspondante. La tangente hyperbolique d'un nombre est calculée comme suit :

$$\tanh(x) = \frac{\sinh(x)}{\cosh(x)}$$

**Remarque :** Si la valeur  $x$  est trop élevée, une erreur de dépassement est renvoyée. Cela se produit si  $\tanh(x)$  dépasse la valeur maximale à virgule flottante sur 32 bits. Si  $\cosh(x)$  est égal à zéro, TANH renvoie la valeur maximale à virgule flottante sur 32 bits.

## Exemples

<code>TEMP = TANH(PI), TEMP = TANH(PI, 0) ou TEMP = TANH(PI, RADIAN)</code> Crée une colonne nommée TEMP contenant la valeur un.
<code>TEMP = TANH(V1)</code> Crée une colonne nommée TEMP, où chaque valeur correspond à la tangente hyperbolique (en radians) du contenu de la colonne V1.
<code>TEMP = TANH(V1:V3, 1) ou TEMP = TANH(V1:V3, DEGREE)</code> Crée trois colonnes nommées TEMP, VX et VY. Les valeurs de la colonne TEMP correspondent aux tangentes hyperboliques du contenu de la colonne V1, les valeurs de la colonne VX correspondent aux tangentes hyperboliques du contenu de la colonne V2 et les valeurs de la colonne VY correspondent aux tangentes hyperboliques du contenu de la colonne V3. Toutes les valeurs sont en degrés.
<code>TEMP = TANH(V1[1:5]:V2)</code> Crée deux colonnes nommées TEMP et VX, contenant des valeurs dans les lignes 1 à 5 (les autres cibles sont vides). Les valeurs de la colonne TEMP correspondent aux tangentes hyperboliques des lignes correspondantes de la colonne V1 et les valeurs de la colonne VX correspondent aux tangentes hyperboliques des lignes correspondantes de la colonne V2. Toutes les valeurs sont en radians.

## Fonctions associées

Fonction	Description
ATAN	Calcule l'arc tangente du contenu de la plage de données spécifiée
COSH	Calcule le cosinus hyperbolique du contenu de la plage de données spécifiée
COT	Calcule la cotangente du contenu de la plage de données spécifiée
SINH	Calcule le sinus hyperbolique du contenu de la plage de données spécifiée
TAN	Calcule la tangente du contenu de la plage de données spécifiée

---

## TO



### Syntaxe

```
begin TO end begin...end
```

### Paramètres

begin

Numéro de début de la plage à créer. Il peut s'agir d'une valeur constante de type entier ou d'une expression convertie en constante entière.

end

Numéro de fin de la plage à créer. Il peut s'agir d'une valeur constante de type entier ou d'une expression convertie en constante entière.

### Description

TO crée une colonne contenant les valeurs entières commençant par begin et se terminant par end. Cette fonction macro permet de définir la variable de temps dans les fonctions récursives (voir la fonction macro INIT).

**Remarque :** L'opérateur TO peut être abrégé en trois petits points (...).

### Exemples

TEMP = 1 TO 10 ou TEMP = 1...10 Crée une colonne nommée TEMP contenant les valeurs 1 à 10.
TEMP = 0 to -10 Crée une colonne nommée TEMP contenant les valeurs 0 à 10.

## Fonctions associées

Fonction	Description
COLUMN	Crée des colonnes, en concaténant verticalement les valeurs en entrée dans chaque colonne
MERGE	Crée un groupe de données en concaténant horizontalement les valeurs en entrée

---

## TOTAL

### Syntaxe

TOTAL(data [, keyword])

### Paramètres

data

Valeurs numériques dont la somme doit être calculée. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de data, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

keyword

Ce mot clé facultatif détermine la manière dont le calcul est effectué sur la plage de données d'entrée. Sélectionnez l'une des valeurs suivantes :

ALL - Effectue le calcul sur toutes les cibles de data (valeur par défaut)

COL - Effectue le calcul séparément pour chaque colonne de data

ROW - Effectue le calcul séparément pour chaque ligne de data

Pour plus de détails sur l'utilisation des mots clés dans IBM Campaign, voir «Spécifications de format», à la page 9.

Pour plus de détails sur l'utilisation des mots clés dans IBM PredictiveInsight, voir «Spécifications de format», à la page 27.

**Remarque :** De nombreuses fonctions de macro utilisent les paramètres de mot clé {ALL | COL | ROW}. Ces mots clés ne s'appliquent pas dans **IBM Campaign** car les données d'entrées correspondent toujours à une colonne ou à une zone. La macro se comporte toujours comme si le mot clé COL a été spécifié. Par conséquent, vous n'avez pas besoin de spécifier ces mots clés lorsque vous utilisez **IBM Campaign**.

### Description

TOTAL calcule la somme de toutes les cibles de la plage de données spécifiée.

**Remarque :** TOTAL est identique à la fonction macro SUM.

## Exemples

<code>TEMP = TOTAL(3)</code>	Crée une colonne nommée TEMP contenant la valeur trois.
<code>TEMP = TOTAL((COLUMN(3, 5, 1)))</code>	
	Crée une colonne nommée TEMP contenant la valeur neuf.
<code>TEMP = TOTAL(V1)</code>	
	Crée une colonne nommée TEMP contenant une valeur unique qui correspond à la somme du contenu de la colonne V1.
<code>TEMP = TOTAL(V1:V3)</code>	
	Crée une colonne nommée TEMP contenant une valeur unique qui correspond à la somme du contenu des colonnes V1, V2 et V3.
<code>TEMP = TOTAL(V1[1:5]:V4)</code>	
	Crée une colonne nommée TEMP contenant une valeur unique qui correspond à la somme des cibles des lignes 10 à 20 des colonnes V1 à V4.
<code>TEMP = TOTAL(V1:V3, COL)</code>	
	Crée trois colonnes nommées TEMP, VX et VY. La valeur unique de la colonne TEMP correspond à la somme du contenu de la colonne V1, la valeur unique de la colonne VX correspond à la somme du contenu de la colonne V2 et la valeur unique de la colonne VY correspond à la somme du contenu de la colonne V3.
<code>TEMP = TOTAL(V1[1:5]:V3, COL)</code>	
	Crée trois colonnes nommées TEMP, VX et VY contenant chacune une seule valeur. La valeur de la colonne TEMP correspond à la somme des cibles des lignes 1 à 5 de la colonne V1, la valeur de la colonne VX correspond à la somme des cibles des lignes 1 à 5 de la colonne V2, et la valeur de la colonne VY correspond à la somme des cibles des lignes 1 à 5 de la colonne V3.
<code>TEMP = TOTAL(V1:V3, ROW)</code>	
	Crée une colonne nommée TEMP dans laquelle chaque entrée de cible correspond à la somme de la ligne correspondante dans les colonnes V1, V2 et V3.
<code>TEMP = TOTAL(V1[1:5]:V3, ROW)</code>	
	Crée une colonne nommée TEMP, dans laquelle les cibles des lignes 1 à 5 contiennent la somme de la ligne correspondante dans les colonnes V1 à V3. Les autres cibles de TEMP sont vides.

## Fonctions associées

Fonction	Description
AVG ou MEAN	Calcule la moyenne arithmétique d'une plage de cibles
AVG_DEV	Calcule l'écart moyen d'une plage de cibles

---

## TRANSPOSE





## Syntaxe

TRANPOSE(data)

## Paramètres

data

Valeurs numériques ou de chaîne à transposer. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de data, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

## Description

TRANPOSE transpose la plage de données spécifiée. Il change les orientations horizontale et verticale de la plage de données (la première ligne de data devient la première colonne, la deuxième ligne devient la deuxième colonne, etc.).

**Remarque :** La plage de données transposée doit être rectangulaire. Les cibles vides d'une colonne numérique sont remplacées par des zéros. Les cibles vides d'une colonne de type chaîne sont remplacées par la chaîne vide ("").

## Exemples

TEMP = TRANPOSE(COLUMN(1,2,3))
Crée trois colonnes nommées TEMP, VX et VY, contenant chacune une valeur unique 1, 2 et 3.
TEMP = TRANPOSE(MERGE("a", "b"))
Crée une colonne nommée TEMP contenant a et b.
TEMP = TRANPOSE(V1)
Crée une colonne pour chaque ligne de la colonne V1. Chaque colonne contient une valeur ; correspondante de la colonne V1.
TEMP = TRANPOSE(V1:V3)
Crée une colonne pour chaque ligne de la colonne la plus longue entre V1, V2 et V3. Chaque colonne comporte trois lignes contenant les valeurs transposées de V1:V3.
TEMP = TRANPOSE(V1[10:15])
Crée six colonnes, contenant chacune une ligne. La première colonne contient la valeur de V1[10], la deuxième colonne contient V1[11], etc.
TEMP = TRANPOSE(V1[50:99]:V2)
Crée 100 colonnes. Chaque colonne comporte deux lignes contenant les valeurs transposées des lignes 50 à 99 des colonnes V1 et V2.

## Fonctions associées

Fonction	Description
COLUMN	Crée des colonnes, en concaténant verticalement les valeurs en entrée dans chaque colonne

Fonction	Description
MERGE	Crée un groupe de données en concaténant horizontalement les valeurs en entrée

## TRUNCATE

### Syntaxe

TRUNCATE(data)

### Paramètres

data

Valeurs numériques à tronquer. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de data, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

### Description

TRUNCATE calcule la partie entière de chaque valeur dans la plage de données spécifiée. Il renvoie une nouvelle colonne par colonne d'entrée, chacune contenant la partie entière (sans partie décimale) des nombres dans la colonne d'entrée correspondante.

**Remarque :** Les fonctions macro FRACTION et TRUNCATE sont complémentaires car leur somme correspond aux valeurs d'origine.

### Exemples

TEMP = TRUNCATE(4.3)
Crée une colonne nommée TEMP contenant la valeur 4.
TEMP = TRUNCATE(2.9)
Crée une colonne nommée TEMP contenant la valeur -2.
TEMP = TRUNCATE(V1)
Crée une colonne nommée TEMP, dans laquelle chaque valeur correspond à la partie décimale du contenu de la colonne V1.
TEMP = TRUNCATE(V1:V3)
Crée trois colonnes nommées TEMP, VX et VY. Les valeurs de la colonne TEMP correspondent aux parties tronquées de la colonne V1, les valeurs de la colonne VX correspondent aux parties tronquées de la colonne V2 et les valeurs de la colonne VY correspondent aux parties tronquées de la colonne V3.
TEMP = TRUNCATE(V1[10:20])
Crée une colonne nommée TEMP, dans laquelle les 11 premières cibles contiennent les parties tronquées des valeurs des lignes 10 à 20 de la colonne V1. Les autres cibles de TEMP sont vides.

```
TEMP = TRUNCATE(V1[50:99]:V2)
```

Crée deux colonnes nommées TEMP et VX, contenant des valeurs dans les lignes 1 à 50 (les autres cibles sont vides). Les valeurs de la colonne TEMP correspondent aux parties tronquées des lignes de la colonne V1 et les valeurs de la colonne VX correspondent aux parties tronquées des valeurs de la colonne V2.

## Fonctions associées

Fonction	Description
CEILING	Calcule le plafond de chaque valeur de la plage de données spécifiée
FLOOR	Calcule le plancher de chaque valeur de la plage de données spécifiée
FRACTION	Renvoie la partie décimale de chaque valeur de la plage de données spécifiée

---

## UPPER

### Syntaxe

```
UPPER(data)
```

### Paramètres

data

Valeur de chaîne à convertir en majuscules.

### Description

UPPER convertit chaque valeur de chaîne de la plage de données spécifiée en majuscules. Il renvoie une nouvelle colonne par colonne d'entrée, chacune contenant la chaîne en majuscules dans la colonne d'entrée correspondante.

Cette macro est disponible dans IBM Interact.

### Exemples

```
Temp = UPPER "gold"
```

Crée une colonne nommée Temp contenant "GOLD".

```
TEMP = UPPER( "jan 15, 1997")
```

Crée une colonne nommée TEMP, qui contient la chaîne de texte ASCII " JAN 15, 1997 ".

```
TEMP = UPPER( "Pressure")
```

Crée une colonne nommée TEMP, qui contient la chaîne de texte ASCII "PRESSURE".

```
TEMP = UPPER(V1)
```

Crée une colonne nommée TEMP contenant les caractères en majuscules de chaque chaîne de la colonne V1.

---

## VARIANCE

### Syntaxe

VARIANCE(data [, keyword])

### Paramètres

data

Valeurs numériques dont la variance doit être calculée. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de data, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

keyword

Ce mot clé facultatif détermine la manière dont le calcul est effectué sur la plage de données d'entrée. Sélectionnez l'une des valeurs suivantes :

ALL - Effectue le calcul sur toutes les cibles de data (valeur par défaut)

COL - Effectue le calcul séparément pour chaque colonne de data

ROW - Effectue le calcul séparément pour chaque ligne de data

Pour plus de détails sur l'utilisation des mots clés dans IBM Campaign, voir «Spécifications de format», à la page 9.

Pour plus de détails sur l'utilisation des mots clés dans IBM PredictiveInsight, voir «Spécifications de format», à la page 27.

**Remarque :** De nombreuses fonctions macro acceptent les paramètres de mot clé {ALL | COL | ROW}. Ces mots clés ne s'appliquent pas dans **IBM Campaign** car les données d'entrées correspondent toujours à une colonne ou à une zone. La macro se comporte toujours comme si le mot clé COL a été spécifié. Par conséquent, vous n'avez pas besoin de spécifier ces mots clés lorsque vous utilisez **IBM Campaign**.

### Description

VARIANCE calcule la variance de toutes les valeurs de la plage de données spécifiée. La variance est égale au carré de l'écart type. La variance est calculée comme suit :

$$\frac{1}{n - 1} \sum_{j=1}^n (x_j - \text{mean})^2$$

où les  $x$  représentent les échantillons,  $n$  représente le nombre d'échantillons et  $\text{mean}$  est la moyenne de la distribution.

**Remarque :** Si le nombre d'échantillons  $n = 1$ , VARIANCE renvoie une erreur.

## Exemples

<p>TEMP = VARIANCE(MERGE(3, 4, 5)) ou TEMP = VARIANCE(MERGE(3, 4, 5), ALL)</p> <p>Crée une colonne nommée TEMP contenant la valeur 0.67.</p>
<p>TEMP = VARIANCE(MERGE(-10, 5, 10))</p> <p>Crée une colonne nommée TEMP contenant la valeur 72.2.</p>
<p>TEMP = VARIANCE(V1)</p> <p>Crée une colonne nommée TEMP contenant une valeur unique qui correspond à la variance du contenu de la colonne V1.</p>
<p>TEMP = VARIANCE(V1:V3)</p> <p>Crée une colonne nommée TEMP contenant une valeur unique qui correspond à la variance du contenu des colonnes V1, V2 et V3.</p>
<p>TEMP = VARIANCE(V1[10:20])</p> <p>Crée une colonne nommée TEMP contenant une valeur unique qui correspond à la variance des cibles des lignes 10 à 20 de la colonne V1.</p>
<p>TEMP = VARIANCE(V1[1:5]:V4)</p> <p>Crée une colonne nommée TEMP contenant une valeur unique qui correspond à la variance des cibles des lignes 1 à 5 des colonnes V1 à V4.</p>
<p>TEMP = VARIANCE(V1:V3, COL)</p> <p>Crée trois colonnes nommées TEMP, VX et VY. La valeur unique de la colonne TEMP correspond à la variance du contenu de la colonne V1, la valeur unique de la colonne VX correspond à la variance du contenu de la colonne V2 et la valeur unique de la colonne VY correspond à la variance du contenu de la colonne V3.</p>
<p>TEMP = VARIANCE(MERGE(1,4), COL)</p> <p>Crée deux colonnes nommées TEMP et VX, contenant chacune la valeur zéro.</p>
<p>TEMP = VARIANCE_(V1[1:5]:V3, COL) ou TEMP = VARIANCE(V1[1:5]:V3[1:5], COL)</p> <p>Crée trois colonnes nommées TEMP, VX et VY contenant chacune une seule valeur. La valeur de la colonne TEMP correspond à la variance des cibles des lignes 1 à 5 de la colonne V1, la valeur de la colonne VX correspond à la variance des cibles des lignes 1 à 5 de la colonne V2, et la valeur de la colonne VY correspond à la variance des cibles des lignes 1 à 5 de la colonne V3.</p>
<p>TEMP = VARIANCE(V1:V3, ROW)</p> <p>Crée une colonne nommée TEMP dans laquelle chaque entrée de cible correspond à la variance de la ligne correspondante dans les colonnes V1, V2 et V3.</p>
<p>TEMP = VARIANCE(V1[1:5]:V3,ROW) ou TEMP = VARIANCE(V1[1:5]:V3[1:5], ROW)</p> <p>Crée une colonne nommée TEMP, dans laquelle les cibles des lignes 1 à 5 contiennent la variance de la ligne correspondante dans les colonnes V1 à V3. Les autres cibles de TEMP sont vides.</p>

## Fonctions associées

Fonction	Description
KURTOSIS	Calcule le kurtosis d'une plage de cibles

Fonction	Description
SKEW	Calcule l'asymétrie de la distribution d'une plage de cibles

---

## WEEKDAY

### Syntaxe

WEEKDAY(data [, conversion\_keyword])

### Paramètres

data

Dates de texte ASCII à convertir en valeurs numériques représentant les jours de la semaine (1-7). Il peut s'agir de texte ASCII entre guillemets, d'une colonne de texte, d'une plage de cibles contenant du texte ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de data, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

conversion\_keyword

Ce mot clé facultatif indique comment interpréter les formats de texte des dates et des heures. Sélectionnez l'une des valeurs suivantes :

1 - mm/dd/yy (valeur par défaut)

2 - dd-mmm-yy

3 - mm/dd/yy hh:mm

Si ce paramètre n'est pas spécifié, la valeur par défaut est 1.

### Description

WEEKDAY convertit les valeurs de texte dans la plage de données spécifiée en valeurs numériques représentant les jours de la semaine, à l'aide du format spécifié pour convertir les dates et les heures. Le nombre 0 correspond à dimanche, 1 à lundi, et ainsi de suite jusqu'à 6 pour samedi. Si une chaîne de texte ne peut pas être analysée à l'aide du mot clé conversion\_keyword spécifié, WEEKDAY renvoie une erreur.

### Exemples

```
TEMP = WEEKDAY("1/1/95")
```

Crée une colonne nommée TEMP contenant le nombre 0 (1er janvier 1995 est un dimanche).

```
TEMP = WEEKDAY(V1, 2)
```

Crée une colonne nommée TEMP contenant les nombres des jours de la semaine pour les chaînes de texte de la colonne V1. Toutes les chaînes de texte de la colonne V1 doivent être au format dd-mmm-yy (sinon des ??? sont renvoyés).

```
TEMP = WEEKDAY(V1:V3, 3)
```

Crée trois colonnes nommées TEMP, VX et VY. La colonne TEMP contient des nombres représentant les jours de la semaine des chaînes de texte de la colonne V1. La colonne VX contient des nombres représentant les jours de la semaine des chaînes de texte de la colonne V2. La colonne VY contient des nombres représentant les jours de la semaine des chaînes de texte de la colonne V3. Toutes les chaînes de texte des colonnes V1 à V3 doivent se présenter sous la forme mm/dd/yy hh:mm (sinon, des ??? sont renvoyés).

```
TEMP = WEEKDAY(V1[10:20]:V2, 10)
```

Crée deux colonnes nommées TEMP et VX. La colonne TEMP contient des nombres représentant les jours de la semaine des chaînes de texte des lignes 10 à 20 de la colonne V1. La colonne VX contient des nombres représentant les jours de la semaine des chaînes de texte des lignes 10 à 20 de la colonne V2. Toutes les chaînes de texte doivent se présenter sous la forme mm/dd/yy (sinon, des ??? sont renvoyés).

## Fonctions associées

Fonction	Description
NUMBER	Convertit les chaînes de texte ASCII des heures et des dates en valeurs numériques

---

## WEEKDAYOF

### Syntaxe

```
WEEKDAYOF(date_string [, input_format])
```

### Paramètres

date\_string

Texte représentant une date valide.

input\_format

Un des mots clés du tableau ci-après, spécifiant le format de date de date\_string.

### Description

WEEKDAYOF renvoie le jour de la semaine sous la forme d'un nombre compris entre 0 et 6 (dimanche = 0, lundi = 1, et ainsi de suite) pour la date indiquée par date\_string. Si input\_format n'est pas fourni, le mot clé par défaut DELIM\_M\_D\_Y est utilisé.

### Exemples

WEEKDAYOF("08312000", MMDDYYYY) renvoie le nombre 4 car jeudi est le quatrième jour de la semaine.

**Remarque :** Pour des informations supplémentaires sur les formats de date valides, voir «DATE», à la page 77.

## Fonctions associées

Fonction	Description
DAYOF	Renvoie le jour du mois sous la forme d'un nombre.
MONTHOF	Renvoie le mois de l'année sous la forme d'un nombre.
YEAROF	Renvoie l'année sous la forme d'un nombre.

---

## XOR

### Syntaxe

data1 XOR data2

### Paramètres

data1

Entiers non négatifs dont le XOR bit à bit doit être calculé avec les valeurs de data2. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Pour la définition du format de data, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

data2

Entiers non négatifs dont le XOR bit à bit doit être calculé avec les valeurs de data1. Il peut s'agir d'une valeur constante, d'une colonne, d'une plage de cibles ou d'une expression convertie comme l'une de ces valeurs. Le nombre de colonnes dans data2 doit correspondre au nombre de colonnes dans data1, à moins que data2 ne soit une constante. Pour la définition du format de data, voir la section "Paramètres des fonctions macro" du chapitre du présent document correspondant à votre produit IBM.

### Description

XOR calcule le XOR bit à bit entre les deux plages de données spécifiées. Il renvoie une nouvelle colonne pour chaque colonne d'entrée, chacune contenant la colonne correspondante de data1 dont le XOR bit à bit avec la colonne correspondante de data2 a été calculé (à savoir, le XOR bit à bit de la première colonne de data1 avec la première colonne de data, celui de la deuxième colonne avec la deuxième colonne et ainsi de suite).

Si data2 est une constante, le XOR bit à bit de chaque valeur de data1 avec cette valeur est calculé. Si data2 contient une ou plusieurs colonnes, les calculs sont effectués ligne par ligne entre une colonne de data1 et une colonne de data2. Le XOR bit à bit de la première ligne de data1 avec la première ligne de data2 est calculé, la seconde ligne avec la seconde ligne, etc. Ce calcul ligne par ligne génère un résultat pour chaque ligne jusqu'à la dernière valeur de la colonne la plus courte.

**Remarque :** La précision de cette fonction macro est limitée aux valeurs entières inférieures à  $2^{24}$ . Les valeurs négative sont interdites.



## Exemples

TEMP = 3 XOR 7
Crée une colonne nommée TEMP contenant la valeur quatre (le XOR bit à bit de 011 et 111 est égal à 100).
TEMP = V1 XOR 8
Crée une colonne nommée TEMP, dans laquelle chaque valeur correspond au XOR bit à bit du contenu de la colonne V1 avec la valeur binaire 1000.
TEMP = V1 XOR V1
Crée une colonne nommée TEMP ne contenant que des zéros (un XOR entre toute valeur et elle-même génère zéro).
TEMP = V1 XOR V2
Crée une colonne nommée TEMP, dans laquelle chaque valeur est la valeur de ligne de la colonne V1 dont le XOR bit à bit est calculé avec la valeur de ligne correspondante de la colonne V2.
TEMP = V1:V3 XOR V4:V6
Crée trois colonnes nommées TEMP, VX et VY. La colonne TEMP contient les valeurs de V1 dont le XOR bit à bit avec les valeurs de ligne correspondantes de la colonne V4 a été calculé. La colonne VX contient les valeurs résultant du XOR bit à bit des colonnes V2 et V5. La colonne VY contient les valeurs résultant du XOR bit à bit des colonnes V3 et V6.
TEMP = V1[10:20] XOR V2 ou TEMP = V1[10:20] XOR V2[1:11]
Crée une colonne nommée TEMP, dans laquelle les 11 premières cibles contiennent le résultat du XOR bit à bit des valeurs des lignes 10 à 20 de la colonne V1 avec les valeurs des lignes 1 à 11 de la colonne V2. Les autres cibles de TEMP sont vides.

## Fonctions associées

Fonction	Description
BIT_AND	Calcule le AND bit à bit entre deux plages de données spécifiées
BIT_NOT	Calcule le NOT bit à bit du contenu de la plage de données spécifiée
BIT_OR	Calcule le OR bit à bit entre deux plages de données spécifiées

---

## XTAB



### Syntaxe

XTAB(col1, col2 [, operator\_keyword, numeric\_col3])

### Paramètres

col1

Première colonne à partir de laquelle xtab doit être généré. Il peut s'agir d'une constante, d'une colonne, d'une plage de cibles à une colonne ou de toute expression convertie comme l'une de ces valeurs.

col2

Deuxième colonne à partir de laquelle xtab doit être généré. Il peut s'agir d'une constante, d'une colonne, d'une plage de cibles à une colonne ou de toute expression convertie comme l'une de ces valeurs.

operator\_keyword

Un des mots clés d'opérateur valides (voir ci-dessous).

numeric\_col3

Troisième colonne à partir de laquelle xtab doit être généré. Il peut s'agir d'une constante, d'une colonne, d'une plage de cibles à une colonne ou de toute expression convertie comme l'une de ces valeurs contenant une valeur numérique.

## Description

XTAB calcule les valeurs distinctes dans col1 et col2. Ensuite, il calcule le mot clé operator\_keyword de numeric\_col3 à l'intersection de chaque valeur col1 avec chaque valeur col2.

Par défaut, le mot clé operator\_keyword est COUNTOF, auquel cas numeric\_col3 n'est pas utilisé.

Les mots clés operator\_keywords possibles sont les suivants :

COUNTOF - renvoie le nombre d'enregistrements à chaque intersection.

COUNTZERO - renvoie le nombre d'enregistrements à chaque intersection pour laquelle numeric\_col3 est égal à 0.

COUNTNONZERO - renvoie le nombre d'enregistrements à chaque intersection pour laquelle numeric\_col3 n'est pas égal à 0.

COUNTNULL - renvoie le nombre d'enregistrements à chaque intersection pour laquelle numeric\_col3 est NULL.

MINOF - renvoie la plus petite valeur de numeric\_col3 à chaque intersection ; renvoie la valeur manquante s'il n'existe aucune valeur à l'intersection.

MAXOF - renvoie la plus grande valeur de numeric\_col3 à chaque intersection ; renvoie la valeur manquante s'il n'existe aucune valeur à l'intersection.

SUMOF - renvoie la somme de toutes les valeurs numeric\_col3 à chaque intersection.

AVGOF - renvoie la moyenne de toutes les valeurs numeric\_col3 non NULL à chaque intersection.

STDEVOF - renvoie l'écart type de toutes les valeurs numeric\_col3 non NULL à chaque intersection.

**Remarque :** Le calcul effectué par la fonction macro XTAB peut être assez long si la quantité de données est importante. Une barre de progression du calcul est affichée jusqu'à la fin du calcul. Si vous décidez d'annuler le calcul, cliquez sur le "X" dans la barre de progression et supprimez la définition de fonction contenant la fonction macro XTAB.

## Exemples

```
TEMP=XTAB(V1,V2)
```

Crée une série de lignes et de colonnes qui calculent le nombre de valeurs distinctes aux intersections des colonnes V1 et V2.

```
TEMP=XTAB(V4,V5, SUMOF V6)
```

Crée une série de lignes et de colonnes qui représentent l'intersection des valeurs distinctes des colonnes V4 et V5. La mesure à chaque intersection correspond à la somme des valeurs de la colonne V6 pour les lignes correspondant à cette intersection.

---

## YEAROF

### Syntaxe

```
YEAROF(date_string [, input_format])
```

### Paramètres

date\_string

Texte représentant une date valide.

input\_format

Un des mots clés du tableau ci-après, spécifiant le format de date de date\_string.

### Description

YEAROF renvoie l'année sous la forme d'un nombre pour la date indiquée par date\_string. Si input\_format n'est pas fourni, le mot clé par défaut DELIM\_M\_D\_Y est utilisé.

### Exemples

```
YEAROF("31082000", DDMYYYY) renvoie le nombre 2000.
```

Pour des informations supplémentaires sur les formats de date valides, voir «DATE», à la page 77.

### Fonctions associées

Fonction	Description
DAYOF	Renvoie le jour du mois sous la forme d'un nombre.
MONTHOF	Renvoie le mois de l'année sous la forme d'un nombre.

<b>Fonction</b>	<b>Description</b>
WEEKDAYOF	Renvoie le jour de la semaine sous la forme d'un nombre.

---

## Comment contacter le support technique IBM

Si vous rencontrez un problème que vous ne parvenez pas à résoudre en consultant la documentation, le correspondant désigné pour le support technique de votre entreprise peut contacter le support technique IBM. Prenez connaissance des informations ci-dessous pour faire en sorte que votre problème soit résolu de manière efficace et fructueuse.

Si vous n'êtes pas le correspondant désigné pour le support technique de votre entreprise, contactez votre administrateur IBM pour obtenir plus d'informations.

### Informations à réunir

Avant de contacter le support technique IBM, vous devez collecter les informations suivantes :

- Brève description de la nature du problème rencontré
- Les messages d'erreur détaillés qui apparaissent lorsque l'erreur se produit,
- La liste des étapes complètes permettant de reproduire l'erreur.
- Les fichiers journaux, fichiers de session, fichiers de configuration et fichiers de données associés.
- Les informations relatives à votre environnement (produit et système), que vous pouvez obtenir en suivant la procédure décrite sous "Informations système".

### Informations système

Lorsque vous appelez le support technique IBM, vous êtes généralement invité à fournir des informations sur votre environnement.

Si votre problème ne vous empêche pas de vous connecter, la plupart de ces informations sont disponibles à la page **A propos de**, qui fournit des informations relatives aux applications IBM que vous avez installées.

Pour accéder à la page **A propos de**, sélectionnez **Aide > A propos de**. Si cette page n'est pas accessible, vous pouvez obtenir le numéro de version de chaque application IBM en consultant le fichier `version.txt` se trouvant dans le répertoire d'installation des applications concernées.

### Informations de contact pour le support technique IBM

Pour savoir comment contacter le support technique IBM, consultez le site Web de support technique IBM : ([http://www-947.ibm.com/support/entry/portal/open\\_service\\_request](http://www-947.ibm.com/support/entry/portal/open_service_request)).



---

## Remarques

Le présent document peut contenir des informations ou des références concernant certains produits, logiciels ou services IBM non annoncés dans ce pays. Pour plus de détails, référez-vous aux documents d'annonce disponibles dans votre zone, ou adressez-vous à votre partenaire commercial IBM. Toute référence à un produit, logiciel ou service IBM n'implique pas que seul ce produit, logiciel ou service puisse être utilisé. Tout autre élément fonctionnellement équivalent peut être utilisé, s'il n'enfreint aucun droit d'IBM. Il est de la responsabilité de l'utilisateur d'évaluer et de vérifier lui-même les installations et applications réalisées avec des produits, logiciels ou services non expressément référencés par IBM.

IBM peut détenir des brevets ou des demandes de brevet couvrant les produits mentionnés dans le présent document. La remise de ce document ne vous accorde aucun droit de licence sur ces brevets ou demandes de brevet. Si vous désirez recevoir des informations concernant l'acquisition de licences, veuillez en faire la demande par écrit à l'adresse suivante :

IBM Director of Licensing  
IBM Corporation  
North Castle Drive  
Armonk, NY 10504-1785  
U.S.A.

Pour le Canada, veuillez adresser votre courrier à :

IBM Director of Commercial Relations  
IBM Canada Ltd  
3600 Steeles Avenue East  
Markham, Ontario  
L3R 9Z7  
Canada

Les informations sur les licences concernant les produits utilisant un jeu de caractères double octet peuvent être obtenues par écrit à l'adresse suivante :

Intellectual Property Licensing  
Legal and Intellectual Property Law  
IBM Japan Ltd.  
1623-14, Shimotsuruma, Yamato-shi  
Kanagawa 242-8502 Japan

Le paragraphe suivant ne s'applique ni au Royaume-Uni, ni dans aucun pays dans lequel il serait contraire aux lois locales. LE PRESENT DOCUMENT EST LIVRE EN L'ETAT SANS AUCUNE GARANTIE EXPLICITE OU IMPLICITE. IBM DECLINE NOTAMMENT TOUTE RESPONSABILITE RELATIVE A CES INFORMATIONS EN CAS DE CONTREFAÇON AINSI QU'EN CAS DE DEF AUT D'APTITUDE A L'EXECUTION D'UN TRAVAIL DONNE. Certaines juridictions n'autorisent pas l'exclusion des garanties implicites, auquel cas l'exclusion ci-dessus ne vous sera pas applicable.

Le présent document peut contenir des inexactitudes ou des coquilles. Ce document est mis à jour périodiquement. Chaque nouvelle édition inclut les mises à jour. IBM peut, à tout moment et sans préavis, changer les produits et logiciels décrits dans ce document.

Les références à des sites Web non IBM sont fournies à titre d'information uniquement et n'impliquent en aucun cas une adhésion aux données qu'ils contiennent. Les éléments figurant sur ces sites Web ne font pas partie des éléments du présent produit IBM et l'utilisation de ces sites relève de votre seule responsabilité.

IBM pourra utiliser ou diffuser, de toute manière qu'elle jugera appropriée et sans aucune obligation de sa part, tout ou partie des informations qui lui seront fournies.

Les licenciés souhaitant obtenir des informations permettant : (i) l'échange des données entre des logiciels créés de façon indépendante et d'autres logiciels (dont celui-ci), et (ii) l'utilisation mutuelle des données ainsi échangées, doivent adresser leur demande à :

IBM Corporation  
170 Tracer Lane  
Waltham, MA 02451  
U.S.A.

Ces informations peuvent être soumises à des conditions particulières, prévoyant notamment le paiement d'une redevance.

Le logiciel sous licence décrit dans ce document et tous les éléments sous licence disponibles s'y rapportant sont fournis par IBM conformément aux dispositions de l'ICA, des Conditions internationales d'utilisation des logiciels IBM ou de tout autre accord équivalent.

Les données de performance indiquées dans ce document ont été déterminées dans un environnement contrôlé. Par conséquent, les résultats peuvent varier de manière significative selon l'environnement d'exploitation utilisé. Certaines mesures évaluées sur des systèmes en cours de développement ne sont pas garanties sur tous les systèmes disponibles. En outre, elles peuvent résulter d'extrapolations. Les résultats peuvent donc varier. Il incombe aux utilisateurs de ce document de vérifier si ces données sont applicables à leur environnement d'exploitation.

Les informations concernant des produits non IBM ont été obtenues auprès des fournisseurs de ces produits, par l'intermédiaire d'annonces publiques ou via d'autres sources disponibles. IBM n'a pas testé ces produits et ne peut confirmer l'exactitude de leurs performances ni leur compatibilité. Elle ne peut recevoir aucune réclamation concernant des produits non IBM. Toute question concernant les performances de produits non IBM doit être adressée aux fournisseurs de ces produits.

Toute instruction relative aux intentions d'IBM pour ses opérations à venir est susceptible d'être changée ou annulée sans préavis, et doit être considérée uniquement comme un objectif.

Tous les tarifs indiqués sont les prix de vente actuels suggérés par IBM et sont susceptibles d'être changés sans préavis. Les tarifs appliqués peuvent varier selon les revendeurs.



Le présent document peut contenir des exemples de données et de rapports utilisés couramment dans l'environnement professionnel. Ces exemples mentionnent des noms fictifs de personnes, de sociétés, de marques ou de produits à des fins illustratives ou explicatives uniquement. Toute ressemblance avec des noms d'individus, de sociétés ou des données réelles serait purement fortuite.

#### LICENCE DE COPYRIGHT :

Le présent logiciel contient des exemples de programmes d'application en langage source destinés à illustrer les techniques de programmation sur différentes plateformes d'exploitation. Vous avez le droit de copier, de modifier et de distribuer ces exemples de programmes sous quelque forme que ce soit et sans paiement d'aucune redevance à IBM, à des fins de développement, d'utilisation, de vente ou de distribution de programmes d'application conformes aux interfaces de programme d'application pour lesquels ils ont été écrits ou aux interfaces de programmation IBM. Ces exemples de programmes n'ont pas été rigoureusement testés dans toutes les conditions. Par conséquent, IBM ne peut garantir expressément ou implicitement la fiabilité, la maintenabilité ou le fonctionnement de ces programmes. Les exemples de programme sont fournis "EN L'ETAT", sans garantie d'aucune sorte. IBM ne sera en aucun cas responsable des dommages liés à l'utilisation de ces exemples de programme.

Si vous visualisez ces informations en ligne, il se peut que les photographies et illustrations en couleur n'apparaissent pas à l'écran.

---

## Marques

IBM, le logo IBM et [ibm.com](http://ibm.com) sont des marques d'International Business Machines dans de nombreux pays. Les autres noms de produits et de services peuvent être des marques déposées d'IBM ou d'autres sociétés. La liste actualisée de toutes les marques d'IBM est disponible sur la page Web «Copyright and trademark information» à l'adresse [www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml).

---

## Règles de confidentialité et conditions d'utilisation

Les produits IBM Software, notamment les logiciels sous forme de services ("Offres logicielles"), peuvent utiliser des cookies ou d'autres technologies pour collecter des informations d'utilisation en vue d'améliorer l'expérience de l'utilisateur final, d'ajuster les interactions avec l'utilisateur final ou à d'autres fins. Un cookie est une donnée qu'un site Web peut envoyer à votre navigateur et qui peut ensuite être stockée sur votre ordinateur sous la forme d'une balise identifiant ce dernier. Dans la plupart des cas, aucune information personnelle n'est collectée par ces cookies. Si vous utilisez une offre logicielle qui vous permet de collecter des informations personnelles via des cookies et des technologies similaires, tenez compte des spécificités suivantes.

En fonction de la configuration déployée, cette offre logicielle peut utiliser des cookies de session et des cookies permanents qui collectent le nom d'utilisateur de chaque utilisateur ainsi que d'autres informations personnelles à des fins de gestion des sessions, de convivialité améliorée pour l'utilisateur ou d'autres objectifs de suivi de l'utilisation ou fonctionnels. Ces cookies peuvent être désactivés mais leur désactivation élimine également la fonctionnalité qu'ils activent.

Diverses juridictions régulent la collecte d'informations personnelles via les cookies et autres technologies similaires. Si la configuration déployée pour cette offre

logicielle vous permet, en tant que client, de collecter des informations personnelles d'utilisateurs finaux via des cookies et autres technologies, vous devez rechercher les conseils juridiques sur les lois applicables à ces collectes de données, notamment toutes les exigences relatives à l'indication de consignes et de consentements, le cas échéant.

IBM exige que les clients (1) fournissent un lien clair et visible vers les conditions d'utilisation du site Web du client (par exemple les règles de confidentialité) qui comprenne un lien vers les collectes de données et les pratiques d'utilisation d'IBM et du client, (2) indiquent que des cookies et des gifs et pixels invisibles sont placés sur l'ordinateur du visiteur par IBM pour le compte du client avec une explication de l'objectif de cette technologie et (3), selon les conditions requises par la loi, obtiennent le consentement des visiteurs du site Web avant de placer les cookies et les gifs et pixels invisibles par le client ou IBM sur leurs unités.

Pour plus d'informations sur l'utilisation des diverses technologies, notamment des cookies, reportez-vous à la section intitulée "Cookies, Web Beacons and Other Technologies" d'IBM's Online Privacy Statement, à l'adresse : <http://www.ibm.com/privacy/details/us/en>.



