

# Microsoft Excel Function Reference

courtesy of

# Office Articles

[www.OfficeArticles.com](http://www.OfficeArticles.com)

## Table of Contents

---

Click to go!

[Database Functions](#)

[Financial Functions](#)

[Date and Time Functions](#)

[Logical Functions](#)

[External Functions](#)

[Lookup Functions](#)

[Engineering Functions](#)

[Math Functions](#)

[Information Functions](#)

[Statistical Functions](#)

[Text and Data Functions](#)

## Database Functions

### ***D AVERAGE(database,field,criteria)***

Averages the values in a column in a list or database that match conditions you specify.

### ***D COUNT(database,field,criteria)***

Counts the cells that contain numbers in a column in a list or database that match conditions you specify.

### ***D COUNTA(database,field,criteria)***

Counts all of the nonblank cells in a column in a list or database that match conditions you specify.

### ***D GET(database,field,criteria)***

Extracts a single value from a column in a list or database that matches conditions you specify.

### ***D MAX(database,field,criteria)***

Returns the largest number in a column in a list or database that matches conditions you specify.

### ***D MIN(database,field,criteria)***

Returns the smallest number in a column in a list or database that matches conditions you specify.

### ***D PRODUCT(database,field,criteria)***

Multiplies the values in a column in a list or database that match conditions you specify.

### ***D STDEV(database,field,criteria)***

Estimates the standard deviation of a population based on a sample, using the numbers in a column in a list or database that match conditions you specify.

**DSTDEVP(*database,field,criteria*)**

Calculates the standard deviation of a population based on the entire population, using the numbers in a column in a list or database that match conditions you specify.

**DSUM(*database,field,criteria*)**

Adds the numbers in a column in a list or database that match conditions you specify.

**DVAR(*database,field,criteria*)**

Estimates the variance of a population based on a sample, using the numbers in a column in a list or database that match conditions you specify.

**DVARP(*database,field,criteria*)**

Calculates the variance of a population based on the entire population, using the numbers in a column in a list or database that match conditions you specify.

**GETPIVOTDATA(*pivot\_table,name*)**

Returns data stored in a PivotTable report. You can use GETPIVOTDATA to retrieve summary data from a PivotTable report, provided the summary data is visible in the report.

## Date and Time Functions

### ***DATE(year,month,day)***

Returns the serial number that represents a particular date.

### ***DATEDIF(start\_date,end\_date,unit)***

Calculates the number of days, months, or years between two dates. This function is provided for compatibility with Lotus 1-2-3.

### ***DATEVALUE(date\_text)***

Returns the serial number of the date represented by date\_text. Use DATEVALUE to convert a date represented by text to a serial number.

### ***DAY(serial\_number)***

Returns the day of a date, represented by a serial number. The day is given as an integer ranging from 1 to 31.

### ***DAYS360(start\_date,end\_date,method)***

Returns the number of days between two dates based on a 360-day year (twelve 30-day months), which is used in some accounting calculations. Use this function to help compute payments if your accounting system is based on twelve 30-day months.

### ***EDATE(start\_date,months)\****

Returns the serial number that represents the date that is the indicated number of months before or after a specified date (the start\_date). Use EDATE to calculate maturity dates or due dates that fall on the same day of the month as the date of issue.

### **EOMONTH(*start\_date,months*)\***

Returns the serial number for the last day of the month that is the indicated number of months before or after start\_date. Use EOMONTH to calculate maturity dates or due dates that fall on the last day of the month.

### **HOUR(*serial\_number*)**

Returns the hour of a time value. The hour is given as an integer, ranging from 0 (12:00 A.M.) to 23 (11:00 P.M.).

### **MINUTE(*serial\_number*)**

Returns the minutes of a time value. The minute is given as an integer, ranging from 0 to 59.

### **MONTH(*serial\_number*)**

Returns the month of a date represented by a serial number. The month is given as an integer, ranging from 1 (January) to 12 (December).

### **NETWORKDAYS(*start\_date,end\_date,holidays*)\***

Returns the number of whole working days between start\_date and end\_date. Working days exclude weekends and any dates identified in holidays. Use NETWORKDAYS to calculate employee benefits that accrue based on the number of days worked during a specific term.

### **NOW( )**

Returns the serial number of the current date and time.

### **SECOND(*serial\_number*)**

Returns the seconds of a time value. The second is given as an integer in the range 0 (zero) to 59.

### ***TIME(hour,minute,second)***

Returns the decimal number for a particular time. The decimal number returned by TIME is a value ranging from 0 to 0.99999999, representing the times from 0:00:00 (12:00:00 A.M.) to 23:59:59 (11:59:59 P.M.).

### ***TIMEVALUE(time\_text)***

Returns the decimal number of the time represented by a text string. The decimal number is a value ranging from 0 (zero) to 0.99999999, representing the times from 0:00:00 (12:00:00 A.M.) to 23:59:59 (11:59:59 P.M.).

### ***TODAY( )***

Returns the serial number of the current date. The serial number is the date-time code used by Microsoft Excel for date and time calculations.

### ***WEEKDAY(serial\_number,return\_type)***

Returns the day of the week corresponding to a date. The day is given as an integer, ranging from 1 (Sunday) to 7 (Saturday), by default.

### ***WEEKNUM(serial\_num,return\_type)\****

Returns a number that indicates where the week falls numerically within a year.

### ***WORKDAY(start\_date,days,holidays)\****

Returns a number that represents a date that is the indicated number of working days before or after a date (the starting date). Working days exclude weekends and any dates identified as holidays. Use WORKDAY to exclude weekends or holidays when you calculate invoice due dates, expected delivery times, or the number of days of work performed. To view the number as a date, click Cells on the Format menu, click Date in the Category box, and then click a date format in the Type box.

***YEAR(serial\_number)***

Returns the year corresponding to a date. The year is returned as an integer in the range 1900-9999.

***YEARFRAC(start\_date,end\_date,basis)\****

Calculates the fraction of the year represented by the number of whole days between two dates (the start\_date and the end\_date). Use the YEARFRAC worksheet function to identify the proportion of a whole year's benefits or obligations to assign to a specific term.

## External Functions

### ***CALL(register\_id,argument1,...)***

Calls a procedure in a dynamic link library or code resource. Use this syntax only with a previously registered code resource, which uses arguments from the REGISTER function.

### ***CALL(file\_text,resource,type\_text,argument1,...)***

Calls a procedure in a dynamic link library or code resource. Use this syntax to simultaneously register and call a code resource for the Macintosh.

### ***CALL(module\_text,procedure,type\_text,argument1,...)***

Calls a procedure in a dynamic link library or code resource. Use this syntax to simultaneously register and call a code resource for Windows machines.

### ***REGISTER.ID(file\_text,resource,type\_text)***

Returns the register ID of the specified dynamic link library (DLL) or code resource that has been previously registered. If the DLL or code resource has not been registered, this function registers the DLL or code resource and then returns the register ID for the Macintosh.

### ***REGISTER.ID(module\_text,procedure,type\_text)***

Returns the register ID of the specified dynamic link library (DLL) or code resource that has been previously registered. If the DLL or code resource has not been registered, this function registers the DLL or code resource and then returns the register ID for Windows.

### ***SQL.REQUEST(connection\_string,output\_ref,driver\_prompt,query\_text,col\_names\_logical)\****

Connects with an external data source, and runs a query from a worksheet. SQL.REQUEST then returns the result as an array without the need for macro programming. If this function is not available, you must install the Microsoft Excel ODBC add-in (XLODBC.XLA).

# Engineering Functions

## ***BESSEL***(*x,n*)\*

Returns the modified Bessel function, which is equivalent to the Bessel function evaluated for purely imaginary arguments.

## ***BESSELJ***(*x,n*)\*

Returns the Bessel function.

## ***BESSELK***(*x,n*)\*

Returns the modified Bessel function, which is equivalent to the Bessel functions evaluated for purely imaginary arguments.

## ***BESSELY***(*x,n*)\*

Returns the Bessel function, which is also called the Weber function or the Neumann function.

## ***BIN2DEC***(*number*)\*

Converts a binary number to decimal.

## ***BIN2HEX***(*number,places*)\*

Converts a binary number to hexadecimal.

## ***BIN2OCT***(*number,places*)\*

Converts a binary number to octal.

## ***COMPLEX***(*real\_num,i\_num,suffix*)\*

Converts real and imaginary coefficients into a complex number of the form  $x + yi$  or  $x + yj$ .

***CONVERT(number,from\_unit,to\_unit)\****

Converts a number from one measurement system to another. For example, CONVERT can translate a table of distances in miles to a table of distances in kilometers.

***DEC2BIN(number,places)\****

Converts a decimal number to binary.

***DEC2HEX(number,places)\****

Converts a decimal number to hexadecimal.

***DEC2OCT(number,places)\****

Converts a decimal number to octal.

***DELTA(number1,number2)\****

Tests whether two values are equal. Returns 1 if number1 = number2; returns 0 otherwise. Use this function to filter a set of values. For example, by summing several DELTA functions you calculate the count of equal pairs. This function is also known as the Kronecker Delta function.

***ERF(lower\_limit,upper\_limit)\****

Returns the error function integrated between lower\_limit and upper\_limit.

***ERFC(x)\****

Returns the complementary ERF function integrated between x and infinity.

***GESTEP(number,step)\****

Returns 1 if number = step; returns 0 (zero) otherwise. Use this function to filter a set of values. For example, by summing several GESTEP functions you calculate the count of values that exceed a threshold.

***HEX2BIN(number,places)\****

Converts a hexadecimal number to binary.

***HEX2DEC(number)\****

Converts a hexadecimal number to decimal.

***HEX2OCT(number,places)\****

Converts a hexadecimal number to octal.

***IMABS(inumber)\****

Returns the absolute value (modulus) of a complex number in  $x + yi$  or  $x + yj$  text format.

***IMAGINARY(inumber)\****

Returns the imaginary coefficient of a complex number in  $x + yi$  or  $x + yj$  text format.

***IMARGUMENT(inumber)\****

Returns the argument

***IMCONJUGATE(inumber)\****

Returns the complex conjugate of a complex number in  $x + yi$  or  $x + yj$  text format.

***IMCOS(inumber)\****

Returns the cosine of a complex number in  $x + yi$  or  $x + yj$  text format.

***IMDIV(inumber1,inumber2)\****

Returns the quotient of two complex numbers in  $x + yi$  or  $x + yj$  text format.

***IMEXP(inumber)\****

Returns the exponential of a complex number in  $x + yi$  or  $x + yj$  text format.

### ***IMLN(inumber)\****

Returns the natural logarithm of a complex number in  $x + yi$  or  $x + yj$  text format.

### ***IMLOG10(inumber)\****

Returns the common logarithm (base 10) of a complex number in  $x + yi$  or

## Information Functions

### ***ISLOGICAL(value)***

Returns TRUE if Value refers to a logical value.

### ***ISNONTEXT(value)***

Returns TRUE if Value refers to any item that is not text. (Note that this function returns TRUE if value refers to a blank cell.)

### ***IMLOG2(inumber)\****

Returns the base-2 logarithm of a complex number in  $x + yi$  or  $x + yj$  text format.

### ***IMPOWER(inumber,number)\****

Returns a complex number in  $x + yi$  or  $x + yj$  text format raised to a power.

### ***IMPRODUCT(inumber1,inumber2,...)\****

Returns the product of 2 to 29 complex numbers in  $x + yi$  or  $x + yj$  text format.

### ***IMREAL(inumber)\****

Returns the real coefficient of a complex number in  $x + yi$  or  $x + yj$  text format.

### ***IMSIN(inumber)\****

Returns the sine of a complex number in  $x + yi$  or  $x + yj$  text format.

### ***IMSQRT(inumber)\****

Returns the square root of a complex number in  $x + yi$  or  $x + yj$  text format.

### ***IMSUB(inumber1,inumber2)\****

Returns the difference of two complex numbers in  $x + yi$  or  $x + yj$  text format.

***IMSUM(inumber1,inumber2,...)\****

Returns the sum of two or more complex numbers in x + yi or x + yj text format.

***OCT2BIN(number,places)\****

Converts an octal number to binary.

***OCT2DEC(number)\****

Converts an octal number to decimal.

***OCT2HEX(number,places)***

Converts an octal number to hexadecimal.

## Financial Functions

***ACCRINT(issue,first\_interest,Settlement,rate,par,frequency,basis)\****

Returns the accrued interest for a security that pays periodic interest.

***ACCRINTM(issue,maturity,rate,par,basis)\****

Returns the accrued interest for a security that pays interest at maturity.

***AMORDEGRC(cost,date\_purchased,first\_period,salvage,period,rate,basis)\****

Returns the depreciation for each accounting period. This function is provided for the French accounting system. If an asset is purchased in the middle of the accounting period, the prorated depreciation is taken into account. The function is similar to AMORLINC, except that a depreciation coefficient is applied in the calculation depending on the life of the assets.

***AMORLINC(cost,date\_purchased,first\_period,salvage,period,rate,basis)\****

Returns the depreciation for each accounting period. This function is provided for the French accounting system. If an asset is purchased in the middle of the accounting period, the prorated depreciation is taken into account.

***COUPDAYBS(settlement,maturity,frequency,basis)\****

Returns the number of days from the beginning of the coupon period to the settlement date.

***COUPDAYS(settlement,maturity,frequency,basis)\****

Returns the number of days in the coupon period that contains the settlement date.

***COUPDAYSNC(settlement,maturity,frequency,basis)\****

Returns the number of days from the settlement date to the next coupon date.

***COUPNCD(settlement,maturity,frequency,basis)\****

Returns a number that represents the next coupon date after the settlement date. To view the number as a date, click Cells on the Format menu, click Date in the Category box, and then click a date format in the Type box.

***COUPNUM(settlement,maturity,frequency,basis)\****

Returns the number of coupons payable between the settlement date and maturity date, rounded up to the nearest whole coupon.

***COUPPCD(settlement,maturity,frequency,basis)\****

Returns a number that represents the previous coupon date before the settlement date. To view the number as a date, click Cells on the Format menu, click Date in the Category box, and then click a date format in the Type box.

***CUMIPMT(rate,nper,pv,start\_period,end\_period,type)\****

Returns the cumulative interest paid on a loan between start\_period and end\_period.

***CUMPRINC(rate,nper,pv,start\_period,end\_period,type)\****

Returns the cumulative principal paid on a loan between start\_period and end\_period.

***DB(cost,salvage,life,period,month)***

Returns the depreciation of an asset for a specified period using the fixed-declining balance method.

***DDB(cost,salvage,life,period,factor)***

Returns the depreciation of an asset for a specified period using the double-declining balance method or some other method you specify.

***DISC(settlement,maturity,pr,redemption,basis)\****

Returns the discount rate for a security.

**DOLLARDE(*fractional\_dollar,fraction*)\***

Converts a dollar price expressed as a fraction into a dollar price expressed as a decimal number. Use DOLLARDE to convert fractional dollar numbers, such as securities prices, to decimal numbers.

**DOLLARFR(*decimal\_dollar,fraction*)\***

Converts a dollar price expressed as a decimal number into a dollar price expressed as a fraction. Use DOLLARFR to convert decimal numbers to fractional dollar numbers, such as securities prices.

**DURATION(*settlement,maturity,coupon yld,frequency,basis*)\***

Returns the Macauley duration for an assumed par value of \$100. Duration is defined as the weighted average of the present value of the cash flows and is used as a measure of a bond price's response to changes in yield.

**EFFECT(*nominal\_rate,npery*)\***

Returns the effective annual interest rate, given the nominal annual interest rate and the number of compounding periods per year.

**FV(*rate,nper,pmt,pv,type*)**

Returns the future value of an investment based on periodic, constant payments and a constant interest rate.

**FVSCHEDULE(*principal,schedule*)\***

Returns the future value of an initial principal after applying a series of compound interest rates. Use FVSCHEDULE to calculate future value of an investment with a variable or adjustable rate.

**INTRATE(*settlement,maturity,investment,redemption,basis*)\***

Returns the interest rate for a fully invested security.

### ***IPMT(rate,per,nper,pv,fv,type)***

Returns the interest payment for a given period for an investment based on periodic, constant payments and a constant interest rate. For a more complete description of the arguments in IPMT and for more information about annuity functions, see PV.

### ***IRR(values,guess)***

Returns the internal rate of return for a series of cash flows represented by the numbers in values. These cash flows do not have to be even, as they would be for an annuity. However, the cash flows must occur at regular intervals, such as monthly or annually. The internal rate of return is the interest rate received for an investment consisting of payments (negative values) and income (positive values) that occur at regular periods.

### ***ISPMT(rate,per,nper,pv)***

Calculates the interest paid during a specific period of an investment. This function is provided for compatibility with Lotus 1-2-3.

### ***MDURATION(settlement,maturity,coupon,yld,frequency,basis)\****

Returns the modified duration for a security with an assumed par value of \$100.

### ***MIRR(values,finance\_rate,reinvest\_rate)***

Returns the modified internal rate of return for a series of periodic cash flows. MIRR considers both the cost of the investment and the interest received on reinvestment of cash.

### ***NOMINAL(effect\_rate,npety)\****

Returns the nominal annual interest rate, given the effective rate and the number of compounding periods per year.

### ***NPER(rate, pmt, pv, fv, type)***

Returns the number of periods for an investment based on periodic, constant payments and a constant interest rate.

***NPV(rate,value1,value2, ...)***

Calculates the net present value of an investment by using a discount rate and a series of future payments (negative values) and income (positive values).

***ODDFPRICE(settlement,maturity,issue,first\_coupon,rate,yld,redemption,frequency,basis)\****

Returns the price per \$100 face value of a security having an odd (short or long) first period.

***ODDFYIELD(settlement,maturity,issue,first\_coupon,rate,pr,redemption,frequency,basis)\****

Returns the yield of a security that has an odd (short or long) first period.

***ODDLPRICE(settlement,maturity,last\_interest,rate,yld,redemption,frequency,basis)\****

Returns the price per \$100 face value of a security having an odd (short or long) last coupon period.

***ODDLYIELD(settlement,maturity,last\_interest,rate,pr,redemption,frequency,basis)\****

Returns the yield of a security that has an odd (short or long) last period.

***PMT(rate,nper,pv,fv,type)***

Calculates the payment for a loan based on constant payments and a constant interest rate.

***PPMT(rate,per,nper,pv,fv,type)***

Returns the payment on the principal for a given period for an investment based on periodic, constant payments and a constant interest rate.

***PRICE(settlement,maturity,rate,yld,redemption,frequency,basis)\****

Returns the price per \$100 face value of a security that pays periodic interest.

***PRICEDISC(settlement,maturity,discount,redemption,basis)\****

Returns the price per \$100 face value of a discounted security.

***PRICEMAT(settlement,maturity,issue,rate,yld,basis)\****

Returns the price per \$100 face value of a security that pays interest at maturity.

***PV(rate,nper,pmt,fv,type)***

Returns the present value of an investment. The present value is the total amount that a series of future payments is worth now. For example, when you borrow money, the loan amount is the present value to the lender.

***RATE(nper,pmt,pv,fv,type,guess)***

Returns the interest rate per period of an annuity. RATE is calculated by iteration and can have zero or more solutions. If the successive results of RATE do not converge to within 0.0000001 after 20 iterations, RATE returns the #NUM! error value.

***RECEIVED(settlement,maturity,investment,discount,basis)\****

Returns the amount received at maturity for a fully invested security.

***SLN(cost,salvage,life)***

Returns the straight-line depreciation of an asset for one period.

***SYD(cost,salvage,life,per)***

Returns the sum-of-years' digits depreciation of an asset for a specified period.

***TBILLEQ(settlement,maturity,discount)\****

Returns the bond-equivalent yield for a Treasury bill.

***TBILLPRICE(settlement,maturity,discount)\****

Returns the price per \$100 face value for a Treasury bill.

***TBILLYIELD(settlement,maturity,pr)\****

Returns the yield for a Treasury bill.

**VDB(cost,salvage,life,start\_period,end\_period,factor,no\_switch)**

Returns the depreciation of an asset for any period you specify, including partial periods, using the double-declining balance method or some other method you specify. VDB stands for variable declining balance.

**XIRR(values,dates,guess)\***

Returns the internal rate of return for a schedule of cash flows that is not necessarily periodic. To calculate the internal rate of return for a series of periodic cash flows, use the IRR function.

**XNPV(rate,values,dates)\***

Returns the net present value for a schedule of cash flows that is not necessarily periodic. To calculate the net present value for a series of cash flows that is periodic, use the NPV function.

**YIELD(settlement,maturity,rate,pr,redemption,frequency,basis)\***

Returns the yield on a security that pays periodic interest. Use YIELD to calculate bond yield.

**YIELDDISC(settlement,maturity,pr,redemption,basis)\***

Returns the annual yield for a discounted security.

**YIELDMAT(settlement,maturity,issue,rate,pr,basis)\***

Returns the annual yield of a security that pays interest at maturity.

**CELL(info\_type,reference)**

Returns information about the formatting, location, or contents of the upper-left cell in a reference.

**COUNTBLANK(range)**

Counts empty cells in a specified range of cells.

***ERROR.TYPE(error\_val)\****

Returns a number corresponding to one of the error values in Microsoft Excel or returns the #N/A error if no error exists. You can use ERROR.TYPE in an IF function to test for an error value and return a text string, such as a message, instead of the error value.

***INFO(type\_text)***

Returns information about the current operating environment.

***ISNUMBER(value)***

Returns TRUE if Value refers to a number.

***ISBLANK(value)***

Returns TRUE if Value refers to an empty cell.

***ISERROR(value)***

Returns TRUE if Value refers to any error value (#N/A, #VALUE!, #REF!, #DIV/0!, #NUM!, #NAME?, or #NULL!).

***ISEVEN(number)\****

Returns TRUE if number is even, or FALSE if number is odd.

***ISODD(number)\****

Returns TRUE if number is odd, or FALSE if number is even.

***ISTEXT(value)***

Returns TRUE if Value refers to text.

***N(value)***

Returns a value converted to a number.

### **NA( )**

Returns the error value #N/A. #N/A is the error value that means "no value is available." Use NA to mark empty cells. By entering #N/A in cells where you are missing information, you can avoid the problem of unintentionally including empty cells in your calculations. (When a formula refers to a cell containing #N/A, the formula returns the #N/A error value.)

### **TYPE(value)**

Returns the type of value. Use TYPE when the behavior of another function depends on the type of value in a particular cell.

## Logical Functions

### ***AND(logical1,logical2,...)***

Returns TRUE if all its arguments are TRUE; returns FALSE if one or more arguments is FALSE.

### ***FALSE( )***

Returns the logical value FALSE.

### ***IF(logical\_test,value\_if\_true,value\_if\_false)***

Returns one value if a condition you specify evaluates to TRUE and another value if it evaluates to FALSE.

### ***NOT(logical)***

Reverses the value of its argument. Use NOT when you want to make sure a value is not equal to one particular value.

### ***OR(logical1,logical2,...)***

Returns TRUE if any argument is TRUE; returns FALSE if all arguments are FALSE.

### ***TRUE( )***

Returns the logical value TRUE.

# Lookup Functions

## ***ADDRESS(row\_num,column\_num,abs\_num,a1,sheet\_text)***

Creates a cell address as text, given specified row and column numbers.

## ***AREAS(reference)***

Returns the number of areas in a reference. An area is a range of contiguous cells or a single cell.

## ***CHOOSE(index\_num,value1,value2,...)***

Uses index\_num to return a value from the list of value arguments. Use CHOOSE to select one of up to 29 values based on the index number. For example, if value1 through value7 are the days of the week, CHOOSE returns one of the days when a number between 1 and 7 is used as index\_num.

## ***COLUMN(reference)***

Returns the column number of the given reference.

## ***COLUMNS(array)***

Returns the number of columns in an array or reference.

## ***HLOOKUP(lookup\_value,table\_array,row\_index\_num,range\_lookup)***

Searches for a value in the top row of a table or an array of values, and then returns a value in the same column from a row you specify in the table or array. Use HLOOKUP when your comparison values are located in a row across the top of a table of data, and you want to look down a specified number of rows. Use VLOOKUP when your comparison values are located in a column to the left of the data you want to find.

### ***HYPERNLINK(link\_location,friendly\_name)***

Creates a shortcut or jump that opens a document stored on a network server, an intranet, or the Internet. When you click the cell that contains the HYPERLINK function, Microsoft Excel opens the file stored at link\_location.

### ***INDEX(array,row\_num,column\_num)***

Returns the value of a specified cell or array of cells within array.

### ***INDEX(reference,row\_num,column\_num,area\_num)***

Returns a reference to a specified cell or cells within reference.

### ***INDIRECT(ref\_text,a1)***

Returns the reference specified by a text string. References are immediately evaluated to display their contents. Use INDIRECT when you want to change the reference to a cell within a formula without changing the formula itself.

### ***LOOKUP(lookup\_value,lookup\_vector,result\_vector)***

Returns a value either from a one-row or one-column range. This vector form of LOOKUP looks in a one-row or one-column range (known as a vector) for a value and returns a value from the same position in a second one-row or one-column range. Included for compatibility with other worksheets. Use VLOOKUP instead.

### ***LOOKUP(lookup\_value,array)***

Returns a value from an array. The array form of LOOKUP looks in the first row or column of an array for the specified value and returns a value from the same position in the last row or column of the array. Included for compatibility with other spreadsheet programs. Use VLOOKUP instead.

### ***MATCH(lookup\_value,lookup\_array,match\_type)***

Returns the relative position of an item in an array that matches a specified value in a specified order. Use MATCH instead of one of the LOOKUP functions when you need the position of an item in a range instead of the item itself.

### ***OFFSET(reference,rows,cols,height,width)***

Returns a reference to a range that is a specified number of rows and columns from a cell or range of cells. The reference that is returned can be a single cell or a range of cells. You can specify the number of rows and the number of columns to be returned.

### ***ROW(reference)***

Returns the row number of a reference.

### ***ROWS(array)***

Returns the number of rows in a reference or array.

### ***TRANSPOSE(array)***

Returns a vertical range of cells as a horizontal range, or vice versa. TRANSPOSE must be entered as an array formula in a range that has the same number of rows and columns, respectively, as array has columns and rows. Use TRANSPOSE to shift the vertical and horizontal orientation of an array on a worksheet. For example, some functions, such as LINEST, return horizontal arrays. LINEST returns a horizontal array of the slope and Y-intercept for a line.

### ***VLOOKUP(lookup\_value,table\_array,col\_index\_num,range\_lookup)***

Searches for a value in the leftmost column of a table, and then returns a value in the same row from a column you specify in the table. Use VLOOKUP instead of HLOOKUP when your comparison values are located in a column to the left of the data you want to find.

## Math Functions

### ***ABS(number)***

Returns the absolute value of a number. The absolute value of a number is the number without its sign.

### ***ACOS(number)***

Returns the arccosine of a number. The arccosine is the angle whose cosine is number. The returned angle is given in radians in the range 0 (zero) to pi.

### ***ACOSH(number)***

Returns the inverse hyperbolic cosine of a number. Number must be greater than or equal to 1. The inverse hyperbolic cosine is the value whose hyperbolic cosine is number, so ACOSH(COSH(number)) equals number.

### ***ASIN(number)***

Returns the arcsine of a number. The arcsine is the angle whose sine is number. The returned angle is given in radians in the range -pi/2 to pi/2.

### ***ASINH(number)***

Returns the inverse hyperbolic sine of a number. The inverse hyperbolic sine is the value whose hyperbolic sine is number, so ASINH(SINH(number)) equals number.

### ***ATAN(number)***

Returns the arctangent of a number. The arctangent is the angle whose tangent is number. The returned angle is given in radians in the range -pi/2 to pi/2.

### ***ATAN2(x\_num,y\_num)***

Returns the arctangent of the specified x- and y-coordinates. The arctangent is the angle from the x-axis to a line containing the origin (0, 0) and a point with coordinates (x\_num, y\_num). The angle is given in radians between -pi and pi, excluding -pi.

### ***ATANH(number)***

Returns the inverse hyperbolic tangent of a number. Number must be between -1 and 1 (excluding -1 and 1). The inverse hyperbolic tangent is the value whose hyperbolic tangent is number, so ATANH(TANH(number)) equals number.

### ***CEILING(number,significance)***

Returns number rounded up, away from zero, to the nearest multiple of significance. For example, if you want to avoid using pennies in your prices and your product is priced at \$4.42, use the formula =CEILING(4.42,0.05) to round prices up to the nearest nickel.

### ***COMBIN(number,number\_chosen)***

Returns the number of combinations for a given number of items. Use COMBIN to determine the total possible number of groups for a given number of items.

### ***COS(number)***

Returns the cosine of the given angle.

### ***COSH(number)***

Returns the hyperbolic cosine of a number.

### ***COUNTIF(range,criteria)***

Counts the number of cells within a range that meet the given criteria.

### ***DEGREES(angle)***

Converts radians into degrees.

### **EVEN(number)**

Returns number rounded up to the nearest even integer. You can use this function for processing items that come in twos. For example, a packing crate accepts rows of one or two items. The crate is full when the number of items, rounded up to the nearest two, matches the crate's capacity.

### **EXP(number)**

Returns e raised to the power of number. The constant e equals 2.71828182845904, the base of the natural logarithm.

### **FACT(number)**

Returns the factorial of a number. The factorial of a number is equal to  $1*2*3*...* \text{number}$ .

### **FACTDOUBLE(number)\***

Returns the double factorial of a number.

### **FLOOR(number,significance)**

Rounds number down, toward zero, to the nearest multiple of significance.

### **GCD(number1,number2, ...)\***

Returns the greatest common divisor of two or more integers. The greatest common divisor is the largest integer that divides both number1 and number2 without a remainder.

### **INT(number)**

Rounds a number down to the nearest integer.

### **ISNA(value)**

Returns TRUE if Value refers to the #N/A (value not available) error value.

### ***LCM(number1,number2, ...)\****

Returns the least common multiple of integers. The least common multiple is the smallest positive integer that is a multiple of all integer arguments number1, number2, and so on. Use LCM to add fractions with different denominators.

### ***LN(number)***

Returns the natural logarithm of a number. Natural logarithms are based on the constant e (2.71828182845904).

### ***LOG(number,base)***

Returns the logarithm of a number to the base you specify.

### ***LOG10(number)***

Returns the base-10 logarithm of a number.

### ***MDETERM(array)***

Returns the matrix determinant of an array.

### ***MINVERSE(array)***

Returns the inverse matrix for the matrix stored in an array.

### ***MMULT(array1,array2)***

Returns the matrix product of two arrays. The result is an array with the same number of rows as array1 and the same number of columns as array2.

### ***MOD(number,divisor)***

Returns the remainder after number is divided by divisor. The result has the same sign as divisor.

### ***MROUND(number,multiple)\****

Returns a number rounded to the desired multiple.

### **MULTINOMIAL(*number1,number2, ...*)\***

Returns the ratio of the factorial of a sum of values to the product of factorials.

### **ODD(*number*)**

Returns number rounded up to the nearest odd integer.

### **PI( )**

Returns the number 3.14159265358979, the mathematical constant pi, accurate to 15 digits.

### **POWER(*number,power*)**

Returns the result of a number raised to a power.

### **PRODUCT(*number1,number2, ...*)**

Multiplies all the numbers given as arguments and returns the product.

### **QUOTIENT(*numerator,denominator*)\***

Returns the integer portion of a division. Use this function when you want to discard the remainder of a division.

### **RADIANS(*angle*)**

Converts degrees to radians.

### **RAND( )**

Returns an evenly distributed random number greater than or equal to 0 and less than 1. A new random number is returned every time the worksheet is calculated.

### **RANDBETWEEN(*bottom,top*)\***

Returns a random number between the numbers you specify. A new random number is returned every time the worksheet is calculated.

***ROMAN(number,form)***

Converts an arabic numeral to roman, as text.

***ROUND(number,num\_digits)***

Rounds a number to a specified number of digits.

***ROUNDDOWN(number,num\_digits)***

Rounds a number down, toward zero.

***ROUNDUP(number,num\_digits)***

Rounds a number up, away from 0 (zero).

***SERIESSUM(x,n,m,coefficients)\****

Returns the sum of a power series based on the formula:

***SIGN(number)***

Determines the sign of a number. Returns 1 if the number is positive, zero (0) if the number is 0, and -1 if the number is negative.

***SIN(number)***

Returns the sine of the given angle.

***SINH(number)***

Returns the hyperbolic sine of a number.

***SQRT(number)***

Returns a positive square root.

***SQRTPI(number)\****

Returns the square root of (number \* pi).

### **SUBTOTAL(function\_num,ref1,ref2,...)**

Returns a subtotal in a list or database. It is generally easier to create a list with subtotals using the Subtotals command (Data menu). Once the subtotal list is created, you can modify it by editing the SUBTOTAL function.

### **SUM(number1,number2, ...)**

Adds all the numbers in a range of cells.

### **SUMIF(range,criteria,sum\_range)**

Adds the cells specified by a given criteria.

### **SUMPRODUCT(array1,array2,array3, ...)**

Multiplies corresponding components in the given arrays, and returns the sum of those products.

### **SUMSQ(number1,number2, ...)**

Returns the sum of the squares of the arguments.

### **SUMX2MY2(array\_x,array\_y)**

Returns the sum of the difference of squares of corresponding values in two arrays.

### **SUMX2PY2(array\_x,array\_y)**

Returns the sum of the sum of squares of corresponding values in two arrays. The sum of the sum of squares is a common term in many statistical calculations.

### **SUMXMY2(array\_x,array\_y)**

Returns the sum of squares of differences of corresponding values in two arrays.

### **TAN(number)**

Returns the tangent of the given angle.

### ***TANH(number)***

Returns the hyperbolic tangent of a number.

### ***TRUNC(number,num\_digits)***

Truncates a number to an integer by removing the fractional part of the number.

## Statistical Functions

### ***AVEDEV(number1,number2,...)***

Returns the average of the absolute deviations of data points from their mean. AVEDEV is a measure of the variability in a data set.

### ***AVERAGE(number1,number2,...)***

Returns the average (arithmetic mean) of the arguments.

### ***AVERAGEA(value1,value2,...)***

Calculates the average (arithmetic mean) of the values in the list of arguments. In addition to numbers, text and logical values such as TRUE and FALSE are included in the calculation.

### ***BETADIST(x,alpha,beta,A,B)***

Returns the cumulative beta probability density function. The cumulative beta probability density function is commonly used to study variation in the percentage of something across samples, such as the fraction of the day people spend watching television.

### ***BETAINV(probability,alpha,beta,A,B)***

Returns the inverse of the cumulative beta probability density function. That is, if probability = BETADIST(x,...), then BETAINV(probability,...) = x. The cumulative beta distribution can be used in project planning to model probable completion times given an expected completion time and variability.

### ***BINOMDIST(number\_s,trials,probability\_s,cumulative)***

Returns the individual term binomial distribution probability. Use BINOMDIST in problems with a fixed number of tests or trials, when the outcomes of any trial are only success or failure, when trials are independent, and when the probability of success is constant throughout the

experiment. For example, BINOMDIST can calculate the probability that two of the next three babies born are male.

### ***CHIDIST(x,degrees\_freedom)***

Returns the one-tailed probability of the chi-squared distribution. The  $\chi^2$  distribution is associated with a  $\chi^2$  test. Use the  $\chi^2$  test to compare observed and expected values. For example, a genetic experiment might hypothesize that the next generation of plants will exhibit a certain set of colors. By comparing the observed results with the expected ones, you can decide whether your original hypothesis is valid.

### ***CHIINV(probability,degrees\_freedom)***

Returns the inverse of the one-tailed probability of the chi-squared distribution. If probability = CHIDIST(x,...), then CHIINV(probability,...) = x. Use this function to compare observed results with expected ones to decide whether your original hypothesis is valid.

### ***CHITEST(actual\_range,expected\_range)***

Returns the test for independence. CHITEST returns the value from the chi-squared ( $\chi^2$ ) distribution for the statistic and the appropriate degrees of freedom. You can use  $\chi^2$  tests to determine whether hypothesized results are verified by an experiment.

### ***CONFIDENCE(alpha,standard\_dev,size)***

Returns the confidence interval for a population mean. The confidence interval is a range on either side of a sample mean. For example, if you order a product through the mail, you can determine, with a particular level of confidence, the earliest and latest the product will arrive.

### ***CORREL(array1,array2)***

Returns the correlation coefficient of the array1 and array2 cell ranges. Use the correlation coefficient to determine the relationship between two properties. For example, you can examine the relationship between a location's average temperature and the use of air conditioners.

### **COUNT(value1,value2,...)**

Counts the number of cells that contain numbers and numbers within the list of arguments. Use COUNT to get the number of entries in a number field in a range or array of numbers.

### **COUNTA(value1,value2,...)**

Counts the number of cells that are not empty and the values within the list of arguments. Use COUNTA to count the number of cells that contain data in a range or array.

### **COVAR(array1,array2)**

Returns covariance, the average of the products of deviations for each data point pair. Use covariance to determine the relationship between two data sets. For example, you can examine whether greater income accompanies greater levels of education.

### **CRITBINOM(trials,probability\_s,alpha)**

Returns the smallest value for which the cumulative binomial distribution is greater than or equal to a criterion value. Use this function for quality assurance applications. For example, use CRITBINOM to determine the greatest number of defective parts that are allowed to come off an assembly line run without rejecting the entire lot.

### **DEVSQ(number1,number2,...)**

Returns the sum of squares of deviations of data points from their sample mean.

### **EXPONDIST(x,lambda,cumulative)**

Returns the exponential distribution. Use EXPONDIST to model the time between events, such as how long an automated bank teller takes to deliver cash. For example, you can use EXPONDIST to determine the probability that the process takes at most 1 minute.

### **FDIST(x,degrees\_freedom1,degrees\_freedom2)**

Returns the F probability distribution. You can use this function to determine whether two data sets have different degrees of diversity. For example, you can examine test scores given to men

and women entering high school and determine if the variability in the females is different from that found in the males.

### ***FINV(probability,degrees\_freedom1,degrees\_freedom2)***

Returns the inverse of the F probability distribution. If  $p = \text{FDIST}(x, \dots)$ , then  $\text{FINV}(p, \dots) = x$ .

### ***FISHER(x)***

Returns the Fisher transformation at  $x$ . This transformation produces a function that is approximately normally distributed rather than skewed. Use this function to perform hypothesis testing on the correlation coefficient.

### ***FISHERINV(y)***

Returns the inverse of the Fisher transformation. Use this transformation when analyzing correlations between ranges or arrays of data. If  $y = \text{FISHER}(x)$ , then  $\text{FISHERINV}(y) = x$ .

### ***ISERR(value)***

Returns TRUE if Value refers to any error value except #N/A.

### ***ISREF(value)***

Returns TRUE if Value refers to a reference.

### ***FORECAST(x,known\_y's,known\_x's)***

Calculates, or predicts, a future value by using existing values. The predicted value is a y-value for a given x-value. The known values are existing x-values and y-values, and the new value is predicted by using linear regression. You can use this function to predict future sales, inventory requirements, or consumer trends.

**FREQUENCY(data\_array,bins\_array)**

Calculates how often values occur within a range of values, and then returns a vertical array of numbers. For example, use FREQUENCY to count the number of test scores that fall within ranges of scores. Because FREQUENCY returns an array, it must be entered as an array formula.

**FTEST(array1,array2)**

Returns the result of an F-test. An F-test returns the one-tailed probability that the variances in array1 and array2 are not significantly different. Use this function to determine whether two samples have different variances. For example, given test scores from public and private schools, you can test whether these schools have different levels of diversity.

**GAMMADIST(x,alpha,beta,cumulative)**

Returns the gamma distribution. You can use this function to study variables that may have a skewed distribution. The gamma distribution is commonly used in queuing analysis.

**GAMMAINV(probability,alpha,beta)**

Returns the inverse of the gamma cumulative distribution. If p = GAMMADIST(x,...), then GAMMAINV(p,...) = x.

**GAMMALN(x)**

Returns the natural logarithm of the gamma function, G(x).

**GEOMEAN(number1,number2, ...)**

Returns the geometric mean of an array or range of positive data. For example, you can use GEOMEAN to calculate average growth rate given compound interest with variable rates.

**GROWTH(known\_y's,known\_x's,new\_x's,const)**

Calculates predicted exponential growth by using existing data. GROWTH returns the y-values for a series of new x-values that you specify by using existing x-values and y-values. You can

also use the GROWTH worksheet function to fit an exponential curve to existing x-values and y-values.

### ***HARMEAN(number1,number2, ...)***

Returns the harmonic mean of a data set. The harmonic mean is the reciprocal of the arithmetic mean of reciprocals.

### ***HYPGEOMDIST(sample\_s,number\_sample,population\_s,number\_population)***

Returns the hypergeometric distribution. HYPGEOMDIST returns the probability of a given number of sample successes, given the sample size, population successes, and population size. Use HYPGEOMDIST for problems with a finite population, where each observation is either a success or a failure, and where each subset of a given size is chosen with equal likelihood.

### ***INTERCEPT(known\_y's,known\_x's)***

Calculates the point at which a line will intersect the y-axis by using existing x-values and y-values. The intercept point is based on a best-fit regression line plotted through the known x-values and known y-values. Use the intercept when you want to determine the value of the dependent variable when the independent variable is 0 (zero). For example, you can use the INTERCEPT function to predict a metal's electrical resistance at 0°C when your data points were taken at room temperature and higher.

### ***KURT(number1,number2, ...)***

Returns the kurtosis of a data set. Kurtosis characterizes the relative peakedness or flatness of a distribution compared with the normal distribution. Positive kurtosis indicates a relatively peaked distribution. Negative kurtosis indicates a relatively flat distribution.

### ***LARGE(array,k)***

Returns the k-th largest value in a data set. You can use this function to select a value based on its relative standing. For example, you can use LARGE to return the highest, runner-up, or third-place score.

***LINEST(known\_y's,known\_x's,const,stats)***

Calculates the statistics for a line by using the "least squares" method to calculate a straight line that best fits your data, and returns an array that describes the line. Because this function returns an array of values, it must be entered as an array formula. For more information about array formulas, click

***LOGEST(known\_y's,known\_x's,const,stats)***

In regression analysis, calculates an exponential curve that fits your data and returns an array of values that describes the curve. Because this function returns an array of values, it must be entered as an array formula. For more information about array formulas, click

***LOGINV(probability,mean,standard\_dev)***

Returns the inverse of the lognormal cumulative distribution function of x, where  $\ln(x)$  is normally distributed with parameters mean and standard\_dev. If  $p = \text{LOGNORMDIST}(x, \dots)$  then  $\text{LOGINV}(p, \dots) = x$ .

***LOGNORMDIST(x,mean,standard\_dev)***

Returns the cumulative lognormal distribution of x, where  $\ln(x)$  is normally distributed with parameters mean and standard\_dev. Use this function to analyze data that has been logarithmically transformed.

***MAX(number1,number2,...)***

Returns the largest value in a set of values.

***MAXA(value1,value2,...)***

Returns the largest value in a list of arguments. Text and logical values such as TRUE and FALSE are compared as well as numbers.

**MEDIAN(number1,number2, ...)**

Returns the median of the given numbers. The median is the number in the middle of a set of numbers; that is, half the numbers have values that are greater than the median, and half have values that are less.

**MIN(number1,number2, ...)**

Returns the smallest number in a set of values.

**MINA(value1,value2,...)**

Returns the smallest value in the list of arguments. Text and logical values such as TRUE and FALSE are compared as well as numbers.

**MODE(number1,number2, ...)**

Returns the most frequently occurring, or repetitive, value in an array or range of data. Like MEDIAN, MODE is a location measure.

**NEGBINOMDIST(number\_f,number\_s,probability\_s)**

Returns the negative binomial distribution. NEGBINOMDIST returns the probability that there will be number\_f failures before the number\_s-th success, when the constant probability of a success is probability\_s. This function is similar to the binomial distribution, except that the number of successes is fixed, and the number of trials is variable. Like the binomial, trials are assumed to be independent.

**NORMDIST(x,mean,standard\_dev,cumulative)**

Returns the normal cumulative distribution for the specified mean and standard deviation. This function has a very wide range of applications in statistics, including hypothesis testing.

**NORMINV(probability,mean,standard\_dev)**

Returns the inverse of the normal cumulative distribution for the specified mean and standard deviation.

**NORMSDIST(z)**

Returns the standard normal cumulative distribution function. The distribution has a mean of 0 (zero) and a standard deviation of one. Use this function in place of a table of standard normal curve areas.

**NORMSINV(probability)**

Returns the inverse of the standard normal cumulative distribution. The distribution has a mean of zero and a standard deviation of one.

**PEARSON(array1,array2)**

Returns the Pearson product moment correlation coefficient, r, a dimensionless index that ranges from -1.0 to 1.0 inclusive and reflects the extent of a linear relationship between two data sets.

**PERCENTILE(array,k)**

Returns the k-th percentile of values in a range. You can use this function to establish a threshold of acceptance. For example, you can decide to examine candidates who score above the 90th percentile.

**PERCENTRANK(array,x,significance)**

Returns the rank of a value in a data set as a percentage of the data set. This function can be used to evaluate the relative standing of a value within a data set. For example, you can use PERCENTRANK to evaluate the standing of an aptitude test score among all scores for the test.

**PERMUT(number,number\_chosen)**

Returns the number of permutations for a given number of objects that can be selected from number objects. A permutation is any set or subset of objects or events where internal order is significant. Permutations are different from combinations, for which the internal order is not significant. Use this function for lottery-style probability calculations.

***POISSON(x,mean,cumulative)***

Returns the Poisson distribution. A common application of the Poisson distribution is predicting the number of events over a specific time, such as the number of cars arriving at a toll plaza in 1 minute.

***PROB(x\_range,prob\_range,lower\_limit,upper\_limit)***

Returns the probability that values in a range are between two limits. If upper\_limit is not supplied, returns the probability that values in x\_range are equal to lower\_limit.

***QUARTILE(array,quart)***

Returns the quartile of a data set. Quartiles often are used in sales and survey data to divide populations into groups. For example, you can use QUARTILE to find the top 25 percent of incomes in a population.

***RANK(number,ref,order)***

Returns the rank of a number in a list of numbers. The rank of a number is its size relative to other values in a list. (If you were to sort the list, the rank of the number would be its position.)

***RSQ(known\_y's,known\_x's)***

Returns the square of the Pearson product moment correlation coefficient through data points in known\_y's and known\_x's. The r-squared value can be interpreted as the proportion of the variance in y attributable to the variance in x.

***SKEW(number1,number2,...)***

Returns the skewness of a distribution. Skewness characterizes the degree of asymmetry of a distribution around its mean. Positive skewness indicates a distribution with an asymmetric tail extending toward more positive values. Negative skewness indicates a distribution with an asymmetric tail extending toward more negative values.

***SLOPE(known\_y's,known\_x's)***

Returns the slope of the linear regression line through data points in known\_y's and known\_x's. The slope is the vertical distance divided by the horizontal distance between any two points on the line, which is the rate of change along the regression line.

***SMALL(array,k)***

Returns the k-th smallest value in a data set. Use this function to return values with a particular relative standing in a data set.

***STANDARDIZE(x,mean,standard\_dev)***

Returns a normalized value from a distribution characterized by mean and standard\_dev.

***STDEV(number1,number2,...)***

Estimates standard deviation based on a sample. The standard deviation is a measure of how widely values are dispersed from the average value (the mean).

***STDEVA(value1,value2,...)***

Estimates standard deviation based on a sample. The standard deviation is a measure of how widely values are dispersed from the average value (the mean). Text and logical values such as TRUE and FALSE are included in the calculation.

***STDEVP(number1,number2,...)***

Calculates standard deviation based on the entire population given as arguments. The standard deviation is a measure of how widely values are dispersed from the average value (the mean).

***STDEVPA(value1,value2,...)***

Calculates standard deviation based on the entire population given as arguments, including text and logical values. The standard deviation is a measure of how widely values are dispersed from the average value (the mean).

***STEYX(known\_y's,known\_x's)***

Returns the standard error of the predicted y-value for each x in the regression. The standard error is a measure of the amount of error in the prediction of y for an individual x.

***TDIST(x,degrees\_freedom,tails)***

Returns the Percentage Points (probability) for the Student t-distribution where a numeric value (x) is a calculated value of t for which the Percentage Points are to be computed. The t-distribution is used in the hypothesis testing of small sample data sets. Use this function in place of a table of critical values for the t-distribution.

***TINV(probability,degrees\_freedom)***

Returns the t-value of the Student's t-distribution as a function of the probability and the degrees of freedom.

***TREND(known\_y's,known\_x's,new\_x's,const)***

Returns values along a linear trend. Fits a straight line (using the method of least squares) to the arrays known\_y's and known\_x's. Returns the y-values along that line for the array of new\_x's that you specify.

***TRIMMEAN(array,percent)***

Returns the mean of the interior of a data set. TRIMMEAN calculates the mean taken by excluding a percentage of data points from the top and bottom tails of a data set. You can use this function when you wish to exclude outlying data from your analysis.

***TTEST(array1,array2,tails,type)***

Returns the probability associated with a Student's t-Test. Use TTEST to determine whether two samples are likely to have come from the same two underlying populations that have the same mean.

**VAR(*number1,number2,...*)**

Estimates variance based on a sample.

**VARA(*value1,value2,...*)**

Estimates variance based on a sample. In addition to numbers, text and logical values such as TRUE and FALSE are included in the calculation.

**VARP(*number1,number2,...*)**

Calculates variance based on the entire population.

**VARPA(*value1,value2,...*)**

Calculates variance based on the entire population. In addition to numbers, text and logical values such as TRUE and FALSE are included in the calculation.

**WEIBULL(*x,alpha,beta,cumulative*)**

Returns the Weibull distribution. Use this distribution in reliability analysis, such as calculating a device's mean time to failure.

**ZTEST(*array,x,sigma*)**

Returns the two-tailed P-value of a z-test. The z-test generates a standard score for x with respect to the data set, array, and returns the two-tailed probability for the normal distribution. You can use this function to assess the likelihood that a particular observation is drawn from a particular population.

## Text and Data Functions

### ***ASC(text)***

Changes full-width (double-byte) English letters or katakana within a character string to half-width (single-byte) characters.

### ***CHAR(number)***

Returns the character specified by a number. Use CHAR to translate code page numbers you might get from files on other types of computers into characters.

### ***CLEAN(text)***

Removes all nonprintable characters from text. Use CLEAN on text imported from other applications that contains characters that may not print with your operating system. For example, you can use CLEAN to remove some low-level computer code that is frequently at the beginning and end of data files and cannot be printed.

### ***CODE(text)***

Returns a numeric code for the first character in a text string. The returned code corresponds to the character set used by your computer.

### ***CONCATENATE(text1, text2,...)***

Joins several text strings into one text string.

### ***DOLLAR(number,decimals)***

Converts a number to text using currency format, with the decimals rounded to the specified place. The format used is \$#,##0.00\_);(\$#,##0.00).

### ***EXACT(text1,text2)***

Compares two text strings and returns TRUE if they are exactly the same, FALSE otherwise. EXACT is case-sensitive but ignores formatting differences. Use EXACT to test text being entered into a document.

### ***FIND(find\_text,within\_text,start\_num)***

FIND finds one text string (find\_text) within another text string (within\_text), and returns the number of the starting position of find\_text, from the first character of within\_text. You can also use SEARCH to find one text string within another, but unlike SEARCH, FIND is case sensitive and doesn't allow wildcard characters.

### ***FINDB(find\_text,within\_text,start\_num)***

FINDB finds one text string (find\_text) within another text string (within\_text), and returns the number of the starting position of find\_text, based on the number of bytes each character uses, from the first character of within\_text. This function is for use with double-byte characters. You can also use SEARCHB to find one text string within another.

### ***FIXED(number,decimals,no\_commas)***

Rounds a number to the specified number of decimals, formats the number in decimal format using a period and commas, and returns the result as text.

### ***JIS(text)***

Changes half-width (single-byte) English letters or katakana within a character string to full-width (double-byte) characters.

### ***LEFT(text,num\_chars)***

LEFT returns the first character or characters in a text string, based on the number of characters you specify.

***LEFTB(text,num\_bytes)***

LEFTB returns the first character or characters in a text string, based on the number of bytes you specify. This function is for use with double-byte characters.

***LEN(text)***

LEN returns the number of characters in a text string.

***LENB(text)***

LENB returns the number of bytes used to represent the characters in a text string. This function is for use with double-byte characters.

***LOWER(text)***

Converts all uppercase letters in a text string to lowercase.

***MID(text,start\_num,num\_chars)***

MID returns a specific number of characters from a text string, starting at the position you specify, based on the number of characters you specify.

***MIDB(text,start\_num,num\_bytes)***

MIDB returns a specific number of characters from a text string, starting at the position you specify, based on the number of bytes you specify. This function is for use with double-byte characters.

***PHONETIC(reference)***

Extracts the phonetic (furigana) characters from a text string.

***PROPER(text)***

Capitalizes the first letter in a text string and any other letters in text that follow any character other than a letter. Converts all other letters to lowercase letters.

### ***REPLACE(old\_text,start\_num,num\_chars,new\_text)***

REPLACE replaces part of a text string, based on the number of characters you specify, with a different text string.

### ***REPLACEB(old\_text,start\_num,num\_bytes,new\_text)***

REPLACEB replaces part of a text string, based on the number of bytes you specify, with a different text string. This function is for use with double-byte characters.

### ***REPT(text,number\_times)***

Repeats text a given number of times. Use REPT to fill a cell with a number of instances of a text string.

### ***RIGHT(text,num\_chars)***

RIGHT returns the last character or characters in a text string, based on the number of characters you specify.

### ***RIGHTB(text,num\_bytes)***

RIGHTB returns the last character or characters in a text string, based on the number of bytes you specify. This function is for use with double-byte characters.

### ***SEARCH(find\_text,within\_text,start\_num)***

SEARCH returns the number of the character at which a specific character or text string is first found, beginning with start\_num. Use SEARCH to determine the location of a character or text string within another text string so that you can use the MID or REPLACE functions to change the text.

### ***SEARCHB(find\_text,within\_text,start\_num)***

SEARCHB also finds one text string (find\_text) within another text string (within\_text), and returns the number of the starting position of find\_text. The result is based on the number of

bytes each character uses, beginning with start\_num. This function is for use with double-byte characters You can also use FINDB to find one text string within another.

### **SUBSTITUTE(*text,old\_text,new\_text,instance\_num*)**

Substitutes new\_text for old\_text in a text string. Use SUBSTITUTE when you want to replace specific text in a text string; use REPLACE when you want to replace any text that occurs in a specific location in a text string.

### **T(*value*)**

Returns the text referred to by value.

### **TEXT(*value,format\_text*)**

Converts a value to text in a specific number format.

### **TRIM(*text*)**

Removes all spaces from text except for single spaces between words. Use TRIM on text that you have received from another application that may have irregular spacing.

### **UPPER(*text*)**

Converts text to uppercase.

### **VALUE(*text*)**

Converts a text string that represents a number to a number.

### **YEN(*number,decimals*)**

Converts a number to text, using the ¥ (yen) currency format, with the number rounded to a specified place.