



TI-Nspire™ CAS **TI-Nspire™ CX CAS** **Guía de Referencia**

Esta guía corresponde a la versión 3.2 del software TI-Nspire™. Para obtener la versión más reciente de la documentación, visite el sitio education.ti.com/guides.

Información importante

Excepto por lo que se establezca expresamente en contrario en la Licencia que se incluye con el programa, Texas Instruments no otorga ninguna garantía, ni expresa ni implícita, incluidas pero sin limitarse a cualquier garantía implícita de comerciabilidad e idoneidad con un propósito en particular, en relación con cualquier programa o material impreso, y hace dichos materiales disponibles únicamente "tal y como se encuentran". En ningún caso Texas Instruments será responsable en relación con ninguna persona de daños especiales, colaterales, incidentales o consecuenciales en conexión con o que surjan de la compra o el uso de estos materiales, y la responsabilidad única y exclusiva de Texas Instruments, independientemente de la forma de acción, no excederá la cantidad estipulada en la licencia para el programa. Asimismo, Texas Instruments no será responsable de ninguna reclamación de ningún tipo en contra del uso de estos materiales por parte de cualquier otro individuo.

Licencia

Favor de ver la licencia completa instalada en

C:\Program Files\TI Education\<TI-Nspire™ Product Name>\license.

© 2006 - 2012 Texas Instruments Incorporated

Contents

Plantillas de expresiones

Plantilla de fracciones	1
Plantilla de exponentes	1
Plantilla de raíz cuadrada	1
Plantilla de raíz enésima	1
e plantilla de exponentes	2
Plantilla de logística	2
Plantilla de compuesto de variables (2 piezas)	2
Plantilla de compuesto de variables (N piezas)	2
Sistema de plantilla de 2 ecuaciones	3
Sistema de plantilla de N ecuaciones	3
Plantilla de valor absoluto	3
plantilla gg°mm'ss.ss''	3
Plantilla de matriz (2 x 2)	3
Plantilla de matriz (1 x 2)	4
Plantilla de matriz (2 x 1)	4
Plantilla de matriz (m x n)	4
Plantilla de suma (Σ)	4
Plantilla de producto (Π)	4
Plantilla de primera derivada	5
Plantilla de segunda derivada	5
Plantilla de enésima derivada	5
Plantilla de integral definida	5
Plantilla de integral indefinida	5
Plantilla de límite	6

Listado alfabético

A

abs()	7
amortTbl() (tablaAmort)	7
and (y)	7
angle()	8
ANOVA	8
ANOVA2way (ANOVA2vías)	9
Ans	11
approx()	11
▶approxFraction()	11
approxRational()	11
arccos()	11
arccosh()	12
arccot()	12
arccoth()	12
arccsc()	12
arccsch()	12
arcLen()	12
arcsec()	12
arcsech()	12
arcsin()	12
arcsinh()	12
arctan()	12
arctanh()	12
augment()	12
avgRC()	13

B

bal()	13
▶Base2	14
▶Base10	14
▶Base16	15
binomCdf()	15
binomPdf()	15

C

ceiling() (techo)	15
centralDiff()	16
cFactor()	16
char()	17
charPoly()	17
χ^2 2way	17
χ^2 Cdf()	18
χ^2 GOF	18
χ^2 Pdf()	18
ClearAZ (LimpiarAZ)	19
ClrErr (LimpErr)	19
colAugment()	19
colDim()	19
colNorm()	19
comDenom()	20
completeSquare()	21
conj()	21
constructMat()	21
CopyVar	22
corrMat()	22
▶cos	22
cos()	23
cos ⁻¹ ()	24
cosh()	24
cosh ⁻¹ ()	25
cot()	25
cot ⁻¹ ()	25
coth()	26
coth ⁻¹ ()	26
count()	26
countif() (conteoSí)	27
cPolyRoots() (RaícesPolic)	27
crossP()	27
csc()	28
csc ⁻¹ ()	28
csch()	28
csch ⁻¹ ()	28
cSolve() (solucionC)	29
CubicReg	31
cumulativeSum()	31
Cycle	32
▶Cylind	32
cZeros()	32

D

dbd()	34
▶DD	34

►Decimal	34
Define (Definir)	35
Define LibPriv	35
Define LibPub	36
deltaList()	36
deltaTmpCnv()	36
DelVar	36
delVoid() (borrarInválido)	36
derivative()	36
deSolve() (resolverEd)	37
det()	38
diag()	38
dim()	38
Disp	39
►DMS (►GMS)	39
domain() (dominio)	39
dominantTerm()	40
dotP() (pPunto)	40
E	
e^()	41
eff()	41
eigVC() (vcProp)	41
eigVI() (vlProp)	42
Else (Más)	42
Elseif (MásSi)	42
EndFor (TerminarPara)	42
EndFunc (TerminarFunc)	42
EndIf (TerminarSi)	42
EndLoop (TerminarBucle)	42
EndPrgm (TerminarPrgm)	42
EndTry (TerminarIntentar)	42
EndWhile (TerminarMientras)	43
euler()	43
exact()	43
Exit (Salir)	44
►exp	44
exp()	44
expList()	45
expand() (expandir)	45
expr()	46
ExpReg	46
F	
factor()	47
FCdf()	48
Fill (Llenar)	48
FiveNumSummary (ResumenNúmCinco)	49
floor() (piso)	49
fMax()	49
fMin()	50
For (Para)	50
format()	51
fPart() (parteF)	51
FPdf()	51
freqTableList()	52
frequency (frecuencia)	52
FTest_2Samp	52
Func	53

G	
gcd() (mcd)	53
geomCdf()	54
geomPdf()	54
getDenom()	54
getLangInfo() (obtnfoldioma)	54
getLockInfo()	55
getMode()	55
getNum()	56
getType()	56
getVarInfo()	56
Goto (IrA)	57
►Grad	57

I	
identity() (identidad)	58
If (Si)	58
lffn()	59
imag()	59
impDif()	60
Indirección	60
inString() (enCadena)	60
int()	60
intDiv()	60
integral	60
interpolate()	61
inv χ^2 ()	61
invF()	61
invNorm()	61
invT()	61
iPart()	62
irr()	62
isPrime()	62
isVoid() (esInválido)	62

E	
Lbl (Etiqu)	63
lcm() (mínimo común múltiplo)	63
left() (izquierda)	63
libShortcut() (accesoDirectoLib)	64
límit() o lím()	64
LinRegBx	65
LinRegMx	66
LinRegIntervals	67
LinRegtTest	68
linSolve()	69
Δ List()	69
listMat()	69
►ln	69
ln()	70
LnReg	70
Local	71
Lock (Bloquear)	71
log()	72
►logbase	72
Logistic	73
LogisticD	74
Loop (Bucle)	75
LU (BA)	75

M			
matList()	76	Product (PI)	94
max()	76	product()	94
mean() (media)	76	propFrac()	95
median() (mediana)	77	Q	
MedMed	77	QR	95
mid()	78	QuadReg (RegCuad)	96
min()	78	QuartReg (RegCuart)	97
mirr()	78	R	
mod()	79	R▶P0()	98
mRow() (filaM)	79	R▶Pr()	98
mRowAdd() (agrFilaM)	79	▶Rad	98
MultReg	79	rand() (aleat)	98
MultRegIntervals	80	randBin() (binAleat)	99
MultRegTests (PruebasRegMult)	80	randInt() (entAleat)	99
N		randMat() (matAleat)	99
nand	81	randNorm() (normAleat)	99
nCr()	82	randPoly() (poliAleat)	99
nDerivative()	82	randSamp() (muestAleat)	100
newList() (nuevaLista)	82	RandSeed (SemillaAleat)	100
newMat()	83	real()	100
nfMax()	83	▶Rect	100
nfMin()	83	ref()	101
nInt()	83	remain() (rest)	102
nom()	84	Request (Solicitar)	102
nor	84	RequestStr (SolicitarCad)	103
norm()	84	Return (Producir)	104
normalLine() (líneaNormal)	85	right() (derecha)	104
normCdf() (CdfNormal)	85	rk23()	105
normPdf()	85	root()	105
not	85	rotate()	106
nPr() (prN)	86	round() (redondear)	106
npv() (vpn)	86	rowAdd() (agrFila)	107
nSolve() (solucionN)	86	rowDim() (dimFila)	107
O		rowNorm()	107
OneVar	87	rowSwap() (cambioFila)	107
or	88	rref()	107
ord()	88	S	
P		sec()	108
P▶Rx()	89	sec ⁻¹ ()	108
P▶Ry()	89	sech()	108
PassErr (PasarErr)	89	sech ⁻¹ ()	109
piecewise() (compuestoDeVariables)	89	seq() (secuen)	109
poissCdf()	90	seqGen()	110
poissPdf()	90	seqn()	110
▶Polar	90	series()	111
polyCoeffs()	91	setMode() (configModo)	112
polyDegree() (gradoPoli)	91	shift() (cambiar)	113
polyEval() (evalPoli)	91	sign()	114
polyGcd()	92	simult()	114
polyQuotient() (cocientePoli)	92	▶sin (▶sen)	115
polyRemainder() (restoPoli)	92	sin() (sen)	115
polyRoots() (raicesPoli)	93	sin ⁻¹ () (sen ⁻¹)	116
PowerReg (RegPot)	93	sinh() (senh)	116
Prgm	94	sinh ⁻¹ () (senh ⁻¹)	116
prodSeq()	94	SinReg	117
		solve() (solucion)	118

SortA (OrdenarA)	120
SortD (OrdenarD)	120
►Sphere (►Esfera)	121
sqrt()	121
stat.results (resultados estadísticas)	122
stat.values	123
stDevPop() (desvEstPob)	123
stDevSamp() (desvEstMuest)	123
Stop (Detener)	124
Almacenar	124
string() (cadena)	124
subMat()	124
Suma (Sigma)	124
sum()	125
sumIf() (sumaSi)	125
secSuma()	125
system()	125

T

T (trasponer)	126
tan()	126
tan ⁻¹ ()	127
tangentLine()	127
tanh()	127
tanh ⁻¹ ()	128
taylor()	128
tCdf()	128
tCollect()	129
tExpand()	129
Text	129
Then (Entonces)	129
tInterval (intervaloT)	130
tInterval_2Samp (intervaloT_2Muest)	130
tmpCnv()	131
ΔtmpCnv()	131
tPdf() (PdFT)	131
trace() (trazado)	132
Try (Intentar)	132
tTest (pruebaT)	133
tTest_2Samp (pruebaT_2Muest)	133
tvmFV()	134
tvmI()	134
tvmN()	134
tvmPmt	134
tvmPV()	134
TwoVar (DosVar)	135

U

unitV()	137
unLock (desbloquear)	137

V

varPop()	137
varSamp() (varMuest)	138

W

warnCodes()	138
when() (cuando)	138
While (Mientras)	139

X

xor	139
-----------	-----

Z

zeros()	140
zInterval (intervaloZ)	142
zInterval_1Prop (intervaloZ_1Prop)	142
zInterval_2Prop (intervaloZ_2Prop)	143
zInterval_2Samp (intervaloZ_2Muest)	143
zTest (pruebaZ)	144
zTest_1Prop (pruebaZ_1Prop)	144
zTest_2Prop (pruebaZ_2Prop)	145
zTest_2Samp (pruebaZ_2Muest)	145

Símbolos

+ (agregar)	147
-(sustraer)	147
· (multiplicar)	148
/ (dividir)	148
^ (potencia)	149
x ² (cuadrado)	150
.+ (punto agregar)	150
.- (punto sust.)	150
.. (punto mult.)	150
./ (punto dividir)	151
.^ (punto potencia)	151
-(negar)	151
% (porcentaje)	152
= (igual)	152
≠ (no igual)	153
< (menor que)	153
≤ (menor o igual)	153
> (mayor que)	153
≥ (mayor o igual)	154
⇒ (implicación lógica)	154
⇔ (implicación doble lógica, XNOR)	154
! (factorial)	154
& (adjuntar)	154
d() (derivada)	155
∫() (integral)	155
√() (raíz cuadrada)	156
Π() (secProd)	157
Σ() (secSuma)	157
ΣInt()	158
ΣPrn() (ΣCap)	159
# (indirección)	159
E (notación científica)	159
g (gradián)	160
r (radián)	160
° (grado)	160
°, ', " (grado/minuto/segundo)	161
∠ (ángulo)	161
' (primo)	161
_ (guión bajo como un elemento vacío)	162
_ (guión bajo como designador de unidad)	162
► (convertir)	162
10 [^] ()	162

$^{-1}$ (recíproco)	163
(operador restrictivo)	163
→ (almacenar)	164
:= (asignar)	165
© (comentario)	165
0b, 0h	165

Elementos vacíos (inválidos)

Cálculos que incluyen elementos inválidos	166
Argumentos de lista que contienen elementos inválidos	166

Accesos directos para ingresar expresiones matemáticas

Jerarquía de EOS™ (Sistema Operativo de Ecuaciones)

Códigos y mensajes de error

Códigos y mensajes de advertencia

Soporte y Servicio de Texas Instruments

TI-Nspire™ CAS Guía de Referencia

Esta guía presenta las plantillas, las funciones, los comandos y los operadores disponibles para evaluar expresiones matemáticas.

Plantillas de expresiones

Las plantillas de expresiones ofrecen una manera fácil de ingresar expresiones matemáticas en una notación matemática estándar. Cuando se inserta una plantilla, ésta aparece en la línea de ingreso con pequeños bloques en las posiciones donde se pueden ingresar elementos. Un cursor muestra cuál elemento se puede ingresar.

Use las teclas de flechas o presione **[tab]** para mover el cursor a cada posición del elemento, y escriba un valor o una expresión para el elemento. Presione **[enter]** o **[ctrl] [enter]** para evaluar la expresión.

Plantilla de fracciones

[ctrl] [÷] teclas



Nota: Vea también **/ (dividir)**, página 148.

Ejemplo:

$$\frac{12}{8 \cdot 2} = \frac{3}{4}$$

Plantilla de exponentes

[^] teclas



Nota: Escriba el primer valor, presione **[^]** y después escriba el exponente. Para regresar el cursor a la línea base, presione la flecha derecha **[▶]**.

Nota: Vea también **^ (potencia)**, página 149.

Ejemplo:

$$2^3 = 8$$

Plantilla de raíz cuadrada

[ctrl] [x²] teclas



Nota: Vea también **√()** (**raíz cuadrada**), página 156.

Ejemplo:

$$\sqrt{4} = 2$$
$$\sqrt{\{9, a, 4\}} = \{3, \sqrt{a}, 2\}$$

Plantilla de raíz enésima

[ctrl] [^] teclas



Nota: Vea también **root()**, página 105.

Ejemplo:

$$\sqrt[3]{8} = 2$$
$$\sqrt[3]{\{8, 27, b\}} = \left\{ \frac{1}{2, 3, b} \right\}$$

e plantilla de exponentes

tecla

 e^{\square} Exponencial natural e elevado a una potencia**Nota:** Vea también $e^{\wedge}()$, página 41.

Ejemplo:

e^1	e
$e^1.$	2.71828182846

Plantilla de logística

tecla

 $\log_{\square}(\square)$

Calcula la logística para una base especificada. Para un predeterminado de base 10, omitir la base.

Nota: Vea también $\logistic()$, página 72.

Ejemplo:

$\log_{\downarrow}(2.)$	0.5
-------------------------	-----

Plantilla de compuesto de variables (2 piezas)

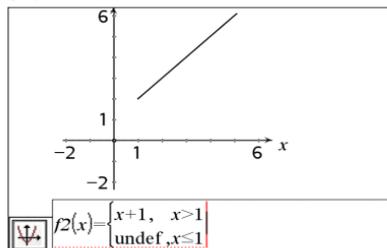
Catálogo >

 $\left\{ \begin{array}{l} \square, \square \\ \square, \square \end{array} \right.$

Permite crear expresiones y condiciones para una función de compuesto de variables de dos-piezas. Para agregar una pieza, haga clic en la plantilla y repita la plantilla.

Nota: Vea también $piecewise()$, página 89.

Ejemplo:

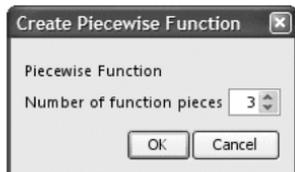
**Plantilla de compuesto de variables (N piezas)**

Catálogo >

Permite crear expresiones y condiciones para una función de compuesto de variables de N -piezas. Indicadores para N .

Ejemplo:

Vea el ejemplo de plantilla de compuesto de variables (2 piezas).

**Nota:** Vea también $piecewise()$, página 89.

Sistema de plantilla de 2 ecuaciones

Catálogo > 



Crea un sistema de dos lineales. Para agregar una fila a un sistema existente, haga clic en la plantilla y repita la plantilla.

Nota: Vea también `system()`, página 125.

Ejemplo:

$$\text{solve}\left(\left\{\begin{array}{l} x+y=0 \\ x-y=5 \end{array}, x, y\right\}, x=\frac{5}{2} \text{ and } y=-\frac{5}{2}\right)$$

$$\text{solve}\left(\left\{\begin{array}{l} y=x^2-2 \\ x+2\cdot y=-1 \end{array}, x, y\right\}, x=-\frac{3}{2} \text{ and } y=\frac{1}{4} \text{ or } x=1 \text{ and } y=-1\right)$$

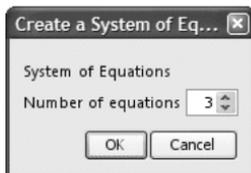
Sistema de plantilla de N ecuaciones

Catálogo > 

Permite crear un sistema de N lineales. Indicadores para N .

Ejemplo:

Vea el ejemplo de Sistema de plantilla de ecuaciones (2 piezas).



Nota: Vea también `system()`, página 125.

Plantilla de valor absoluto

Catálogo > 



Nota: Vea también `abs()`, página 7.

Ejemplo:

$$\left\{ \left| 2, -3, 4, -4^3 \right| \right\} \quad \{ 2, 3, 4, 64 \}$$

plantilla gg°mm'ss.ss"

Catálogo > 



Permite ingresar ángulos en el formato `gg°mm'ss.ss"`, donde **gg** es el número de grados decimales, **mm** es el número de minutos y **ss.ss** es el número de segundos.

Ejemplo:

$$30^\circ 15' 10'' \quad \frac{10891 \cdot \pi}{64800}$$

Plantilla de matriz (2 x 2)

Catálogo > 



Crea una matriz de 2 x 2

Ejemplo:

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \cdot a \quad \begin{bmatrix} a & 2 \cdot a \\ 3 \cdot a & 4 \cdot a \end{bmatrix}$$

Plantilla de matriz (1 x 2)

Catálogo > 

$$\begin{bmatrix} 0 & 0 \end{bmatrix}$$

Ejemplo:

$$\text{crossP}(\begin{bmatrix} 1 & 2 \end{bmatrix}, \begin{bmatrix} 3 & 4 \end{bmatrix}) \quad \begin{bmatrix} 0 & 0 & -2 \end{bmatrix}$$

Plantilla de matriz (2 x 1)

Catálogo > 

$$\begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Ejemplo:

$$\begin{bmatrix} 5 \\ 8 \end{bmatrix} \cdot 0.01 \quad \begin{bmatrix} 0.05 \\ 0.08 \end{bmatrix}$$

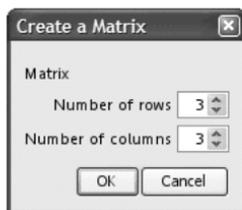
Plantilla de matriz (m x n)

Catálogo > 

La plantilla aparece después de que se le indica especificar el número de filas y columnas.

Ejemplo:

$$\text{diag} \left(\begin{bmatrix} 4 & 2 & 6 \\ 1 & 2 & 3 \\ 5 & 7 & 9 \end{bmatrix} \right) \quad \begin{bmatrix} 4 & 2 & 9 \end{bmatrix}$$



Nota: Si se crea una matriz con un número grande de filas y columnas, puede llevarse unos cuantos segundos en aparecer.

Plantilla de suma (Σ)

Catálogo > 

$$\sum_{i=0}^0 (i)$$

Ejemplo:

$$\sum_{n=3}^7 (n) \quad 25$$

Nota: Vea también $\Sigma()$ (**sumaSec**), página 157.

Plantilla de producto (Π)

Catálogo > 

$$\prod_{i=0}^0 (i)$$

Ejemplo:

$$\prod_{n=1}^5 \left(\frac{1}{n} \right) \quad \frac{1}{120}$$

Nota: Vea también $\Pi()$ (**prodSec**), página 157.

Plantilla de primera derivadaCatálogo > 

$$\frac{d}{dx}(\square)$$

La plantilla de primera derivada también se puede usar para calcular la primera derivada en un punto.

Nota: Vea también **d()** (**derivada**), página 155.

Ejemplo:

$$\frac{d}{dx}(x^3) \quad 3 \cdot x^2$$

$$\frac{d}{dx}(x^3)|_{x=3} \quad 27$$

Plantilla de segunda derivadaCatálogo > 

$$\frac{d^2}{dx^2}(\square)$$

La plantilla de segunda derivada también se puede usar para calcular la segunda derivada en un punto.

Nota: Vea también **d()** (**derivada**), página 155.

Ejemplo:

$$\frac{d^2}{dx^2}(x^3) \quad 6 \cdot x$$

$$\frac{d^2}{dx^2}(x^3)|_{x=3} \quad 18$$

Plantilla de enésima derivadaCatálogo > 

$$\frac{d^{\square}}{d\square^{\square}}(\square)$$

La plantilla de enésima derivada se puede usar para calcular la enésima derivada.

Nota: Vea también **d()** (**derivada**), página 155.

Ejemplo:

$$\frac{d^3}{dx^3}(x^3)|_{x=3} \quad 6$$

Plantilla de integral definidaCatálogo > 

$$\int_{\square}^{\square} \square d\square$$

Nota: Vea también **∫()** (**integral()**), página 155.

Ejemplo:

$$\int_a^b x^2 dx \quad \frac{b^3}{3} - \frac{a^3}{3}$$

Plantilla de integral indefinidaCatálogo > 

$$\int \square d\square$$

Nota: Vea también **∫()** (**integral()**), página 155.

Ejemplo:

$$\int x^2 dx \quad \frac{x^3}{3}$$

$$\lim_{\square \rightarrow \square} (\square)$$

Ejemplo:

$$\lim_{x \rightarrow 5} (2 \cdot x + 3) = 13$$

Use - o (-) para el límite de la izquierda. Use + para el límite de la derecha.

Nota: Vea también **limit()**, página 64.

Listado alfabético

Los elementos cuyos nombres no son alfabéticos (como +, ! y >) se enumeran al final de esta sección, comenzando en la página 147. A menos que se especifique lo contrario, todos los ejemplos en esta sección se realizaron en el modo de reconfiguración predeterminado, y se supone que todas las variables no están definidas.

A

abs()

Catálogo >

abs(Expr1) ⇒ expresión

abs(Lista1) ⇒ lista

abs(Matriz1) ⇒ matriz

Entrega el valor absoluto del argumento.

Nota: Vea también **Plantilla de valor absoluto**, página 3.

Si el argumento es un número complejo, entrega el módulo del número.

Nota: Todas las variables indefinidas se tratan como variables reales.

$\left \left\{ \frac{\pi}{2}, \frac{\pi}{3} \right\} \right $	$\left\{ \frac{\pi}{2}, \frac{\pi}{3} \right\}$
$ 2-3 \cdot i $	$\sqrt{13}$
$ z $	$ z $
$ x+y \cdot i $	$\sqrt{x^2+y^2}$

amortTbl() (tablaAmort)

Catálogo >

amortTbl(NPgo, N, I, VP, [Pgo], [VF], [PpA], [CpA], [PgoAl], [valorRedondo]) ⇒ matriz

La función de amortización que entrega una matriz como una tabla de amortización para un conjunto de argumentos de TVM.

NPgo es el número de pagos a incluirse en la tabla. La tabla comienza con el primer pago.

N, I, VP, Pgo, VF, PpA, CpA, and PgoAl se describen en la tabla de argumentos de VTD, página 135.

- Si se omite Pgo, se predetermina a $Pgo = \mathbf{tvmPmt}(N, I, VP, VF, PpA, CpA, PgoAl)$.
- Si se omite VF, se predetermina a $VF = 0$.
- Los predeterminados para PpA, CpA y PgoAl son los mismos que para las funciones de TVM.

valorRedondo especifica el número de lugares decimales para el redondeo. Predeterminado=2.

Las columnas en la matriz de resultado están en este orden: Número de pago, cantidad pagada a interés, cantidad pagada a capital y balance.

El balance desplegado en la fila n es el balance después del pago n.

Se puede usar la matriz de salida como entrada para las otras funciones de amortización $\Sigma \mathbf{Int}()$ y $\Sigma \mathbf{Prn}()$, página 158y **bal()**, página 13.

amortTbl(12,60,10,5000,,,12,12)

0	0.	0.	5000.
1	-41.67	-64.57	4935.43
2	-41.13	-65.11	4870.32
3	-40.59	-65.65	4804.67
4	-40.04	-66.2	4738.47
5	-39.49	-66.75	4671.72
6	-38.93	-67.31	4604.41
7	-38.37	-67.87	4536.54
8	-37.8	-68.44	4468.1
9	-37.23	-69.01	4399.09
10	-36.66	-69.58	4329.51
11	-36.08	-70.16	4259.35
12	-35.49	-70.75	4188.6

and (y)

Catálogo >

ExprBooleana1 and ExprBooleana2 ⇒ expresión Booleana

ListaBooleana1 and ListaBooleana2 ⇒ Lista Booleana

MatrizBooleana1 and MatrizBooleana2 ⇒ Matriz Booleana

Entrega verdadero o falso o una forma simplificada del ingreso original.

$x \geq 3$ and $x \geq 4$	$x \geq 4$
$\{x \geq 3, x \leq 0\}$ and $\{x \geq 4, x \leq -2\}$	$\{x \geq 4, x \leq -2\}$

and (y)

Catálogo >

Entero1 and Entero2 \Rightarrow entero

Compara dos enteros reales bit por bit usando una operación **y**. En forma interna, ambos enteros se convierten en números binarios de 64 bits firmados. Cuando se comparan los bits correspondientes, el resultado es 1 si ambos bits son 1; de otro modo, el resultado es 0. El valor producido representa los resultados de los bits, y se despliega de acuerdo con el modo de Base.

Se pueden ingresar enteros en cualquier base de números. Para un ingreso binario o hexadecimal, se debe usar el prefijo 0b ó 0h, respectivamente. Sin un prefijo, los enteros se tratan como decimales (base 10).

En modo de base hexadecimal:

0h7AC36 and 0h3D5F 0h2C16

Importante: Cero, no la letra 0.

En modo de base binaria:

0b100101 and 0b100 0b100

En modo de base decimal:

37 and 0b100 4

Nota: Un ingreso binario puede tener hasta 64 dígitos (sin contar el prefijo 0b). Un ingreso hexadecimal puede tener hasta 16 dígitos.

angle()

Catálogo >

angle(Expr1) \Rightarrow expresión

Entrega el ángulo del argumento, interpretando el argumento como un número complejo.

Nota: Todas las variables indefinidas se tratan como variables reales.

En modo de ángulo en Grados:

angle(0+2*i*) 90

En modo de ángulo en Gradianes:

angle(0+3*i*) 100

En modo de ángulo en Radianes:

angle(1+i) $\frac{\pi}{4}$

angle(z) $\frac{-\pi \cdot (\text{sign}(z)-1)}{2}$

angle(x+i·y) $\frac{\pi \cdot \text{sign}(y)}{2} - \tan^{-1}\left(\frac{x}{y}\right)$

angle({{1+2*i*, 3+0*i*, 0-4*i*}}) $\left\{ \frac{\pi}{2}, -\tan^{-1}\left(\frac{1}{2}\right), 0, \frac{\pi}{2} \right\}$

angle(Lista1) \Rightarrow lista

angle(Matriz1) \Rightarrow matriz

Entrega una lista o matriz de ángulos de los elementos en *Lista1* o *Matriz1*, interpretando cada elemento como un número complejo que representa un punto de coordenada bidimensional o rectangular.

ANOVA

Catálogo >

ANOVA Lista1, Lista2[, Lista3, ..., Lista20][, Bandera]

Realiza un análisis unidireccional de la varianza para comparar las medias de dos a 20 poblaciones. Un resumen de resultados se almacena en la variable *stat.results*. (Vea la página 122.)

Bandera=0 para Datos, Bandera=1 para Estadísticas

Variable de salida	Descripción
stat.F	Valor de F estadístico
stat.ValP	Nivel más bajo de significancia en el cual la hipótesis nula se puede rechazar
stat.df	Grados de libertad de los grupos

Variable de salida	Descripción
stat.SS	Suma de cuadrados de los grupos
stat.MS	Cuadrados medios de los grupos
stat.dfError	Grados de libertad de los errores
stat.SSError	Suma de cuadrados de los errores
stat.MSError	Cuadrado medio de los errores
stat.sp	Desviación estándar agrupada
stat.xbarlista	Media de la entrada de las listas
stat.ListaCBajo	95% de intervalos de confianza para la media de cada lista de entrada
stat.ListaCAlto	95% de intervalos de confianza para la media de cada lista de entrada

ANOVA2way (ANOVA2vías)

Catálogo > 

ANOVA2way Lista1,Lista2[,Lista3,...,Lista10][,LevRow]

Genera un análisis bidireccional de la varianza para comparar las medias de dos a 10 poblaciones. Un resumen de resultados se almacena en la variable *stat.results*. (Vea la página 122.)

LevRow=0 para bloque

LevRow=2,3,...,Len-1, para factor dos, donde

Len=largo(Lista1)=largo(Lista2) = ... = largo(Lista10) y *Len* l

LevRow ∈ {2,3,...}

Salidas: Diseño de bloque

Variable de salida	Descripción
stat.F	F estadístico del factor de columna
stat.ValP	Nivel más bajo de significancia en el cual la hipótesis nula se puede rechazar
stat.df	Grados de libertad del factor de columna
stat.SS	Suma de cuadrados del factor de columna
stat.MS	Cuadrados medios para el factor de columna
stat.BloqF	F estadístico para el factor
stat.BloqValP	Probabilidad más baja a la cual la hipótesis nula se puede rechazar
stat.dfBloque	Grados de libertad del factor
stat.SSBloque	Suma de cuadrados para el factor
stat.MSBloque	Cuadrados medios para el factor
stat.dfError	Grados de libertad de los errores
stat.SSError	Suma de cuadrados de los errores
stat.MSError	Cuadrados medios para los errores
stat.s	Desviación estándar del error

Salidas del FACTOR DE COLUMNA

Variable de salida	Descripción
stat.Fcol	F estadístico del factor de columna
stat.ValPCol	Valor de probabilidad del factor de columna
stat.dfCol	Grados de libertad del factor de columna
stat.SSCol	Suma de cuadrados del factor de columna
stat.MSCol	Cuadrados medios para el factor de columna

Salidas del FACTOR DE FILAS

Variable de salida	Descripción
stat.FFila	F estadístico del factor de fila
stat.ValPFila	Valor de probabilidad del factor de fila
stat.dfFila	Grados de libertad del factor de fila
stat.SSFila	Suma de cuadrados del factor de fila
stat.MSFila	Cuadrados medios para el factor de fila

Salidas de INTERACCIÓN

Variable de salida	Descripción
stat.FInterac	F estadístico de la interacción
stat.ValPInterac	Valor de probabilidad de la interacción
stat.dfInterac	Grados de libertad de la interacción
stat.SSInterac	Suma de cuadrados de la interacción
stat.MSInterac	Cuadrados medios para la interacción

Salidas de ERROR

Variable de salida	Descripción
stat.dfError	Grados de libertad de los errores
stat.SSError	Suma de cuadrados de los errores
stat.MSError	Cuadrados medios para los errores
s	Desviación estándar del error

Ans

ctrl (-) teclas

Ans \Rightarrow *valor*

Entrega el resultado de la expresión evaluada más recientemente.

56	56
56+4	60
60+4	64

approx()

Catálogo >

approx(*Expr1*) \Rightarrow *expresión*Entrega la evaluación del argumento como una expresión que contiene valores decimales, cuando es posible, independientemente del modo **Auto** o **Aproximado** actual.

Esto es equivalente a ingresar el argumento y presionar

.

$\text{approx}\left(\frac{1}{3}\right)$	0.333333
$\text{approx}\left\{\left\{\frac{1}{3}, \frac{1}{9}\right\}\right\}$	{0.333333, 0.111111}
$\text{approx}\{\{\sin(\pi), \cos(\pi)\}\}$	{0., -1.}
$\text{approx}(\{\sqrt{2}, \sqrt{3}\})$	[1.41421 1.73205]
$\text{approx}\left(\left\{\frac{1}{3}, \frac{1}{9}\right\}\right)$	[0.333333 0.111111]
$\text{approx}\{\{\sin(\pi), \cos(\pi)\}\}$	{0., -1.}
$\text{approx}(\{\sqrt{2}, \sqrt{3}\})$	[1.41421 1.73205]

approx(*Lista1*) \Rightarrow *lista***approx**(*Lista1*) \Rightarrow *lista*Entrega una lista o *matriz* donde cada elemento se ha evaluado a un valor decimal, cuando es posible.**▶approxFraction()**

Catálogo >

Expr ▶**approxFraction**(*[Tol]*) \Rightarrow *expresión**Lista* ▶**approxFraction**(*[Tol]*) \Rightarrow *lista**Matriz* ▶**approxFraction**(*[Tol]*) \Rightarrow *matriz*Entrega la entrada como una fracción, usando una tolerancia de *Tol*. Si *Tol* se omite, se usa una tolerancia de 5.E-14.**Nota:** Se puede insertar esta función desde el teclado de la computadora al escribir @>**approxFraction**(...).

$\frac{1}{2} + \frac{1}{3} + \tan(\pi)$	0.833333
$0.8333333333333333 \blacktriangleright \text{approxFraction}(5.E-14)$	$\frac{5}{6}$
$\{\pi, 1.5\} \blacktriangleright \text{approxFraction}(5.E-14)$	$\left\{\frac{5419351}{1725033}, \frac{3}{2}\right\}$

approxRational()

Catálogo >

approxRational(*Expr1*, *Tol*) \Rightarrow *expresión***approxRational**(*Lista1*, *Tol*) \Rightarrow *lista***approxRational**(*Matriz1*, *Tol*) \Rightarrow *matriz*Entrega el argumento como una fracción usando una tolerancia de *Tol*. Si *Tol* se omite, se usa una tolerancia de 5.E-14.

$\text{approxRational}(0.333, 5 \cdot 10^{-5})$	$\frac{333}{1000}$
$\text{approxRational}(\{0.2, 0.33, 4.125\}, 5.E-14)$	$\left\{\frac{1}{5}, \frac{33}{100}, \frac{33}{8}\right\}$

arccos()Vea \cos^{-1} , página 24.

arccosh() Vea \cosh^{-1} , página 25.

arccot() Vea \cot^{-1} , página 25.

arccoth() Vea \coth^{-1} , página 26.

arccsc() Vea \csc^{-1} , página 28.

arcsch() Vea csch^{-1} , página 28.

arcLen() Catálogo > 

arcLen(*Expr1*, *Var*, *Iniciar*, *Terminar*) \Rightarrow expresión

Entrega la longitud de arco de *Expr1* desde *Iniciar* a *Terminar* con respecto de la variable *Var*.

La longitud de arco se calcula como una integral suponiendo una definición de modo de función.

$$\begin{array}{l} \text{arcLen}(\cos(x), x, 0, \pi) \quad 3.8202 \\ \text{arcLen}(f(x), x, a, b) \quad \int_a^b \sqrt{\left(\frac{d}{dx}(f(x))\right)^2 + 1} dx \end{array}$$

arcLen(*Lista1*, *Var*, *Iniciar*, *Terminar*) \Rightarrow lista

Entrega una lista de longitudes de arco de cada elemento de *Lista1* desde *Iniciar* hasta *Terminar* con respecto de *Var*.

$$\text{arcLen}(\{\sin(x), \cos(x)\}, x, 0, \pi) \quad \{3.8202, 3.8202\}$$

arcsec() Vea \sec^{-1} , página 108.

arcsech() Vea sech , página 109.

arcsin() Vea \sin , página 116.

arsinh() Vea \sinh , página 116.

arctan() Vea \tan , página 127.

arctanh() Vea \tanh , página 128.

augment() Catálogo > 

augment(*Lista1*, *Lista2*) \Rightarrow lista

Entrega una nueva lista que es *Lista2* adjuntada al final de *Lista1*.

$$\text{augment}(\{1, -3, 2\}, \{5, 4\}) \quad \{1, -3, 2, 5, 4\}$$

augment()

Catálogo >

augment(*Matriz1*, *Matriz2*) ⇒ *matriz*

Entrega una nueva matriz que es *Matriz2* adjuntada a *Matriz1*. Cuando se usa el carácter " , " las matrices deben tener dimensiones de fila iguales, y *Matriz2* se adjunta a *Matriz1* como nuevas columnas. No altera *Matriz1* o *Matriz2*.

$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \rightarrow m1$	$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$
$\begin{bmatrix} 5 \\ 6 \end{bmatrix} \rightarrow m2$	$\begin{bmatrix} 5 \\ 6 \end{bmatrix}$
augment (<i>m1</i> , <i>m2</i>)	$\begin{bmatrix} 1 & 2 & 5 \\ 3 & 4 & 6 \end{bmatrix}$

avgRC()

Catálogo >

avgRC(*Expr1*, *Var* [=Valor] [, *Paso*]) ⇒ *expresión***avgRC**(*Expr1*, *Var* [=Valor] [, *Lista1*]) ⇒ *lista***avgRC**(*Lista1*, *Var* [=Valor] [, *Paso*]) ⇒ *lista***avgRC**(*Matriz1*, *Var* [=Valor] [, *Paso*]) ⇒ *matriz*

Entrega el cociente diferencial progresivo (tasa de cambio promedio).

Expr1 puede ser un nombre de función definido por el usuario (vea **Func**).Cuando se especifica el *Valor*, se eliminan todas las asignaciones anteriores de la variable o cualquier sustitución "|" para la variable.*Paso* es el valor del paso. Si se omite *Paso* se predetermina a 0.001.Tome en cuenta que la función similar **centralDiff()** usa el cociente diferencial central.

avgRC ($f(x)$, x , h)	$\frac{f(x+h)-f(x)}{h}$
avgRC ($\sin(x)$, x , h) $ x=2$	$\frac{\sin(h+2)-\sin(2)}{h}$
avgRC (x^2-x+2 , x)	$2 \cdot (x-0.4995)$
avgRC (x^2-x+2 , x ,0.1)	$2 \cdot (x-0.45)$
avgRC (x^2-x+2 , x ,3)	$2 \cdot (x+1)$

B**bal()**

Catálogo >

bal(*NPgo*,*N*,*I*,*VP* [,*Pgo*], [*VF*], [*PpA*], [*CpA*], [*PgoAl*], [*valorRedondo*]) ⇒ *valor***bal**(*NPgo*,*tablaAmort*) ⇒ *valor*

Función de amortización que calcula el balance del programa después de un pago especificado.

N, *I*, *VP*, *Pgo*, *VF*, *PpA*, *CpA* y *PgoAl* se describen en la tabla de argumentos de VTD, página 135.*NPgo* especifica el número de pago después del cual usted desea que los datos se calculen.*N*, *I*, *VP*, *Pgo*, *VF*, *PpA*, *CpA* y *PgoAl* se describen en la tabla de argumentos de VTD, página 135.

- Si se omite *Pgo*, se predetermina a $Pgo = \mathbf{tvmPmt}(N, I, VP, VF, PpA, CpA, PgoAl)$.
- Si se omite *VF*, se predetermina a *VF*=0.
- Los predeterminados para *PpA*, *CpA* y *PgoAl* son los mismos que para las funciones de VTD.

valorRedondo especifica el número de lugares decimales para el redondeo. Predeterminado=2.**bal**(*NPgo*,*tablaAmort*) calcula el balance después del número de pago *NPgo*, basado en la tabla de amortización *tablaAmort*. El argumento *tablaAmort* debe ser una matriz en la forma descrita bajo **amortTbl()**, página 2.**Nota:** Vea también **ΣInt()** y **ΣPrn()**, página 158.

bal (5,6,5.75,5000,,12,12)	833.11
tbl:=amortTbl (6,6,5.75,5000,,12,12)	
$\begin{bmatrix} 0 & 0. & 0. & 5000. \\ 1 & -23.35 & -825.63 & 4174.37 \\ 2 & -19.49 & -829.49 & 3344.88 \\ 3 & -15.62 & -833.36 & 2511.52 \\ 4 & -11.73 & -837.25 & 1674.27 \\ 5 & -7.82 & -841.16 & 833.11 \\ 6 & -3.89 & -845.09 & -11.98 \end{bmatrix}$	
bal (4, tbl)	1674.27

Entero | ►Base2 ⇒ entero

Nota: Se puede insertar este operador desde el teclado de la computadora al escribir @>Base2.

Convierte *Entero1* en un número binario. Los número binarios o hexadecimales siempre tienen un prefijo 0b ó 0h, respectivamente.

— Cero, no la letra O, seguida de b o de h.

0b númeroBinario

0h númeroHexadecimal

— Un número binario puede tener hasta 64 dígitos.
Un número hexadecimal puede tener hasta 16.

Sin un prefijo, *Entero1* se trata como decimal (base 10). El resultado se despliega en binario, independientemente del modo de la Base.

Los números negativos se despliegan en forma de "complemento de dos". Por ejemplo:

-1 se despliega como

0hFFFFFFFFFFFFFF en modo de base Hexadecimal

0b111...111 (64 1's) en modo de base Binaria

-2⁶³ se despliega como

0h8000000000000000 en modo de base Hexadecimal

0b100...000 (63 ceros) en modo de base Binaria

Si se ingresa un entero decimal que está fuera del rango de una forma binaria de 64 bits firmada, se usa una operación de módulo simétrico para llevar el valor al rango apropiado. Considere los siguientes ejemplos de valores fuera del rango.

2⁶³ se convierte en -2⁶³ y se despliega como

0h8000000000000000 en modo de base Hexadecimal

0b100...000 (63 ceros) en modo de base Binaria

2⁶⁴ se convierte en 0 y se despliega como

0h0 en modo de base Hexadecimal

0b0 en modo de base Binaria

-2⁶³ - 1 se convierte en 2⁶³ - 1 y se despliega como

0h7FFFFFFFFFFFFFFF en modo de base Hexadecimal

0b111...111 (64 1's) en modo de base Binaria

256 ►Base2	0b100000000
0h1F ►Base2	0b11111

Entero | ►Base10 ⇒ entero

Nota: Se puede insertar este operador desde el teclado de la computadora al escribir @>Base10.

Convierte *Integer1* en un número decimal (base 10). El ingreso binario o hexadecimal siempre debe tener un prefijo 0b ó 0h, respectivamente.

0b númeroBinario

0h númeroHexadecimal

Cero, no la letra O, seguida de b o de h.

Un número binario puede tener hasta 64 dígitos. Un número hexadecimal puede tener hasta 16.

Sin un prefijo, *Integer1* se trata como decimal. El resultado se despliega en decimal, independientemente del modo de la Base.

0b10011 ►Base10	19
0h1F ►Base10	31

►Base16

Catálogo >

Entero1 ►Base16 ⇒ entero**Nota:** Se puede insertar este operador desde el teclado de la computadora al escribir @>Base16.Convierte *Entero1* en un número hexadecimal. Los números binarios y hexadecimales siempre tienen un prefijo 0b ó 0h, respectivamente.

0b númeroBinario

0h númeroHexadecimal

Cero, no la letra O, seguida de b o de h.

Un número binario puede tener hasta 64 dígitos. Un número hexadecimal puede tener hasta 16.

Sin un prefijo, *Integer1* se trata como decimal (base 10). El resultado se despliega en hexadecimal, independientemente del modo de la Base.

Si se ingresa un entero decimal que es demasiado grande para una forma binaria de 64 bits firmada, se usa una operación de módulo simétrico para llevar el valor al rango apropiado. Para obtener más información, vea ►Base2, página 14.

256►Base16	0h100
0b111100001111►Base16	0hF0F

binomCdf()

Catálogo >

binomCdf(*n,p*) ⇒ número**binomCdf**(*n,p,limiteInferior,limiteSuperior*) ⇒ número si *limiteInferior* y *limiteSuperior* son números, *lista* si *limiteInferior* y *limiteSuperior* son listas**binomCdf**(*n,p,limiteSuperior*) para $P(0 \leq X \leq \text{limiteSuperior})$ ⇒ número si *limiteSuperior* es un número, *lista* si *limiteSuperior* es una listaGenera una probabilidad acumulativa para la distribución binómica discreta con *n* número de pruebas y probabilidad *p* de éxito en cada prueba.Para $P(X \leq \text{limiteSuperior})$, configure *limiteInferior*=0**binomPdf()**

Catálogo >

binomPdf(*n,p*) ⇒ número**binomPdf**(*n,p,XVal*) ⇒ número si *XVal* es un número, *lista* si *XVal* es una listaGenera una probabilidad para la distribución binómica discreta con *n* número de pruebas y probabilidad *p* de éxito en cada prueba.**C****ceiling() (techo)**

Catálogo >

ceiling(*Expr1*) ⇒ enteroEntrega el entero más cercano que es \geq el argumento.

El argumento puede ser un número real o complejo.

Nota: Vea también **floor()**.**ceiling**(*Lista1*) ⇒ lista**ceiling**(*Matriz1*) ⇒ matriz

Entrega una lista o matriz del techo de cada elemento.

ceiling (.456)	1.
-----------------------	----

ceiling {{-3.1,1,2.5}}	{-3.,1,3.}
ceiling $\left(\begin{matrix} 0 & -3.2 \cdot i \\ 1.3 & 4 \end{matrix}\right)$	$\begin{matrix} 0 & -3 \cdot i \\ 2. & 4 \end{matrix}$

centralDiff()

Catálogo >

centralDiff(*Expr1*, *Var* [=Valor], [*Paso*]) ⇒ expresión**centralDiff**(*Expr1*, *Var* [, *Paso*]) | *Var* = Valor ⇒ expresión**centralDiff**(*Expr1*, *Var* [=Valor], [*Lista*]) ⇒ lista**centralDiff**(*Lista1*, *Var* [=Valor], [*Paso*]) ⇒ lista**centralDiff**(*Matriz1*, *Var* [=Valor], [*Paso*]) ⇒ matriz

Entrega la derivada numérica usando la fórmula del cociente diferencial central.

Cuando se especifica el *Valor*, se eliminan todas las asignaciones anteriores de la variable o cualquier sustitución "I" para la variable.*Paso* es el valor del paso. Si se omite *Paso*, se predetermina a 0.001.Al usar *Lista1* o *Matriz1*, la operación se mapea a lo largo de los valores en la lista y a lo largo de los elementos de la matriz.**Nota:** Vea también **avgRC()** y **d()**.

$$\frac{\text{centralDiff}(\cos(x), x, h) - (\cos(x-h) - \cos(x+h))}{2 \cdot h}$$

$$\lim_{h \rightarrow 0} (\text{centralDiff}(\cos(x), x, h)) = -\sin(x)$$

$$\text{centralDiff}(x^3, x, 0.01) = 3 \cdot (x^2 + 0.000033)$$

$$\text{centralDiff}(\cos(x), x) \Big|_{x=\frac{\pi}{2}} = -1$$

$$\text{centralDiff}(x^2, x, \{0.01, 0.1\}) = \{2 \cdot x, 2 \cdot x\}$$

cFactor()

Catálogo >

cFactor(*Expr1*, *Var*) ⇒ expresión**cFactor**(*Lista1*, *Var*) ⇒ lista**cFactor**(*Matriz1*, *Var*) ⇒ matriz**cFactor**(*Expr1*) entrega *Expr1* factorizado con respecto de todas sus variables sobre un denominador común.*Expr1* se factoriza tanto como es posible hacia los factores racionales lineales, incluso si esto introduce nuevos número no reales. Esta alternativa es apropiada si se desea una factorización con respecto de más de una variable.**cFactor**(*Expr1*, *Var*) entrega *Expr1* factorizado con respecto de la variable *Var*.*Expr1* se factoriza tanto como es posible hacia factores que son lineales en *Var*, quizá con constantes no reales, incluso si esto introduce constantes irracionales o subexpresiones que son irracionales en otras variables.Los factores y sus términos se clasifican con *Var* como la variable principal. Se recopilan potencias similares de *Var* en cada factor. Incluya *Var* si se necesita la factorización con respecto de sólo esa variable y usted está dispuesto a aceptar expresiones irracionales en otras variables para incrementar la factorización con respecto de *Var*. Podría haber cierta factorización incidental con respecto de otras variables.Para la configuración automática del modo **Auto** o **Aproximado**, incluyendo *Var*, también permite la aproximación con coeficientes de punto flotante, donde los coeficientes irracionales no se pueden expresar en forma explícita concisamente en términos de funciones integradas. Incluso cuando hay sólo una variable, incluyendo *Var*, puede producir una factorización más completa.**Nota:** Vea también **factor()**.

$$\text{cFactor}(a^3 \cdot x^2 + a \cdot x^2 + a^3 + a) = a \cdot (a+i) \cdot (a+i) \cdot (x+i) \cdot (x+i)$$

$$\text{cFactor}\left(x^2 + \frac{4}{9}\right) = \frac{(3 \cdot x + 2 \cdot i) \cdot (3 \cdot x + 2 \cdot i)}{9}$$

$$\text{cFactor}(x^2 + 3) = x^2 + 3$$

$$\text{cFactor}(x^2 + a) = x^2 + a$$

$$\text{cFactor}(a^3 \cdot x^2 + a \cdot x^2 + a^3 + a \cdot x) = a \cdot (a^2 + 1) \cdot (x+i) \cdot (x+i)$$

$$\text{cFactor}(x^2 + 3, x) = (x + \sqrt{3} \cdot i) \cdot (x + \sqrt{3} \cdot i)$$

$$\text{cFactor}(x^2 + a, x) = (x + \sqrt{a} \cdot i) \cdot (x + \sqrt{a} \cdot i)$$

$$\text{cFactor}(x^5 + 4 \cdot x^4 + 5 \cdot x^3 - 6 \cdot x - 3) = x^5 + 4 \cdot x^4 + 5 \cdot x^3 - 6 \cdot x - 3$$

$$\text{cFactor}(x^5 + 4 \cdot x^4 + 5 \cdot x^3 - 6 \cdot x - 3, x) = (x - 0.964673) \cdot (x + 0.611649) \cdot (x + 2.12543) \cdot (x^2 + \dots)$$

Para ver el resultado completo, presione y luego use para mover el cursor.

char()

Catálogo >

char(Entero) \Rightarrow caracter

Entrega una cadena de caracteres que contiene el caracter numerado *Entero* desde el conjunto de caracteres del dispositivo portátil. El rango válido para *Entero* es 0–65535.

char (38)	"&"
char (65)	"A"

charPoly()

Catálogo >

charPoly(matrizCuadrada, Var) \Rightarrow expresión polinómica**charPoly**(matrizCuadrada, Expr) \Rightarrow expresión polinómica**charPoly**(matrizCuadrada1, Matriz2) \Rightarrow expresión polinómica

Entrega el polinomio característico de *matrizCuadrada*. El polinomio característico de $n \times n$ matriz *A*, denotado por $p_A(\lambda)$, es el polinomio definido por

$$p_A(\lambda) = \det(\lambda \cdot I - A)$$

donde *I* denota la matriz de identidad $n \times n$.

matrizCuadrada1 y *matrizCuadrada2* deben tener dimensiones iguales.

$m := \begin{bmatrix} 1 & 3 & 0 \\ 2 & -1 & 0 \\ -2 & 2 & 5 \end{bmatrix}$	$\begin{bmatrix} 1 & 3 & 0 \\ 2 & -1 & 0 \\ -2 & 2 & 5 \end{bmatrix}$
charPoly (<i>m</i> , <i>x</i>)	$-x^3 + 5 \cdot x^2 + 7 \cdot x - 35$
charPoly (<i>m</i> , $x^2 + 1$)	$-x^6 + 2 \cdot x^4 + 14 \cdot x^2 - 24$
charPoly (<i>m</i> , <i>m</i>)	0

 χ^2 2way

Catálogo >

 χ^2 2way matrizObs**chi22way** matrizObs

Resuelve una prueba χ^2 para la asociación en la tabla bidireccional de conteos en la matriz observada *matrizObs*. Un resumen de resultados se almacena en la variable *stat.results*. (Vea la página 122.)

Para obtener información sobre el efecto de los elementos vacíos en una matriz, vea "Elementos vacíos (inválidos)" en la página 166.

Variable de salida	Descripción
stat. χ^2	Estadísticas cuadradas de Ji: suma (observada - esperada) ² /esperada
stat.ValP	Nivel más bajo de significancia en el cual la hipótesis nula se puede rechazar
stat.df	Grados de libertad para las estadísticas cuadradas de ji
stat.ExpMat	Matriz de tabla de conteo elemental esperada, suponiendo una hipótesis nula
stat.CompMat	Matriz de contribuciones de estadísticas cuadradas de ji elementales

χ^2 Cdf()Catálogo > 

χ^2 Cdf(*limiteInferior*,*limiteSuperior*,*df*) \Rightarrow número si *limiteInferior* y *limiteSuperior* son números, lista si *limiteInferior* y *limiteSuperior* son listas

chi2Cdf(*limiteInferior*,*limiteSuperior*,*df*) \Rightarrow número si *limiteInferior* y *limiteSuperior* son números, lista si *limiteInferior* y *limiteSuperior* son listas

Genera la probabilidad de distribución χ^2 entre *limiteInferior* y *limiteSuperior* para grados específicos de libertad *df*.

Para $P(X \leq \text{limiteSuperior})$, configure *limiteInferior* = 0.

Para obtener información sobre el efecto de los elementos vacíos en una lista, vea "Elementos vacíos (inválidos)" en la página 166.

 χ^2 GOFCatálogo > 

χ^2 GOF *listaObs*,*listaExp*,*df*

chi2GOF *listaObs*,*listaExp*,*df*

Realiza una prueba para confirmar que los datos de la muestra son de una población que cumple con una distribución especificada.

listaObs es una lista de conteos y debe contener enteros. Un resumen de resultados se almacena en la variable *stat.results*. (Vea la página 122.)

Para obtener información sobre el efecto de los elementos vacíos en una lista, vea "Elementos vacíos (inválidos)" en la página 166.

Variable de salida	Descripción
stat. χ^2	Estadísticas cuadradas de Ji: $\text{suma}((\text{observada} - \text{esperada})^2 / \text{esperada})$
stat.ValP	Nivel más bajo de significancia en el cual la hipótesis nula se puede rechazar
stat.df	Grados de libertad para las estadísticas cuadradas de ji
stat.ListaComp	Contribuciones de estadísticas cuadradas de ji elementales

 χ^2 Pdf()Catálogo > 

χ^2 Pdf(*XVal*,*df*) \Rightarrow número si *XVal* es un número, lista si *XVal* es una lista

chi2Pdf(*XVal*,*df*) \Rightarrow número si *XVal* es un número, lista si *XVal* es una lista

Genera la función de densidad de probabilidad (pdf) para la distribución χ^2 a un valor especificado *XVal* para los grados de libertad especificados *df*.

Para obtener información sobre el efecto de los elementos vacíos en una lista, vea "Elementos vacíos (inválidos)" en la página 166.

ClearAZ (LimpiarAZ)Catálogo > **ClearAZ**

Limpia todas las variables de carácter único en el espacio del problema actual.

Si una o más de las variables están bloqueadas, este comando despliega un mensaje de error y borra únicamente las variables no bloqueadas. Vea **unLock**, página 137.

$5 \rightarrow b$	5
b	5
ClearAZ	Done
b	b

ClrErr (LimpErr)Catálogo > **ClrErr**

Limpia el estado del error y configura *Codigerr* de la variable del sistema a cero.

La cláusula **Else** del bloque **Try...Else...EndTry** debe usar **ClrErr** o **PassErr**. Si el error se debe procesar o ignorar, use **ClrErr**. Si no se sabe qué hacer con el error, use **PassErr** para enviarlo al siguiente manipulador de errores. Si no hay ningún otro manipulador de errores **Try...Else...EndTry** pendiente, el cuadro de diálogo de error se desplegará como normal.

Nota: Vea también **PasErr**, página 89, y **Try**, página 132.

Nota para ingresar el ejemplo: En la aplicación de Calculadora en el dispositivo portátil, usted puede ingresar definiciones en líneas múltiples al presionar  en lugar de **enter** al final de cada línea. En el teclado de la computadora, presione y sostenga **Alt** y presione **Ingresar**.

Para consultar un ejemplo de **ClrErr**, vea el Ejemplo 2 bajo el comando **Try**, página 132.

colAugment()Catálogo > 

colAugment(*Matriz1*, *Matriz2*) \Rightarrow *matriz*

Entrega una nueva matriz que es *Matriz2* adjuntada a *Matriz1*. Las matrices deben tener dimensiones de columna iguales, y *Matriz2* se adjunta a *Matriz1* como nuevas filas. No altera *Matriz1* o *Matriz2*.

$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \rightarrow m1$	$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$
$\begin{bmatrix} 5 & 6 \end{bmatrix} \rightarrow m2$	$\begin{bmatrix} 5 & 6 \end{bmatrix}$
colAugment ($m1, m2$)	$\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}$

colDim()Catálogo > 

colDim(*Matriz*) \Rightarrow *expresión*

Entrega el número de columnas contenidas en *Matriz*.

Nota: Vea también **rowDim**().

colDim $\left(\begin{bmatrix} 0 & 1 & 2 \\ 3 & 4 & 5 \end{bmatrix}\right)$	3
---	---

colNorm()Catálogo > 

colNorm(*Matriz*) \Rightarrow *expresión*

Entrega el máximo de las sumas de los valores absolutos de los elementos en las columnas en *Matriz*.

Nota: Los elementos de matriz indefinida no están permitidos. Vea también **rowNorm**().

$\begin{bmatrix} 1 & -2 & 3 \\ 4 & 5 & -6 \end{bmatrix} \rightarrow mat$	$\begin{bmatrix} 1 & -2 & 3 \\ 4 & 5 & -6 \end{bmatrix}$
colNorm (<i>mat</i>)	9

comDenom(*Expr1*,*Var*) ⇒ *expresión*

comDenom(*List1*,*Var*) ⇒ *lista*

comDenom(*Matriz1*,*Var*) ⇒ *matriz*

comDenom(*Expr1*) entrega una proporción reducida de un numerador completamente expandido sobre un denominador completamente expandido.

comDenom(*Expr1*,*Var*) entrega una proporción reducida del numerador y el denominador expandidos con respecto de *Var*. Los términos y sus factores se clasifican con *Var* como la variable principal. Se recopilan potencias similares de *Var*. Puede haber cierta factorización incidental de los coeficientes recopilados. Se compara para omitir *Var*, con frecuencia esto ahorra tiempo, memoria y espacio de pantalla, mientras que hace la expresión más comprensible. También hace que las operaciones subsiguientes en el resultado sean más rápidas y que haya menos probabilidad de que se agote la memoria.

Si *Var* no ocurre en *Expr1*, **comDenom**(*Expr1*,*Var*) entrega una proporción reducida de un numerador no expandido sobre un denominador no expandido. Por lo general, dichos resultados incluso ahorran más tiempo, memoria y espacio de pantalla. Tales resultados parcialmente factorizados también hacen que las operaciones subsiguientes en el resultado sean más rápidas y que haya mucho menos probabilidad de que se agote la memoria.

Incluso cuando no hay ningún denominador, la función **comden** es con frecuencia una manera rápida de lograr la factorización parcial si **factor()** es demasiado lento o si se agota la memoria.

Sugerencia: Ingrese esta definición de la función **comden()** y pruébela en forma rutinaria como una alternativa para **comDenom()** y **factor()**.

$$\text{comDenom}\left(\frac{y^2+y}{(x+1)^2}+y^2+y\right)$$

$$\frac{x^2 \cdot y^2 + x^2 \cdot y + 2 \cdot x \cdot y^2 + 2 \cdot x \cdot y + 2 \cdot y^2 + 2 \cdot y}{x^2 + 2 \cdot x + 1}$$

$$\text{comDenom}\left(\frac{y^2+y}{(x+1)^2}+y^2+y, x\right)$$

$$\frac{x^2 \cdot y \cdot (y+1) + 2 \cdot x \cdot y \cdot (y+1) + 2 \cdot y \cdot (y+1)}{x^2 + 2 \cdot x + 1}$$

$$\text{comDenom}\left(\frac{y^2+y}{(x+1)^2}+y^2+y, y\right)$$

$$\frac{y^2 \cdot (x^2 + 2 \cdot x + 2) + y \cdot (x^2 + 2 \cdot x + 2)}{x^2 + 2 \cdot x + 1}$$

Define *comden*(*exprn*)=**comDenom**(*exprn*,*abc*)
Done

$$\text{comden}\left(\frac{y^2+y}{(x+1)^2}+y^2+y\right) \frac{(x^2+2 \cdot x+2) \cdot y \cdot (y+1)}{(x+1)^2}$$

$$\text{comden}\left(1234 \cdot x^2 \cdot (y^3-y) + 2468 \cdot x \cdot (y^2-1)\right)$$

$$1234 \cdot x \cdot (x \cdot y + 2) \cdot (y^2 - 1)$$

completeSquare()Catálogo > **completeSquare**(*ExprOEcn*, *Var*) expresión o ecuación \Rightarrow **completeSquare**(*ExprOEcn*, *Var*^{°Potencia}) expresión o ecuación \Rightarrow **completeSquare**(*ExprOEcn*, *Var1*, *Var2* [...]) expresión o ecuación \Rightarrow **completeSquare**(*ExprOEcn*, {*Var1*, *Var2* [...]}) expresión o ecuación \Rightarrow Convierte una expresión polinomial cuadrática de la forma $a \cdot x^2 + b \cdot x + c$ en la forma $a \cdot (x-h)^2 + k$

- 0 -

Convierte una ecuación cuadrática de la forma $a \cdot x^2 + b \cdot x + c = d$ en la forma $a \cdot (x-h)^2 = k$

El primer argumento debe ser una expresión o ecuación cuadrática en forma estándar con respecto del segundo argumento.

El Segundo argumento debe ser un término de una variable sencilla o un término de una variable sencilla elevado a una potencia racional, por ejemplo x , y^2 o $z^{(1/3)}$.La tercera y cuarta sintaxis intentan completar el cuadrado con respecto de las variables *Var1*, *Var2* [...]).

$\text{completeSquare}(x^2+2 \cdot x+3, x)$	$(x+1)^2+2$
---	-------------

$\text{completeSquare}(x^2+2 \cdot x=3, x)$	$(x+1)^2=4$
---	-------------

$\text{completeSquare}(x^6+2 \cdot x^3+3, x^3)$	$(x^3+1)^2+2$
---	---------------

$\text{completeSquare}(x^2+4 \cdot x+y^2+6 \cdot y+3=0, x, y)$	$(x+2)^2+(y+3)^2=10$
--	----------------------

$\text{completeSquare}(3 \cdot x^2+2 \cdot y+7 \cdot y^2+4 \cdot x=3, \{x, y\})$	$3 \cdot \left(x+\frac{2}{3}\right)^2+7 \cdot \left(y+\frac{1}{7}\right)^2=\frac{94}{21}$
--	---

$\text{completeSquare}(x^2+2 \cdot x \cdot y, x, y)$	$(x+y)^2-y^2$
--	---------------

conj()Catálogo > **conj**(*Expr1*) \Rightarrow expresión**conj**(*Lista1*) \Rightarrow lista**conj**(*Matriz1*) \Rightarrow matriz

Entrega el complejo conjugado del argumento.

Nota: Todas las variables indefinidas se tratan como variables reales.

$\text{conj}(1+2 \cdot i)$	$1-2 \cdot i$
----------------------------	---------------

$\text{conj}\left(\begin{bmatrix} 2 & 1-3 \cdot i \\ -i & -7 \end{bmatrix}\right)$	$\begin{bmatrix} 2 & 1+3 \cdot i \\ i & -7 \end{bmatrix}$
--	---

$\text{conj}(z)$	\bar{z}
------------------	-----------

$\text{conj}(x+i \cdot y)$	$x-y \cdot i$
----------------------------	---------------

constructMat()Catálogo > **constructMat**(*Expr*, *Var1*, *Var2*, *numFilas*, *numCols*) \Rightarrow matriz

Entrega una matriz basada en los argumentos.

Expr es una expresión en las variables *Var1* y *Var2*. Los elementos en la matriz resultante se forman al evaluar *Expr* para cada valor incrementado de *Var1* y *Var2*.*Var1* se incrementa automáticamente desde **1** a *numFilas*. Dentro de cada fila, *Var2* se incrementa desde **1** a *numCols*.

$\text{constructMat}\left(\frac{1}{i+j}, i, j, 3, 4\right)$	<table border="1"> <tr><td>$\frac{1}{2}$</td><td>$\frac{1}{3}$</td><td>$\frac{1}{4}$</td><td>$\frac{1}{5}$</td></tr> <tr><td>$\frac{1}{3}$</td><td>$\frac{1}{4}$</td><td>$\frac{1}{5}$</td><td>$\frac{1}{6}$</td></tr> <tr><td>$\frac{1}{4}$</td><td>$\frac{1}{5}$</td><td>$\frac{1}{6}$</td><td>$\frac{1}{7}$</td></tr> </table>	$\frac{1}{2}$	$\frac{1}{3}$	$\frac{1}{4}$	$\frac{1}{5}$	$\frac{1}{3}$	$\frac{1}{4}$	$\frac{1}{5}$	$\frac{1}{6}$	$\frac{1}{4}$	$\frac{1}{5}$	$\frac{1}{6}$	$\frac{1}{7}$
$\frac{1}{2}$	$\frac{1}{3}$	$\frac{1}{4}$	$\frac{1}{5}$										
$\frac{1}{3}$	$\frac{1}{4}$	$\frac{1}{5}$	$\frac{1}{6}$										
$\frac{1}{4}$	$\frac{1}{5}$	$\frac{1}{6}$	$\frac{1}{7}$										

CopyVar

Catálogo > 

CopyVar *Var1*, *Var2*

CopyVar *Var1*, *Var2*.

CopyVar *Var1*, *Var2* copia el valor de la variable *Var1* a la variable *Var2*, creando *Var2* si es necesario. La variable *Var1* debe tener un valor.

Si *Var1* es el nombre de una función existente definida por el usuario, copia la definición de esa función a la función *Var2*. La función *Var1* se debe definir.

Var1 debe cumplir con los requisitos de nombramiento de la variable o debe ser una expresión de indirección que se simplifica a un nombre de variable que cumple con los requisitos.

CopyVar *Var1*, *Var2*. copia todos los miembros del grupo de la variable *Var1*. al grupo *Var2*, creando *Var2*, si es necesario.

Var1. debe ser el nombre de un grupo de variables existente, como los resultados de las estadísticas *stat.nn* o las variables creadas usando la función **LibShortcut()**. Si *Var2*. ya existe, este comando reemplaza todos los miembros que son comunes para ambos grupos y agrega los miembros que no existen todavía. Si uno o más miembros de *Var2*. están bloqueados, todos los miembros de *Var2*. se dejan sin cambios.

Define $a(x) = \frac{1}{x}$ Done

Define $b(x) = x^2$ Done

CopyVar *a,c*: $c(4)$ $\frac{1}{4}$

CopyVar *b,c*: $c(4)$ 16

aa.a: =45 45

aa.b: =6.78 6.78

aa.c: =8.9 8.9

getVarInfo()

<i>aa.a</i>	"NUM"	" <input type="text" value=""/>
<i>aa.b</i>	"NUM"	" <input type="text" value=""/>
<i>aa.c</i>	"NUM"	" <input type="text" value=""/>

CopyVar *aa,bb*. Done

getVarInfo()

<i>aa.a</i>	"NUM"	" <input type="text" value=""/>
<i>aa.b</i>	"NUM"	" <input type="text" value=""/>
<i>aa.c</i>	"NUM"	" <input type="text" value=""/>
<i>bb.a</i>	"NUM"	" <input type="text" value=""/>
<i>bb.b</i>	"NUM"	" <input type="text" value=""/>
<i>bb.c</i>	"NUM"	" <input type="text" value=""/>

corrMat()

Catálogo > 

corrMat(*Lista1*,*Lista2*[,...[,*Lista20*]])

Genera la matriz de correlación para la matriz aumentada [*Lista1*, *Lista2*, ..., *Lista20*].

►cos

Catálogo > 

Expr ►cos

Nota: Se puede insertar este operador desde el teclado de la computadora al escribir @>cos.

Representa *Expr* en términos de coseno. Este es un operador de conversión de despliegue. Se puede usar únicamente al final de la línea de ingreso.

►cos reduce todas las potencias de

$\sin(\dots)$ módulo $1 - \cos(\dots)^2$

de manera que cualquier potencia restante de $\cos(\dots)$ tiene exponentes en el rango (0, 2). Entonces, el resultado estará libre de $\sin(\dots)$ si y sólo si $\sin(\dots)$ ocurre en la expresión dada únicamente para potencias iguales.

Nota: Este operador de conversión no está soportado en los modos de Ángulo en Grados o Gradianes. Antes de usarlo, asegúrese de que el modo de Ángulo está configurado a Radianes y que *Expr* no contiene referencias explícitas para ángulos en grados o gradianes.

$(\sin(x))^2 \blacktriangleright \cos$ $1 - (\cos(x))^2$

cos()**cos**(Expr1) ⇒ expresión**cos**(Lista1) ⇒ lista**cos**(Expr1) entrega el coseno del argumento como una expresión.**cos**(Lista1) entrega una lista de cosenos de todos los elementos en Lista1.**Nota:** El argumento se interpreta como un ángulo en grados, gradianes o radianes, de acuerdo con la configuración del modo del ángulo actual. Se puede usar °, G o R para anular el modo de ángulo en forma temporal.

En modo de ángulo en Grados:

$$\cos\left(\frac{\pi}{4}\right) \quad \frac{\sqrt{2}}{2}$$

$$\cos(45) \quad \frac{\sqrt{2}}{2}$$

$$\cos(\{0,60,90\}) \quad \left\{1, \frac{1}{2}, 0\right\}$$

En modo de ángulo en Gradianes:

$$\cos(\{0,50,100\}) \quad \left\{1, \frac{\sqrt{2}}{2}, 0\right\}$$

En modo de ángulo en Radianes:

$$\cos\left(\frac{\pi}{4}\right) \quad \frac{\sqrt{2}}{2}$$

$$\cos(45^\circ) \quad \frac{\sqrt{2}}{2}$$

cos(matrizCuadrada1) ⇒ matrizCuadradaEntrega el coseno de la matriz de *matrizCuadrada1*. Esto no es lo mismo que calcular el coseno de cada elemento.Cuando una función escalar f(A) opera en *matrizCuadrada1* (A), el resultado se calcula por medio del algoritmo:Compute los valores propios (λ_j) y los vectores propios (V_j) de A.*matrizCuadrada1* debe ser diagonalizable. Asimismo, no puede tener variables simbólicas a las que no se ha asignado un valor.

Forme las matrices:

$$B = \begin{bmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & \lambda_n \end{bmatrix} \text{ and } X = [V_1, V_2, \dots, V_n]$$

Luego $A = X B X^{-1}$ y $f(A) = X f(B) X^{-1}$. Por ejemplo, $\cos(A) = X \cos(B) X^{-1}$ donde: $\cos(B) =$

$$\begin{bmatrix} \cos(\lambda_1) & 0 & \dots & 0 \\ 0 & \cos(\lambda_2) & \dots & 0 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & \cos(\lambda_n) \end{bmatrix}$$

Todos los cálculos se realizan usando aritmética de punto flotante.

En modo de ángulo en Radianes:

$$\cos\left(\begin{bmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{bmatrix}\right)$$

$$\begin{bmatrix} 0.212493 & 0.205064 & 0.121389 \\ 0.160871 & 0.259042 & 0.037126 \\ 0.248079 & -0.090153 & 0.218972 \end{bmatrix}$$

cos⁻¹()**cos⁻¹(Expr1)** ⇒ expresión**cos⁻¹(Lista1)** ⇒ lista

En modo de ángulo en Grados:

$$\cos^{-1}(1) \quad 0$$

cos⁻¹(Expr1) entrega el ángulo cuyo coseno es *Expr1* como una expresión.

En modo de ángulo en Gradianes:

$$\cos^{-1}(0) \quad 100$$

cos⁻¹(Valor1) entrega el ángulo cuyo coseno es *Valor1*.**cos⁻¹(Lista1)** entrega una lista de cosenos inversos de cada elemento de *Lista1*.

En modo de ángulo en Radianes:

$$\cos^{-1}\{0,0.2,0.5\} \quad \left\{ \frac{\pi}{2}, 1.36944, 1.0472 \right\}$$

Nota: El resultado se entrega como un ángulo en grados, gradianes o radianes, de acuerdo con la configuración del modo del ángulo actual.**Nota:** Se puede insertar esta función desde el teclado al escribir **arccos (...)**.**cos⁻¹(matrizCuadrada1)** ⇒ matrizCuadrada

En el modo de ángulo en Radianes y el Formato Complejo Rectangular:

Entrega el coseno inverso de la matriz de *matrizCuadrada1*. Esto no es lo mismo que calcular el coseno inverso de cada elemento. Para obtener información acerca del método de cálculo, consulte **cos()**.

$$\cos^{-1}\left(\begin{bmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{bmatrix}\right)$$

matrizCuadrada1 debe ser diagonalizable. El resultado siempre contiene números de punto flotante.

$$\begin{bmatrix} 1.73485+0.064606 \cdot i & -1.49086+2.10514 \\ -0.725533+1.51594 \cdot i & 0.623491+0.77836 \cdot i \\ -2.08316+2.63205 \cdot i & 1.79018-1.27182 \cdot i \end{bmatrix}$$

Para ver el resultado completo, presione **▲** y luego use **◀** y **▶** para mover el cursor.**cosh()****cosh(Expr1)** ⇒ expresión**cosh(Lista1)** ⇒ lista

$$\cosh\left(\left(\frac{\pi}{4}\right)r\right) \quad \cosh(45)$$

cosh(Expr1) entrega el coseno hiperbólico del argumento como una expresión.**cosh(Lista1)** entrega una lista de cosenos hiperbólicos de cada elemento de *Lista1*.**cosh(matrizCuadrada1)** ⇒ matrizCuadrada

En modo de ángulo en Radianes:

Entrega el coseno hiperbólico de la matriz de *matrizCuadrada1*. Esto no es lo mismo que calcular el coseno hiperbólico de cada elemento. Para obtener información acerca del método de cálculo, consulte **cos()**.

$$\cosh\left(\begin{bmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{bmatrix}\right)$$

matrizCuadrada1 debe ser diagonalizable. El resultado siempre contiene números de punto flotante.

$$\begin{bmatrix} 421.255 & 253.909 & 216.905 \\ 327.635 & 255.301 & 202.958 \\ 226.297 & 216.623 & 167.628 \end{bmatrix}$$

cosh⁻¹()

Catálogo >

cosh⁻¹(Expr1) ⇒ expresión**cosh⁻¹(Lista1)** ⇒ lista**cosh⁻¹(Expr1)** entrega el coseno hiperbólico inverso del argumento como una expresión.**cosh⁻¹(Lista1)** entrega una lista de cosenos hiperbólicos inversos de cada elemento de *Lista1*.**Nota:** Se puede insertar esta función desde el teclado al escribir **arccosh** (...).**cosh⁻¹(matrizCuadrada1)** ⇒ matrizCuadradaEntrega el coseno hiperbólico inverso de la matriz de *matrizCuadrada1*. Esto no es lo mismo que calcular el coseno hiperbólico inverso de cada elemento. Para obtener información acerca del método de cálculo, consulte **cos()**.*matrizCuadrada1* debe ser diagonalizable. El resultado siempre contiene números de punto flotante.

$$\frac{\cosh^{-1}(1)}{\cosh^{-1}\{1,2,1,3\}} = \frac{0}{\{0,1.37286,\cosh^{-1}(3)\}}$$

En el modo de ángulo en Radianes y en el Formato Complejo Rectangular:

$$\cosh^{-1}\begin{pmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{pmatrix} = \begin{bmatrix} 2.52503+1.73485\cdot i & -0.009241-1.49086i \\ 0.486969-0.725533\cdot i & 1.66262+0.623491i \\ -0.322354-2.08316\cdot i & 1.26707+1.79018i \end{bmatrix}$$

Para ver el resultado completo, presione \blacktriangle y luego use \blacktriangleleft y \blacktriangleright para mover el cursor.**cot()****cot(Expr1)** ⇒ expresión**cot(Lista1)** ⇒ listaEntrega la cotangente de *Expr1* o entrega una lista de cotangentes de todos los elementos en *Lista1*.**Nota:** El argumento se interpreta como un ángulo en grados, gradianes o radianes, de acuerdo con la configuración del modo del ángulo actual. Se puede usar °, G o R para anular el modo de ángulo en forma temporal.

En modo de ángulo en Grados:

$$\cot(45) = 1$$

En modo de ángulo en Gradianes:

$$\cot(50) = 1$$

En modo de ángulo en Radianes:

$$\cot\{1,2,1,3\} = \left\{ \frac{1}{\tan(1)}, -0.584848, \frac{1}{\tan(3)} \right\}$$

cot⁻¹()**cot⁻¹(Expr1)** ⇒ expresión**cot⁻¹(Lista1)** ⇒ listaEntrega el ángulo cuya cotangente es *Expr1* o entrega una lista que contiene las cotangentes inversas de cada elemento de *Lista1*.**Nota:** El resultado se entrega como un ángulo en grados, gradianes o radianes, de acuerdo con la configuración del modo del ángulo actual.**Nota:** Se puede insertar esta función desde el teclado al escribir **arccot** (...).

En modo de ángulo en Grados:

$$\cot^{-1}(1) = 45$$

En modo de ángulo en Gradianes:

$$\cot^{-1}(1) = 50$$

En modo de ángulo en Radianes:

$$\cot^{-1}(1) = \frac{\pi}{4}$$

coth()Catálogo > **coth**(Expr1) ⇒ expresión**coth**(Lista1) ⇒ lista

Entrega la cotangente hiperbólica de Expr1 o entrega una lista de cotangentes hiperbólicas de todos los elementos de Lista1.

$\text{coth}(1.2)$	1.19954
$\text{coth}\{1,3.2\}$	$\left\{ \frac{1}{\tanh(1)}, 1.00333 \right\}$

coth⁻¹()Catálogo > **coth⁻¹**(Expr1) ⇒ expresión**coth⁻¹**(Lista1) ⇒ lista

Entrega la cotangente hiperbólica inversa de Expr1 o entrega una lista que contiene las cotangentes hiperbólicas inversas de cada elemento de Lista1.

Nota: Se puede insertar esta función desde el teclado al escribir **arccoth** (...).

$\text{coth}^{-1}(3.5)$	0.293893
$\text{coth}^{-1}\{-2.2,1.6\}$	$\left\{ \frac{-\ln(3)}{2}, 0.518046, \frac{\ln\left(\frac{7}{5}\right)}{2} \right\}$

count()Catálogo > **count**(Valor1oLista1 [,Valor2oLista2 [,...]]) ⇒ valor

Entrega el conteo acumulado de todos los elementos en los argumentos que se evalúan a valores numéricos.

Cada argumento puede ser una expresión, valor, lista o matriz. Se puede mezclar tipos de datos y usar argumentos de varias dimensiones.

Para una lista, matriz o rango de celdas, cada elemento se evalúa para determinar si se debe incluir en el conteo.

Dentro de la aplicación Listas y Hoja de Cálculo, se puede usar un rango de celdas en lugar de cualquier argumento.

Los elementos vacíos (anulados) se ignoran. Para obtener más información sobre elementos vacíos, vea la página 166.

$\text{count}(2,4,6)$	3
$\text{count}\{2,4,6\}$	3
$\text{count}\left(2,\{4,6\},\begin{bmatrix} 8 & 10 \\ 12 & 14 \end{bmatrix}\right)$	7
$\text{count}\left(\frac{1}{2},3+4\cdot i,\text{undef},"hello",x+5,\text{sign}(0)\right)$	2

En el último ejemplo, sólo 1/2 y 3+4*i* se cuentan. Los argumentos restantes, suponiendo que *x* no está definida, no se evalúan a valores numéricos.

countif() (conteoSí)Catálogo > **countif**(*Lista*,*Criterios*) ⇒ *valor*

Entrega el conteo acumulado de todos los elementos en *Lista* que cumplen con los *Criterios* especificados.

Los *criterios* pueden ser:

- Un valor, una expresión o una cadena. Por ejemplo, **3** cuenta sólo aquellos elementos en *Lista* que se simplifican al valor 3.
- Una expresión Booleana que contiene el símbolo **?** como un marcador de posición para cada elemento. Por ejemplo, **?<5** cuenta sólo aquellos elementos en *Lista* que son menos de 5.

Dentro de la aplicación Listas y Hoja de Cálculo, se puede usar un rango de celdas en lugar de *Lista*.

Los elementos vacíos (anulados) en la lista se ignoran. Para obtener más información sobre elementos vacíos, vea la página 166.

Nota: Vea también **sumif()**, página 125, y **frequency()**, página 52.

$$\text{countif}\left(\{1,3,"abc",\text{undef},3,1\},3\right) \quad 2$$

Cuenta el número de elementos iguales a 3.

$$\text{countif}\left(\{"abc","def","abc",3\},"dif"\right) \quad 1$$

Cuenta el número de elementos iguales a "dif".

$$\text{countif}\left(\{x^{-2},x^{-1},1,x,x^2\},x\right) \quad 1$$

Cuenta el número de elementos iguales a x ; este ejemplo supone que la variable x es indefinida.

$$\text{countif}\left(\{1,3,5,7,9\},?<5\right) \quad 2$$

Cuenta 1 y 3.

$$\text{countif}\left(\{1,3,5,7,9\},2<?<8\right) \quad 3$$

Cuenta 3, 5 y 7.

$$\text{countif}\left(\{1,3,5,7,9\},?<4 \text{ or } ?>6\right) \quad 4$$

Cuenta 1, 3, 7 y 9.

cPolyRoots() (RaícesPolíC)Catálogo > **cPolyRoots**(*Poli*,*Var*) ⇒ *lista***cPolyRoots**(*ListaDeCoefs*) ⇒ *lista*

La primera sintaxis, **cPolyRoots**(*Poli*,*Var*), entrega una lista de raíces complejas del polinomio *Poli* con respecto de la variable *Var*.

Poli debe ser un polinomio en una variable.

La segunda sintaxis, **cPolyRoots**(*ListaDeCoefs*), entrega una lista de raíces complejas para los coeficientes en *ListaDeCoefs*.

Nota: Vea también **polyRoots()**, página 93.

$$\text{cPolyRoots}\left(y^3+1,y\right) \quad \{-1\}$$

$$\text{cPolyRoots}\left(y^3+1,y\right) \quad \left\{-1, \frac{1}{2} - \frac{\sqrt{3}}{2}i, \frac{1}{2} + \frac{\sqrt{3}}{2}i\right\}$$

$$\text{cPolyRoots}\left(x^2+2x+1,x\right) \quad \{-1,-1\}$$

$$\text{cPolyRoots}\left(\{1,2,1\}\right) \quad \{-1,-1\}$$

crossP()Catálogo > **crossP**(*Lista1*, *Lista2*) ⇒ *lista*

Entrega el producto cruzado de *Lista1* y *Lista2* como una lista.

Lista1 y *Lista2* deben tener una dimensión igual, y la dimensión debe ser 2 ó 3.

$$\text{crossP}\left(\{a1,b1\},\{a2,b2\}\right) \quad \{0,0,a1\cdot b2-a2\cdot b1\}$$

$$\text{crossP}\left(\{0.1,2.2,-5\},\{1,-0.5,0\}\right) \quad \{-2.5,-5,-2.25\}$$

crossP(*Vector1*, *Vector2*) ⇒ *vector*

Entrega un vector de fila o columna (dependiendo de los argumentos) que es el producto cruzado de *Vector1* y *Vector2*.

Tanto *Vector1* como *Vector2* deben ser vectores de fila, o ambos deben ser vectores de columna. Ambos vectores deben tener una dimensión igual, y la dimensión debe ser 2 ó 3.

$$\text{crossP}\left(\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix},\begin{bmatrix} 3 & 6 & -3 \end{bmatrix}\right) \quad \begin{bmatrix} -3 & 6 & -3 \end{bmatrix}$$

$$\text{crossP}\left(\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix},\begin{bmatrix} 3 & 4 \end{bmatrix}\right) \quad \begin{bmatrix} 0 & 0 & -2 \end{bmatrix}$$

csc()
 **tecla**
csc(*Expr1*) ⇒ expresión**csc**(*Lista1*) ⇒ lista

Entrega la cosecante de *Expr1* o entrega una lista que contiene las cosecantes de todos los elementos en *Lista1*.

En modo de ángulo en Grados:

$$\text{csc}(45) \quad \frac{\sqrt{2}}{2}$$

En modo de ángulo en Gradianes:

$$\text{csc}(50) \quad \frac{\sqrt{2}}{2}$$

En modo de ángulo en Radianes:

$$\text{csc}\left(\left\{1, \frac{\pi}{2}, \frac{\pi}{3}\right\}\right) \quad \left\{\frac{1}{\sin(1)}, 1, \frac{2\sqrt{3}}{3}\right\}$$

csc⁻¹()
 **tecla**
csc⁻¹(*Expr1*) ⇒ expresión**csc⁻¹**(*Lista1*) ⇒ lista

Entrega el ángulo cuya cosecante es *Expr1* o entrega una lista que contiene las cosecantes inversas de cada elemento de *Lista1*.

Nota: El resultado se entrega como un ángulo en grados, gradianes o radianes, de acuerdo con la configuración del modo del ángulo actual.

Nota: Se puede insertar esta función desde el teclado al escribir **arccsc**(...).

En modo de ángulo en Grados:

$$\text{csc}^{-1}(1) \quad 90$$

En modo de ángulo en Gradianes:

$$\text{csc}^{-1}(1) \quad 100$$

En modo de ángulo en Radianes:

$$\text{csc}^{-1}\left(\left\{1, 4, 6\right\}\right) \quad \left\{\frac{\pi}{2}, \sin^{-1}\left(\frac{1}{4}\right), \sin^{-1}\left(\frac{1}{6}\right)\right\}$$

csch()
Catálogo > 
csch(*Expr1*) ⇒ expresión**csch**(*Lista1*) ⇒ lista

Entrega la cosecante hiperbólica de *Expr1* o entrega una lista de cosecantes hiperbólicas de todos los elementos de *Lista1*.

$$\text{csch}(3) \quad \frac{1}{\sinh(3)}$$

$$\text{csch}\left(\left\{1, 2, 1, 4\right\}\right) \quad \left\{\frac{1}{\sinh(1)}, 0.248641, \frac{1}{\sinh(4)}\right\}$$

csch⁻¹()
Catálogo > 
csch⁻¹(*Expr1*) ⇒ expresión**csch⁻¹**(*Lista1*) ⇒ lista

Entrega la cosecante hiperbólica inversa de *Expr1* o entrega una lista que contiene las cosecantes hiperbólicas inversas de cada elemento de *Lista1*.

Nota: Se puede insertar esta función desde el teclado al escribir **arccsch**(...).

$$\text{csch}^{-1}(1) \quad \sinh^{-1}(1)$$

$$\text{csch}^{-1}\left(\left\{1, 2, 1, 3\right\}\right) \quad \left\{\sinh^{-1}(1), 0.459815, \sinh^{-1}\left(\frac{1}{3}\right)\right\}$$

cSolve(Ecuación, Var) \Rightarrow expresión Booleana

cSolve(Ecuación, Var=Cálculo) \Rightarrow expresión Booleana

cSolve(Desigualdad, Var) \Rightarrow expresión Booleana

Entrega soluciones complejas posibles de una ecuación o desigualdad para *Var*. La meta es producir posibles para todas las soluciones reales y no reales. Incluso si la *Ecuación* es real, **cSolve()** permite resultados no reales en Formato Complejo de resultado Real.

Aunque todas las variables no definidas que no **cSolve** con un guión bajo () se procesan como si fueran reales, **cSolve()** puede solucionar ecuaciones polinómicas para soluciones complejas.

cSolve() configura temporalmente el dominio para complejas durante la solución, incluso si el dominio actual es real. En el dominio complejo, las potencias fraccionarias que tienen denominadores noes usan el principal en lugar del ramal real. En consecuencia, las soluciones de **solve()** para las ecuaciones que incluyen dichas potencias fraccionarias no son necesariamente un subconjunto de aquellas de **cSolve()**.

cSolve() comienza con métodos simbólicos exactos. **cSolve()** también usa factorización polinómica compleja aproximada iterativa, de ser necesario.

Nota: Vea también **cZeros()**, **solve()** y **zeros()**.

Nota: Si la *Ecuación* no es polinómica con funciones como **abs()**, **angle()**, **conj()**, **real()** o **imag()**, usted debe poner un guión bajo (presione **ctrl** ) al final de *Var*. De manera predeterminada, una variable se trata como un valor real.

Si se usa *var_*, la variable se trata como complejo.

También se debe usar *var_* para cualquier otra variable en la *Ecuación* que pudiera tener valores irrales. De otro modo, usted puede recibir resultados inesperados.

cSolve(Ecn1 and Ecn2 [and ...],

VarOCálculo1, *VarOCálculo2* [, ...]) \Rightarrow expresión Booleana

cSolve(SistemaDeEcns, *VarOCálculo1*,

VarOCálculo2 [, ...]) \Rightarrow expresión Booleana

Entrega soluciones complejas posibles para las ecuaciones algebraicas simultáneas, donde cada *varOCálculo* especifica una variable que usted desea solucionar.

De manera opcional, se puede especificar un cálculo inicial para una variable. Cada *varOCálculo* debe tener la forma:

variable

- 0 -

variable = número real o irreal

Por ejemplo, *x* es válida y también lo es *x=3+i*.

Si todas las ecuaciones son polinomios y usted NO especifica cualquier cálculo inicial, **cSolve()** usa el método de eliminación de léxico Gröbner/Buchberger para intentar determinar **todas** las soluciones complejas.

$$\text{cSolve}(x^3=1,x)$$

$$x = \frac{1}{2} + \frac{\sqrt{3}}{2} \cdot i \text{ or } x = \frac{1}{2} - \frac{\sqrt{3}}{2} \cdot i \text{ or } x = -1$$

$$\text{solve}(x^3=-1,x) \quad x = -1$$

$$\text{cSolve}\left(x^{\frac{1}{3}}=-1,x\right) \quad \text{false}$$

$$\text{solve}\left(x^{\frac{1}{3}}=-1,x\right) \quad x = -1$$

En modo de Dígitos de Despliegue de Fijo 2:

$$\text{exact}(\text{cSolve}(x^5+4 \cdot x^4+5 \cdot x^3-6 \cdot x-3=0,x))$$

$$x \cdot (x^4+4 \cdot x^3+5 \cdot x^2-6)=3$$

$$\text{cSolve}(Ans,x)$$

$$x = -1.11+1.07 \cdot i \text{ or } x = -1.11-1.07 \cdot i \text{ or } x = -2.$$

Para ver el resultado completo, presione \blacktriangle y luego use \blacktriangleleft y \blacktriangleright para mover el cursor.

z se trata como real:

$$\text{cSolve}(\text{conj}(z)=1+i,z) \quad z = 1+i$$

z_ se trata como complejo:

$$\text{cSolve}(\text{conj}(z_)=1+i,z_) \quad z_ = 1-i$$

Nota: Los siguientes ejemplos usan un guión bajo (presione **ctrl** ) de manera que las variables se tratarán como complejas.

Las soluciones complejas pueden incluir soluciones tanto reales como irreales, como en el ejemplo de la derecha.

$$\text{cSolve}\left(u_{-} \cdot v_{-} - u_{-} = v_{-} \text{ and } v_{-}^2 = u_{-}, \{u_{-}, v_{-}\}\right)$$

$$u_{-} = \frac{1}{2} + \frac{\sqrt{3}}{2} \cdot i \text{ and } v_{-} = \frac{1}{2} - \frac{\sqrt{3}}{2} \cdot i \text{ or } u_{-} = \frac{1}{2} - i$$

Para ver el resultado completo, presione \blacktriangle y luego use \blacktriangleleft para mover el cursor.

Las ecuaciones polinómicas simultáneas pueden tener variables extras que no tienen ningún valor, aunque representan valores numéricos dados que podrían sustituirse más adelante.

$$\text{cSolve}\left(u_{-} \cdot v_{-} - u_{-} = c_{-} \cdot v_{-} \text{ and } v_{-}^2 = u_{-}, \{u_{-}, v_{-}\}\right)$$

$$u_{-} = \frac{-(\sqrt{1-4 \cdot c_{-} + 1})^2}{4} \text{ and } v_{-} = \frac{\sqrt{1-4 \cdot c_{-} + 1}}{2} \text{ or } u_{-} =$$

Para ver el resultado completo, presione \blacktriangle y luego use \blacktriangleleft para mover el cursor.

También se pueden incluir variables de solución que no aparecen en las ecuaciones. Estas soluciones muestran cómo las familias de soluciones podrían contener constantes arbitrarias de la forma ck , donde k es un sufijo de entero desde 1 hasta 255.

$$\text{cSolve}\left(u_{-} \cdot v_{-} - u_{-} = v_{-} \text{ and } v_{-}^2 = u_{-}, \{u_{-}, v_{-}, w_{-}\}\right)$$

$$u_{-} = \frac{1}{2} + \frac{\sqrt{3}}{2} \cdot i \text{ and } v_{-} = \frac{1}{2} - \frac{\sqrt{3}}{2} \cdot i \text{ and } w_{-} = c8 \text{ or } u_{-}$$

Para ver el resultado completo, presione \blacktriangle y luego use \blacktriangleleft para mover el cursor.

Para sistemas polinómicos, el tiempo de cálculo o el agotamiento de memoria pueden depender ampliamente del orden en el cual se enumeran las variables de solución. Si su elección inicial agota la memoria o su paciencia, intente volver a arreglar las variables en las ecuaciones ylo en la lista $varOCálculo$.

$$\text{cSolve}\left(u_{-} + v_{-} = e^{w_{-}} \text{ and } u_{-} - v_{-} = i, \{u_{-}, v_{-}\}\right)$$

$$u_{-} = \frac{e^{w_{-} + i}}{2} \text{ and } v_{-} = \frac{e^{w_{-} - i}}{2}$$

Si usted no incluye ningún cálculo y si cualquier ecuación no es polinómica en cualquier variable, pero todas las ecuaciones son lineales en todas las variables de solución, **cSolve()** usa la eliminación Gaussiana para tratar de determinar todas las soluciones.

$$\text{cSolve}\left(e^{z_{-}} = w_{-} \text{ and } w_{-} = z_{-}^2, \{w_{-}, z_{-}\}\right)$$

$$w_{-} = 0.494866 \text{ and } z_{-} = -0.703467$$

Si un sistema no es ni polinómico en todas sus variables ni lineal en sus variables de solución, **cSolve()** determina como máximo una solución usando un método iterativo aproximado. Para hacer esto, el número de variables de solución debe igualar el número de ecuaciones, y todas las demás variables en las ecuaciones deben simplificarse a números.

$$\text{cSolve}\left(e^{z_{-}} = w_{-} \text{ and } w_{-} = z_{-}^2, \{w_{-}, z_{-} = 1 + i\}\right)$$

$$w_{-} = 0.149606 + 4.8919 \cdot i \text{ and } z_{-} = 1.58805 + 1 \cdot i$$

Con frecuencia es necesario un cálculo irreal para determinar una solución irreal. Por convergencia, un cálculo podría tener que ser más bien cercano a una solución.

Para ver el resultado completo, presione \blacktriangle y luego use \blacktriangleleft para mover el cursor.

CubicReg X , Y , [$Frec$] [, $Categoría$, $Incluir$]

Resuelve la regresión polinómica cúbica $y = a \cdot x^3 + b \cdot x^2 + c \cdot x + d$ en listas X y Y con frecuencia $Frec$. Un resumen de resultados se almacena en la variable $stat.results$. (Vea la página 122.)

Todas las listas deben tener una dimensión igual, excepto por $Incluir$.

X y Y son listas de variables independientes y dependientes.

$Frec$ es una lista opcional de valores de frecuencia. Cada elemento en $Frec$ especifica la frecuencia de la ocurrencia para cada punto de datos X y Y correspondientes. El valor predeterminado es 1. Todos los elementos deben ser enteros ≥ 0 .

$Categoría$ es una lista de códigos de categoría para los datos X y Y correspondientes.

$Incluir$ es una lista de uno o más códigos de categoría. Sólo aquellos elementos de datos cuyo código de categoría está incluido en esta lista están incluidos en el cálculo.

Para obtener información sobre el efecto de los elementos vacíos en una lista, vea "Elementos vacíos (inválidos)" en la página 166.

Variable de salida	Descripción
stat.EcnReg	Ecuación de regresión: $a \cdot x^3 + b \cdot x^2 + c \cdot x + d$
stat.a, stat.b, stat.c, stat.d	Coefficientes de regresión
stat.R ²	Coefficiente de determinación
stat.Resid	Residuales de la regresión
stat.XReg	La lista de puntos de datos en <i>Lista X</i> modificada se usa de hecho en la regresión con base en las restricciones de las Categorías $Frec$, <i>Lista de Categorías Incluir</i>
stat.YReg	La lista de puntos de datos en <i>Lista Y</i> modificada se usa de hecho en la regresión con base en las restricciones de las Categorías $Frec$, <i>Lista de Categorías Incluir</i>
stat.FrecReg	Lista de frecuencias correspondientes a $stat.XReg$ y $stat.YReg$

cumulativeSum()**cumulativeSum(Lista1)** \Rightarrow lista

Entrega una lista de sumas acumulativas de los elementos en *Lista1* comenzando en el elemento 1.

cumulativeSum(Matriz1) \Rightarrow matriz

Entrega una matriz de sumas acumulativas de los elementos en *Matriz1*. Cada elemento está en la suma acumulativa de la columna desde la parte superior hasta la parte inferior.

Un elemento vacío (anulado) en *Lista1* o *Matriz1* produce un elemento anulado en la lista o matriz resultante. Para obtener más información sobre elementos vacíos, vea la página 166.

$$\text{cumulativeSum}\{ \{1,2,3,4\} \} \quad \{1,3,6,10\}$$

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix} \rightarrow m1 \quad \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}$$

$$\text{cumulativeSum}(m1) \quad \begin{bmatrix} 1 & 2 \\ 4 & 6 \\ 9 & 12 \end{bmatrix}$$

Cycle

Catálogo >

Cycle

Transfiere el control de inmediato a la siguiente iteración del bucle actual (**For**, **While**, o **Loop**).

Cycle no está permitido afuera de las tres estructuras de bucles ((**For**, **While**, o **Loop**)).

Nota para ingresar el ejemplo: En la aplicación de Calculadora en el dispositivo portátil, usted puede ingresar definiciones en líneas múltiples al presionar en lugar de al final de cada línea. En el teclado de la computadora, presione y sostenga **Alt** y presione **Ingresar**.

Lista de funciones que suma los enteros desde 1 hasta 100, saltándose 50.

```
Define g() $\leftarrow$ Func
Local temp,i
0  $\rightarrow$  temp
For i,1,100,1
If i=50
Cycle
temp+i  $\rightarrow$  temp
EndFor
Return temp
EndFunc
```

$g()$ 5000

Cylind

Catálogo >

Vector Cylind

Nota: Se puede insertar este operador desde el teclado de la computadora al escribir **@>Cylind**.

Despliega el vector de fila o columna en forma cilíndrica $[r, \angle \theta, z]$.

Vector debe tener exactamente tres elementos. Puede ser una fila o una columna.

$[2 \ 2 \ 3]$ Cylind $\left[2 \cdot \sqrt{2} \ \angle \frac{\pi}{4} \ 3 \right]$

cZeros()

Catálogo >

cZeros(*Expr*, *Var*) \Rightarrow lista

Entrega una lista de valores reales e irreales posibles de *Var* que hacen $Expr=0$. **cZeros()** hace esto al calcular **expList(cSolve(Expr=0,Var),Var)**. De otro modo, **cZeros()** es similar a **zeros()**.

Nota: Vea también **cSolve()**, **solve()** y **zeros()**.

Nota: Si *Expr* no es polinómica con funciones como **abs()**, **angle()**, **conj()**, **real()** o **imag()**, usted debe poner un guión bajo (presione) al final de *Var*. De manera predeterminada, una variable se trata como un valor real. Si se usa *var_*, la variable se trata como complejo.

También se debe usar *var_* para cualquier otra variable en *Expr* que pudiera tener valores irreales. De otro modo, usted puede recibir resultados inesperados.

cZeros({*Expr*1, *Expr*2 [, ...]},
{*Var*Ocálculo1,*Var*Ocálculo2 [, ...]}) \Rightarrow matriz

Entrega las posibles posiciones donde las expresiones son cero en forma simultánea. Cada *Var*Ocálculo especifica un desconocido cuyo valor usted busca.

De manera opcional, se puede especificar un cálculo inicial para una variable. Cada *Var*Ocálculo debe tener la forma:

variable
- 0 -
variable = número real o irreal

Por ejemplo, *x* es válida y también lo es $x=3+i$.

En modo de Dígitos de Despliegue de Fijo 3:

$cZeros(x^5+4 \cdot x^4+5 \cdot x^3-6 \cdot x-3,x)$
 $\{-1.1138+1.07314 \cdot i,-1.1138-1.07314 \cdot i,-2 \cdot$

Para ver el resultado completo, presione y luego use y para mover el cursor.

z se trata como real:

$cZeros(conj(z)-1-i,z)$ $\{1+i\}$

z_ se trata como complejo:

$cZeros(conj(z_)-1-i,z_)$ $\{1-i\}$

Si todas las expresiones son polinomios y usted NO especifica cualquier cálculo inicial, **cZeros()** usa el método de eliminación de léxico Gröbner/Buchberger para intentar determinar **todos** los ceros complejos.

Los ceros complejos pueden incluir ceros tanto reales como irreales, como en el ejemplo de la derecha.

Cada fila de la matriz resultante representa un cero alterno, con los componentes ordenados igual que la lista *VarOCálculo* lista. Para extraer una fila, index de la matriz con *[fila]*.

Nota: Los siguientes ejemplos usan un guión bajo (presione **ctrl** **[_]**) de manera que las variables se tratarán como complejas.

$$\text{cZeros}\left(\left\{u_- \cdot v_- - u_- v_- v_-^2 + u_-\right\}, \left\{u_-, v_-\right\}\right)$$

$$\begin{bmatrix} 0 & 0 \\ \frac{1}{2} \frac{\sqrt{3}}{2} \cdot i & \frac{1}{2} \frac{\sqrt{3}}{2} \cdot i \\ \frac{1}{2} + \frac{\sqrt{3}}{2} \cdot i & \frac{1}{2} \frac{\sqrt{3}}{2} \cdot i \end{bmatrix}$$

Extraer la fila 2:

$$\text{Ans}[2] \quad \begin{bmatrix} \frac{1}{2} \frac{\sqrt{3}}{2} \cdot i & \frac{1}{2} \frac{\sqrt{3}}{2} \cdot i \end{bmatrix}$$

Los polinomios simultáneos pueden tener variables extras que no tienen ningún valor, aunque representan valores numéricos dados que podrían sustituirse más adelante.

$$\text{cZeros}\left(\left\{u_- \cdot v_- - u_- c_- v_- v_-^2 + u_-\right\}, \left\{u_-, v_-\right\}\right)$$

$$\begin{bmatrix} 0 & 0 \\ \frac{-\sqrt{1-4 \cdot c_- - 1}}{4} & \frac{-\sqrt{1-4 \cdot c_- - 1}}{2} \\ \frac{-\sqrt{1-4 \cdot c_- + 1}}{4} & \frac{\sqrt{1-4 \cdot c_- + 1}}{2} \end{bmatrix}$$

Usted también puede incluir variables desconocidas que no aparecen en las expresiones. Estos ceros muestran cómo las familias de ceros podrían contener constantes arbitrarias de la forma *ck*, donde *k* es un sufijo de entero desde 1 hasta 255.

Para sistemas polinómicos, el tiempo de cálculo o el agotamiento de memoria pueden depender ampliamente del orden en el cual se enumeran los desconocidos. Si su elección inicial agota la memoria o su paciencia, intente volver a arreglar las variables en las expresiones y/o en la lista *VarOCálculo*.

$$\text{cZeros}\left(\left\{u_- \cdot v_- - u_- v_- v_-^2 + u_-\right\}, \left\{u_-, v_-, w_-\right\}\right)$$

$$\begin{bmatrix} 0 & 0 & c4 \\ \frac{1}{2} \frac{\sqrt{3}}{2} \cdot i & \frac{1}{2} \frac{\sqrt{3}}{2} \cdot i & c4 \\ \frac{1}{2} + \frac{\sqrt{3}}{2} \cdot i & \frac{1}{2} \frac{\sqrt{3}}{2} \cdot i & c4 \end{bmatrix}$$

Si usted no incluye ningún cálculo y si cualquier expresión no es polinómica en cualquier variable, pero todas las expresiones son lineales en todos los desconocidos, **cZeros()** usa la eliminación Gaussiana para tratar de determinar todos los ceros.

$$\text{cZeros}\left(\left\{u_- + v_- e^{w_-}, u_- v_- - i\right\}, \left\{u_-, v_-\right\}\right)$$

$$\begin{bmatrix} e^{w_- + i} & e^{w_- - i} \\ 2 & 2 \end{bmatrix}$$

Si un sistema no es ni polinómico en todas sus variables ni lineal en sus desconocidos, **cZeros()** determina como máximo un cero usando un método iterativo aproximado. Para hacer esto, el número de desconocidos debe igualar el número de expresiones, y todas las demás variables en las expresiones deben simplificarse a números.

$$\text{cZeros}\left(\left\{e^{z_-} - w_-, w_- z_-^2\right\}, \left\{w_-, z_-\right\}\right)$$

$$[0.494866 \quad -0.703467]$$

Con frecuencia es necesario un cálculo irreal para determinar un cero irreal. Por convergencia, un cálculo podría tener que ser más bien cercano a un cero.

$$\text{cZeros}\left(\left\{e^{z_-} - w_-, w_- z_-^2\right\}, \left\{w_-, z_- = 1 + i\right\}\right)$$

$$[0.149606 + 4.8919 \cdot i \quad 1.58805 + 1.54022 \cdot i]$$

D

dbd()

Catálogo >

dbd(*fecha1, fecha2*) ⇒ *valor*

Entrega el número de días entre *fecha1* y *fecha2* usando el método de conteo de días reales.

fecha1 y *fecha2* pueden ser números dentro del rango de las fechas en el calendario estándar. Si tanto *fecha1* como *fecha2* son listas, deberán tener la misma longitud.

Tanto *fecha1* como *fecha2* deben estar entre los años 1950 a 2049.

Usted puede ingresar las fechas en uno de dos formatos. La colocación decimal se diferencia entre los formatos de fecha.

MM.DDAA (formato que se usa de manera común en los Estados Unidos)

DDMM.AA (formato que se usa de manera común en Europa)

$\text{dbd}(12.3103, 1.0104)$	1
$\text{dbd}(1.0107, 6.0107)$	151
$\text{dbd}(3112.03, 101.04)$	1
$\text{dbd}(101.07, 106.07)$	151

►DD

Catálogo >

Expr1 ►DD ⇒ *valor*

Lista1 ►DD ⇒ *lista*

Matriz1 ►DD ⇒ *matriz*

Nota: Usted puede insertar este operador desde el teclado de la computadora al escribir @>DD.

Entrega el decimal equivalente del argumento expresado en grados. El argumento es un número, lista o matriz que se interpreta por medio de la configuración del modo de Ángulo en gradianes, radianes o grados.

En modo de ángulo en Grados:

$(1.5^\circ) \blacktriangleright \text{DD}$	1.5°
$(45^\circ 22' 14.3'') \blacktriangleright \text{DD}$	45.3706°
$(\{45^\circ 22' 14.3'', 60^\circ 0' 0''\}) \blacktriangleright \text{DD}$	$\{45.3706^\circ, 60^\circ\}$

En modo de ángulo en Gradianes:

$1 \blacktriangleright \text{DD}$	$\frac{9}{10}^\circ$
-----------------------------------	----------------------

En modo de ángulo en Radianes:

$(1.5) \blacktriangleright \text{DD}$	85.9437°
---------------------------------------	-----------------

►Decimal

Catálogo >

Expresión1 ►Decimal ⇒ *expresión*

Lista1 ►Decimal ⇒ *expresión*

Matriz1 ►Decimal ⇒ *expresión*

Nota: Usted puede insertar este operador desde el teclado de la computadora al escribir @>Decimal.

Despliega el argumento en forma decimal. Este operador se puede usar únicamente al final de la línea de ingreso.

$\frac{1}{3} \blacktriangleright \text{Decimal}$	0.333333
--	----------

Define (Definir)

Catálogo > 

Define *Var = Expresión*

Define *Función(Param1, Param2, ...) = Expresión*

Define la variable *Var* o la función definida por el usuario *Función*.

Los parámetros, como *Param1*, proporcionan marcadores de posición para pasar argumentos a la función. Cuando llame a una función definida por el usuario, usted deberá suministrar argumentos (por ejemplo, valores o variables) que correspondan a los parámetros. Cuando se llama, la función evalúa la *Expresión* usando los argumentos provistos.

Var y *Función* no pueden ser el nombre de una variable de sistema o de una función o un comando integrado.

Nota: Esta forma de **Define** es equivalente a ejecutar la expresión: *expresión* → *Función(Param1, Param2)*.

Define *Función(Param1, Param2, ...) = Func*

Bloque

EndFunc

Define *Programa(Param1, Param2, ...) = Prgm*

Bloque

EndPrgm

En esta forma, la función o el programa definido por el usuario puede ejecutar un bloque de varias sentencias.

Bloque puede ser una sentencia sencilla o una serie de sentencias en líneas separadas. *Bloque* también puede incluir expresiones e instrucciones (como **If**, **Then**, **Else**, y **For**).

Nota para ingresar el ejemplo: En la aplicación de Calculadora en el dispositivo portátil, usted puede ingresar definiciones en líneas múltiples al presionar  en lugar de **enter** al final de cada línea. En el teclado de la computadora, presione y sostenga **Alt** y presione **Ingresar**.

Nota: Vea también **Define LibPriv**, página 35 y **Define LibPub**, página 36.

Define $g(x,y)=2 \cdot x-3 \cdot y$	Done
$g(1,2)$	-4
$1 \rightarrow a: 2 \rightarrow b: g(a,b)$	-4
Define $h(x)=\text{when}(x<2,2 \cdot x-3,-2 \cdot x+3)$	Done
$h(-3)$	-9
$h(4)$	-5

Define $g(x,y)=\text{Func}$	Done
If $x>y$ Then	
Return x	
Else	
Return y	
EndIf	
EndFunc	

$g(3,-7)$	3
-----------	---

Define $g(x,y)=\text{Prgm}$	
If $x>y$ Then	
Disp x , " greater than ", y	
Else	
Disp x , " not greater than ", y	
EndIf	
EndPrgm	

Done

$g(3,-7)$	
	3 greater than -7
	Done

Define LibPriv

Catálogo > 

Define LibPriv *Var = Expresión*

Define LibPriv *Función(Param1, Param2, ...) = Expresión*

Define LibPriv *Función(Param1, Param2, ...) = Func*

Bloque

EndFunc

Define LibPriv *Programa(Param1, Param2, ...) = Prgm*

Bloque

EndPrgm

Opera igual que **Define**, excepto porque define una variable de librería privada, función o programa. Las funciones y los programas privados no aparecen en el Catálogo.

Nota: Vea también **Define**, página 35 y **Define LibPub**, página 36.

Define LibPub

Catálogo > 

Define LibPub *Var* = *Expresión*

Define LibPub *Función*(*Param1*, *Param2*, ...) = *Expresión*

Define LibPub *Función*(*Param1*, *Param2*, ...) = **Func**

Bloque

EndFunc

Define LibPub *Programa*(*Param1*, *Param2*, ...) = **Prgm**

Bloque

EndPrgm

Opera igual que **Define**, excepto porque define una variable de librería pública, función o programa. Las funciones y los programas públicos aparecen en el Catálogo después de que la librería se ha guardado y actualizado.

Nota: Vea también **Define**, página 35 y **Define LibPriv**, página 35.

deltaList()

Vea **ΔList()**, página 69.

deltaTmpCnv()

Vea **ΔtmpCnv()**, página 131.

DelVar

Catálogo > 

DelVar *Var1* [, *Var2*] [, *Var3*] ...

DelVar *Var*.

Borra la variable o el grupo de variables especificado de la memoria.

Si una o más de las variables están bloqueadas, este comando despliega un mensaje de error y borra únicamente las variables no bloqueadas. Vea **unlock**, página 137.

DelVar *Var*. borra todos los miembros del grupo de variables *Var*. (como las estadísticas *stat.nn* los resultados o las variables que se crean con el uso de **LibShortcut()** función). El punto (.) en esta forma de comando **DelVar** lo limita a borrar un grupo de variables; la variable sencilla *Var* no se ve afectada.

$2 \rightarrow a$	2									
$(a+2)^2$	16									
DelVar a	Done									
$(a+2)^2$	$(a+2)^2$									
aa.a:=45	45									
aa.b:=5.67	5.67									
aa.c:=78.9	78.9									
getVarInfo()	<table border="1"> <tbody> <tr> <td>aa.a</td> <td>"NUM"</td> <td>"</td> </tr> <tr> <td>aa.b</td> <td>"NUM"</td> <td>"</td> </tr> <tr> <td>aa.c</td> <td>"NUM"</td> <td>"</td> </tr> </tbody> </table>	aa.a	"NUM"	" 	aa.b	"NUM"	" 	aa.c	"NUM"	" 
aa.a	"NUM"	" 								
aa.b	"NUM"	" 								
aa.c	"NUM"	" 								
DelVar aa.	Done									
getVarInfo()	"NONE"									

delVoid() (borrInVálido)

Catálogo > 

delVoid(*Lista1*) ⇒ *lista*

Entrega una lista que tiene el contenido de *Lista1* con todos los elementos (nulos) vacíos eliminados.

Para obtener más información sobre elementos vacíos, vea la página 166.

delVoid{ {1,void,3} } { 1,3 }

derivative()

Vea *d()*, página 155.

deSolve(EDO1erO2oGrado, Var, depVar)

⇒ una solución general

Entrega una ecuación que especifica en forma explícita o implícita una solución general para la ecuación diferencial ordinaria (EDO) de 1er o 2o grado. En la EDO:

- Use un símbolo primo (presione ) para denotar la 1a derivada de la variable dependiente con respecto de la variable independiente.
- Use dos símbolos primos para denotar la segunda derivada correspondiente.

El símbolo primo se usa para las derivadas dentro de resolverEd() únicamente. En otros casos, use **d**.

La solución general de una ecuación de 1er grado contiene una constante arbitraria de la forma ck , donde k es un sufijo de entero desde 1 hasta 255. La solución de una ecuación de 2o grado contiene dos constantes.

Aplice **solve()** para una solución implícita si desea tratar de convertirla en una o más soluciones explícitas equivalentes.

Cuando compare sus resultados con las soluciones del libro de texto o del manual, tome en cuenta que los diferentes métodos introducen constantes arbitrarias en distintos puntos en el cálculo, lo que puede producir soluciones generales diferentes.

deSolve(EDO1erGrado and condInic, Var, depVar)

⇒ una solución particular

Entrega una solución particular que satisface la EDO1erGrado y la condInic. Por lo general esto es más fácil que determinar una solución general, al sustituir los valores iniciales, solucionar la constante arbitraria y luego sustituir ese valor en la solución general.

condInic es una ecuación de la forma:

depVar (valorInicialIndependiente) = valorInicialDependiente

El valorInicialIndependiente y el valorInicialDependiente pueden ser variables como x_0 y y_0 que no tienen ningún valor almacenado. La diferenciación implícita puede ayudar a verificar las soluciones implícitas.

deSolve(EDO2oGrado and condInic1 and condInic2, Var, depVar)

⇒ una solución particular

Entrega una solución particular que satisface la EDO de 2o Grado y tiene un valor especificado de la variable dependiente y su primera derivada en un punto.

Para condInic1, use la forma:

depVar (valorInicialIndependiente) = valorInicialDependiente

Para condInic2, use la forma:

depVar (valorInicialIndependiente) = valorInicial1aDerivada

$$\text{deSolve}(y''+2y'+y=x^2, x, y)$$

$$y = (c3 \cdot x + c4) \cdot e^{-x} + x^2 - 4 \cdot x + 6$$

$$\text{right}(Ans) \rightarrow temp \quad (c3 \cdot x + c4) \cdot e^{-x} + x^2 - 4 \cdot x + 6$$

$$\frac{d^2}{dx^2}(temp) + 2 \cdot \frac{d}{dx}(temp) + temp - x^2$$

$$\text{DeIVar } temp \quad Done$$

$$\text{deSolve}(y' = (\cos(y))^2 \cdot x, x, y) \quad \tan(y) = \frac{x^2}{2} + c4$$

$$\text{solve}(Ans, y)$$

$$y = \tan^{-1}\left(\frac{x^2 + 2 \cdot c4}{2}\right) + n3 \cdot \pi$$

$$Ans | c4 = c - 1 \text{ and } n3 = 0$$

$$y = \tan^{-1}\left(\frac{x^2 + 2 \cdot (c - 1)}{2}\right)$$

$$\sin(y) = (y \cdot e^x + \cos(y)) \cdot y' \rightarrow ode$$

$$\sin(y) = (e^x \cdot y + \cos(y)) \cdot y'$$

$$\text{deSolve}(ode \text{ and } y(0) = 0, x, y) \rightarrow soln$$

$$\frac{-(2 \cdot \sin(y) + y^2)}{2} = (e^x - 1) \cdot e^{-x} \cdot \sin(y)$$

$$\text{soln} | x = 0 \text{ and } y = 0 \quad true$$

$$ode | y' = \text{impDif}(soln, x, y) \quad true$$

$$\text{DeIVar } ode, soln \quad Done$$

$$\text{deSolve}(y'' = y^2 \text{ and } y(0) = 0 \text{ and } y'(0) = 0, t, y)$$

$$\frac{3}{2} \cdot y^{\frac{4}{3}} = t$$

$$\text{solve}(Ans, y)$$

$$y = \frac{2}{4} \cdot \frac{4}{3} \cdot (3 \cdot t)^{\frac{3}{4}} \text{ and } t \geq 0$$

deSolve() (resolverEd)

Catálogo >

deSolve(EDO2oGrado and bndCond1 and condBnd2, Var, depVar) ⇒ una solución particular

Entrega una solución particular que satisface la EDO2oGrado y tiene valores especificados en dos puntos diferentes.

$$\text{deSolve}\left(w^{n-2}\cdot w^{\frac{1}{x^2}} + \left(9 + \frac{2}{x^2}\right) \cdot w \cdot x \cdot e^x \text{ and } w\left(\frac{\pi}{6}\right) = 0 \text{ and } w\left(\frac{\pi}{3}\right) = 0, x, w\right)$$

$$w = \frac{x \cdot e^x}{(\ln(e))^2 + 9} + \frac{e^3 \cdot x \cdot \cos(3 \cdot x)}{(\ln(e))^2 + 9} - \frac{e^6 \cdot x \cdot \sin(3 \cdot x)}{(\ln(e))^2 + 9}$$

det()

Catálogo >

det(matrizCuadrada[, Tolerancia]) ⇒ expresión

Entrega la determinante de *matrizCuadrada*.

De manera opcional, cualquier elemento de matriz se trata como cero si su valor absoluto es menor que la *Tolerancia*. Esta tolerancia se usa sólo si la matriz tiene ingresos de punto flotante y no contiene ninguna variable simbólica a la que no se le haya asignado un valor. De otro modo, la *Tolerancia* se ignora.

- Si usted usa **ctrl** **enter** o configura el modo **Auto o Aproximado** para aproximar, los cálculos se realizan al usar la aritmética de punto flotante.
- Si la *Tolerancia* se omite o no se usa, la tolerancia predeterminada se calcula como:

$$5E-14 \cdot \max(\text{dim}(\text{matrizCuadrada}) \cdot \text{rowNorm}(\text{matrizCuadrada}))$$

$$\det\begin{pmatrix} a & b \\ c & d \end{pmatrix} \quad a \cdot d - b \cdot c$$

$$\det\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \quad -2$$

$$\det\left(\text{identity}(3) - x \cdot \begin{pmatrix} 1 & -2 & 3 \\ -2 & 4 & 1 \\ -6 & -2 & 7 \end{pmatrix}\right)$$

$$-(98 \cdot x^3 - 55 \cdot x^2 + 12 \cdot x - 1)$$

$$\begin{bmatrix} 1. \text{E}20 & 1 \\ 0 & 1 \end{bmatrix} \rightarrow \text{mat1} \quad \begin{bmatrix} 1. \text{E}20 & 1 \\ 0 & 1 \end{bmatrix}$$

$$\det(\text{mat1}) \quad 0$$

$$\det(\text{mat1}, .1) \quad 1. \text{E}20$$

diag()

Catálogo >

diag(Lista) ⇒ matriz

diag(matrizFila) ⇒ matriz

diag(matrizColumna) ⇒ matriz

Entrega una matriz con los valores en la lista o matriz de argumentos en su diagonal principal.

diag(matrizCuadrada) ⇒ matrizFila

Entrega una matriz de filas que contiene los elementos de la diagonal principal de *matrizCuadrada*.

matrizCuadrada debe ser cuadrada.

$$\text{diag}([2 \ 4 \ 6]) \quad \begin{bmatrix} 2 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 6 \end{bmatrix}$$

$$\begin{bmatrix} 4 & 6 & 8 \\ 1 & 2 & 3 \\ 5 & 7 & 9 \end{bmatrix} \quad \begin{bmatrix} 4 & 6 & 8 \\ 1 & 2 & 3 \\ 5 & 7 & 9 \end{bmatrix}$$

$$\text{diag}(\text{Ans}) \quad \begin{bmatrix} 4 & 2 & 9 \end{bmatrix}$$

dim()

Catálogo >

dim(Lista) ⇒ entero

Entrega la dimensión de *Lista*.

dim(Matriz) ⇒ lista

Entrega las dimensiones de la matriz como una lista de dos elementos (filas, columnas).

dim(Cadena) ⇒ entero

Entrega el número de caracteres contenidos en la cadena de caracteres *Cadena*.

$$\text{dim}(\{0, 1, 2\}) \quad 3$$

$$\text{dim}\begin{pmatrix} 1 & -1 \\ 2 & -2 \\ 3 & 5 \end{pmatrix} \quad \{3, 2\}$$

$$\text{dim}(\text{"Hello"}) \quad 5$$

$$\text{dim}(\text{"Hello "&"there"}) \quad 11$$

Disp [*exprOCadena1*] [, *exprOCadena2*] ...

Despliega los argumentos en el historial de la Calculadora. Los argumentos se despliegan en sucesión, con espacios pequeños como separadores.

Es útil principalmente con programas y funciones para asegurar en despliegue de cálculos intermedios.

Nota para ingresar el ejemplo: En la aplicación de Calculadora en el dispositivo portátil, usted puede ingresar definiciones en líneas múltiples al presionar en lugar de al final de cada línea. En el teclado de la computadora, presione y sostenga **Alt** y presione **Ingresar**.

```
Define chars(start,end)=Prgm
  For i,start,end
  Disp i," ",char(i)
EndFor
EndPrgm
```

Done

chars(240,243)

240 ð

241 ñ

242 ò

243 ó

Done

DMS (►GMS)

Expr ►DMS

Lista ►DMS

Matriz ►DMS

Nota: Usted puede insertar este operador desde el teclado de la computadora al escribir @>DMS.

Interpreta el argumento como un ángulo y despliega el número GMS (GGGGG°MM'SS.ss") equivalente. Vea °, ', "" en la página 161 para el formato GMS (grado, minutos, segundos).

Nota: ►DMS se convertirá de radianes a grados cuando se use en el modo de Radián. Si la entrada va seguida de un símbolo de grados °, no ocurrirá ninguna conversión. Usted puede usar ►DMS sólo al final de una línea de ingreso.

En modo de ángulo en Grados:

$(45.371) \blacktriangleright \text{DMS}$	$45^{\circ}22'15.6''$
$(\{45.371,60\}) \blacktriangleright \text{DMS}$	$\{45^{\circ}22'15.6'',60^{\circ}\}$

domain() (dominio)

domain(*Expr1*, *Var*) \Rightarrow *expresión*

Devuelve el dominio de *Expr1* con respecto a *Var*.

domain() puede utilizarse para examinar los dominios de las funciones. Se restringe a un dominio real y finito.

Esta funcionalidad presenta limitaciones debido a defectos en los algoritmos de simplificación algebraicos para computadora y algoritmos solucionadores.

Algunas funciones no pueden ser utilizadas como argumentos para **domain()**, sin importar si aparecen explícitamente o dentro de las variables y funciones definidas por el usuario: En el siguiente ejemplo, la expresión no puede simplificarse porque $\int()$ no es una función permitida.

$$\text{domain}\left(\int \frac{x}{t} dt, x\right) \rightarrow \text{domain}\left(\int \frac{x}{t} dt, x\right)$$

$\text{domain}(x^2, x)$	$-\infty < x < \infty$
$\text{domain}\left(\frac{x+1}{x^2+2 \cdot x}, x\right)$	$x \neq -2$ and $x \neq 0$
$\text{domain}\left(\left(\sqrt{x}\right)^2, x\right)$	$0 \leq x < \infty$
$\text{domain}\left(\frac{1}{x+y}, y\right)$	$y \neq -x$

dominantTerm()

Catálogo >

dominantTerm(Expr1, Var [, Punto]) ⇒ expresión**dominantTerm**(Expr1, Var [, Punto]) | Var > Punto

⇒ expresión

dominantTerm(Expr1, Var [, Punto]) | Var < Punto

⇒ expresión

Entrega el término dominante de la representación de una serie de potencia de Expr1 expandida alrededor de Punto. El término dominante es aquel cuya magnitud crece con más rapidez cerca de Var = Punto. La potencia resultante de (Var - Punto) puede tener un exponente negativo y/o fraccional. El coeficiente de esta potencia puede incluir logaritmos de (Var - Punto) y otras funciones de Var que están dominadas por todas las potencias de (Var - Punto) teniendo el mismo signo de exponente.

Punto se predetermina a 0. Punto puede ser ∞ o -∞, en cuyos casos el término dominante será el término que tiene el exponente más grande de Var en lugar del exponente más pequeño de Var.

dominantTerm(...) entrega "**dominantTerm(...)**" si no puede determinar tal representación, como para singularidades esenciales como $\sin(1/z)$ en $z=0$, $e^{-1/z}$ en $z=0$, o e^z en $z = \infty$ o $-\infty$.

Si la serie o una de sus derivadas tiene una discontinuidad de salto en un Punto, es probable que el resultado contenga subexpresiones del signo de forma (...) o abs(...) para una variable de expansión real o (-1)^{parte real de z} para una variable de expansión compleja, que es una que termina con "...". Si usted pretende usar el término dominante sólo para valores en un lado de Punto, entonces anexe a **dominantTerm(...)** el apropiado de "**| Var > Punto**", "**| Var < Punto**", "**| Var ≥ Punto**" o "**| Var ≤ Punto**" para obtener un resultado más simple.

dominantTerm() se distribuye sobre listas y matrices del 1er argumento.

dominantTerm() es útil cuando usted desea conocer la expresión más simple posible que sea asintótica para otra expresión como Var → Punto. **dominantTerm()** también es útil cuando no es obvio cuál será el grado del primer término no-cero de una serie, y usted no desea calcular iterativamente, ya sea de manera interactiva o por medio de un bucle de programa.

Nota: Vea también **series()**, página 111.

$$\text{dominantTerm}(\tan(\sin(x)) - \sin(\tan(x)), x)$$

$$\frac{x^7}{30}$$

$$\text{dominantTerm}\left(\frac{1 - \cos(x-1)}{(x-1)^3}, x, 1\right)$$

$$\frac{1}{2 \cdot (x-1)}$$

$$\text{dominantTerm}\left(x^{-2} \cdot \tan\left(x^{\frac{1}{3}}\right), x\right)$$

$$\frac{1}{5x^3}$$

$$\text{dominantTerm}(\ln(x^x - 1) \cdot x^{-2}, x)$$

$$\frac{\ln(x \cdot \ln(x))}{x^2}$$

$$\text{dominantTerm}(e^{-z}, z)$$

$$\text{dominantTerm}(e^{-z}, z, 0)$$

$$\text{dominantTerm}\left(1 + \frac{1}{n}, n, \infty\right)$$

$$e$$

$$\text{dominantTerm}\left(\tan^{-1}\left(\frac{1}{x}\right), x, 0\right)$$

$$\frac{\pi \cdot \text{sign}(x)}{2}$$

$$\text{dominantTerm}\left(\tan^{-1}\left(\frac{1}{x}\right), x, x > 0\right)$$

$$\frac{\pi}{2}$$

dotP() (pPunto)

Catálogo >

dotP(Lista1, Lista2) ⇒ expresión

Entrega el producto "punto" de dos listas.

$$\text{dotP}(\{a, b, c\}, \{d, e, f\})$$

$$a \cdot d + b \cdot e + c \cdot f$$

$$\text{dotP}(\{1, 2\}, \{5, 6\})$$

$$17$$

dotP(Vector1, Vector2) ⇒ expresión

Entrega el producto punto de dos vectores.

$$\text{dotP}([a \ b \ c], [d \ e \ f])$$

$$a \cdot d + b \cdot e + c \cdot f$$

$$\text{dotP}([1 \ 2 \ 3], [4 \ 5 \ 6])$$

$$32$$

Ambos deben ser vectores de fila, o ambos deben ser vectores de columna.

E

e^0 [ex] tecla

e^(Expr1) ⇒ expresión

Entrega **e** elevado a la potencia de *Expr1*.

Nota: Vea también **plantilla de exponente e**, página 2.

Nota: Presionar **[ex]** para desplegar **e^** (es diferente de presionar el carácter **[E]** en el teclado).

Usted puede ingresar un número complejo en la forma polar $re^{i\theta}$. Sin embargo, use esta forma sólo en el modo de ángulo en Radianes; esto causa un error de Dominio en el modo de ángulo en Grados o en Gradianes.

e^(Lista1) ⇒ lista

Entrega **e** elevado a la potencia de cada elemento en *Lista1*.

e^(matrizCuadrada1) ⇒ *matrizCuadrada*

Entrega el exponencial de la matriz de *matrizCuadrada1*. Esto no es lo mismo que calcular **e** elevado a la potencia de cada elemento. Para obtener información acerca del método de cálculo, consulte **cos()**.

matrizCuadrada1 debe ser diagonalizable. El resultado siempre contiene números de punto flotante.

e^1	e
$e^{1.}$	2.71828
e^{3^2}	e^9

$e\{1,1,0.5\}$	$\{e,2.71828,1.64872\}$
----------------	-------------------------

$e \begin{bmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{bmatrix}$	$\begin{bmatrix} 782.209 & 559.617 & 456.509 \\ 680.546 & 488.795 & 396.521 \\ 524.929 & 371.222 & 307.879 \end{bmatrix}$
--	---

eff() Catálogo > [ex]

eff(tasaNominal,CpA) ⇒ valor

Función financiera que convierte la tasa de interés nominal *tasaNominal* en una tasa efectiva anual, donde *CpA* se da como el número de periodos de capitalización por año.

tasaNominal debe ser un número real y *CpA* debe ser un número real > 0.

Nota: Vea también **nom()**, página 84.

eff(5.75,12)	5.90398
---------------------	---------

eigVc() (vcProp) Catálogo > [ex]

eigVc(matrizCuadrada) ⇒ *matriz*

Entrega una matriz que contiene los vectores propios para una *matrizCuadrada* real o compleja, donde cada columna en el resultado corresponde a un valor propio. Tome en cuenta que un vector propio no es único; puede escalarse por medio de cualquier factor constante. Los vectores propios se normalizan, lo que significa que si $V = [x_1, x_2, \dots, x_n]$, entonces:

$$x_1^2 + x_2^2 + \dots + x_n^2 = 1$$

matrizCuadrada se balancea primero con transformaciones de similitud hasta que las normas de fila y columna están tan cerca del mismo valor como es posible. La *matrizCuadrada* se reduce entonces a una forma de Hessenberg superior y los vectores propios se generan o se obtienen por medio de la factorización de Schur.

En Formato Complejo Rectangular:

$\begin{bmatrix} -1 & 2 & 5 \\ 3 & -6 & 9 \\ 2 & -5 & 7 \end{bmatrix} \rightarrow mI$	$\begin{bmatrix} -1 & 2 & 5 \\ 3 & -6 & 9 \\ 2 & -5 & 7 \end{bmatrix}$
eigVc(mI)	$\begin{bmatrix} -0.800906 & 0.767947 & (\\ 0.484029 & 0.573804+0.052258 \cdot i & 0.5738 \\ 0.352512 & 0.262687+0.096286 \cdot i & 0.2626 \end{bmatrix}$

Para ver el resultado completo, presione **▲** y luego use **◀▶** para mover el cursor.

eigVl() (vlProp)

Catálogo >

eigVl(matrizCuadrada) ⇒ lista

Entrega una lista de valores propios de una *matrizCuadrada* real o compleja.

matrizCuadrada se balancea primero con transformaciones de similitud hasta que las normas de fila y columna están tan cerca del mismo valor como es posible. La *matrizCuadrada* se reduce entonces a una forma de Hessenberg superior y los vectores propios se generan o se obtienen por medio de la matriz de Hessenberg superior.

En modo de formato complejo Rectangular:

$$\begin{bmatrix} -1 & 2 & 5 \\ 3 & -6 & 9 \\ 2 & -5 & 7 \end{bmatrix} \rightarrow mL \qquad \begin{bmatrix} -1 & 2 & 5 \\ 3 & -6 & 9 \\ 2 & -5 & 7 \end{bmatrix}$$

$$\text{eigVl}(mL) \\ \{-4.40941, 2.20471 + 0.763006 \cdot i, 2.20471 - 0.763006 \cdot i\}$$

Para ver el resultado completo, presione \blacktriangle y luego use \blacktriangleleft y \blacktriangleright para mover el cursor.

Else (Más)

Vea If, página 58.

Elseif (MásSi)

Catálogo >

If ExprBooleana1 Then

Bloque1

Elseif ExprBooleana2 Then

Bloque2

⋮

Elseif ExprBooleanaN Then

BloqueN

Endif

⋮

Nota para ingresar el ejemplo: En la aplicación de Calculadora en el dispositivo portátil, usted puede ingresar definiciones en líneas múltiples al presionar $\boxed{\leftarrow}$ en lugar de $\boxed{\text{enter}}$ al final de cada línea. En el teclado de la computadora, presione y sostenga **Alt** y presione **Ingresar**.

Define $g(x) = \text{Func}$ If $x \leq -5$ Then

Return 5

Elseif $x > -5$ and $x < 0$ ThenReturn $-x$ Elseif $x \geq 0$ and $x \neq 10$ ThenReturn x Elseif $x = 10$ Then

Return 3

EndIf

EndFunc

Done

EndFor (TerminarPara)

Vea For, página 50.

EndFunc (TerminarFunc)

Vea Func, página 53.

EndIf (TerminarSi)

Vea If, página 58.

EndLoop (TerminarBucle)

Vea Loop, página 75.

EndPrgm (TerminarPrgm)

Vea Prgm, página 94.

EndTry (TerminarIntentar)

Vea Try, página 132.

euler()

Catálogo > 

euler(Expr, Var, varDep, {Var0, VarMax}, var0Dep, PasoVar [, pasoEuler]) matriz \Rightarrow

euler(SistemaDeExpr, Var, ListaDeVarsDep, {Var0, VarMax}, ListaDeVars0Dep, PasoVar [, pasoEuler]) matriz \Rightarrow

euler(ListaDeExpr, Var, ListaDeVarsDep, {Var0, VarMax}, ListaDeVars0Dep, PasoVar [, pasoEuler]) matriz \Rightarrow

Use el método de Euler para resolver el sistema

$$\frac{d \text{ depVar}}{d \text{ Var}} = \text{Expr}(\text{Var}, \text{varDep})$$

con $\text{varDep}(\text{Var}0)=\text{var}0\text{Dep}$ en el intervalo $[\text{Var}0, \text{VarMax}]$. Entregará una matriz cuya primera fila define los valores del resultado de Var y cuya segunda fila define el valor del primer componente de solución a los valores de Var correspondientes, y así sucesivamente.

Expr es el lado derecho que define la ecuación diferencial ordinaria (EDO).

SistemaDeExpr es el sistema de lados derechos que define el sistema de EDOs (corresponde al orden de variables dependientes en *ListaDeVarsDep*).

ListaDeExpr es una lista de lados derechos que define el sistema de EDOs (corresponde al orden de variables dependientes en *ListaDeVarsDep*).

Var es la variable independiente.

ListaDeVarsDep es una lista de variables dependientes.

{*Var0, VarMax*} es una lista de dos elementos que le dice a la función que se integre de *Var0* a *VarMax*.

ListaDeVars0Dep es una lista de valores iniciales para variables dependientes.

PasoVar es un número distinto de cero de manera que **sign**(*PasoVar*) = **sign**(*VarMax-Var0*) y las soluciones se entregan a $\text{Var}0+i \cdot \text{PasoVar}$ para todos $i=0,1,2,\dots$ de tal manera que $\text{Var}0+i \cdot \text{PasoVar}$ está en $[\text{var}0, \text{VarMax}]$ (puede que no haya un valor de solución en *VarMax*).

pasoEuler es un entero positivo (predeterminado a 1) que define el número de pasos de Euler entre los valores de resultado. El tamaño del paso real utilizado por el método de Euler es $\text{PasoVar} / \text{pasoEuler}$.

Ecuación diferencial:

$$y' = 0.001 \cdot y \cdot (100 - y) \text{ y } y(0) = 10$$

$$\text{euler}\left\{0.001 \cdot y \cdot (100 - y), t, y, \{0, 100\}, 10, 1\right\}$$

0.	1.	2.	3.	4.
10.	10.9	11.8712	12.9174	14.042

Para ver el resultado completo, presione \blacktriangle y después use \blacktriangleleft y

\blacktriangleright para mover el cursor.

Compare el resultado anterior con la solución exacta de CAS obtenido al usar de **Resolver()** y **genSec()**:

$$\text{deSolve}\{y' = 0.001 \cdot y \cdot (100 - y) \text{ and } y(0) = 10, t, y\}$$

$$y = \frac{100 \cdot (1.10517)^t}{(1.10517)^t + 9}$$

$$\text{seqGen}\left(\frac{100 \cdot (1.10517)^t}{(1.10517)^t + 9}, t, y, \{0, 100\}\right)$$

$$\{10., 10.9367, 11.9494, 13.0423, 14.2189\}$$

Sistema de ecuaciones:

$$y1' = y1 + 0.1 \cdot y1 \cdot y2$$

$$y2' = 3 \cdot y2 - y1 \cdot y2$$

$$\text{con } y1(0) = 2 \text{ y } y2(0) = 5$$

$$\text{euler}\left\{\begin{matrix} -y1 + 0.1 \cdot y1 \cdot y2 \\ 3 \cdot y2 - y1 \cdot y2 \end{matrix}, t, \{y1, y2\}, \{0.5\}, \{2.5\}, 1\right\}$$

0.	1.	2.	3.	4.	5.
2.	1.	1.	3.	27.	243.
5.	10.	30.	90.	90.	-2070.

exact()

Catálogo > 

exact(Expr1 [, Tolerancia]) \Rightarrow expresión

exact(Lista1 [, Tolerancia]) \Rightarrow lista

exact(Matriz1 [, Tolerancia]) \Rightarrow matriz

Usa aritmética de modo Exacto para producir, cuando es posible, el equivalente de número racional del argumento.

Tolerancia especifica la tolerancia para la conversión; la predeterminada es 0 (cero).

$$\text{exact}(0.25) \quad \frac{1}{4}$$

$$\text{exact}(0.333333) \quad \frac{333333}{1000000}$$

$$\text{exact}(0.333333, 0.001) \quad \frac{1}{3}$$

$$\text{exact}(3.5 \cdot x + y) \quad \frac{7 \cdot x}{2} + y$$

$$\text{exact}(\{0.2, 0.33, 4.125\}) \quad \left\{\frac{1}{5}, \frac{33}{100}, \frac{33}{8}\right\}$$

Exit (Salir)

Catálogo >

Exit

Sale del bloque **For**, **While**, o **Loop**.

Exit no está permitido afuera de las tres estructuras de bucles (**For**, **While**, o **Loop**).

Nota para ingresar el ejemplo: En la aplicación de Calculadora en el dispositivo portátil, usted puede ingresar definiciones en líneas múltiples al presionar en lugar de al final de cada línea. En el teclado de la computadora, presione y sostenga **Alt** y presione **Ingresar**.

Listado de funciones:

Define $g()$ = Func	Done
Local $temp, i$	
$0 \rightarrow temp$	
For $i, 1, 100, 1$	
$temp + i \rightarrow temp$	
If $temp > 20$ Then	
Exit	
EndIf	
EndFor	
EndFunc	
$g()$	21

Expr

Catálogo >

Expr Expr

Representa la *Expr* en términos del exponencial natural e . Este es un operador de conversión de despliegue. Se puede usar únicamente al final de la línea de ingreso.

Nota: Usted puede insertar este operador desde el teclado de la computadora al escribir $@>exp$.

$\frac{d}{dx}(e^x + e^{-x})$	$2 \cdot \sinh(x)$
$2 \cdot \sinh(x) \gg exp$	$e^{-x} \cdot (e^{2 \cdot x} - 1)$

exp()

tecla

exp(*Expr1*) \Rightarrow expresión

Entrega e elevado a la potencia de *Expr1*.

Nota: Vea también la plantilla exponencial e , página 2.

Usted puede ingresar un número complejo en la forma polar $re^{i\theta}$. Sin embargo, use esta forma sólo en el modo de ángulo en Radianes; esto causa un error de Dominio en el modo de ángulo en Grados o en Gradianes.

exp(*Lista1*) \Rightarrow lista

Entrega e elevada a la potencia de cada elemento en *Lista1*.

e^1	e
$e^{1.}$	2.71828
e^{3^2}	e^9
$e\{1, 1., 0.5\}$	$\{e, 2.71828, 1.64872\}$

exp(*matrizCuadrada1*) \Rightarrow *matrizCuadrada*

Entrega el exponencial de la matriz de *matrizCuadrada1*. Esto no es lo mismo que calcular e elevado a la potencia de cada elemento. Para obtener información acerca del método de cálculo, consulte **cos()**.

matrizCuadrada1 debe ser diagonalizable. El resultado siempre contiene números de punto flotante.

$\begin{bmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{bmatrix}$	$\begin{bmatrix} 782.209 & 559.617 & 456.509 \\ 680.546 & 488.795 & 396.521 \\ 524.929 & 371.222 & 307.879 \end{bmatrix}$
--	---

exp►list()Catálogo > **exp►list**(*Expr*, *Var*) ⇒ lista

Examina la *Expr* para las ecuaciones que están separadas por la palabra "or", y entrega una lista que contiene los lados derechos de las ecuaciones de la forma $Var=Expr$. Esto le brinda una forma fácil de extraer algunos valores de solución incrustados en los resultados de las funciones **solve()**, **cSolve()**, **fMin()**, y **fMax()**.

Nota: **exp►list()** no es necesaria con las funciones **zeros()** y **cZeros()** porque entregan una lista de valores de solución en forma directa.

Usted puede insertar esta función desde el teclado al escribir **exp@>list (...)**.

$$\begin{array}{l} \text{solve}(x^2-x-2=0,x) \quad x=-1 \text{ or } x=2 \\ \text{exp►list}(\text{solve}(x^2-x-2=0,x),x) \quad \{-1,2\} \end{array}$$

expand() (expandir)Catálogo > **expand**(*Expr1* [, *Var*]) ⇒ expresión**expand**(*Lista1* [, *Var*]) ⇒ lista**expand**(*Matriz1* [, *Var*]) ⇒ matriz

expand(*Expr1*) entrega *Expr1* expandida con respecto de todas sus variables. La expansión es una expansión polinómica para los polinomios y una expansión de fracción parcial para las expresiones racionales.

La meta de **expand()** es transformar *Expr1* en una suma y/o diferencia de términos sencillos. En contraste, la meta de **factor()** es transformar *Expr1* en un producto y/o cociente de factores sencillos.

expand(*Expr1*, *Var*) entrega *Expr1* expandida con respecto de *Var*. Se recopilan potencias similares de *Var*. Los términos y sus factores se ordenan con *Var* como la variable principal. Puede haber cierta factorización o expansión incidental de los coeficientes recopilados. Se compara para omitir *Var*, con frecuencia esto ahorra tiempo, memoria y espacio de pantalla, mientras que hace la expresión más comprensible.

$$\begin{array}{l} \text{expand}((x+y+1)^2) \quad x^2+2\cdot x\cdot y+2\cdot x\cdot y^2+2\cdot y+1 \\ \text{expand}\left(\frac{x^2-x+y^2-y}{x^2\cdot y^2-x^2\cdot y-x\cdot y^2+x\cdot y}\right) \quad \frac{1}{x-1} - \frac{1}{x} + \frac{1}{y-1} - \frac{1}{y} \end{array}$$

$$\begin{array}{l} \text{expand}((x+y+1)^2,y) \quad y^2+2\cdot y\cdot(x+1)+(x+1)^2 \\ \text{expand}((x+y+1)^2,x) \quad x^2+2\cdot x\cdot(y+1)+(y+1)^2 \\ \text{expand}\left(\frac{x^2-x+y^2-y}{x^2\cdot y^2-x^2\cdot y-x\cdot y^2+x\cdot y}\right) \quad \frac{1}{y-1} - \frac{1}{y} + \frac{1}{x\cdot(x-1)} \\ \text{expand}(Ans,x) \quad \frac{1}{x-1} - \frac{1}{x} + \frac{1}{y\cdot(y-1)} \end{array}$$

Incluso cuando hay sólo una variable, al usar *Var* se puede hacer la factorización del denominador que se usa para la expansión de la fracción parcial más completa.

Sugerencia: Para expresiones racionales, **propFrac()** es una alternativa más rápida aunque menos extrema para **expand()**.

Nota: Vea también **comDenom()** para un numerador expandido sobre un denominador expandido.

$$\begin{array}{l} \text{expand}\left(\frac{x^3+x^2-2}{x^2-2}\right) \quad \frac{2\cdot x}{x^2-2} + x+1 \\ \text{expand}(Ans,x) \quad \frac{1}{x-\sqrt{2}} + \frac{1}{x+\sqrt{2}} + x+1 \end{array}$$

expand() (expandir)

Catálogo >

expand(Expr1, [Var]) también distribuye logaritmos y potencias fraccionales independientemente de *Var*. Para una distribución incrementada de logaritmos y potencias fraccionales, podrían ser necesarias restricciones de desigualdad para garantizar que algunos factores son no negativos.

expand(Expr1, [Var]) también distribuye valores absoluto, **sign()**, y exponenciales, independientemente de *Var*.

Nota: Vea también **tExpand()** para suma de ángulo trigonométrico y expansión de ángulo múltiple.

$$\frac{\ln(2 \cdot x \cdot y) + \sqrt{2 \cdot x \cdot y}}{\text{expand}(Ans)} = \frac{\ln(2 \cdot x \cdot y) + \sqrt{2 \cdot x \cdot y}}{\ln(x \cdot y) + \sqrt{2 \cdot \sqrt{x \cdot y} + \ln(2)}}$$

$$\text{expand}(Ans) | y \geq 0$$

$$\frac{\ln(x) + \sqrt{2 \cdot \sqrt{x} \cdot \sqrt{y} + \ln(y)} + \ln(2)}{\text{sign}(x \cdot y) + |x \cdot y| + e^{2 \cdot x + y}}$$

$$\frac{e^{2 \cdot x + y} + \text{sign}(x \cdot y) + |x \cdot y|}{\text{expand}(Ans)}$$

$$\text{sign}(x) \cdot \text{sign}(y) + |x| \cdot |y| + (e^x)^2 \cdot e^y$$

expr()

Catálogo >

expr(Cadena) \Rightarrow expresión

Entrega la cadena de caracteres contenida en *Cadena* como una expresión y la ejecuta de inmediato.

$$\frac{\text{expr}("1+2+x^2+x")}{\text{expr}("expand((1+x)^2)")}$$

$$\frac{x^2+x+3}{x^2+2 \cdot x+1}$$

"Define cube(x)=x^3" \rightarrow *funcstr*

"Define cube(x)=x^3"

$$\frac{\text{expr}(funcstr)}{\text{cube}(2)}$$

$$\frac{Done}{8}$$

ExpReg

Catálogo >

ExpReg *X*, *Y* [, [*Frec*] [, [*Categoría*, *Incluir*]]

Genera la regresión exponencial $y = a \cdot (b)^x$ en listas *X* y *Y* con frecuencia *Frec*. Un resumen de resultados se almacena en la variable *stat.results*. (Vea la página 122.)

Todas las listas deben tener una dimensión igual, excepto por *Incluir*.

X y *Y* son listas de variables independientes y dependientes.

Frec es una lista opcional de valores de frecuencia. Cada elemento en *Frec* especifica la frecuencia de la ocurrencia para cada punto de datos *X* y *Y* correspondientes. El valor predeterminado es 1. Todos los elementos deben ser enteros ≥ 0 .

Categoría es una lista de códigos de categoría para los datos *X* y *Y* correspondientes.

Incluir es una lista de uno o más códigos de categoría. Sólo aquellos elementos de datos cuyo código de categoría está incluido en esta lista están incluidos en el cálculo.

Para obtener información sobre el efecto de los elementos vacíos en una lista, vea "Elementos vacíos (inválidos)" en la página 166.

Variable de salida	Descripción
stat.EcnReg	Ecuación de regresión: $a \cdot (b)^x$
stat.a, stat.b	Coefficientes de regresión

Variable de salida	Descripción
stat.r ²	Coefficiente de determinación lineal para datos transformados
stat.r	Coefficiente de correlación para datos transformados (x, ln(y))
stat.Resid	Residuales asociados con el modelo exponencial
stat.TransResid	Residuales asociadas con el ajuste lineal de datos transformados
stat.XReg	La lista de puntos de datos en <i>Lista X</i> modificada se usa de hecho en la regresión con base en las restricciones de las Categorías <i>Frec</i> , <i>Lista de Categorías Incluir</i>
stat.YReg	La lista de puntos de datos en <i>Lista Y</i> modificada se usa de hecho en la regresión con base en las restricciones de las Categorías <i>Frec</i> , <i>Lista de Categorías Incluir</i>
stat.FrecReg	Lista de frecuencias correspondientes a <i>stat.XReg</i> y <i>stat.YReg</i>

F

factor()

Catálogo > 

factor(*Expr1*, *Var*) ⇒ expresión

factor(*Lista1*, *Var*) ⇒ lista

factor(*Matriz1*, *Var*) ⇒ matriz

factor(*Expr1*) entrega *Expr1* factorizado con respecto de todas sus variables sobre un denominador común.

Expr1 se factoriza tanto como es posible hacia los factores racionales lineales sin introducir nuevas subexpresiones no reales. Esta alternativa es apropiada si se desea una factorización con respecto de más de una variable.

factor(*Expr1*, *Var*) entrega *Expr1* factorizado con respecto de la variable *Var*.

Expr1 se factoriza tanto como es posible hacia factores reales que son lineales en *Var*, incluso si introduce constantes irracionales o subexpresiones que son irracionales en otras variables.

Los factores y sus términos se clasifican con *Var* como la variable principal. Se recopilan potencias similares de *Var* en cada factor. Incluya *Var* si se necesita la factorización con respecto de sólo esa variable y usted está dispuesto a aceptar expresiones irracionales en otras variables para incrementar la factorización con respecto de *Var*. Podría haber cierta factorización incidental con respecto de otras variables.

Para la configuración Automática del modo **Auto o Aproximado**, incluyendo *Var* permite la aproximación con coeficientes de punto flotante, donde los coeficientes irracionales no se pueden expresar en forma explícita concisamente en términos de funciones integradas. Incluso cuando hay sólo una variable, incluyendo *Var*, puede producir una factorización más completa.

Nota: Vea también **comDenom()** para obtener una forma rápida de lograr una factorización parcial cuando **factor()** no es lo suficientemente rápido o si agota la memoria.

Nota: Vea también **cFactor()** para factorizar hasta los coeficientes complejos en busca de factores lineales.

$$\frac{\text{factor}(a^3 \cdot x^2 - a \cdot x^2 - a^3 + a)}{a \cdot (a-1) \cdot (a+1) \cdot (x-1) \cdot (x+1)}$$

$$\frac{\text{factor}(x^2+1)}{x^2+1}$$

$$\frac{\text{factor}(x^2-4)}{(x-2) \cdot (x+2)}$$

$$\frac{\text{factor}(x^2-3)}{x^2-3}$$

$$\frac{\text{factor}(x^2-a)}{x^2-a}$$

$$\frac{\text{factor}(a^3 \cdot x^2 - a \cdot x^2 - a^3 + a, x)}{a \cdot (a^2-1) \cdot (x-1) \cdot (x+1)}$$

$$\frac{\text{factor}(x^2-3, x)}{(x+\sqrt{3}) \cdot (x-\sqrt{3})}$$

$$\frac{\text{factor}(x^2-a, x)}{(x+\sqrt{a}) \cdot (x-\sqrt{a})}$$

$$\frac{\text{factor}(x^5+4 \cdot x^4+5 \cdot x^3-6 \cdot x-3)}{x^5+4 \cdot x^4+5 \cdot x^3-6 \cdot x-3}$$

$$\frac{\text{factor}(x^5+4 \cdot x^4+5 \cdot x^3-6 \cdot x-3, x)}{(x-0.964673) \cdot (x+0.6111649) \cdot (x+2.12543) \cdot (x^2)}$$

factor()

Catálogo >

factor(*númeroRacional*) entrega el número racional factorizado en primos. Para números compuestos, el tiempo de cómputo aumenta exponencialmente con el número de dígitos en el segundo factor más grande. Por ejemplo, factorizar un entero de 30 dígitos podría llevarse más de un día, y factorizar un número de 100 dígitos podría llevarse más de un siglo.

<code>factor(152417172689)</code>	123457 · 1234577
<code>isPrime(152417172689)</code>	false

Para detener el cálculo manualmente:

- **Windows®:** Mantenga presionada la tecla **F12** y presione **Intro** varias veces.
- **Macintosh®:** Mantenga presionada la tecla **FS** y presione **Intro** varias veces.
- **Dispositivo portátil:** Mantenga presionada la tecla **on** y presione varias veces.

Si usted simplemente desea determinar si un número es primo, use **isPrime()** en su lugar. Es mucho más rápido, en particular si *númeroRacional* no es primo y si el segundo factor más grande tiene más de cinco dígitos.

F Cdf()

Catálogo >

F Cdf(*limiteInferior, limiteSuperior, númerodf, denomdf*) ⇒ *número* si *limiteInferior* y *limiteSuperior* son números, lista si *limiteInferior* y *limiteSuperior* son listas

FCdf(*limiteInferior, limiteSuperior, númerodf, denomdf*) ⇒ *número* si *limiteInferior* y *limiteSuperior* son números, lista si *limiteInferior* y *limiteSuperior* son listas

Calcula la probabilidad de la distribución **F** entre el *Limite inferior* y *Limite Superior* para los grados de libertad *dfNumer* y *dfDenom* especificados.

Para $P(X \leq \text{Limite superior})$, establecer *Limite Inferior*=0.

Fill (Llenar)

Catálogo >

Fill *Expr, varMatriz* ⇒ *matriz*

Reemplaza cada elemento en la variable *varMatriz* con *Expr*. *varMatriz* ya debe existir.

$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$	→ <i>amatrix</i>	$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$
Fill 1.01, <i>amatrix</i>		Done
<i>amatrix</i>		$\begin{bmatrix} 1.01 & 1.01 \\ 1.01 & 1.01 \end{bmatrix}$

Fill *Expr, varLista* ⇒ *lista*

Reemplaza cada elemento en la variable *varLista* con *Expr*. *varLista* ya debe existir.

$\{1, 2, 3, 4, 5\}$	→ <i>alist</i>	$\{1, 2, 3, 4, 5\}$
Fill 1.01, <i>alist</i>		Done
<i>alist</i>		$\{1.01, 1.01, 1.01, 1.01, 1.01\}$

FiveNumSummary X , [$Frec$], [$Categoría$, $Incluir$]

Proporciona una versión abreviada de las estadísticas de 1 variable en la lista X . Un resumen de resultados se almacena en la variable $stat.results$. (Vea la página 122.)

X representa una lista que contiene los datos.

$Frec$ es una lista opcional de valores de frecuencia. Cada elemento en $Frec$ especifica la frecuencia de la ocurrencia para cada punto de datos X y Y correspondientes. El valor predeterminado es 1.

$Categoría$ es una lista de códigos de categoría numérica para los datos X correspondientes.

$Incluir$ es una lista de uno o más códigos de categoría. Sólo aquellos elementos de datos cuyo código de categoría está incluido en esta lista están incluidos en el cálculo.

Un elemento (inválido) vacío en cualquiera de las listas X , $Frec$, o $Categoría$ da como resultado un inválido para el elemento correspondiente de todas esas listas. Para obtener más información sobre elementos vacíos, vea la página 166.

Variable de salida	Descripción
stat.MinX	Mínimo de valores x .
stat.C ₁ X	1er Cuartil de x .
stat.MedianaX	Mediana de x .
stat.C ₃ X	3er Cuartil de x .
stat.MaxX	Máximo de valores x .

floor() (piso)

floor($Expr1$) \Rightarrow entero

Entrega el entero más grande que es \leq el argumento. Esta función es idéntica a **int()**.

El argumento puede ser un número real o complejo.

floor($Lista1$) \Rightarrow lista

floor($Matriz1$) \Rightarrow matriz

Entrega una lista o matriz del piso de cada elemento.

Nota: Vea también **ceiling()** e **int()**.

$$\text{floor}(-2.14) \quad -3.$$

$$\text{floor}\left(\left\{\frac{3}{2}, 0, -5.3\right\}\right) \quad \{1, 0, -6.\}$$

$$\text{floor}\left(\begin{pmatrix} 1.2 & 3.4 \\ 2.5 & 4.8 \end{pmatrix}\right) \quad \begin{pmatrix} 1. & 3. \\ 2. & 4. \end{pmatrix}$$

fMax()

fMax($Expr$, Var) \Rightarrow expresión Booleana

fMax($Expr$, Var , $limiteInferior$)

fMax($Expr$, Var , $limiteInferior$, $limiteSuperior$)

fMax($Expr$, Var) | $limiteInferior \leq Var \leq limiteSuperior$

Entrega una expresión Booleana que especifica valores candidatos de Var que maximizan $Expr$ o ubican su límite superior menor.

$$\text{fMax}\left(1-(x-a)^2-(x-b)^2, x\right) \quad x = \frac{a+b}{2}$$

$$\text{fMax}\left(.5 \cdot x^3 - x - 2, x\right) \quad x = \infty$$

fMax()

Catálogo >

Puede utilizar el operador restrictivo ("|") para restringir el intervalo de solución o especificar otras restricciones.

$$fMax(0.5 \cdot x^3 - x - 2, x) | x \leq 1 \quad x = -0.816497$$

Para la configuración aproximada del modo **Auto o Aproximado**, **fMax()** busca iterativamente un máximo local aproximado. Con frecuencia esto es más rápido, en particular si usted usa el operador "|" para restringir la búsqueda a un intervalo relativamente pequeño que contiene exactamente un máximo local.

Nota: Vea también **fMin()** y **Max()**.

fMin()

Catálogo >

fMin(Expr, Var) ⇒ expresión Booleana

fMin(Expr, Var, limiteInferior)

fMin(Expr, Var, limiteInferior, limiteSuperior)

fMin(Expr, Var) | limiteInferior ≤ Var ≤ limiteSuperior

$$fMin(1 - (x-a)^2 - (x-b)^2, x) \quad x = \infty \text{ or } x = -\infty$$

$$fMin(0.5 \cdot x^3 - x - 2, x) | x \geq 1 \quad x = 1$$

Entrega una expresión Booleana que especifica valores candidatos de *Var* que minimizan *Expr* o ubican su límite inferior mayor.

Puede utilizar el operador restrictivo ("|") para restringir el intervalo de solución o especificar otras restricciones.

Para la configuración aproximada del modo **Auto o Aproximado**, **fMin()** busca iterativamente un mínimo local aproximado. Con frecuencia esto es más rápido, en particular si usted usa el operador "|" para restringir la búsqueda a un intervalo relativamente pequeño que contiene exactamente un mínimo local.

Nota: Vea también **fMax()** y **min()**.

For (Para)

Catálogo >

For *Var, Bajo, Alto* [, *Paso*]
Bloque

EndFor

Ejecuta las sentencias en *Bloque* iterativamente para cada valor de *Var*, desde *Bajo* hasta *Alto*, en incrementos de *Paso*.

Var no debe ser una variable de sistema.

Paso puede ser positivo o negativo. El valor predeterminado es 1.

Bloque puede ser una sentencia sencilla o una serie de sentencias separadas con el carácter ";".

Nota para ingresar el ejemplo: En la aplicación de Calculadora en el dispositivo portátil, usted puede ingresar definiciones en líneas múltiples al presionar en lugar de al final de cada línea. En el teclado de la computadora, presione y sostenga **Alt** y presione **Ingresar**.

```
Define g()=Func
  Local tempsum,step,i
  0 → tempsum
  1 → step
  For i,1,100,step
    tempsum+i → tempsum
  EndFor
EndFunc
```

$$g() \quad 5050$$

format()Catálogo > **format**(*Expr*, *cadenaFormato*) ⇒ *cadena*

Entrega *Expr* como una cadena de caracteres con base en la plantilla de formato.

Expr debe simplificarse a un número.

cadenaFormato es una cadena y debe ser en la forma: "F[n]", "S[n]", "E[n]", "G[n][c]", donde [] indican porciones adicionales.

F[n]: Formato fijo. n es el número de dígitos a desplegar después del punto decimal.

S[n]: Formato científico. n es el número de dígitos a desplegar después del punto decimal.

E[n]: Formato de ingeniería. n es el número de dígitos después del primer dígito significativo. El exponente se ajusta a un múltiplo de tres, y el punto decimal se mueve hacia la derecha por cero, uno o dos dígitos.

G[n][c]: Igual que el formato fijo, pero también separa los dígitos hacia la izquierda de la raíz en grupos de tres. c especifica el carácter del separador del grupo y se predetermina a una coma. Si c es un punto, la raíz se mostrará como una coma.

[Rc]: Cualquiera de los especificadores anteriores puede tener un sufijo con la bandera de la raíz Rc, donde c es un carácter sencillo que especifica qué sustituir para el punto de la raíz.

<code>format(1.234567, "f3")</code>	"1.235"
<code>format(1.234567, "s2")</code>	"1.23E0"
<code>format(1.234567, "e3")</code>	"1.235E0"
<code>format(1.234567, "g3")</code>	"1.235"
<code>format(1234.567, "g3")</code>	"1,234.567"
<code>format(1.234567, "g3,r:")</code>	"1:235"

fPart() (parteF)Catálogo > **fPart**(*Expr*) ⇒ *expresión***fPart**(*Lista*) ⇒ *lista***fPart**(*Matriz*) ⇒ *matriz*

Entrega la parte fraccional del argumento.

Para una lista o matriz, entrega las partes fraccionales de los elementos.

El argumento puede ser un número real o complejo.

<code>fPart(-1.234)</code>	-0.234
<code>fPart({1,-2.3,7.003})</code>	{0,-0.3,0.003}

Fpdf()Catálogo > 

Fpdf(*XVal*, *númerodf*, *denomdf*) ⇒ *número* si *XVal* es un número, *lista* si *XVal* es una lista

Resuelve la probabilidad de distribución **F** en *XVal* para los *númerodf* (grados de libertad) y *denomdf* especificados.

freqTable►list()Catálogo > **freqTable►list**(*Lista1*,*listaEnteroFrec*) ⇒ *lista*

Entrega una lista que contiene los elementos desde *Lista1* expandida de acuerdo con las frecuencias en *listaEnteroFrec*. Esta función se puede usar para construir una tabla de frecuencia para la aplicación de Datos y Estadísticas.

Lista1 puede ser cualquier lista válida.

listaEnteroFrec debe tener la misma dimensión que *Lista1* y debe contener sólo elementos enteros no negativos. Cada elemento especifica el número de veces que el elemento de *Lista1* correspondiente se repetirá en la lista de resultados. Un valor de cero excluye el elemento de *Lista1* correspondiente.

Nota: Usted puede insertar esta función desde el teclado de la computadora al escribir **freqTable►list**(...).

Los elementos vacíos (anulados) se ignoran. Para obtener más información sobre elementos vacíos, vea la página 166.

$$\text{freqTable►list}\left(\{1,2,3,4\},\{1,4,3,1\}\right)$$

$$\{1,2,2,2,2,3,3,3,4\}$$

$$\text{freqTable►list}\left(\{1,2,3,4\},\{1,4,0,1\}\right)$$

$$\{1,2,2,2,2,4\}$$
frequency (frecuencia)Catálogo > **frequency**(*Lista1*,*listaCajones*) ⇒ *lista*

Entrega una lista que contiene los conteos de los elementos en *Lista1*. Los conteos se basan en los rangos (cajones) que usted define en *listaCajones*.

Si *listaCajones* es {*b*(1), *b*(2), ..., *b*(*n*)}, los rangos especificados son {*?≤b*(1), *b*(1)<*?≤b*(2), ..., *b*(*n*-1)<*?≤b*(*n*), *b*(*n*)>*?*}. La lista resultante es un elemento más largo que *listaCajones*.

Cada elemento del resultado corresponde al número de elementos de *Lista1* que están en el rango de ese cajón. Expresado en términos de la función **countff()**, el resultado es { **conteoSi**(*lista*, *?≤b*(1)), **conteoSi**(*lista*, *b*(1)<*?≤b*(2)), ..., **conteoSi**(*lista*, *b*(*n*-1)<*?≤b*(*n*)), **conteoSi**(*lista*, *b*(*n*)>*?*)}.

Los elementos de *Lista1* que no pueden estar "colocados en un cajón" se ignoran. Los elementos (inválidos) vacíos también se ignoran. Para obtener más información sobre elementos vacíos, vea la página 166.

Dentro de la aplicación Listas y Hoja de Cálculo, usted puede usar un rango de celdas en lugar de ambos argumentos.

Nota: Vea también **countff()**, página 27.

$$\text{datalist}:=\{1,2,e,3,\pi,4,5,6,\text{"hello"},7\}$$

$$\{1,2,2,71828,3,3.14159,4,5,6,\text{"hello"},7\}$$

$$\text{frequency}\{\text{datalist},\{2.5,4.5\}\}$$

$$\{2,4,3\}$$

Explicación del resultado:

2 elementos de *listaDatos* son ≤ 2.5

4 elementos de *listaDatos* son > 2.5 y ≤ 4.5

3 elementos de *listaDatos* son > 4.5

El elemento "hola" es una cadena y no se puede colocar en ninguno de los cajones definidos.

FTest_2SampCatálogo > **FTest_2Samp** *Lista1*,*Lista2*,*Frec1*,*Frec2*,*Hipot*]]**FTest_2Samp** *Lista1*,*Lista2*,*Frec1*,*Frec2*,*Hipot*]]

(Entrada de lista de datos)

FTest_2Samp *sx1*,*n1*,*sx2*,*n2*,*Hipot*]]**FTest_2Samp** *sx1*,*n1*,*sx2*,*n2*,*Hipot*]]

(Entrada de estadísticas de resumen)

Realiza una prueba F° de dos muestras. Un resumen de resultados se almacena en la variable *stat.results*. (Vea la página 122.)

Para $H_a: \sigma_1 > \sigma_2$, configurar *Hipot*>0

Para $H_a: \sigma_1 \neq \sigma_2$ (predeterminado), configurar *Hipot*=0

Para $H_a: \sigma_1 < \sigma_2$, configurar *Hipot*<0

Para obtener información sobre el efecto de los elementos vacíos en una lista, vea "Elementos vacíos (inválidos)" en la página 166.

Variable de salida	Descripción
stat.F	Estadística F calculada para la secuencia de datos
stat.ValP	Nivel más bajo de significancia en el cual la hipótesis nula se puede rechazar
stat.númerodf	grados de libertad del numerador = $n-1$
stat.denomdf	grados de libertad del denominador = $n-1$
stat.sx1, stat.sx2	Desviaciones estándar de muestra de las secuencias de datos en <i>Lista 1</i> y <i>Lista 2</i>
stat.x1_bar stat.x2_bar	Muestra significa las secuencias de datos en <i>Lista 1</i> y <i>Lista 2</i>
stat.n1, stat.n2	Tamaño de las muestras

Func

Catálogo >

Func

Bloque

EndFunc

Plantilla para crear una función definida por el usuario.

Bloque puede ser una sentencia sencilla, una serie de sentencias separadas con el carácter ":" o una serie de sentencias en líneas separadas. La función puede usar la instrucción **Return** para producir un resultado específico.

Nota para ingresar el ejemplo: En la aplicación de Calculadora en el dispositivo portátil, usted puede ingresar definiciones en líneas múltiples al presionar en lugar de al final de cada línea. En el teclado de la computadora, presione y sostenga **Alt** y presione **Ingresar**.

Defina una función de compuesto de variables:

Define $g(x)=\text{Func}$

Done

If $x < 0$ Then

Return $3 \cdot \cos(x)$

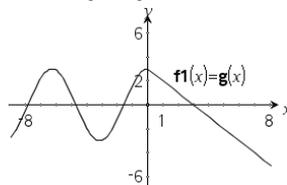
Else

Return $3-x$

EndIf

EndFunc

Resultado de graficar $g(x)$



G

gcd() (mcd)

Catálogo >

$\text{gcd}(\text{Número1}, \text{Número2}) \Rightarrow$ expresión

Entrega el máximo común divisor de los dos argumentos. El **gcd** de dos fracciones es el **gcd** de sus numeradores dividido entre el **lcm** de sus denominadores.

En el modo de Auto o Aproximado, el **gcd** de los números de punto flotante es 1.0.

$\text{gcd}(\text{Lista1}, \text{Lista2}) \Rightarrow$ lista

Entrega los máximos comunes divisores de los elementos correspondientes en *Lista 1* y *Lista 2*.

$\text{gcd}(18, 33)$

3

$\text{gcd}(\{12, 14, 16\}, \{9, 7, 5\})$

$\{3, 7, 1\}$

gcd() (mcd)

Catálogo >

gcd(Matriz1, Matriz2) ⇒ matriz

Entrega los máximos comunes divisores de los elementos correspondientes en Matriz1 y Matriz2.

$$\text{gcd}\left(\begin{bmatrix} 2 & 4 \\ 6 & 8 \end{bmatrix}, \begin{bmatrix} 4 & 8 \\ 12 & 16 \end{bmatrix}\right) \quad \begin{bmatrix} 2 & 4 \\ 6 & 8 \end{bmatrix}$$

geomCdf()

Catálogo >

geomCdf(p, límiteInferior, límiteSuperior) ⇒ número si límiteInferior y límiteSuperior son números, lista si límiteInferior y límiteSuperior son listas**geomCdf**(p, límiteSuperior) para

P(1 ≤ X ≤ límiteSuperior) ⇒ número si límiteSuperior es un número, lista si límiteSuperior es una lista

Resuelve una probabilidad geométrica acumulativa desde límiteInferior hasta límiteSuperior con la probabilidad de éxito p especificada.

Para P(X ≤ límiteSuperior), configure límiteInferior = 1.

geomPdf()

Catálogo >

geomPdf(p, XVal) ⇒ número si XVal es un número, lista si XVal es una lista

Resuelve una probabilidad en XVal, el número de la prueba en la que ocurre el primer éxito, para la distribución geométrica discreta con la probabilidad de éxito p.

getDenom()

Catálogo >

getDenom(Expr1) ⇒ expresión

Transforma el argumento en una expresión que tiene un denominador común reducido, y después entrega su denominador.

$\text{getDenom}\left(\frac{x+2}{y-3}\right)$	$y-3$
$\text{getDenom}\left(\frac{2}{7}\right)$	7
$\text{getDenom}\left(\frac{1}{x} + \frac{y^2+y}{y^2}\right)$	$x \cdot y$

getLangInfo() (obtnfoldioma)

Catálogo >

getLangInfo() ⇒ cadena

Entrega una cadena que corresponde al nombre corto del idioma activo actualmente. Por ejemplo, usted puede usarlo en un programa o una función para determinar el idioma actual.

Inglés = "en"
 Danés = "da"
 Alemán = "de"
 Finlandés = "fi"
 Francés = "fr"
 Italiano = "it"
 Holandés = "nl"
 Holandés belga = "nl_BE"
 Noruego = "no"
 Portugués = "pt"
 Español = "es"
 Sueco = "sv"

$$\text{getLangInfo}() \quad \text{"en"}$$

getLockInfo()

Catálogo >

getLockInfo(*Var*) ⇒ *valor*

Entrega el estado de bloqueada/desbloqueada actual de la variable *Var*.

valor = 0: *Var* está desbloqueada o no existe.

valor = 1: *Var* está bloqueada y no se puede modificar ni borrar.

Vea **Lock**, página 71 y **unLock**, página 137.

<i>a</i> :=65	65
Lock <i>a</i>	Done
getLockInfo(<i>a</i>)	1
<i>a</i> :=75	"Error: Variable is locked."
DelVar <i>a</i>	"Error: Variable is locked."
Unlock <i>a</i>	Done
<i>a</i> :=75	75
DelVar <i>a</i>	Done

getMode()

Catálogo >

getMode(*EnteroNombreModo*) ⇒ *valor***getMode**(0) ⇒ *lista*

getMode(*EnteroNombreModo*) entrega un valor que representa la configuración actual del modo *EnteroNombreModo*.

getMode(0) entrega una lista que contiene pares de números.

Cada par consiste en un entero de modo y un entero de configuración.

Para obtener un listado de modos y sus configuraciones, consulte la tabla de abajo.

Si usted guarda las configuraciones con **getMode**(0) → *var*, podrá usar **setMode**(*var*) en una función o un programa para restaurar temporalmente las configuraciones dentro de la ejecución de la función o el programa únicamente. Vea **setMode**(0), página 112.

getMode(0)	{ 1,1,2,1,3,1,4,1,5,1,6,1,7,1,8,1 }
getMode(1)	1
getMode(8)	1

Modo Nombre	Modo Entero	Cómo configurar enteros
Desplegar dígitos	1	1=Flotante, 2=Flotante1, 3=Flotante2, 4=Flotante3, 5=Flotante4, 6=Flotante5, 7=Flotante6, 8=Flotante7, 9=Flotante8, 10=Flotante9, 11=Flotante10, 12=Flotante11, 13=Flotante12, 14=Fijo0, 15=Fijo1, 16=Fijo2, 17=Fijo3, 18=Fijo4, 19=Fijo5, 20=Fijo6, 21=Fijo7, 22=Fijo8, 23=Fijo9, 24=Fijo10, 25=Fijo11, 26=Fijo12
Ángulo	2	1=Radián, 2=Grado, 3=Gradián
Formato exponencial	3	1=Normal, 2=Científico, 3=Ingeniería
Real o Complejo	4	1=Real, 2=Rectangular, 3=Polar
Auto o Aprox.	5	1=Auto, 2=Aproximado, 3=Exacto
Formato de Vector	6	1=Rectangular, 2=Cilíndrico, 3=Esférico
Base	7	1=Decimal, 2=Hexagonal, 3=Binario
Sistema de unidad	8	1=SI, 2=Ing/EEUU

getNum()

Catálogo >

getNum(*Expr1*) ⇒ *expresión*

Transforma el argumento en una expresión que tiene un denominador común reducido, y después entrega su numerador.

$\text{getNum}\left(\frac{x+2}{y-3}\right)$	$x+2$
$\text{getNum}\left(\frac{2}{7}\right)$	2
$\text{getNum}\left(\frac{1}{x} + \frac{1}{y}\right)$	$x+y$

getType()

Catálogo >

getType(*var*) *cadena* ⇒Entrega una cadena que indica el tipo de datos de la variable *var*.Si *var* no se ha definido, entrega la cadena "NINGUNA".

$\{1,2,3\} \rightarrow temp$	$\{1,2,3\}$
$\text{getType}(temp)$	"LIST"
$2.3 \cdot i \rightarrow temp$	$3 \cdot i$
$\text{getType}(temp)$	"EXPR"
$\text{DelVar } temp$	<i>Done</i>
$\text{getType}(temp)$	"NONE"

getVarInfo()

Catálogo >

getVarInfo() ⇒ *matriz o cadena***getVarInfo**(*CadenaNombreLib*) ⇒ *matriz o cadena***getVarInfo**() entrega una matriz de información (nombre de variable, tipo, accesibilidad de librería y estado de bloqueada/desbloqueada) para todas las variables y los objetos de librería definidos en el problema actual.Si no hay ninguna variable definida, **getVarInfo**() entrega la cadena "NINGUNA".**getVarInfo**(*CadenaNombreLib*) entrega una matriz de información para todos los objetos de librería definidos en la librería *CadenaNombreLib*. *CadenaNombreLib* debe ser una cadena (texto encerrado entre comillas) o una variable de cadena.Si la librería *CadenaNombreLib* no existe, ocurrirá un error.

$\text{getVarInfo}()$	"NONE"
Define $x=5$	<i>Done</i>
Lock x	<i>Done</i>
Define LibPriv $y=\{1,2,3\}$	<i>Done</i>
Define LibPub $z(x)=3 \cdot x^2 - x$	<i>Done</i>
$\text{getVarInfo}()$	$\begin{bmatrix} x & \text{"NUM"} & \text{"{ }"} & 1 \\ y & \text{"LIST"} & \text{"LibPriv"} & 0 \\ z & \text{"FUNC"} & \text{"LibPub"} & 0 \end{bmatrix}$
$\text{getVarInfo}(tmp3)$	"Error: Argument must be a string"
$\text{getVarInfo}("tmp3")$	$\begin{bmatrix} volcyl2 & \text{"NONE"} & \text{"LibPub"} & 0 \end{bmatrix}$

getVarInfo()

Catálogo >

Tome en cuenta el ejemplo de la izquierda, en el cual el resultado de **getVarInfo()** se asigna a la variable *vs*. Intentar desplegar la fila 2 ó la fila 3 de *vs* entrega un error de "Lista o matriz inválida" porque al menos uno de los elementos en esas filas (variable *b*, por ejemplo) se reevalúa a una matriz.

Este error también podría ocurrir cuando se usa *Ans* para reevaluar un resultado de **getVarInfo()**.

El sistema arroja el error anterior porque la versión actual del software no soporta una estructura de matriz generalizada donde un elemento de una matriz puede ser una matriz o una lista.

$a:=1$	1															
$b:=[1\ 2]$	$[1\ 2]$															
$c:=[1\ 3\ 7]$	$[1\ 3\ 7]$															
$vs:=\text{getVarInfo}()$	<table border="1"> <tr> <td><i>a</i></td> <td>"NUM"</td> <td>"{"</td> <td>"}"</td> <td>0</td> </tr> <tr> <td><i>b</i></td> <td>"MAT"</td> <td>"{"</td> <td>"}"</td> <td>0</td> </tr> <tr> <td><i>c</i></td> <td>"MAT"</td> <td>"{"</td> <td>"}"</td> <td>0</td> </tr> </table>	<i>a</i>	"NUM"	"{"	"}"	0	<i>b</i>	"MAT"	"{"	"}"	0	<i>c</i>	"MAT"	"{"	"}"	0
<i>a</i>	"NUM"	"{"	"}"	0												
<i>b</i>	"MAT"	"{"	"}"	0												
<i>c</i>	"MAT"	"{"	"}"	0												
$vs[1]$	$[1\ \text{"NUM"}\ \text{"{"}\ \text{"}"]$															
$vs[1,1]$	1															
$vs[2]$	"Error: Invalid list or matrix"															
$vs[2,1]$	$[1\ 2]$															

Goto (IrA)

Catálogo >

Goto nombreEtiqueta

Transfiere el control a la etiqueta *nombreEtiqueta*.

nombreEtiqueta se debe definir en la misma función al usar una instrucción **Lbl**.

Nota para ingresar el ejemplo: En la aplicación de Calculadora en el dispositivo portátil, usted puede ingresar definiciones en líneas múltiples al presionar en lugar de al final de cada línea. En el teclado de la computadora, presione y sostenga **Alt** y presione **Ingresar**.

Define $g()$ =Func	Done
Local <i>temp,i</i>	
$0 \rightarrow temp$	
$1 \rightarrow i$	
Lbl top	
$temp+i \rightarrow temp$	
If $i < 10$ Then	
$i+1 \rightarrow i$	
Goto top	
EndIf	
Return temp	
EndFunc	
$g()$	55

►Grad

Catálogo >

Expr1 ► **Grad** \Rightarrow *expresión*

Convierte *Expr1* para la medida de ángulo en gradianes.

Nota: Usted puede insertar este operador desde el teclado de la computadora al escribir **@>Grad**.

En modo de ángulo en Grados:	
$(1.5) \blacktriangleright \text{Grad}$	$(1.66667)^{\circ}$
En modo de ángulo en Radianes:	
$(1.5) \blacktriangleright \text{Grad}$	$(95.493)^{\circ}$

identity() (identidad)Catálogo > **identity(Entero)** \Rightarrow matrizEntrega la matriz de identidad con una dimensión de *Entero*.*Entero* debe ser un entero positivo.

identity(4)	1	0	0	0
	0	1	0	0
	0	0	1	0
	0	0	0	1

If (Si)Catálogo > **If** *SentenciaExprBooleana***If** *ExprBooleana* **Then***Bloque***Endif**

Si *ExprBooleana* se evalúa como verdadera, ejecuta una sentencia sencilla *Sentencia* o el bloque de sentencias *Bloque* antes de continuar con la ejecución.

Si *ExprBooleana* se evalúa como falsa, continúa la ejecución sin ejecutar la sentencia o el bloque de sentencias.

Bloque puede ser una sentencia sencilla o una secuencia de sentencias separadas con el caracter ";".

Nota para ingresar el ejemplo: En la aplicación de Calculadora en el dispositivo portátil, usted puede ingresar definiciones en líneas múltiples al presionar  en lugar de  al final de cada línea. En el teclado de la computadora, presione y sostenga **Alt** y presione **Ingresar**.

If *ExprBooleana* **Then***Bloque1***Else***Bloque2***Endif**

Si *ExprBooleana* se evalúa como verdadera, ejecuta *Bloque1* y luego se salta *Bloque2*.

Si *ExprBooleana* se evalúa como falsa, se salta *Bloque1* pero ejecuta *Bloque2*.

Bloque1 y *Bloque2* pueden ser una sentencia sencilla.

```
Define g(x)=Func                               Done
    If x<0 Then
        Return x2
    EndIf
EndFunc
```

g(-2)	4
--------------	---

```
Define g(x)=Func                               Done
    If x<0 Then
        Return -x
    Else
        Return x
    EndIf
EndFunc
```

g(12)	12
--------------	----

g(-12)	12
---------------	----

If *ExprBooleana1* **Then**
Bloque1
Elseif *ExprBooleana2* **Then**
Bloque2
 ⋮
Elseif *ExprBooleanaN* **Then**
BloqueN
Endif

Permite la ramificación. Si *ExprBooleana1* se evalúa como verdadera, ejecuta *Bloque1*. Si *ExprBooleana1* se evalúa como falsa, evalúa *ExprBooleana2*, y así sucesivamente.

```
Define g(x)=Func
  If x<5 Then
    Return 5
  ElseIf x>5 and x<0 Then
    Return -x
  ElseIf x≥0 and x≠10 Then
    Return x
  ElseIf x=10 Then
    Return 3
  EndIf
EndFunc
```

Done

$g(-4)$	4
$g(10)$	3

ifFn(*ExprBooleana*, *Valor_Si_verdadero* [, *Valor_Si_falso* [, *Valor_Si_desconocido*]]) \Rightarrow expresión, lista o matriz

Evalúa la expresión booleana *ExprBooleana* (o cada elemento de *ExprBooleana*) y produce un resultado con base en las siguientes reglas:

- *ExprBooleana* puede probar un valor sencillo, una lista o una matriz.
- Si un elemento de *ExprBooleana* se evalúa como verdadero, entrega el elemento correspondiente de *Valor_Si_verdadero*.
- Si un elemento de *ExprBooleana* se evalúa como falso, entrega el elemento correspondiente de *Valor_Si_falso*. Si usted omite *Valor_Si_falso*, entrega indef.
- Si un elemento de *ExprBooleana* no es ni verdadero ni falso, entrega el elemento correspondiente *Valor_Si_desconocido*. Si usted omite *Valor_Si_desconocido*, entrega indef.
- Si el segundo, tercer o cuarto argumento de la función **ifFn()** es una expresión sencilla, la prueba Booleana se aplica a cada posición en *ExprBooleana*.

Nota: Si la sentencia *ExprBooleana* simplificada incluye una lista o matriz, todos los demás argumentos de la lista o matriz deben tener la(s) misma(s) dimensión(es), y el resultado tendrá la(s) misma(s) dimensión(es).

```
ifFn({1,2,3}<2.5,{5,6,7},{8,9,10})
      {5,6,10}
```

El valor de prueba de **1** es menor que 2.5, entonces su elemento *Valor_Si_Verdadero* correspondiente de **5** se copia en la lista de resultados.

El valor de prueba de **2** es menor que 2.5, entonces su elemento *Valor_Si_Verdadero* correspondiente de **6** se copia en la lista de resultados.

Valor de prueba de **3** no es menor que 2.5, entonces su elemento *Valor_Si_Falso* correspondiente de **10** se copia en la lista de resultados.

```
ifFn({1,2,3}<2.5,4,{8,9,10})      {4,4,10}
```

Valor_Si_verdadero es un valor sencillo y corresponde a cualquier posición seleccionada.

```
ifFn({1,2,3}<2.5,{5,6,7})          {5,6,undef}
```

Valor_Si_falso no está especificado. Se usa Indeterminado o indefinido.

```
ifFn({2,"a"<2.5,{6,7},{9,10},"err")
      {6,"err"}
```

Un elemento seleccionado de *Valor_Si_verdadero*. Un elemento seleccionado de *Valor_Si_desconocido*.

imag(*Expr1*) \Rightarrow expresión

Entrega la parte imaginaria del argumento.

Nota: Todas las variables indefinidas se tratan como variables reales. Vea también **real()**, página . 100

```
imag(1+2·i)      2
```

```
imag(z)          0
```

```
imag(x+i·y)     y
```

imag()Catálogo > **imag(Lista1)** ⇒ lista

Entrega una lista de las partes imaginarias de los elementos.

$\text{imag}(\{-3, 4 - i, i\})$	$\{0, -1, 1\}$
---------------------------------	----------------

imag(Matriz1) ⇒ matriz

Entrega una matriz de las partes imaginarias de los elementos.

$\text{imag}\left(\begin{bmatrix} a & b \\ i \cdot c & i \cdot d \end{bmatrix}\right)$	$\begin{bmatrix} 0 & 0 \\ c & d \end{bmatrix}$
--	--

impDif()Catálogo > **impDif(Ecuación, Var, varDepend[, Ord])**

⇒ expresión

donde la orden *Ord* se predetermina en 1.

Resuelve la derivada implícita para las ecuaciones en las que una variable se define en forma implícita en términos de otra.

$\text{impDif}(x^2 + y^2 = 100, x, y)$	$\frac{-x}{y}$
--	----------------

Indirección

Vea #(), página 159.

inString() (enCadena)Catálogo > **inString(cadenaBúsqueda, subCadena[, Iniciar])** ⇒ enteroEntrega la posición de caracteres en la cadena *cadenaBúsqueda* en la cual comienza la primera ocurrencia de la cadena *subCadena*.*Iniciar*, si se incluye, especifica la posición de caracteres dentro de *cadenaBúsqueda* donde comienza la búsqueda. Predeterminado = 1 (el primer carácter de *cadenaBúsqueda*).Si *cadenaBúsqueda* no contiene *subCadena* o *Iniciar* es > la longitud de *cadenaBúsqueda*, entrega cero.

$\text{inString}(\text{"Hello there"}, \text{"the"})$	7
$\text{inString}(\text{"ABCEFG"}, \text{"D"})$	0

int()Catálogo > **int(Expr)** ⇒ entero**int(Lista1)** ⇒ lista**int(Matriz1)** ⇒ matrizEntrega el entero más grande que es menor que o igual al argumento. Esta función es idéntica a **floor()**.

El argumento puede ser un número real o complejo.

Para una lista o matriz, entrega el entero más grande de los elementos.

$\text{int}(-2.5)$	-3.
$\text{int}([-1.234 \ 0 \ 0.37])$	$[-2. \ 0 \ 0.]$

intDiv()Catálogo > **intDiv(Número1, Número2)** ⇒ entero**intDiv(Lista1, Lista2)** ⇒ lista**intDiv(Matriz1, Matriz2)** ⇒ matrizEntrega la parte del entero signado de (*Número1* ÷ *Número2*).

Para listas y matrices, entrega la parte del entero signado de (argumento 1 ÷ argumento 2) para cada par de elementos.

$\text{intDiv}(-7, 2)$	-3
$\text{intDiv}(4, 5)$	0
$\text{intDiv}(\{12, -14, -16\}, \{5, 4, -3\})$	$\{2, -3, 5\}$

integral

Vea ∫(), página 155.

interpolate()Catálogo > **interpolate**(valorX, listaX, listaY, ListaPrimaY) lista ⇒

Esta función hace lo siguiente:

Dadas *listaX*, *listaY*=**f**(*listaX*) y *ListaPrimaY*=**f'**(*listaX*) para cierta función desconocida **f**, se usa una interpolación cúbica para aproximar la función **f** al *valorX*. Se supone que *listaX* es una lista de números monótonicamente crecientes o decrecientes, aunque esta función puede entregar un valor incluso cuando no lo es. Esta función avanza a través de *listaX* en busca de un intervalo [*listaX*[*i*], *listaX*[*i*+1]] que contenga un *valorX*. Si encuentra dicho intervalo, entrega un valor interpolado para **f**(*valorX*); de otro modo, entrega **undef**.

listaX, *listaY* y *ListaPrimaY* deben tener la misma dimensión ≥ 2 y contener expresiones que se simplifiquen a números.

valorX puede ser una variable indefinida, un número o una lista de números.

Ecuación diferencial:

$$y' = -3y + 6t + 5 \text{ y } y(0) = 5$$

$$rk := rk23(-3y + 6t + 5, t, y, \{0, 10\}, 5, 1)$$

0.	1.	2.	3.	4.
5.	3.19499	5.00394	6.99957	9.00593

Para ver el resultado completo, presione  ydespués use   para mover el cursor.Use la función `interpolat()` para calcular los valores de la función para la `listavlorx`:

```
xvaelist:=seq(i,i,0,10,0.5)
{0,0.5,1.,1.5,2.,2.5,3.,3.5,4.,4.5,5.,5.5,6.,6.5,
xlist:=mat▶list(rk[1])
{0.,1.,2.,3.,4.,5.,6.,7.,8.,9.,10.}
ylist:=mat▶list(rk[2])
{5.,3.19499,5.00394,6.99957,9.00593,10.9976
yprimelist:=-3y+6t+5|y=ylist and t=xlist
{-10.,1.41503,1.98819,2.00129,1.98221,2.006
interpolat(xvaelist,xlist,ylist,yprimelist)
{5.,2.67062,3.19499,4.02782,5.00394,6.00011▶
```

invχ²()Catálogo > **invχ²**(Área,df)**invChi2**(Área,df)

Resuelve la función de probabilidad (ji cuadrado) acumulativa Inversa χ^2 especificada por el grado de libertad, *df* para un *Área* dada debajo de la curva.

invF()Catálogo > **invF**(Área,númerodf,denomdf)**invF**(Área,númerodf,denomdf)

resuelve la función de distribución de **F** acumulativa Inversa especificada por *númerodf* y *denomdf* para un *Área* dada bajo la curva.

invNorm()Catálogo > **invNorm**(Área,μ,σ)

Resuelve la función de distribución normal acumulativa inversa para un *Área* dada bajo la curva de distribución normal especificada por μ y σ .

invt()Catálogo > **invt**(Área,df)

Resuelve la función de probabilidad del estudiante t acumulativa Inversa especificada por el grado de libertad *df* para un *Área* dada bajo la curva.

iPart()

Catálogo >

iPart(Expr) ⇒ entero**iPart**(Lista1) ⇒ lista**iPart**(Matriz1) ⇒ matriz

Entrega la parte de entero del argumento.

Para listas y matrices, entrega la parte de entero de cada elemento.

El argumento puede ser un número real o complejo.

$iPart(-1.234)$	-1.
$iPart\left(\left\{\frac{3}{2}, -2.3, 7.003\right\}\right)$	$\{1, -2., 7.\}$

irr()

Catálogo >

irr(CF0, ListaFE [, FrecFE]) ⇒ valor

La función financiera que calcula la tasa interna de rendimiento de una inversión.

FE0 es el flujo de efectivo inicial en tiempo 0; debe ser un número real.

ListaFE es una lista de cantidades de flujo de efectivo después del flujo de efectivo inicial FE0.

FrecFE es una lista opcional en la cual cada elemento especifica la frecuencia de ocurrencia para una cantidad de flujo de efectivo (consecutivo) agrupado, que es el elemento correspondiente de la ListaFE. La predeterminada es 1; si usted ingresa valores, éstos deben ser enteros positivos < 10,000.

Nota: Vea también **mirr()**, página 78.

$list1 := \{6000, -8000, 2000, -3000\}$	$\{6000, -8000, 2000, -3000\}$
$list2 := \{2, 2, 2, 1\}$	$\{2, 2, 2, 1\}$
$irr(5000, list1, list2)$	-4.64484

isPrime()

Catálogo >

isPrime(Número) ⇒ expresión de constante Booleana

Entrega verdadero o falso para indicar si número es un número entero ≥ 2 que es divisible equitativamente sólo entre sí mismo y 1.

Si Número excede alrededor de 306 dígitos y no tiene ningún factor ≤ 1021, **isPrime**(Número) despliega un mensaje de error.Si usted sólo desea determinar si Número es primo, use **isPrime()** en lugar de **factor()**. Es mucho más rápido, en particular si Número no es primo y si tiene un segundo factor más grande que excede alrededor de cinco dígitos.**Nota para ingresar el ejemplo:** En la aplicación de Calculadora en el dispositivo portátil, usted puede ingresar definiciones en líneas múltiples al presionar en lugar de **enter** al final de cada línea. En el teclado de la computadora, presione y sostenga **Alt** y presione **Ingresar**.

$isPrime(5)$	true
$isPrime(6)$	false

Función para encontrar el siguiente primo después de un número especificado:

Define $nextprim(n) =$	Func	Done
	Loop	
	$n+1 \rightarrow n$	
	If $isPrime(n)$	
	Return n	
	EndLoop	
	EndFunc	
$nextprim(7)$		11

isVoid() (esInválido)

Catálogo >

isVoid(Var) ⇒ expresión de constante Booleana**isVoid**(Expr) ⇒ expresión de constante Booleana**isVoid**(Lista) ⇒ expresiones de constante Booleana

Entrega verdadero o falso para indicar si el argumento es un tipo de datos inválido.

Para obtener más información sobre elementos inválidos, vea la página 166.

$a := _$	-
$isVoid(a)$	true
$isVoid(\{1, _, 3\})$	$\{false, true, false\}$

E

Lbl (Etiqu)

Catálogo >

Lbl nombreEtiqueta

Define una etiqueta con el nombre *nombreEtiqueta* dentro de una función.

Usted puede usar una instrucción **Goto** *nombreEtiqueta* para transferir el control a la instrucción que sigue inmediatamente a la etiqueta.

nombreEtiqueta debe cumplir con los mismos requisitos de nombrado que un nombre de variable.

Nota para ingresar el ejemplo: En la aplicación de Calculadora en el dispositivo portátil, usted puede ingresar definiciones en líneas múltiples al presionar en lugar de al final de cada línea. En el teclado de la computadora, presione y sostenga **Alt** y presione **Ingresar**.

```
Define g() $\Rightarrow$ Func Done
  Local temp,i
  0  $\rightarrow$  temp
  1  $\rightarrow$  i
  Lbl top
  temp+i  $\rightarrow$  temp
  If i<10 Then
  i+1  $\rightarrow$  i
  Goto top
  EndIf
  Return temp
EndFunc
```

$g()$ 55

lcm() (mínimo común múltiplo)

Catálogo >

lcm(Número1, Número2) \Rightarrow expresión

lcm(Lista1, Lista2) \Rightarrow lista

lcm(Matriz1, Matriz2) \Rightarrow matriz

Entrega el mínimo común múltiplo de los dos argumentos. El **lcm** de dos fracciones es el **lcm** de sus numeradores dividido entre el **gcd** de sus denominadores. El **lcm** de los números de punto flotante fraccional es su producto.

Para dos listas o matrices, entrega los mínimos comunes múltiplos de los elementos correspondientes.

$lcm(6,9)$ 18

$lcm\left(\left\{\frac{1}{3}, -14, 16\right\}, \left\{\frac{2}{15}, 7, 5\right\}\right)$ $\left\{\frac{2}{3}, 14, 80\right\}$

left() (izquierda)

Catálogo >

left(cadenaFuente[, Num]) \Rightarrow cadena

Entrega los caracteres de *Num* del extremo izquierdo contenidos en una cadena de caracteres *cadenaFuente*.

Si usted omite *Num*, entrega toda la *cadenaFuente*.

left(Lista1[, Num]) \Rightarrow lista

Entrega los elementos de *Num* del extremo izquierdo contenidos en *Lista1*.

Si usted omite *Num*, entrega toda la *Lista1*.

left(Comparación) \Rightarrow expresión

Entrega el lado del extremo izquierdo de una ecuación o desigualdad.

$left("Hello", 2)$ "He"

$left(\{1, 3, -2, 4\}, 3)$ $\{1, 3, -2\}$

$left(x < 3)$ x

LinRegBx $X, Y, [Frec], [Categoría, Incluir]$

Resuelve la regresión lineal $y = a + b \cdot x$ en las listas X y Y con frecuencia $Frec$. Un resumen de resultados se almacena en la variable *resultados.estad*. (Vea la página 122.)

Todas las listas deben tener una dimensión igual, excepto por *Incluir*.

X y Y son listas de variables independientes y dependientes.

$Frec$ es una lista opcional de valores de frecuencia. Cada elemento en $Frec$ especifica la frecuencia de la ocurrencia para cada punto de datos X y Y correspondientes. El valor predeterminado es 1. Todos los elementos deben ser enteros ≥ 0 .

$Categoría$ es una lista de códigos de categoría para los datos X y Y correspondientes.

Incluir es una lista de uno o más códigos de categoría. Sólo aquellos elementos de datos cuyo código de categoría está incluido en esta lista están incluidos en el cálculo.

Para obtener información sobre el efecto de los elementos vacíos en una lista, vea "Elementos vacíos (inválidos)" en la página 166.

Variable de salida	Descripción
stat.EcnReg	Ecuación de regresión: $a + b \cdot x$
stat.a, stat.b	Coefficientes de regresión
stat.r ²	Coefficiente de determinación
stat.r	Coefficiente de correlación
stat.Resid	Residuales de la regresión
stat.XReg	La lista de puntos de datos en <i>Lista X</i> modificada se usa de hecho en la regresión con base en las restricciones de las Categorías <i>Frec</i> , <i>Lista de Categorías Incluir</i>
stat.YReg	La lista de puntos de datos en <i>Lista Y</i> modificada se usa de hecho en la regresión con base en las restricciones de las Categorías <i>Frec</i> , <i>Lista de Categorías Incluir</i>
stat.FrecReg	Lista de frecuencias correspondientes a <i>stat.XReg</i> y <i>stat.YReg</i>

LinRegMx $X, Y, [Frec], [Categoría, Incluir]$

Resuelve la regresión lineal $y = m \cdot x + b$ en las listas X y Y con frecuencia $Frec$. Un resumen de resultados se almacena en la variable $stat.results$. (Vea la página 122.)

Todas las listas deben tener una dimensión igual, excepto por *Incluir*.

X y Y son listas de variables independientes y dependientes.

$Frec$ es una lista opcional de valores de frecuencia. Cada elemento en $Frec$ especifica la frecuencia de la ocurrencia para cada punto de datos X y Y correspondientes. El valor predeterminado es 1. Todos los elementos deben ser enteros ≥ 0 .

$Categoría$ es una lista de códigos de categoría para los datos X y Y correspondientes.

Incluir es una lista de uno o más códigos de categoría. Sólo aquellos elementos de datos cuyo código de categoría está incluido en esta lista están incluidos en el cálculo.

Para obtener información sobre el efecto de los elementos vacíos en una lista, vea "Elementos vacíos (inválidos)" en la página 166.

Variable de salida	Descripción
stat.EcnReg	Ecuación de regresión: $y = m \cdot x + b$
stat.m, stat.b	Coefficientes de regresión
stat.r ²	Coefficiente de determinación
stat.r	Coefficiente de correlación
stat.Resid	Residuales de la regresión
stat.XReg	La lista de puntos de datos en <i>Lista X</i> modificada se usa de hecho en la regresión con base en las restricciones de las Categorías <i>Frec</i> , <i>Lista de Categorías Incluir</i>
stat.YReg	La lista de puntos de datos en <i>Lista Y</i> modificada se usa de hecho en la regresión con base en las restricciones de las Categorías <i>Frec</i> , <i>Lista de Categorías Incluir</i>
stat.FrecReg	Lista de frecuencias correspondientes a <i>stat.XReg</i> y <i>stat.YReg</i>

LinRegIntervals $X, Y[, F[, 0[, NivC]]]$

Para Pendiente. Resuelve en un intervalo de confianza de nivel C para la pendiente.

LinRegIntervals $X, Y[, F[, 1, val[X[, nivC]]]$

Para Respuesta. Resuelve un valor "y" previsto en un intervalo de predicción de nivel C para una observación sencilla, así como un intervalo de confianza de nivel C para la respuesta promedio.

Un resumen de resultados se almacena en la variable *stat.results*. (Vea la página 122.)

Todas las listas deben tener una dimensión igual.

X y Y son listas de variables independientes y dependientes.

F es una lista opcional de valores de frecuencia. Cada elemento en F especifica la frecuencia de la ocurrencia para cada punto de datos X y Y correspondientes. El valor predeterminado es 1. Todos los elementos deben ser enteros ≥ 0 .

Para obtener información sobre el efecto de los elementos vacíos en una lista, vea "Elementos vacíos (inválidos)" en la página 166.

Variable de salida	Descripción
stat.EcnReg	Ecuación de regresión: $a+b \cdot x$
stat.a, stat.b	Coefficientes de regresión
stat.df	Grados de libertad
stat.r ²	Coefficiente de determinación
stat.r	Coefficiente de correlación
stat.Resid	Residuales de la regresión

Únicamente para un tipo de pendiente

Variable de salida	Descripción
[stat.CBajo, stat.CAlto]	Intervalo de confianza para la pendiente.
stat.ME	Margen de error del intervalo de confianza
stat.EEPendiente	Error estándar de pendiente
stat.s	Error estándar sobre la línea

Para tipo de Respuesta únicamente

Variable de salida	Descripción
[stat.CBajo, stat.CAlto]	Intervalo de confianza para la respuesta promedio
stat.ME	Margen de error del intervalo de confianza
stat.EE	Error estándar de respuesta promedio

Variable de salida	Descripción
[stat.PredBaja, stat.PredAlta]	Intervalo de predicción para una observación sencilla
stat.MEPred	Margen de error del intervalo de predicción
stat.EEPred	Error estándar para la predicción
stat. \hat{y}	$a + b \cdot \text{val}X$

LinRegTTest

Catálogo > 

LinRegTTest $X, Y[, Frec[, Hipot]]$

Resuelve una regresión lineal en las listas X y Y y una prueba t en el valor de la pendiente β y el coeficiente de correlación ρ para la ecuación $y = \alpha + \beta x$. Prueba la hipótesis nula $H_0: \beta = 0$ (equivalentemente, $\rho = 0$) contra una de las tres hipótesis alternativas.

Todas las listas deben tener una dimensión igual.

X y Y son listas de variables independientes y dependientes.

$Frec$ es una lista opcional de valores de frecuencia. Cada elemento en $Frec$ especifica la frecuencia de la ocurrencia para cada punto de datos X y Y correspondientes. El valor predeterminado es 1. Todos los elementos deben ser enteros ≥ 0 .

$Hipot$ es un valor opcional que especifica una de las tres hipótesis alternativas contra la cual se probará la hipótesis nula ($H_0: \beta = \rho = 0$).

Para $H_a: \beta \neq 0$ y $\rho \neq 0$ (predeterminada), configuran $Hipot = 0$

Para $H_a: \beta < 0$ y $\rho < 0$, configuran $Hipot < 0$

Para $H_a: \beta > 0$ y $\rho > 0$, configuran $Hipot > 0$

Un resumen de resultados se almacena en la variable $stat.results$. (Vea la página 122.)

Para obtener información sobre el efecto de los elementos vacíos en una lista, vea "Elementos vacíos (inválidos)" en la página 166.

Variable de salida	Descripción
stat.EcnReg	Ecuación de regresión: $a + b \cdot x$
stat.t	t -Estadística para prueba de significancia
stat.ValP	Nivel más bajo de significancia en el cual la hipótesis nula se puede rechazar
stat.df	Grados de libertad
stat.a, stat.b	Coefficientes de regresión
stat.s	Error estándar sobre la línea
stat.EEPendiente	Error estándar de pendiente
stat. r^2	Coefficiente de determinación
stat.r	Coefficiente de correlación
stat.Resid	Residuales de la regresión

linSolve()

Catálogo >

linSolve(*SistemaDeEcnLineales*, *Var1*, *Var2*, ...) ⇒ lista**linSolve**(*EcnLineal1* and *EcnLineal2* and ...,*Var1*, *Var2*, ...) ⇒ lista**linSolve**(*EcnLineal1*, *EcnLineal2*, ..., {*Var1*, *Var2*, ...})

⇒ lista

linSolve(*SistemaDeEcnLineales*, {*Var1*, *Var2*, ...})

⇒ lista

linSolve(*EcnLineal1* and *EcnLineal2* and ...,{*Var1*, *Var2*, ...}) ⇒ lista**linSolve**(*EcnLineal1*, *EcnLineal2*, ..., {*Var1*, *Var2*, ...})

⇒ lista

Entrega una lista de soluciones para las variables *Var1*, *Var2*, ...

El primer argumento se debe evaluar para un sistema de ecuaciones lineales o una ecuación lineal sencilla. De otro modo, ocurrirá un error de argumento.

Por ejemplo, evaluar **linSolve**($x=1$ y $x=2$, x) produce un resultado de "Error de Argumento".

$$\text{linSolve}\left(\left\{\begin{array}{l} 2x+4y=3 \\ 5x-3y=7 \end{array}\right\}, \{x,y\}\right) \quad \left\{\begin{array}{l} 37 \\ 26 \end{array}, \frac{1}{26}\right\}$$

$$\text{linSolve}\left(\left\{\begin{array}{l} 2x=3 \\ 5x-3y=7 \end{array}\right\}, \{x,y\}\right) \quad \left\{\frac{3}{2}, \frac{1}{6}\right\}$$

$$\text{linSolve}\left(\left\{\begin{array}{l} \text{apple}+4\text{pear}=23 \\ 5\text{apple}-\text{pear}=17 \end{array}\right\}, \{\text{apple},\text{pear}\}\right)$$

$$\left\{\frac{13}{3}, \frac{14}{3}\right\}$$

$$\text{linSolve}\left(\left\{\begin{array}{l} \text{apple}+4\frac{\text{pear}}{3}=14 \\ -\text{apple}+\text{pear}=6 \end{array}\right\}, \{\text{apple},\text{pear}\}\right)$$

$$\left\{\frac{36}{13}, \frac{114}{13}\right\}$$

ΔList()

Catálogo >

ΔList(*Lista1*) ⇒ lista**Nota:** Usted puede insertar esta función desde el teclado al escribir **Δlist**(...).Entrega una lista que contiene las diferencias entre los elementos consecutivos en *Lista1*. Cada elemento de *Lista1* se sustrae del siguiente elemento de *Lista1*. La lista resultante siempre es un elemento más corto que la *Lista1* original.

$$\Delta\text{List}(\{20,30,45,70\}) \quad \{10,15,25\}$$

list▶mat()

Catálogo >

list▶mat(*Lista* [, *elementosPorFila*]) ⇒ matrizEntrega una matriz llenada fila por fila con los elementos de *Lista1*.*elementosPorFila*, si están incluidos, especifica el número de elementos por fila. El predeterminado es el número de elementos en *Lista* (una fila).Si *Lista* no llena la matriz resultante, se agregan ceros.**Nota:** Usted puede insertar esta función desde el teclado de la computadora al escribir **list▶mat**(...).

$$\text{list}\blacktriangleright\text{mat}(\{1,2,3\}) \quad \begin{array}{|c|c|c|} \hline 1 & 2 & 3 \\ \hline \end{array}$$

$$\text{list}\blacktriangleright\text{mat}(\{1,2,3,4,5\},2) \quad \begin{array}{|c|c|} \hline 1 & 2 \\ \hline 3 & 4 \\ \hline 5 & 0 \\ \hline \end{array}$$

ln

Catálogo >

Expr ▶ln ⇒ expresiónCausa la entrada *Expr* a convertirse en una expresión que contiene sólo logaritmos naturales (ln).**Nota:** Usted puede insertar este operador desde el teclado de la computadora al escribir **▶ln**.

$$\left(\log_{10}(x)\right)\blacktriangleright\ln \quad \frac{\ln(x)}{\ln(10)}$$

ln()ctrl e^x teclas**ln(Expr1)** ⇒ expresión**ln(Lista)** ⇒ lista

Entrega el logaritmo natural del argumento.

Para una lista, entrega los logaritmos naturales de los elementos.

ln(matrizCuadrada1) ⇒ matrizCuadradaEntrega el logaritmo natural de la matriz de *matrizCuadrada1*. Esto no es lo mismo que calcular el logaritmo natural de cada elemento. Para obtener información acerca del método de cálculo, consulte **cos()** en.*matrizCuadrada1* debe ser diagonalizable. El resultado siempre contiene números de punto flotante.

$$\ln(2.) \quad 0.693147$$

Si el modo de formato complejo es Real:

$$\ln\{-3, 1.2, 5\}$$

"Error: Non-real calculation"

Si el modo de formato complejo es Rectangular:

$$\ln\{-3, 1.2, 5\} \quad \{\ln(3)+\pi \cdot i, 0.182322, \ln(5)\}$$

En el modo de ángulo en Radianes y el formato complejo Rectangular:

$$\ln \begin{pmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{pmatrix}$$

$$\begin{bmatrix} 1.83145+1.73485 \cdot i & 0.009193-1.49086 \\ 0.448761-0.725533 \cdot i & 1.06491+0.623491 \cdot i \\ -0.266891-2.08316 \cdot i & 1.12436+1.79018 \cdot i \end{bmatrix}$$

Para ver el resultado completo, presione \blacktriangle y luego use \blacktriangleleft y \blacktriangleright para mover el cursor.**LnReg**

Catálogo >

LnReg X, Y1, [Frec] I, Categoría, Incluir]]Resuelve la regresión logarítmica $y = a+b \cdot \ln(x)$ en las listas X y Y con frecuencia Frec. Un resumen de resultados se almacena en la variable *stat.results*. (Vea la página 122.)Todas las listas deben tener una dimensión igual, excepto por *Incluir*.

X y Y son listas de variables independientes y dependientes.

Frec es una lista opcional de valores de frecuencia. Cada elemento en *Frec* especifica la frecuencia de la ocurrencia para cada punto de datos X y Y correspondientes. El valor predeterminado es 1. Todos los elementos deben ser enteros ≥ 0 .*Categoría* es una lista de códigos de categoría para los datos X y Y correspondientes.*Incluir* es una lista de uno o más códigos de categoría. Sólo aquellos elementos de datos cuyo código de categoría está incluido en esta lista están incluidos en el cálculo.

Para obtener información sobre el efecto de los elementos vacíos en una lista, vea "Elementos vacíos (inválidos)" en la página 166.

Variable de salida	Descripción
stat.EcnReg	Ecuación de regresión: $a+b \cdot \ln(x)$
stat.a, stat.b	Coefficientes de regresión
stat.r ²	Coefficiente de determinación lineal para datos transformados
stat.r	Coefficiente de correlación para datos transformados ($\ln(x)$, y)

Variable de salida	Descripción
stat.Resid	Residuales asociados con el modelo logarítmico
stat.TransResid	Residuales asociadas con el ajuste lineal de datos transformados
stat.XReg	La lista de puntos de datos en <i>Lista X</i> modificada se usa de hecho en la regresión con base en las restricciones de las Categorías <i>Frec</i> , <i>Lista de Categorías Incluir</i>
stat.YReg	La lista de puntos de datos en <i>Lista Y</i> modificada se usa de hecho en la regresión con base en las restricciones de las Categorías <i>Frec</i> , <i>Lista de Categorías Incluir</i>
stat.FrecReg	Lista de frecuencias correspondientes a <i>stat.XReg</i> y <i>stat.YReg</i>

Local

Catálogo >

Local *Var1*, *Var2* [, *Var3*] ...

Declara las *vars* especificadas como variables locales. Esas variables existen sólo durante la evaluación de una función y se borran cuando la función termina la ejecución.

Nota: Las variables locales ahorran memoria porque sólo existen en forma temporal. Asimismo, no alteran ninguno de los valores de variable global existentes. Las variables locales se deben usar para los bucles y para guardar temporalmente los valores en una función de líneas múltiples, ya que las modificaciones en las variables globales no están permitidas en una función.

Nota para ingresar el ejemplo: En la aplicación de Calculadora en el dispositivo portátil, usted puede ingresar definiciones en líneas múltiples al presionar en lugar de al final de cada línea. En el teclado de la computadora, presione y sostenga **Alt** y presione **Ingresar**.

```
Define rollcount()=Func
    Local i
    1 → i
    Loop
    If randInt(1,6)=randInt(1,6)
    Goto end
    i+1 → i
    EndLoop
    Lbl end
    Return i
EndFunc
```

Done

<i>rollcount</i> ()	16
<i>rollcount</i> ()	3

Lock (Bloquear)

Catálogo >

Lock *Var1*, *Var2* [, *Var3*] ...

Lock *Var*.

Bloquea las variables o el grupo de variables especificado. Las variables bloqueadas no se pueden modificar ni borrar.

Usted no puede bloquear o desbloquear la variable de sistema *Ans*, y no puede bloquear los grupos de variables de sistema *stat*. o *tvm*.

Nota: El comando **Lock** limpia el historial de Deshacer/Rehacer cuando se aplica a variables no bloqueadas.

Vea **unLock**, página 137y **getLockInfo()**, página 55.

<i>a</i> :=65	65
Lock <i>a</i>	Done
getLockInfo(<i>a</i>)	1
<i>a</i> :=75	"Error: Variable is locked."
DelVar <i>a</i>	"Error: Variable is locked."
Unlock <i>a</i>	Done
<i>a</i> :=75	75
DelVar <i>a</i>	Done

log()ctrl 10^x teclas**log**(*Expr1* [, *Expr2*]) \Rightarrow expresión**log**(*Lista1* [, *Expr2*]) \Rightarrow listaEntrega el logaritmo *Expr2* base del primer argumento.**Nota:** Vea también **Plantilla de logaritmos**, página 2.Para una lista, entrega el logaritmo *Expr2* base de los elementos.

Si el segundo argumento se omite, se usa 10 como la base.

$$\log_{10} (2.) \quad 0.30103$$

$$\log_4 (2.) \quad 0.5$$

$$\log_3 (10) - \log_3 (5) \quad \log_3 (2)$$

Si el modo de formato complejo es Real:

$$\log_{10} (\{-3, 1.2, 5\}) \quad \text{Non-real result}$$

Si el modo de formato complejo es Rectangular:

$$\log_{10} (\{-3, 1.2, 5\})$$

$$\left\{ \log_{10} (3) + 1.36438 \cdot i, 0.079181, \log_{10} (5) \right\}$$

En el modo de ángulo en Radianes y el formato complejo Rectangular:

$$\log_{10} \begin{bmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 0.795387 + 0.753438 \cdot i & 0.003993 - 0.6474 \cdot i \\ 0.194895 - 0.315095 \cdot i & 0.462485 + 0.2707 \cdot i \\ -0.115909 - 0.904706 \cdot i & 0.488304 + 0.7774 \cdot i \end{bmatrix}$$

Para ver el resultado completo, presione \blacktriangle y luego use \blacktriangleleft y \blacktriangleright para mover el cursor.**log**(*matrizCuadrada1* [, *Expr*]) \Rightarrow *matrizCuadrada*Entrega el logaritmo *Expr* base de la matriz de *matrizCuadrada1*. Esto no es lo mismo que calcular el logaritmo *Expr* base de cada elemento. Para obtener información acerca del método de cálculo, consulte **cos()**.*matrizCuadrada1* debe ser diagonalizable. El resultado siempre contiene números de punto flotante.

Si el argumento base se omite, se usa 10 como la base.

logbase

Catálogo >

logbase(*Expr1*) \Rightarrow expresiónCausa la Expresión de entrada a simplificarse a una expresión utilizando la base *Expr1*.**Nota:** Usted puede insertar este operador desde el teclado de la computadora al escribir $\text{>logbase}(\dots)$.

$$\log_3 (10) - \log_5 (5) \blacktriangleright \text{logbase}(5)$$

$$\frac{\left(\log_5 (3) - \log_5 (2) - 1 \right)}{\log_5 (3)}$$

Logistic $X, Y, [Frec] [, Categoría, Incluir]$

Resuelve la regresión logística $y = (c/(1+a \cdot e^{bx})+d)$ en las listas X y Y con frecuencia $Frec$. Un resumen de resultados se almacena en la variable *stat.results*. (Vea la página 122.)

Todas las listas deben tener una dimensión igual, excepto por *Incluir*.

X y Y son listas de variables independientes y dependientes.

$Frec$ es una lista opcional de valores de frecuencia. Cada elemento en $Frec$ especifica la frecuencia de la ocurrencia para cada punto de datos X y Y correspondientes. El valor predeterminado es 1. Todos los elementos deben ser enteros ≥ 0 .

$Categoría$ es una lista de códigos de categoría para los datos X y Y correspondientes.

$Incluir$ es una lista de uno o más códigos de categoría. Sólo aquellos elementos de datos cuyo código de categoría está incluido en esta lista están incluidos en el cálculo.

Para obtener información sobre el efecto de los elementos vacíos en una lista, vea "Elementos vacíos (inválidos)" en la página 166.

Variable de salida	Descripción
stat.EcnReg	Ecuación de regresión: $c/(1+a \cdot e^{bx}+d)$
stat.a, stat.b, stat.c	Coefficientes de regresión
stat.Resid	Residuales de la regresión
stat.XReg	La lista de puntos de datos en la <i>Lista X</i> modificada que se usa en realidad en la regresión con base en las restricciones de las Categorías <i>Frec</i> , <i>Lista de Categorías Incluir</i>
stat.YReg	La lista de puntos de datos en la <i>Lista Y</i> modificada que se usa en realidad en la regresión con base en las restricciones de las Categorías <i>Frec</i> , <i>Lista de Categorías Incluir</i>
stat.FrecReg	Lista de frecuencias correspondientes a <i>stat.XReg</i> y <i>stat.YReg</i>

LogisticD $X, Y [, [Iteraciones], [Frec] [, Categoría, Incluir]]$

Resuelve la regresión logística $y = c/(1 + a \cdot e^{bx})$ en las listas X y Y con frecuencia $Frec$, utilizando un número específico de $Iteraciones$. Un resumen de resultados se almacena en la variable $stat.results$. (Vea la página 122.)

Todas las listas deben tener una dimensión igual, excepto por *Incluir*.

X y Y son listas de variables independientes y dependientes.

$Frec$ es una lista opcional de valores de frecuencia. Cada elemento en $Frec$ especifica la frecuencia de la ocurrencia para cada punto de datos X y Y correspondientes. El valor predeterminado es 1. Todos los elementos deben ser enteros ≥ 0 .

$Categoría$ es una lista de códigos de categoría para los datos X y Y correspondientes.

Incluir es una lista de uno o más códigos de categoría. Sólo aquellos elementos de datos cuyo código de categoría está incluido en esta lista están incluidos en el cálculo.

Para obtener información sobre el efecto de los elementos vacíos en una lista, vea "Elementos vacíos (inválidos)" en la página 166.

Variable de salida	Descripción
stat.EcnReg	Ecuación de regresión: $c/(1 + a \cdot e^{bx})$
stat.a, stat.b, stat.c, stat.d	Coefficientes de regresión
stat.Resid	Residuales de la regresión
stat.XReg	La lista de puntos de datos en la <i>Lista X</i> modificada que se usa en realidad en la regresión con base en las restricciones de las Categorías <i>Frec</i> , <i>Lista de Categorías Incluir</i>
stat.YReg	La lista de puntos de datos en la <i>Lista Y</i> modificada que se usa en realidad en la regresión con base en las restricciones de las Categorías <i>Frec</i> , <i>Lista de Categorías Incluir</i>
stat.FrecReg	Lista de frecuencias correspondientes a <i>stat.XReg</i> y <i>stat.YReg</i>

Loop

Bloque

EndLoop

Ejecuta en forma repetida las sentencias en el *Bloque*. Tome en cuenta que el bucle se ejecutará sin parar, a menos que se ejecute una instrucción **Goto** o **Exit** dentro del *Bloque*.

Bloque es una secuencia de sentencias separadas con el caracter ":".

Nota para ingresar el ejemplo: En la aplicación de Calculadora en el dispositivo portátil, usted puede ingresar definiciones en líneas múltiples al presionar en lugar de al final de cada línea. En el teclado de la computadora, presione y sostenga **Alt** y presione **Ingresar**.

Define $rollcount(i)=Func$

Local i

$i \rightarrow i$

Loop

If $randInt(1,6)=randInt(1,6)$

Goto end

$i+1 \rightarrow i$

EndLoop

Lbl end

Return i

EndFunc

Done

$rollcount(i)$	16
$rollcount(i)$	3

LU (BA)

LU *Matriz*, *matrizB*, *matrizA*, *matrizP*[, Tol]

Calcula la descomposición BA (baja-alta) de Doolittle de una matriz real o compleja. La matriz triangular baja se almacena en *matrizB*, la matriz triangular alta en *matrizA* y la matriz de permutación (que describe los cambios de fila realizados durante el cálculo) en *matrizP*.

$$matrizB \cdot matrizA = matrizP \cdot matriz$$

De manera opcional, cualquier elemento de matriz se trata como cero si su valor absoluto es menor que la *Tolerancia*. Esta tolerancia se usa sólo si la matriz tiene ingresos de punto flotante y no contiene ninguna variable simbólica a la que no se le haya asignado un valor. De otro modo, la *Tolerancia* se ignora.

- Si usted usa o configura el modo **Auto o Aproximado** para aproximar, los cálculos se realizan al usar la aritmética de punto flotante.
- Si la *Tolerancia* se omite o no se usa, la tolerancia predeterminada se calcula como:
 $5E-14 \cdot \max(\dim(Matriz)) \cdot \text{normaFila}(Matriz)$

El algoritmo de factorización **LU** usa un pivoteo parcial con intercambios de filas.

$\begin{bmatrix} 6 & 12 & 18 \\ 5 & 14 & 31 \\ 3 & 8 & 18 \end{bmatrix} \rightarrow m1$	$\begin{bmatrix} 6 & 12 & 18 \\ 5 & 14 & 31 \\ 3 & 8 & 18 \end{bmatrix}$
---	--

LU $m1, lower, upper, perm$ Done

<i>lower</i>	$\begin{bmatrix} 1 & 0 & 0 \\ \frac{5}{6} & 1 & 0 \\ \frac{1}{2} & \frac{1}{2} & 1 \end{bmatrix}$
--------------	---

<i>upper</i>	$\begin{bmatrix} 6 & 12 & 18 \\ 0 & 4 & 16 \\ 0 & 0 & 1 \end{bmatrix}$
--------------	--

<i>perm</i>	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$
-------------	---

$\begin{bmatrix} m & n \\ o & p \end{bmatrix} \rightarrow m1$	$\begin{bmatrix} m & n \\ o & p \end{bmatrix}$
---	--

LU $m1, lower, upper, perm$ Done

<i>lower</i>	$\begin{bmatrix} 1 & 0 \\ \frac{m}{o} & 1 \\ o & \end{bmatrix}$
--------------	---

<i>upper</i>	$\begin{bmatrix} o & p \\ 0 & n - \frac{m \cdot p}{o} \end{bmatrix}$
--------------	--

<i>perm</i>	$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$
-------------	--

M

mat►list()

Catálogo > 

mat►list(*Matriz*) ⇒ *lista*

Entrega una lista completada con los elementos de *Matriz*. Los elementos se copian desde *Matriz* fila por fila.

Nota: Usted puede insertar esta función desde el teclado de la computadora al escribir **mat@>list**(...).

$$\begin{array}{l} \text{mat►list}([1 \ 2 \ 3]) \qquad \qquad \qquad \{1,2,3\} \\ \begin{array}{|c|c|c|} \hline 1 & 2 & 3 \\ \hline 4 & 5 & 6 \\ \hline \end{array} \rightarrow m1 \qquad \qquad \begin{array}{|c|c|c|} \hline 1 & 2 & 3 \\ \hline 4 & 5 & 6 \\ \hline \end{array} \\ \text{mat►list}(m1) \qquad \qquad \qquad \{1,2,3,4,5,6\} \end{array}$$

max()

Catálogo > 

max(*Expr1*, *Expr2*) ⇒ *expresión*

max(*Lista1*, *Lista2*) ⇒ *lista*

max(*Matriz1*, *Matriz2*) ⇒ *matriz*

Entrega el máximo de los dos argumentos. Si los argumentos son dos listas de matrices, entrega una lista de matriz que contiene el valor máximo de cada par de elementos correspondientes.

max(*Lista*) ⇒ *expresión*

Entrega el elemento máximo en *lista*.

max(*Matriz1*) ⇒ *matriz*

Entrega un vector de fila que contiene el elemento máximo de cada columna en *Matriz1*.

Los elementos vacíos (anulados) se ignoran. Para obtener más información sobre elementos vacíos, vea la página 166.

Nota: Vea también **fMax()** y **min()**.

$$\begin{array}{l} \text{max}(2.3, 1.4) \qquad \qquad \qquad 2.3 \\ \text{max}(\{1,2\}, \{-4,3\}) \qquad \qquad \{1,3\} \end{array}$$

$$\text{max}(\{0,1,-7,1.3,0.5\}) \qquad \qquad 1.3$$

$$\text{max}\left(\begin{array}{|c|c|c|} \hline 1 & -3 & 7 \\ \hline -4 & 0 & 0.3 \\ \hline \end{array}\right) \qquad \qquad [1 \ 0 \ 7]$$

mean() (media)

Catálogo > 

mean(*Lista*, *listaFrec*) ⇒ *expresión*

Entrega la media de los elementos en *Lista*.

Cada elemento de *listaFrec* cuenta el número de ocurrencias consecutivas del elemento correspondiente en *Lista*.

mean(*Matriz1*, *matrizFrec*) ⇒ *matriz*

Entrega un vector de fila de las medias de todas las columnas en *Matriz1*.

Cada elemento de *matrizFrec* cuenta el número de ocurrencias consecutivas del elemento correspondiente en *Matriz1*.

Los elementos vacíos (anulados) se ignoran. Para obtener más información sobre elementos vacíos, vea la página 166.

$$\text{mean}(\{0.2, 0, 1, -0.3, 0.4\}) \qquad \qquad 0.26$$

$$\text{mean}(\{1,2,3\}, \{3,2,1\}) \qquad \qquad \frac{5}{3}$$

En formato de vector Rectangular:

$$\text{mean}\left(\begin{array}{|c|c|} \hline 0.2 & 0 \\ \hline -1 & 3 \\ \hline 0.4 & -0.5 \\ \hline \end{array}\right) \qquad \qquad [-.133333 \ 0.833333]$$

$$\text{mean}\left(\begin{array}{|c|c|} \hline \frac{1}{5} & 0 \\ \hline -1 & 3 \\ \hline \frac{2}{5} & \frac{-1}{2} \\ \hline \end{array}\right) \qquad \qquad \begin{array}{|c|c|} \hline \frac{-2}{15} & \frac{5}{6} \\ \hline \end{array}$$

$$\text{mean}\left(\begin{array}{|c|c|} \hline 1 & 2 \\ \hline 3 & 4 \\ \hline 5 & 6 \\ \hline \end{array}, \begin{array}{|c|c|} \hline 5 & 3 \\ \hline 4 & 1 \\ \hline 6 & 2 \\ \hline \end{array}\right) \qquad \qquad \begin{array}{|c|c|} \hline \frac{47}{15} & \frac{11}{3} \\ \hline \end{array}$$

median() (mediana)Catálogo > **median(Lista₁, listaFrec)** ⇒ expresiónEntrega la mediana de los elementos en *Lista*.Cada elemento de *listaFrec* cuenta el número de ocurrencias consecutivas del elemento correspondiente en *Lista*.

$$\text{median}(\{0.2, 0, 1, -0.3, 0.4\}) \quad 0.2$$

median(Matriz1_i, matrizFrec) ⇒ matrizEntrega un vector de fila que contiene las medianas de las columnas en *Matriz1*.Cada elemento de *matrizFrec* cuenta el número de ocurrencias consecutivas del elemento correspondiente en *Matriz1*.

$$\text{median} \left(\begin{bmatrix} 0.2 & 0 \\ 1 & -0.3 \\ 0.4 & -0.5 \end{bmatrix} \right) \quad [0.4 \quad -0.3]$$

Notas:

- Todos los ingresos en la lista o matriz se deben simplificar a números.
- Los elementos vacíos (inválidos) en la lista o matriz se ignoran. Para obtener más información sobre elementos vacíos, vea la página 166.

MedMedCatálogo > **MedMed X₁Y₁, Frec** [, Categoría, Incluir]Genera la línea media-mediana $y = (m \cdot x + b)$ en las listas *X* y *Y* con frecuencia *Frec*. Un resumen de resultados se almacena en la variable *stat.results*. (Vea la página 122.)Todas las listas deben tener una dimensión igual, excepto por *Incluir*.*X* y *Y* son listas de variables independientes y dependientes.*Frec* es una lista opcional de valores de frecuencia. Cada elemento en *Frec* especifica la frecuencia de la ocurrencia para cada punto de datos *X* y *Y* correspondientes. El valor predeterminado es 1. Todos los elementos deben ser enteros ≥ 0 .*Categoría* es una lista de códigos de categoría para los datos *X* y *Y* correspondientes.*Incluir* es una lista de uno o más códigos de categoría. Sólo aquellos elementos de datos cuyo código de categoría está incluido en esta lista están incluidos en el cálculo.

Para obtener información sobre el efecto de los elementos vacíos en una lista, vea "Elementos vacíos (inválidos)" en la página 166.

Variable de salida	Descripción
stat.EcnReg	Ecuación de la recta mediana-mediana: $m \cdot x + b$
stat.m, stat.b	Coefficientes del modelo
stat.Resid	Residuales desde la recta mediana-mediana
stat.XReg	La lista de puntos de datos en la <i>Lista X</i> modificada que se usa en realidad en la regresión con base en las restricciones de las Categorías <i>Frec</i> , <i>Lista de Categorías Incluir</i>
stat.YReg	La lista de puntos de datos en la <i>Lista Y</i> modificada que se usa en realidad en la regresión con base en las restricciones de las Categorías <i>Frec</i> , <i>Lista de Categorías Incluir</i>
stat.FrecReg	Lista de frecuencias correspondientes a <i>stat.XReg</i> y <i>stat.YReg</i>

mid()Catálogo > **mid**(cadenaFuente, Iniciar[, Contar]) ⇒ cadena

Entrega caracteres de *Conteo* de la cadena de caracteres *cadenaFuente*, comenzando con el número de caracteres *Iniciar*.

Si se omite *Conteo* o es mayor que la dimensión de *cadenaFuente*, entrega todos los caracteres de *cadenaFuente*, comenzando con el número de caracteres *Iniciar*.

El *Conteo* debe ser ≥ 0 . Si *Conteo* = 0, entrega una cadena vacía.

mid(listaFuente, Iniciar [, Conteo]) ⇒ lista

Entrega elementos de *Conteo* de *listaFuente*, comenzando con el número de elementos del *Inicio*.

Si se omite *Conteo* o es mayor que la dimensión de *listaFuente*, entrega todos los elementos de *listaFuente*, comenzando con el número de elementos del *Inicio*.

El *Conteo* debe ser ≥ 0 . Si *Conteo* = 0, entrega una lista vacía.

mid(listaCadenaFuente, Iniciar[, Conteo]) ⇒ lista

Entrega cadenas de *Conteo* de la lista de cadenas *listaCadenaFuente*, comenzando con el número de elementos del *Inicio*.

$\text{mid}(\text{"Hello there"}, 2)$	"ello there"
$\text{mid}(\text{"Hello there"}, 7, 3)$	"the"
$\text{mid}(\text{"Hello there"}, 1, 5)$	"Hello"
$\text{mid}(\text{"Hello there"}, 1, 0)$	"{"

$\text{mid}(\{9, 8, 7, 6\}, 3)$	{7, 6}
$\text{mid}(\{9, 8, 7, 6\}, 2, 2)$	{8, 7}
$\text{mid}(\{9, 8, 7, 6\}, 1, 2)$	{9, 8}
$\text{mid}(\{9, 8, 7, 6\}, 1, 0)$	{{"}}

$\text{mid}(\{"A", "B", "C", "D"\}, 2, 2)$	{"B", "C"}
--	------------

mín()Catálogo > **mín**(Expr1, Expr2) ⇒ expresión**mín**(Lista1, Lista2) ⇒ lista**mín**(Matriz1, Matriz2) ⇒ matriz

Entrega el mínimo de los dos argumentos. Si los argumentos son dos listas o matrices, entrega una lista o matriz que contiene el valor mínimo de cada par de elementos correspondientes.

mín(Lista) ⇒ expresión

Entrega el elemento mínimo de *Lista*.

mín(Matriz1) ⇒ matriz

Entrega un vector de fila que contiene el elemento mínimo de cada columna en *Matriz1*.

Nota: Vea también **fMín()** y **max()**.

$\text{mín}(2.3, 1.4)$	1.4
$\text{mín}(\{1, 2\}, \{-4, 3\})$	{-4, 2}

$\text{mín}(\{0, 1, -7, 1.3, 0.5\})$	-7
--------------------------------------	----

$\text{mín}\left(\begin{bmatrix} 1 & -3 & 7 \\ -4 & 0 & 0.3 \end{bmatrix}\right)$	[-4 -3 0.3]
---	-------------

mirr()Catálogo > **mirr**(tasaFinanciación, tasaReinversión, FE0, listaFE[, frecFE])

La función financiera que entrega la tasa interna de rendimiento modificada de una inversión.

tasaFinanciación es la tasa de interés que usted paga sobre las cantidades de flujo de efectivo.

tasaReinversión es la tasa de interés a la que se reinvierten los flujos de efectivo.

FE0 es el flujo de efectivo inicial en tiempo 0; debe ser un número real.

ListaFE es una lista de cantidades de flujo de efectivo después del flujo de efectivo inicial *FE0*.

FrecFE es una lista opcional en la cual cada elemento especifica la frecuencia de ocurrencia para una cantidad de flujo de efectivo (consecutivo) agrupado, que es el elemento correspondiente de la *ListaFE*. La predeterminada es 1; si usted ingresa valores, éstos deben ser enteros positivos < 10,000.

Nota: Vea también **irr()**, página 62.

$\text{list1} = \{6000, -8000, 2000, -3000\}$	$\{6000, -8000, 2000, -3000\}$
$\text{list2} = \{2, 2, 2, 1\}$	$\{2, 2, 2, 1\}$
$\text{mirr}(4.65, 12, 5000, \text{list1}, \text{list2})$	13.41608607

mod()

Catálogo >

mod(*Expr1*, *Expr2*) ⇒ *expresión***mod**(*Lista1*, *Lista2*) ⇒ *lista***mod**(*Matriz1*, *Matriz2*) ⇒ *matriz*

Entrega el segundo argumento del módulo del primer argumento conforme se define por medio de las identidades:

$$\text{mod}(x, 0) = x$$

$$\text{mod}(x, y) = x - y \text{ piso}(x/y)$$

Cuando el segundo argumento no es cero, el resultado es periódico en ese argumento. El resultado es cero o tiene el mismo signo que el segundo argumento.

Si los argumentos son dos listas o dos matrices, entrega una lista o matriz que contiene el módulo de cada par de elementos correspondientes.

Nota: Vea también **remain()**, página . 102

$\text{mod}(7,0)$	7
$\text{mod}(7,3)$	1
$\text{mod}(-7,3)$	2
$\text{mod}(7,-3)$	-2
$\text{mod}(-7,-3)$	-1
$\text{mod}(\{12, -14, 16\}, \{9, 7, -5\})$	$\{3, 0, -4\}$

mRow() (filaM)

Catálogo >

mRow(*Expr*, *Matriz1*, *Índice*) ⇒ *matriz*

Entrega una copia de *Matriz1* con cada elemento en la fila *Índice* de *Matriz1* multiplicado por *Expr*.

$\text{mRow}\left(\frac{-1}{3}, \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, 2\right)$	$\begin{bmatrix} 1 & 2 \\ -1 & -4 \\ 3 & 3 \end{bmatrix}$
---	---

mRowAdd() (agrFilaM)

Catálogo >

mRowAdd(*Expr*, *Matriz1*, *Índice1*, *Índice2*) ⇒ *matriz*

Entrega una copia de *Matriz1* con cada elemento en la fila *Índice2* de *Matriz1* reemplazado por:

$$\text{Expr} \cdot \text{fila } \text{Índice1} + \text{fila } \text{Índice2}$$

$\text{mRowAdd}\left(-3, \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, 1, 2\right)$	$\begin{bmatrix} 1 & 2 \\ 0 & -2 \\ 3 & 4 \end{bmatrix}$
$\text{mRowAdd}\left(n, \begin{bmatrix} a & b \\ c & d \end{bmatrix}, 1, 2\right)$	$\begin{bmatrix} a & b \\ a \cdot n + c & b \cdot n + d \\ c & d \end{bmatrix}$

MultReg

Catálogo >

MultReg *Y*, *X1*[, *X2*[, *X3*, ..., [, *X10*]]]

Calcula la regresión lineal múltiple de la lista *Y* en las listas *X1*, *X2*, ..., *X10*. Un resumen de resultados se almacena en la variable *resultados.estad*. (Vea la página 122.)

Todas las listas deben tener una dimensión igual.

Para obtener información sobre el efecto de los elementos vacíos en una lista, vea "Elementos vacíos (inválidos)" en la página 166.

Variable de salida	Descripción
stat.EcnReg	Ecuación de regresión: $b_0 + b_1 \cdot x_1 + b_2 \cdot x_2 + \dots$
stat.b0, stat.b1, ...	Coefficientes de regresión
stat.R ²	Coefficiente de determinación múltiple
stat.ŷLista	\hat{y} Lista = $b_0 + b_1 \cdot x_1 + \dots$
stat.Resid	Residuales de la regresión

MultRegIntervals $Y, X1[,X2[,X3,...[,X10]]], listaValX[, nivelC]$

Computa un valor y previsto, un intervalo de predicción de nivel C para una observación sencilla, así como un intervalo de confianza de nivel C para la respuesta media.

Un resumen de resultados se almacena en la variable *stat.results*.
(Vea la página 122.)

Todas las listas deben tener una dimensión igual.

Para obtener información sobre el efecto de los elementos vacíos en una lista, vea “Elementos vacíos (inválidos)” en la página 166.

Variable de salida	Descripción
stat.EcnReg	Ecuación de regresión: $b_0 + b_1 \cdot x_1 + b_2 \cdot x_2 + \dots$
stat. \hat{y}	Un estimado de punto: $\hat{y} = b_0 + b_1 \cdot x_1 + \dots$ para <i>listaValX</i>
stat.dfError	Grados de libertad de error
stat.CBajo, stat.CAlto	Intervalo de confianza para una respuesta media
stat.ME	Margen de error del intervalo de confianza
stat.EE	Error estándar de respuesta media
stat.PredBaja, stat.PredAlta	Intervalo de predicción para una observación sencilla
stat.MEPred	Margen de error del intervalo de predicción
stat.EEPred	Error estándar para la predicción
stat.ListaB	Lista de coeficientes de regresión, $\{b_0, b_1, b_2, \dots\}$
stat.Resid	Residuales de la regresión

MultRegTests (PruebasRegMult)**MultRegTests** $Y, X1[,X2[,X3,...[,X10]]]$

La prueba de regresión lineal múltiple resuelve una regresión lineal múltiple sobre los datos dados y proporciona la estadística de la prueba F global y las estadísticas de la prueba t para los coeficientes.

Un resumen de resultados se almacena en la variable *stat.results*.
(Vea la página 122.)

Para obtener información sobre el efecto de los elementos vacíos en una lista, vea “Elementos vacíos (inválidos)” en la página 166.

Salidas

Variable de salida	Descripción
stat.EcnReg	Ecuación de regresión: $b_0 + b_1 \cdot x_1 + b_2 \cdot x_2 + \dots$
stat.F	Estadística de la prueba F global
stat.ValP	Valor P asociado con la estadística de F global

Variable de salida	Descripción
stat.R ²	Coficiente de determinación múltiple
stat.AjustR ²	Coficiente de determinación múltiple ajustado
stat.s	Desviación estándar del error
stat.DW	Estadística de Durbin-Watson; se usa para determinar si la autocorrelación de primer grado está presente en el modelo
stat.dffReg	Grados de libertad de la regresión
stat.SCReg	Suma de cuadrados de la regresión
stat.CMReg	Cuadrado medio de la regresión
stat.dffError	Grados de libertad de error
stat.SSError	Suma de cuadrados del error
stat.CMError	Cuadrado medio del error
stat.ListaB	{b ₀ ,b ₁ ,...} Lista de coeficientes
stat.ListaT	Lista de estadísticas t, una para cada coeficiente en la ListaB
stat.ListaP	Valores P de la lista para cada estadística t
stat.ListaEE	Lista de errores estándar para los coeficientes en la ListaB
stat.ŷLista	ŷLista = b ₀ +b ₁ •x ₁ + . . .
stat.Resid	Residuales de la regresión
stat.ResidE	Residuales estandarizados; se obtienen al dividir un residual entre su desviación estándar
stat.DistCook	Distancia de Cook; medida de la influencia de una observación con base en el residual y el apalancamiento
stat.Apalancamiento	Medida de cuán lejos están los valores de la variable independiente de sus valores medios

N

nand

teclas  

BooleanExpr1 **nand** *BooleanExpr2* devuelve expresión

booleana

BooleanList1 **nand** *BooleanList2* devuelve lista booleana

BooleanMatrix1 **nand** *BooleanMatrix2* devuelve matriz booleana

$x \geq 3$ and $x \geq 4$

$x \geq 4$

$x \geq 3$ nand $x \geq 4$

$x < 4$

Devuelve la negación de una operación **and** lógica en los dos argumentos. Devuelve verdadero, falso o una forma simplificada de la ecuación.

Para listas y matrices, devuelve comparaciones elemento por elemento.

nandteclas  *Entero1 nand Entero2* \Rightarrow entero

Compara dos números reales enteros bit a bit utilizando una operación **nand**. Internamente, ambos números enteros se convierten en números binarios de 64 bit con signos. Cuando se comparan bits correspondientes, el resultado es 1 si ambos bits son 1; de lo contrario el resultado es 0. El valor devuelto representa los resultados bit, y se muestran según el modelo Base.

Puede ingresar los números enteros en cualquier base numérica. Para una entrada binaria o hexadecimal, debe utilizar el prefijo 0b o 0h respectivamente. Sin un prefijo, se trata a los números enteros como decimales (base 10).

3 and 4	0
3 nand 4	-1
$\{1,2,3\}$ and $\{3,2,1\}$	$\{1,2,1\}$
$\{1,2,3\}$ nand $\{3,2,1\}$	$\{-2,-3,-2\}$

nCr()Catálogo >  **nCr**(*Expr1*, *Expr2*) \Rightarrow expresión

Para entero *Expr1* y *Expr2* con $Expr1 \geq Expr2 \geq 0$, **nCr()** es el número de combinaciones de los elementos de *Expr1* tomadas de *Expr2* a la vez. (Esto también se conoce como un coeficiente binomial). Ambos argumentos pueden ser enteros o expresiones simbólicas.

nCr(*Expr*, 0) \Rightarrow 1**nCr**(*Expr*, enteroNeg) \Rightarrow 0**nCr**(*Expr*, enteroPos) $\Rightarrow Expr \cdot (Expr-1) \dots$ $(Expr - \text{enteroPos} + 1) \text{ enteroPos}!$ **nCr**(*Expr*, noEntero) \Rightarrow expresión! $((Expr - \text{noEntero})! \cdot \text{noEntero}!)$ **nCr**(*Lista1*, *Lista2*) \Rightarrow lista

Entrega una lista de combinaciones con base en los pares de elementos correspondientes en las dos listas. Los argumentos deben tener el mismo tamaño que la lista.

nCr(*Matriz1*, *Matriz2*) \Rightarrow matriz

Entrega una matriz de combinaciones con base en los pares de elementos correspondientes en las dos matrices. Los argumentos deben tener el mismo tamaño que la matriz.

$nCr(z,3)$	$\frac{z \cdot (z-2) \cdot (z-1)}{6}$
Ans z=5	10
$nCr(z,c)$	$\frac{z!}{c! \cdot (z-c)!}$
Ans	$\frac{1}{c!}$
$nPr(z,c)$	$c!$

$nCr(\{5,4,3\},\{2,4,2\})$	$\{10,1,3\}$
----------------------------	--------------

$nCr\left(\begin{bmatrix} 6 & 5 \\ 4 & 3 \end{bmatrix}, \begin{bmatrix} 2 & 2 \\ 2 & 2 \end{bmatrix}\right)$	$\begin{bmatrix} 15 & 10 \\ 6 & 3 \end{bmatrix}$
--	--

nDerivative()Catálogo >  **nDerivative**(*Expr1*, *Var=Valor*[, *Orden*]) \Rightarrow valor**nDerivative**(*Expr1*, *Var*[, *Orden*]) | *Var=Valor* \Rightarrow valor

Entrega la derivada numérica calculada con el uso de métodos de autodiferenciación.

Cuando se especifica el *Valor*, se eliminan todas las asignaciones anteriores de la variable o cualquier sustitución "|" para la variable.

El *Orden* de la derivada debe ser 1 ó 2.

$nDerivative(x ,x=1)$	1
$nDerivative(x ,x) x=0$	undef
$nDerivative(\sqrt{x-1},x) x=1$	undef

newList() (nuevaLista)Catálogo >  **newList**(*elementosNum*) \Rightarrow lista

Entrega una lista con una dimensión de *elementosNum*. Cada elemento es cero.

$newList(4)$	$\{0,0,0,0\}$
--------------	---------------

newMat()Catálogo > **newMat**(filasNum, columnasNum) ⇒ matrizEntrega una matriz de ceros con la dimensión *filasNum* por *columnasNum*.

$\text{newMat}(2,3)$	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$
----------------------	--

nfMax()Catálogo > **nfMax**(Expr, Var) ⇒ valor**nfMax**(Expr, Var, limiteInferior) ⇒ valor**nfMax**(Expr, Var, limiteInferior, limiteSuperior) ⇒ valor**nfMax**(Expr, Var) | $\text{limiteInferior} \leq \text{Var} \leq \text{limiteSuperior}$ ⇒ valorEntrega un valor numérico candidato de la variable *Var* donde ocurre el local máximo de *Expr*.Si proporciona el *limite inferior* y el *limite superior*, la función buscará en el intervalo cerrado [*limite inferior*, *limite superior*] el valor del máximo local en la función.**Nota:** Vea también **fMax()** y **d()**.

$\text{nfMax}(-x^2-2 \cdot x-1, x)$	-1.
$\text{nfMax}(0.5 \cdot x^3-x-2, x, -5, 5)$	-0.816497

nfMin()Catálogo > **nfMin**(Expr, Var) ⇒ valor**nfMin**(Expr, Var, limiteInferior) ⇒ valor**nfMin**(Expr, Var, limiteInferior, limiteSuperior) ⇒ valor**nfMin**(Expr, Var) | $\text{limiteInferior} \leq \text{Var} \leq \text{limiteSuperior}$ ⇒ valorEntrega un valor numérico candidato de la *Var* donde ocurre el local mínimo de *Expr*.Si proporciona el *limite inferior* y el *limite superior*, la función buscará en el intervalo cerrado [*limite inferior*, *limite superior*] el valor del mínimo local en la función.**Nota:** Vea también **fMin()** y **d()**.

$\text{nfMin}(x^2+2 \cdot x+5, x)$	-1.
$\text{nfMin}(0.5 \cdot x^3-x-2, x, -5, 5)$	0.816497

nInt()Catálogo > **nInt**(Expr1, Var, Inferior, Superior) ⇒ expresión

Si el integrando *Expr1* no contiene ninguna variable que no sea *Var*, y si *Inferior* y *Superior* son constantes, positiva ∞ o negativa ∞ , entonces **nInt()** entrega una aproximación de $\int(\text{Expr1}, \text{Var}, \text{Inferior}, \text{Superior})$. Esta aproximación es un promedio ponderado de algunos valores muestra del integrando en el intervalo $\text{Inferior} < \text{Var} < \text{Superior}$.

La meta es seis dígitos significativos. El logaritmo adaptable termina cuando parece probable que la meta se ha alcanzado, o bien cuando parece improbable que las muestras adicionales producirán una mejora importante.

Se desplegará una advertencia ("Exactitud cuestionable") cuando parece que la meta no se ha alcanzado.

Anide **nInt()** para hacer una integración numérica múltiple. Los límites de la integración pueden depender de las variables de integración afuera de los mismos.

Nota: Vea también **f()**, página 155.

$\text{nInt}(e^{-x^2}, x, -1, 1)$	1.49365
-----------------------------------	---------

$\text{nInt}(\cos(x), x, \pi, \pi+1 \cdot \text{E} \cdot 12)$	-1.04144E-12
$\int_{\pi}^{\pi+10^{-12}} \cos(x) dx$	$-\sin\left(\frac{1}{1000000000000}\right)$

$\text{nInt}\left(\text{nInt}\left(\frac{e^{-x \cdot y}}{\sqrt{x^2-y^2}}, y, -x, x\right), x, 0, 1\right)$	3.30423
--	---------

nom()

Catálogo >

nom(*tasaEfectiva*, *CpA*) \Rightarrow *valor*

Función financiera que convierte la tasa de interés efectiva anual *tasaEfectiva* en una tasa nominal, con *CpA* dado como el número de periodos compuestos por año.

tasaEfectiva debe ser un número real y *CpA* debe ser un número real > 0.

Nota: Vea también **eff()**, página 41.

nom (5.90398,12)	5.75
-------------------------	------

nor

teclas

BooleanoExpr1 **nor** *BooleanoExpr2* devuelve *expresión booleana*

BooleanaLista1 **nor** *BooleanaLista2* devuelve *lista booleana*

BooleanaMatriz1 **nor** *BooleanaMatriz2* devuelve *matriz booleana*

$x \geq 3$ or $x \geq 4$	$x \geq 3$
--------------------------	------------

$x \geq 3$ nor $x \geq 4$	$x < 3$
---------------------------	---------

Devuelve la negación de una operación **or** lógica en los dos argumentos. Devuelve verdadero, falso o una forma simplificada de la ecuación.

Para listas y matrices, devuelve comparaciones elemento por elemento.

Entero1 **nor** *Entero2* \Rightarrow *entero*

Compara dos números reales enteros bit a bit utilizando una operación **nor**. Internamente, ambos números enteros se convierten en números binarios de 64 bit y con signos. Cuando se comparan bits correspondientes, el resultado es 1 si ambos bits son 1; de lo contrario el resultado es 0. El valor devuelto representa los resultados bit, y se muestran según el modelo Base.

Puede ingresar los números enteros en cualquier base numérica. Para una entrada binaria o hexadecimal, debe utilizar el prefijo 0b o 0h respectivamente. Sin un prefijo, se trata a los números enteros como decimales (base 10).

3 or 4	7
--------	---

3 nor 4	-8
---------	----

{1,2,3} or {3,2,1}	{3,2,3}
--------------------	---------

{1,2,3} nor {3,2,1}	{-4,-3,-4}
---------------------	------------

norm()

Catálogo >

norm(*Matriz*) \Rightarrow *expresión*

norm(*Vector*) \Rightarrow *expresión*

Entrega la norma Frobenius.

$\text{norm}\left(\begin{bmatrix} a & b \\ c & d \end{bmatrix}\right)$	$\sqrt{a^2+b^2+c^2+d^2}$
--	--------------------------

$\text{norm}\left(\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}\right)$	$\sqrt{30}$
--	-------------

$\text{norm}\left(\begin{bmatrix} 1 & 2 \end{bmatrix}\right)$	$\sqrt{5}$
---	------------

$\text{norm}\left(\begin{bmatrix} 1 \\ 2 \end{bmatrix}\right)$	$\sqrt{5}$
--	------------

nPr() (prN)

Catálogo >

nPr(*Expr1*, *Expr2*) ⇒ expresión

Para entero *Expr1* y *Expr2* con $Expr1 \geq Expr2 \geq 0$, **nPr()** es el número de permutaciones de los elementos de *Expr1* tomadas de *Expr2* a la vez. Ambos argumentos pueden ser enteros o expresiones simbólicas.

nPr(*Expr*, 0) ⇒ 1**nPr**(*Expr*, enteroNeg) ⇒ $1/((Expr+1) \cdot (Expr+2) \dots)$
(expresión-enteroNeg)**nPr**(*Expr*, enteroPos) ⇒ $Expr \cdot (Expr-1) \dots$
(*Expr*-enteroPos+1)**nPr**(*Expr*, noEntero) ⇒ $Expr! / (Expr-noEntero)!$ **nPr**(*Lista1*, *Lista2*) ⇒ lista

Entrega una lista de permutaciones con base en los pares de elementos correspondientes en las dos listas. Los argumentos deben tener el mismo tamaño que la lista.

nPr(*Matriz1*, *Matriz2*) ⇒ matriz

Entrega una matriz de permutaciones con base en los pares de elementos correspondientes en las dos matrices. Los argumentos deben tener el mismo tamaño que la matriz.

$$\frac{nPr(z,3)}{Ans|z=5} = \frac{z \cdot (z-2) \cdot (z-1)}{60}$$

$$\frac{nPr(z,-3)}{1} = \frac{1}{(z+1) \cdot (z+2) \cdot (z+3)}$$

$$\frac{nPr(z,c)}{z!} = \frac{z!}{(z-c)!}$$

$$\frac{Ans \cdot nPr(z-c,c)}{1}$$

$$\frac{nPr(\{5,4,3\}, \{2,4,2\})}{\{20,24,6\}}$$

$$\frac{nPr\left(\begin{bmatrix} 6 & 5 \\ 4 & 3 \end{bmatrix}, \begin{bmatrix} 2 & 2 \\ 2 & 2 \end{bmatrix}\right)}{\begin{bmatrix} 30 & 20 \\ 12 & 6 \end{bmatrix}}$$

npv() (vpn)

Catálogo >

npv(*TasaInterés*, *FEO*, *ListaFE*, *FrecFE*)

Función financiera que calcula el valor presente neto; la suma de los valores presentes para las entradas y salidas de efectivo. Un resultado positivo para el vpv indica una inversión rentable.

tasaInterés es la tasa por la que se descuentan los flujos de efectivo (el costo del dinero) durante un período.

FEO es el flujo de efectivo inicial en tiempo 0; debe ser un número real.

ListaFE es una lista de cantidades de flujo de efectivo después del flujo de efectivo inicial *FEO*.

FrecFE es una lista en la cual cada elemento especifica la frecuencia de ocurrencia para una cantidad de flujo de efectivo (consecutivo) agrupado, que es el elemento correspondiente de la *ListaFE*. La predeterminada es 1; si usted ingresa valores, éstos deben ser enteros positivos < 10,000.

$$\frac{list1 := \{6000, -8000, 2000, -3000\}}{\{6000, -8000, 2000, -3000\}}$$

$$\frac{list2 := \{2, 2, 2, 1\}}{\{2, 2, 2, 1\}}$$

$$\frac{npv(10, 5000, list1, list2)}{4769.91}$$

nSolve() (solucionN)

Catálogo >

nSolve(*Ecuación*, *Var*[=*Cálculo*]) ⇒ número de error_cadena**nSolve**(*Ecuación*, *Var*[=*Cálculo*], límiteInferior)

⇒ número de error_cadena

nSolve(*Ecuación*, *Var*[=*Cálculo*], límiteInferior, límiteSuperior)

⇒ número de error_cadena

nSolve(*Ecuación*, *Var*[=*Cálculo*]) |límiteInferior ≤ *Var* ≤ límiteSuperior

⇒ número de error_cadena

Busca iterativamente una solución numérica real aproximada para *Ecuación* para su variable uno. Especifique la variable como:

variable

- 0 -

variable = número real

Por ejemplo, x es válida y también lo es x=3.

$$\frac{nSolve(x^2 + 5 \cdot x - 25 = 9, x)}{3.84429}$$

$$\frac{nSolve(x^2 = 4, x = -1)}{-2}$$

$$\frac{nSolve(x^2 = 4, x = 1)}{2}$$

Nota: Si hay varias soluciones, usted puede usar un cálculo para ayudar a encontrar una solución particular.

nSolve() (solucionN)Catálogo > 

nSolve() con frecuencia es mucho más rápido que **solve()** o **zeros()**, en particular si el operador "|" se usa para restringir la búsqueda a un intervalo pequeño que contiene exactamente una solución sencilla.

nSolve() intenta determinar un punto donde la residual es cero o dos puntos relativamente cercanos donde la residual tiene signos opuestos y la magnitud de la residual no es excesiva. Si no puede lograr esto al usar un número modesto de puntos de muestra, entrega la cadena "ninguna solución encontrada".

Nota: Vea también **cSolve()**, **cZeros()**, **solve()**, y **zeros()**.

$$\text{nSolve}(x^2+5\cdot x-25=9,x)|x<0 \quad -8.84429$$

$$\text{nSolve}\left(\frac{(1+r)^{24}-1}{r}=26,r\right)|r>0 \text{ and } r<0.25$$

0.006886

$$\text{nSolve}(x^2=-1,x) \quad \text{"No solution found"}$$

O**OneVar**Catálogo > 

OneVar [1,]X[,][Frec][,Categoría,Incluir]

OneVar [n,]X1,X2[X3[...[,X20]]]

Calcula estadísticas de 1 variable en hasta 20 listas. Un resumen de resultados se almacena en la variable *stat.results*. (Vea la página 122.)

Todas las listas deben tener una dimensión igual, excepto por *Incluir*.

Frec es una lista opcional de valores de frecuencia. Cada elemento en *Frec* especifica la frecuencia de la ocurrencia para cada punto de datos *X* y *Y* correspondientes. El valor predeterminado es 1. Todos los elementos deben ser enteros ≥ 0 .

Categoría es una lista de códigos de categoría numérica para los valores *X* correspondientes.

Incluir es una lista de uno o más códigos de categoría. Sólo aquellos elementos de datos cuyo código de categoría está incluido en esta lista están incluidos en el cálculo.

Un elemento (inválido) vacío en cualquiera de las listas *X*, *Frec* o *Categoría* da como resultado un inválido para el elemento correspondiente de todas esas listas. Un elemento vacío en cualquiera de las listas *X1* a *X20* da como resultado vacío para el elemento correspondiente de todas esas listas. Para obtener más información sobre elementos vacíos, vea la página 166.

Variable de salida	Descripción
stat. \bar{X}	Media de valores x
stat. Σx	Suma de valores x
stat. Σx^2	Suma de valores x^2
stat.ex	Desviación estándar muestra de x
stat. σx	Desviación estándar de población de x
stat.n	Número de puntos de datos
stat.MinX	Mínimo de valores x
stat.C ₇ X	1er Cuartil de x
stat.MedianaX	Mediana de x

Variable de salida	Descripción
stat.C3X	3er Cuartil de x
stat.MaxX	Máximo de valores x
stat.SCX	Suma de cuadrados de desviaciones de la media de x

or

Catálogo > 

BooleanaExpr1 **or** *BooleanaExpr2* devuelve expresión booleana
BooleanaLista1 **or** *BooleanaLista2* devuelve lista booleana
BooleanaMatriz1 **or** *BooleanaMatriz2* devuelve matriz booleana

Entrega verdadero o falso o una forma simplificada del ingreso original.

Entrega verdadero si cualquiera de las expresiones o ambas se simplifican a verdadero. Entrega falso si ambas expresiones se evalúan a falso.

Nota: Vea **xor**.

Nota para ingresar el ejemplo: En la aplicación de Calculadora en el dispositivo portátil, usted puede ingresar definiciones en líneas múltiples al presionar  en lugar de  al final de cada línea. En el teclado de la computadora, presione y sostenga **Alt** y presione **Ingresar**.

Entero1 **or** *Entero2* ⇒ *entero*

Compara dos enteros reales bit por bit usando una or operación. En forma interna, ambos enteros se convierten en números binarios de 64 bits firmados. Cuando se comparan los bits correspondientes, el resultado es 1 si cualquiera de los bits es 1; el resultado es 0 sólo si ambos bits son 0. El valor producido representa los resultados de los bits, y se despliega de acuerdo con el modo de Base.

Se pueden ingresar enteros en cualquier base de números. Para un ingreso binario o hexadecimal, se debe usar el prefijo 0b or 0h, respectivamente. Sin un prefijo, los enteros se tratan como decimales (base 10).

Si se ingresa un entero decimal que es demasiado grande para una forma binaria de 64 bits firmada, se usa una operación de módulo simétrico para llevar el valor al rango apropiado. Para obtener más información, vea **Base2**, página 14.

Nota: Vea **xor**.

$x \geq 3$ or $x \geq 4$ $x \geq 3$

```
Define g(x)=Func Done
If x<=0 or x>=5
Goto end
Return x*3
Lbl end
EndFunc
```

$g(3)$ 9

$g(0)$ A function did not return a value

En modo de base hexadecimal:

0h7AC36 or 0h3D5F 0h7BD7F

Importante: Cero, no la letra O.

En modo de base binaria:

0b100101 or 0b100 0b100101

Nota: Un ingreso binario puede tener hasta 64 dígitos (sin contar el prefijo 0b). Un ingreso hexadecimal puede tener hasta 16 dígitos.

ord()

Catálogo > 

ord(*Cadena*) ⇒ *entero*

ord(*Lista1*) ⇒ *lista*

Entrega el código numérico del primer carácter en la cadena de caracteres *Cadena*, o bien una lista de los primeros caracteres de cada elemento de la lista.

ord("hello") 104

char(104) "h"

ord(**char**(24)) 24

ord({"alpha", "beta"}) {97,98}

P

P►Rx()

Catálogo >

P►Rx(*rExpr*, *θExpr*) ⇒ expresión

P►Rx(*rLista*, *θLista*) ⇒ lista

P►Rx(*rMatriz*, *θMatriz*) ⇒ matriz

Entrega la coordenada x equivalente del par (r, θ).

Nota: El argumento θ se interpreta como un ángulo en grados, gradianes o radianes, de acuerdo con el modo de ángulo actual. Si el argumento es una expresión, usted puede usar $\frac{\circ}{3}$ o $\frac{r}{\text{r}}$ para anular la configuración del modo de ángulo en forma temporal.

Nota: Usted puede insertar esta función desde el teclado de la computadora al escribir **P@>Rx** (...).

En modo de ángulo en Radianes:

$$\text{P►Rx}(r, \theta) \quad \cos(\theta) \cdot r$$

$$\text{P►Rx}(4, 60^\circ) \quad 2$$

$$\text{P►Rx}\left\{\{-3, 10, 1.3\}, \left\{\frac{\pi}{3}, \frac{\pi}{4}, 0\right\}\right\} \quad \left\{\frac{-3}{2}, 5\sqrt{2}, 1.3\right\}$$

P►Ry()

Catálogo >

P►Ry(*rExpr*, *θExpr*) ⇒ expresión

P►Ry(*rLista*, *θLista*) ⇒ lista

P►Ry(*rMatriz*, *θMatriz*) ⇒ matriz

Entrega la coordenada y equivalente del par (r, θ).

Nota: El argumento θ se interpreta como un ángulo en grados, radianes o gradianes, de acuerdo con el modo de ángulo actual. Si el argumento es una expresión, usted puede usar $\frac{\circ}{3}$ o $\frac{r}{\text{r}}$ para anular la configuración del modo de ángulo en forma temporal.

Nota: Usted puede insertar esta función desde el teclado de la computadora al escribir **P@>Ry** (...).

En modo de ángulo en Radianes:

$$\text{P►Ry}(r, \theta) \quad \sin(\theta) \cdot r$$

$$\text{P►Ry}(4, 60^\circ) \quad 2\sqrt{3}$$

$$\text{P►Ry}\left\{\{-3, 10, 1.3\}, \left\{\frac{\pi}{3}, \frac{\pi}{4}, 0\right\}\right\} \quad \left\{\frac{-3\sqrt{3}}{2}, -5\sqrt{2}, 0\right\}$$

PassErr (PasarErr)

Catálogo >

PassErr

Pasa un error al siguiente nivel.

Si la variable de sistema *códigoErr* es cero, **PassErr** no hace nada.

La cláusula **Else** del bloque **Try...Else...EndTry** debe usar **ClrErr** o **PassErr**. Si el error se debe procesar o ignorar, use **ClrErr**. Si no se sabe qué hacer con el error, use **PassErr** para enviarlo al siguiente manipulador de errores. Si no hay ningún otro manipulador de errores **Try...Else...EndTry** pendiente, el cuadro de diálogo de error se desplegará como normal.

Nota: Ver también **BorrarErr**, página 19 e **intento**, page 132.

Nota para ingresar el ejemplo: En la aplicación calculadora del dispositivo portátil, puede ingresar definiciones de múltiples líneas presionando en vez de al final de cada línea. En el teclado de la computadora, apriete **Alt** y presione **Enter**.

Para ver un ejemplo de **PassErr**, vea el Ejemplo 2 bajo el comando **Intentar**, página 132.

piecewise() (compuestoDeVariables)

Catálogo >

piecewise(*Expr1* [, *Cond1* [, *Expr2* [, *Cond2* [, ...]]])

Entrega definiciones para una función de compuesto de variables en la forma de una lista. Usted también puede crear definiciones de compuesto de variables al usar una plantilla.

Nota: Vea también **Plantilla de compuesto de variables**, página 2.

$$\text{Define } p(x) = \begin{cases} x, & x > 0 \\ \text{undef}, & x \leq 0 \end{cases} \quad \text{Done}$$

$$p(1) \quad 1$$

$$p(-1) \quad \text{undef}$$

poissCdf()

Catálogo >

poissCdf(λ , *límiteInferior*, *límiteSuperior*) \Rightarrow número si *límiteInferior* y *límiteSuperior* son números, lista si *límiteInferior* y *límiteSuperior* son listas

poissCdf(λ , *límiteSuperior*) para $P(0 \leq X \leq \textit{límiteSuperior}) \Rightarrow$ número si *límiteSuperior* es un número, lista si *límiteSuperior* es una lista

Resuelve una probabilidad acumulativa para la distribución de Poisson discreta con una media especificada λ .

Para $P(X \leq \textit{límiteSuperior})$, configure *límiteInferior*=0

poissPdf()

Catálogo >

poissPdf(λ , *ValX*) \Rightarrow número si *ValX* es un número, lista si *ValX* es una lista

Resuelve una probabilidad para la distribución de Poisson discreta con la media especificada λ .

Polar

Catálogo >

Vector ► **Polar**

Nota: Usted puede insertar este operador desde el teclado de la computadora al escribir @>Polar.

Despliega el *vector* en forma polar [$r \angle \theta$]. El vector debe ser de dimensión 2 y puede ser una fila o una columna.

Nota: ► **Polar** es una instrucción de formato de despliegue, no una función de conversión. Usted puede usarla sólo al final de una línea de ingreso, y no actualiza *ans*.

Nota: Vea también ► **Rect**, página 100.

valorComplejo ► **Polar**

Despliega el *valorComplejo* en forma polar.

- El modo de ángulo en grados entrega ($r \angle \theta$).
- El modo de ángulo en radianes entrega $re^{i\theta}$.

valorComplejo puede tener cualquier forma compleja. Sin embargo, un ingreso de $re^{i\theta}$ causa un error en el modo de ángulo en grados.

Nota: Usted debe usar los paréntesis para un ingreso polar ($r \angle \theta$).

$[1 \ 3.] \blacktriangleright \text{Polar}$	$[3.16228 \ \angle \ 1.24905]$
$[x \ y] \blacktriangleright \text{Polar}$	$\left[\sqrt{x^2+y^2} \ \angle \ \frac{\pi \cdot \text{sign}(y)}{2} - \tan^{-1}\left(\frac{x}{y}\right) \right]$

En modo de ángulo en Radianes:

$(3+4 \cdot i) \blacktriangleright \text{Polar}$	$e^{i \cdot \left(\frac{\pi}{2} - \tan^{-1}\left(\frac{3}{4}\right) \right)} \cdot 5$
$\left(\left(4 \ \frac{\pi}{3} \right) \right) \blacktriangleright \text{Polar}$	$e^{\frac{i \cdot \pi}{3}} \cdot 4$

En modo de ángulo en Gradianes:

$(4 \cdot i) \blacktriangleright \text{Polar}$	$(4 \ \angle \ 100)$
--	----------------------

En modo de ángulo en Grados:

$(3+4 \cdot i) \blacktriangleright \text{Polar}$	$\left(5 \ \angle \ 90 - \tan^{-1}\left(\frac{3}{4}\right) \right)$
--	--

polyCoeffs()Catálogo > **polyCoeffs**(*Poli* [, *Var*]) ⇒ *lista*

Entrega una lista de los coeficientes del polinomio *Poli* con respecto de la variable *Var*.

Poli debe ser una expresión polinómica en *Var*. Recomendamos que usted no omita *Var* a menos que *Poli* sea una expresión en una variable sencilla.

$$\text{polyCoeffs}(4 \cdot x^2 - 3 \cdot x + 2, x) \quad \{4, -3, 2\}$$

$$\text{polyCoeffs}((x-1)^2 \cdot (x+2)^3) \quad \{1, 4, 1, -10, -4, 8\}$$

Expande el polinomio y selecciona *x* para la *Var*omitida.

$$\text{polyCoeffs}((x+y+z)^2, x) \quad \{1, 2 \cdot (y+z), (y+z)^2\}$$

$$\text{polyCoeffs}((x+y+z)^2, y) \quad \{1, 2 \cdot (x+z), (x+z)^2\}$$

$$\text{polyCoeffs}((x+y+z)^2, z) \quad \{1, 2 \cdot (x+y), (x+y)^2\}$$

polyDegree() (gradoPoli)Catálogo > **polyDegree**(*Poli* [, *Var*]) ⇒ *valor*

Entrega el grado de la expresión polinómica *Poli* con respecto de la variable *Var*. Si usted omite *Var*, la función **polyDegree()** selecciona una predeterminada de las variables contenidas en el polinomio *Poli*.

Poli debe ser una expresión polinómica en *Var*. Recomendamos que usted no omita *Var* a menos que *Poli* sea una expresión en una variable sencilla.

$$\text{polyDegree}(5) \quad 0$$

$$\text{polyDegree}(\ln(2) + \pi, x) \quad 0$$

Polinomios constantes

$$\text{polyDegree}(4 \cdot x^2 - 3 \cdot x + 2, x) \quad 2$$

$$\text{polyDegree}((x-1)^2 \cdot (x+2)^3) \quad 5$$

$$\text{polyDegree}((x+y^2+z^3)^2, x) \quad 2$$

$$\text{polyDegree}((x+y^2+z^3)^2, y) \quad 4$$

$$\text{polyDegree}((x-1)^{10000}, x) \quad 10000$$

El grado se puede extraer a pesar de que en los coeficientes no se puede. Esto es porque el grado se puede extraer sin expandir el polinomio.

polyEval() (evalPoli)Catálogo > **polyEval**(*Lista1*, *Expr1*) ⇒ *expresión***polyEval**(*Lista1*, *Lista2*) ⇒ *expresión*

Interpreta el primer argumento como el coeficiente de un polinomio de grado descendente, y entrega el polinomio evaluado para el valor del segundo argumento.

$$\text{polyEval}(\{a, b, c\}, x) \quad a \cdot x^2 + b \cdot x + c$$

$$\text{polyEval}(\{1, 2, 3, 4\}, 2) \quad 26$$

$$\text{polyEval}(\{1, 2, 3, 4\}, \{2, 7\}) \quad \{26, -262\}$$

polyGcd()Catálogo > **polyGcd**(*Expr1, Expr2*) \Rightarrow expresión

Entrega el máximo común divisor de los dos argumentos.

Expr1 y *Expr2* deben ser expresiones polinómicas.

No se permite lista, matriz ni argumentos Booleanos

$\text{polyGcd}(100,30)$	10
$\text{polyGcd}(x^2-1,x-1)$	$x-1$
$\text{polyGcd}(x^3-6 \cdot x^2+11 \cdot x-6, x^2-6 \cdot x+8)$	$x-2$

polyQuotient() (cocientePoli)Catálogo > **polyQuotient**(*Poli1, Poli2* [, *Var*]) \Rightarrow expresiónEntrega el cociente del polinomio *Poli1* dividido entre el polinomio *Poli2* con respecto de la variable *Var* especificada.*Poli1* y *Poli2* deben ser expresiones polinómicas en *Var*.Recomendamos que usted no omita *Var* a menos que *Poli1* y *Poli2* sean expresiones en la misma variable sencilla.

$\text{polyQuotient}(x-1,x-3)$	1
$\text{polyQuotient}(x-1,x^2-1)$	0
$\text{polyQuotient}(x^2-1,x-1)$	$x+1$
$\text{polyQuotient}(x^3-6 \cdot x^2+11 \cdot x-6, x^2-6 \cdot x+8)$	x
$\text{polyQuotient}((x-y) \cdot (y-z), x+y+z, x)$	$y-z$
$\text{polyQuotient}((x-y) \cdot (y-z), x+y+z, y)$	$2 \cdot x - y + 2 \cdot z$
$\text{polyQuotient}((x-y) \cdot (y-z), x+y+z, z)$	$-(x-y)$

polyRemainder() (restoPoli)Catálogo > **polyRemainder**(*Poli1, Poli2* [, *Var*]) \Rightarrow expresiónEntrega el resto del polinomio *Poli1* dividido entre el polinomio *Poli2* con respecto de la variable *Var* especificada.*Poli1* y *Poli2* deben ser expresiones polinómicas en *Var*.Recomendamos que usted no omita *Var* a menos que *Poli1* y *Poli2* sean expresiones en la misma variable sencilla.

$\text{polyRemainder}(x-1,x-3)$	2
$\text{polyRemainder}(x-1,x^2-1)$	$x-1$
$\text{polyRemainder}(x^2-1,x-1)$	0
$\text{polyRemainder}((x-y) \cdot (y-z), x+y+z, x)$	$-(y-z) \cdot (2 \cdot y + z)$
$\text{polyRemainder}((x-y) \cdot (y-z), x+y+z, y)$	$-2 \cdot x^2 - 5 \cdot x \cdot z - 2 \cdot z^2$
$\text{polyRemainder}((x-y) \cdot (y-z), x+y+z, z)$	$(x-y) \cdot (x+2 \cdot y)$

polyRoots() (raícesPoli)

Catálogo >

polyRoots(*Poli*, *Var*) ⇒ lista**polyRoots**(*ListaDeCoefs*) ⇒ lista

La primera sintaxis, **polyRoots**(*Poli*, *Var*), entrega una lista de raíces reales del polinomio *Poli* con respecto de la variable *Var*. Si no existe ninguna raíz real, entrega una lista vacía: {}.

Poli debe ser un polinomio en una variable.

La segunda sintaxis, **polyRoots**(*ListaDeCoefs*), entrega una lista de raíces reales para los coeficientes en *ListaDeCoefs*.

Nota: Vea también **cPolyRoots()**, página 27.

$$\begin{array}{l} \text{polyRoots}(y^3+1,y) \qquad \{-1\} \\ \hline \text{cPolyRoots}(y^3+1,y) \\ \left\{-1, \frac{1}{2} - \frac{\sqrt{3}}{2}i, \frac{1}{2} + \frac{\sqrt{3}}{2}i\right\} \\ \hline \text{polyRoots}(x^2+2x+1,x) \qquad \{-1,-1\} \\ \hline \text{polyRoots}(\{1,2,1\}) \qquad \{-1,-1\} \end{array}$$

PowerReg (RegPot)

Catálogo >

PowerReg *X*, *Y* [, *Frec*] [, *Categoría*, *Incluir*]

Resuelve la regresión de potencia $y = a \cdot (x)^{ab}$ en listas *X* y *Y* con frecuencia *Frec*. Un resumen de resultados se almacena en la variable *stat.results*. (Vea la página 122.)

Todas las listas deben tener una dimensión igual, excepto por *Incluir*.

X y *Y* son listas de variables independientes y dependientes.

Frec es una lista opcional de valores de frecuencia. Cada elemento en *Frec* especifica la frecuencia de la ocurrencia para cada punto de datos *X* y *Y* correspondientes. El valor predeterminado es 1. Todos los elementos deben ser enteros ≥ 0 .

Categoría es una lista de códigos de categoría para los datos *X* y *Y* correspondientes.

Incluir es una lista de uno o más códigos de categoría. Sólo aquellos elementos de datos cuyo código de categoría está incluido en esta lista están incluidos en el cálculo.

Para obtener información sobre el efecto de los elementos vacíos en una lista, vea "Elementos vacíos (inválidos)" en la página 166.

Variable de salida	Descripción
stat.EcnReg	Ecuación de regresión: $a \cdot (x)^{ab}$
stat.a, stat.b	Coefficientes de regresión
stat.r ²	Coefficiente de determinación lineal para datos transformados
stat.r	Coefficiente de correlación para datos transformados (ln(x), ln(y))
stat.Resid	Residuales asociados con el modelo de potencia
stat.TransResid	Residuales asociadas con el ajuste lineal de datos transformados
stat.XReg	La lista de puntos de datos en la <i>Lista X</i> modificada que se usa en realidad en la regresión con base en las restricciones de las Categorías <i>Frec</i> , <i>Lista de Categorías</i> e <i>Incluir</i>
stat.YReg	La lista de puntos de datos en la <i>Lista Y</i> modificada que se usa en realidad en la regresión con base en las restricciones de las Categorías <i>Frec</i> , <i>Lista de Categorías</i> e <i>Incluir</i>
stat.FrecReg	Lista de frecuencias correspondientes a <i>stat.XReg</i> y <i>stat.YReg</i>

Prgm*Bloque***EndPrgm**

Plantilla para crear un programa definido por el usuario. Se debe usar con el comando **Define**, **Define LibPub**, o **Define LibPriv**.

Bloque puede ser una sentencia sencilla, una serie de sentencias separadas con el caracter ":" o una serie de sentencias en líneas separadas.

Nota para ingresar el ejemplo: En la aplicación de Calculadora en el dispositivo portátil, usted puede ingresar definiciones en líneas múltiples al presionar  en lugar de  al final de cada línea. En el teclado de la computadora, presione y sostenga **Alt** y presione **Ingresar**.

Calcular MCD y desplegar los resultados intermedios.

Define $\text{proggcd}(a,b)=\text{Prgm}$

```
Local d
While b≠0
d:=mod(a,b)
a:=b
b:=d
Disp a, " ", b
EndWhile
Disp "GCD=", a
EndPrgm
```

Done

$\text{proggcd}(4560,450)$

450 60

60 30

30 0

GCD=30

Done

prodSeq()

Vea $\Pi()$, página 157.

Product (PI)

Vea $\Pi()$, página 157.

product()

Catálogo > 

product(*Lista*, *Iniciar*, *Terminar*) \Rightarrow *expresión*

Entrega el producto de los elementos contenidos en *Lista*. *Inicio* y *Término* son opcionales. Especifican un rango de elementos.

$\text{product}(\{1,2,3,4\})$ 24

$\text{product}(\{2,x,y\})$ $2 \cdot x \cdot y$

$\text{product}(\{4,5,8,9\},2,3)$ 40

product(*Matriz**1*, *Iniciar*, *Terminar*) \Rightarrow *matriz*

Entrega un vector de fila que contiene los productos de los elementos en las columnas de *Matriz**1*. *Inicio* y *término* son opcionales. Especifican un rango de filas.

Los elementos vacíos (inválidos) se ignoran. Para obtener más información sobre elementos vacíos, vea la página 166.

$\text{product}\left(\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}\right)$ [28 80 162]

$\text{product}\left(\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix},1,2\right)$ [4 10 18]

propFrac()

Catálogo >

propFrac(Expr1[, Var]) \Rightarrow expresión

propFrac(número_racional) entrega número_racional como la suma de un entero y una fracción que tiene el mismo signo y una magnitud de denominador mayor que la magnitud del numerador.

propFrac(expresión_racional, Var) entrega la suma de las proporciones apropiadas y un polinomio con respecto de Var. El grado de Var en el denominador excede el grado de Var en el numerador en cada proporción apropiada. Se recopilan potencias similares de Var. Los términos y sus factores se ordenan con Var como la variable principal.

Si se omite Var, se realiza una expansión de la fracción apropiada con respecto de la variable más principal. Entonces los coeficientes de la parte polinómica se tornan apropiados con respecto de su variable más principal primero y así sucesivamente.

Para expresiones racionales, **propFrac()** es una alternativa más rápida aunque menos extrema para **expand()**.

Usted puede usar la función **propFrac()** para representar fracciones mezcladas y demostrar la suma y la resta de fracciones mezcladas.

$$\text{propFrac}\left(\frac{4}{3}\right) \quad 1 + \frac{1}{3}$$

$$\text{propFrac}\left(\frac{-4}{3}\right) \quad -1 - \frac{1}{3}$$

$$\text{propFrac}\left(\frac{x^2+x+1}{x+1} + \frac{y^2+y+1}{y+1}, x\right) \quad \frac{1}{x+1} + x + \frac{y^2+y+1}{y+1}$$

$$\text{propFrac}(\text{Ans}) \quad \frac{1}{x+1} + x + \frac{1}{y+1} + y$$

$$\text{propFrac}\left(\frac{11}{7}\right) \quad 1 + \frac{4}{7}$$

$$\text{propFrac}\left(3 + \frac{1}{11} + 5 + \frac{3}{4}\right) \quad 8 + \frac{37}{44}$$

$$\text{propFrac}\left(3 + \frac{1}{11} - \left(5 + \frac{3}{4}\right)\right) \quad -2 - \frac{29}{44}$$

Q**QR**

Catálogo >

QR Matriz, matrizQ, matrizR[, Tol]

Calcula la factorización de QR de Householder de una matriz real o una matriz compleja. Las matrices Q y R resultantes se almacenan en la Matriz especificada. La matriz Q es unitaria. La matriz R es triangular superior.

De manera opcional, cualquier elemento de matriz se trata como cero si su valor absoluto es menor que la Tolerancia. Esta tolerancia se usa sólo si la matriz tiene ingresos de punto flotante y no contiene ninguna variable simbólica a la que no se le haya asignado un valor. De otro modo, la Tolerancia se ignora.

- Si usted usa o configura el modo **Auto o Aproximado** para aproximar, los cálculos se realizan al usar la aritmética de punto flotante.
- Si la Tolerancia se omite o no se usa, la tolerancia predeterminada se calcula como:
 $5E-14 \cdot \max(\dim(\text{Matriz})) \cdot \text{normaFila}(\text{Matriz})$

El número de punto flotante (9.) en m1 causa que los resultados se calculen en forma de punto flotante.

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9. \end{bmatrix} \rightarrow m1 \quad \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9. \end{bmatrix}$$

QR m1, qm, rm Done

qm	0.123091	0.904534	0.408248
	0.492366	0.301511	-0.816497
	0.86164	-0.301511	0.408248

rm	8.12404	9.60114	11.0782
	0.	0.904534	1.80907
	0.	0.	0.

ClearAZ Done

La factorización de QR se resuelve numéricamente al usar transformaciones de Householder. La solución simbólica se resuelve al usar Gram-Schmidt. Las columnas en *nombreMatQ* son los vectores de base ortonormal que extienden el espacio definido por la *matriz*.

$$\begin{bmatrix} m & n \\ o & p \end{bmatrix} \rightarrow m1 \quad \begin{bmatrix} m & n \\ o & p \end{bmatrix}$$

QR *m1,qm,rm* Done

$$qm = \begin{bmatrix} \frac{m}{\sqrt{m^2+o^2}} & \frac{-\text{sign}(m \cdot p - n \cdot o) \cdot o}{\sqrt{m^2+o^2}} \\ \frac{o}{\sqrt{m^2+o^2}} & \frac{m \cdot \text{sign}(m \cdot p - n \cdot o)}{\sqrt{m^2+o^2}} \end{bmatrix}$$

$$rm = \begin{bmatrix} \sqrt{m^2+o^2} & \frac{m \cdot n + o \cdot p}{\sqrt{m^2+o^2}} \\ 0 & \frac{m \cdot p - n \cdot o}{\sqrt{m^2+o^2}} \end{bmatrix}$$

QuadReg (RegCua)

QuadReg *X,Y [, Frec] [, Categoría, Incluir]*

Resuelve la regresión polinómica cuadrática $y = a \cdot x^2 + b \cdot x + c$ en las listas *X* y *Y* con frecuencia *Frec*. Un resumen de resultados se almacena en la variable *stat.results*. (Vea la página 122.)

Todas las listas deben tener una dimensión igual, excepto por *Incluir*.

X y *Y* son listas de variables independientes y dependientes.

Frec es una lista opcional de valores de frecuencia. Cada elemento en *Frec* especifica la frecuencia de la ocurrencia para cada punto de datos *X* y *Y* correspondientes. El valor predeterminado es 1. Todos los elementos deben ser enteros ≥ 0 .

Categoría es una lista de códigos de categoría para los datos *X* y *Y* correspondientes.

Incluir es una lista de uno o más códigos de categoría. Sólo aquellos elementos de datos cuyo código de categoría está incluido en esta lista están incluidos en el cálculo.

Para obtener información sobre el efecto de los elementos vacíos en una lista, vea "Elementos vacíos (inválidos)" en la página 166.

Variable de salida	Descripción
stat.EcnReg	Ecuación de regresión: $a \cdot x^2 + b \cdot x + c$
stat.a, stat.b, stat.c	Coefficientes de regresión
stat.R ²	Coefficiente de determinación
stat.Resid	Residuales de la regresión
stat.XReg	La lista de puntos de datos en la <i>Lista X</i> modificada que se usa en realidad en la regresión con base en las restricciones de las Categorías <i>Frec</i> , <i>Lista de Categorías Incluir</i>
stat.YReg	La lista de puntos de datos en la <i>Lista Y</i> modificada que se usa en realidad en la regresión con base en las restricciones de las Categorías <i>Frec</i> , <i>Lista de Categorías Incluir</i>
stat.FrecReg	Lista de frecuencias correspondientes a <i>stat.XReg</i> y <i>stat.YReg</i>

QuartReg $X, Y [, Frec] [, Categoría, Incluir]$

Resuelve la regresión polinómica cuártica

$y = a \cdot x^4 + b \cdot x^3 + c \cdot x^2 + d \cdot x + e$ en las listas X y Y con frecuencia $Frec$. Un resumen de resultados se almacena en la variable $stat.results$. (Vea la página 122.)

Todas las listas deben tener una dimensión igual, excepto por $Incluir$.

X y Y son listas de variables independientes y dependientes.

$Frec$ es una lista opcional de valores de frecuencia. Cada elemento en $Frec$ especifica la frecuencia de la ocurrencia para cada punto de datos X y Y correspondientes. El valor predeterminado es 1. Todos los elementos deben ser enteros ≥ 0 .

$Categoría$ es una lista de códigos de categoría para los datos X y Y correspondientes.

$Incluir$ es una lista de uno o más códigos de categoría. Sólo aquellos elementos de datos cuyo código de categoría está incluido en esta lista están incluidos en el cálculo.

Para obtener información sobre el efecto de los elementos vacíos en una lista, vea "Elementos vacíos (inválidos)" en la página 166.

Variable de salida	Descripción
stat.EcnReg	Ecuación de regresión: $a \cdot x^4 + b \cdot x^3 + c \cdot x^2 + d \cdot x + e$
stat.a, stat.b, stat.c, stat.d, stat.e	Coefficientes de regresión
stat.R ²	Coefficiente de determinación
stat.Resid	Residuales de la regresión
stat.XReg	La lista de puntos de datos en la <i>Lista X</i> modificada que se usa en realidad en la regresión con base en las restricciones de las Categorías $Frec$, <i>Lista de Categoríaae Incluir</i>
stat.YReg	La lista de puntos de datos en la <i>Lista Y</i> modificada que se usa en realidad en la regresión con base en las restricciones de las Categorías $Frec$, <i>Lista de Categoríaae Incluir</i>
stat.FrecReg	Lista de frecuencias correspondientes a $stat.XReg$ y $stat.YReg$

R

R▶Pθ()

Catálogo >

R▶Pθ (*ExprX*, *ExprY*) ⇒ *expresión*

R▶Pθ (*ListaX*, *ListaY*) ⇒ *lista*

R▶Pθ (*MatrizX*, *MatrizY*) ⇒ *matriz*

Entrega la coordenada θ equivalente de los argumentos del par (x,y) .

Nota: El resultado se entrega como un ángulo en grados, gradianes o radianes, de acuerdo con la configuración del modo del ángulo actual.

Nota: Usted puede insertar esta función desde el teclado de la computadora al escribir **R@>Ptheta** (...).

En modo de ángulo en Grados:

$$\mathbf{R}\blacktriangleright\mathbf{P}\theta(x,y) \quad 90 \cdot \text{sign}(y) - \tan^{-1}\left(\frac{x}{y}\right)$$

En modo de ángulo en Gradianes:

$$\mathbf{R}\blacktriangleright\mathbf{P}\theta(x,y) \quad 100 \cdot \text{sign}(y) - \tan^{-1}\left(\frac{x}{y}\right)$$

En modo de ángulo en Radianes:

$$\mathbf{R}\blacktriangleright\mathbf{P}\theta(3,2) \quad \tan^{-1}\left(\frac{2}{3}\right)$$

$$\mathbf{R}\blacktriangleright\mathbf{P}\theta\left([3 \ -4 \ 2], \left[0 \ \frac{\pi}{4} \ 1.5\right]\right) \\ \left[0 \ \tan^{-1}\left(\frac{16}{\pi}\right) + \frac{\pi}{2} \ 0.643501\right]$$

R▶Pr()

Catálogo >

R▶Pr (*ExprX*, *ExprY*) ⇒ *expresión*

R▶Pr (*ListaX*, *ListaY*) ⇒ *lista*

R▶Pr (*MatrizX*, *MatrizY*) ⇒ *matriz*

Entrega la coordenada r equivalente de los argumentos del par (x,y) .

Nota: Usted puede insertar esta función desde el teclado de la computadora al escribir **R@>Pr** (...).

En modo de ángulo en Radianes:

$$\mathbf{R}\blacktriangleright\mathbf{P}r(3,2) \quad \sqrt{13}$$

$$\mathbf{R}\blacktriangleright\mathbf{P}r(x,y) \quad \sqrt{x^2+y^2}$$

$$\mathbf{R}\blacktriangleright\mathbf{P}r\left([3 \ -4 \ 2], \left[0 \ \frac{\pi}{4} \ 1.5\right]\right) \\ \left[3 \ \frac{\sqrt{\pi^2+256}}{4} \ 2.5\right]$$

▶Rad

Catálogo >

Expr▶**Rad** ⇒ *expresión*

Convierte el argumento en la medida de ángulo en radianes.

Nota: Usted puede insertar este operador desde el teclado de la computadora al escribir **@>Rad**.

En modo de ángulo en Grados:

$$(1.5)\blacktriangleright\mathbf{Rad} \quad (0.02618)r$$

En modo de ángulo en Gradianes:

$$(1.5)\blacktriangleright\mathbf{Rad} \quad (0.023562)r$$

rand() (aleat)

Catálogo >

rand() ⇒ *expresión*

rand(#Pruebas) ⇒ *lista*

rand() entrega un valor aleatorio entre 0 y 1.

rand(#Pruebas) entrega una lista que contiene valores aleatorios de #Pruebas entre 0 y 1.

Configura la semilla del número aleatorio.

$$\mathbf{RandSeed} \ 1147 \quad Done$$

$$\mathbf{rand}(2) \quad \{0.158206, 0.717917\}$$

randBin() (binAleat)

Catálogo >

randBin(n, p) \Rightarrow expresión**randBin**($n, p, \#Pruebas$) \Rightarrow lista**randBin**(n, p) entrega un número real aleatorio a partir de una distribución Binomial especificada.**randBin**($n, p, \#Pruebas$) entrega una lista que contiene números reales aleatorios de $\#Pruebas$ a partir de una distribución Binomial especificada.

randBin (80,.5)	34.
randBin (80,.5,3)	{47.,41.,46.}

randInt() (entAleat)

Catálogo >

randInt(*limiteInferior*,*limiteSuperior*) \Rightarrow expresión**randInt**(*limiteInferior*,*limiteSuperior*, $\#Pruebas$) \Rightarrow lista**randInt**(*limiteInferior*,*limiteSuperior*) entrega un entero aleatorio dentro del rango especificado por los límites del entero del *limiteInferior* y el *limiteSuperior*.**randInt**(*limiteInferior*,*limiteSuperior*, $\#Pruebas$) entrega una lista que contiene enteros aleatorios de $\#Pruebas$ dentro del rango especificado.

randInt (3,10)	7.
randInt (3,10,4)	{8.,9.,4.,4.}

randMat() (matAleat)

Catálogo >

randMat(*filasNum*,*columnasNum*) \Rightarrow matriz

Entrega una matriz de enteros entre -9 y 9 de la dimensión especificada.

Ambos argumentos se deben simplificar a enteros.

RandSeed 1147	Done									
randMat (3,3)	<table border="1"> <tr><td>8</td><td>-3</td><td>6</td></tr> <tr><td>-2</td><td>3</td><td>-6</td></tr> <tr><td>0</td><td>4</td><td>-6</td></tr> </table>	8	-3	6	-2	3	-6	0	4	-6
8	-3	6								
-2	3	-6								
0	4	-6								

Nota: Los valores en esta matriz cambiarán cada vez que usted presione **enter**.**randNorm() (normAleat)**

Catálogo >

randNorm(μ, σ) \Rightarrow expresión**randNorm**($\mu, \sigma, \#Pruebas$) \Rightarrow lista**randNorm**(μ, σ) entrega un número decimal a partir de la distribución normal especificada. Podría ser cualquier número real, pero se concentrará fuertemente en el intervalo $[\mu - 3 \cdot \sigma, \mu + 3 \cdot \sigma]$.**randNorm**($\mu, \sigma, \#Pruebas$) entrega una lista que contiene números decimales de $\#Pruebas$ a partir de la distribución normal especificada.

RandSeed 1147	Done
randNorm (0,1)	0.492541
randNorm (3,4.5)	-3.54356

randPoly() (poliAleat)

Catálogo >

randPoly(*Var*,*Orden*) \Rightarrow expresiónEntrega un polinomio en *Var* del *Orden* específico. Los coeficientes son enteros aleatorios en el rango -9 a 9. El coeficiente principal no será cero.*El Orden* debe ser 0-99.

RandSeed 1147	Done
randPoly ($x, 5$)	$-2 \cdot x^5 + 3 \cdot x^4 - 6 \cdot x^3 + 4 \cdot x - 6$

randSamp() (muestAleat)

Catálogo >

randSamp(Lista,#Pruebas[,noReempl]) \Rightarrow lista

Entrega una lista que contiene una muestra aleatoria de las pruebas #Pruebas a partir de la Lista con una opción para el reemplazo de la muestra (noReempl=0), o ningún reemplazo de la muestra (noReempl=1). El predeterminado es con reemplazo de la muestra.

Define list3={1,2,3,4,5}	Done
Define list4=randSamp(list3,6)	Done
list4	{5.,1.,3.,3.,4.,4.}

RandSeed (SemillaAleat)

Catálogo >

RandSeed Número

Si el Número = 0, configura las semillas al predeterminado de fábrica para el generador de números aleatorios. Si el Número \neq 0, se usa para generar dos semillas, que se almacenan en las variables de sistema semilla1 y semilla2.

RandSeed 1147	Done
rand()	0.158206

real()

Catálogo >

real(Expr1) \Rightarrow expresión

Entrega la parte real del argumento.

Nota: Todas las variables indefinidas se tratan como variables reales. Vea también **imag()**, página 59.

real(2+3 <i>i</i>)	2
real(z)	z
real(x+i*y)	x

real(Lista1) \Rightarrow lista

Entrega las partes reales de todos los elementos.

real({a+i b ,3,i})	{a,3,0}
---------------------------	---------

real(Matriz1) \Rightarrow matriz

Entrega las partes reales de todos los elementos.

real($\begin{pmatrix} a+ib & 3 \\ c & i \end{pmatrix}$)	$\begin{bmatrix} a & 3 \\ c & 0 \end{bmatrix}$
---	--

►Rect

Catálogo >

Vector ►Rect

Nota: Usted puede insertar este operador desde el teclado de la computadora al escribir @>Rect.

Despliega el **Vector** en forma rectangular [x, y, z]. El vector debe ser de dimensión 2 ó 3 y puede ser una fila o una columna.

Nota: ►Rect es una instrucción de formato de despliegue, no una función de conversión. Usted puede usarla sólo al final de una línea de ingreso, y no actualiza ans.

Nota: Vea también ►Polar, página 59.

$\left(\begin{pmatrix} 3 & \angle \frac{\pi}{4} & \angle \frac{\pi}{6} \end{pmatrix} \right) \blacktriangleright \text{Rect}$	$\left[\begin{array}{ccc} 3 \cdot \sqrt{2} & 3 \cdot \sqrt{2} & 3 \cdot \sqrt{3} \\ 4 & 4 & 2 \end{array} \right]$
$a \angle b \angle c$	$[a \cdot \cos(b) \cdot \sin(c) \quad a \cdot \sin(b) \cdot \sin(c) \quad a \cdot \cos(c)]$

►Rect

Catálogo >

valorComplejo ►Rect

Despliega el *valorComplejo* en forma rectangular a+bi. El *valorComplejo* puede tener cualquier forma compleja. Sin embargo, un ingreso de $re^{i\theta}$ causa un error en el modo de ángulo en grados.

Nota: Usted debe usar los paréntesis para un ingreso polar ($r\angle\theta$).

En modo de ángulo en Radianes:

$\left(\frac{\pi}{3}\right)$	$\frac{\pi}{3}$
$\left(4 \cdot e^{\frac{\pi}{3}}\right)$ ►Rect	$4 \cdot e^{\frac{\pi}{3}}$
$\left(4 \angle \frac{\pi}{3}\right)$ ►Rect	$2+2 \cdot \sqrt{3} \cdot i$

En modo de ángulo en Gradianes:

$\left(\left(1 \angle 100\right)\right)$ ►Rect	i
--	-----

En modo de ángulo en Grados:

$\left(\left(4 \angle 60\right)\right)$ ►Rect	$2+2 \cdot \sqrt{3} \cdot i$
---	------------------------------

Nota: Para escribir \angle , selecciónelo desde la lista de símbolos en el Catálogo.

ref()

Catálogo >

ref(MatrizI, Tol) ⇒ matriz

Entrega la forma escalonada por filas de *MatrizI*.

De manera opcional, cualquier elemento de matriz se trata como cero si su valor absoluto es menor que la *Tolerancia*. Esta tolerancia se usa sólo si la matriz tiene ingresos de punto flotante y no contiene ninguna variable simbólica a la que no se le haya asignado un valor. De otro modo, la *Tolerancia* se ignora.

- Si usted usa **ctrl** **enter** o configura el modo **Auto o Aproximado** para aproximar, los cálculos se realizan al usar la aritmética de punto flotante.
- Si la *Tolerancia* se omite o no se usa, la tolerancia predeterminada se calcula como:
 $5E-14 \cdot \max(\dim(\text{MatrizI})) \cdot \text{normFila}(\text{MatrizI})$

Evite los elementos indefinidos en *MatrizI*. Pueden conllevar a resultados inesperados.

Por ejemplo, si *a* es indefinido en la siguiente expresión, aparecerá un mensaje de advertencia y el resultado se muestra como:

$$\text{ref}\left(\begin{bmatrix} a & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\right) \Rightarrow \begin{bmatrix} 1 & \frac{1}{a} & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

La advertencia aparece porque el elemento generalizado $1/a$ no sería válido para $a=0$.

Puede evitar esto almacenando un valor *a* de antemano o utilizando el operador restrictivo "!" para sustituir un valor, tal como se muestra en el siguiente ejemplo.

$$\text{ref}\left(\begin{bmatrix} a & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \mid a=0\right) \Rightarrow \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

Nota: Vea también **rref()**, página 107.

$\text{ref}\left(\begin{bmatrix} -2 & -2 & 0 & -6 \\ 1 & -1 & 9 & -9 \\ -5 & 2 & 4 & -4 \end{bmatrix}\right)$	$\begin{bmatrix} 1 & \frac{-2}{5} & \frac{-4}{5} & \frac{4}{5} \\ 0 & 1 & \frac{4}{7} & \frac{11}{7} \\ 0 & 0 & 1 & \frac{-62}{71} \end{bmatrix}$
$\begin{bmatrix} a & b & c \\ e & f & g \end{bmatrix} \rightarrow mI$	$\begin{bmatrix} a & b & c \\ e & f & g \end{bmatrix}$
$\text{ref}(mI)$	$\begin{bmatrix} a & b & c \\ e & f & g \end{bmatrix}$

remain() (rest)

Catálogo >

remain(*Expr1*, *Expr2*) \Rightarrow *expresión***remain**(*Lista1*, *Lista2*) \Rightarrow *lista***remain**(*Matriz1*, *Matriz2*) \Rightarrow *matriz*

Entrega el resto del primer argumento con respecto del segundo argumento conforme se definen por medio de las identidades:

rest(*x*,0) \times rest(*x*,*y*) \times -*y* · partel(*x*/*y*)Como consecuencia, tome en cuenta que **remain**(-*x*,*y*)= -**remain**(*x*,*y*). El resultado es cero o tiene el mismo signo que el primer argumento.**Nota:** Vea también **mod()**, página 79.

remain(7,0)	7
remain(7,3)	1
remain(-7,3)	-1
remain(7,-3)	1
remain(-7,-3)	-1
remain({12,-14,16},{9,7,-5})	{3,0,1}
remain($\begin{pmatrix} 9 & -7 \\ 6 & 4 \end{pmatrix}, \begin{pmatrix} 4 & 3 \\ 4 & -3 \end{pmatrix}$)	$\begin{pmatrix} 1 & -1 \\ 2 & 1 \end{pmatrix}$

Request (Solicitar)

Catálogo >

Request *cadenaIndicadora*, *var* [, *DespBandera* [, *varEstado*]]**Request** *cadenaIndicadora*, *func*(*arg1*, ...*argn*)
[, *DespBandera* [, *varEstado*]]

Comando de programación: Pausa el programa y despliega un cuadro de diálogo que contiene el mensaje *cadenaIndic* y un cuadro de entrada para la respuesta del usuario.

Cuando el usuario escribe una respuesta y hace clic en **OK**, el contenido del cuadro de entrada se asigna a la variable *var*.

Si el usuario hace clic en **Cancelar**, el programa procede sin aceptar ninguna entrada. El programa usa el valor anterior de *var* si *var* ya se había definido.

El argumento *DespBandera* opcional puede ser cualquier expresión.

- Si *DespBandera* se omite o se evalúa a **1**, el mensaje de indicación y la respuesta del usuario se despliegan en el historial de la Calculadora.
- Si *DespBandera* se evalúa a **0**, la indicación y la respuesta no se despliegan en el historial.

Defina un programa:

```
Define request_demo=Prgm
Request "Radius: ",r
Disp "Area = ",pi*r^2
EndPrgm
```

Ejecute el programa y escriba una respuesta:

request_demo()

Resultado después de seleccionar **OK**:

```
Radius: 6/2
Area= 28.2743
```

Request (Solicitar)

Catálogo > 

El argumento *varEstado* opcional le da al programa una manera de determinar cómo el usuario descartó el cuadro de diálogo. Tome en cuenta que *varEstado* requiere el argumento *DespBandera*.

- Si el usuario hizo clic en **OK** o presionó **Intro** o **Ctrl+Intro**, la variable *varEstado* estará configurada a un valor de **1**.
- De otra manera, la variable *varEstado* se configura a un valor de **0**.

El argumento *func()* permite que un programa almacene la respuesta del usuario como una definición de función. Esta sintaxis opera como si el usuario ejecutara el comando:

Define *func(arg1, ...argn)* = respuesta del usuario

Entonces el programa puede usar la función definida *func()*. La *cadenaIndic* debería guiar al usuario para ingresar una *respuesta del usuario* apropiada que complete la definición de función.

Nota: Usted puede usar el comando **Request** dentro de un programa definido por el usuario, pero no dentro de una función.

Para detener un programa que contiene un **Request** comando adentro de un bucle infinito:

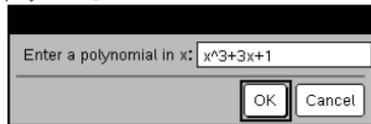
- **Windows®:** Mantenga presionada la tecla **F12** y presione **Intro** varias veces.
- **Macintosh®:** Mantenga presionada la tecla **F5** y presione **Intro** varias veces.
- **Receptor portátil:** Mantenga presionada la tecla  y presione  varias veces.

Nota: Vea también **RequestStr**, página 103.

Defina un programa:

```
Define polynomial()=Prgm
Request "Enter a polynomial in x:",p(x)
Disp "Real roots are: ",polyRoots(p(x),x)
EndPrgm
```

Ejecute el programa y escriba una respuesta:
polynomial()



Resultado después de seleccionar **OK**:

```
Enter a polynomial in x: x^3+3x+1
Real roots are: {-0.322185}
```

RequestStr (SolicitarCad)

Catálogo > 

RequestStr *cadenaIndic, var[, DespBandera]*

Comando de programación: Opera en forma idéntica a la primera sintaxis del comando **Request**, excepto que la respuesta del usuario siempre se interpreta como una cadena. En contraste, el comando **Request** interpreta la respuesta como una expresión, a menos que el usuario la encierre entre comillas ("").

Nota: Usted puede usar el comando **RequestStr** dentro de un programa definido por el usuario, pero no dentro de una función.

Para detener un programa que contiene un **RequestStr** comando adentro de un bucle infinito:

- **Windows®:** Mantenga presionada la tecla **F12** y presione **Intro** varias veces.
- **Macintosh®:** Mantenga presionada la tecla **F5** y presione **Intro** varias veces.
- **Receptor portátil:** Mantenga presionada la tecla  y presione  varias veces.

Nota: Vea también **Request**, página 102.

Defina un programa:

```
Define requestStr_demo()=Prgm
RequestStr "Your name:",name,0
Disp "Response has ",dim(name)," characters."
EndPrgm
```

Ejecute el programa y escriba una respuesta:
requestStr_demo()



El resultado después de seleccionar **OK** (Tome en cuenta que el argumento *DespBandera* de **0** omite la indicación y la respuesta del historial):

```
requestStr_demo()
```

```
Response has 5 characters.
```

Return (Producir)Catálogo > **Return** [*Expr*]

Entrega *Expr* como el resultado de la función. Use dentro de un bloque de **Func...EndFunc**.

Nota: Use **Return** sin un argumento dentro de un bloque de **Prgm...EndPrgm** para salir de un programa.

Nota para ingresar el ejemplo: En la aplicación de Calculadora en el dispositivo portátil, usted puede ingresar definiciones en líneas múltiples al presionar  en lugar de  al final de cada línea. En el teclado de la computadora, presione y sostenga **Alt** y presione **Ingresar**.

 Define $factoral(nn) = \text{Func}$
Local *answer, count* $1 \rightarrow \text{answer}$ For *count, 1, nn**answer*·*count* \rightarrow *answer*

EndFor

Return *answer*

EndFunc

Done

 $factoral(3)$

6

right() (derecha)Catálogo > **right**(*Lista1*, *Núm*) \Rightarrow *lista*

Entrega los elementos de *Num* del extremo derecho contenidos en *Lista1*.

Si usted omite *Num*, entrega toda la *Lista1*.

right(*cadenaFuente*, *Num*) \Rightarrow *cadena*

Entrega los caracteres de *Num* del extremo derecho contenidos en una cadena de caracteres *cadenaFuente*.

Si usted omite *Num*, entrega toda la *cadenaFuente*.

right(*Comparación*) \Rightarrow *expresión*

Entrega el lado derecho de una ecuación o desigualdad.

 $right(\{1, 3, -2, 4\}, 3)$
 $\{3, -2, 4\}$

 $right("Hello", 2)$

"lo"

 $right(x < 3)$

3

rk23(Expr, Var, varDep, {Var0, VarMax}, var0Dep, PasoVar [, toldif]) matriz \Rightarrow

rk23(SistemaDeExpr, Var, ListaDeVarsDep, {Var0, VarMax}, ListaDeVars0Dep, PasoVar [, toldif]) matriz \Rightarrow

rk23(SistemaDeExpr, Var, ListaDeVarsDep, {Var0, VarMax}, ListaDeVars0Dep, PasoVar [, toldif]) matriz \Rightarrow

Use el método de Runge-Kutta para resolver el sistema

$$\frac{d \text{depVar}}{d \text{Var}} = \text{Expr}(\text{Var}, \text{varDep})$$

con $\text{varDep}(\text{Var0}) = \text{var0Dep}$ en el intervalo $[\text{Var0}, \text{VarMax}]$. Entrega una matriz cuya primera fila define los valores de resultado de Var conforme se definen por medio de PasoVar . La segunda fila define el valor del primer componente de solución a los valores de Var correspondientes, y así sucesivamente.

Expr es el lado derecho que define la ecuación diferencial ordinaria (EDO).

SistemaDeExpr es un sistema de lados derechos que define el sistema de EDOs (corresponde al orden de variables dependientes en *ListaDeVarsDep*).

ListaDeExpr es una lista de lados derechos que define el sistema de EDOs (corresponde al orden de variables dependientes en *ListaDeVarsDep*).

Var es la variable independiente.

ListaDeVarsDep es una lista de variables dependientes.

{*Var0*, *VarMax*} es una lista de dos elementos que le dice a la función que se integre de *Var0* a *VarMax*.

ListaDeVars0Dep es una lista de valores iniciales para variables dependientes.

Si *PasoVar* se evalúa a un número distinto de cero: $\text{signo}(\text{PasoVar}) = \text{signo}(\text{VarMax} - \text{Var0})$ y las soluciones se entregan a $\text{Var0} + i \cdot \text{PasoVar}$ para todos $i=0,1,2,\dots$ de tal manera que $\text{Var0} + i \cdot \text{PasoVar}$ está en $[\text{var0}, \text{VarMax}]$ (puede que no haya un valor de solución en *VarMax*).

Si *PasoVar* se evalúa a cero, las soluciones se entregan a los valores de *Var* de "Runge-Kutta".

tol dif es la tolerancia de error (predeterminado a 0.001).

Ecuación diferencial:

$$y' = 0.001 \cdot y \cdot (100 - y) \text{ y } y(0) = 10$$

$$\text{rk23}\left(0.001 \cdot y \cdot (100 - y), t, y, \{0, 100\}, 10, 1\right)$$

0.	1.	2.	3.	4.
10.	10.9367	11.9493	13.042	14.2

Para ver el resultado completo, presione \blacktriangle y después use \blacktriangleleft \blacktriangleright para mover el cursor.

La misma ecuación con *tol dif* configurada a $1.E-6$

$$\text{rk23}\left(0.001 \cdot y \cdot (100 - y), t, y, \{0, 100\}, 10, 1, 1.E-6\right)$$

0.	1.	2.	3.	4.
10.	10.9367	11.9495	13.0423	14.2189

Compare el resultado anterior con la solución exacta de CAS obtenido al usar de *Resolver()* y *genSec()*:

$$\text{deSolve}\left(y' = 0.001 \cdot y \cdot (100 - y) \text{ and } y(0) = 10, t, y\right)$$

$$y = \frac{100 \cdot (1.10517)^t}{(1.10517)^t + 9}$$

$$\text{seqGen}\left(\frac{100 \cdot (1.10517)^t}{(1.10517)^t + 9}, t, y, \{0, 100\}\right)$$

$$\{10., 10.9367, 11.9494, 13.0423, 14.2189, 15.4\}$$

Sistema de ecuaciones:

$$\begin{cases} y1' = y1 + 0.1 \cdot y1 \cdot y2 \\ y2' = 3 \cdot y2 - y1 \cdot y2 \end{cases}$$

con $y1(0) = 2$ y $y2(0) = 5$

$$\text{rk23}\left(\begin{cases} -y1 + 0.1 \cdot y1 \cdot y2 \\ 3 \cdot y2 - y1 \cdot y2 \end{cases}, t, \{y1, y2\}, \{0, 5\}, \{2, 5\}, 1\right)$$

0.	1.	2.	3.	4.
2.	1.94103	4.78694	3.25253	1.82848
5.	16.8311	12.3133	3.51112	6.27245

root(Expr) \Rightarrow raíz

root(Expr1, Expr2) \Rightarrow raíz

root(Expr) entrega la raíz cuadrada de Expr.

root(Expr1, Expr2) entrega la raíz de Expr2 de Expr1. Expr1 puede ser una constante de punto flotante compleja, un entero o una constante racional compleja, o bien una expresión simbólica general.

Nota: Vea también **Plantilla de raíz N-ésima**, página 1.

$\sqrt[3]{8}$	2
$\sqrt[3]{3}$	$\frac{1}{3^3}$
$\sqrt[3]{3}$	1.44225

rowAdd() (agrFila)Catálogo > **rowAdd**(*Matriz1*, *indiceR1*, *indiceR2*) ⇒ *matriz*Entrega una copia de *Matriz1* con la fila *indiceR2* reemplazada por la suma de las filas *indiceR1* e *indiceR2*.

$\text{rowAdd}\left(\begin{bmatrix} 3 & 4 \\ -3 & -2 \end{bmatrix}, 1, 2\right)$	$\begin{bmatrix} 3 & 4 \\ 0 & 2 \end{bmatrix}$
$\text{rowAdd}\left(\begin{bmatrix} a & b \\ c & d \end{bmatrix}, 1, 2\right)$	$\begin{bmatrix} a & b \\ a+c & b+d \end{bmatrix}$

rowDim() (dimFila)Catálogo > **rowDim**(*Matriz*) ⇒ *expresión*Entrega el número de filas en *Matriz*.**Nota:** Vea también **colDim()**, página 19.

$\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix} \rightarrow m1$	$\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}$
$\text{rowDim}(m1)$	3

rowNorm()Catálogo > **rowNorm**(*Matriz*) ⇒ *expresión*Entrega el máximo de las sumas de los valores absolutos de los elementos en las filas en *Matriz*.**Nota:** Todos los elementos de la matriz se deben simplificar a números. Vea también **colNorm()**, página 19.

$\text{rowNorm}\left(\begin{bmatrix} -5 & 6 & -7 \\ 3 & 4 & 9 \\ 9 & -9 & -7 \end{bmatrix}\right)$	25
--	----

rowSwap() (cambioFila)Catálogo > **rowSwap**(*Matriz1*, *indiceR1*, *indiceR2*) ⇒ *matriz*Entrega *Matriz1* con las filas *indiceR1* e *indiceR2* intercambiadas.

$\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix} \rightarrow mat$	$\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}$
$\text{rowSwap}(mat, 1, 3)$	$\begin{bmatrix} 5 & 6 \\ 3 & 4 \\ 1 & 2 \end{bmatrix}$

rref()Catálogo > **rref**(*Matriz1*, *Tol*) ⇒ *matriz*Entrega la forma escalonada reducida por filas de *Matriz1*.

$\text{rref}\left(\begin{bmatrix} -2 & -2 & 0 & -6 \\ 1 & -1 & 9 & -9 \\ -5 & 2 & 4 & -4 \end{bmatrix}\right)$	$\begin{bmatrix} 1 & 0 & 0 & \frac{66}{71} \\ 0 & 1 & 0 & \frac{147}{71} \\ 0 & 0 & 1 & \frac{-62}{71} \end{bmatrix}$
--	---

rref()

Catálogo >

De manera opcional, cualquier elemento de matriz se trata como cero si su valor absoluto es menor que la *Tolerancia*. Esta tolerancia se usa sólo si la matriz tiene ingresos de punto flotante y no contiene ninguna variable simbólica a la que no se le haya asignado un valor. De otro modo, la *Tolerancia* se ignora.

$$\text{rref}\left(\begin{bmatrix} a & b & x \\ c & d & y \end{bmatrix}\right) \quad \begin{bmatrix} a & b & x \\ c & d & y \end{bmatrix}$$

- Si usted usa **ctrl** **enter** o configura el modo **Auto o Aproximado** para aproximar, los cálculos se realizan al usar la aritmética de punto flotante.
- Si la *Tolerancia* se omite o no se usa, la tolerancia predeterminada se calcula como:
 $5E-14 \cdot \max(\dim(\text{Matriz}1)) \cdot \text{normFila}(\text{Matriz}1)$

Nota: Vea también **rref()**, página 101.

S**sec()** **tecla**

sec(*Expr1*) \Rightarrow expresión

sec(*Lista1*) \Rightarrow lista

Entrega la secante de *Expr1* o entrega una lista que contiene las secantes de todos los elementos en *Lista1*.

Nota: El argumento se interpreta como un ángulo en grados, gradianes o radianes, de acuerdo con la configuración del modo del ángulo actual. Se puede usar $^{\circ}$, $^{\text{G}}$, o $^{\text{r}}$ para anular el modo de ángulo en forma temporal.

En modo de ángulo en Grados:

$$\begin{array}{l} \text{sec}(45) \\ \text{sec}(\{1,2,3,4\}) \end{array} \quad \begin{array}{l} \sqrt{2} \\ \left\{ \frac{1}{\cos(1)}, 1.00081, \frac{1}{\cos(4)} \right\} \end{array}$$

sec⁻¹() **tecla**

sec⁻¹(*Expr1*) \Rightarrow expresión

sec⁻¹(*Lista1*) \Rightarrow lista

Entrega el ángulo cuya secante es *Expr1* o entrega una lista que contiene las secantes inversas de cada elemento de *Lista1*.

Nota: El resultado se entrega como un ángulo en grados, gradianes o radianes, de acuerdo con la configuración del modo del ángulo actual.

Nota: Usted puede insertar esta función desde el teclado al escribir **arcsec**(...).

En modo de ángulo en Grados:

$$\text{sec}^{-1}(1) \quad 0$$

En modo de ángulo en Gradianes:

$$\text{sec}^{-1}(\sqrt{2}) \quad 50$$

En modo de ángulo en Radianes:

$$\text{sec}^{-1}(\{1,2,5\}) \quad \left\{ 0, \frac{\pi}{3}, \cos^{-1}\left(\frac{1}{5}\right) \right\}$$

sech()

Catálogo >

sech(*Expr1*) \Rightarrow expresión

sech(*Lista1*) \Rightarrow lista

Entrega la secante hiperbólica de *Expr1* o entrega una lista que contiene las secantes hiperbólicas de los elementos de *Lista1*.

$$\text{sech}(3) \quad \frac{1}{\cosh(3)}$$

$$\text{sech}(\{1,2,3,4\}) \quad \left\{ \frac{1}{\cosh(1)}, 0.198522, \frac{1}{\cosh(4)} \right\}$$

sech⁻¹()

Catálogo >

sech⁻¹(Expr1) ⇒ expresión**sech⁻¹(Valor1)** ⇒ valor**sech⁻¹(Lista1)** ⇒ lista

Entrega la secante hiperbólica inversa de *Expr1* o entrega una lista que contiene las secantes hiperbólicas inversas de cada elemento de *Lista1*.

Nota: Usted puede insertar esta función desde el teclado al escribir **arcsech (...)**.

En el modo de ángulo en Radianes y el modo complejo Rectangular:

$\text{sech}^{-1}(1)$	0
$\text{sech}^{-1}(\{1, -2, 2.1\})$	$\left\{0, \frac{2 \cdot \pi}{3} \cdot i, 8. \epsilon^{-15} + 1.07448 \cdot i\right\}$

seq() (secuen)

Catálogo >

seq(Expr, Var, Bajo, Alto, Paso) ⇒ lista

Incrementa *Var* desde *Bajo* hasta *Alto* por un incremento de *Paso*, evalúa *Expr* y entrega los resultados como una lista. Los contenidos originales de *Var* están ahí todavía después de que se completa **seq()**.

El valor predeterminado para *Paso* = 1.

$\text{seq}(n^2, n, 1, 6)$	$\{1, 4, 9, 16, 25, 36\}$
$\text{seq}\left(\frac{1}{n}, n, 1, 10, 2\right)$	$\left\{1, \frac{1}{3}, \frac{1}{5}, \frac{1}{7}, \frac{1}{9}\right\}$
$\text{sum}\left(\text{seq}\left(\frac{1}{n^2}, n, 1, 10, 1\right)\right)$	$\frac{1968329}{1270080}$

Presione **Ctrl+Ingresar** (Macintosh@:

⌘+Ingresar) para evaluar:

$\text{sum}\left(\text{seq}\left(\frac{1}{n^2}, n, 1, 10, 1\right)\right)$	1.54977
--	---------

seqGen()

Catálogo > 

seqGen(*Expr*, *Var*, *varDep*, {*Var0*, *VarMax*}, [*ListaDeTérminosInic* [, *PasoVar* [, *ValorMax*]]) *lista* ⇒

Genera una lista de términos para la secuencia $varDep(Var)=Expr$ como sigue: Incrementa la variable independiente *Var* desde *Var0* hasta *VarMax* por medio de *PasoVar*, evalúa $varDep(Var)$ para los valores correspondientes de *Var* usando la fórmula *Expr* y *ListaDeTérminosInic*, y entrega los resultados como una lista.

seqGen(*ListaOSistemaDeExpr*, *Var*, *ListaDeVarsDep*, {*Var0*, *VarMax*}, [*MatrizDeTérminosInic* [, *PaspVar* [, *ValorMax*]]) *matriz* ⇒

Genera una matriz de términos para un sistema (o una lista) de secuencias $ListaDeVarsDep(Var)=ListaOSistemaDeExpr$ como sigue: Incrementa la variable independiente *Var* desde *Var0* hasta *VarMax* por medio de *PasoVar*, evalúa $ListaDeVarsDep(Var)$ para los valores correspondientes de *Var* usando la fórmula *ListaOSistemaDeExpr* y *MatrizDeTérminosInic*, y entrega los resultados como una matriz.

Los contenidos originales de *Var* no cambian después de que se completa **seqGen**().

El valor predeterminado para *PasoVar* = 1.

Genera los 5 primeros términos de la secuencia $u(n) = u(n-1)^2/2$, con $u(1)=2$ y *PasoVar*=1.

$$\text{seqGen}\left(\frac{u(n-1)^2}{n}, n, u, \{1, 5\}, \{2\}\right)$$

$$\left\{2, 2, \frac{4}{3}, \frac{4}{9}, \frac{16}{405}\right\}$$

Ejemplo en el que *Var0*=2:

$$\text{seqGen}\left(\frac{u(n-1)+1}{n}, n, u, \{2, 5\}, \{3\}\right)$$

$$\left\{3, \frac{4}{3}, \frac{7}{12}, \frac{19}{60}\right\}$$

Ejemplo en el que el término inicial es simbólico:

$$\text{seqGen}\{u(n-1)+2, n, u, \{1, 5\}, \{a\}\}$$

$$\{a, a+2, a+4, a+6, a+8\}$$

Sistema de dos secuencias:

$$\text{seqGen}\left(\left\{\frac{1}{n}, \frac{u_2(n-1)}{2} + u_1(n-1)\right\}, n, \{u_1, u_2\}, \{1, 5\}, \left[\begin{array}{c} _ \\ 2 \end{array}\right]\right)$$

$$\left[\begin{array}{ccccc} 1 & \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \\ 2 & 2 & \frac{3}{2} & \frac{13}{12} & \frac{19}{24} \end{array}\right]$$

Nota: El Vacío () en la matriz de términos iniciales anterior se usa para indicar que el término inicial para $u_1(n)$ se calcula utilizando la fórmula de secuencia explícita $u_1(n)=1/n$.

seqn()

Catálogo > 

seqn(*Expr*(*u*, *n* [, *ListaDeTérminosInic* [, *nMax* [, *ValorMax*]]) *lista* ⇒

Genera una lista de términos para una secuencia $u(n)=Expr(u, n)$ como sigue: Incrementa *n* desde 1 hasta *nMax* por 1, evalúa $u(n)$ para los valores correspondientes de *n* usando la fórmula *Expr*(*u*, *n*) y *ListaDeTérminosInic*, y entrega los resultados como una lista.

seqn(*Expr*(*n* [, *nMax* [, *ValorMax*]]) *lista* ⇒

Genera una lista de términos para una secuencia no recursiva $u(n)=Expr(n)$ como sigue: Incrementa *n* desde 1 hasta *nMax* por 1, evalúa $u(n)$ para los valores correspondientes de *n* usando la fórmula *Expr*(*n*) y entrega los resultados como una lista.

Si *nMax* falta, *nMax* se configura a 2500

Si *nMax*=0, *nMax* se configura a 2500

Nota: **seqn**() llama **seqGen**() con $n0=1$ y $npaso=1$

Genera los 6 primeros términos de la secuencia $u(n) = u(n-1)/2$, con $u(1)=2$.

$$\text{seqn}\left(\frac{u(n-1)}{n}, \{2\}, 6\right)$$

$$\left\{2, 1, \frac{1}{3}, \frac{1}{12}, \frac{1}{60}, \frac{1}{360}\right\}$$

$$\text{seqn}\left(\frac{1}{n^2}, 6\right)$$

$$\left\{1, \frac{1}{4}, \frac{1}{9}, \frac{1}{16}, \frac{1}{25}, \frac{1}{36}\right\}$$

series(Expr1, Var, Orden [, Punto]) ⇒ expresión

series(Expr1, Var, Orden [, Punto]) | Var > Punto ⇒ expresión

series(Expr1, Var, Orden [, Punto]) | Var < Punto ⇒ expresión

Entrega una representación de serie de potencia truncada de Expr1 expandida alrededor de Punto a través del grado Orden. Orden puede ser cualquier número racional. Las potencias resultantes de (Var - Punto) pueden incluir exponentes negativos y/o fraccionales. Los coeficientes de estas potencias pueden incluir logaritmos de (Var - Punto) y otras funciones de Var que están dominadas por todas las potencias de (Var - Punto) teniendo el mismo signo de exponente.

Punto se predetermina a 0. Punto puede ser ∞ o -∞, en cuyos casos la expansión es por medio del grado Orden en 1/(Var - Punto).

series(...) entrega "series(...)" si es incapaz de determinar tal representación, como para singularidades esenciales como $\sin(1/z)$ en $z=0$, $e^{-1/z}$ en $z=0$ ó e^z en $z = \infty$ o $-\infty$.

Si la serie o una de sus derivadas tiene una discontinuidad de salto en Punto, es probable que el resultado contenga subexpresiones del signo de forma(...) o abs(...) para una variable de expansión real o (-1)^{pisos(...angle(...))} para una variable de expansión compleja, que es una que termina con "...". Si usted intenta usar la serie sólo para los valores en un lado de Punto, entonces añada el apropiado de " | Var > Punto", " | Var < Punto", " | Var ≥ Punto" o " | Var ≤ Punto" para obtener un resultado más sencillo.

series() puede proporcionar aproximaciones simbólicas para integrales indefinidas e integrales definidas para las cuales de otro modo no se pueden obtener soluciones simbólicas.

series() se distribuye sobre listas y matrices del 1er argumento.

series() es una versión generalizada de **taylor**().

Conforme se ilustra por medio del último ejemplo de la derecha, la corriente abajo de las rutinas de despliegue del resultado producido por serie(...) podría reorganizar los términos de manera que el término dominante no sea el del extremo izquierdo.

Nota: Vea también **dominantTerm**(), página 40.

$$\text{series}\left(\frac{1-\cos(x-1)}{(x-1)^2}, x, 4, 1\right)$$

$$\frac{1}{2} \frac{(x-1)^2}{24} + \frac{(x-1)^4}{720}$$

$$\text{series}(\ln(x^x-1), x, 2)$$

$$\ln(x \cdot \ln(x)) + \frac{x \cdot \ln(x)}{2} + \frac{x^2 \cdot (\ln(x))^2}{24}$$

$$\text{series}\left(e^{\frac{-1}{z-1}}, z, 1, 0, 0\right)$$

$$\text{series}\left(e^{\frac{-1}{z-1}}, z, 1, 0, 0\right)$$

$$\text{series}\left(\left(1+\frac{1}{n}\right)^n, n, 2, \infty\right)$$

$$e - \frac{e}{2 \cdot n} + \frac{11 \cdot e}{24 \cdot n^2}$$

$$\text{series}\left(\tan^{-1}\left(\frac{1}{x}\right), x, 5\right), x > 0$$

$$\frac{\pi}{2} - x + \frac{x^3}{3} - \frac{x^5}{5}$$

$$\text{series}\left(\int \frac{\sin(x)}{x} dx, x, 6\right)$$

$$x - \frac{x^3}{18} + \frac{x^5}{600}$$

$$\text{series}\left(\int_0^x \sin(x \cdot \sin(t)) dt, x, 7\right)$$

$$\frac{x^3}{2} - \frac{x^5}{24} + \frac{29 \cdot x^7}{720}$$

$$\text{series}\left(\left(1+e^x\right)^2, x, 2, 1\right)$$

$$e \cdot (2 \cdot e + 1) \cdot (x-1)^2 + (2 \cdot e^2 + 2 \cdot e) \cdot (x-1) + (e+1)^2$$

setMode(enteroNombreModo, enteroConfig) ⇒ entero
setMode(lista) ⇒ lista de enteros

Sólo es válido dentro de una función o un programa.

setMode(enteroNombreModo, enteroConfig) configura en forma temporal el modo *enteroNombreModo* a la nueva configuración *enteroConfig* entrega un entero correspondiente a la configuración original de ese modo. El cambio está limitado a la duración de la ejecución del programa/la función.

enteroNombreModo especifica cuál modo usted desea configurar. Debe ser uno de los enteros de modo de la tabla de abajo.

enteroConfig especifica la nueva configuración para el modo. Debe ser uno de los enteros de configuración que se enumeran abajo para el modo específico que usted está configurando.

setMode(lista) le permite cambiar varias configuraciones. *lista* contiene pares de enteros de modo y enteros de configuración.

setMode(lista) entrega una lista similar cuyos pares de enteros representan los modos y las configuraciones originales.

Si usted ha guardado todas las configuraciones de modo con **getMode(0)** → *var*, podrá usar **setMode(var)** para restaurar esas configuraciones hasta que la función o el programa exista. Vea **getMode()**, página 55.

Nota: Las configuraciones del modo actual se pasan a las subrutinas llamadas. Si cualquier subrutina cambia una configuración del modo, el cambio de modo se perderá cuando el control regrese a la rutina de llamada.

Nota para ingresar el ejemplo: En la aplicación de Calculadora en el dispositivo portátil, usted puede ingresar definiciones en líneas múltiples al presionar  en lugar de **enter** al final de cada línea. En el teclado de la computadora, presione y sostenga **Alt** y presione **Ingresar**.

Despliegue el valor aproximado de π usando la configuración predeterminada para Desplegar Dígitos, y luego despliegue π con una configuración de Fijo2. Revise para ver que el predeterminado esté restaurado después de que se ejecute el programa.

```
Define prog1()=Prgm                                Done
    Disp approx( $\pi$ )
    setMode(1,16)
    Disp approx( $\pi$ )
EndPrgm
-----
prog1()
-----
3.14159
3.14
-----
Done
```

Modo Nombre	Modo Entero	Cómo configurar enteros
Desplegar dígitos	1	1=Flotante, 2=Flotante1, 3=Flotante2, 4=Flotante3, 5=Flotante4, 6=Flotante5, 7=Flotante6, 8=Flotante7, 9=Flotante8, 10=Flotante9, 11=Flotante10, 12=Flotante11, 13=Flotante12, 14=Fijo0, 15=Fijo1, 16=Fijo2, 17=Fijo3, 18=Fijo4, 19=Fijo5, 20=Fijo6, 21=Fijo7, 22=Fijo8, 23=Fijo9, 24=Fijo10, 25=Fijo11, 26=Fijo12
Ángulo	2	1=Radián, 2=Grado, 3=Gradián
Formato exponencial	3	1=Normal, 2=Científico, 3=Ingeniería
Real o Complejo	4	1=Real, 2=Rectangular, 3=Polar
Auto o Aprox.	5	1=Auto, 2=Aproximado, 3=Exacto
Formato de Vector	6	1=Rectangular, 2=Cilíndrico, 3=Esférico
Base	7	1=Decimal, 2=Hexagonal, 3=Binario
Sistema de unidad	8	1=SI, 2=Ing/EEUU

shift(EnteroI [,#deCambios]) ⇒ entero

Cambia los bits en un entero binario. Usted puede ingresar *EnteroI* en cualquier base de números; se convierte automáticamente en una forma binaria de 64 bits signada. Si la magnitud de *EnteroI* es demasiado grande para esta forma, una operación de módulo simétrico lo lleva dentro del rango. Para obtener más información, vea ▶**Base2**, página 14.

Si *#deCambios* es positivo, el cambio es hacia la izquierda. Si *#deCambios* es negativo, el cambio es hacia la derecha. El predeterminado es -1 (cambiar a la derecha un bit).

En un cambio a la derecha, el bit del extremo derecho se elimina y 0 ó 1 se inserta para coincidir con el bit del extremo izquierdo. En un cambio a la izquierda, el bit del extremo izquierdo se elimina y 0 ó 1 se inserta como el bit del extremo derecho.

Por ejemplo, en un cambio a la derecha:

Cada bit cambia a la derecha.

0b0000000000000111101011000011010

Inserta 0 si el bit del extremo izquierdo es 0, ó 1 si el bit del extremo izquierdo es 1.

produce:

0b00000000000000111101011000011010

El resultado se despliega de acuerdo con el modo de la Base. Los ceros líderes no se muestran.

shift(Lista1 [,#deCambios]) ⇒ lista

Entrega una copia de *Lista1* cambiada a la derecha o a la izquierda por elementos de *#de Cambios*. No altera *Lista1*.

Si *#deCambios* es positivo, el cambio es hacia la izquierda. Si *#deCambios* es negativo, el cambio es hacia la derecha. El predeterminado es -1 (cambiar a la derecha un elemento).

Los elementos introducidos al principio o al final de *lista* por medio del cambio están configurados al símbolo "undef".

shift(Cadena1 [,#deCambios]) ⇒ cadena

Entrega una copia de *Cadena1* cambiada a la derecha o a la izquierda por caracteres de *#de Cambios*. No altera *Cadena1*.

Si *#deCambios* es positivo, el cambio es hacia la izquierda. Si *#deCambios* es negativo, el cambio es hacia la derecha. El predeterminado es -1 (cambiar a la derecha un carácter).

Los caracteres introducidos al principio o al final de *cadena* por medio del cambio están configurados a un espacio.

En modo de base binaria:

shift(0b1111010110000110101)	0b111101011000011010
shift(256,1)	0b1000000000

En modo de base hexadecimal:

shift(0h78E)	0h3C7
shift(0h78E,-2)	0h1E3
shift(0h78E,2)	0h1E38

Importante: Para ingresar un número binario o hexadecimal, use siempre el prefijo 0b ó 0h (cero, no la letra O).

En modo de base decimal:

shift({1,2,3,4})	{undef,1,2,3}
shift({1,2,3,4},-2)	{undef,undef,1,2}
shift({1,2,3,4},2)	{3,4,undef,undef}

shift("abcd")	" abc"
shift("abcd",-2)	" ab"
shift("abcd",1)	"bcd "

sign()

Catálogo >

sign(*Expr1*) ⇒ *expresión***sign**(*Lista1*) ⇒ *lista***sign**(*Matriz1*) ⇒ *matriz*Para *Expr1* real o compleja, entrega $\text{Expr1}/\text{abs}(\text{Expr1})$ cuando $\text{Expr1} \neq 0$.Entrega 1 si *Expr1* es positiva.Entrega -1 si *Expr1* es negativa.**sign(0)** entrega ± 1 si el modo de formato complejo es Real; de otro modo, se entrega a sí mismo.**sign(0)** representa el círculo de unidad en el dominio complejo.

Para una lista o matriz, entrega los signos de todos los elementos.

$\text{sign}(-3,2)$	-1.
$\text{sign}\{2,3,4,-5\}$	$\{1,1,1,-1\}$
$\text{sign}(1+ x)$	1

Si el modo de formato complejo es Real:

$\text{sign}[-3 \ 0 \ 3]$	$[-1 \ \pm 1 \ 1]$
---------------------------	--------------------

simult()

Catálogo >

simult(*matrizCoeff*, *vectorConst*[, *Tol*]) ⇒ *matriz*

Entrega un vector de columna que contiene las soluciones para un sistema de ecuaciones lineales.

Nota: Vea también **linSolve()**, página 69.*matrizCoeff* debe ser una matriz cuadrada que contiene los coeficientes de las ecuaciones.*vectorConst* debe tener el mismo número de filas (misma dimensión) que *matrizCoeff* y contener las constantes.De manera opcional, cualquier elemento de matriz se trata como cero si su valor absoluto es menor que la *Tolerancia*. Esta tolerancia se usa sólo si la matriz tiene ingresos de punto flotante y no contiene ninguna variable simbólica a la que no se le haya asignado un valor. De otro modo, la *Tolerancia* se ignora.

- Si usted configura el modo **Auto o Aproximado** en Aproximado, los cálculos se hacen usando aritmética de punto flotante.
- Si la *Tolerancia* se omite o no se usa, la tolerancia predeterminada se calcula como:
 $5E-14 \cdot \max(\dim(\text{matrizCoeff})) \cdot \text{normaFila}(\text{matrizCoeff})$

Solución para x y y:

$x + 2y = 1$

$3x + 4y = -1$

$\text{simult}\left(\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, \begin{bmatrix} 1 \\ -1 \end{bmatrix}\right)$	$\begin{bmatrix} -3 \\ 2 \end{bmatrix}$
---	---

La solución es $x=-3$ y $y=2$.

Solución:

$ax + by = 1$

$cx + dy = 2$

$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \rightarrow \text{matriz1}$	$\begin{bmatrix} a & b \\ c & d \end{bmatrix}$
$\text{simult}\left(\text{matriz1}, \begin{bmatrix} 1 \\ 2 \end{bmatrix}\right)$	$\begin{bmatrix} -(2 \cdot b - d) \\ a \cdot d - b \cdot c \\ 2 \cdot a - c \\ a \cdot d - b \cdot c \end{bmatrix}$

simult(*matrizCoeff*, *matrizConst*[, *Tol*]) ⇒ *matriz*

Soluciona varios sistemas de ecuaciones lineales, donde cada sistema tiene los mismos coeficientes de ecuaciones pero constantes diferentes.

Cada columna en *matrizConst* debe contener las constantes para un sistema de ecuaciones. Cada columna en la matriz resultante contiene la solución para el sistema correspondiente.

Solucionar:

$x + 2y = 1$

$3x + 4y = -1$

$x + 2y = 2$	
$3x + 4y = -3$	
$\text{simult}\left(\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, \begin{bmatrix} 1 & 2 \\ -1 & -3 \end{bmatrix}\right)$	$\begin{bmatrix} -3 & -7 \\ 2 & \frac{9}{2} \\ 2 & 2 \end{bmatrix}$

Para el primer sistema, $x=-3$ y $y=2$. Para el segundo sistema, $x=-7$ y $y=9/2$.

Expr ▶sin

Nota: Usted puede insertar este operador desde el teclado de la computadora al escribir sin .

Representa *Expr* en términos de seno. Este es un operador de conversión de despliegue. Se puede usar únicamente al final de la línea de ingreso.

▶sin reduce todas las potencias de $\cos(\dots)$ módulo $1 - \sin(\dots)^2$ de manera que cualquier potencia restante de $\sin(\dots)$ tiene exponentes en el rango (0, 2). Entonces, el resultado estará libre de $\cos(\dots)$ si y sólo si $\cos(\dots)$ ocurre en la expresión dada únicamente para potencias iguales.

Nota: Este operador de conversión no está soportado en los modos de Ángulo en Grados o Gradianes. Antes de usarlo, asegúrese de que el modo de Ángulo está configurado a Radianes y que *Expr* no contiene referencias explícitas para ángulos en grados o gradianes.

$$(\cos(x))^2 \blacktriangleright \sin \quad 1 - (\sin(x))^2$$

sin() (sen)



sin(*Expr1*) \Rightarrow expresión

sin(*Lista1*) \Rightarrow lista

sin(*Expr1*) entrega el seno del argumento como una expresión.

sin(*Lista1*) entrega una lista de senos de todos los elementos en *Lista1*.

Nota: El argumento se interpreta como un ángulo en grados, gradianes o radianes, de acuerdo con el modo del ángulo actual. Usted puede usar $^{\circ}$, $^{\text{G}}$ o $^{\text{r}}$ para anular la configuración del modo de ángulo en forma temporal.

En modo de ángulo en Grados:

$$\sin\left(\frac{\pi}{4}\right) \quad \frac{\sqrt{2}}{2}$$

$$\sin(45) \quad \frac{\sqrt{2}}{2}$$

$$\sin(\{0,60,90\}) \quad \left\{0, \frac{\sqrt{3}}{2}, 1\right\}$$

En modo de ángulo en Gradianes:

$$\sin(50) \quad \frac{\sqrt{2}}{2}$$

En modo de ángulo en Radianes:

$$\sin\left(\frac{\pi}{4}\right) \quad \frac{\sqrt{2}}{2}$$

$$\sin(45^{\circ}) \quad \frac{\sqrt{2}}{2}$$

sin(*matrizCuadrada1*) \Rightarrow *matrizCuadrada*

Entrega el seno de la matriz de *matrizCuadrada1*. Esto no es lo mismo que calcular el seno de cada elemento. Para obtener información acerca del método de cálculo, consulte **cos()**.

matrizCuadrada1 debe ser diagonalizable. El resultado siempre contiene números de punto flotante.

En modo de ángulo en Radianes:

$$\sin\left(\begin{bmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{bmatrix}\right) \quad \begin{bmatrix} 0.9424 & -0.04542 & -0.031999 \\ -0.045492 & 0.949254 & -0.020274 \\ -0.048739 & -0.00523 & 0.961051 \end{bmatrix}$$

sin⁻¹() (sen⁻¹)**sin⁻¹(Expr1)** ⇒ expresión**sin⁻¹(Lista1)** ⇒ lista**sin⁻¹(Expr1)** entrega el ángulo cuyo seno es Expr1 como una expresión.**sin⁻¹(Lista1)** entrega una lista de senos inversos de cada elemento de Lista1.**Nota:** El resultado se entrega como un ángulo en grados, gradianes o radianes, de acuerdo con la configuración del modo del ángulo actual.**Nota:** Usted puede insertar esta función desde el teclado al escribir **arcosen (...)**.**sin⁻¹(matrizCuadrada1)** ⇒ matrizCuadradaEntrega el seno inverso de la matriz de *matrizCuadrada1*. Esto no es lo mismo que calcular el seno inverso de cada elemento. Para obtener información acerca del método de cálculo, consulte **cos()**.*matrizCuadrada1* debe ser diagonalizable. El resultado siempre contiene números de punto flotante.

En modo de ángulo en Grados:

$$\sin^{-1}(1) \quad 90$$

En modo de ángulo en Gradianes:

$$\sin^{-1}(1) \quad 100$$

En modo de ángulo en Radianes:

$$\sin^{-1}\{0,0,2,0,5\} \quad \{0,0,201358,0,523599\}$$

En el modo de ángulo en Radianes y el modo de formato complejo Rectangular:

$$\sin^{-1}\begin{pmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{pmatrix} \begin{bmatrix} -0.164058-0.064606 \cdot i & 1.49086-2.1051 \cdot i \\ 0.725533-1.51594 \cdot i & 0.947305-0.77830 \cdot i \\ 2.08316-2.63205 \cdot i & -1.79018+1.2718 \cdot i \end{bmatrix}$$

Para ver el resultado completo, presione **▲** y luego use **◀▶** para mover el cursor.**sinh() (senh)****sinh(Expr1)** ⇒ expresión**sinh(Lista1)** ⇒ lista**sinh(Expr1)** entrega el seno hiperbólico del argumento como una expresión.**sinh(Lista1)** entrega una lista de los senos hiperbólicos de cada elemento de Lista1.**sinh(matrizCuadrada1)** ⇒ matrizCuadradaEntrega el seno hiperbólico de la matriz de *matrizCuadrada1*. Esto no es lo mismo que calcular el seno hiperbólico de cada elemento. Para obtener información acerca del método de cálculo, consulte **cos()**.*matrizCuadrada1* debe ser diagonalizable. El resultado siempre contiene números de punto flotante.

$$\sinh(1.2) \quad 1.50946$$

$$\sinh\{0,1,2,3\} \quad \{0,1.50946,10.0179\}$$

En modo de ángulo en Radianes:

$$\sinh\begin{pmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{pmatrix} \begin{bmatrix} 360.954 & 305.708 & 239.604 \\ 352.912 & 233.495 & 193.564 \\ 298.632 & 154.599 & 140.251 \end{bmatrix}$$

sinh⁻¹() (senh⁻¹)**sinh⁻¹(Expr1)** ⇒ expresión**sinh⁻¹(Lista1)** ⇒ lista**sinh⁻¹(Expr1)** entrega el seno hiperbólico inverso del argumento como una expresión.**sinh⁻¹(Lista1)** entrega una lista de los senos hiperbólicos inversos de cada elemento de Lista1.**Nota:** Usted puede insertar esta función desde el teclado al escribir **arcosenh (...)**.

$$\sinh^{-1}(0) \quad 0$$

$$\sinh^{-1}\{0,2,1,3\} \quad \{0,1.48748,\sinh^{-1}(3)\}$$

sinh⁻¹() (senh⁻¹)

Catálogo >

sinh⁻¹(matrizCuadrada1) ⇒ matrizCuadrada

Entrega el seno hiperbólico inverso de la matriz de *matrizCuadrada1*. Esto no es lo mismo que calcular el seno hiperbólico inverso de cada elemento. Para obtener información acerca del método de cálculo, consulte **cos()**.

matrizCuadrada1 debe ser diagonalizable. El resultado siempre contiene números de punto flotante.

En modo de ángulo en Radianes:

$$\sinh^{-1} \begin{pmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{pmatrix} = \begin{bmatrix} 0.041751 & 2.15557 & 1.1582 \\ 1.46382 & 0.926568 & 0.112557 \\ 2.75079 & -1.5283 & 0.57268 \end{bmatrix}$$

SinReg

Catálogo >

SinReg *X*, *Y* [, [*Iteraciones*], [*Periodo*] [, [*Categoría*, *Incluir*]]

Resuelve la regresión sinusoidal en las listas *X* y *Y*. Se almacena un resumen de resultados en la variable *stat.results*. (Vea la página 122.)

Todas las listas deben tener una dimensión igual, excepto por *Incluir*.

X y *Y* son listas de variables independientes y dependientes.

Iteraciones es un valor que especifica el número máximo de veces (1 a 16) que se intentará una solución. Si se omite, se usa 8. Por lo general, los valores más grandes dan como resultado una mejor exactitud, pero tiempos de ejecución más largos, y viceversa.

Periodo especifica un periodo estimado. Si se omite, la diferencia entre los valores en *X* deberán ser iguales y estar en orden secuencial. Si usted especifica el *Periodo*, las diferencias entre los valores *x* pueden ser desiguales.

Categoría es una lista de códigos de categoría para los datos *X* y *Y* correspondientes.

Incluir es una lista de uno o más códigos de categoría. Sólo aquellos elementos de datos cuyo código de categoría está incluido en esta lista están incluidos en el cálculo.

El resultado de **SinReg** siempre es en radianes, independientemente de la configuración del modo de ángulo.

Para obtener información sobre el efecto de los elementos vacíos en una lista, vea "Elementos vacíos (inválidos)" en la página 166.

Variable de salida	Descripción
stat.EcnReg	Ecuación de Regresión: $a \cdot \sin(bx+c)+d$
stat.a, stat.b, stat.c, stat.d	Coefficientes de regresión
stat.Resid	Residuales de la regresión
stat.XReg	La lista de puntos de datos en la <i>Lista X</i> modificada que se usa en realidad en la regresión con base en las restricciones de las Categorías <i>Frec</i> , <i>Lista de Categorías Incluir</i>
stat.YReg	La lista de puntos de datos en la <i>Lista Y</i> modificada que se usa en realidad en la regresión con base en las restricciones de las Categorías <i>Frec</i> , <i>Lista de Categorías Incluir</i>
stat.FrecReg	Lista de frecuencias correspondientes a <i>stat.XReg</i> y <i>stat.YReg</i>

solve(Ecuación, Var) \Rightarrow expresión Booleana

solve(Ecuación, Var=Cálculo) \Rightarrow expresión Booleana

solve(Desigualdad, Var) \Rightarrow expresión Booleana

Entrega soluciones reales posibles de una ecuación o una desigualdad para Var. La meta es producir posibles soluciones. Sin embargo, podría haber ecuaciones o desigualdades para las cuales el número de soluciones es infinito.

Las posibles soluciones podrían no ser soluciones finitas reales para algunas combinaciones de valores para variables indefinidas.

Para la configuración de Auto del modo **Auto o Aproximado**, la meta es producir soluciones exactas cuando son concisas, y complementadas por medio de búsquedas iterativas con aritmética aproximada cuando las soluciones exactas son imprácticas.

Debido a la cancelación predeterminada del máximo común divisor del numerador y el denominador de las proporciones, las soluciones podrían ser soluciones sólo en el límite de uno o ambos lados.

Para las desigualdades de los tipos \geq , \leq , $<0 >$, las soluciones explícitas son improbables, a menos que la desigualdad sea lineal y que contenga sólo Var.

Para el modo Exacto, las porciones que no se pueden solucionar se entregan como una ecuación o desigualdades implícita.

Utilice el operador restrictivo ("|") para restringir el intervalo de solución u otras variables que se dan en la ecuación o desigualdad. Cuando encuentre una solución en un intervalo, puede utilizar los operadores de desigualdad para excluir dicho intervalo de búsquedas futuras.

se entrega falso cuando no se encuentra ninguna solución real. Se entrega verdadero si **solve()** puede determinar que cualquier valor real finito de Var satisface la ecuación o desigualdad.

Dado que **solve()** siempre entrega un resultado Booleano, usted puede usar "and", "or" y "not" para combinar los resultados de **solve()** entre sí o con otras expresiones Booleanas.

Las soluciones podrían contener una constante indefinida nueva única de la forma nj , donde j es un entero en el intervalo 1–255. Dichas variables designan un entero arbitrario.

En el modo Real, las potencias fraccionarias que tienen denominadores impares indican sólo una rama real. De otra manera, varias expresiones ramificadas como las potencias fraccionarias, los logaritmos y las funciones trigonométricas inversas indican sólo una rama principal. En consecuencia, **solve()** produce sólo las soluciones correspondientes a esa rama real o principal.

Nota: Vea también **cSolve()**, **cZeros()**, **nSolve()**, y **zeros()**.

$$\text{solve}(a \cdot x^2 + b \cdot x + c = 0, x)$$

$$x = \frac{\sqrt{b^2 - 4 \cdot a \cdot c} - b}{2 \cdot a} \text{ or } x = \frac{-\sqrt{b^2 - 4 \cdot a \cdot c} - b}{2 \cdot a}$$

$$\text{Ans}|a=1 \text{ and } b=1 \text{ and } c=1$$

$$x = \frac{-1 + \sqrt{3}}{2} \cdot i \text{ or } x = \frac{-1 - \sqrt{3}}{2} \cdot i$$

$$\text{solve}((x-a) \cdot e^x = x \cdot (x-a), x)$$

$$x=a \text{ or } x=0.567143$$

$$(x+1) \cdot \frac{x-1}{x-1} + x-3$$

$$2 \cdot x-2$$

$$\text{solve}(5 \cdot x-2 \geq 2 \cdot x, x)$$

$$x \geq \frac{2}{3}$$

$$\text{exact}(\text{solve}((x-a) \cdot e^x = x \cdot (x-a), x))$$

$$e^x + x = 0 \text{ or } x = a$$

En modo de ángulo en Radianes:

$$\text{solve}\left(\tan(x) = \frac{1}{x}, x\right) | x > 0 \text{ and } x < 1$$

$$x = 0.860334$$

$$\text{solve}(x = x + 1, x)$$

false

$$\text{solve}(x = x, x)$$

true

$$2 \cdot x - 1 \leq 1 \text{ and solve}(x^2 \neq 9, x)$$

$$x \neq -3 \text{ and } x \leq 1$$

En modo de ángulo en Radianes:

$$\text{solve}(\sin(x) = 0, x)$$

$$x = n1 \cdot \pi$$

$$\text{solve}\left(x^{\frac{1}{3}} = -1, x\right)$$

$$x = -1$$

$$\text{solve}(\sqrt{x} = 2, x)$$

false

$$\text{solve}(\sqrt{x} = -2, x)$$

$$x = 4$$

solve(*Ecn1* and *Ecn2* [and ...], *VarOCálculo1*,

VarOCálculo2 [, ...]) ⇒ expresión Booleana

solve(*SistemaDeEcns*, *VarOCálculo1*,

VarOCálculo2 [, ...]) ⇒ expresión Booleana

solve({*Ecn1*, *Ecn2* [,...]} {*VarOCálculo1*, *VarOCálculo2* [, ...]})

⇒ expresión Booleana

Entrega posibles soluciones reales para las ecuaciones algebraicas simultáneas, donde cada *VarOCálculo* especifica una variable que usted desea solucionar.

Usted puede separar las ecuaciones con el operador **and** o puede ingresar un *SistemaDeEcns* al usar una plantilla del Catálogo. El número de argumentos *VarOCálculo* debe coincidir con el número de ecuaciones. De manera opcional, se puede especificar un cálculo inicial para una variable. Cada *VarOCálculo* debe tener la forma:

variable

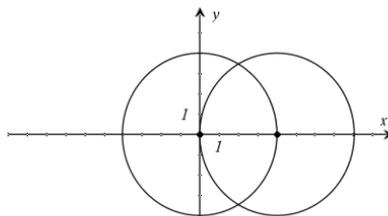
- 0 -

variable = número real o no real

Por ejemplo, *x* es válida y también lo es *x=3*.

Si todas las ecuaciones son polinomios y usted NO especifica cualquier cálculo inicial, **solve()** usa el método de eliminación de léxico Gröbner/Buchberger para intentar determinar todas las soluciones reales.

Por ejemplo, supongamos que usted tiene un círculo de radio *r* en el origen y otro círculo de radio *r* centrado donde el primer círculo cruza el eje *x* positivo. Use **solve()** para encontrar las intersecciones.



Conforme se ilustra con *r* en el ejemplo de la derecha, las ecuaciones polinómicas simultáneas pueden tener variables extras que no tienen ningún valor, aunque representan valores numéricos dados que podrían sustituirse más adelante.

$$\text{solve}\left(x^2+y^2=r^2 \text{ and } (x-r)^2+y^2=r^2, \{x,y\}\right)$$

$$x=\frac{r}{2} \text{ and } y=\frac{\sqrt{3}\cdot r}{2} \text{ or } x=\frac{r}{2} \text{ and } y=\frac{-\sqrt{3}\cdot r}{2}$$

Usted también (o en lugar de) puede incluir variables de solución que no aparecen en las ecuaciones. Por ejemplo, usted puede incluir *z* como una variable de solución para extender el ejemplo anterior a dos cilindros intersectados paralelos del radio *r*.

$$\text{solve}\left(x^2+y^2=r^2 \text{ and } (x-r)^2+y^2=r^2, \{x,y,z\}\right)$$

$$x=\frac{r}{2} \text{ and } y=\frac{\sqrt{3}\cdot r}{2} \text{ and } z=c1 \text{ or } x=\frac{r}{2} \text{ and } y=$$

Estas soluciones de cilindros ilustran cómo las familias de soluciones podrían contener constantes arbitrarias de la forma *ck*, donde *k* es un sufijo de entero desde 1 hasta 255.

Para ver el resultado completo, presione  y luego use  para mover el cursor.

Para sistemas polinómicos, el tiempo de cálculo o el agotamiento de memoria pueden depender ampliamente del orden en el cual se enumeran las variables de solución. Si su elección inicial agota la memoria o su paciencia, intente volver a arreglar las variables en las ecuaciones y/o en la lista *varOCálculo*.

Si usted no incluye ningún cálculo y si cualquier ecuación no es polinómica en cualquier variable, pero todas las ecuaciones son lineales en todas las variables de solución, **solve()** usa la eliminación Gaussiana para tratar de determinar todas las soluciones reales.

$$\text{solve}\left(x+e^z\cdot y=1 \text{ and } x-y=\sin(z), \{x,y\}\right)$$

$$x=\frac{e^z\cdot \sin(z)+1}{e^z+1} \text{ and } y=\frac{-(\sin(z)-1)}{e^z+1}$$

solve() (solucion)

Catálogo >

Si un sistema no es ni polinómico en todas sus variables ni lineal en sus variables de solución, **solve()** determina como máximo una solución usando un método iterativo aproximado. Para hacer esto, el número de variables de solución debe igualar el número de ecuaciones, y todas las demás variables en las ecuaciones deben simplificarse a números.

Cada variable de solución comienza en su valor calculado si hay uno; de otro modo, comienza en 0.0.

Use cálculos para buscar las soluciones adicionales de una en una. Por convergencia, un cálculo puede tener que ser más bien cercano a una solución.

$$\text{solve}(e^z \cdot y = 1 \text{ and } -y = \sin(z), \{y, z\})$$

$$y = 2.812E-10 \text{ and } z = 21.9911 \text{ or } y = 0.001871$$

Para ver el resultado completo, presione \blacktriangle y luego use \blacktriangleleft y \blacktriangleright para mover el cursor.

$$\text{solve}(e^z \cdot y = 1 \text{ and } -y = \sin(z), \{y, z = 2 \cdot \pi\})$$

$$y = 0.001871 \text{ and } z = 6.28131$$

SortA (OrdenarA)

Catálogo >

SortA *Lista1* [, *Lista2*] [, *Lista3*] ...**SortA** *Vector1* [, *Vector2*] [, *Vector3*] ...

Ordena los elementos del primer argumento en orden ascendente.

Si usted incluye argumentos adicionales, ordena los elementos de cada uno, de manera que sus nuevas posiciones coinciden con las nuevas posiciones de los elementos en el primer argumento.

Todos los argumentos deben ser nombres de listas o vectores. Todos los argumentos deben tener dimensiones iguales.

Los elementos vacíos (inválidos) dentro del primer argumento se mueven a la parte inferior. Para obtener más información sobre elementos vacíos, vea la página 166.

$\{2, 1, 4, 3\} \rightarrow list1$	$\{2, 1, 4, 3\}$
SortA <i>list1</i>	Done
<i>list1</i>	$\{1, 2, 3, 4\}$
$\{4, 3, 2, 1\} \rightarrow list2$	$\{4, 3, 2, 1\}$
SortA <i>list2, list1</i>	Done
<i>list2</i>	$\{1, 2, 3, 4\}$
<i>list1</i>	$\{4, 3, 2, 1\}$

SortD (OrdenarD)

Catálogo >

SortD *Lista1* [, *Lista2*] [, *Lista3*] ...**SortD** *Vector1* [, *Vector2*] [, *Vector3*] ...

Idéntico a **SortA**, excepto que **SortD** ordena los elementos en orden descendente.

Los elementos vacíos (inválidos) dentro del primer argumento se mueven a la parte inferior. Para obtener más información sobre elementos vacíos, vea la página 166.

$\{2, 1, 4, 3\} \rightarrow list1$	$\{2, 1, 4, 3\}$
$\{1, 2, 3, 4\} \rightarrow list2$	$\{1, 2, 3, 4\}$
SortD <i>list1, list2</i>	Done
<i>list1</i>	$\{4, 3, 2, 1\}$
<i>list2</i>	$\{3, 4, 1, 2\}$

►Sphere (►Esfera)

Catálogo >

Véctor ►Sphere

Nota: Usted puede insertar este operador desde el teclado de la computadora al escribir @>Sphere.

Despliega el vector de fila o columna en forma esférica [$\rho \angle \theta \angle \phi$].

Véctor debe ser de dimensión 3 y puede ser un vector de fila o de columna.

Nota: ►Sphere es una instrucción de formato de despliegue, no una función de conversión. Usted puede usarla sólo al final de una línea de ingreso.

Presione **Ctrl+Ingresar** (Macintosh®):

⌘+Ingresar para evaluar:

$$\left[\begin{array}{ccc} 1 & 2 & 3 \end{array} \right] \text{►Sphere} \\ \left[3.74166 \quad \angle 1.10715 \quad \angle 0.640522 \right]$$

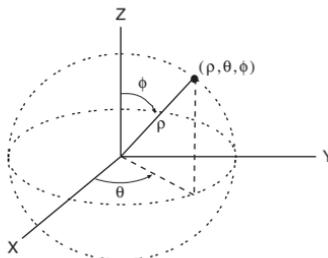
Presione **Ctrl+Ingresar** (Macintosh®):

⌘+Ingresar para evaluar:

$$\left(\begin{array}{ccc} 2 & \angle \frac{\pi}{4} & 3 \end{array} \right) \text{►Sphere} \\ \left[3.60555 \quad \angle 0.785398 \quad \angle 0.588003 \right]$$

Presione

$$\left(\begin{array}{ccc} 2 & \angle \frac{\pi}{4} & 3 \end{array} \right) \text{►Sphere} \\ \left[\sqrt{13} \quad \angle \frac{\pi}{4} \quad \angle \cos^{-1} \left(\frac{3 \cdot \sqrt{13}}{13} \right) \right]$$



sqrt()

Catálogo >

sqrt(Expr1) ⇒ expresión

sqrt(Lista1) ⇒ lista

Entrega la raíz cuadrada del argumento.

Para una lista, entrega las raíces cuadradas de todos los elementos en Lista1.

Nota: Vea también **Plantilla de raíz cuadrada**, página 1.

$$\sqrt{4} \quad 2 \\ \sqrt{\{9,a,4\}} \quad \{3,\sqrt{a},2\}$$

stat.results

Despliega los resultados de un cálculo de estadísticas.

Los resultados se despliegan como un conjunto de pares de valores de nombres. Los nombres específicos que se muestran dependen de la función o del comando de estadísticas evaluado de manera más reciente.

Usted puede copiar un nombre o valor y pegarlo en otras ubicaciones.

Nota: Evite definir variables que usan los mismos nombres que aquellos que se usan para análisis estadístico. En algunos casos, podría ocurrir una condición de error. Los nombres de variable que se usan para análisis estadístico se enumeran en la tabla de abajo.

$$xlist = \{1, 2, 3, 4, 5\} \quad \{1, 2, 3, 4, 5\}$$

$$ylist = \{4, 8, 11, 14, 17\} \quad \{4, 8, 11, 14, 17\}$$

LinRegMx *xlist, ylist, 1: stat.results*

"Title"	"Linear Regression (mx+b)"
"RegEqn"	"m*x+b"
"m"	3.2
"b"	1.2
"r ² "	0.996109
"r"	0.998053
"Resid"	" {... } "

<i>stat.values</i>	"Linear Regression (mx+b)"
	"m*x+b"
	3.2
	1.2
	0.996109
	0.998053
	" {-0.4, 0.4, 0.2, 0., -0.2} "

stat.a
stat.AdjR²
stat.b
stat.b0
stat.b1
stat.b2
stat.b3
stat.b4
stat.b5
stat.b6
stat.b7
stat.b8
stat.b9
stat.b10
stat.bList
stat.χ²
stat.c
stat.CLower
stat.CLowerList
stat.CompList
stat.CompMatrix
stat.CookDist
stat.CUpper
stat.CUpperList
stat.d

stat.dFDenom
stat.dFBlock
stat.dFCol
stat.dFError
stat.dFInteract
stat.dFReg
stat.dFNumer
stat.dFRow
stat.DW
stat.e
stat.ExpMatrix
stat.F
stat.FBlock
stat.Fcol
stat.FInteract
stat.FreqReg
stat.Frow
stat.Leverage
stat.LowerPred
stat.LowerVal
stat.m
stat.MaxX
stat.MaxY
stat.ME
stat.MedianX

stat.MedianY
stat.MEPred
stat.MinX
stat.MinY
stat.MS
stat.MSBlock
stat.MSCol
stat.MSError
stat.MSInteract
stat.MSReg
stat.MSRow
stat.n
stat.β
stat.β₁
stat.β₂
stat.βDiff
stat.PList
stat.PVal
stat.PValBlock
stat.PValCol
stat.PValInteract
stat.PValRow
stat.Q1X
stat.Q1Y
stat.Q3X

stat.Q3Y
stat.r
stat.r²
stat.RegEqn
stat.Resid
stat.ResidTrans
stat.σx
stat.σy
stat.σx1
stat.σx2
stat.Σx
stat.Σx²
stat.Σxy
stat.Σy
stat.Σy²
stat.s
stat.SE
stat.SEList
stat.SEPred
stat.sResid
stat.SEslope
stat.sp
stat.SS
stat.SSBlock

stat.SSCol
stat.SSX
stat.SSY
stat.SSError
stat.SSInteract
stat.SSReg
stat.SSRow
stat.tList
stat.UpperPred
stat.UpperVal
stat.X̄
stat.X̄1
stat.X̄2
stat.X̄Diff
stat.X̄List
stat.XReg
stat.XVal
stat.XValList
stat.ȳ
stat.ŷ
stat.ŶList
stat.YReg

Nota: Cada vez que la aplicación de Listas y Hoja de Cálculo calcula resultados estadísticos, copia las variables del grupo "stat." a un grupo "stat#.", donde # es un número que se incrementa en forma automática. Esto le permite mantener los resultados anteriores mientras realiza varios cálculos.

stat.valuesVea el ejemplo de **stat.results**.

Despliega una matriz de los valores calculados para la función o el comando de estadísticas evaluado de manera más reciente.

A diferencia de **stat.results**, **stat.values** omite los nombres asociados con los valores.

Usted puede copiar un valor y pegarlo en otras ubicaciones.

stDevPop() (desvEstPob)

stDevPop(Lista[, listaFrec]) ⇒ expresión

Entrega la desviación estándar de población de los elementos en *Lista*.

Cada elemento de *listaFrec* cuenta el número de ocurrencias consecutivas del elemento correspondiente en *Lista*.

Nota: *Lista* debe tener al menos dos elementos. Los elementos vacíos (inválidos) se ignoran. Para obtener más información sobre elementos vacíos, vea la página 166.

stDevPop(Matriz1[, matrizFrec]) ⇒ matriz

Entrega un vector de fila de las desviaciones estándar de población las columnas en *Matriz1*.

Cada elemento de *matrizFrec* cuenta el número de ocurrencias consecutivas del elemento correspondiente en *Matriz1*.

Nota: *Matriz1* debe tener al menos dos filas. Los elementos vacíos (inválidos) se ignoran. Para obtener más información sobre elementos vacíos, vea la página 166.

En modos de ángulo en Radianes y auto:

$$\text{stDevPop}\left(\left\{a, b, c\right\}\right) = \frac{\sqrt{2 \cdot \left(a^2 - a \cdot (b+c) + b^2 - b \cdot c + c^2\right)}}{3}$$

$$\text{stDevPop}\left(\left\{1, 2, 5, -6, 3, -2\right\}\right) = \frac{\sqrt{465}}{6}$$

$$\text{stDevPop}\left(\left\{1, 3, 2, 5, -6, 4\right\}, \left\{3, 2, 5\right\}\right) = 4.11107$$

$$\text{stDevPop}\left(\begin{bmatrix} 1 & 2 & 5 \\ -3 & 0 & 1 \\ 5 & 7 & 3 \end{bmatrix}, \begin{bmatrix} 4 \cdot \sqrt{6} & \sqrt{78} & 2 \cdot \sqrt{6} \\ 3 & 3 & 3 \end{bmatrix}\right) = \begin{bmatrix} 2.52608 & 5.21506 \end{bmatrix}$$

stDevSamp() (desvEstMuest)

stDevSamp(Lista[, listaFrec]) ⇒ expresión

Entrega la desviación estándar muestra de los elementos en *Lista*.

Cada elemento de *listaFrec* cuenta el número de ocurrencias consecutivas del elemento correspondiente en *Lista*.

Nota: *Lista* debe tener al menos dos elementos. Los elementos vacíos (inválidos) se ignoran. Para obtener más información sobre elementos vacíos, vea la página 166.

$$\text{stDevSamp}\left(\left\{a, b, c\right\}\right) = \frac{\sqrt{3 \cdot \left(a^2 - a \cdot (b+c) + b^2 - b \cdot c + c^2\right)}}{3}$$

$$\text{stDevSamp}\left(\left\{1, 2, 5, -6, 3, -2\right\}\right) = \frac{\sqrt{62}}{2}$$

$$\text{stDevSamp}\left(\left\{1, 3, 2, 5, -6, 4\right\}, \left\{3, 2, 5\right\}\right) = 4.33345$$

stDevSamp() (desvEstMuest)

Catálogo >

stDevSamp(*MatrizI*, *matrizFrec*) ⇒ *matriz*Entrega un vector de fila de las desviaciones estándar muestra de las columnas en *MatrizI*.Cada elemento de *matrizFrec* cuenta el número de ocurrencias consecutivas del elemento correspondiente en *MatrizI*.**Nota:** *MatrizI* debe tener al menos dos filas. Los elementos vacíos (inválidos) se ignoran. Para obtener más información sobre elementos vacíos, vea la página 166.

stDevPop	$\begin{bmatrix} 1 & 2 & 5 \\ -3 & 0 & 1 \\ 5 & 7 & 3 \end{bmatrix}$	$[3.26599 \quad 2.94392 \quad 1.63299]$
stDevPop	$\begin{bmatrix} -1.2 & 5.3 \\ 2.5 & 7.3 \\ 6 & -4 \end{bmatrix}, \begin{bmatrix} 4 & 2 \\ 3 & 3 \\ 1 & 7 \end{bmatrix}$	$[2.52608 \quad 5.21506]$

Stop (Detener)

Catálogo >

Stop

Comando de programación: Termina el programa.

Stop no está permitido en las funciones.**Nota para ingresar el ejemplo:** En la aplicación de Calculadora en el dispositivo portátil, usted puede ingresar definiciones en líneas múltiples al presionar en lugar de al final de cada línea. En el teclado de la computadora, presione y sostenga **Alt** y presione **Ingresar**.

<i>i</i> :=0	0
Define <i>prog1</i> ()=Prgm	<i>Done</i>
For <i>i</i> ,1,10,1	
If <i>i</i> =5	
Stop	
EndFor	
EndPrgm	
<i>prog1</i> ()	<i>Done</i>
<i>i</i>	5

Almacenar

Vea → (almacenar), página 164.

string() (cadena)

Catálogo >

string(*Expr*) ⇒ *cadena*Simplifica *Expr* y entrega el resultado de una cadena de caracteres.

string(1.2345)	"1.2345"
string(1+2)	"3"
string(cos(x)+√3)	"cos(x)+√(3)"

subMat()

Catálogo >

subMat(*MatrizI*, *iniciarFila*, *iniciarCol*, *terminarFila*, *terminarCol*)
⇒ *matriz*Entrega la submatriz especificada de *MatrizI*.Predeterminados: *iniciarFila*=1, *iniciarCol*=1, *terminarFila*=última fila, *terminarCol*=última columna.

$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \rightarrow m1$	$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$
subMat(<i>m1</i> ,2,1,3,2)	$\begin{bmatrix} 4 & 5 \\ 7 & 8 \end{bmatrix}$
subMat(<i>m1</i> ,2,2)	$\begin{bmatrix} 5 & 6 \\ 8 & 9 \end{bmatrix}$

Suma (Sigma)Vea $\Sigma()$, página 157.

sum()

Catálogo >

sum(Lista[, Iniciar[, Terminar]]) ⇒ expresiónEntrega la suma de todos los elementos en *Lista*.*Inicio* y *Término* son opcionales. Especifican un rango de elementos.Cualquier argumento inválido produce un resultado inválido. Los elementos vacíos (inválidos) en *Lista* se ignoran. Para obtener más información sobre elementos vacíos, vea la página 166.

$\text{sum}\left(\{1,2,3,4,5\}\right)$	15
$\text{sum}\left(\{a,2 \cdot a,3 \cdot a\}\right)$	$6 \cdot a$
$\text{sum}\left(\text{seq}(n,n,1,10)\right)$	55
$\text{sum}\left(\{1,3,5,7,9\},3\right)$	21

sum(Matriz[, Iniciar[, Terminar]]) ⇒ matrizEntrega un vector de fila que contiene las sumas de todos los elementos en las columnas de *Matriz1*.*Inicio* y *Término* son opcionales. Especifican un rango de filas.Cualquier argumento inválido produce un resultado inválido. Los elementos vacíos (inválidos) en *Matriz1* se ignoran. Para obtener más información sobre elementos vacíos, vea la página 166.

$\text{sum}\left(\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}\right)$	$[5 \ 7 \ 9]$
$\text{sum}\left(\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}\right)$	$[12 \ 15 \ 18]$
$\text{sum}\left(\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix},2,3\right)$	$[11 \ 13 \ 15]$

sumIf() (sumaSi)

Catálogo >

sumIf(Lista,Criterios[, ListaSuma]) ⇒ valorEntrega la suma acumulada de todos los elementos en *Lista* que cumplen con los *Criterios* especificados. De manera opcional, usted puede especificar una lista alterna, *listaSuma*, para proporcionar los elementos a acumular.*Lista* puede ser una expresión, lista o matriz. *ListaSuma*, si se especifica, debe tener la(s) misma(s) dimensión(es) que *Lista*.*Los criterios* pueden ser:

- Un valor, una expresión o una cadena. Por ejemplo, **34** acumula sólo aquellos elementos en *Lista* que se simplifican al valor 34.
- Una expresión Booleana que contiene el símbolo **?** como un marcador de posición para cada elemento. Por ejemplo, **?<10** acumula sólo aquellos elementos en *Lista* que son menos de 10.

Cuando un elemento de *Lista* cumple con los *Criterios*, el elemento se agrega a la suma acumulativa. Si usted incluye *listaSuma*, el elemento correspondiente de *listaSuma* se agrega a la suma en su lugar.Dentro de la aplicación de Listas y Hoja de Cálculo, usted puede usar un rango de celdas en lugar de *Lista* y *listaSuma*.

Los elementos vacíos (inválidos) se ignoran. Para obtener más información sobre elementos vacíos, vea la página 166.

Nota: Vea también **countIf()**, página 27.

$\text{sumIf}\left(\{1,2,e,3,\pi,4,5,6\},2.5<?<4.5\right)$	$e+\pi+7$
$\text{sumIf}\left(\{1,2,3,4\},2<?<5,\{10,20,30,40\}\right)$	70

secSuma()Vea $\Sigma()$, página 157.**system()**

Catálogo >

system(Ecn1 [, Ecn2 [, Ecn3 [, ...]])**system**(Expr1 [, Expr2 [, Expr3 [, ...]])

Entrega un sistema de ecuaciones, formateado como una lista. Usted también puede crear un sistema al usar una plantilla.

Nota: Vea también **Sistema de ecuaciones**, página 3.

$\text{solve}\left(\begin{cases} x+y=0 \\ x-y=8 \end{cases},x,y\right)$	$x=4$ and $y=-4$
---	------------------

T

T (trasponer)

Catálogo > 

$MatrizI^T \Rightarrow$ matriz

Entrega el traspuesto conjugado complejo de $MatrizI$.

Nota: Usted puede insertar este operador desde el teclado de la computadora al escribir @t.

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}^T \quad \begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{bmatrix}$$

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}^T \quad \begin{bmatrix} a & c \\ b & d \end{bmatrix}$$

$$\begin{bmatrix} 1+i & 2+i \\ 3+i & 4+i \end{bmatrix}^T \quad \begin{bmatrix} 1-i & 3-i \\ 2-i & 4-i \end{bmatrix}$$

tan()

 tecla

$\tan(\text{Expr1}) \Rightarrow$ expresión

$\tan(\text{Lista1}) \Rightarrow$ lista

$\tan(\text{Expr1})$ entrega la tangente del argumento como una expresión.

$\tan(\text{Lista1})$ entrega una lista de las tangentes de todos los elementos en Lista1 .

Nota: El argumento se interpreta como un ángulo en grados, gradianes o radianes, de acuerdo con el modo del ángulo actual. Usted puede usar $^{\circ}$, $^{\text{G}}$ o $^{\text{r}}$ para anular la configuración del modo de ángulo en forma temporal.

En modo de ángulo en Grados:

$$\tan\left(\frac{\pi}{4}\right) \quad 1$$

$$\tan(45) \quad 1$$

$$\tan(\{0,60,90\}) \quad \{0,\sqrt{3},\text{undef}\}$$

En modo de ángulo en Gradianes:

$$\tan\left(\frac{\pi}{4}\right) \quad 1$$

$$\tan(50) \quad 1$$

$$\tan(\{0,50,100\}) \quad \{0,1,\text{undef}\}$$

En modo de ángulo en Radianes:

$$\tan\left(\frac{\pi}{4}\right) \quad 1$$

$$\tan(45^{\circ}) \quad 1$$

$$\tan\left(\left\{\pi, \frac{\pi}{3}, \pi, \frac{\pi}{4}\right\}\right) \quad \{0,\sqrt{3},0,1\}$$

$\tan(\text{matrizCuadrada1}) \Rightarrow$ matrizCuadrada

Entrega la tangente de la matriz de matrizCuadrada1 . Esto no es lo mismo que calcular la tangente de cada elemento. Para obtener información acerca del método de cálculo, consulte **cos()**.

matrizCuadrada1 debe ser diagonalizable. El resultado siempre contiene números de punto flotante.

En modo de ángulo en Radianes:

$$\tan\left(\begin{bmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{bmatrix}\right) \quad \begin{bmatrix} -28.2912 & 26.0887 & 11.1142 \\ 12.1171 & -7.83536 & -5.48138 \\ 36.8181 & -32.8063 & -10.4594 \end{bmatrix}$$

tan⁻¹()**tan⁻¹(Expr1)** ⇒ expresión**tan⁻¹(Lista1)** ⇒ lista**tan⁻¹(Expr1)** entrega el ángulo cuya tangente es *Expr1* como una expresión.**tan⁻¹(Lista1)** entrega una lista de las tangentes inversas de cada elemento de *Lista1*.**Nota:** El resultado se entrega como un ángulo en grados, gradianes o radianes, de acuerdo con la configuración del modo del ángulo actual.**Nota:** Usted puede insertar esta función desde el teclado al escribir **arccotan** (...).**tan⁻¹(matrizCuadrada1)** ⇒ *matrizCuadrada*Entrega la tangente inversa de la matriz de *matrizCuadrada1*. Esto no es lo mismo que calcular la tangente inversa de cada elemento. Para obtener información acerca del método de cálculo, consulte **cos()**.*matrizCuadrada1* debe ser diagonalizable. El resultado siempre contiene números de punto flotante.

En modo de ángulo en Grados:

$$\tan^{-1}(1) \quad 45$$

En modo de ángulo en Gradianes:

$$\tan^{-1}(1) \quad 50$$

En modo de ángulo en Radianes:

$$\tan^{-1}\{0,0.2,0.5\} \quad \{0,0.197396,0.463648\}$$

En modo de ángulo en Radianes:

$$\tan^{-1}\left(\begin{bmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{bmatrix}\right) \quad \begin{bmatrix} -0.083658 & 1.26629 & 0.62263 \\ 0.748539 & 0.630015 & -0.070012 \\ 1.68608 & -1.18244 & 0.455126 \end{bmatrix}$$

tangentLine()**tangentLine(Expr1,Var,Punto)** ⇒ expresión**tangentLine(Expr1,Var=Punto)** ⇒ expresiónEntrega la línea tangente para la curva representada por *Expr1* en el punto especificado en *Var=Punto*.Asegúrese de que la variable independiente no está definida. Por ejemplo, Si $f(x)=5$ y $x=3$, entonces **tangentLine(f(x),x,2)** entrega "falso".

$$\text{tangentLine}(x^2,x,1) \quad 2 \cdot x - 1$$

$$\text{tangentLine}((x-3)^2-4,x=3) \quad -4$$

$$\text{tangentLine}\left(x\frac{1}{3},x=0\right) \quad x=0$$

$$\text{tangentLine}(\sqrt{x^2-4},x=2) \quad \text{undef}$$

$$x:=3: \text{tangentLine}(x^2,x,1) \quad 5$$

tanh()**tanh(Expr1)** ⇒ expresión**tanh(Lista1)** ⇒ lista**tanh(Expr1)** entrega la tangente hiperbólica del argumento como una expresión.**tanh(Lista1)** entrega una lista de las tangentes hiperbólicas de cada elemento de *Lista1*.**tanh(matrizCuadrada1)** ⇒ *matrizCuadrada*Entrega la tangente hiperbólica de la matriz de *matrizCuadrada1*. Esto no es lo mismo que calcular la tangente hiperbólica de cada elemento. Para obtener información acerca del método de cálculo, consulte **cos()**.*matrizCuadrada1* debe ser diagonalizable. El resultado siempre contiene números de punto flotante.

$$\tanh(1.2) \quad 0.833655$$

$$\tanh(\{0,1\}) \quad \{0,\tanh(1)\}$$

En modo de ángulo en Radianes:

$$\tanh\left(\begin{bmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{bmatrix}\right) \quad \begin{bmatrix} -0.097966 & 0.933436 & 0.425972 \\ 0.488147 & 0.538881 & -0.129382 \\ 1.28295 & -1.03425 & 0.428817 \end{bmatrix}$$

tanh⁻¹()

Catálogo >

tanh(Expr1) ⇒ expresión **tanh⁻¹**(Lista1) ⇒ lista**tanh⁻¹**(Expr1) entrega la tangente hiperbólica inversa del argumento como una expresión.**tanh⁻¹**(Lista1) entrega una lista de las tangentes hiperbólicas inversas de cada elemento de Lista1.**Nota:** Usted puede insertar esta función desde el teclado al escribir **arctanh** (...).**tanh⁻¹**(matrizCuadrada1) ⇒ matrizCuadradaEntrega la tangente hiperbólica inversa de la matriz de matrizCuadrada1. Esto no es lo mismo que calcular la tangente hiperbólica inversa de cada elemento. Para obtener información acerca del método de cálculo, consulte **cos()**.

matrizCuadrada1 debe ser diagonalizable. El resultado siempre contiene números de punto flotante.

En formato complejo Rectangular:

$$\begin{array}{l} \text{tanh}^{-1}(0) \qquad \qquad \qquad 0 \\ \text{tanh}^{-1}(\{1,2,1,3\}) \\ \left\{ \text{undef}, 0.518046 - 1.5708 \cdot i, \frac{\ln(2)}{2} - \frac{\pi \cdot i}{2} \right\} \end{array}$$

En el modo de ángulo en Radianes y el formato complejo Rectangular:

$$\begin{array}{l} \text{tanh}^{-1} \left(\begin{pmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{pmatrix} \right) \\ \left[\begin{array}{ll} -0.099353 + 0.164058 \cdot i & 0.267834 - 1.4908 \\ -0.087596 - 0.725533 \cdot i & 0.479679 - 0.94730 \\ 0.511463 - 2.08316 \cdot i & -0.878563 + 1.7901 \end{array} \right] \end{array}$$

Para ver el resultado completo, presione **▲** y luego use **◀** y **▶** para mover el cursor.**taylor()**

Catálogo >

taylor(Expr1, Var, Orden[, Punto]) ⇒ expresiónEntrega el polinomio de Taylor solicitado. El polinomio incluye términos de no cero de grados del entero desde cero por medio del Orden en (Var menos Punto). **taylor**() se entrega a sí mismo si no hay ninguna serie de potencias truncada de este orden, o si requeriría exponentes negativos o fraccionarios. Use sustitución y/o multiplicación temporal por una potencia de (Var menos Punto) para determinar más series de potencias generales.

Punto se predetermina a cero y es el punto de expansión.

Conforme se ilustra por medio del último ejemplo de la derecha, la corriente abajo de las rutinas de despliegue del resultado producido por **taylor**(...) podría reorganizar los términos de manera que el término dominante no sea el del extremo izquierdo.

$$\begin{array}{l} \text{taylor}(e^{\sqrt{x}}, x, 2) \qquad \qquad \text{taylor}(e^{\sqrt{x}}, x, 2, 0) \\ \text{taylor}(e^t, t, 4) | t = \sqrt{x} \qquad \qquad \frac{3}{24} + \frac{x^2}{6} + \frac{x}{2} + \sqrt{x} + 1 \\ \text{taylor}\left(\frac{1}{x \cdot (x-1)}, x, 3\right) \qquad \text{taylor}\left(\frac{1}{x \cdot (x-1)}, x, 3, 0\right) \\ \text{expand}\left(\frac{\text{taylor}\left(\frac{x}{x \cdot (x-1)}, x, 4\right)}{x}\right) \\ -x^3 - x^2 - x - \frac{1}{x} \end{array}$$

$$\begin{array}{l} \text{taylor}\left(\left(1 + e^x\right)^2, x, 2, 1\right) \\ e \cdot (2 \cdot e + 1) \cdot (x - 1)^2 + (2 \cdot e^2 + 2 \cdot e) \cdot (x - 1) + (e + 1)^2 \end{array}$$

tCdf()

Catálogo >

tCdf(límiteInferior, límiteSuperior, df) ⇒ número si el límiteInferior y el límiteSuperior son números, lista si el límiteInferior y el límiteSuperior son listas

Resuelve la probabilidad de distribución de Student-t entre el límiteInferior y el límiteSuperior para los grados de libertad especificados df.

Para P(X ≤ límiteSuperior), configure límiteInferior = -∞.

tCollect()Catálogo > **tCollect**(Expr1) \Rightarrow expresión

Entrega una expresión en la cual los productos y las potencias de enteros de senos y cosenos se convierten en una combinación lineal de senos y cosenos de ángulos múltiples, sumas de ángulos y diferencias de ángulos. La transformación convierte los polinomios trigonométricos en una combinación lineal de sus armónicos.

En ocasiones, **tCollect()** alcanzará sus metas cuando la simplificación trigonométrica predeterminada no lo logre. **tCollect()** tiende a revertir las transformaciones realizadas por **tExpand()**. En ocasiones, al aplicar **tExpand()** a un resultado de **tCollect()**, o viceversa, en dos pasos independientes se simplifica una expresión.

$$\begin{aligned} \text{tCollect}(\cos(\alpha)^2) &= \frac{\cos(2\alpha)+1}{2} \\ \text{tCollect}(\sin(\alpha)\cdot\cos(\beta)) &= \frac{\sin(\alpha-\beta)+\sin(\alpha+\beta)}{2} \end{aligned}$$

tExpand()Catálogo > **tExpand**(Expr1) \Rightarrow expresión

Entrega una expresión en la cual los senos y cosenos de ángulos múltiples de enteros, sumas de ángulos y diferencias de ángulos se expanden. Debido a la identidad $(\sin(x))^2+(\cos(x))^2=1$, existen muchos resultados equivalentes posibles. En consecuencia, un resultado podría diferir de un resultado mostrado en otras publicaciones.

En ocasiones, **tExpand()** alcanzará sus metas cuando la simplificación trigonométrica predeterminada no lo logre. **tExpand()** tiende a revertir las transformaciones realizadas por **tCollect()**. En ocasiones, al aplicar **tCollect()** a un resultado de **tExpand()**, o viceversa, en dos pasos independientes se simplifica una expresión.

Nota: El ajuste al modo de Grados por $\pi/180$ interfiere con la capacidad de **tExpand()** para reconocer formas expandibles. Para obtener mejores resultados, **tExpand()** se debe usar en el modo de Radián.

$$\begin{aligned} \text{tExpand}(\sin(3\cdot\phi)) &= 4\cdot\sin(\phi)\cdot(\cos(\phi))^2-\sin(\phi) \\ \text{tExpand}(\cos(\alpha-\beta)) &= \cos(\alpha)\cdot\cos(\beta)+\sin(\alpha)\cdot\sin(\beta) \end{aligned}$$

TextCatálogo > **Text** indicarCad[, DespBandera]

Comando de programación: Pausa el programa y despliega la cadena de caracteres *indicarCad* en un cuadro de diálogo.

Cuando el usuario selecciona **OK**, la ejecución del programa continúa.

El argumento *bandera* opcional puede ser cualquier expresión.

- Si *DespBandera* se omite o se evalúa a **1**, el mensaje de texto se agrega al historial de la Calculadora.
- Si *DespBandera* se evalúa a **0**, el mensaje de texto no se agrega al historial.

Si el programa necesita una respuesta escrita del usuario, consulte **Request**, página 102o **RequestStr**, página 103.

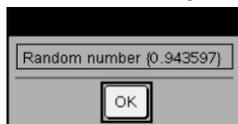
Nota: Usted puede usar este comando dentro de un programa definido por el usuario, pero no dentro de una función.

Defina un programa que pause para desplegar cada uno de cinco números aleatorios en un cuadro de diálogo. Dentro de la plantilla Prgm...TerminarPrgm, llene cada línea al presionar  en lugar de **enter**. En el teclado de la computadora, presione y sostenga **Alt** y presione **Ingresar**.

```
Define text_demo()=Prgm
For i,1,5
  string:="Random number " & string(rand(i))
  Text string
EndFor
EndPrgm
```

Ejecute el programa:
text_demo()

Muestra de un cuadro de diálogo:

**Then (Entonces)**Vea **If**, página 58.

tInterval (intervalO)Catálogo > **tInterval** *Lista[,Frec[,nivelC]]*

(Entrada de lista de datos)

tInterval $\bar{x},sx,n[,nivelC]$

(Entrada de estadísticas de resumen)

Resuelve un intervalo de confianza t . Un resumen de resultados se almacena en la variable *stat.results*. (Vea la página 122.)

Para obtener información sobre el efecto de los elementos vacíos en una lista, vea "Elementos vacíos (inválidos)" en la página 166.

Variable de salida	Descripción
stat.CBajo, stat.CAlto	Intervalo de confianza para una media de población desconocida
stat. \bar{x}	Media de la muestra de la secuencia de datos de la distribución aleatoria normal
stat.ME	Margen de error
stat.df	Grados de libertad
stat. σ_x	Desviación estándar muestra
stat.n	Longitud de la secuencia de datos con media de la muestra muestra

tInterval_2Samp (intervalO_2Muest)Catálogo > **tInterval_2Samp***Lista1,Lista2[,Frec1[,Frec2[,nivelC[,Agrupado]]]]*

(Entrada de lista de datos)

tInterval_2Samp $\bar{x}_1,sx_1,n_1,\bar{x}_2,sx_2,n_2[,nivelC[,Agrupado]]$

(Entrada de estadísticas de resumen)

Resuelve un intervalo de confianza t de dos muestras. Un resumen de resultados se almacena en la variable *stat.results*. (Vea la página 122.)*Agrupado=1* agrupa las varianzas; *Agrupado=0* no agrupa las varianzas.

Para obtener información sobre el efecto de los elementos vacíos en una lista, vea "Elementos vacíos (inválidos)" en la página 166.

Variable de salida	Descripción
stat.CBajo, stat.CAlto	Intervalo de confianza que contiene la probabilidad de distribución del nivel de confianza
stat. $\bar{x}_1-\bar{x}_2$	Medias de las muestras de las secuencias de datos de la distribución aleatoria normal
stat.ME	Margen de error
stat.df	Grados de libertad
stat. \bar{x}_1 , stat. \bar{x}_2	Medias muestra de las secuencias de datos de la distribución aleatoria normal
stat. σ_{x1} , stat. σ_{x2}	Desviaciones estándar muestra para <i>Lista 1</i> y <i>Lista 2</i>

Variable de salida	Descripción
stat.n1, stat.n2	Número de muestras en las secuencias de datos
stat.sp	La desviación estándar agrupada. Calculada cuando <i>Agrupado</i> = Sí

tmpCnv()

Catálogo > 

tmpCnv(Expr °unidadTemp, °unidadTemp2)

⇒ expresión °unidadTemp2

Convierte un valor de temperatura especificado por Expr de una unidad a otra. Las unidades de temperatura válidas son:

°C Celsius

°F Fahrenheit

°K Kelvin

°R Rankine

Para escribir °, selecciónelo de entre los símbolos del Catálogo.

para escribir °, presione  .

Por ejemplo, 100 °C se convierte a 212 °F.

Para convertir un rango de temperatura, use **ΔtmpCnv()** en su lugar.

$\text{tmpCnv}(100 \cdot \text{°C}, \text{°F})$ 212. °F

$\text{tmpCnv}(32 \cdot \text{°F}, \text{°C})$ 0. °C

$\text{tmpCnv}(0 \cdot \text{°C}, \text{°K})$ 273.15 °K

$\text{tmpCnv}(0 \cdot \text{°F}, \text{°R})$ 459.67 °R

Nota: Usted puede usar el Catálogo para seleccionar las unidades de temperatura.

ΔtmpCnv()

Catálogo > 

ΔtmpCnv(Expr °unidadTemp, °unidadTemp2)

⇒ expresión °unidadTemp2

Nota: Usted puede insertar esta función desde el teclado al escribir **cnvTempDelta** (...).

Convierte un rango de temperatura (la diferencia entre dos valores de temperatura) especificado por Expr de una unidad a otra. Las unidades de temperatura válidas son:

°C Celsius

°F Fahrenheit

°K Kelvin

°R Rankine

Para ingresar °, selecciónelo desde la Paleta de Símbolos o escriba **@d**.

Para escribir °, presione  .

1 °C y 1 °K tienen la misma magnitud, al igual que 1 °F y 1 °R. Sin embargo, 1 °C es 9/5 tan grande como 1 °F.

Por ejemplo, un rango de 100 °C (desde 0 °C hasta 100 °C) es equivalente a un rango de 180 °F.

Para convertir un valor de temperatura particular en lugar de un rango, use **tmpCnv()**.

$\Delta\text{tmpCnv}(100 \cdot \text{°C}, \text{°F})$ 180. °F

$\Delta\text{tmpCnv}(180 \cdot \text{°F}, \text{°C})$ 100. °C

$\Delta\text{tmpCnv}(100 \cdot \text{°C}, \text{°K})$ 100. °K

$\Delta\text{tmpCnv}(100 \cdot \text{°F}, \text{°R})$ 100. °R

$\Delta\text{tmpCnv}(1 \cdot \text{°C}, \text{°F})$ 1.8 °F

Nota: Usted puede usar el Catálogo para seleccionar las unidades de temperatura.

tPdf() (PdfT)

Catálogo > 

tPdf(ValX,d) ⇒ número si ValX es un número, lista si ValX es una lista

Resuelve la función de densidad de probabilidad (pdf) para la distribución de Student-*t* a un valor *x* especificado con grados de libertad *d*/especificados.

trace() (trazado)

Catálogo >

trace(*matriz*:Cuadrada) ⇒ *expresión*Entrega el trazado (suma de todos los elementos de la diagonal principal) de *matriz*:Cuadrada.

$$\text{trace} \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix} \quad 15$$

$$\text{trace} \begin{pmatrix} a & 0 \\ 1 & a \end{pmatrix} \quad 2 \cdot a$$

Try (Intentar)

Catálogo >

Try*bloque1***Else***bloque2***EndTry**

Ejecuta el *bloque1* a menos que ocurra un error. La ejecución del programa se transfiere al *bloque2* si ocurre un error en el *bloque1*. La variable de sistema *códigoErr* contiene el código del error para permitir que el programa ejecute la recuperación del error. Para obtener una lista de códigos de error, vea "Códigos y mensajes de error", página 172.

bloque1 y *bloque2* pueden ser una sentencia sencilla o una serie de sentencias separadas por el caracter ":".

Nota para ingresar el ejemplo: En la aplicación de Calculadora en el dispositivo portátil, usted puede ingresar

definiciones en líneas múltiples al presionar en lugar de al final de cada línea. En el teclado de la computadora, presione y sostenga **Alt** y presione **Ingresar**.

```
Define prog1()=Prgm
  Try
  z:=z+1
  Disp "z incremented."
  Else
  Disp "Sorry, z undefined."
  EndTry
EndPrgm
```

Done

z:=1:prog1()

z incremented.

Done

DelVar z:prog1()

Sorry, z undefined.

Done

Ejemplo 2

Para ver los comandos **Try**, **ClrErr**, y **PassErr** en operación, ingrese el programa *valspropios()* que se muestra a la derecha. Ejecute el programa al ejecutar cada una de las siguientes expresiones.

$$\text{eigenvals} \left(\begin{bmatrix} -3 \\ -41 \\ 5 \end{bmatrix}, \begin{bmatrix} -1 & 2 & -3.1 \end{bmatrix} \right)$$

$$\text{eigenvals} \left(\begin{bmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \end{bmatrix}, \begin{bmatrix} 1 \\ 2 \end{bmatrix} \right)$$

Nota: Vea también **ClrErr**, página 19y **PassErr**, página 89.

```
Defina valspropios(a,b)=Prgm
© El programa valspropios(A,B) despliega los valores propios de
Try
  Disp "A= ",a
  Disp "B= ",b
  Disp " "
  Disp "Los valores propios de A-B son:",eigVl(a*b)
Else
  If errCode=230 Then
  Disp "Error: El producto de A-B debe ser una matriz
cuadrada"
  ClrErr
  Else
  PassErr
  EndIf
EndTry
EndPrgm
```

tTest $\mu_0, Lista[, Frec[, Hipot]]$

(Entrada de lista de datos)

tTest $\mu_0, \bar{x}, sx, n[, Hipot]$

(Entrada de estadísticas de resumen)

Realiza una prueba de hipótesis para una sola media de población desconocida μ cuando la desviación estándar de población, σ se desconoce. Un resumen de resultados se almacena en la variable *stat.results*. (Vea la página 122.)

Pruebe $H_0: \mu = \mu_0$, contra uno de los siguientes:

Para $H_a: \mu < \mu_0$, configure *Hipot*<0

Para $H_a: \mu \neq \mu_0$ (predeterminado), configure *Hipot*=0

Para $H_a: \mu > \mu_0$, configure *Hipot*>0

Para obtener información sobre el efecto de los elementos vacíos en una lista, vea "Elementos vacíos (inválidos)" en la página 166.

Variable de salida	Descripción
stat.t	$(\bar{x} - \mu_0) / (\text{desvest} / \text{sqrt}(n))$
stat.ValP	Nivel más bajo de significancia en el cual la hipótesis nula se puede rechazar
stat.df	Grados de libertad
stat. \bar{x}	Media de muestra de la secuencia de datos en <i>Lista</i>
stat.ex	Desviación estándar muestra de la secuencia de datos
stat.n	Tamaño de la muestra

tTest_2Samp (pruebaT_2Muest)**tTest_2Samp** *Lista1, Lista2[, Frec1[, Frec2[, Hipot[, Agrupado]]]]*

(Entrada de lista de datos)

tTest_2Samp $\bar{x}_1, sx1, n1, \bar{x}_2, sx2, n2[, Hipot[, Agrupado]]$

(Entrada de estadísticas de resumen)

Resuelve una prueba *T* de dos muestras. Un resumen de resultados se almacena en la variable *stat.results*. (Vea la página 122.)

Pruebe $H_0: \mu_1 = \mu_2$, contra uno de los siguientes:

Para $H_a: \mu_1 < \mu_2$, configure *Hipot*<0

Para $H_a: \mu_1 \neq \mu_2$ (predeterminado), configure *Hipot*=0

Para $H_a: \mu_1 > \mu_2$, configure *Hipot*>0

Agrupado=1 agrupa las varianzas

Agrupado=0 no agrupa las varianzas

Para obtener información sobre el efecto de los elementos vacíos en una lista, vea "Elementos vacíos (inválidos)" en la página 166.

Variable de salida	Descripción
stat.t	Valor normal estándar resuelto para la diferencia de las medias
stat.ValP	Nivel más bajo de significancia en el cual la hipótesis nula se puede rechazar
stat.df	Grados de libertad para la estadística T
stat. \bar{x} 1, stat. \bar{x} 2	Medias muestra de las secuencias de datos en <i>Lista 1</i> y <i>Lista 2</i>
stat.sx1, stat.sx2	Desviaciones estándar de muestras de las secuencias de datos en <i>Lista 1</i> y <i>Lista 2</i>
stat.n1, stat.n2	Tamaño de las muestras
stat.sp	La desviación estándar agrupada. Calculada cuando <i>Agrupado</i> =1.

tvfV()

Catálogo > 

tvfV($N, I, VP, Pgo, [PpA], [CpA], [PgoA]$) \Rightarrow valor

La función financiera que calcula el valor futuro del dinero.

Nota: Los argumentos que se usan en las funciones del VTD se describen en la tabla de argumentos del VTD, página 135. Vea también **amortTbl()**, página 7.

$$\text{tvfV}(120, 5, 0, -500, 12, 12) \quad 77641.1$$

tvml()

Catálogo > 

tvml($N, VP, Pgo, [PpA], [CpA], [PgoA]$) \Rightarrow valor

La función financiera que calcula la tasa de interés por año.

Nota: Los argumentos que se usan en las funciones del VTD se describen en la tabla de argumentos del VTD, página 135. Vea también **amortTbl()**, página 7.

$$\text{tvml}(240, 100000, -1000, 0, 12, 12) \quad 10.5241$$

tvnN()

Catálogo > 

tvnN($N, I, VP, Pgo, [PpA], [CpA], [PgoA]$) \Rightarrow valor

La función financiera que calcula el número de periodos de pago.

Nota: Los argumentos que se usan en las funciones del VTD se describen en la tabla de argumentos del VTD, página 135. Vea también **amortTbl()**, página 7.

$$\text{tvnN}(5, 0, -500, 77641, 12, 12) \quad 120.$$

tvmpmt()

Catálogo > 

tvmpmt($N, I, VP, VF, [PpA], [CpA], [PgoA]$) \Rightarrow valor

La función financiera que calcula la cantidad de cada pago.

Nota: Los argumentos que se usan en las funciones del VTD se describen en la tabla de argumentos del VTD, página 135. Vea también **amortTbl()**, página 7.

$$\text{tvmpmt}(60, 4, 30000, 0, 12, 12) \quad -552.496$$

tvmpv()

Catálogo > 

tvmpv($N, I, Pgo, VF, [PpA], [CpA], [PgoA]$) \Rightarrow valor

La función financiera que calcula el valor presente.

Nota: Los argumentos que se usan en las funciones del VTD se describen en la tabla de argumentos del VTD, página 135. Vea también **amortTbl()**, página 7.

$$\text{tvmpv}(48, 4, -500, 30000, 12, 12) \quad -3426.7$$

argumento del VTD*	Descripción	Tipo de datos
N	Número de periodos de pago	número real
I	tasa de interés anual	número real
VP	Valor presente	número real
Pgo	cantidad de pago	número real
VF	Valor futuro	número real
PpA	Pagos por año, predeterminado=1	entero > 0
CpA	Periodos de capitalización por año, predeterminado=1	entero > 0
$PgoAl$	Pago vencido al final o al principio de cada periodo, predeterminado=final	entero (0=final, 1=principio)

* Estos nombres de argumento de valor tiempo del dinero son similares a los nombres de variable del VTD (como **vtd.vp** y **vtd.pgo**) que se usan en el solucionador financiero de la aplicación de la Calculadora. Sin embargo, las funciones financieras no almacenan sus valores o resultados de argumento para las variables del VTD.

TwoVar (DosVar)

Catálogo > 

TwoVar $X, Y, [Frec] [, Categoría, Incluir]$

Calcula las estadísticas de DosVar. Un resumen de resultados se almacena en la variable *stat.results*. (Vea la página 122.)

Todas las listas deben tener una dimensión igual, excepto por *Incluir*.

X y Y son listas de variables independientes y dependientes.

Frec es una lista opcional de valores de frecuencia. Cada elemento en *Frec* especifica la frecuencia de la ocurrencia para cada punto de datos X y Y correspondientes. El valor predeterminado es 1. Todos los elementos deben ser enteros ≥ 0 .

Categoría es una lista de códigos de categoría numérica para los datos de X y Y correspondientes.

Incluir es una lista de uno o más códigos de categoría. Sólo aquellos elementos de datos cuyo código de categoría está incluido en esta lista están incluidos en el cálculo.

Un elemento (inválido) vacío en cualquiera de las listas X , *Frec* o *Categoría* da como resultado un inválido para el elemento correspondiente de todas esas listas. Un elemento vacío en cualquiera de las listas $X1$ a $X20$ da como resultado un inválido para el elemento correspondiente de todas esas listas. Para obtener más información sobre elementos vacíos, vea la página 166.

Variable de salida	Descripción
stat. \bar{x}	Media de valores x
stat. Σx	Suma de valores x
stat. Σx^2	Suma de valores x^2
stat. s_x	Desviación estándar de muestra de x
stat. σ_x	Desviación estándar de población de x

Variable de salida	Descripción
stat.n	Número de puntos de datos
stat. \bar{y}	Media de valores y
stat. Σy	Suma de valores y
stat. Σy^2	Suma de valores y^2
stat.sy	Desviación estándar de muestra de y
stat. σy	Desviación estándar de población de y
stat. Σxy	Suma de los valores $x \cdot y$
stat.r	Coefficiente de correlación
stat.MinX	Mínimo de valores x
stat.C ₁ X	1er Cuartil de x
stat.MedianaX	Mediana de x
stat.C ₃ X	3er Cuartil de x
stat.MaxX	Máximo de valores x
stat.MinY	Mínimo de valores y
stat.C ₁ Y	1er Cuartil de y
stat.MedY	Mediana de y
stat.C ₃ Y	3er Cuartil de y
stat.MaxY	Máximo de valores y
stat. $\Sigma(x-\bar{x})^2$	Suma de cuadrados de desviaciones de la media de x
stat. $\Sigma(y-\bar{y})^2$	Suma de cuadrados de desviaciones de la media de y

U

unitV()

Catálogo >

unitV(*Vector1*) ⇒ *vector*

Entrega un vector de unidad de fila o de columna, dependiendo de la forma de *Vector1*.

Vector1 debe ser una matriz de fila sencilla o una matriz de columna sencilla.

$$\text{unitV}\left(\begin{bmatrix} a & b & c \end{bmatrix}\right) = \left[\frac{a}{\sqrt{a^2+b^2+c^2}} \quad \frac{b}{\sqrt{a^2+b^2+c^2}} \quad \frac{c}{\sqrt{a^2+b^2+c^2}} \right]$$

$$\text{unitV}\left(\begin{bmatrix} 1 & 2 & 1 \end{bmatrix}\right) = \left[\frac{\sqrt{6}}{6} \quad \frac{\sqrt{6}}{3} \quad \frac{\sqrt{6}}{6} \right]$$

$$\text{unitV}\left(\begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}\right) = \begin{bmatrix} \frac{\sqrt{14}}{14} \\ \frac{14}{\sqrt{14}} \\ \frac{7}{3\sqrt{14}} \\ 14 \end{bmatrix}$$

Para ver el resultado completo, presione \blacktriangle y luego use \blacktriangleleft y \blacktriangleright para mover el cursor.

unLock (desbloquear)

Catálogo >

unLock *Var1*[, *Var2*] [, *Var3*] ...

unLock *Var*.

Desbloquea las variables o el grupo de variables especificado. Las variables bloqueadas no se pueden modificar ni borrar.

Vea **Lock**, página 71 y **getLockInfo()**, página 55.

$a:=65$	65
Lock <i>a</i>	Done
getLockInfo(<i>a</i>)	1
$a:=75$	"Error: Variable is locked."
DelVar <i>a</i>	"Error: Variable is locked."
Unlock <i>a</i>	Done
$a:=75$	75
DelVar <i>a</i>	Done

V

varPop()

Catálogo >

varPop(*Lista*[, *listaFrec*]) ⇒ *expresión*

Entrega la varianza de población de *Lista*.

Cada elemento de *listaFrec* cuenta el número de ocurrencias consecutivas del elemento correspondiente en *Lista*.

Nota: *Lista* debe contener al menos dos elementos.

Si un elemento en cualquiera de las listas está vacío (inválido), ese elemento se ignora, y el elemento correspondiente en la otra lista también se ignora. Para obtener más información sobre elementos vacíos, vea la página 166.

varPop({5,10,15,20,25,30})	$\frac{875}{12}$
Ans·1.	72.9167

varSamp() (varMuest)

Catálogo >

varSamp(*Lista1*, *listaFrec*) ⇒ *expresión*Entrega la varianza muestra de *Lista*.Cada elemento de *listaFrec* cuenta el número de ocurrencias consecutivas del elemento correspondiente en *Lista*.**Nota:** *Lista* debe contener al menos dos elementos.

Si un elemento en cualquiera de las listas está vacío (inválido), ese elemento se ignora, y el elemento correspondiente en la otra lista también se ignora. Para obtener más información sobre elementos vacíos, vea la página 166.

$$\text{varSamp}\left(\{a,b,c\}\right) = \frac{a^2 - a \cdot (b+c) + b^2 - b \cdot c + c^2}{3}$$

$$\text{varSamp}\left(\{1,2,5, -6,3, -2\}\right) = \frac{31}{2}$$

$$\text{varSamp}\left(\{1,3,5\}, \{4,6,2\}\right) = \frac{68}{33}$$

varSamp(*Matriz1*, *matrizFrec*) ⇒ *matriz*Entrega un vector de fila que contiene la varianza muestra de cada columna en *Matriz1*.Cada elemento de *matrizFrec* cuenta el número de ocurrencias consecutivas del elemento correspondiente en *Matriz1*.

Si un elemento en cualquiera de las matrices está vacío (inválido), ese elemento se ignora, y el elemento correspondiente en la otra matriz también se ignora. Para obtener más información sobre elementos vacíos, vea la página 166.

Nota: *Matriz1* debe contener al menos dos filas.

$$\text{varSamp}\left(\begin{bmatrix} 1 & 2 & 5 \\ -3 & 0 & 1 \\ .5 & .7 & 3 \end{bmatrix}\right) = \begin{bmatrix} 4.75 & 1.03 & 4 \end{bmatrix}$$

$$\text{varSamp}\left(\begin{bmatrix} -1.1 & 2.2 \\ 3.4 & 5.1 \\ -2.3 & 4.3 \end{bmatrix}, \begin{bmatrix} 6 & 3 \\ 2 & 4 \\ 5 & 1 \end{bmatrix}\right) = \begin{bmatrix} 3.91731 & 2.08411 \end{bmatrix}$$

W**warnCodes()**

Catálogo >

warnCodes(*Expr1*, *VarEstado*) *expresión* ⇒Evalúa la expresión *Expr1*, entrega el resultado y almacena los códigos de cualquier advertencia generada en la variable de lista *varEstado*. Si no se genera ninguna advertencia, esta función asigna a *varEstado* una lista vacía.*Expr1* puede ser cualquier expresión matemática de TI-Nspire™ o de CAS de TI-Nspire™. Usted no puede usar un comando o asignación como *Expr1*.*VarEstado* debe ser un nombre de variable válido.

Para obtener una lista de códigos de advertencia y mensajes asociados, vea la página 177.

$$\text{warnCodes}\left(\text{solve}\left(\sin(10 \cdot x) = \frac{x^2}{x}, x\right), \text{warn}\right)$$

$$x = -0.84232 \text{ or } x = -0.706817 \text{ or } x = -0.285234 \text{ or } x = 0$$

$$\text{warn} = \{10007, 10009\}$$

Para ver el resultado completo, presione \blacktriangle y después use \blacktriangleleft y \blacktriangleright para mover el cursor.**when() (cuando)**

Catálogo >

when(*Condición*, *resultadoVerdadero* [, *resultadoFalso*], *resultadoDesconocido*) ⇒ *expresión*Entrega un *resultadoVerdadero*, *resultadoFalso* o *resultadoDesconocido*, dependiendo de si la *Condición* es verdadera, falsa o desconocida. Entrega la entrada si hay muy pocos argumentos para especificar el resultado apropiado.Omita tanto el *resultadoFalso* como el *resultadoDesconocido* para hacer una expresión definida sólo en la región donde la *Condición* es verdadera.Use un **undef** *resultadoFalso* para definir una expresión que se grafique sólo en un intervalo.

$$\text{when}(x < 0, x + 3) | x = 5 \quad \text{undef}$$

when() (cuando)

Catálogo >

when() es útil para definir funciones recursivas.

$\text{when}(n > 0, n \cdot \text{factoral}(n-1), 1) \rightarrow \text{factoral}(n)$	Done
$\text{factoral}(3)$	6
3!	6

While (Mientras)

Catálogo >

While Condición

Bloque

EndWhileEjecuta las sentencias en *Bloque* siempre y cuando la *Condición* sea verdadera.*Bloque* puede ser una sentencia sencilla o una secuencia de sentencias separadas con el caracter ":".

Nota para ingresar el ejemplo: En la aplicación de Calculadora en el dispositivo portátil, usted puede ingresar definiciones en líneas múltiples al presionar en lugar de **enter** al final de cada línea. En el teclado de la computadora, presione y sostenga **Alt** y presione **Ingresar**.

Define $\text{sum_of_recip}(n) = \text{Func}$	Local $i, \text{tempsum}$
	$1 \rightarrow i$
	$0 \rightarrow \text{tempsum}$
	While $i \leq n$
	$\text{tempsum} + \frac{1}{i} \rightarrow \text{tempsum}$
	$i + 1 \rightarrow i$
	EndWhile
	Return tempsum
	EndFunc
	Done
$\text{sum_of_recip}(3)$	$\frac{11}{6}$

X**xor**

Catálogo >

BooleanaExpr1 xor BooleanaExpr2 devuelve expresión booleana

true xor true false

BooleanaLista1 xor BooleanaLista2 devuelve lista booleana

5 > 3 xor 3 > 5 true

*BooleanaMatriz1 xor BooleanaMatriz2 devuelve matriz booleana*Entrega verdadero si *ExprBooleana1* es verdadera y *ExprBooleana2* es falsa, o viceversa.

Entrega falso si ambos argumentos son verdaderos o si ambos son falsos. Entrega una expresión Booleana simplificada si cualquiera de los argumentos no se puede resolver a verdadero o falso.

Nota: Vea **or**, página 88.

Entero1 **xor** Entero2 \Rightarrow entero

Compara dos enteros reales bit por bit usando una operación **xor**. En forma interna, ambos enteros se convierten en números binarios de 64 bits firmados. Cuando se comparan los bits correspondientes, el resultado es 1 si cualquiera de los bits (pero no ambos) es 1; el resultado es 0 si ambos bits son 0 ó ambos bits son 1. El valor producido representa los resultados de los bits, y se despliega de acuerdo con el modo de Base.

Se pueden ingresar enteros en cualquier base de números. Para un ingreso binario o hexadecimal, se debe usar el prefijo 0b ó 0h, respectivamente. Sin un prefijo, los enteros se tratan como decimales (base 10).

Si se ingresa un entero decimal que es demasiado grande para una forma binaria de 64 bits firmada, se usa una operación de módulo simétrico para llevar el valor al rango apropiado. Para obtener más información, vea **Base2**, página 14.

Nota: Vea **or**, página 88.

En modo de base hexadecimal:

Importante: Utilice el número cero, no la letra "0".

0h7AC36 xor 0h3D5F 0h79169

En modo de base binaria:

0b100101 xor 0b100 0b100001

Nota: Un ingreso binario puede tener hasta 64 dígitos (sin contar el prefijo 0b). Un ingreso hexadecimal puede tener hasta 16 dígitos.

Z

zeros()

zeros(Expr, Var) \Rightarrow lista

zeros(Expr, Var=Cálculo) \Rightarrow lista

Entrega una lista de valores reales posibles de Var que hacen Expr=0. **zeros()** hace esto al resolver

expList(solve(Expr=0, Var), Var).

Para algunos propósitos, la forma de resultado para **zeros()** es más conveniente que la de **solve()**. Sin embargo, la forma de resultado de **zeros()** no puede expresar soluciones implícitas, soluciones que requieren desigualdades o soluciones que no implican Var.

Nota: Vea también **cSolve()**, **cZeros()**, y **solve()**.

zeros({Expr1, Expr2},

{VarOCálculo1, VarOCálculo2 [, ...]}) \Rightarrow matriz

Entrega ceros reales posibles de las expresiones algebraicas simultáneas, donde cada VarOCálculo especifica un desconocido cuyo valor usted busca.

De manera opcional, se puede especificar un cálculo inicial para una variable. Cada VarOCálculo debe tener la forma:

variable

- 0 -

variable = número real o noreal

Por ejemplo, x es válida y también lo es x=3.

$$\text{zeros}\left(a \cdot x^2 + b \cdot x + c, x\right)$$

$$\left\{ \frac{\sqrt{b^2 - 4 \cdot a \cdot c} - b}{2 \cdot a}, \frac{-\left(\sqrt{b^2 - 4 \cdot a \cdot c} + b\right)}{2 \cdot a} \right\}$$

$$a \cdot x^2 + b \cdot x + c | x = \text{Ans}[2]$$

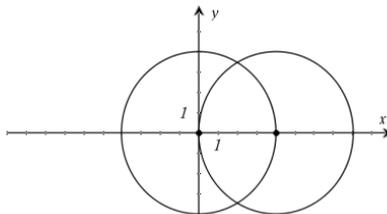
$$\text{exact}\left(\text{zeros}\left(a \cdot \left(e^x + x\right) \cdot \left(\text{sign}(x) - 1\right), x\right)\right)$$

$$\text{exact}\left(\text{solve}\left(a \cdot \left(e^x + x\right) \cdot \left(\text{sign}(x) - 1\right) = 0, x\right)\right)$$

$$e^x + x = 0 \text{ or } x > 0 \text{ or } a = 0$$

Si todas las expresiones son polinomios y usted NO especifica cualquier cálculo inicial, **zeros()** usa el método de eliminación de léxico Gröbner/Buchberger para intentar determinar todos los ceros reales.

Por ejemplo, supongamos que usted tiene un círculo de radio e en el origen y otro círculo de radio r centrado donde el primer círculo cruza el eje x positivo. Use **zeros()** para encontrar las intersecciones.



Conforme se ilustra con r en el ejemplo de la derecha, las expresiones polinómicas simultáneas pueden tener variables extras que no tienen ningún valor, aunque representan valores numéricos dados que podrían sustituirse más adelante.

Cada fila de la matriz resultante representa un cero alterno, con los componentes ordenados igual que la lista *varOC* cálculo list. Para extraer una fila, indexe la matriz con [fila].

$$\text{zeros}\left(\left\{x^2+y^2-r^2,(x-r)^2+y^2-r^2\right\},\{x,y\}\right)$$

$$\begin{bmatrix} r & -\sqrt{3}\cdot r \\ 2 & 2 \\ r & \sqrt{3}\cdot r \\ 2 & 2 \end{bmatrix}$$

Extraer la fila 2:

$$\text{Ans}[2] \quad \begin{bmatrix} r & \sqrt{3}\cdot r \\ 2 & 2 \end{bmatrix}$$

Usted también (o en lugar de) puede incluir incógnitas que no aparecen en las expresiones. Por ejemplo, usted puede incluir z como una incógnita para extender el ejemplo anterior a dos cilindros intersectados paralelos del radio r . Los ceros de los cilindros ilustran cómo las familias de ceros podrían contener constantes arbitrarias en la forma ck , donde k es un sufijo de entero desde 1 hasta 255.

$$\text{zeros}\left(\left\{x^2+y^2-r^2,(x-r)^2+y^2-r^2\right\},\{x,y,z\}\right)$$

$$\begin{bmatrix} r & -\sqrt{3}\cdot r & c1 \\ 2 & 2 & \\ r & \sqrt{3}\cdot r & c1 \\ 2 & 2 & \end{bmatrix}$$

Para sistemas polinómicos, el tiempo de cálculo o el agotamiento de memoria pueden depender ampliamente del orden en el cual se enumeran los desconocidos. Si su elección inicial agota la memoria o su paciencia, intente volver a arreglar las variables en las expresiones y o en la lista *varOC* cálculo .

Si usted no incluye ningún cálculo y si cualquier expresión no es polinómica en cualquier variable, pero todas las expresiones son lineales en todas las incógnitas, **zeros()** usa la eliminación Gaussiana para tratar de determinar todos los ceros reales.

$$\text{zeros}\left(\left\{x+e^z\cdot y-1,x-y-\sin(z)\right\},\{x,y\}\right)$$

$$\begin{bmatrix} e^z\cdot \sin(z)+1 & -(\sin(z)-1) \\ e^z+1 & e^z+1 \end{bmatrix}$$

Si un sistema no es ni polinómico en todas sus variables ni lineal en sus incógnitas, **zeros()** determina como máximo un cero usando un método iterativo aproximado. Para hacer esto, el número de desconocidos debe igualar el número de expresiones, y todas las demás variables en las expresiones deben simplificarse a números.

Cada incógnita comienza en su valor calculado si hay uno; de otro modo, comienza en 0.0.

Use cálculos para buscar ceros adicionales de uno en uno. Por convergencia, un cálculo puede tener que ser más bien cercano a un cero.

$$\text{zeros}\left(\left\{x+e^z\cdot y-1,x-y-\sin(z)\right\},\{y,z\}\right)$$

$$\begin{bmatrix} 0.041458 & 3.18306 \\ 0.001871 & 6.28131 \\ 2.812\text{E}^{-10} & 21.9911 \end{bmatrix}$$

$$\text{zeros}\left(\left\{e^z\cdot y-1,x-y-\sin(z)\right\},\{y,z=2\cdot\pi\}\right)$$

$$\begin{bmatrix} 0.001871 & 6.28131 \end{bmatrix}$$

zInterval (intervaloZ)Catálogo > **zInterval** $\sigma, Lista [, Frec [, nivelC]]$

(Entrada de lista de datos)

zInterval $\sigma, \bar{X}, n [, nivelC]$

(Entrada de estadísticas de resumen)

Resuelve un intervalo de confianza Z . Un resumen de resultados se almacena en la variable *stat.results*. (Vea la página 122.)

Para obtener información sobre el efecto de los elementos vacíos en una lista, vea "Elementos vacíos (inválidos)" en la página 166.

Variable de salida	Descripción
stat.CBajo, stat.CAlto	Intervalo de confianza para una media de población desconocida
stat. \bar{X}	Media muestra de la secuencia de datos de la distribución aleatoria normal
stat.ME	Margen de error
stat.ex	Desviación estándar muestra
stat.n	Longitud de la secuencia de datos con media muestra
stat. σ	Desviación estándar de población conocida para la secuencia de datos <i>Lista</i>

zInterval_1Prop (intervaloZ_1Prop)Catálogo > **zInterval_1Prop** $x, n [, nivelC]$ Resuelve un intervalo de confianza Z de una proporción. Un resumen de resultados se almacena en la variable *stat.results*. (Vea la página 122.) x es un entero no negativo.

Para obtener información sobre el efecto de los elementos vacíos en una lista, vea "Elementos vacíos (inválidos)" en la página 166.

Variable de salida	Descripción
stat.CBajo, stat.CAlto	Intervalo de confianza que contiene la probabilidad de distribución del nivel de confianza
stat. \hat{p}	La proporción de éxitos calculada
stat.ME	Margen de error
stat.n	Número de muestras en la secuencia de datos

zInterval_2Prop (intervaloZ_2Prop)Catálogo > **zInterval_2Prop** $x1, n1, x2, n2, nivelC]$

Resuelve un intervalo de confianza Z de dos proporciones. Un resumen de resultados se almacena en la variable *stat.results*. (Vea la página 122.)

$x1$ y $x2$ son enteros no negativos.

Para obtener información sobre el efecto de los elementos vacíos en una lista, vea "Elementos vacíos (inválidos)" en la página 166.

Variable de salida	Descripción
stat.CBajo, stat.CAlto	Intervalo de confianza que contiene la probabilidad de distribución del nivel de confianza
stat.pDif	La diferencia entre proporciones calculada
stat.ME	Margen de error
stat.p1	Estimación de proporción de primera muestra
stat.p2	Estimación de proporción de segunda muestra
stat.n1	Tamaño de la muestra en una secuencia de datos
stat.n2	Tamaño de la muestra en la secuencia de datos de dos

zInterval_2Samp (intervaloZ_2Muest)Catálogo > **zInterval_2Samp** $\sigma_1, \sigma_2, Lista1, Lista2, Frec1, Frec2, nivelC]]$

(Entrada de lista de datos)

zInterval_2Samp $\sigma_1, \sigma_2, \bar{x}1, n1, \bar{x}2, n2, nivelC]$

(Entrada de estadísticas de resumen)

Resuelve un intervalo de confianza Z de dos muestras. Un resumen de resultados se almacena en la variable *stat.results*. (Vea la página 122.)

Para obtener información sobre el efecto de los elementos vacíos en una lista, vea "Elementos vacíos (inválidos)" en la página 166.

Variable de salida	Descripción
stat.CBajo, stat.CAlto	Intervalo de confianza que contiene la probabilidad de distribución del nivel de confianza
stat.x1-x2	Medias muestra de las secuencias de datos de la distribución aleatoria normal
stat.ME	Margen de error
stat.x1, stat.x2	Medias muestra de las secuencias de datos de la distribución aleatoria normal
stat.sx1, stat.sx2	Desviaciones estándar muestras para <i>Lista 1</i> y <i>Lista 2</i>
stat.n1, stat.n2	Número de muestras en las secuencias de datos
stat.r1, stat.r2	Desviaciones estándar de población conocidas para <i>Lista 1</i> y <i>Lista 2</i>

zTest $\mu_0, \sigma, \text{Lista}, [\text{Frec}, \text{Hipot}]$

(Entrada de lista de datos)

zTest $\mu_0, \sigma, \bar{x}, n, [\text{Hipot}]$

(Entrada de estadísticas de resumen)

Realiza una prueba z con frecuencia *listaFrec*. Un resumen de resultados se almacena en la variable *stat.results*. (Vea la página 122.)

Pruebe $H_0: \mu = \mu_0$, contra uno de los siguientes:

Para $H_a: \mu < \mu_0$, configure *Hipot*<0

Para $H_a: \mu \neq \mu_0$ (predeterminado), configure *Hipot*=0

Para $H_a: \mu > \mu_0$, configure *Hipot*>0

Para obtener información sobre el efecto de los elementos vacíos en una lista, vea "Elementos vacíos (inválidos)" en la página 166.

Variable de salida	Descripción
stat.z	$(\bar{x} - \mu_0) / (\sigma / \sqrt{n})$
stat.Valor P	Probabilidad más baja a la cual la hipótesis nula se puede rechazar
stat. \bar{x}	Media de muestra de la secuencia de datos en <i>Lista</i>
stat.ex	Desviación estándar de muestras de la secuencia de datos. Sólo se entrega para la entrada de <i>Datos</i> .
stat.n	Tamaño de la muestra

zTest_1Prop (pruebaZ_1Prop)**zTest_1Prop** $p_0, x, n, [\text{Hipot}]$

Resuelve una prueba Z de una proporción. Un resumen de resultados se almacena en la variable *stat.results*. (Vea la página 122.)

x es un entero no negativo.

Pruebe $H_0: p = p_0$ contra uno de los siguientes:

Para $H_a: p > p_0$, configure *Hipot*>0

Para $H_a: p \neq p_0$ (predeterminado), configure *Hipot*=0

Para $H_a: p < p_0$, configure *Hipot*<0

Para obtener información sobre el efecto de los elementos vacíos en una lista, vea "Elementos vacíos (inválidos)" en la página 166.

Variable de salida	Descripción
stat.p0	Proporción poblacional de la hipótesis
stat.z	Valor normal estándar calculado para la proporción
stat.ValP	Nivel más bajo de significancia en el cual la hipótesis nula se puede rechazar
stat. \hat{p}	Proporción muestral estimada
stat.n	Tamaño de la muestra

zTest_2Prop (pruebaZ_2Prop)Catálogo > **zTest_2Prop** $x1, n1, x2, n2[, Hipot]$

Resuelve una prueba Z de dos proporciones. Un resumen de resultados se almacena en la variable *stat.results*. (Vea la página 122.)

$x1$ y $x2$ son enteros no negativos.

Pruebe $H_0: p1 = p2$, contra uno de los siguientes:

Para $H_a: p1 > p2$, configure *Hipot*>0

Para $H_a: p1 \neq p2$ (predeterminado), configure *Hipot*=0

Para $H_a: p < p0$, configure *Hipot*<0

Para obtener información sobre el efecto de los elementos vacíos en una lista, vea "Elementos vacíos (inválidos)" en la página 166.

Variable de salida	Descripción
stat.z	Valor normal estándar calculado para la diferencia de las proporciones
stat.ValP	Nivel más bajo de significancia en el cual la hipótesis nula se puede rechazar
stat. $\hat{p}1$	Estimación de proporción de primera muestra
stat. $\hat{p}2$	Estimación de proporción de segunda muestra
stat. \hat{p}	Estimación de proporción de muestras agrupadas
stat.n1, stat.n2	Número de muestras tomadas en las pruebas 1 y 2

zTest_2Samp (pruebaZ_2Muest)Catálogo > **zTest_2Samp** $\sigma_1, \sigma_2, Lista1, Lista2[, Frec1[, Frec2[, Hipot]]]$

(Entrada de lista de datos)

zTest_2Samp $\sigma_1, \sigma_2, \bar{x}1, n1, \bar{x}2, n2[, Hipot]$

(Entrada de estadísticas de resumen)

Resuelve una prueba Z de dos muestras. Un resumen de resultados se almacena en la variable *stat.results*. (Vea la página 122.)

Pruebe $H_0: \mu1 = \mu2$, contra uno de los siguientes:

Para $H_a: \mu1 < \mu2$, configure *Hipot*<0

Para $H_a: \mu1 \neq \mu2$ (predeterminado), configure *Hipot*=0

Para $H_a: \mu1 > \mu2$, *Hipot*>0

Para obtener información sobre el efecto de los elementos vacíos en una lista, vea "Elementos vacíos (inválidos)" en la página 166.

Variable de salida	Descripción
stat.z	Valor normal estándar computado para la diferencia de las medias
stat.ValP	Nivel más bajo de significancia en el cual la hipótesis nula se puede rechazar
stat. $\bar{x}1$, stat. $\bar{x}2$	Muestras de las medias de las secuencias de datos en <i>Lista1</i> y <i>Lista2</i>

Variable de salida	Descripción
stat.sx1, stat.sx2	Desviaciones estándar de muestras de las secuencias de datos en <i>Lista1</i> y <i>Lista2</i>
stat.n1, stat.n2	Tamaño de las muestras

Símbolos

+ (agregar)

 tecla

$Expr1 + Expr2 \Rightarrow$ expresión

Entrega la suma de los dos argumentos.

56	56
56+4	60
60+4	64
64+4	68
68+4	72

$Lista1 + Lista2 \Rightarrow$ lista

$Matriz1 + Matriz2 \Rightarrow$ matriz

Entrega una lista (o matriz) que contiene las sumas de los elementos correspondientes en $Lista1$ y $Lista2$ (o $Matriz1$ y $Matriz2$).

Las dimensiones de los argumentos deben ser iguales.

$\left\{ 22, \pi, \frac{\pi}{2} \right\} \rightarrow I1$	$\left\{ 22, \pi, \frac{\pi}{2} \right\}$
$\left\{ 10, 5, \frac{\pi}{2} \right\} \rightarrow I2$	$\left\{ 10, 5, \frac{\pi}{2} \right\}$
$I1+I2$	$\{ 32, \pi+5, \pi \}$
$Ans + \{ \pi, -5, \pi \}$	$\{ \pi+32, \pi, 0 \}$
$\begin{bmatrix} a & b \\ c & d \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$	$\begin{bmatrix} a+1 & b \\ c & d+1 \end{bmatrix}$

$Expr + Lista1 \Rightarrow$ lista

$Lista1 + Expr \Rightarrow$ lista

Entrega una lista que contiene las sumas de $Expr$ y cada elemento en $Lista1$.

$15 + \{ 10, 15, 20 \}$	$\{ 25, 30, 35 \}$
$\{ 10, 15, 20 \} + 15$	$\{ 25, 30, 35 \}$

$Expr + Matriz1 \Rightarrow$ matriz

$Matriz1 + Expr \Rightarrow$ matriz

Entrega una matriz con $Expr$ agregada a cada elemento en la diagonal de $Matriz1$. $Matriz1$ debe ser cuadrada.

Nota: Use $+$ (punto más) para agregar una expresión a cada elemento.

$20 + \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$	$\begin{bmatrix} 21 & 2 \\ 3 & 24 \end{bmatrix}$
---	--

- (sustraer)

 tecla

$Expr1 - Expr2 \Rightarrow$ expresión

Entrega $Expr1$ menos $Expr2$.

6-2	4
$\pi - \frac{\pi}{6}$	$\frac{5 \cdot \pi}{6}$

$Lista1 - Lista2 \Rightarrow$ lista

$Matriz1 - Matriz2 \Rightarrow$ matriz

Sustraer a cada elemento en $Lista2$ (o $Matriz2$) del elemento correspondiente en $Lista1$ (o $Matriz1$) y entrega los resultados.

Las dimensiones de los argumentos deben ser iguales.

$\left\{ 22, \pi, \frac{\pi}{2} \right\} - \left\{ 10, 5, \frac{\pi}{2} \right\}$	$\{ 12, \pi-5, 0 \}$
$\begin{bmatrix} 3 & 4 \end{bmatrix} - \begin{bmatrix} 1 & 2 \end{bmatrix}$	$\begin{bmatrix} 2 & 2 \end{bmatrix}$

$Expr - Lista1 \Rightarrow$ lista

$Lista1 - Expr \Rightarrow$ lista

Sustraer a cada elemento de $Lista1$ de $Expr$ o sustraer $Expr$ de cada elemento de $Lista1$ y entrega una lista de los resultados.

$15 - \{ 10, 15, 20 \}$	$\{ 5, 0, -5 \}$
$\{ 10, 15, 20 \} - 15$	$\{ -5, 0, 5 \}$

-(sustraer)

[- tecla]

 $Expr - Matriz1 \Rightarrow matriz$ $Matriz1 - Expr \Rightarrow matriz$

$Expr - Matriz1$ entrega una matriz de $Expr$ veces la matriz de identidad menos $Matriz1$. La $Matriz1$ debe ser cuadrada.

$Matriz1 - Expr$ entrega una matriz de $Expr$ veces la matriz de identidad sustraída de $Matriz1$. La $Matriz1$ debe ser cuadrada.

Nota: Use - (punto menos) para sustraer una expresión de cada elemento.

$$20 - \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \quad \begin{bmatrix} 19 & -2 \\ -3 & 16 \end{bmatrix}$$

·(multiplicar)

[x] tecla

 $Expr1 \cdot Expr2 \Rightarrow expresión$

Entrega el producto de los dos argumentos.

$$2 \cdot 3.45 \quad 6.9$$

$$x \cdot y \cdot x \quad x^2 \cdot y$$

 $Lista1 \cdot Lista2 \Rightarrow lista$

Entrega una lista que contiene los productos de los elementos correspondientes en $Lista1$ y $Lista2$.

Las dimensiones de las listas deben ser iguales.

$$\{1, 2, 3\} \cdot \{4, 5, 6\} \quad \{4, 10, 18\}$$

$$\left\{ \frac{2}{a}, \frac{3}{a} \right\} \cdot \left\{ a^2, \frac{b}{3} \right\} \quad \left\{ 2 \cdot a, \frac{b}{2} \right\}$$

 $Matriz1 \cdot Matriz2 \Rightarrow matriz$

Entrega el producto de la matriz de $Matriz1$ y $Matriz2$.

El número de columnas en $Matriz1$ debe igualar el número de filas en $Matriz2$.

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \cdot \begin{bmatrix} a & d \\ b & e \\ c & f \end{bmatrix}$$

$$\begin{bmatrix} a+2 \cdot b+3 \cdot c & d+2 \cdot e+3 \cdot f \\ 4 \cdot a+5 \cdot b+6 \cdot c & 4 \cdot d+5 \cdot e+6 \cdot f \end{bmatrix}$$

 $Expr \cdot Lista1 \Rightarrow lista$ $Lista1 \cdot Expr \Rightarrow lista$

Entrega una lista que contiene los productos de $Expr$ y cada elemento en $Lista1$.

$$\pi \cdot \{4, 5, 6\} \quad \{4 \cdot \pi, 5 \cdot \pi, 6 \cdot \pi\}$$

 $Expr \cdot Matriz1 \Rightarrow matriz$ $Matriz1 \cdot Expr \Rightarrow matriz$

Entrega una matriz que contiene los productos de $Expr$ y cada elemento en $Matriz1$.

Nota: Use · (punto multiplicar) para multiplicar una expresión por cada elemento.

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \cdot 0.01 \quad \begin{bmatrix} 0.01 & 0.02 \\ 0.03 & 0.04 \end{bmatrix}$$

$$1 \cdot \text{identity}(3) \quad \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

/ (dividir)

[÷] tecla

 $Expr1 / Expr2 \Rightarrow expresión$

Entrega el cociente de $Expr1$ dividido entre $Expr2$.

Nota: Vea también **Plantilla de fracciones**, página 1.

$$\frac{2}{3.45} \quad .57971$$

$$\frac{x^3}{x} \quad x^2$$

/ (dividir)

 **tecla**
 $Lista1 / Lista2 \Rightarrow lista$

Entrega una lista que contiene los cocientes de *Lista1* divididos entre *Lista2*.

$$\frac{\{1.,2,3\}}{\{4,5,6\}} \quad \left\{ 0,25, \frac{2}{5}, \frac{1}{2} \right\}$$

Las dimensiones de las listas deben ser iguales.

 $Expr / Lista1 \Rightarrow lista$
 $Lista1 / Expr \Rightarrow lista$

Entrega una lista que contiene los cocientes de *Expr* divididos entre *Lista1* o de *Lista1* divididos entre *Expr*.

$$\frac{a}{\{3,a,\sqrt{a}\}} \quad \left\{ \frac{a}{3}, 1, \sqrt{a} \right\}$$

$$\frac{\{a,b,c\}}{a \cdot b \cdot c} \quad \left\{ \frac{1}{b \cdot c}, \frac{1}{a \cdot c}, \frac{1}{a \cdot b} \right\}$$

 $Matriz1 / Expr \Rightarrow matriz$

Entrega una matriz que contiene los cocientes de *Matriz1/Expr*.

$$\frac{\begin{bmatrix} a & b & c \end{bmatrix}}{a \cdot b \cdot c} \quad \begin{bmatrix} \frac{1}{b \cdot c} & \frac{1}{a \cdot c} & \frac{1}{a \cdot b} \end{bmatrix}$$

Nota: Use . / (punto dividir) para dividir una expresión entre cada elemento.

^ (potencia)

 **tecla**
 $Expr1 \wedge Expr2 \Rightarrow expresión$
 $Lista1 \wedge Lista2 \Rightarrow lista$

Entrega el primer argumento elevado a la potencia del segundo argumento.

$$4^2 \quad 16$$

$$\{a,2,c\} \{1,b,3\} \quad \{a,2^b,c^3\}$$

Nota: Vea también **Plantilla de exponentes**, página 1.

Para una lista, entrega los elementos en *Lista1* elevados a la potencia de los elementos correspondientes en *Lista2*.

En el dominio real, las potencias fraccionarias que han reducido los exponentes con denominadores impares usan la rama real contra la rama principal para el modo complejo.

 $Expr \wedge Lista1 \Rightarrow lista$

Entrega *Expr* elevada a la potencia de los elementos en *Lista1*.

$$p \{a,2,-3\} \quad \left\{ p^a, p^2, \frac{1}{p^3} \right\}$$

 $Lista1 \wedge Expr \Rightarrow lista$

Entrega los elementos en *Lista1* elevados a la potencia de *Expr*.

$$\{1,2,3,4\}^{-2} \quad \left\{ 1, \frac{1}{4}, \frac{1}{9}, \frac{1}{16} \right\}$$

 $matriz:Cuadrada1 \wedge entero \Rightarrow matriz$

Entrega *matriz:Cuadrada1* elevada a la potencia del *entero*.

matriz:Cuadrada1 debe ser una matriz cuadrada.

Si *entero* = -1, resuelve la matriz inversa.

Si *entero* < -1, resuelve la matriz inversa a una potencia positiva apropiada.

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}^2 \quad \begin{bmatrix} 7 & 10 \\ 15 & 22 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}^{-1} \quad \begin{bmatrix} -2 & 1 \\ 3 & -1 \\ 2 & 2 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}^{-2} \quad \begin{bmatrix} \frac{11}{4} & -5 \\ 2 & 2 \\ -15 & 7 \\ 4 & 4 \end{bmatrix}$$

x² (cuadrado)**x² tecla***Expr*² ⇒ expresión

Entrega el cuadrado del argumento.

4^2	16
-------	----

*Lista*² ⇒ listaEntrega una lista que contiene los cuadrados de los elementos en la *Lista1*.

$\{2,4,6\}^2$	$\{4,16,36\}$
---------------	---------------

*matrizCuadrada*² ⇒ matrizEntrega el cuadrado de la matriz de *matrizCuadrada1*. Esto no es lo mismo que calcular el cuadrado de cada elemento. Use $\wedge 2$ para calcular el cuadrado de cada elemento.

$\begin{bmatrix} 2 & 4 & 6 \\ 3 & 5 & 7 \\ 4 & 6 & 8 \end{bmatrix}^2$	$\begin{bmatrix} 40 & 64 & 88 \\ 49 & 79 & 109 \\ 58 & 94 & 130 \end{bmatrix}$
---	--

$\begin{bmatrix} 2 & 4 & 6 \\ 3 & 5 & 7 \\ 4 & 6 & 8 \end{bmatrix} \wedge 2$	$\begin{bmatrix} 4 & 16 & 36 \\ 9 & 25 & 49 \\ 16 & 36 & 64 \end{bmatrix}$
--	--

+ (punto agregar)**+ teclas***Matriz1* + *Matriz2* ⇒ matriz*Expr* + *Matriz1* ⇒ matriz*Matriz1* + *Matriz2* entrega una matriz que es la suma de cada par de elementos correspondientes en *Matriz1* y *Matriz2*.*Expr* + *Matriz1* entrega una matriz que es la suma de *Expr* y cada elemento en *Matriz1*.

$\begin{bmatrix} a & 2 \\ b & 3 \end{bmatrix} + \begin{bmatrix} c & 4 \\ 5 & d \end{bmatrix}$	$\begin{bmatrix} a+c & 6 \\ b+5 & d+3 \end{bmatrix}$
---	--

$x + \begin{bmatrix} c & 4 \\ 5 & d \end{bmatrix}$	$\begin{bmatrix} x+c & x+4 \\ x+5 & x+d \end{bmatrix}$
--	--

- (punto sust.)**- teclas***Matriz1* - *Matriz2* ⇒ matriz*Expr* - *Matriz1* ⇒ matriz*Matriz1* - *Matriz2* entrega una matriz que es la diferencia entre cada par de elementos correspondientes en *Matriz1* y *Matriz2*.*Expr* - *Matriz1* entrega una matriz que es la diferencia de *Expr* y cada elemento en *Matriz1*.

$\begin{bmatrix} a & 2 \\ b & 3 \end{bmatrix} - \begin{bmatrix} c & 4 \\ d & 5 \end{bmatrix}$	$\begin{bmatrix} a-c & -2 \\ b-d & -2 \end{bmatrix}$
---	--

$x - \begin{bmatrix} c & 4 \\ d & 5 \end{bmatrix}$	$\begin{bmatrix} x-c & x-4 \\ x-d & x-5 \end{bmatrix}$
--	--

. (punto mult.)**. teclas***Matriz1* . *Matriz2* ⇒ matriz*Expr* . *Matriz1* ⇒ matriz*Matriz1* . *Matriz2* entrega una matriz que es el producto de cada par de elementos correspondientes en *Matriz1* y *Matriz2*.*Expr* . *Matriz1* entrega una matriz que contiene los productos de *Expr* y cada elemento en *Matriz1*.

$\begin{bmatrix} a & 2 \\ b & 3 \end{bmatrix} . \begin{bmatrix} c & 4 \\ 5 & d \end{bmatrix}$	$\begin{bmatrix} a \cdot c & 8 \\ 5 \cdot b & 3 \cdot d \end{bmatrix}$
---	--

$x . \begin{bmatrix} a & b \\ c & d \end{bmatrix}$	$\begin{bmatrix} a \cdot x & b \cdot x \\ c \cdot x & d \cdot x \end{bmatrix}$
--	--

% (porcentaje)ctrl  teclas $Expr1 \% \Rightarrow$ expresión $Lista1 \% \Rightarrow$ lista $Matriz1 \% \Rightarrow$ matrizEntrega $\frac{argument}{100}$

Para una lista o matriz, entrega una lista o matriz con cada elemento dividido entre 100.

Presione **Ctrl+Ingresar**   (Macintosh@:**%+Ingresar**) para evaluar: 13% 0.13 Presione **Ctrl+Ingresar**   (Macintosh@:**%+Ingresar**) para evaluar: $\{\{1,10,100\}\}\%$ $\{0.01,0.1,1.\}$ **= (igual)** teclas $Expr1 = Expr2 \Rightarrow$ expresión Booleana $Lista1 = Lista2 \Rightarrow$ lista Booleana $Matriz1 = Matriz2 \Rightarrow$ matriz BooleanaEntrega verdadero si $Expr1$ se determina como igual a $Expr2$.Entrega falso si $Expr1$ se determina como no igual a $Expr2$.

Cualquier otra cosa entrega una forma simplificada de la ecuación.

Para listas y matrices, entrega comparaciones elemento por elemento.

Nota para ingresar el ejemplo: En la aplicación de Calculadora en el dispositivo portátil, usted puede ingresar definiciones en líneas múltiples al presionar  en lugar de  al final de cada línea. En el teclado de la computadora, presione y sostenga **Alt** y presione **Ingresar**.

Ejemplo de función que usa símbolos de prueba matemática: =, ≠, <, ≤, >, ≥

Define $g(x)=Func$ If $x \leq -5$ Then

Return 5

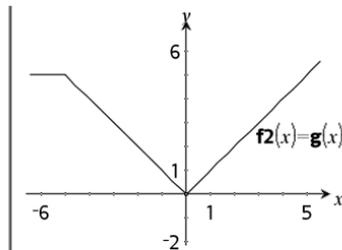
ElseIf $x > -5$ and $x < 0$ ThenReturn $-x$ ElseIf $x \geq 0$ and $x \neq 10$ ThenReturn x ElseIf $x = 10$ Then

Return 3

EndIf

EndFunc

Done

Resultado de graficar $g(x)$  $f2(x)=g(x)$

≠ (no igual)  **teclas** $Expr1 \neq Expr2 \Rightarrow$ expresión Booleana

Vea "=" (igual) ejemplo.

 $Lista1 \neq Lista2 \Rightarrow$ lista Booleana $Matriz1 \neq Matriz2 \Rightarrow$ matriz BooleanaEntrega verdadero si $Expr1$ se determina como no igual a $Expr2$.Entrega si $Expr1$ se determina como igual a $Expr2$.

Cualquier otra cosa entrega una forma simplificada de la ecuación.

Para listas y matrices, entrega comparaciones elemento por elemento.

Nota: Usted puede insertar este operador desde el teclado al escribir
/=**< (menor que)**  **teclas** $Expr1 < Expr2 \Rightarrow$ expresión Booleana

Vea "=" (igual) ejemplo.

 $Lista1 < Lista2 \Rightarrow$ lista Booleana $Matriz1 < Matriz2 \Rightarrow$ matriz BooleanaEntrega verdadero si $Expr1$ se determina como menor que $Expr2$.Entrega falso si $Expr1$ se determina como mayor que o igual a $Expr2$.

Cualquier otra cosa entrega una forma simplificada de la ecuación.

Para listas y matrices, entrega comparaciones elemento por elemento.

≤ (menor o igual)  **teclas** $Expr1 \leq Expr2 \Rightarrow$ expresión Booleana

Vea "=" (igual) ejemplo.

 $Lista1 \leq Lista2 \Rightarrow$ lista Booleana $Matriz1 \leq Matriz2 \Rightarrow$ matriz BooleanaEntrega verdadero si $Expr1$ se determina como menor que o igual a $Expr2$.Entrega falso si $Expr1$ se determina como mayor que $Expr2$.

Cualquier otra cosa entrega una forma simplificada de la ecuación.

Para listas y matrices, entrega comparaciones elemento por elemento.

Nota: Usted puede insertar este operador desde el teclado al escribir
<=**> (mayor que)**  **teclas** $Expr1 > Expr2 \Rightarrow$ expresión Booleana

Vea "=" (igual) ejemplo.

 $Lista1 > Lista2 \Rightarrow$ lista Booleana $Matriz1 > Matriz2 \Rightarrow$ matriz BooleanaEntrega verdadero si $Expr1$ se determina como mayor que $Expr2$.Entrega falso si $Expr1$ se determina como menor que o igual a $Expr2$.

Cualquier otra cosa entrega una forma simplificada de la ecuación.

Para listas y matrices, entrega comparaciones elemento por elemento.

≥ (mayor o igual)

ctrl [=] teclas

 $Expr1 \geq Expr2 \Rightarrow$ expresión Booleana

Vea "=" (igual) ejemplo.

 $Lista1 \geq Lista2 \Rightarrow$ lista Booleana $Matriz1 \geq Matriz2 \Rightarrow$ matriz BooleanaEntrega verdadero si $Expr1$ se determina como mayor que o igual a $Expr2$.Entrega falso si $Expr1$ se determina como menor que $Expr2$.

Cualquier otra cosa entrega una forma simplificada de la ecuación.

Para listas y matrices, entrega comparaciones elemento por elemento.

Nota: Usted puede insertar este operador desde el teclado al escribir \geq **\Rightarrow (implicación lógica)**

teclas ctrl [=]

 $BooleanaExpr1 \Rightarrow BooleanaExpr2$ devuelve expresión booleana $5 > 3$ or $3 > 5$

true

 $BooleanaLista1 \Rightarrow BooleanaLista2$ devuelve lista booleana $BooleanaMatriz1 \Rightarrow BooleanaMatriz2$ devuelve matriz booleana $5 > 3 \Rightarrow 3 > 5$

false

 $Entero1 \Rightarrow Entero2$ devuelve Entero 3 or 4

7

Evalúa la expresión **not** <argumento1> or <argumento2> y devuelve verdadero, falso o una forma simplificada de la ecuación. $3 \Rightarrow 4$

-4

Para listas y matrices, devuelve comparaciones elemento por elemento.

 $\{1,2,3\}$ or $\{3,2,1\}$ $\{3,2,3\}$ $\{1,2,3\} \Rightarrow \{3,2,1\}$ $\{-1,-1,-3\}$ **Nota:** Puede insertar este operador con el teclado al escribir \Rightarrow **\Leftrightarrow (implicación doble lógica, XNOR)**

teclas [=] ctrl

 $BooleanaExpr1 \Leftrightarrow BooleanaExpr2$ devuelve expresión booleana $5 > 3$ xor $3 > 5$

true

 $BooleanaLista1 \Leftrightarrow BooleanaLista2$ devuelve lista booleana $BooleanaMatriz1 \Leftrightarrow BooleanaMatriz2$ devuelve matriz booleana $5 > 3 \Leftrightarrow 3 > 5$

false

 $Entero1 \Leftrightarrow Entero2$ devuelve Entero 3 xor 4

7

Devuelve la negación de una **XOR** operación booleana en los dos argumentos. Devuelve verdadero, falso o una forma simplificada de la ecuación. $3 \Leftrightarrow 4$

-8

Para listas y matrices, devuelve comparaciones elemento por elemento.

 $\{1,2,3\}$ xor $\{3,2,1\}$ $\{2,0,2\}$ $\{1,2,3\} \Leftrightarrow \{3,2,1\}$ $\{-3,-1,-3\}$ **Nota:** Puede insertar este operador con el teclado al escribir \Leftrightarrow **! (factorial)**

[?!] tecla

 $Expr1! \Rightarrow$ expresión $5!$

120

 $Lista1! \Rightarrow$ lista $Matriz1! \Rightarrow$ matriz $\{\{5,4,3\}\}!$ $\{120,24,6\}$

Entrega el factorial del argumento.

Para una lista o matriz, entrega una lista o una matriz de factoriales de los elementos.

 $\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}!$ $\begin{bmatrix} 1 & 2 \\ 6 & 24 \end{bmatrix}$ **& (adjuntar)**

ctrl [&] teclas

 $Cadena1 \& Cadena2 \Rightarrow$ cadena

"Hello " & "Nick"

"Hello Nick"

Entrega una cadena de texto que es $Cadena2$ adjuntada a $Cadena1$.

d() (derivada)Catálogo >  $d(\text{Expr1}, \text{Var}, \text{Orden}) \Rightarrow$ expresión $d(\text{Lista1}, \text{Var}, \text{Orden}) \Rightarrow$ lista $d(\text{Matriz1}, \text{Var}, \text{Orden}) \Rightarrow$ matriz

Entrega la primera derivada del primer argumento con respecto de la variable *Var*.

Orden, si se incluye, debe ser un entero. Si el orden es menor que cero, el resultado será una antiderivada.

Nota: Usted puede insertar esta función desde el teclado al escribir **derivative** (...).

d() no sigue el mecanismo de evaluación normal de simplificar completamente sus argumentos y luego aplicar la definición de función a estos argumentos completamente simplificados. En su lugar, **d()** realiza los siguientes pasos:

1. Simplificar el segundo argumento sólo hasta el punto en que no conlleva a una *n* variable.
2. Simplificar el primer argumento sólo hasta el punto en que no recupera ningún valor almacenado para la variable determinada por medio del paso 1.
3. Determinar la derivada simbólica del resultado del paso 2 con respecto de la variable del paso 1.

Si la variable del paso 1 tiene un valor almacenado o un valor especificado por el operador restrictivo ("|"), sustituya dicho valor en el resultado del paso 3.

Nota: Vea también **Primera derivada**, página 5;
Segunda derivada, página 5 o **N-ésima derivada**, página 5.

$$\frac{d}{dx}(f(x) \cdot g(x)) = \frac{d}{dx}(f(x)) \cdot g(x) + \frac{d}{dx}(g(x)) \cdot f(x)$$

$$\frac{d}{dy} \left(\frac{d}{dx}(x^2 \cdot y^3) \right) = 6 \cdot y^2 \cdot x$$

$$\frac{d}{dx} \left(\left\{ x^2, x^3, x^4 \right\} \right) = \left\{ 2 \cdot x, 3 \cdot x^2, 4 \cdot x^3 \right\}$$

f() (integral)Catálogo >  $f(\text{Expr1}, \text{Var}, \text{Baja}, \text{Alta}) \Rightarrow$ expresión $f(\text{Expr1}, \text{Var}, \text{Constante}) \Rightarrow$ expresión

Entrega la integral de *Expr1* con respecto de la variable *Var* de *Baja* a *Alta*.

Nota: Vea también **Plantilla de integral definida o indefinida**, página 5.

Nota: Usted puede insertar esta función desde el teclado al escribir **integral** (...).

Si se omiten *Baja* y *Alta*, entrega una antiderivada. Se omite una constante simbólica de integración, a menos que usted proporcione el argumento de la *Constante*.

$$\int_a^b x^2 dx = \frac{b^3}{3} - \frac{a^3}{3}$$

$$\int x^2 dx = \frac{x^3}{3}$$

$$f(a \cdot x^2, x, c) = \frac{a \cdot x^3}{3} + c$$

Las antiderivadas igualmente válidas podrían diferir por una constante numérica. Dicha constante podría estar oculta, en particular cuando una antiderivada contiene logaritmos o funciones trigonométricas inversas. Por otra parte, las expresiones constantes de compuesto de variables en ocasiones se agregan para hacer válida una antiderivada sobre un intervalo más grande que la fórmula usual.

$\int()$ (integral)

Catálogo >

$\int()$ se entrega a sí mismo para piezas de *Expr1* que no puede determinar como una combinación finita explícita de sus funciones y operadores integrados.

Cuando usted proporciona *Baja* y *Alta*, se hace un intento de localizar cualquier discontinuidad o derivada discontinua en el intervalo $Baja < Var < Alta$ y de subdividir el intervalo en esos lugares.

Para la configuración de Auto del modo **Auto o Aproximado**, se usa la integración numérica donde es aplicable cuando no se puede determinar una antiderivada o un límite.

Para la configuración de Aproximado, primero se intenta la integración numérica, si aplica. Las antiderivadas se buscan sólo donde dicha integración numérica no es aplicable o falla.

$$\int b \cdot e^{-x^2} + \frac{a}{x^2+a^2} dx \quad b \cdot \int e^{-x^2} dx + \tan^{-1}\left(\frac{x}{a}\right)$$

Presione **Ctrl+Ingresar** (Macintosh®: **⌘+Ingresar**) para evaluar:

$$\int_{-1}^1 e^{-x^2} dx \quad 1.49365$$

$\int()$ se puede anidar para hacer integrales múltiples. Los límites de la integración pueden depender de las variables de integración afuera de los mismos.

Nota: Vea también **nInt()**, página 83.

$$\int_0^a \int_0^x \ln(x+y) dy dx \quad \frac{a^2 \cdot \ln(a)}{2} + a^2 \cdot \left(\ln(2) - \frac{3}{4} \right)$$

 $\sqrt{}$ (raíz cuadrada) **teclas**

$\sqrt{}$ (*Expr1*) \Rightarrow expresión

$\sqrt{}$ (*Lista1*) \Rightarrow lista

Entrega la raíz cuadrada del argumento.

Para una lista, entrega las raíces cuadradas de todos los elementos en *Lista1*.

Nota: Usted puede insertar esta función desde el teclado al escribir **sqrt** (...).

Nota: Vea también **Plantilla de raíz cuadrada**, página 1.

$$\sqrt{4} \quad 2$$

$$\sqrt{\{9,4\}} \quad \{3,\sqrt{4}\}$$

$\Pi()$ (secProd)Catálogo >  $\Pi(\text{Expr1}, \text{Var}, \text{Baja}, \text{Alta}) \Rightarrow$ expresión**Nota:** Usted puede insertar esta función desde el teclado al escribir `prodSeq` (...).Evalúa *Expr1* para cada valor de *Var* de *Baja* a *Alta* entrega el producto de los resultados.**Nota:** Vea también **Plantilla de producto (Π)**, página 4.

$$\prod_{n=1}^5 \left(\frac{1}{n} \right) = \frac{1}{120}$$

$$\prod_{k=1}^n (k^2) = (n!)^2$$

$$\prod_{n=1}^5 \left\{ \left(\frac{1}{n}, n, 2 \right) \right\} = \left\{ \frac{1}{120}, 120, 32 \right\}$$

 $\Pi(\text{Expr1}, \text{Var}, \text{Baja}, \text{Baja}-1) \Rightarrow 1$ $\Pi(\text{Expr1}, \text{Var}, \text{Baja}, \text{Alta})$ $\Rightarrow \mathbf{1/\Pi}(\text{Expr1}, \text{Var}, \text{Alta}+1, \text{Baja}-1)$ if $\text{Alta} < \text{Baja}-1$

Las fórmulas del producto utilizadas se derivan de la siguiente referencia:

Ronald L. Graham, Donald E. Knuth y Oren Patashnik. *Matemáticas Concretas: Una Fundación para las Ciencias de la Computación*. Lectura, Massachusetts: Addison-Wesley, 1994.

$$\prod_{k=4}^3 (k) = 1$$

$$\prod_{k=4}^1 \left(\frac{1}{k} \right) = 6$$

$$\prod_{k=4}^1 \left(\frac{1}{k} \right) \cdot \prod_{k=2}^4 \left(\frac{1}{k} \right) = \frac{1}{4}$$

 $\Sigma()$ (secSuma)Catálogo >  $\Sigma(\text{Expr1}, \text{Var}, \text{Baja}, \text{Alta}) \Rightarrow$ expresión**Nota:** Usted puede insertar esta función desde el teclado al escribir `secSuma` (...).Evalúa *Expr1* para cada valor de *Var* de *Baja* a *Alta* entrega la suma de los resultados.**Nota:** Vea también **Plantilla de suma**, página 4.

$$\sum_{n=1}^5 \left(\frac{1}{n} \right) = \frac{137}{60}$$

$$\sum_{k=1}^n (k^2) = \frac{n \cdot (n+1) \cdot (2 \cdot n+1)}{6}$$

$$\sum_{n=1}^{\infty} \left(\frac{1}{n^2} \right) = \frac{\pi^2}{6}$$

$\Sigma()$ (secSuma)Catálogo >  $\Sigma(\text{Expr1}, \text{Var}, \text{Baja}, \text{Alta}-1) \Rightarrow 0$ $\Sigma(\text{Expr1}, \text{Var}, \text{Baja}, \text{Alta})$ $\Rightarrow -\Sigma(\text{Expr1}, \text{Var}, \text{Alta}+1, \text{Baja}-1)$ si $\text{Alta} < \text{Baja}-1$

Las fórmulas de la sumatoria utilizadas se derivan de la siguiente referencia:

Ronald L. Graham, Donald E. Knuth y Oren Patashnik. *Matemáticas Concretas: Una Fundación para las Ciencias de la Computación*. Lectura, Massachusetts: Addison-Wesley, 1994.

$$\sum_{k=4}^3 (k) \quad 0$$

$$\sum_{k=4}^1 (k) \quad -5$$

$$\sum_{k=4}^1 (k) + \sum_{k=2}^4 (k) \quad 4$$

 $\Sigma\text{Int}()$ Catálogo > 

$\Sigma\text{Int}(\text{NPgo1}, \text{NPgo2}, N, I, \text{VP}, \text{Pgo}, \text{VF}, \text{PpA}, \text{CpA}, \text{PgoAl}, \text{PgoAl}, \text{IvalorRedondo}) \Rightarrow \text{valor}$

$\Sigma\text{Int}(\text{NPgo1}, \text{NPgo2}, \text{tablaAmort}) \Rightarrow \text{valor}$

La función de amortización que calcula la suma del interés durante un rango de pagos específico.

NPgo1 y NPgo2 definen los límites iniciales y finales del rango de pagos.

$N, I, \text{VP}, \text{Pgo}, \text{VF}, \text{PpA}, \text{CpA}, \text{PgoAl}$ se describen en la tabla de argumentos de VTD, página 135.

- Si se omite Pgo , se predetermina a $\text{Pgo} = \text{tvmPmt}(N, I, \text{VP}, \text{VF}, \text{PpA}, \text{CpA}, \text{PgoAl})$.
- Si se omite VF , se predetermina a $\text{VF} = 0$.
- Los predeterminados para $\text{PpA}, \text{CpA}, \text{PgoAl}$ son los mismos que para las funciones de VTD.

IvalorRedondo especifica el número de lugares decimales para el redondeo. Predeterminado=2.

$\Sigma\text{Int}(\text{NPgo1}, \text{NPgo2}, \text{tablaAmort})$ calcula la suma del interés con base en la tabla de amortización tablaAmort . El argumento tablaAmort debe ser una matriz en la forma descrita bajo $\text{amortTbl}()$, página 7.

Nota: Vea también $\Sigma\text{Prn}()$, abajo y $\text{Bal}()$, página 13.

$\Sigma\text{Int}(1, 3, 12, 4.75, 20000, , 12, 12) \quad -213.48$

$\text{tbl} := \text{amortTbl}(12, 12, 4.75, 20000, , 12, 12)$

0	0.	0.	20000.
1	-77.49	-1632.43	18367.6
2	-71.17	-1638.75	16728.8
3	-64.82	-1645.1	15083.7
4	-58.44	-1651.48	13432.2
5	-52.05	-1657.87	11774.4
6	-45.62	-1664.3	10110.1
7	-39.17	-1670.75	8439.32
8	-32.7	-1677.22	6762.1
9	-26.2	-1683.72	5078.38
10	-19.68	-1690.24	3388.14
11	-13.13	-1696.79	1691.35
12	-6.55	-1703.37	-12.02

$\Sigma\text{Int}(1, 3, \text{tbl}) \quad -213.48$

ΣPrn() (ΣCap)Catálogo 

$\Sigma\text{Prn}(NPgo1, NPgo2, N, I, VP, [Pgo], [VF], [PpA], [CpA], [PgoA], [valorRedondo]) \Rightarrow \text{valor}$

$\Sigma\text{Prn}(NPgo1, NPgo2, \text{tablaAmort}) \Rightarrow \text{valor}$

La función de amortización que calcula la suma del capital durante un rango de pagos específico.

$NPgo1$ y $NPgo2$ definen los límites iniciales y finales del rango de pagos.

$N, I, VP, Pgo, VF, PpA, CpA, PgoA$ se describen en la tabla de argumentos de VTD, página 135.

- Si se omite Pgo , se predetermina a $Pgo = \text{tvmPmt}(N, I, VP, VF, PpA, CpA, PgoA)$.
- Si se omite VF , se predetermina a $VF = 0$.
- Los predeterminados para $PpA, CpA, PgoA$ son los mismos que para las funciones de VTD.

valorRedondo especifica el número de lugares decimales para el redondeo. Predeterminado=2.

$\Sigma\text{Prn}(NPgo1, NPgo2, \text{tablaAmort})$ calcula la suma del interés con base en la tabla de amortización tablaAmort . El argumento tablaAmort debe ser una matriz en la forma descrita bajo **amortTbl()**, página 7.

Nota: Vea también **ΣInt()**, arriba y **Bal()**, página 13.

$$\Sigma\text{Prn}(1, 3, 12, 4.75, 20000, 12, 12) \quad -4916.28$$

$$\text{tbl} := \text{amortTbl}(12, 12, 4.75, 20000, 12, 12)$$

0	0.	0.	20000.
1	-77.49	-1632.43	18367.57
2	-71.17	-1638.75	16728.82
3	-64.82	-1645.1	15083.72
4	-58.44	-1651.48	13432.24
5	-52.05	-1657.87	11774.37
6	-45.62	-1664.3	10110.07
7	-39.17	-1670.75	8439.32
8	-32.7	-1677.22	6762.1
9	-26.2	-1683.72	5078.38
10	-19.68	-1690.24	3388.14
11	-13.13	-1696.79	1691.35
12	-6.55	-1703.37	-12.02

$$\Sigma\text{Prn}(1, 3, \text{tbl}) \quad -4916.28$$

(indirección)ctrl  teclas

$\# \text{cadenaNomVar}$

Se refiere a la variable cuyo nombre es cadenaNomVar . Esto le permite usar cadenas para crear nombres de variable dentro de una función.

$$\#("x" \& "y" \& "z") \quad xyz$$

Crea o se refiere a la variable xyz.

$$10 \rightarrow r \quad 10$$

$$"r" \rightarrow sI \quad "r"$$

$$\#sI \quad 10$$

Entrega el valor de la variable (r) cuyo nombre se almacena en la variable s1.

E (notación científica) teclas

mantisaEexponente

Ingresa un número en la notación científica. El número se interpreta como $\text{mantisa} \times 10^{\text{exponente}}$.

Sugerencia: Si usted desea ingresar una potencia de 10 sin causar un resultado de valor decimal, use 10^{entero} .

Nota: Usted puede insertar este operador desde el teclado de la computadora al escribir @E. Por ejemplo, escriba 2.3@E4 para ingresar 2.3E4.

$$23000. \quad 23000.$$

$$2300000000. + 4.1E15 \quad 4.1E15$$

$$3 \cdot 10^4 \quad 30000$$

g (gradián) **tecla** $Expr1^g \Rightarrow$ expresión $Lista1^g \Rightarrow$ lista $Matriz1^g \Rightarrow$ matriz

Esta función le proporciona una manera de especificar un ángulo en gradianes mientras está en el modo de Grados o Radianes.

En el modo de ángulo en Radianes, multiplica $Expr1$ por $\pi/200$.

En el modo de ángulo en Grados, multiplica $Expr1$ por $g/100$.

En el modo de Gradianes, entrega $Expr1$ sin cambios.

Nota: Usted puede insertar este símbolo desde el teclado de la computadora al escribir @g.

En modo de Grados, Gradianes o Radianes:

$$\frac{\cos(50^g)}{\frac{\sqrt{2}}{2}}$$

$$\frac{\cos(\{0, 100^g, 200^g\})}{\{1, 0, -1\}}$$

r (radián) **tecla** $Expr1^r \Rightarrow$ expresión $Lista1^r \Rightarrow$ lista $Matriz1^r \Rightarrow$ matriz

Esta función le proporciona una manera de especificar un ángulo en radianes mientras está en el modo de Grados o Gradianes.

En el modo de ángulo en Grados, multiplica el argumento por $180/\pi$.

En el modo de ángulo en Radianes, entrega el argumento sin cambios.

En el modo de Gradianes, multiplica el argumento por $200/\pi$.

Sugerencia: Use r si usted desea forzar los radianes en una definición de función independientemente del modo que prevalece cuando se usa la función.

Nota: Usted puede insertar este símbolo desde el teclado de la computadora al escribir @x.

En modo de ángulo en Grados, Gradianes o Radianes:

$$\frac{\cos\left(\frac{\pi}{4^r}\right)}{\frac{\sqrt{2}}{2}}$$

$$\frac{\cos\left(\left\{0^r, \frac{\pi}{12}r, -(\pi)^r\right\}\right)}{\left\{1, \frac{(\sqrt{3+1})\sqrt{2}}{4}, -1\right\}}$$

o (grado) **tecla** $Expr1^o \Rightarrow$ expresión $Lista1^o \Rightarrow$ lista $Matriz1^o \Rightarrow$ matriz

Esta función le proporciona una manera de especificar un ángulo en grados mientras está en el modo de Gradianes o Radianes.

En el modo de ángulo en Radianes, multiplica el argumento por $\pi/180$.

En el modo de ángulo en Grados, entrega el argumento sin cambios.

En el modo de ángulo en Gradianes, multiplica el argumento por $10/9$.

Nota: Usted puede insertar este símbolo desde el teclado de la computadora al escribir @d.

En modo de ángulo en Grados, Gradianes o Radianes:

$$\frac{\cos(45^o)}{\frac{\sqrt{2}}{2}}$$

En modo de ángulo en Radianes:

Presione **Ctrl+Ingresar**  **enter** (Macintosh@: **⌘+Ingresar**) para evaluar:

$$\frac{\cos\left(\left\{0, \frac{\pi}{4}, 90^o, 30.12^o\right\}\right)}{\{1, .0.707107, 0., .0.864976\}}$$

° , ' , '' (grado/minuto/segundo)

ctrl teclas

gg°mm'ss.ss'' ⇒ expresión

En modo de ángulo en Grados:

gg Un número positivo o negativo

$$25^{\circ}13'17.5'' \quad 25.2215$$

mm Un número no negativo

$$25^{\circ}30' \quad \frac{51}{2}$$

ss.ss Un número no negativo

Entrega $gg+(mm/60)+(ss.ss/3600)$.

Este formato de ingreso de base-60 le permite:

- Ingresar un ángulo en grados/minutos/segundos sin importar le modo de ángulo actual.
- Ingrese el tiempo como horas/minutos/segundos.

Nota: Siga ss.ss con dos apóstrofes (''), no con el símbolo de comillas (").

∠ (ángulo)

ctrl teclas

[Radio,∠θ_Ángulo] ⇒ vector
(entrada polar)

En el modo de Radianes y en el formato del vector configure a: rectangular

[Radio,∠θ_Ángulo,Z_Coordenada] ⇒ vector
(entrada cilíndrica)

$$\left[5 \angle 60^{\circ} \angle 45^{\circ} \right] \quad \left[\frac{5 \cdot \sqrt{2}}{4} \quad \frac{5 \cdot \sqrt{6}}{4} \quad \frac{5 \cdot \sqrt{2}}{2} \right]$$

[Radio,∠θ_Ángulo,∠θ_Ángulo] ⇒ vector
(entrada esférica)

cilíndrico

Entrega las coordenadas como un vector dependiendo de la configuración del modo del Formato del Vector: rectangular, cilíndrica o esférica.

$$\left[5 \angle 60^{\circ} \angle 45^{\circ} \right] \quad \left[\frac{5 \cdot \sqrt{2}}{2} \quad \angle \frac{\pi}{3} \quad \frac{5 \cdot \sqrt{2}}{2} \right]$$

Nota: Usted puede insertar este símbolo desde el teclado de la computadora al escribir @<.

esférico

$$\left[5 \angle 60^{\circ} \angle 45^{\circ} \right] \quad \left[5 \angle \frac{\pi}{3} \angle \frac{\pi}{4} \right]$$

(Magnitud ∠_Ángulo) ⇒ valorComplejo
(entrada polar)

En el modo de ángulo en Radianes y el formato complejo Rectangular:

Ingresar un valor complejo en la forma polar ($r\angle\theta$). El *Ángulo* se interpreta de acuerdo con la configuración del modo del *Ángulo* actual.

$$5+3 \cdot i \left(10 \angle \frac{\pi}{4} \right) \quad 5-5 \cdot \sqrt{2} + (3-5 \cdot \sqrt{2}) \cdot i$$

Presione **Ctrl+Ingresar** (Macintosh@):

⌘+Ingresar para evaluar:

$$5+3 \cdot i \left(10 \angle \frac{\pi}{4} \right) \quad -2.07107-4.07107 \cdot i$$

' (primo)

tecla

variable '
variable ''

Ingresar un símbolo primo en una ecuación diferencial. Un símbolo primo sencillo denota una ecuación diferencial de 1er grado, dos símbolos primos denotan una de 2o grado, y así sucesivamente.

$$\text{deSolve}\left(y''=y^2 \text{ and } y(0)=0 \text{ and } y'(0)=0,t,y\right) \quad \frac{2 \cdot y^{\frac{4}{3}}}{3} = t$$

_ (guión bajo como designador de unidad)

ctrl  teclas*Expr_Unidad*

Designa las unidades para una *Expr*. Todos los nombres de unidad deben comenzar con un guión bajo.

Usted puede usar unidades predefinidas o crear sus propias unidades. Para una lista de unidades predefinidas, abra el Catálogo y despliegue la pestaña de Conversiones de Unidades. Usted puede seleccionar nombres de unidades desde el Catálogo o escribir los nombres de unidades directamente.

Variable_

Cuando la *Variable* no tiene ningún valor, se trata como si representara un número complejo. En forma predeterminada, sin el *_*, la variable se trata como real.

Si la *Variable* tiene un valor, el *_* se ignora y la *Variable* retiene su tipo de datos original.

Nota: Usted puede almacenar un número complejo para una variable sin usar *_*. Sin embargo, para obtener mejores resultados en los cálculos como **cSolve()** y **cZeros()**, se recomienda el *_*.

$$3 \cdot _m \blacktriangleright _ft \qquad 9.84252 \cdot _ft$$

Nota: Usted puede encontrar el símbolo de conversión, \blacktriangleright , en el

Catálogo. Haga clic en  y luego haga clic en

Operadores Matemáticos.

Supongamos que *z* es indefinido:

$\text{real}(z)$	z
$\text{real}(z_)$	$\text{real}(z_)$
$\text{imag}(z)$	0
$\text{imag}(z_)$	$\text{imag}(z_)$

 \blacktriangleright (convertir)ctrl  teclas

$$\text{Expr_Unidad1} \blacktriangleright _Unidad2 \Rightarrow \text{Expr_Unidad2}$$

Convierte una expresión de una unidad a otra.

El carácter de guión bajo *_* designa las unidades. Las unidades deben estar en la misma categoría, como Longitud o Área.

Para una lista de unidades predefinidas, abra el Catálogo y despliegue la pestaña de Conversiones de Unidades:

- Usted puede seleccionar un nombre de unidad desde la lista.
- Usted puede seleccionar el operador de conversión, \blacktriangleright , desde la parte superior de la lista.

Usted también puede escribir los nombres de unidades manualmente. Para escribir "*_*" cuando escriba nombres de unidades en el dispositivo portátil, presione  .

Nota: Para convertir unidades de temperatura, use **tmpCnv()** y **Δ tmpCnv()**. El operador de conversión \blacktriangleright no maneja unidades de temperatura.

$$3 \cdot _m \blacktriangleright _ft \qquad 9.84252 \cdot _ft$$

 $10^{\wedge}()$ Catálogo > 

$$10^{\wedge}(\text{Expr1}) \Rightarrow \text{expresión}$$

$$10^{\wedge}(\text{Lista1}) \Rightarrow \text{lista}$$

Entrega 10 elevado a la potencia del argumento.

Para una lista, entrega 10 elevado a la potencia de los elementos en *Lista1*.

$10^{1.5}$	31.6228
$10^{\{0, -2.2, a\}}$	$\left\{1, \frac{1}{100}, 100, 10^a\right\}$

10^()

Catálogo > 

$10^{(matrizCuadrada1)} \Rightarrow matrizCuadrada$

Entrega 10 elevado a la potencia de *matrizCuadrada1*. Esto no es lo mismo que calcular 10 elevado a la potencia de cada elemento. Para obtener información acerca del método de cálculo, consulte **cos()**.

matrizCuadrada1 debe ser diagonalizable. El resultado siempre contiene números de punto flotante.

$$10^{\begin{bmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{bmatrix}}$$

$$\begin{bmatrix} 1.14336\text{E}7 & 8.17155\text{E}6 & 6.67589\text{E}6 \\ 9.95651\text{E}6 & 7.11587\text{E}6 & 5.81342\text{E}6 \\ 7.65298\text{E}6 & 5.46952\text{E}6 & 4.46845\text{E}6 \end{bmatrix}$$

^-1(recíproco)

Catálogo > 

$Expr1^{-1} \Rightarrow expresión$

$Lista1^{-1} \Rightarrow lista$

Entrega el recíproco del argumento.

Para una lista, entrega los recíprocos de los elementos en *Lista1*.

$$(3.1)^{-1} \quad 0.322581$$

$$\{a, 4, -0.1, x, -2\}^{-1} \quad \left\{ \frac{1}{a}, \frac{1}{4}, -10., \frac{1}{x}, \frac{-1}{2} \right\}$$

$matrizCuadrada1^{-1} \Rightarrow matrizCuadrada$

Entrega el inverso de *matrizCuadrada*.

matrizCuadrada1 debe ser una matriz cuadrada no singular.

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}^{-1} \quad \begin{bmatrix} -2 & 1 \\ 3 & -1 \\ 2 & 2 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 2 \\ a & 4 \end{bmatrix}^{-1} \quad \begin{bmatrix} -2 & 1 \\ a-2 & a-2 \\ a & -1 \\ 2 \cdot (a-2) & 2 \cdot (a-2) \end{bmatrix}$$

| (operador restrictivo)

teclas  

$Expr | BooleanaExpr1 [\text{and } BooleanaExpr2]...$

$Expr | BooleanaExpr1 [\text{or } BooleanaExpr2]...$

El símbolo de restricción ("|") funciona como un operador binario. El operando a la izquierda de | es una expresión. El operando a la derecha de | especifica una o más relaciones que deben afectar la simplificación de la expresión. Las relaciones múltiples luego de | deben estar unidas por "and" lógica u operadores "or".

El operador restrictivo proporciona tres funciones básicas:

- Sustituciones
- Restricciones de intervalos
- Exclusiones

Las sustituciones tienen la forma de una igualdad, tal como $x=3$ o $y=\sin(x)$. Para ser más efectiva, el lado izquierdo debe ser una variable simple. $Expr | Variable = el\ valor$ sustituirá el valor para cada ocurrencia de la *Variable* en la *Expr*.

$$\frac{x+1|x=3}{4}$$

$$\frac{x+y|x=\sin(y)}{\sin(y)+y}$$

$$\frac{x+y|\sin(y)=x}{x+y}$$

$$\frac{x^3-2 \cdot x+7 \rightarrow f(x)}{Done}$$

$$\frac{f(x)|x=\sqrt{3}}{\sqrt{3}+7}$$

$$\frac{(\sin(x))^2+2 \cdot \sin(x)-6|\sin(x)=d}{d^2+2 \cdot d-6}$$

(operador restrictivo)

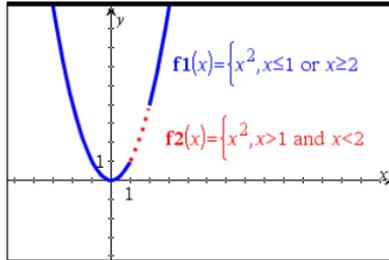
teclas **ctrl**

Las restricciones de intervalo tienen la forma de una o más desigualdades unidas por "and" lógica u operadores "or". Las restricciones de intervalo también permite la simplificación que de otro modo sería inválida o no computable.

$$\text{solve}(x^2-1=0,x)|x>0 \text{ and } x<2 \quad x=1$$

$$\sqrt{x} \cdot \sqrt{\frac{1}{x}} | x > 0 \quad 1$$

$$\sqrt{x} \cdot \sqrt{\frac{1}{x}} \quad \sqrt{\frac{1}{x}} \cdot \sqrt{x}$$



Las exclusiones utilizan el operador relacional "distinto" (\neq) para no tener en cuenta un valor específico. Se utilizan principalmente para excluir una solución exacta al utilizar las funciones **cSolución()**, **cCeros()**, **fMax()**, **fMin()**, **solución()**, **ceros()**, etc.

$$\text{solve}(x^2-1=0,x)|x \neq 1 \quad x=-1$$

→ (almacenar)

teclas **ctrl** **var** **tecla**

Expr → Var

Lista → Var

Matriz → Var

Expr → Función(Parám1,...)

Lista → Función(Parám1,...)

Matriz → Función(Parám1,...)

Si la variable *Var* no existe, la crea y la inicializa para *Expr*, *Lista* o *Matriz*.

Si la variable *Var* ya existe y no está bloqueada o protegida, reemplaza sus contenidos con *Expr*, *Lista* o *Matriz*.

Sugerencia: Si usted planea hacer cálculos simbólicos al usar variables indefinidas, evite almacenar cualquier cosa en las variables de una letra utilizadas comúnmente como a, b, c, x, y, z, y así sucesivamente.

Nota: Usted puede insertar este operador desde el teclado al escribir =: como un acceso directo. Por ejemplo, escriba **pi/4=: myvar**.

$$\frac{\pi}{4} \rightarrow \text{myvar} \quad \frac{\pi}{4}$$

$$2 \cdot \cos(x) \rightarrow y1(x) \quad \text{Done}$$

$$\{1,2,3,4\} \rightarrow \text{lst5} \quad \{1,2,3,4\}$$

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \rightarrow \text{matg} \quad \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$$

$$\text{"Hello"} \rightarrow \text{str1} \quad \text{"Hello"}$$

:= (asignar)ctrl  teclas

Var := Expr
 Var := Lista
 Var := Matriz
 Función(Parám1,...) := Expr
 Función(Parám1,...) := Lista
 Función(Parám1,...) := Matriz

Si la variable *Var* no existe, crea *Var* y la inicializa para *Expr*, *Lista* o *Matriz*.

Si *Var* ya existe y no está bloqueada o protegida, reemplaza sus contenidos con *Expr*, *Lista* o *Matriz*.

Sugerencia: Si usted planea hacer cálculos simbólicos al usar variables indefinidas, evite almacenar cualquier cosa en las variables de una letra utilizadas comúnmente como a, b, c, x, y, z, y así sucesivamente.

$myvar := \frac{\pi}{4}$	$\frac{\pi}{4}$
$y1(x) := 2 \cdot \cos(x)$	Done
$lst5 := \{1, 2, 3, 4\}$	$\{1, 2, 3, 4\}$
$matg := \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$	$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$
$str1 := "Hello"$	"Hello"

Ⓞ (comentario)ctrl  teclas

Ⓞ [texto]

Ⓞ procesa *texto* como una línea de comentario, lo que le permite anotar funciones y programas que usted crea.

Ⓞ puede estar al comienzo y en cualquier parte en la línea. Todo a la derecha de Ⓞ, al final de la línea, es el comentario.

Nota para ingresar el ejemplo: En la aplicación de Calculadora en el dispositivo portátil, usted puede ingresar definiciones en líneas múltiples al presionar  en lugar de  al final de cada línea. En el teclado de la computadora, presione y sostenga **Alt** y presione **Ingresar**.

Define $g(n) = \text{Func}$

Ⓞ Declare variables

Local *i, result*

result := 0

For *i*, 1, *n*, 1 Ⓞ Loop *n* times

result := *result* + i^2

EndFor

Return *result*

EndFunc

Done

$g(3)$	14
--------	----

0b, 0h

0 B teclas, 0 H teclas

0b número Binario

0h número Hexadecimal

Denota un número binario o hexadecimal, respectivamente. Para ingresar un número binario o hexadecimal, usted debe ingresar el prefijo 0b ó 0h independientemente del modo de la Base. Sin un prefijo, un número se trata como decimal (base 10).

Los resultados se despliegan de acuerdo con el modo de la Base.

En modo de base decimal:

0b10+0hF+10	27
-------------	----

En modo de base binaria:

0b10+0hF+10	0b11011
-------------	---------

En modo de base hexadecimal:

0b10+0hF+10	0h1B
-------------	------

Elementos vacíos (inválidos)

Cuando analice datos del mundo real, usted quizá no siempre tenga un conjunto de datos completo. El software TI-Nspire™ CAS permite elementos de datos vacíos, o inválidos, de manera que usted podrá proceder con los datos cercanamente completos en lugar de tener que comenzar de nuevo o descartar los casos incompletos.

Usted puede encontrar un ejemplo de datos que incluye elementos vacíos en el capítulo de Listas y Hoja de Cálculo, bajo "Cómo graficar datos de hoja de cálculo".

La función **delVoid()** le permite eliminar elementos vacíos de una lista. La función **isVoid()** le permite probar un elemento vacío. Para obtener detalles, vea **delVoid()**, página 36y **isVoid()**, página 62.

Nota: Para ingresar un elemento vacío manualmente en una expresión matemática, escriba " _ " o la palabra clave **inválido**. La palabra clave **inválido** se convierte automáticamente en un símbolo " _ " cuando se evalúa la expresión. Para escribir " _ " en el dispositivo portátil, presione

 .

Cálculos que incluyen elementos inválidos

La mayoría de los cálculos que incluyen una entrada inválida producirán un resultado inválido. Vea los casos especiales abajo.

$_$	$_$
$\gcd(100, _)$	$_$
$3 + _$	$_$
$\{5, _, 10\} - \{3, 6, 9\}$	$\{2, _, 1\}$

Argumentos de lista que contienen elementos inválidos

Las siguientes funciones y comandos ignoran (se saltan) los elementos inválidos encontrados en argumentos de lista.

count, **countf**, **cumulativeSum**, **freqTable**, **list**, **frequency**, **max**, **mean**, **median**, **product**, **stDevPop**, **stDevSamp**, **sum**, **sumf**, **varPop**, y **varSamp**, así como cálculos de regresión, **OneVar**, **TwoVar** estadísticas de **FiveNumSummary**, intervalos de confianza y pruebas estadísticas

$\text{sum}\{\{2, _, 3, 5, 6, 6\}\}$	16.6
$\text{median}\{\{1, 2, _, _, 3\}\}$	2
$\text{cumulativeSum}\{\{1, 2, _, 4, 5\}\}$	$\{1, 3, _, 7, 12\}$
$\text{cumulativeSum}\left(\begin{array}{cc} 1 & 2 \\ 3 & _ \\ 5 & 6 \end{array}\right)$	$\begin{array}{cc} 1 & 2 \\ 4 & _ \\ 9 & 8 \end{array}$

SortA y **SortD** mueven todos los elementos vacíos dentro del primer argumento a la parte inferior.

$\{5, 4, 3, _, 1\} \rightarrow \text{list1}$	$\{5, 4, 3, _, 1\}$
$\{5, 4, 3, 2, 1\} \rightarrow \text{list2}$	$\{5, 4, 3, 2, 1\}$
$\text{SortA list1, list2}$	Done
list1	$\{1, 3, 4, 5, _ \}$
list2	$\{1, 3, 4, 5, 2\}$
$\{1, 2, 3, _, 5\} \rightarrow \text{list1}$	$\{1, 2, 3, _, 5\}$
$\{1, 2, 3, 4, 5\} \rightarrow \text{list2}$	$\{1, 2, 3, 4, 5\}$
$\text{SortD list1, list2}$	Done
list1	$\{5, 3, 2, 1, _ \}$
list2	$\{5, 3, 2, 1, 4\}$

Argumentos de lista que contienen elementos inválidos(continued)

En las regresiones, un vacío en una lista X o Y introduce un vacío para el elemento correspondiente del residual.

$I1:=\{1,2,3,4,5\}; I2:=\{2,3,5,6,6\}$	$\{2,3,5,6,6\}$
LinRegMx I1,I2	Done
stat.Resid	$\{0.434286, -, 0.862857, -0.011429, 0.44\}$
stat.XReg	$\{1, -, 3, 4, 5\}$
stat.YReg	$\{2, -, 3, 5, 6, 6\}$
stat.FreqReg	$\{1, -, 1, 1, 1\}$

Una categoría omitida en las regresiones introduce un vacío para el elemento correspondiente del residual.

$I1:=\{1,3,4,5\}; I2:=\{2,3,5,6,6\}$	$\{2,3,5,6,6\}$
cat:="M","M","F","F": incl:="F"	$\{ "F" \}$
LinRegMx I1,I2,1,cat,incl	Done
stat.Resid	$\{ -, -, 0, 0, \}$
stat.XReg	$\{ -, -, 4, 5, \}$
stat.YReg	$\{ -, -, 5, 6, 6 \}$
stat.FreqReg	$\{ -, -, 1, 1, \}$

Una frecuencia de 0 en las regresiones introduce un vacío para el elemento correspondiente del residual.

$I1:=\{1,3,4,5\}; I2:=\{2,3,5,6,6\}$	$\{2,3,5,6,6\}$
LinRegMx I1,I2,{1,0,1,1}	Done
stat.Resid	$\{0.069231, -, -0.276923, 0.207692\}$
stat.XReg	$\{1, -, 4, 5, \}$
stat.YReg	$\{2, -, 5, 6, 6\}$
stat.FreqReg	$\{1, -, 1, 1, \}$

Accesos directos para ingresar expresiones matemáticas

Los accesos directos le permiten ingresar elementos de expresiones matemáticas al escribir en lugar de usar el Catálogo o la Paleta de Símbolos. Por ejemplo, para ingresar la expresión $\sqrt{6}$, usted puede escribir `sqrt(6)` en la línea de ingreso. Cuando usted presiona **enter**, la expresión `sqrt(6)` se cambia a $\sqrt{6}$. Algunos accesos directos son útiles tanto desde el dispositivo portátil como desde el teclado de la computadora. Otros son útiles principalmente desde el teclado de la computadora.

Desde el dispositivo portátil o el teclado de la computadora

Para ingresar esto:	Escriba este acceso directo:
π	pi
θ	theta
∞	infinity
\leq	<=
\geq	>=
\neq	/=
\Rightarrow (implicación lógica)	=>
\Leftrightarrow (implicación doble lógica, XNOR)	<=>
\rightarrow (almacenar operador)	=:
$ $ (valor absoluto)	abs(...)
$\sqrt{\quad}$	sqrt(...)
$d()$	derivative(...)
$\int()$	integral(...)
$\Sigma()$ (Plantilla de sumas)	sumSeq(...)
$\Pi()$ (Plantilla de productos)	prodSeq(...)
$\sin^{-1}()$, $\cos^{-1}()$, ...	arcsin(...), arccos(...), ...
Δ Lista()	deltaList(...)
Δ TmpCnv()	deltaTmpCnv(...)

Desde el teclado de la computadora

Para ingresar esto:	Escriba este acceso directo:
$c1$, $c2$, ... (constantes)	@c1, @c2, ...

Para ingresar esto:	Escriba este acceso directo:
$n1, n2, \dots$ (constantes de enteros)	@n1, @n2, ...
i (constante imaginaria)	@i
e (base de logaritmo natural e)	@e
E (notación científica)	@E
\top (trasponer)	@t
Γ (radianes)	@r
$^\circ$ (grados)	@d
g (gradianes)	@g
\sphericalangle (ángulo)	@<
\blacktriangleright (conversión)	@>
\blacktriangleright Decimal, \blacktriangleright approxFraction(), y así sucesivamente.	@>Decimal, @>approxFraction(), y así sucesivamente.

Jerarquía de EOS™ (Sistema Operativo de Ecuaciones)

Esta sección describe el Sistema Operativo de Ecuaciones (EOS™) que se usa en la tecnología de aprendizaje de matemáticas y ciencias de TI-Nspire™ CAS. Los números, las variables y las funciones se ingresan en una secuencia directa sencilla. El software EOS™ evalúa las expresiones y ecuaciones mediante la agrupación entre paréntesis, y de acuerdo con las prioridades descritas a continuación.

Orden de la evaluación

Nivel	Operador
1	Paréntesis (), paréntesis rectangulares [], corchetes { }
2	Indirección (#)
3	Llamadas de función
4	Operadores posteriores: grados-minutos-segundos (°, ', "), factorial (!), porcentaje (%), radián (°), subíndice ([]), trasponer (°)
5	Exponenciación, operador de potencia (^)
6	Negación (¬)
7	Concatenación de cadenas, (&)
8	Multiplicación (•), división (/)
9	Adición (+), sustracción (-)
10	Relaciones de igualdad: igual (=), no igual (≠ o ≠), menor que (<), menor que o igual (≤ o ≤), mayor que (>), mayor que o igual (≥ o ≥)
11	Lógico not
12	Lógico and
13	Lógico or
14	xor, nor, nand
15	Implicación lógica (⇒)
16	Implicación doble lógica, XNOR (⇔)
17	Operador restrictivo (" ")
18	Almacenar (→)

Paréntesis, paréntesis rectangulares y corchetes

Todos los cálculos dentro de un par de paréntesis, paréntesis rectangulares o corchetes se evalúan primero. Por ejemplo, en la expresión $4(1+2)$, el software EOS™ evalúa primero la parte de la expresión dentro del paréntesis, $1+2$, y luego multiplica el resultado, 3, por 4.

El número de paréntesis, paréntesis rectangulares y corchetes iniciales y finales debe ser el mismo dentro de una expresión o ecuación. Si no es así, se despliega un mensaje de error que indica el elemento faltante. Por ejemplo, $(1+2)/(3+4)$ desplegará el mensaje de error "Faltante".

Nota: Debido a que el software TI-Nspire™ CAS le permite definir sus propias funciones, un nombre de variable seguido de una expresión entre paréntesis se considera como una "llamada de función" en lugar de una multiplicación implícita. Por ejemplo $a(b+c)$ es la función a evaluada por $b+c$. Para multiplicar la expresión $b+c$ por la variable a , use la multiplicación explícita: $a*(b+c)$.

Indirección

El operador de indirección (#) convierte una cadena en un nombre de variable o función. Por ejemplo, $\#("x"&"y"&"z")$ crea un nombre de variable xyz . La indirección también permite la creación y modificación de variables desde dentro de un programa. Por ejemplo, si $10 \rightarrow r$ y $"r" \rightarrow s1$, entonces $\#s1=10$.

Operadores posteriores

Los operadores posteriores son operadores que vienen directamente después de un argumento, como $5!$, 25% ó $60^\circ 15' 45''$. Los argumentos seguidos de un operador posterior se evalúan en el cuarto nivel de prioridad. Por ejemplo, en la expresión $4^3!$, $3!$ se evalúa primero. El resultado, 6, entonces se convierte en el exponente de 4 para producir 4096.

Exponenciación

La exponenciación (^) y la exponenciación elemento por elemento (.^) se evalúan de derecha a izquierda. Por ejemplo, la expresión 2^3^2 se evalúa igual que $2^{(3^2)}$ para producir 512. Esto es diferente de $(2^3)^2$, que es 64.

Negación

Para ingresar un número negativo, presione $\boxed{-}$ seguido del número. Las operaciones posteriores y la exponenciación se realizan antes de la negación. Por ejemplo, el resultado de $-x^2$ es un número negativo, y $-9^2 = -81$. Use paréntesis para cuadrar un número negativo como $(-9)^2$ para producir 81.

Restricción ("|")

El argumento que sigue el operador restrictivo ("|") proporciona una serie de restricciones que afectan la evaluación del argumento que precede al operador.

Códigos y mensajes de error

Cuando ocurre un error, su código se asigna a la variable *códigoErr*. Los programas y funciones definidos por el usuario pueden examinar *códigoErr* para determinar la causa de un error. Para ver un ejemplo del uso de *códigoErr*, vea el Ejemplo 2 bajo el comando **Try**, página 132.

Nota: Algunas condiciones de error aplican sólo a los productos TI-Nspire™ CAS, y algunos aplican sólo a los productos TI-Nspire™.

Código de error	Descripción
10	Una función no produjo un valor
20	Una prueba no resolvió para VERDADERO o FALSO. Por lo general, las variables indefinidas no se pueden comparar. Por ejemplo, la prueba Si $a < b$ causará este error si a o b es indefinido cuando se ejecuta la sentencia Si.
30	El argumento no puede ser un nombre de carpeta.
40	Error de argumento
50	Incongruencia de argumento Dos o más argumentos deben ser del mismo tipo.
60	El argumento debe ser una expresión Booleana o un entero
70	El argumento debe ser un número decimal
90	El argumento debe ser una lista
100	El argumento debe ser una matriz
130	El argumento debe ser una cadena
140	El argumento debe ser un nombre de variable. Asegúrese de que el nombre: <ul style="list-style-type: none">• no comience con un dígito• no contenga espacios o caracteres especiales• no use guión bajo o punto en una manera inválida• no exceda las limitaciones de longitud Vea la sección de la Calculadora en la documentación para obtener más detalles.
160	El argumento debe ser una expresión
165	Las baterías están demasiado bajas para enviar o recibir Instale baterías nuevas antes de enviar o recibir.
170	Límite El límite inferior debe ser menor que el límite superior para definir el intervalo de búsqueda.
180	Salto La tecla  o  se presionó durante un cálculo largo o durante la ejecución del programa.
190	Definición circular Este mensaje se despliega para evitar que la memoria se agote durante el reemplazo infinito de valores de variable durante la simplificación. Por ejemplo, $a+1 \rightarrow a$, donde a es una variable indefinida, causará este error.
200	Expresión de restricción inválida Por ejemplo, $\text{solve}(3x^2-4=0,x) x < 0 \text{ or } x > 5$ produciría este error porque la restricción está separada por "or" en lugar de "and".
210	Tipo de datos inválido Un argumento es del tipo de datos incorrecto.
220	Límite dependiente

Código de error	Descripción
230	Dimensión Un índice de lista o matriz no es válido. Por ejemplo, si la lista {1,2,3,4} está almacenada en L1, entonces L1[5] es un error de dimensión porque L1 sólo contiene cuatro elementos.
235	Error de Dimensión No hay elementos suficientes en las listas.
240	Incongruencia de dimensión Dos o más argumentos deben ser de la misma dimensión. Por ejemplo, {1,2}+{1,2,3} es una incongruencia de dimensión porque las matrices contienen un número de elementos distinto.
250	Dividir por cero
260	Error de dominio Un argumento debe estar en un dominio especificado. Por ejemplo, rand(0) no es válido.
270	Duplicar nombre de variable
280	Else y ElseIf son inválidos afuera del bloque If...EndIf
290	A TerminarIntentar le falta la sentencia Else congruente
295	Iteración excesiva
300	Lista o matriz de 2 ó 3 elementos esperada
310	El primer argumento de nSolve debe ser una ecuación en una variable sencilla. No puede contener una variable no valorada que no sea la variable de interés.
320	El primer argumento de solve o cSolve debe ser una ecuación o desigualdad Por ejemplo, solve(3x ² -4,x) es vacío porque el primer argumento no es una ecuación.
345	Unidades inconsistentes
350	Índice fuera de rango
360	La cadena de indirección no es un nombre de variable válido
380	Ans indefinido O bien el cálculo anterior no creó Ans o no se ingresó ningún cálculo anterior
390	Asignación inválida
400	Valor de asignación inválido
410	Comando inválido
430	Inválido para las configuraciones del modo actual
435	Cálculo inválido
440	multiplicación implícita inválida Por ejemplo, x(x+1) es inválido; mientras que, x*(x+1) es la sintaxis correcta. Esto es para evitar una confusión entre la multiplicación implícita y la definición de la función.
450	Inválido en una función o expresión actual Sólo ciertos comandos son válidos en una función definida por el usuario
490	Inválido en el bloque Try..EndTry
510	Lista o matriz inválida
550	Inválido afuera de la función o el programa Un número de comandos no es válido afuera de una función o un programa. Por ejemplo, Local no se puede usar, a menos que sea una función o un programa.
560	Inválido afuera de los bloques Loop..EndLoop, For...EndFor, o While...EndWhile. Por ejemplo, el comando Exit es válido sólo dentro de estos bloques de bucle.

Código de error	Descripción
565	Inválido afuera del programa
570	nombre de ruta inválido Por ejemplo, \var es inválida.
575	Complejo polar inválido
580	Referencia de programa inválida Los programas no se pueden referenciar dentro de funciones o expresiones como $1+p(x)$ donde p es un programa.
600	Tabla inválida
605	uso de unidades inválido
610	Nombre de variable inválido en una sentencia Local
620	Nombre de variable o función inválido
630	Referencia de variable inválida
640	Sintaxis de vector inválida
650	Transmisión de enlace Una transmisión entre dos unidades no se completó. Verifique que el cable de conexión esté bien conectado en ambos extremos.
665	Matriz no diagonalizable
670	Memoria Baja 1. Borre algunos datos en este documento 2. Guarde y cierre este documento Si 1 y 2 fallan, extraiga y reinserte las baterías
672	Agotamiento de recursos
673	Agotamiento de recursos
680	(Faltante
690) Faltante
700	" Faltantes
710] Faltante
720	} Faltante
730	Sintaxis del bloque inicio o final faltante
740	Entonces faltante en el bloque If..EndIf
750	El nombre no es una función o un programa
765	Ninguna función seleccionada
780	No se encontró ninguna solución
800	Resultado no real Por ejemplo, si el software está en la configuración Real, $\sqrt{-1}$ es inválido. Para permitir resultados complejos, cambie la Configuración del Modo "Real o Complejo" a RECTANGULAR O POLAR.
830	Desbordamiento
850	Programa no encontrado No se pudo encontrar una referencia de programa adentro de otro programa en la ruta provista durante la ejecución.

Código de error	Descripción
855	Las funciones de tipo aleatorio no se permiten en la representación gráfica
860	Recursión demasiado profunda
870	variable de nombre o sistema reservada
900	Error de argumento El modelo mediana-mediana no se pudo aplicar al conjunto de datos.
910	Error de sintaxis
920	Texto no encontrado
930	Muy pocos argumentos Uno o más argumentos faltantes en la función o el comando.
940	Demasiados argumentos La expresión o ecuación contiene un número de argumentos excesivo y no se puede evaluar.
950	Demasiados subíndices
955	Demasiadas variables indefinidas
960	La variable no está definida No hay ningún valor asignado a la variable. Use uno de los siguientes comandos: <ul style="list-style-type: none"> • alm → • := • Define para asignar valores a las variables
965	SO sin licencia
970	Variable en uso, así que las referencias o los cambios no se permiten
980	La variable está protegida
990	Nombre de variable inválido Asegúrese de que el nombre no exceda las limitaciones de longitud
1000	Dominio de variables de ventana
1010	Zoom
1020	Error interno
1030	Violación de memoria protegida
1040	Función no soportada. Esta función requiere del Sistema de Álgebra de Computadora. Pruebe TI-Nspire™ CAS.
1045	Operador no soportado. Este operador requiere del Sistema de Álgebra de Computadora. Pruebe TI-Nspire™ CAS.
1050	Característica no soportada. Este operador requiere del Sistema de Álgebra de Computadora. Pruebe TI-Nspire™ CAS.
1060	El argumento de entrada debe ser numérico. Sólo las entradas que contienen valores numéricos están permitidos.
1070	Argumento de función trigonométrica demasiado grande para una reducción exacta
1080	Uso de Ans no soportado. Esta aplicación no soporta Ans.
1090	La función no está definida. Use uno de los siguientes comandos: <ul style="list-style-type: none"> • Define • := • alm → para definir una función.

Código de error	Descripción
1100	Cálculo no real Por ejemplo, si el software está en la configuración Real, $\sqrt{(-1)}$ es inválido. Para permitir resultados complejos, cambie la Configuración del Modo "Real o Complejo" a RECTANGULAR O POLAR.
1110	Límites inválidos
1120	Ningún cambio de signo
1130	El argumento no puede ser una lista o matriz
1140	Error de argumento El primer argumento debe ser una expresión polinómica en el segundo argumento. Si el segundo argumento se omite, el software intenta seleccionar un predeterminado.
1150	Error de argumento Los primeros dos argumentos deben ser expresiones polinómicas en el tercer argumento. Si el tercer argumento se omite, el software intenta seleccionar un predeterminado.
1160	nombre de ruta de librería inválido Un nombre de ruta debe ser en la forma $xxx\text{עניני}$, donde: <ul style="list-style-type: none"> • La parte xxx puede tener de 1 a 16 caracteres. • La parte עניני puede tener de 1 a 15 caracteres. Vea la sección de Librería en la documentación para obtener más detalles.
1170	Uso de nombre de ruta de librería inválido <ul style="list-style-type: none"> • No se puede asignar un valor a un nombre de ruta al usar Define, $:=$ o $\text{alm} \rightarrow$. • Un nombre de ruta no se puede declarar como una variable Local o usarse como un parámetro en una definición de función o de programa.
1180	Nombre de variable de librería inválido. Asegúrese de que el nombre: <ul style="list-style-type: none"> • No contenga un punto • No comience con un guión bajo • No exceda de 15 caracteres Vea la sección de Librería en la documentación para obtener más detalles.
1190	Documento de librería no encontrado: <ul style="list-style-type: none"> • Verifique que la librería esté en la carpeta MiLib. • Actualice Librerías. Vea la sección de Librería en la documentación para obtener más detalles.
1200	Variable de librería no encontrada: <ul style="list-style-type: none"> • Verifique que la variable de librería existe en el primer problema en la librería. • Asegúrese de que la variable de librería se ha definido como LibPub o LibPriv. • Actualice Librerías. Vea la sección de Librería en la documentación para obtener más detalles.
1210	Nombre de acceso directo de librería inválido. Asegúrese de que el nombre: <ul style="list-style-type: none"> • No contenga un punto • No comience con un guión bajo • No exceda de 16 caracteres • No es un nombre reservado Vea la sección de Librería en la documentación para obtener más detalles.
1220	Error de dominio: Las funciones tangentLine y normalLine sólo soportan funciones valoradas reales.
1230	Error de dominio. Los operadores de conversión trigonométrica no están soportados en los modos de ángulo en Grados o Gradianes.
1250	Error de Argumento Use un sistema de ecuaciones lineales. Ejemplo de un sistema de dos ecuaciones lineales con variables x y y : $3x + 7y = 5$ $2y - 5x = -1$

Código de error	Descripción
1260	Error de Argumento: El primer argumento de nfMin o nfMax debe ser una expresión en una variable sencilla. No puede contener una variable no valorada que no sea la variable de interés.
1270	Error de Argumento El Orden de la derivada debe ser igual a 1 ó 2.
1280	Error de Argumento Use un polinomio en forma expandida en una variable.
1290	Error de Argumento Use un polinomio en una variable.
1300	Error de Argumento Los coeficientes del polinomio se deben evaluar a valores numéricos.
1310	Error de argumento: Una función no se pudo evaluar para uno o más de sus argumentos.
1380	Error de argumento: No se permiten llamadas anidadas en la función del dominio().

Códigos y mensajes de advertencia

Usted puede usar la función **warnCodes()** para almacenar los códigos de las advertencias generadas al evaluar una expresión. Esta tabla enumera cada código de advertencia numérico y su mensaje asociado.

Para obtener un ejemplo de cómo almacenar códigos de advertencia, vea **warnCodes()**, página [138](#).

Código de advertencia	Mensaje
10000	La operación podría introducir soluciones falsas.
10001	Diferenciar una ecuación puede producir una ecuación falsa.
10002	Solución cuestionable
10003	Exactitud cuestionable
10004	La operación podría perder las soluciones.
10005	cResolver podría especificar más ceros.
10006	Resolver puede especificar más ceros.
10007	Es posible que existan más soluciones. Intente especificar límites superiores o inferiores correctos y/o un punto inicial. Ejemplos utilizando la función solución(): <ul style="list-style-type: none"> • solución(Ecuación, Var=Estimar) limitInferior<Var<limiteSuperior • solución(Ecuación, Var) limitInferior<Var<limiteSuperior • solución(Ecuación, Var=Estimar)
10008	El dominio del resultado podría ser más pequeño que el dominio de la entrada.
10009	El dominio del resultado podría ser más GRANDE que el dominio de la entrada.
10012	Cálculo no real
10013	∞^0 ó indef^0 reemplazado por 1

Código de advertencia	Mensaje
10014	indef^0 reemplazado por 1
10015	1^∞ ó 1^{indef} reemplazado por 1
10016	1^{indef} reemplazado por 1
10017	Desbordamiento reemplazado por ∞ o $-\infty$
10018	La operación requiere y entrega un valor de 64 bits.
10019	Agotamiento del recurso, la simplificación podría estar incompleta.
10020	Argumento de función de trigonometría demasiado grande para una reducción exacta.
10021	La entrada contiene un parámetro indefinido. El resultado podría no ser válido para todos los posibles valores de parámetro.
10022	Especificar los límites inferiores y superiores apropiados podrían producir una solución.
10023	El escalador se ha multiplicado por la matriz de identidad.
10024	Resultado obtenido usando aritmética aproximada.
10025	La equivalencia no se puede verificar en el modo EXACTO.
10026	La restricción se podría ignorar. Especifique la restricción en la forma " $\text{"\ "} \text{" Constante de SímboloPruebaMat de Variable'}$ o un conjunto de estas formas, por ejemplo ' $x < 3$ y $x > -12$ '

Soporte y Servicio

Soporte y Servicio de Texas Instruments

Para los EE.UU. y Canadá:

Para obtener información general

Página Principal: education.ti.com

Base de conocimientos y preguntas por correo electrónico: education.ti.com/support

Teléfono: (800) TI-CARES / (800) 842-2737
Para los EE.UU., Canadá, México, Puerto Rico y las Islas Vírgenes únicamente

Información internacional: education.ti.com/international

Para obtener soporte técnico

Base de Conocimientos y soporte por correo electrónico: education.ti.com/support

Teléfono (no gratuito): (972) 917-8324

Para servicio (hardware) de producto

Cientes en los EE.UU., Canadá, México, Puerto Rico y las Islas Vírgenes: Siempre contacte a Soporte Técnico de Texas Instruments antes de devolver el producto para servicio.

Para todos los demás países:

Para obtener información general

Para obtener más información sobre los productos y servicios de TI, contacte a TI por correo electrónico o visite la dirección en Internet de TI.

Preguntas por correo electrónico: ti-cares@ti.com

Página Principal: education.ti.com

Información sobre servicio y garantía

Para obtener información sobre la duración y los términos de la garantía, o bien sobre el servicio para el producto, consulte el certificado de garantía incluido con este producto o contacte a su vendedor o distribuidor local de Texas Instruments.

Index

Symbols

\wedge , potencia 149
 \wedge^{-1} , recíproco 163
_, designación de unidad 162
:=, asignar 165
!, factorial 154
.^, punto potencia 151
.*, punto multiplicación 150
.+, punto agregar 150
.-, punto sustracción 150
.:, punto división 151
' , notación en minuto 161
' , primo 161
" , notación en segundo 161
 \leq , menor que o igual 153
©, comentario 165
 Δ lista(), diferencia de lista 69
 $^\circ$, grados/minutos/segundos 161
 $^\circ$, notación en grados 160
►, convertir unidades 162
>, integral 155
<, raíz cuadrada 156
, no igual 153
-, sustraer 147
 \div , dividir 148
 Π , producto 157
 $\Sigma()$, suma 157
 \Leftrightarrow , implicación lógica doble 154
 \Rightarrow , implicación lógica 154, 168
*, multiplicar 148
&, adjuntar 154
 \rightarrow , almacenar 164
#, indirección 159
#, operador de indirección 171
%, porcentaje 152
+, agregar 147
<, menor que 153
=, igual 152
>, mayor que 153
 \geq , mayor que o igual 154
|, operador restrictivo 163

Numerics

0b, indicador binario 165

0h, indicador hexadecimal 165

10^(), potencia de diez 162

►Fracciónaprox() 11

A

abs(), valor absoluto 7

accesoDirectoLib(), crear accesos directos para objetos de librería 64

adjuntar, & 154

agregar, + 147

agrFila(), agregado de fila de matriz 107

agrFilaM(), multiplicación y suma de fila de matriz 79

aleat(), número aleatorio 98

aleatoria

matriz, matAleat() 99

norma, normAleat() 99

aleatorio

polinomio, poliAleat() 99

semilla de número, SemillaAleat 100

and, Boolean operator 7

ángulo, ángulo() 8

ángulo(), ángulo 8

ANOVA, análisis de varianza unidireccional 8

ANOVA2vías, análisis de varianza bidireccional 9

Ans, última respuesta 11

aprox(), aproximado 11, 12

aproximado, aprox() 11, 12

arccos() 11

arccosh() 12

arccot() 12

arccoth() 12

arccsc() 12

arccsch() 12

arcoseno, $\cos^{-1}()$ 24

arcoseno, $\sin^{-1}()$ 116

arcotangente, $\tan^{-1}()$ 127

arcsec() 12

arcsech() 12

arcsin() 12
arcsinh() 12
arctan() 12
arctanh() 12
argumentos del VTD 135
argumentos en funciones del VTD 135
aumentar(), aumentar/concatenar 12
aumentar/concatenar, aumentar() 12
aumentCol 19

B

BA, descomposición baja-alta de matriz 75
►Base10, se despliega como entero decimal 14
►Base16, se despliega como hexadecimal 15
►Base2, se despliega como binario 14
binAleat, número aleatorio 99
binario
 indicador, 0b 165
 se despliega, ►Base2 14
binomCdf() 15
binomPdf() 15
bloquear variables y grupos de variables 71
Bloquear, bloquear variable o grupo de variables 71
Boolean operators
 and 7
borrar
 elementos inválidos de la lista 36
borrInval(), eliminar los elementos inválidos 36
BorrVar, borrar variable 36
Bucle, bucle 75
bucle, Bucle 75
BxRegLin, regresión lineal 65

C

χ^2 2vias 17
 χ^2 Cdf() 18
 χ^2 GOF 18
 χ^2 Pdf() 18

cadena
 dimensión, dim() 38
 longitud 38
cadena de caracteres, car() 17
cadena de formato, formato() 51
cadena med, med() 78
cadena(), expresión para cadena 124
cadenas
 adjuntar, & 154
 cadena de caracteres, car() 17
 cadena med, med() 78
 cadena para expresión, expr() 46, 72
 cambiar, cambiar() 113
 código de caracter, ord() 88
 cómo formatear 51
 cómo usar para crear nombres de variable 171
 dentro, EnCadena 60
 derecha, derecha() 104
 expresión para cadena, cadena() 124
 formato, formato() 51
 indirección, # 159
 izquierda, izquierda() 63
 rotar, rotar() 105, 106
cambiar, cambiar() 113
cambiar(), cambiar 113
cambioFila(), cambio de fila de matriz 107
car(), cadena de caracteres 17
caracteres
 cadena, car() 17
 código numérico, ord() 88
Cdf() 48
Cdfgeom() 54
CdfNormal() 85
CdfT(), probabilidad de distribución de student-*t* 128
ceros, ceros() 140
ceros(), ceros 140
cerosC(), ceros complejos 32
Ciclo, ciclo 32
ciclo, Ciclo 32
►Cilind, se despliega como vector cilíndrico 32
clear

error, ClrErr 19
 ClrErr, clear error 19
 ▶cnvTmp() 131
 cnvTmp() 131
 códigos y mensajes de advertencia 177
 códigos y mensajes de error 172
 coefPoli() 91
 comando de Texto 129
 comando Detener 124
 combinaciones, nCr() 82
 comentario, © 165
 cómo almacenar
 símbolo, → 164, 165
 cómo borrar
 variable, BorrVar 36
 cómo definir
 función o programa privado 35
 función o programa público 36
 cómo desbloquear variables y grupos de variables 137
 cómo ordenar
 ascendente, OrdenarA 120
 descendente, OrdenarD 120
 cómo programar
 definir programa, Prgm 94
 desplegar datos, Desp 39
 pasar error, PasarErr 89
 complejo
 ceros, cerosC() 32
 conjugado, conj() 21
 factor, FactorC() 16
 solucionar, solucionC() 29
 completeSquare(), complete square 21
 compuestoDeVariables() 89
 con, | 163
 configuraciones de modo,
 obtModo() 55
 configuraciones, obtener actual 55
 conj(), complejo conjugado 21
 constante
 en solucion() 118
 constantes
 accesos directos para 168
 en ceros() 141
 en cerosC() 33
 en resolverEd() 37
 en solucion() 119
 en solucionC() 30
 construir matriz, construMat() 21
 construMat(), construir matriz 21
 contar días entre fechas, def() 34
 conteo condicional de elementos en una lista, conteo() 27
 conteo de elementos en una lista, conteo() 26
 conteo(), conteo de elementos en una lista 26
 conteoSi(), conteo condicional de elementos en una lista 27
 conTmpDelta() 36
 convertir
 ▶Grad 57
 ▶Rad 98
 unidades 162
 coordenada x rectangular, P▶Rx() 89
 coordenada y rectangular, P▶Ry() 89
 copiar variable o función, CopiarVar 22
 ▶cos, se despliega en términos de coseno 22
 cos(), coseno 23
 cos⁻¹, arcoseno 24
 coseno
 despliega la expresión en términos de 22
 coseno, cos() 23
 cosh(), coseno hiperbólico 24
 cosh⁻¹(), arcoseno hiperbólico 25
 cot(), cotangente 25
 cot⁻¹(), arcotangente 25
 cotangente, cot() 25
 coth(), cotangente hiperbólica 26
 coth⁻¹(), arcotangente hiperbólica 26
 csc(), cosecante 28
 csc⁻¹(), cosecante inversa 28
 csch(), cosecante hiperbólica 28
 csch⁻¹(), cosecante hiperbólica inversa 28
 cuando, cuando() 138
 cuando(), cuando 138

D

`d()`, primera derivada 155
►DD, se despliega como ángulo decimal 34
►Decimal, despliega el resultado como decimal 34
decimal
 despliegue de ángulo, ►DD 34
 se despliega como entero, ►Base10 14
`def()`, días entre fechas 34
Definir 35
Definir LibPriv 35
Definir LibPub 36
Definir, definir 35
definir, Definir 35
`denomCom()`, denominador común 20
denominador 20
denominador común, `denomCom()` 20
densidad de probabilidad de student-*t*, `PdfT()` 131
densidad de probabilidad, `PdfNorm()` 85
dentro de cadena, `enCadena()` 60
derecha, `derecha()` 104
`derecha()`, derecha 104
derivada implícita, `difImp()` 60
derivada o enésima derivada
 plantilla para 5
`derivada()` 36
`derivadaN()`, derivada numérica 82
derivadas
 derivada numérica, `derivadaN()` 82
 derivada numérica, `derivN()` 83
 primera derivada, `d()` 155
desbloquear, desbloquear variable o grupo de variables 137
Desp, desplegar datos 39
desplegar datos, Desp 39
despliegue de grado/minuto/segundo, ►GMS 39
despliegue de vector esférico, ►Esfera 121

despliegue de vector rectangular, ►Rect 100
`desvEstMuest()`, desviación estándar muestra 123
`desvEstPob()`, desviación estándar de población 123
desviación estándar, `desvEst()` 123, 137
`det()`, matriz determinante 38
`diag()`, diagonal de matriz 38
días entre fechas, `def()` 34
`difCentral()` 16
`difImp()`, derivada implícita 60
`dim()`, dimensión 38
`dimCol()`, dimensión de columna de matriz 19
dimensión, `dim()` 38
`dimFila()`, dimensión de fila de matriz 107
distribución normal acumulativa inversa (`normInv()`) 61
distribution functions
 `poissCdf()` 90
`divEnt()`, dividir entero 60
dividir entero, `divEnt()` 60
dividir, ÷ 148
`dominio()`, función del dominio 39
DosVar, resultados de dos variables 135

E

e exponente
 plantilla para 2
e para una potencia, `e^()` 41, 44
`e^()`, e para una potencia 41
e, despliega la expresión de 44
E, exponente 159
ecuaciones simultáneas, `simult()` 114
`ef()`, convertir nominal a tasa efectiva 41
elementos inválidos, eliminar 36
elementos vacíos 166
elementos vacíos (inválidos) 166
eliminar
 elementos inválidos de la lista 36
`enCadena()`, dentro de cadena 60
`ent()`, entero 60

entAleat(), entero aleatorio 99
entero, ent() 60
Entrada, entrada 60
entrada, Entrada 60
EOS (Sistema Operativo de Ecuaciones) 170
errores y solución de problemas
 pasar error, PasarErr 89
errors and troubleshooting
 clear error, ClrErr 19
►Esfera, se despliega como vector esférico 121
esInválido(), prueba para inválido 62
esPrimo(), prueba de primo 62
estad.resultados 122
estad.valores 123
estadísticas
 combinaciones, nCr() 82
 desviación estándar, desvEst() 123, 137
 estadísticas de una variable, UnaVar 87
 factorial, ! 154
 media, media() 76
 mediana, mediana() 77
 norma aleatoria, normAleat() 99
 permutaciones, prN() 86
 resultados de dos variables, DosVar 135
 semilla de número aleatorio, SemillaAleat 100
 varianza, varianza() 138
estadísticas de una variable, UnaVar 87
Etiq, etiqueta 63
etiqueta, Etiq 63
euler(), Euler function 43
evalPoli(), evaluar polinomio 91
evaluación, orden de 170
evaluar polinomio, evalPoli() 91
exacto, exacto() 43
exacto(), exacto 43
exclusión con el operador "|" 163
►exp, despliega e 44
exp(), e para una potencia 44
exp►lista(), expresión para lista 45
expandir, expandir() 45

expandir(), expandir 45
expansión trigonométrica, expanT() 129
expanT(), expansión trigonométrica 129
exponente, E 159
exponentes
 plantilla para 1
expr(), cadena para expresión 46, 72
expresiones
 cadena para expresión, expr() 46, 72
 expresión para lista, exp►lista() 45

F

factor, factor() 47
factor(), factor 47
FactorC(), factor complejo 16
factorial, ! 154
factorización de QR, QR 95
ferf(), forma escalonada reducida por filas 107
filaM(), operación de fila de matriz 79
flnv() 61
fMax(), función máxima 49
fMin(), función mínima 50
fnMáx(), función numérica máxima 83
fnMín(), función numérica mínima 83
forma escalonada por filas, ref() 101
forma escalonada reducida por filas, ferf() 107
formato(), cadena de formato 51
fracción propia, fracProp 95
fracciones
 fracProp 95
 plantilla para 1
fracciones mezcladas, utilizando fracProp() con 95
fracProp, fracción propia 95
frecuencia() 52
Func, función 53
Func, función de programa 53

función de compuesto de variables
(2 piezas)
plantilla para 2

función para determinar dominio,
dominio() 39

funciones

definidas por el usuario 35
función de programa, Func 53
máxima, fMáx() 49
mínima, fMín() 50
parte, parteF() 51

funciones de distribución

binomCdf() 15

binomPdf() 15

χ^2 2vías() 17

χ^2 Cdf() 18

χ^2 GOF() 18

χ^2 Pdf() 18

CdfNormal() 85

CdfT() 128

Inv χ^2 () 61

normlnv() 61

PdfNorm() 85

Pdfpoiss() 90

PdfT() 131

tlnv() 61

funciones definidas por el usuario 35

funciones financieras, vtdl() 134

funciones financieras, vtdN() 134

funciones financieras, vtdPgo() 134

funciones financieras, vtdVF() 134

funciones financieras, vtdVP() 134

funciones y programas definidos por
el usuario 35, 36

funciones y variables

cómo copiar 22

G

g, gradianes 160

getType(), get type of variable 56

►GMS, se despliega como grado/
minuto/segundo 39

►, convertir a ángulo en gradianes 57

gradoPoli() 91

grupos, cómo bloquear y
desbloquear 71, 137

grupos, cómo probar el estado de
bloqueo 55

guion bajo, _ 162

H

hexadecimal

indicador, 0h 165

se despliega, ►Base16 15

hiperbólico

arcoseno, cosh⁻¹() 25

arcoseno, sinh⁻¹() 116

arcotangente, tanh⁻¹() 128

coseno, cosh() 24

seno, sinh() 116

tangente, tanh() 127

I

identidad(), matriz de identidad 58

idioma

obtener información del idioma
54

igual, = 152

imag(), parte imaginaria 59

implicación lógica doble, ⇔ 154

implicación lógica, ⇒ 154, 168

ln(), logaritmo natural 70

indirección, # 159

integral definida

plantilla para 5

integral indefinida

plantilla para 5

integral, > 155

Intentar, comando de manejo de
error 132

interpolate(), interpolate 61

IntervalosRegLin, regresión lineal 67

IntervalosRegMult() 80

intervaloT_2Muest, intervalo de

confianza t-de dos muestras 130

intervaloT, intervalo de confianza t
130

intervaloZ_1Prop, intervalo de
confianza Z de una proporción
142

intervaloZ_2Muest, intervalo de
confianza Z de dos muestras 143

intervaloZ_2Prop, intervalo de
confianza Z de dos proporciones
143

intervaloZ, intervalo de confianza Z
142

intN(), integral numérica 83

inválido, prueba para 62

Inv $\chi^2()$ 61

inverso, \wedge^{-1} 163

ir a, IrA 57

IrA, ir a 57

izquierda, izquierda() 63

izquierda(), izquierda 63

L

LibPriv 35

LibPub 36

librería

crear accesos directos para
objetos 64

límite

lím() 64

límite() 64

plantilla para 6

límite() o lím(), límite 64

LimpiarAZ 19

línea normal, líneaNormal() 85

línea tangente, líneaTangente() 127

líneaNormal() 85

líneaTangente() 127

lista para matriz, lista▶mat() 69

lista, conteo condicional de
elementos en 27

lista, conteo de elementos en 26

lista▶mat(), lista para matriz 69

listaDelta() 36

listas

aumentar/concatenar,

aumentar() 12

cadena med, med() 78

diferencia, Δ lista() 69

diferencias en una lista, Δ lista()
69

elementos vacíos en 166

expresión para lista, exp▶lista()
45

lista para matriz, lista▶mat() 69

lista, nuevaLista() 82

matriz para lista, mat▶lista() 76

mínimo, mín() 78

ordenar ascendente, OrdenarA
120

ordenar descendente, OrdenarD
120

producto cruzado, pCruz() 27

producto punto, pPunto() 40

producto, producto() 94

suma acumulativa,
sumaAcumulativa() 31

sumatoria, suma() 125

Llenar, llenar matriz 48

local, Local 71

Local, variable local 71

logaritmo natural, En() 70

logaritmos 70

Logística

plantilla para 2

Logística, regresión logística 73

LogísticaD, regresión logística 74

Lonarc(), longitud de arco 12

longitud de arco, Lonarc() 12

longitud de cadena 38

M

más si, MásSi 42

más, Más 58

MásSi, más si 42

mat▶lista(), matriz para lista 76

matAleat(), matriz aleatoria 99

matCorr(), matriz de correlación 22

matrices

agregado de fila, agrFila() 107

aleatoria, matAleat() 99

aumentar/concatenar,
aumentar() 12

cambio de fila, cambioFila() 107

cómo llenar, Llenar 48

descomposición baja-alta, BA 75

determinante, det() 38

diagonal, diag() 38

dimensión de columna, dimCol()
19

dimensión de fila, dimFila() 107

dimensión, dim() 38

factorización de QR, QR 95
 forma escalón reducida por filas, `ferf()` 107
 forma escalonada por filas, `ref()` 101
 identidad, `identidad()` 58
 lista para matriz, `listaMat()` 69
 matriz para lista, `matLista()` 76
 mínimo, `mín()` 78
 multiplicación y suma de fila, `agrFilaM()` 79
 norma de columna, `normaCol()` 19
 norma de fila, `normaFila()` 107
 nueva, `nuevaMat()` 83
 operación de fila, `filaM()` 79
 producto, `producto()` 94
 punto agregar, `.+` 150
 punto división, `./` 151
 punto multiplicación, `.*` 150
 punto potencia, `.^` 151
 punto sustracción, `.-` 150
 submatriz, `subMat()` 124, 125
 suma acumulativa, `sumaAcumulativa()` 31
 sumatoria, `suma()` 125
 trasponer, `T` 126
 valorPropio, `vlProp()` 42
 vectorPropio, `vcProp()` 41
 matriz (1 × 2)
 plantilla para 4
 matriz (2 × 1)
 plantilla para 4
 matriz (2 × 2)
 plantilla para 3
 matriz (m × n)
 plantilla para 4
 matriz de correlación, `matCorr()` 22
 matriz de identidad, `identidad()` 58
 matriz para lista, `matLista()` 76
 máximo común divisor, `mcd()` 53
 mayor que o igual, \geq 154
 mayor que, $>$ 153
`mcd()`, máximo común divisor 53
`mcdPoli()` 92
 mcm, mínimo común múltiplo 63
`med()`, cadena med 78
 media, `media()` 76
`media()`, media 76
 mediana, `mediana()` 77
`mediana()`, mediana 77
 MedMed, regresión de línea media-media 77
 menor que o igual, \leq 153
 menor que, $<$ 153
 Mientras, `mientras` 139
`mientras`, `Mientras` 139
`mín()`, mínimo 78
 mínimo común múltiplo, `mcm` 63
 mínimo, `mín()` 78
`mod()`, módulo 79
 modes
 setting, `setMode()` 112
 módulo, `mod()` 79
`muestAleat()` 100
 muestra aleatoria 100
 multiplicar, `*` 148
`MxRegLin`, regresión lineal 66

N

`nand`, operador booleano 81
`nCr()`, combinaciones 82
 negación, cómo ingresar números negativos 171
 no igual, 153
`nom()`, convertir efectiva a tasa nominal 84
`nor`, operador booleano 84
 norma Frobenius, `norma()` 84
`norma()`, norma Frobenius 84
`normaCol()`, norma de columna de matriz 19
`normaFila()`, norma de fila de matriz 107
`normAleat()`, norma aleatoria 99
`normInv()`, distribución normal acumulativa inversa 61
`not`, operador booleano 85
 notación en gradián, ^g 160
 notación en grado/minuto/segundo 161
 notación en grados, ^o 160
 notación en minuto, ' 161
 notación en segundo, " 161
 nueva

lista, nuevaLista() 82
matriz, nuevaMat() 83
nuevaLista(), nueva lista 82
nuevaMat(), nueva matriz 83
numérica
derivada, derivadaN() 82
derivada, derivN() 83
integral, intN() 83
solución, solucionN() 86

O

objetos

crear accesos directos para
librería 64

obtDenom(), obtener/producir
denominador 54

obtener/producir
denominador, obtDenom() 54
información de variables,
obtInfoVar() 54, 56
número, obtNúm() 56

obtInfoBloq(), prueba el estado de
bloqueo de la variable o del
grupo de variables 55

obtInfoIdioma(), obtener/producir
información del idioma 54

obtInfoVar(), obtener/producir
información de variables 56

obtModo(), obtener configuraciones
de modo 55

obtNúm(), obtener/producir
número 56

operador de indirección (#) 171

operador restrictivo "|" 163

operador restrictivo, orden de la
evaluación 170

operadores

orden de evaluación 170

Operadores booleanos

nand 81

nor 84

not 85

or 88

⇔ 154

xor 139

⇒ 154, 168

or (booleano), or 88

or, operador booleano 88

ord(), código de carácter numérico
88

OrdenarA, ordenar ascendente 120

OrdenarD, ordenar descendente 120

P

P►Rx(), coordenada x rectangular 89
P►Ry(), coordenada y rectangular 89
Para 50

Para, para 50

para, Para 50

parte de entero, pEnt() 62

parte imaginaria, imag() 59

parteF(), parte de función 51

pasar error, PasarErr 89

PasarErr, pasar error 89

pCruz(), producto cruzado 27

Pdf() 51

Pdfgeom() 54

PdfNorm() 85

Pdfpoiss() 90

PdfT(), densidad de probabilidad de
student-*t* 131

pEnt(), parte de entero 62

permutaciones, prN() 86

Pgrm, definir programa 94

piecewise() 89

piso, piso() 49

piso(), piso 49

plantillas

derivada o enésima derivada 5

e exponente 2

exponente 1

fracción 1

función de compuesto de
variables (2 piezas) 2

función de compuesto de
variables (N piezas) 2

integral definida 5

integral indefinida 5

límite 6

Logística 2

matriz (1 × 2) 4

matriz (2 × 1) 4

matriz (2 × 2) 3

matriz (m × n) 4

primera derivada 5
 producto (Π) 4
 raíz cuadrada 1
 raíz enésima 1
 segunda derivada 5
 sistema de ecuaciones (2
 ecuaciones) 3
 sistema de ecuaciones (N
 ecuaciones) 3
 suma (Σ) 4
 valor absoluto 3
 poisnCdf() 90
 ▶Polar, se despliega como vector
 polar 90
 polar
 coordenada, R▶Pθ() 98
 coordenada, R▶Pr() 98
 despliegue de vector, ▶Polar 90
 poliAleat(), polinomio aleatorio 99
 poliCar() 17
 polinomio de Taylor, taylor() 128
 polinomios
 aleatorio, poliAleat() 99
 evaluar, evalPoli() 91
 porcentaje, % 152
 potencia de diez, 10^() 162
 potencia, ^ 149
 pPunto(), producto punto 40
 primera derivada
 plantilla para 5
 primo, ' 161
 prN(), permutaciones 86
 probabilidad de distribución de
 student-*t*, CdfT() 128
 probabilidad de distribución normal,
 CdfNormal() 85
 prodSec() 94
 Producir, producir 104
 producir, Producir 104
 producto (Π)
 plantilla para 4
 producto cruzado, pCruz() 27
 producto, Π() 157
 producto, producto() 94
 producto(), producto 94
 programas
 cómo definir una librería privada
 35
 cómo definir una librería pública
 36
 programas y cómo programar
 desplegar pantalla I/O, Desp 39
 intentar, Intentar 132
 terminar intentar,
 TerminarIntentar 132
 terminar programa,
 TerminarPrgm 94
 programs and programming
 clear error, ClrErr 19
 prueba de número primo, esPrimo()
 62
 Prueba F de 2 muestras 52
 prueba para inválido, esInválido()
 62
 Prueba t de regresión lineal múltiple
 80
prueba T, pruebaT 133
 Prueba_2M, prueba F de 2 muestras
 52
 PruebasRegMult() 80
 pruebaT_2Muest, prueba *T* de dos
 muestras 133
 pruebaT, prueba *T* 133
 PruebaTRegLin 68
 pruebaZ 144
 pruebaZ_1Prop, prueba Z de una
 proporción 144
 pruebaZ_2Muest, prueba Z de dos
 muestras 145
 pruebaZ_2Prop, prueba Z de dos
 proporciones 145
 punto
 agregar, .+ 150
 división, .÷ 151
 multiplicación, .* 150
 potencia, .^ 151
 producto, pPunto() 40
 sustracción, .- 150
Q
 QR, factorización de QR 95
R
 r, radián 160
 R▶Pθ(), coordenada polar 98

▶Pr(), coordenada polar 98
 Racionalaprox() 11
 ▶Rad, convertir a ángulo en radianes 98
 radián, r 160
 RaícesPoli() 93
 RaícesPoliC() 27
 raíz cuadrada
 plantilla para 1
 raíz cuadrada, $\sqrt{}$ () 121, 156
 raíz enésima
 plantilla para 1
 real, real() 100
 real(), real 100
 recíproco, \wedge^{-1} 163
 recopilación trigonométrica,
 recopilT() 129
 recopilT(), recopilación
 trigonométrica 129
 ▶Rect, se despliega como vector
 rectangular 100
 redondear, redondear() 106
 redondear(), redondear 106
 ref(), forma escalonada por filas 101
 RegCuad, regresión cuadrática 96
 RegCuart, regresión cuártica 97
 RegCúbica, regresión cúbica 31
 RegExp, regresión exponencial 46
 RegLn, regresión logarítmica 70
 RegMult 79
 RegPot, regresión de potencia 93
 regresión cuadrática, RegCuad 96
 regresión cuártica, RegCuart 97
 regresión cúbica, RegCúbica 31
 regresión de línea media-media
 (MedMed) 77
 regresión de potencia, RegPot 93,
 102, 103, 129
 regresión exponencial, RegExp 46
 regresión lineal, AxRegLin 66
 regresión lineal, BxRegLin 65, 67
 regresión logarítmica, RegLn 70
 regresión logística, Logística 73
 regresión logística, LogísticaD 74
 regresión sinusoidal, RegSin 117
 regresiones
 cuadrática, RegCuad 96
 cuártica, RegCuart 97
 cúbica, RegCúbica 31
 exponencial, RegExp 46
 línea media-media (MedMed) 77
 logarítmica, RegLn 70
 Logística 73
 logística, Logística 74
 RegMult 79
 regresión de potencia, RegPot
 93, 102, 103, 129
 regresión lineal, AxRegLin 66
 regresión lineal, BxRegLin 65, 67
 sinusoidal, RegSin 117
 RegSin, regresión sinusoidal 117
 resolverEd(), solución 37
 respuesta (última), Ans 11
 rest(), resto 102
 resto, rest() 102
 resultado
 se despliega como e 44
 se despliega en términos de
 coseno 22
 se despliega en términos de seno
 115
 resultados de dos variables, DosVar
 135
 resultados, estadísticas 122
 ResumenNúmCinco 49
 right, right() 21, 43, 61, 105, 138
 rk23(), Runge Kutta function 105
 rotar, rotar() 105, 106
 rotar(), rotar 105, 106
 rzcuaad(), raíz cuadrada 121

S

Salir, salir 44
 salir, Salir 44
 Σ Cap() 159
 se despliega como
 ángulo decimal, ▶DD 34
 binario, ▶Base2 14
 grado/minuto/segundo, ▶GMS 39
 hexadecimal, ▶Base16 15
 se despliega como decimal,
 ▶Base10 14
 vector cilíndrico, ▶Cilind 32
 vector esférico, ▶Esfera 121
 vector polar, ▶Polar 90

vector rectangular, ▶Rect 100
 se despliega como vector cilíndrico,
 ▶Cilind 32
 sec(), secante 108
 sec⁻¹(), secante inversa 108
 sech(), secante hiperbólica 108
 sech⁻¹(), secante hiperbólica inversa
 109
 secSuma() 125
 secuenc(), secuencia 109
 secuencia, secuenc() 109
 segunda derivada
 plantilla para 5
 SemillaAleat, semilla de número
 aleatorio 100
 ▶sen, se despliega en términos de
 seno 115
 sen(), seno 115
 sen⁻¹(), arcoseno 116
 senh(), seno hiperbólico 116
 senh⁻¹(), arcoseno hiperbólico 116
 seno
 despliega la expresión en
 términos de 115
 seno, sen() 115
 seqGen() 110
 seqn() 110
 sequence, seq() 110
 serie, serie() 111
 serie(), serie 111
 set
 mode, setMode() 112
 setMode(), set mode 112
 Si, si 58
 si, Si 58
 siFn() 59
 signo, signo() 114
 signo(), signo 114
 simult(), ecuaciones simultáneas 114
 ΣInt() 158
 sistema de ecuaciones (2 ecuaciones)
 plantilla para 3
 sistema de ecuaciones (N
 ecuaciones)
 plantilla para 3
 Sistema Operativo de Ecuaciones
 (EOS) 170
 Solicitar 102
 SolicitarCad 103
 solución, resolverEd() 37
 solucion, solucion() 118
 solucion(), solucion 118
 solucionC(), solucionar complejo 29
 solucionLin() 69
 solucionN(), solución numérica 86
 strings
 right, right() 21, 43, 61, 105, 138
 subMat(), submatriz 124, 125
 submatriz, subMat() 124, 125
 suma (Σ)
 plantilla para 4
 suma acumulativa,
 sumaAcumulativa() 31
 suma de pagos de capital 159
 suma de pagos de interés 158
 suma, Σ() 157
 suma(), sumatoria 125
 sumaAcumulativa(), suma
 acumulativa 31
 sumaSi() 125
 sumatoria, suma() 125
 sustitución con el operador "|" 163
 sustraer, - 147

T

T, trasponer 126
 tabla de amortización, tablaAmort()
 7, 13
 tablaAmort(), tabla de amortización
 7, 13
 tablaFrec() 52
 tan(), tangente 126
 tan⁻¹(), arcotangente 127
 tangente, tan() 126
 tanh(), tangente hiperbólica 127
 tanh⁻¹(), arcotangente hiperbólica
 128
 tasa de cambio promedio, TCprom()
 13
 tasa efectiva, ef() 41
 tasa interna de rendimiento, tirm()
 78
 tasa nominal, nom() 84
 taylor(), polinomio de Taylor 128

TCprom(), tasa de cambio promedio
 13
 techo, techo() 15, 16, 27
 techo(), techo 15
 terminar
 bucle, TerminarBucle 75
 función, TerminarFunc 53
 intentar, TerminarIntentar 132
 mientras, TerminarMientras 139
 para, TerminarPara 50
 programa, TerminarPrgm 94
 si, TerminarSi 58
 terminar bucle, TerminarBucle 75
 terminar función, TerminarFunc 53
 terminar mientras,
 TerminarMientras 139
 terminar si, TerminarSi 58
 TerminarIntentar, terminar intentar
 132
 TerminarMientras, terminar
 mientras 139
 término dominante,
 términoDominante() 40
 términoDominante(), término
 dominante 40
 tInv() 61
 tir(), tasa interna de rendimiento
 tasa interna de rendimiento,
 tir() 62
 tirm(), tasa interna de rendimiento
 modificada 78
 trasponer, \top 126
 trazado() 132

U

UnaVar, estadísticas de una variable
 87
 unidades
 convertir 162

V

valor absoluto
 plantilla para 3
 valor presente neto, vpn() 86
 valor tiempo del dinero, cantidad de
 pago 134
 valor tiempo del dinero, Interés 134

valor tiempo del dinero, número de
 pagos 134
 valor tiempo del dinero, Valor
 Futuro 134
 valor tiempo del dinero, valor
 presente 134
 valores de resultados, estadísticos
 123
 valorPropio, vlProp() 42
 variable
 cómo crear un nombre desde
 una cadena de caracteres 171
 variable local, Local 71
 variables
 borrar, BorrVar 36
 limpie todas las letras únicas 19
 local, Local 71
 variables y funciones
 cómo copiar 22
 variables, cómo bloquear y
 desbloquear 55, 71, 137
 varianza, varianza() 138
 varMuest(), varianza muestra 138
 varPob() 137
 vcProp(), vector propio 41
 vcUnid(), vector de unidad 137
 vector de unidad, vcUnid() 137
 vectores
 producto cruzado, pCruz() 27
 producto de punto, pPunto() 40
 se despliega como vector
 cilindrico, \blacktriangleright Cilind 32
 unidad, vcUnid() 137
 vectorPropio, vcProp() 41
 vlProp(), valorPropio 42
 vpn(), valor presente neto 86
 vtdI() 134
 vtdN() 134
 vtdPgo() 134
 vtdVF() 134
 vtdVP() 134

W

warnCodes(), Warning codes 138

X

x2, cuadrado 150

XNOR 154

xor, exclusivo booleano o 139