

FUNCTIONS IN EXCEL

Financial Functions

ACCRINT This function returns the accrued interest for a security that pays periodic interest.

Syntax: ACCRINT (issue, first interest, settlement, rate, par, frequency, basis).

Issue – security's issues date

First interest - security's first interest date.

Settlement - security's settlement date.

Rate - security's annual coupon rate.

Par - security's par value.

Frequency - No. of coupon payments per year.

Basis - Type of day count basis to use.

ACCRINTM This function returns the accrued interest for a security that pays interest at maturity

Syntax: ACCRINTM (Issue, maturity, rate, par, basis)

Maturity - Security's maturity date.

Explanations: Same as above

AMORDEGRC This function returns the depreciation for each accounting period by using a depreciation coefficient

Syntax: AMORDEGRC (Cost, date_purchased, first_period, salvage, period, rate, basis)

Cost - Cost of asset.

Date purchased - Date of purchase of asset.

First period - Date of the end of first period.

Salvage - Salvage value at the end of life of the asset.

Period – Period, *Rate* - Rate of depreciation.

Basis - Year basis to be used.

AMORLINC This function returns the depreciation for each accounting period.

Syntax: AMORLINC (cost, date_purchased, first_period, salvage, period, rate, basis)

Date_purchased - Date of the purchase of asset.

Explanations: Same as above

COUPDAYBS This function is used to return the number of days from the beginning of the coupon period to the settlement date.

Syntax: COUPDAYBS (settlement, maturity, frequency, basis)

Maturity - Security's maturity date. The maturity date is the date when the security expires.

Basis - Type of day count basis to use.

Explanations: Same as above

COUPDAYS This function is used to return the number of days in the coupon period that contains the settlement date.

Syntax: COUPDAYS (settlement, maturity, frequency, basis)

Explanations: Same as above

COUPDAYSNC[#] This function is used to return the number of days from the settlement date to the next coupon date.

Syntax: COUPDAYSNC (settlement, maturity, frequency, basis)

Explanations: Same as above

COUPNCD[#] This function is used to return the next coupon date after the settlement date.

Syntax: COUPNCD (settlement, maturity, frequency, basis)

Explanations: Same as above

COUPNUM[#] This function is used to return the number of coupons payable between the settlement date and maturity date, rounded up to the nearest whole coupon.

Syntax: COUPNUM (settlement, maturity, frequency, basis)

Explanations: Same as above

COUPPCD[#] This function Returns a number that represents previous coupon date before the settlement date.

Syntax: COUPPCD (settlement, maturity, frequency, basis)

Explanations: Same as above

CUMIPMT[#] This function Returns the cumulative interest paid on a loan between start_period and end_period.

Syntax: CUMIPMT (rate, nper, pv, start_period, end_period, type)

Rate - Interest rate.

Nper - Total no. of payment periods.

Pv - Present value.

Start_period - First period in the calculation.

End_period - Last period in the calculation.

Type - Timing of the payment.

CUMPRINC[#] This function is used to return the cumulative principal paid on a loan between start_period and end_period.

Syntax: CUMPRINC (rate, nper, pv, start_period, end_period, type)

Explanations: Same as above

DB This function is used to return the depreciation of an asset for a specified period using the fixed-declining balance method.

Syntax: DB (cost, salvage, life, period, month)

Cost - Initial cost of asset.

Life - No. of periods over which the asset is being depreciated

Period - Period of depreciation. Period must use the same units as life.

Month – No. of months in the first year.

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

Syntax: DDB (cost, salvage, life, period, factor)

Factor - Rate at which the balance declines.

Explanations: Same as above

Important: All five arguments must be positive numbers

DISC This function is used to return the discount rate for a security

Syntax: DISC (settlement, maturity, pr, redemption, basis)

Pr - Security's price per \$100 face value.

Redemption - Security's redemption value per \$100 face value.

Explanations: Same as above

DOLLARDE[#] This function is used to convert a dollar price, expressed as a fraction, into a dollar price, expressed as a decimal number

Syntax: DOLLARDE (Fractional_dollar, fraction)

Fractional_dollar - Number expressed as a fraction.

Fraction - Integer to use in the denominator of the fraction.

DOLLARFR[#] This function is used to convert 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.

Syntax: DOLLARFR (Decimal_dollara, fraction)

Decimal_dollar - Decimal number.

DURATION This function is used return 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.

Syntax: DURATION (settlement, maturity, coupon, yld, frequency, basis)

Coupon - Security's annual coupon rate.

Yld - Security's annual yield.

Explanations: Same as above

EFFECT This function is used to return the effective annual interest rate, given the nominal annual interest rate and the number of compounding periods per year.

Syntax: EFFECT (nominal_rate, npery)

Nominal_rate - Nominal interest rate.

Npery - No. of compounding periods per year.

FV This function is used to return the future value of an investment based on periodic, constant payments and a constant interest rate.

Syntax: FV (rate, nper, pmt, pv, type)

Pmt - Payment made each period.

Explanations: Same as above

FVSCCHEDULE[#] This function is used to return the future value of an initial principal after applying a series of compound interest rates. Use FVSCCHEDULE to calculate the future value of an investment with a variable or adjustable rate.

Syntax: FVSCCHEDULE (principal, schedule)

Principal - Present value.

Schedule - Array of interest rates to apply.

INTRATE This function is used to return the future value of an initial principal after applying a series of compound interest rates. Use FVSCCHEDULE to calculate the future value of an investment with a variable or adjustable rate.

Syntax: FVSCCHEDULE (principal, schedule)

Explanations: Same as above

IPMT This function is used to return the interest payment for a given period for an investment based on periodic, constant payments and a constant interest rate.

Syntax: IPMT (rate, per, nper, pv, fv, type)

Per - Period for which you want to find the interest and must be in the range 1 to nper.

Type - Number 0 or 1 and indicates when payments are due. If type is omitted, it is assumed to be 0.

Explanations: Same as above

IRR This function is used to return 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.

Syntax: IRR (values, guess)

Values - Array or a reference to cells that contain numbers for which you want to calculate the internal rate of return.

Guess - Number that you guess is close to the result of IRR.

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

Syntax: ISPMT (rate, per, nper, pv)

Explanations: Same as above

MDURATION This function is used to return the modified Macaulay duration for a security with an assumed par value of \$100.

Syntax: MDURATION (settlement, maturity, coupon, yld, frequency, basis)

Explanations: Same as above

MIRR This function is used to return 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.

Syntax: MIRR (values, finance_rate, reinvest_rate)

Finance_rate - Interest rate you pay on the money used in the cash flows.

Reinvest_rate - Interest rate you receive on the cash flows as you reinvest them

NOMINAL This function is used to return the nominal annual interest rate, given the effective rate and the number of compounding periods per year.

Syntax: NOMINAL (Effect_rate, Npery)

Effect_rate - Effective interest rate.

Npery – No. of compounding periods per year.

NPER This function is used to return the number of periods for an investment based on periodic, constant payments and a constant interest rate.

Syntax: PER (rate, pmt, pv, fv, type)

Explanations: Same as above

NPV This function is used to calculate the net present value of an investment by using a discount rate and a series of future payments (negative values) and income (positive values).

Syntax: NPV (rate,value1,value2, ...)

Rate - Rate of discount over the length of one period.

Value1, value2, ... - 1 to 29 arguments representing the payments and income.

ODDFPRICE[#] This function is used to return the price per \$100 face value of a security having an odd (short or long) first period.

Syntax: ODDFPRICE (settlement,maturity,issue,first_coupon,rate,yld,redemption,frequency,basis)

Explanations: Same as above

Redemption - Security's redemption value per \$100 face value.

ODDFYIELD[#] This function is used to return the yield of a security that has an odd (short or long) first period.

Syntax: ODDFYIELD (settlement, maturity, issue, first_coupon, rate, pr, redemption,frequency,basis)

Explanations: Same as above

ODDLPRICE[#] This function is used to return the price per \$100 face value of a security having an odd (short or long) last coupon period.

Syntax: ODDLPRICE (settlement, maturity, last_interest, rate,yld,redemption,frequency,basis)

Explanations: Same as above

ODDLYIELD[#] This function is used to return the yield of a security that has an odd (short or long) last period.

Syntax: ODDLYIELD (settlement, maturity, last_interest, rate, pr, redemption, frequency, basis)

Explanations: Same as above

PMT This function is used to calculate payment of loan based on constant payments & constant int. rate.

Syntax: PMT (rate, nper, pv, fv, type)

Explanations: Same as above

PPMT This function is used to return the payment on the principal for a given period for an investment based on periodic, constant payments and a constant interest rate.

Syntax: PPMT (rate, per, nper, pv, fv, type)

Explanations: Same as above

PRICE This function is used to return price per \$100 FV of a security that pays periodic interest.

Syntax: PRICE (settlement, maturity, rate, yld, redemption, frequency, basis)

Explanations: Same as above

PRICEDISC This function is used to return the price per \$100 face value of a discounted security.

Syntax: PRICEDISC (settlement, maturity, discount, redemption, basis)

Explanations: Same as above

PRICEMAT This function is used to return the price per \$100 face value of a security that pays interest at maturity.

Syntax: PRICEMAT (settlement, maturity, issue, rate, yld, basis)

Explanations: Same as above

PV This function is used to return the present value of an investment. The present value is the total amount that a series of future payments is worth now.

Syntax: PV (rate, nper, pmt, fv, type)

Explanations: Same as above

RATE This function is used to return 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.

Syntax: RATE (nper, pmt, pv, fv, type, guess)

Explanations: Same as above

RECEIVED This function returns the amount received at maturity for a fully invested security.

Syntax: RECEIVED (settlement, maturity, investment, discount, basis)

Investment - Amount invested in the security.
Explanations: Same as above

SLN This function is used to return the straight-line depreciation of an asset for one period.

Syntax: SLN (cost, salvage, life)

Explanations: Same as above

SYD This function returns the sum-of-years' digits depreciation of an asset for a specified period.

Syntax: SYD (cost, salvage, life, per)

Explanations: Same as above

TBILLEQ[#] This function is used to return the bond-equivalent yield for a Treasury bill.

Syntax: TBILLEQ (settlement, maturity, discount)

Settlement - Treasury bill's settlement date.
Maturity - Treasury bill's maturity date.
Discount - Treasury bill's discount rate.

TBILLPRICE[#] This function is used to return the price per \$100 face value for a Treasury bill.

Syntax: TBILLPRICE (settlement, maturity, discount)

Explanations: Same as above

TBILLYIELD[#] This function is used to return the yield for a Treasury bill.

Syntax: TBILLYIELD (settlement, maturity, pr)

Explanations: Same as above

VDB This function is used to return 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.

Syntax: VDB (cost, salvage, life, start_period, end_period, factor, no_switch)

Explanations: Same as above
Factor -Rate at which the balance declines.
No_switch -Logical value specifying whether to switch to straight-line depreciation when depreciation is greater than the declining balance calculation.
All arguments except no_switch must be positive numbers.

XIRR This function is used to return 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.

Syntax: XIRR (values, dates, guess)

Values - Series of cash flows that corresponds to a schedule of payments in dates.

Dates - Schedule of payment dates that corresponds to the cash flow payments.

Guess - Number that you guess is close to the result of XIRR

XNPV This function is used to return NPV for a schedule of cash flows that is not necessarily periodic. To calculate NPV for a series of cash flows that is periodic, use the NPV function.

Syntax: XNPV (rate, values, dates)

Rate - Discount rate (cash flows)
Explanations: Same as above.

YIELD This function is used to return the yield on a security that pays periodic interest. Use YIELD to calculate bond yield.

Syntax: YIELD (settlement, maturity, rate, pr, redemption, frequency, basis)

Rate - Security's annual coupon rate.
Pr - Security's price per \$100 face value.
Redemption - Security's redemption value per \$100 face value.
Explanations: Same as above.

YIELDDISC[#] This function is used to return the annual yield for a discounted security.

Syntax: YIELDDISC (settlement, maturity, pr, redemption, basis)

Explanations: Same as above.

YIELDMAT[#] This function returns the annual yield of a security that pays interest at maturity.

Syntax: YIELDMAT (settlement, maturity, issue, rate, pr, basis)

Explanations: Same as above.
Issue - Security's issue date, expressed as a serial date number.

Statistical Functions

AVEDEV This function is used to return the average of the absolute deviations of data points from their mean. AVEDEV is a measure of the variability in a data set.

Syntax: AVEDEV (number1, number2,...)

Number1, number2, ... - 1 to 30 arguments for which you want the average of the absolute deviations.

AVERAGE This function is used to return the average (arithmetic mean) of its arguments

Syntax: AVERAGE (number1, number2,...)

Explanations: Same as above.

AVERAGEA This function is used to calculate the average (arithmetic mean) of the values in the list of arguments. In addition to numbers, text and logical values are included in the calculation.

Syntax: AVERAGEA (value1, value2,...)

Value1, value2, ... - 1 to 30 cells, ranges of cells, or values for which you want the average.

BETADIST[#] This function is used to return 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.

Syntax: BETADIST (x, alpha, beta, A, B)

X - Value between A and B at which to evaluate the function.

Alpha - Parameter of the distribution.

Beta - Parameter of the distribution.

A - Optional lower bound to the interval of x.

B - Optional upper bound to the interval of x.

BETAINV[#] This function is used to return 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.

Syntax: BETAINV (probability, alpha, beta, A, B)

Explanations: Same as above.

Probability - Probability associated with the beta distribution.

BINOMDIST This function is used to return 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, & 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.

Syntax: BINOMDIST (number_s, trials, probability_s, cumulative)

Number_s - No. of successes in trials.

Trials - No. of independent trials.

Probability_s - Probability of success on each trial.

Cumulative - Logical value that determines the form of the function.

CHIDIST[#] This function is used to return the one-tailed probability of the chi-squared distribution. The χ^2 distribution is associated with a χ^2 test. Use the χ^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.

Syntax: CHIDIST(x, degrees_freedom)

X - Value at which you want to evaluate the distribution.

Degrees_freedom - No. of degrees of freedom.

CHIINV[#] This function is used to return 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 in order to decide whether your original hypothesis is valid.

Syntax: CHIINV (probability, degrees_freedom)

Explanations: Same as above.

Probability - Probability associated with the chi-squared distribution.

CHITEST[#] This function is used to return the test for independence. CHITEST returns the value from the chi-squared (χ^2) distribution for the statistic and the appropriate degrees of freedom. You can use χ^2 tests to determine whether hypothesized results are verified by an experiment.

Syntax: CHITEST (actual_range, expected_range)

Actual_range - Range of data that contains observations to test against expected values.

Expected_range - Range of data that contains the ratio of the product of row totals and column totals to the grand total.

CONFIDENCE This function is used to return 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.

Syntax: CONFIDENCE (alpha, standard_dev, size)

Alpha - Significance level used to compute the confidence level.

Standard_dev - Population standard deviation for the data range and is assumed to be known.

Size - Sample size

CORREL This function is used to return the correlation coefficient of the array1 and array2 cell ranges. Use the correlation coefficient to determine the relationship between two properties.

Syntax: CORREL (array1, array2)

Array1 - Cell range of values.

Array2 - Second cell range of values.

COUNT This function is used to count the number of cells that contain numbers and also numbers within the list of arguments.

Syntax: COUNT (value1, value2,...)

Explanations: Same as above.

COUNTA This function is used to count the number of cells that are not empty and the values within the list of arguments. Use COUNTA to count the no. of cells that contain data in an array.

Syntax: COUNTA (value1, value2,...)

Explanations: Same as above.

COVAR This function is used to return covariance, the average of the products of deviations for each

data point pair. Use covariance to determine the relationship between two data sets.

Syntax: COVAR (array1, array2)

Explanations: Same as above.

CRITBINOM[#] This function is used to return 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.

Syntax: CRITBINOM (trials, probability_s, alpha)

Trials – No. of Bernoulli trials.

Probability_s - Probability of a success on each trial.

Alpha - Criterion value.

DEVSQ This function returns sum of squares of deviations of data points from their sample mean.

Syntax: DEVSQ (number1, number2,...)

EXPONDIST This function is used to return the exponential distribution.

Syntax: EXPONDIST (x, lambda, cumulative)

X - Value of the function.

Lambda - Parameter value.

Cumulative - Logical value that indicates which form of the exponential function to provide.

FDIST[#] This function is used to return the F probability distribution. You can use this function to determine whether two data sets have different degrees of diversity.

Syntax: FDIST (x, degrees_freedom1, degrees_freedom2)

X - Value at which to evaluate the function.

Degrees_freedom1 - Numerator degrees of freedom.

Degrees_freedom2 - Denominator degrees of freedom.

FINV[#] This function is used to return the inverse of the F probability distribution. If $p = \text{FDIST}(x, \dots)$, then $\text{FINV}(p, \dots) = x$.

The F distribution can be used in an F-test that compares the degree of variability in two data sets.

Syntax: FINV (probability, degrees_freedom1, degrees_freedom2)

Explanations: Same as above.

FISHER[#] This function is used to return the Fisher transformation at x. This transformation produces a function that is normally distributed rather than skewed. Use this function to perform hypothesis testing on the correlation coefficient.

Syntax: FISHER(x)

X - Numeric value for which you want the transformation.

FISHERINV[#] This function is used to return the inverse of the Fisher transformation. Use this transformation when analyzing correlations between

ranges or arrays of data. If $y = \text{FISHER}(x)$, then $\text{FISHERINV}(y) = x$.

Syntax: FISHERINV(y)

Y - Value for which you want to perform the inverse of the transformation.

FORECAST This function is used to calculate, 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.

Syntax: FORECAST (x, known_y's, known_x's)

X - Data point for which you want to predict a value.

Known_y's - Dependent array or range of data.

Known_x's - Independent array or range of data.

FREQUENCY This function is used to calculate how often values occur within a range of values, and then returns a vertical array of numbers.

Syntax: FREQUENCY (data_array, bins_array)

Data_array - Array of or reference to a set of values for which you want to count frequencies.

Bins_array - Array of or reference to intervals into which you want to group the values in data_array.

FTEST This function is used to return 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.

Syntax: FTEST (array1, array2)

Explanations: Same as above.

GAMMADIST[#] This function is used to return 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.

Syntax: GAMMADIST (x, alpha, beta, cumulative)

Explanations: Same as above.

GAMMAINV[#] This function is used to return the inverse of the gamma cumulative distribution. You can use this to study a variable whose distribution may be skewed.

Syntax: GAMMAINV (probability, alpha, beta)

Explanations: Same as above.

GAMMALN[#] This function is used to return the natural logarithm of the gamma function, $\Gamma(x)$.

Syntax: GAMMALN(x)

X - Value for which you want to calculate GAMMALN.

GEOMEAN This function is used to return the geometric mean of an array or range of positive data.

Syntax: GEOMEAN (number1, number2,...)

Explanations: Same as above.

GROWTH This function is used to calculate 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.

Syntax: GROWTH (known_y's, known_x's, new_x's, const)

Known_y's - Set of y-values you already know in the relationship $y = b \cdot m^x$.

Known_x's - Optional set of x-values that you may already know in the relationship $y = b \cdot m^x$.

New_x's - New x-values for which you want GROWTH to return corresponding y-values.

Const - Logical value specifying whether to force the constant b to equal 1.

HARMEAN This function is used to return the harmonic mean of a data set. The harmonic mean is the reciprocal of the arithmetic mean of reciprocals.

Syntax: HARMEAN (number1, number2,...)

Explanations: Same as above.

HYPGEOMDIST[#] This function is used to return the hypergeometric distribution.

Syntax: HYPGEOMDIST (sample_s, number_sample, population_s, number_population)

Sample_s - No. of successes in the sample.

Number_sample - Size of the sample.

Population_s - No. of successes in the population.

Number_population - Population size

INTERCEPT[#] This function is used to calculate 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.

Syntax: INTERCEPT (known_y's, known_x's)

Known_y's is the dependent set of observations or data.

Known_x's is the independent set of observations or data.

KURT[#] This function is used to return 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.

Syntax: KURT (number1, number2,...)

Explanations: Same as above.

LARGE[#] This function is used to return the kth largest value in a data set. You can use this function to select a value based on its relative standing.

Syntax: LARGE (array, k)

Array - Array or range of data for which you want to determine the k-th largest value.

K - Position (from the largest) in the array or cell range of data to return.

LINEST[#] This function is used to calculate 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.

The equation for the line is:

$y = mx + b$ or

$y = m_1x_1 + m_2x_2 + \dots + b$ (if there are multiple ranges of x-values)

where the dependent y-value is a function of the independent x-values. The m-values are coefficients corresponding to each x-value, and b is a constant value. Note that y, x, and m can be vectors. The array that LINEST returns is {m_n,m_{n-1},...,m₁,b}. LINEST can also return additional regression statistics.

Syntax: LINEST (known_y's, known_x's, const, stats)

Explanations: Same as above.

Stats - Logical value specifying whether to return additional regression statistics.

LOGEST[#] This function is used to return the parameters of an exponential trend

Syntax: LOGEST (known_y's, known_x's, const, stats)

Explanations: Same as above.

LOGINV[#] This function is used to return 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 $\text{LOGINV}(p, \dots) = x$.

Use the lognormal distribution to analyze logarithmically transformed data.

Syntax: LOGINV (probability, mean, standard_dev)

Explanations: Same as above.

Mean - Mean of ln(x).

Standard_dev - Standard deviation of ln(x).

LOGNORMDIST[#] This function is used to return 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.

Syntax: LOGNORMDIST (x, mean, standard_dev)

Explanations: Same as above.

MAX This function is used to return the largest value in a set of values.

Syntax: MAX (number1, number2,...)

Explanations: Same as above.

MAXA This function is used to return the largest value in a list of arguments. Text and logical values

such as TRUE and FALSE are compared as well as numbers.

MAXA is similar to MINA. For more information, see the examples for MINA.

Syntax: MAXA (value1, value2,...)

Explanations: Same as above.

MEDIAN This function is used to return 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.

Syntax: MEDIAN (number1, number2,...)

Explanations: Same as above.

MIN This function is used to return the smallest number in a set of values.

Syntax: MIN (number1, number2,...)

Explanations: Same as above.

MINA This function is used to return the smallest value in the list of arguments. Text and logical values such as TRUE and FALSE are compared as well as numbers.

Syntax: MINA (value1, value2,...)

Explanations: Same as above.

MODE This function is used to return the most frequently occurring, or repetitive, value in an array or range of data. Like MEDIAN, MODE is a location measure.

Syntax: MODE (number1, number2,...)

Explanations: Same as above.

NEGBINOMDIST This function is used to return the negative binomial distribution.

Syntax: NEGBINOMDIST (number_f, number_s, probability_s)

Number_f - No. of failures.

Number_s - Threshold no. of successes.

Probability_s - Probability of a success.

NORMDIST[#] This function is used to return the normal distribution for the specified mean and standard deviation. This function has a very wide range of applications in statistics, including hypothesis testing.

Syntax: NORMDIST (x, mean, standard_dev, cumulative)

Explanations: Same as above.

NORMINV[#] This function is used to return the inverse of the normal cumulative distribution for the specified mean and standard deviation.

Syntax: NORMINV (probability, mean, standard_dev)

Explanations: Same as above.

NORMSDIST[#] Returns the standard normal cumulative distribution. The distribution has a mean of 0 (zero) and a standard deviation of one.

Syntax: NORMSDIST (z)

Z - Value for which you want the distribution.

NORMSINV[#] This function is used to return the inverse of the standard normal cumulative distribution. The distribution has a mean of zero and a standard deviation of one.

Syntax: NORMSINV (probability)

Probability - Probability corresponding to the normal distribution.

PEARSON This function is used to return 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.

Syntax: PEARSON (array1, array2)

Array1 - Set of independent values.

Array2 - Set of dependent values.

PERCENTILE This function is used to return the k-th percentile of values in a range. You can use this function to establish a threshold of acceptance.

Syntax: PERCENTILE (array, k)

Array - Array or range of data that defines relative standing.

K - Percentile value in the range 0..1, inclusive.

PERCENTRANK[#] This function is used to return 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.

Syntax: PERCENTRANK (array, x, significance)

Array - Array or range of data with numeric values that defines relative standing.

X - Value for which you want to know the rank.

Significance - Optional value that identifies the number of significant digits for the returned percentage value.

PERMUT This function is used to return 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.

Syntax: PERMUT (number, number_chosen)

Number - Integer that describes no. of objects.

Number_chosen - Integer that describes no. of objects in each permutation

POISSON This function is used to return 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.

Syntax: POISSON (x, mean, cumulative)

X - No. of events.

Mean - Expected numeric value.

Cumulative - Logical value that determines the form of the probability distribution returned.

PROB This function is used to return 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.

Syntax: PROB (x_range, prob_range, lower_limit, upper_limit)

X_range - Range of numeric values of x with which there are associated probabilities.

Prob_range - Set of probabilities associated with values in x_range.

Lower_limit - Lower bound on the value for which you want a probability.

Upper_limit - Optional upper bound on the value for which you want a probability.

QUARTILE This function is used to return the quartile of a data set. Quartiles often are used in sales and survey data to divide populations into groups.

Syntax: QUARTILE (array, quart)

Array - Array or cell range of numeric values for which you want the quartile value.

Quart - Indicates which value to return.

RANK This function is used to return 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.)

Syntax: RANK (number, ref, order)

Number - Number whose rank you want to find.

Ref - Array of, or a reference to, a list of numbers. Nonnumeric values in ref are ignored.

Order - Number specifying how to rank number.

RSQ This function is used to return the square of the Pearson product moment correlation coefficient through data points in known_y's and known_x's.

Syntax: RSQ (known_y's, known_x's)

Known_y's - Array or range of data points.

Known_x's - Array or range of data points.

SKEW This function is used to return 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.

Syntax: SKEW (number1, number2,...)

Explanations: Same as above.

SLOPE This function is used to return 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.

Syntax: SLOPE (known_y's, known_x's)

Known_y's - Array of numeric dependent data points.

Known_x's - Set of independent data points.

SMALL[#] This function is used to return the k-th smallest value in a data set. Use this function to return values with a particular relative standing in a data set.

Syntax: SMALL (array, k)

Array - Array/range of numerical data for which you want to determine the k-th smallest value.

K - Position (from the smallest) in the array or range of data to return.

STANDARDIZE[#] This function is used to return a normalized value from a distribution characterized by mean and standard_dev.

Syntax: STANDARDIZE (x, mean, standard_dev)

X - Value you want to normalize.

Mean - Arithmetic mean of distribution.

Standard_dev - Standard deviation of distribution.

STDEV[#] This function is used to estimate standard deviation based on a sample. The standard deviation is a measure of how widely values are dispersed from the average value (the mean).

Syntax: STDEV (number1, number2,...)

Explanations: Same as above.

STDEVA[#] This function is used to estimate 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.

Syntax: STDEVA (value1, value2,...)

Explanations: Same as above.

STDEVP[#] This function is used to calculate 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).

Syntax: STDEVP (number1, number2,...)

Explanations: Same as above.

STDEVPA[#] This function is used to calculate 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).

Syntax: STDEVPA (value1, value2,...)

Explanations: Same as above.

STEYX[#] This function is used to return the standard error of the predicted y-value for each x in the regression. The standard error is a measure of amount of error in prediction of y for an individual x.

Syntax: STEYX (known_y's, known_x's)

Known_y's - Range of dependent data points.

Known_x's - Range of independent data points.

TDIST[#] This function is used to return 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.

Syntax: TDIST (x, degrees_freedom, tails)

X - Numeric value at which to evaluate the distribution.

Degrees_freedom - Integer indicating the number of degrees of freedom.

Tails - No. of distribution tails to return.

TINV[#] This function is used to return the t-value of the Student's t-distribution as a function of the probability and the degrees of freedom.

Syntax: TINV (probability, degrees_freedom)

Probability - Probability associated with the two-tailed Student's t-distribution.

Degrees_freedom - No. of degrees of freedom with which to characterize the distribution.

TREND This function is used to return 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.

Syntax: TREND (known_y's, known_x's, new_x's, const)

Known_y's is the set of y-values you already know in the relationship $y = mx + b$.

Known_x's is an optional set of x-values that you may already know in the relationship $y = mx + b$.

New_x's are new x-values for which you want TREND to return corresponding y-values.

Const is a logical value specifying whether to force the constant b to equal 0.

TRIMMEAN[#] This function is used to return 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.

Syntax: TRIMMEAN (array, percent)

Array is the array or range of values to trim and average.

Percent is the fractional number of data points to exclude from the calculation.

TTEST[#] This function is used to return the probability associated with a Student's t-Test. Use TTEST to determine whether 2 samples are likely to have come from the same two underlying populations that have the same mean.

Syntax: TTEST (array1, array2, tails, type)

Array1 - First data set.

Array2 - Second data set.

Tails - No. of distribution tails.

Type - Kind of t-Test to perform.

VAR This function is used to estimate variance based on a sample.

Syntax: VAR (number1, number2,...)

Number1, number2, ... are 1 to 30 number arguments corresponding to a sample of a population.

VARA This function is used to estimate variance based on a sample. In addition to numbers, text and logical values such as TRUE and FALSE are included in the calculation.

Syntax: VARA (value1, value2,...)

Value1, value2, ... are 1 to 30 value arguments corresponding to a sample of a population.

VARP This function is used to calculate variance based on the entire population.

Syntax: VARP (number1, number2,...)

Number1, number2, ... are 1 to 30 number arguments corresponding to a population.

VARPA This function is used to calculate variance based on the entire population. In addition to numbers, text and logical values such as TRUE and FALSE are included in the calculation.

Syntax: VARPA (value1, value2,...)

Value1, value2, ... are 1 to 30 value arguments corresponding to a population

WEIBULL[#] This function is used to return the Weibull distribution. Use this distribution in reliability analysis, such as calculating a device's mean time to failure.

Syntax: WEIBULL (x, alpha, beta, cumulative)

Explanations: Same as above.

ZTEST[#] This function is used to return 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.

Syntax: ZTEST (array, x, sigma)

Array - Range of data against which to test x.

X - Value to test.

Sigma - Population (known) standard deviation. If omitted, the sample standard deviation is used.

Text and Time Functions

ASC[#] This function is used to changes full-width (double-byte) English letters or katakana within a character string to half-width (single-byte) characters. For Double-byte character set (DBCS) languages, changes half-width (single-byte) English characters to full-width (double-byte) characters.

Syntax: ASC (text)

Text - Text or a reference to a cell that contains the text you want to change.

BAHTTEXT[#] This function is used to convert a number to Thai text and adds a suffix of "Baht."

Syntax: BAHTTEXT (number)

Number - Number you want to convert to text, or a reference to a cell containing a number, or a formula that evaluates to a number

CHAR This function is used to return 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.

Syntax: CHAR (number)

Number - Number between 1 and 255 specifying which character you want.

CLEAN This function is used to remove all nonprintable characters from text.

Syntax: CLEAN (text)

Text - Worksheet information from which you want to remove nonprintable characters.

CODE This function is used to return a numeric code for the first character in a text string. The returned code corresponds to the character set used by your computer.

Syntax: CODE (text)

Text - Text for which you want the code of the first character.

CONCATENATE This function is used to join several text strings into one text string.

Syntax: CONCATENATE (text1, text2,...)

Text1, text2, ... are 1 to 30 text items to be joined into a single text item.

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

Syntax: DOLLAR (number, decimals)

Number - Number, a reference to a cell containing a number, or a formula that evaluates to a number.

Decimals - Number of digits to the right of the decimal point.

EXACT This function 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.

Syntax: EXACT (text1, text2)

Text1 - First text string.

Text2 - Second text string.

FIND This function finds one text value within another (case-sensitive). Also applies to: FINDB

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.

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.

Syntax: FIND (find_text, within_text, start_num) & FINDB (find_text, within_text, start_num)

Find_text - Text you want to find.

Within_text - Text containing the text you want to find.

Start_num - Character at which to start the search.

FIXED This function is used to round 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.

Syntax: FIXED (number, decimals, no_commas)

Number - Number you want to round and convert to text.

Decimals - No. of digits to the right of the decimal point.

No_commas - Logical value that, if TRUE, prevents FIXED from including commas in the returned text.

JIS[#] This function is used to change half-width (single-byte) English letters or katakana within a character string to full-width (double-byte) characters. For Japanese, this function changes half-width (single-byte) English letters or katakana within a character string to full-width (double-byte) characters.

Syntax: JIS (text)

Text - Text or a reference to a cell that contains the text you want to change.

LEFT This function is used to return the leftmost characters from a text value. Also applies to: LEFTB

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

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.

Syntax: LEFT (text, num_chars) & LEFTB (text, num_bytes)

Text - Text string that contains the characters you want to extract.

Num_chars - No. of characters you want LEFT to extract.

Num_bytes - No. of characters you want LEFTB to extract, based on bytes.

LEN This function is used to return the number of characters in a text string. Also applies to: LENB

LEN returns the number of characters in a text string.

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

Syntax: LEN (text) & LENB (text)

Text - Text whose length you want to find. Spaces count as characters.

LOWER This function is used to convert all uppercase letters in a text string to lowercase.

Syntax: LOWER (text)

Text - Text you want to convert to lowercase.

MID This function is used to return a specific number of characters from a text string starting at the position you specify. Also applies to: MIDB

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 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.

Syntax: MID (text, start_num, num_chars) & MIDB (text, start_num, num_bytes)

Explanations: Same as above

Text - Text string containing the characters you want to extract.

Start_num - Position of the first character you want to extract in text.

PHONETIC[#] This function is used to extract the phonetic (furigana) characters from a text string.

Syntax: PHONETIC (reference)

Reference - Text string/Reference to a single cell or range of cells that contain a furigana text string.

PROPER This function is used to capitalize 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.

Syntax: PROPER (text)

Text - Text enclosed in quotation marks, a formula that returns text, or a reference to a cell containing the text you want to partially capitalize

REPLACE This function is used to replace characters within text. Also applies to: REPLACEB

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

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.

Syntax: REPLACE (old_text, start_num, num_chars, new_text) &

REPLACEB (old_text, start_num, num_bytes, new_text)

Explanations: Same as above

Old_text - Text in which you want to replace some characters.

New_text - Text that will replace characters in old_text.

REPT This function is used to repeat text a given number of times.

Syntax: REPT (text, number_times)

Text - Text you want to repeat.

Number_times - Positive number specifying no. of times to repeat text.

RIGHT This function is used to return the rightmost characters from a text value. Also applies to: RIGHTB

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

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.

Syntax: RIGHT (text, num_chars) & RIGHTB (text, num_bytes)

Explanations: Same as above

SEARCH This function is used to find one text value within another. Also applies to: SEARCHB

SEARCH returns the number of the character at which a specific character or text string is first found, beginning with 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.

Syntax: SEARCH (find_text, within_text, start_num) & SEARCHB (find_text, within_text, start_num)

Find_text - Text you want to find.

Within_text - Text in which you want to search for find_text.

Start_num - Character number in within_text at which you want to start searching.

SUBSTITUTE This function is used to substitute new text for old text in a text string. Use

SUBSTITUTE when you want to replace specific text in a text string.

Syntax: SUBSTITUTE (text, old_text, new_text, instance_num)

Explanations: Same as above

Instance_num - Occurrence of old text you want to replace with new text.

T This function is used to return the text referred to by value.

Syntax: T (value)

Value is the value you want to test.

TEXT This function is used to convert a value to text in a specific number format.

Syntax: TEXT (value, format_text)

Value is a numeric value, a formula that evaluates to a numeric value, or a reference to a cell containing a numeric value.

Format_text is a number format in text form.

TRIM This function is used to remove 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.

Syntax: TRIM (text)

Text is the text from which you want spaces removed.

UPPER This function is used to convert text to uppercase.

Syntax: UPPER (text)

Text is the text you want converted to uppercase. Text can be a reference or text string.

VALUE This function is used to convert a text string that represents a number to a number.

Syntax: VALUE (text)

Text is the text enclosed in quotation marks or a reference to a cell containing the text you want to convert.

Maths and Trigonometry

ABS This function is used to return the absolute value of a number. The absolute value of a number is the number without its sign.

Syntax: ABS (number)

Number is the real number of which you want the absolute value.

ACOS This function is used to return the arccosine, or inverse cosine, 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.

Syntax: ACOS (number)

Number is the cosine of the angle you want and must be from -1 to 1.

ACOSH This function is used to return 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.

Syntax: ACOSH (number)

Number is any real number equal to or greater than 1.

ASIN This function is used to return the arcsine, or inverse sine, 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.

Syntax: ASIN (number)

Number is the sine of the angle you want and must be from -1 to 1.

ASINH This function is used to return the inverse hyperbolic sine of a number. The inverse hyperbolic sine is the value whose hyperbolic sine is number, so ASINH(SINH(number)) = number.

Syntax: ASINH (number)

Number is any real number.

ATAN This function is used to return the arctangent, or inverse tangent, of a number. The arctangent is the angle whose tangent is number. Returned angle in radians in range -pi/2 to pi/2.

Syntax: ATAN (number)

Number is the tangent of the angle you want.

ATAN2# This function is used to return the arctangent, or inverse tangent, of the specified x- and y-coordinates. The arctangent is the angle from 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.

Syntax: ATAN2 (x_num, y_num)

X_num - x-coordinate of the point.

Y_num - y-coordinate of the point.

ATANH This function is used to return 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.

Syntax: ATANH (number)

Number is any real number between 1 and -1.

CEILING This function is used to return number rounded up, away from zero, to 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.

Syntax: CEILING (number, significance)

Number - Value you want to round.

Significance - Multiple to which you want to round.

COMBIN This function is used to return 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.

Syntax: COMBIN (number, number_chosen)

Number - No. of items.

Number chosen - No. of items in each combination.

COS This function is used to return the cosine of the given angle.

Syntax: COS (number)

Number is the angle in radians for which you want the cosine.

COSH This function is used to return the hyperbolic cosine of a number.

Syntax: COSH (number)

Number is any real number for which you want to find the hyperbolic cosine.

COUNTIF This function is used to count the no. of cells within a range that meet the given criteria.

Syntax: COUNTIF (range, criteria)

Range - Range of cells from which you want to count cells.
Criteria - Criteria in form of a number, expression, or text that defines which cells will be counted.

DEGREES This function is used to convert radians into degrees.

Syntax: DEGREES (angle)

Angle is the angle in radians that you want to convert.

EVEN This function is used to return number rounded up to the nearest even integer. You can use this function for processing items that come in twos.

Syntax: EVEN (number)

Number is the value to round.

EXP This function is used to return e raised to the power of number. The constant e equals 2.71828182845904, the base of the natural logarithm.

Syntax: EXP (number)

Number is the exponent applied to the base e.

FACT This function is used to return factorial of a number. Factorial = $1*2*3*...*$ number.

Syntax: FACT (number)

Number is nonnegative number you want factorial of. If number is not an integer, it is truncated.

FACTDOUBLE This function is used to return the double factorial of a number.

Syntax: FACTDOUBLE (number)

Number is value for which to return double factorial. If number is not an integer, it is truncated.

FLOOR This function is used to round number down, toward zero, to nearest multiple of significance.

Syntax: FLOOR (number, significance)

Number - Numeric value you want to round.

Significance - Multiple to which you want to round.

GCD This function is used to return greatest common divisor of two or more integers. The greatest common divisor is largest integer that divides both number1 & number2 without a remainder.

Syntax: GCD (number1, number2, ...)

Number1, number2, ... are 1 to 29 values. If any value is not an integer, it is truncated.

INT This function is used to round a number down to the nearest integer.

Syntax: INT (number)

Number is the real number you want to round down to an integer.

LCM This function is used to 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.

Syntax: LCM (number1, number2, ...)

Number1, number2,... are 1 to 29 values for which you want the least common multiple. If value is not an integer, it is truncated.

LN This function is used to return the natural logarithm of a number. Natural logarithms are based on the constant e (2.71828182845904).

Syntax: LN (number)

Number is the positive real number for which you want the natural logarithm.

LOG This function is used to return the logarithm of a number to the base you specify.

Syntax: LOG (number, base)

Number is the positive real number for which you want the logarithm.

Base is the base of the logarithm. If base is omitted, it is assumed to be 10

LOG10 This function is used to return the base-10 logarithm of a number.

Syntax: LOG10 (number)

Number is the positive real number for which you want the base-10 logarithm.

MDETERM This function is used to return the matrix determinant of an array.

Syntax: MDETERM (array)

Array is a numeric array with an equal number of rows and columns.

MINVERSE This function is used to return the inverse matrix for the matrix stored in an array.

Syntax: MINVERSE (array)

Array is a numeric array with an equal number of rows and columns.

MMULT[#] This function is used to return 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.

Syntax: MMULT (array1, array2)

Array1, array2 are the arrays you want to multiply.

MOD This function is used to return the remainder after number is divided by divisor. The result has the same sign as divisor.

Syntax: MOD (number, divisor)

Number is the number for which you want to find the remainder.

Divisor is the number by which you want to divide number.

MROUND This function is used to return a number rounded to the desired multiple.

Syntax: MROUND (number, multiple)

Number - Value to round.

Multiple - Multiple to which you want to round number

MULTINOMIAL[#] This function is used to return the ratio of the factorial of a sum of values to the product of factorials.

Syntax: MULTINOMIAL (number1, number2, ...)

Number1,number2, ... are 1 to 29 values for which you want the multinomial.

ODD This function is used to return number rounded up to the nearest odd integer.

Syntax: ODD (number)

Number is the value to round.

PI[#] This function returns number 3.14159265358979, math. constant pi, accurate to 15 digits.

Syntax: PI ()

POWER This function is used to return the result of a number raised to a power.

Syntax: POWER (number, power)

Number is the base number. It can be any real number.

Power is the exponent to which the base number is raised.

PRODUCT This function multiplies all the numbers given as arguments and returns the product.

Syntax: PRODUCT (number1, number2,...)

Number1, number2, ... are 1 to 30 numbers that you want to multiply.

QUOTIENT This function is used to return the integer portion of a division. Use this function when you want to discard the remainder of a division.

Syntax: QUOTIENT (numerator, denominator)

Numerator - Dividend.

Denominator - Divisor.

RADIANS This function is used to convert degrees to radians.

Syntax: RADIANS (angle)

Angle is an angle in degrees that you want to convert.

RAND This function is used to return 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.

Syntax: RAND ()

RANDBETWEEN[#] This function is used to return a random number between the numbers you specify. A new random number is returned every time the worksheet is calculated.

Syntax: RANDBETWEEN (bottom, top)

Bottom - Smallest integer RANDBETWEEN will return.

Top - Largest integer RANDBETWEEN will return.

ROMAN This function is used to convert an arabic numeral to roman, as text.

Syntax: ROMAN (number, form)

Number - Arabic numeral you want converted.

Form - Number specifying the type of roman numeral you want.

ROUND This function is used to round a number to a specified number of digits.

Syntax: ROUND (number, num_digits)

Number - Number you want to round.

Num_digits - No. of digits to which you want to round number.

ROUNDDOWN This function is used to round a number down, toward zero.

Syntax: ROUNDDOWN (number, num_digits)

Explanations: Same as above.

ROUNDUP This function is used to round a number up, away from zero

Syntax: ROUNDUP (number, num_digits)

Explanations: Same as above.

SERIESSUM[#] This function is used to return the sum of a power series based on the formula:

Many functions can be approximated by a power series expansion.

Syntax: SERIESSUM (x, n, m, coefficients)

X - Input value to the power series.

N - Initial power to which you want to raise x.

M - Step by which to increase n for each term in the series.

Coefficients - Set of coefficients by which each successive power of x is multiplied.

SIGN This function is used to determine 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.

Syntax: SIGN (number)

Number is any real number.

SIN This function is used to return the sine of the given angle.

Syntax: SIN (number)

Number is the angle in radians for which you want the sine.

SINH This function is used to return the hyperbolic sine of a number.

Syntax: SINH (number)

Number - Any real number.

SQRT This function is used to return a positive square root.

Syntax: SQRT (number)

Number is the number for which you want the square root.

SQRTPI[#] This function is used to return the square root of (number * pi).

Syntax: SQRTPI (number)

Number is the number by which pi is multiplied.

SUBTOTAL This function is used to return a subtotal in a list or database.

Syntax: SUBTOTAL (function_num, ref1, ref2,...)

Function_num is the number 1 to 11 that specifies which function to use in calculating subtotals within a list.

SUM This function is used to add all the numbers in a range of cells.

Syntax: SUM (number1, number2, ...)

Number1, number2, ... are 1 to 30 arguments for which you want the total value or sum.

SUMIF This function is used to add the cells specified by a given criteria.

Syntax: SUMIF (range, criteria, sum_range)

Range - Range of cells you want evaluated.

Criteria - Criteria in the form of a number, expression, or text that defines which cells will be added.

Sum_range - Actual cells to sum.

SUMPRODUCT[#] This function is used to multiply corresponding components in the given arrays, and returns the sum of those products.

Syntax: SUMPRODUCT (array1, array2, array3, ...)

Array1, array2, array3, ... are 2 to 30 arrays whose components you want to multiply and then add.

SUMSQ This function is used to return the sum of the squares of the arguments.

Syntax: SUMSQ (number1, number2, ...)

Number1, number2, ... are 1 to 30 arguments for which you want the sum of the squares. You can also use a single array or a reference to an array instead of arguments separated by commas.

SUMX2MY2[#] This function is used to return the sum of the difference of squares of corresponding values in two arrays.

Syntax: SUMX2MY2 (array_x, array_y)

Array_x - First array or range of values.

Array_y - Second array or range of values.

SUMX2PY2[#] This function is used to return 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.

Syntax: SUMX2PY2 (array_x, array_y)

Array_x - First array or range of values.

Array_y - Second array or range of values.

SUMXMY2[#] This function returns sum of squares of differences of corresponding values in 2 arrays.

Syntax: SUMXMY2 (array_x, array_y)

Array_x - First array or range of values.

Array_y - Second array or range of values.

TAN This function is used to return the tangent of the given angle.

Syntax: TAN (number)

Number is the angle in radians for which you want the tangent.

TANH This function is used to return the hyperbolic tangent of a number.

Syntax: TANH (number)

Number is any real number

TRUNC This function is used to truncate a number to an integer by removing fractional part.

Syntax: TRUNC (number, num_digits)

Number is the number you want to truncate.

Num_digits is a number specifying precision of truncation. The default value for num_digits is 0.

Lookup & Reference

ADDRESS This function creates a cell address as text, given specified row and column numbers.

Syntax: ADDRESS (row_num, column_num, abs_num, a1, sheet_text)

Row_num - Row number to use in the cell reference.

Column_num - Column number to use in the cell reference.

Abs_num - Type of reference to return.

AREAS[#] This function returns no. of areas in reference. An area is a range of contiguous/single cell.

Syntax: AREAS (reference)

Reference is a reference to a cell or range of cells and can refer to multiple areas.

CHOOSE[#] This function is used 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.

Syntax: CHOOSE (index_num, value1, value2,...)

Index_num specifies which value argument is selected. Index_num must be a number between 1 and 29, or a formula or reference to a cell containing a number between 1 and 29.

Value1,value2,... are 1 to 29 value arguments from which CHOOSE selects a value or an action to perform based on index_num. The arguments can be numbers, cell references, defined names, formulas, functions, or text.

COLUMN This function is used to return the column number of the given reference.

Syntax: COLUMN (reference)

Reference is the cell or range of cells for which you want the column number.

COLUMNS This function is used to return the number of columns in an array or reference.

Syntax: COLUMNS (array)

Array is an array/array formula, or a reference to range of cells for which you want no. of columns.

HLOOKUP[#] This function is used to search 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.

Syntax: HLOOKUP (lookup_value, table_array, row_index_num, range_lookup)

Lookup_value is the value to be found in the first row of the table. Lookup_value can be a value, a reference, or a text string.

Table_array is a table of information in which data is looked up. Use a reference to a range or a range name.

Row_index_num is the row number in table_array from which the matching value will be returned. A row_index_num of 1 returns the first row value in table_array, a row_index_num of 2 returns the second row value in table_array, and so on.

Range_lookup is a logical value that specifies whether you want HLOOKUP to find an exact match or an approximate match. If TRUE or omitted, an approximate match is returned. In other words, if an exact match is not found, the next largest value that is less than lookup_value is returned.

HYPERLINK[#] This function is used to create 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.

Syntax: HYPERLINK (link_location, friendly_name)

Link_location is the path and file name to the document to be opened as text.

Friendly_name is jump text or numeric value that is displayed in cell. Friendly_name is displayed in blue and is underlined. If friendly_name is omitted, cell displays link_location as the jump text.

INDEX[#] This function is used to return a value or the reference to a value from within a table or range. There are two forms of the INDEX() function: array and reference. The array form always returns a value or an array of values; the reference form always returns a reference.

Syntax 1: INDEX (array, row_num, column_num)

Array is a range of cells or an array constant.

Row_num selects the row in array from which to return a value. If row_num is omitted, column_num is required.

Column_num selects the column in array from which to return a value. If column_num is omitted, row_num is required

Syntax 2: INDEX (reference, row_num, column_num, area_num)

Reference is a reference to one or more cell ranges.

Row_num is the number of the row in reference from which to return a reference.

Column_num is the number of the column in reference from which to return a reference.

INDIRECT[#] This function is used to return the reference specified by a text string. References are immediately evaluated to display their contents.

Syntax: INDIRECT (ref_text, a1)

Ref_text is a reference to a cell that contains an A1-style reference, an R1C1-style reference, a name defined as a reference, or a reference to a cell as a text string. If ref_text is not a valid cell reference, INDIRECT returns the #REF! error value.

A1 is a logical value that specifies what type of reference is contained in the cell ref_text.

LOOKUP[#] This function is used to return a value either from a one-row or one-column range or from an array. The LOOKUP function has two syntax forms: vector and array. The 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. 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.

Syntax 1: LOOKUP (lookup_value, lookup_vector, result_vector)

Lookup_value is a value that LOOKUP searches for in the first vector. Lookup_value can be a number, text, a logical value, or a name or reference that refers to a value.

Lookup_vector is a range that contains only one row or one column. The values in lookup_vector can be text, numbers, or logical values.

Important The values in lookup_vector must be placed in ascending order: ..., -2, -1, 0, 1, 2, ..., A-Z, FALSE, TRUE; otherwise, LOOKUP may not give the correct value. Uppercase and lowercase text are equivalent.

Result_vector is a range that contains only one row or column. It must be the same size as lookup_vector.

Syntax 2: LOOKUP (lookup_value, array)

Lookup_value is a value that LOOKUP searches for in an array. Lookup_value can be a number, text, a logical value, or a name or reference that refers to a value.

Array is a range of cells that contains text, numbers, or logical values that you want to compare with lookup_value. The array form of LOOKUP is very similar to the HLOOKUP and VLOOKUP functions. The difference is that HLOOKUP searches for lookup_value in the first row, VLOOKUP searches in the first column, and LOOKUP searches according to the dimensions of array.

Important The values in array must be placed in ascending order: ..., -2, -1, 0, 1, 2, ..., A-Z, FALSE, TRUE; otherwise, LOOKUP may not give the correct value. Uppercase and lowercase text are equivalent

MATCH[#] This function is used to return 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.

Syntax: MATCH (lookup_value, lookup_array, match_type)

Lookup_value is the value you use to find the value you want in a table.

Lookup_array is a contiguous range of cells containing possible lookup values. Lookup_array must be an array or an array reference.

Match_type is the number -1, 0, or 1. Match_type specifies how Microsoft Excel matches lookup_value with values in lookup_array.

OFFSET[#] This function is used to return 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.

Syntax: OFFSET (reference, rows, cols, height, width)

Reference is the reference from which you want to base the offset. Reference must refer to a cell or range of adjacent cells; otherwise, OFFSET returns the #VALUE! error value.

Rows is the number of rows, up or down, that you want the upper-left cell to refer to. Using 5 as the rows argument specifies that the upper-left cell in the reference is five rows below reference. Rows can be positive (which means below the starting reference) or negative (which means above the starting reference).

Cols is the number of columns, to the left or right, that you want the upper-left cell of the result to refer to. Using 5 as the cols argument specifies that the upper-left cell in the reference is five columns to the right of reference. Cols can be positive (which means to the right of the starting reference) or negative (which means to the left of the starting reference).

Height is the height, in number of rows, that you want the returned reference to be. Height must be a positive number.

Width is the width, in number of columns, that you want the returned reference to be. Width must be a positive number

ROW This function is used to return the row number of a reference.

Syntax: ROW (reference)

Reference is the cell or range of cells for which you want the row number.

ROWS This function is used to return the number of rows in a reference or array.

Syntax: ROWS (array)

Array is an array/array formula, or a reference to a range of cells for which you want no. of rows.

RTD[#] This function is used to retrieve real-time data from a program that supports COM automation.

Syntax: RTD (ProgID, server, topic1, [topic2],...)

ProgID the name of the ProgID of a registered COM automation add-in that has been installed on the local computer. Enclose the name in quotation marks.

server name of the server where the add-in should be run. If there is no server, and the program is run locally, leave the argument blank. Otherwise, enter quotation marks (") around the server name. When using RTD within Visual Basic for Applications (VBA), double quotation marks or the VBA Null String property are required for the server, even if the server is running locally.

topic1, topic2,... 1 to 28 parameters that together represent a unique piece of real-time data.

TRANSPOSE[#] This function returns a vertical range of cells as horizontal range, or vice versa.

Syntax: TRANSPOSE (array)

Array - Array or range of cells on a worksheet that you want to transpose.

VLOOKUP[#] This function is used to search 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.

The V in VLOOKUP stands for "Vertical."

Syntax: VLOOKUP (lookup_value, table_array, col_index_num, range_lookup)

Lookup_value is the value to be found in the first column of the array. Lookup_value can be a value, a reference, or a text string.

Table_array is the table of information in which data is looked up. Use a reference to a range or a range name, such as Database or List.

Col_index_num is the column number in table_array from which the matching value must be returned. A col_index_num of 1 returns the value in the first column in table_array; a col_index_num of 2 returns the value in the second column in table_array, and so on. If col_index_num is less than 1, VLOOKUP returns the #VALUE! error value; if col_index_num is greater than the number of columns in table_array, VLOOKUP returns the #REF! error value.

Range_lookup is a logical value that specifies whether you want VLOOKUP to find an exact match or an approximate match. If TRUE or omitted, an approximate match is returned. In other words, if an exact match is not found, the next largest value that is less than lookup_value is returned. If FALSE, VLOOKUP will find an exact match. If one is not found, the error value #N/A is returned.

Information Functions

CELL This function is used to return information about the formatting, location, or contents of the upper-left cell in a reference.

Syntax: CELL (info_type, reference)

Info_type - Text value that specifies what type of cell information you want.

Reference - Cell that you want information about.

COUNTBLANK This function is used to count empty cells in a specified range of cells.

Syntax: COUNTBLANK (range)

Range is the range from which you want to count the blank cells.

ERROR.TYPE[#] This function is used to return a number corresponding to one of the error values in Microsoft Excel or returns the #N/A error if no error exists.

Syntax: ERROR.TYPE (error_val)

Error_val is the error value whose identifying number you want to find. Although error_val can be the actual error value, it will usually be a reference to cell containing formula that you want to test.

INFO[#] This function is used to return information about the current operating environment.

Syntax: INFO (type_text)

Type_text is text that specifies what type of information you want returned.

ISBLANK[#] This section describes the nine worksheet functions used for testing the type of a value or reference. Each of these functions, referred to collectively as the IS functions, checks the type of

value and returns TRUE or FALSE depending on the outcome.

Syntax: ISBLANK (value), ISERR (value), ISERROR (value), ISLOGICAL (value), ISNA (value),

ISNONTEXT (value), ISNUMBER (value), ISREF (value), ISTEXT (value)

Value is the value you want tested. Value can be a blank (empty cell), error, logical, text, number, or reference value, or a name referring to any of these, that you want to test.

ISERR[#] This section describes the nine worksheet functions used for testing the type of a value or reference. Each of these functions, referred to collectively as the IS functions, checks the type of value and returns TRUE or FALSE depending on the outcome.

Syntax: ISBLANK (value), ISERR (value), ISERROR (value), ISLOGICAL (value), ISNA (value),

ISNONTEXT (value), ISNUMBER (value), ISREF (value), ISTEXT (value)

Value is the value you want tested. Value can be a blank (empty cell), error, logical, text, number, or reference value, or a name referring to any of these, that you want to test.

ISERROR[#] This section describes the nine worksheet functions used for testing the type of a value or reference. Each of these functions, referred to collectively as the IS functions, checks the type of value and returns TRUE or FALSE depending on the outcome.

Syntax: ISBLANK (value), ISERR (value), ISERROR (value), ISLOGICAL (value), ISNA (value),

ISNONTEXT (value), ISNUMBER (value), ISREF (value), ISTEXT (value)

Value is the value you want tested. Value can be a blank (empty cell), error, logical, text, number, or reference value, or a name referring to any of these, that you want to test

ISEVEN This function returns TRUE if number is even, or FALSE if number is odd.

Syntax: ISEVEN (number)

Number is the value to test. If number is not an integer, it is truncated.

ISLOGICAL[#] This section describes the nine worksheet functions used for testing the type of a value or reference. Each of these functions, referred to collectively as the IS functions, checks the type of value and returns TRUE or FALSE depending on the outcome.

Syntax: ISBLANK (value), ISERR (value), ISERROR (value), ISLOGICAL (value), ISNA (value),

ISNONTEXT (value), ISNUMBER (value), ISREF (value), ISTEXT (value)

Value is the value you want tested. Value can be a blank (empty cell), error, logical, text, number, or reference value, or a name referring to any of these, that you want to test.

ISNA[#] This section describes the nine worksheet functions used for testing the type of a value or reference. Each of these functions, referred to collectively as the IS functions, checks the type of value and returns TRUE or FALSE depending on the outcome.

Syntax: ISBLANK (value), ISERR (value), ISERROR (value), ISLOGICAL (value), ISNA (value), ISNONTEXT (value), ISNUMBER (value), ISREF (value), ISTEXT (value)

Value is the value you want tested. Value can be a blank (empty cell), error, logical, text, number, or reference value, or a name referring to any of these, that you want to test.

ISNONTEXT[#] This section describes the nine worksheet functions used for testing the type of a value or reference. Each of these functions, referred to collectively as the IS functions, checks the type of value and returns TRUE or FALSE depending on the outcome.

Syntax: ISBLANK (value), ISERR (value), ISERROR (value), ISLOGICAL (value), ISNA (value), ISNONTEXT (value), ISNUMBER (value), ISREF (value), ISTEXT (value)

Value is the value you want tested. Value can be a blank (empty cell), error, logical, text, number, or reference value, or a name referring to any of these, that you want to test.

ISNUMBER[#] This section describes the nine worksheet functions used for testing the type of a value or reference. Each of these functions, referred to collectively as the IS functions, checks the type of value and returns TRUE or FALSE depending on the outcome.

Syntax: ISBLANK (value), ISERR (value), ISERROR (value), ISLOGICAL (value), ISNA (value), ISNONTEXT (value), ISNUMBER (value), ISREF (value), ISTEXT (value)

Value is the value you want tested. Value can be a blank (empty cell), error, logical, text, number, or reference value, or a name referring to any of these, that you want to test.

ISODD This function returns TRUE if number is odd, or FALSE if number is even.

Syntax: ISODD (number)

Number is the value to test. If number is not an integer, it is truncated.

ISREF[#] This section describes the nine worksheet functions used for testing the type of a value or reference. Each of these functions, referred to collectively as the IS functions, checks the type of value and returns TRUE or FALSE depending on the outcome.

Syntax: ISBLANK (value), ISERR (value), ISERROR (value), ISLOGICAL (value), ISNA (value),

ISNONTEXT (value), ISNUMBER (value), ISREF (value), ISTEXT (value)

Value is the value you want tested. Value can be a blank (empty cell), error, logical, text, number, or reference value, or a name referring to any of these, that you want to test.

ISTEXT[#] This section describes the nine worksheet functions used for testing the type of a value or reference. Each of these functions, referred to collectively as the IS functions, checks the type of value and returns TRUE or FALSE depending on the outcome.

Syntax: ISBLANK (value), ISERR (value), ISERROR (value), ISLOGICAL (value), ISNA (value), ISNONTEXT (value), ISNUMBER (value), ISREF (value), ISTEXT (value)

Value is the value you want tested. Value can be a blank (empty cell), error, logical, text, number, or reference value, or a name referring to any of these, that you want to test.

N This function returns a value converted to a number

Syntax: N (value)

Value is the value you want converted. N converts values listed in the following table.

NA This function is used to return 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.)

Syntax: NA ()

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

Syntax: TYPE (value)

Value can be any Microsoft Excel value, such as a number, text, logical value, and so on.

Logical Functions

AND This function returns TRUE if all its arguments are TRUE; returns FALSE if one or more argument is FALSE.

Syntax: AND (logical1, logical2, ...)

Logical1, logical2, ... are 1 to 30 conditions you want to test that can be either TRUE or FALSE.

FALSE This function returns the logical value FALSE.

Syntax: FALSE ()

IF[#] This function returns one value if a condition you specify evaluates to TRUE and another value if it evaluates to FALSE.

Syntax: IF (logical_test, value_if_true, value_if_false)

Logical_test is any value or expression that can be evaluated to TRUE or FALSE. For example, A10=100 is a logical expression; if the value in cell A10 is equal to 100, the expression evaluates to TRUE. Otherwise, the expression evaluates to FALSE.

Value_if_true is the value that is returned if logical_test is TRUE.

Value_if_true can be another formula.

Value_if_false is the value that is returned if logical_test is FALSE.

NOT[#] This function reverses the value of its argument. Use NOT when you want to make sure a value is not equal to one particular value.

Syntax: NOT (logical)

Logical is a value or expression that can be evaluated to TRUE or FALSE.

OR[#] This function returns TRUE if any arg. is TRUE; returns FALSE if all arguments are FALSE.

Syntax: OR (logical1, logical2,...)

Logical1, logical2,... are 1 to 30 conditions you want to test that can be either TRUE or FALSE

TRUE This function returns the logical value TRUE.

Syntax: TRUE ()

Database Functions

DAVERAGE This function is used to average the values in a column of a list or database that match conditions you specify.

Syntax: DAVERAGE (database, field, criteria)

Database is the range of cells that makes up the list or database. A database is a list of related data in which rows of related information are records, and columns of data are fields. The first row of the list contains labels for each column.

Field indicates which column is used in the function. Field can be given as text with the column label enclosed between double quotation marks, such as "Age" or "Yield," or as a number that represents the position of the column within the list: 1 for the first column, 2 for the second column, and so on.

Criteria is the range of cells that contains the conditions you specify. You can use any range for the criteria argument, as long as it includes at least one column label and at least one cell below the column label for specifying a condition for the column.

DCOUNT This function is used to count the cells that contain numbers in a column of a list or database that match conditions you specify.

The field argument is optional. If field is omitted, DCOUNT counts all records in the database that match the criteria.

Syntax: DCOUNT (database, field, criteria)

Explanations: Same as above

DCOUNTA[#] This function is used to count nonblank cells in a column of a list or database that match conditions you specify.

The field argument is optional. If field is omitted, DCOUNTA counts all records in the database that match the criteria.

Syntax: DCOUNTA (database, field, criteria)

Explanations: Same as above

DGET[#] This function is used to extract a single value from a column of a list or database that matches conditions you specify.

Syntax: DGET (database, field, criteria)

Explanations: Same as above

DMAX This function is used to return the largest number in a column of a list or database that matches conditions you specify.

Syntax: DMAX (database, field, criteria)

Explanations: Same as above

DMIN This function is used to return the smallest number in a column of a list or database that matches conditions you specify.

Syntax: DMIN (database, field, criteria)

Explanations: Same as above

DPRODUCT This function is used to multiply the values in a column of a list or database that match conditions you specify.

Syntax: DPRODUCT (database, field, criteria)

Explanations: Same as above

DSTDEV[#] This function is used to estimate the standard deviation of a population based on a sample by using the numbers in a column of a list or database that match conditions you specify.

Syntax: DSTDEV (database, field, criteria)

Explanations: Same as above

DSTDEVP[#] This function is used to calculate the standard deviation of a population based on the entire population, using numbers in a column of a list/database that match conditions you specify.

Syntax: DSTDEVP (database, field, criteria)

Explanations: Same as above

DSUM This function is used to add numbers in a column of a list/database that match conditions.

Syntax: DSUM (database, field, criteria)

Explanations: Same as above

DVAR This function is used to estimate the variance of a population based on a sample by using the numbers in a column of a list or database that match conditions you specify.

Syntax: DVAR (database, field, criteria)

Explanations: Same as above

DVARP[#] This function is used to calculate variance of a population based on the entire population by using the numbers in a column of a list or database that match conditions you specify.

Syntax: DVARP (database, field, criteria)

Explanations: Same as above

GETPIVOTDATA[#] This function is used to return 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.

Syntax: GETPIVOTDATA (data_field, pivot_table, field1, item1, field2, item2,...)

Data_field is the name, enclosed in quotation marks, for the data field that contains the data you want to retrieve. Pivot_table is a reference to any cell, range of cells, or named range of cells in a PivotTable report. This information is used to determine which PivotTable report contains the data you want to retrieve.

Field1, Item1, Field2, Item2 are one to 14 pairs of field names and item names that describe the data you want to retrieve. The pairs can be in any order. Field names and names for items other than dates and numbers are enclosed in quotation marks. For OLAP PivotTable reports, items can contain the source name of the dimension as well as the source name of the item. A field and item pair for an OLAP PivotTable might look like this:

"[Product]", "[Product].[All Products].[Foods].[Baked Goods]"

Date & Time Functions

DATE This function is used to return the sequential serial number that represents a particular date. If the cell format was General before the function was entered, result is formatted as a date.

Syntax: DATE (year, month, day)

Month is a number representing the month of the year. If month is greater than 12, month adds that number of months to the first month in the year specified.

Day is a number representing the day of the month. If day is greater than the number of days in the month specified, day adds that number of days to the first day in the month.

DATEVALUE This function is used to return the serial number of the date represented by date_text. Use DATEVALUE to convert a date represented by text to a serial number.

Syntax: DATEVALUE (date_text)

Date_text is text that represents a date in a Microsoft Excel date format.

If the year portion of date_text is omitted, DATEVALUE uses the current year from your computer's built-in clock. Time information in date_text is ignored.

DAY This function is used to return the day of a date, represented by a serial number. The day is given as an integer ranging from 1 to 31.

Syntax: DAY (serial_number)

Serial_number is the date of the day you are trying to find. Dates should be entered by using DATE function, or as results of other formulas/functions. Problems can occur if dates entered as text.

DAYS360 This function is used to return 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.

Syntax: DAYS360 (start_date, end_date, method)

Start_date and end_date are the two dates between which you want to know the number of days. If start_date occurs after end_date, DAYS360 returns a negative number. Dates should be entered by using DATE function, or as results of other formulas/functions. Problems can occur if dates are entered as text.

Method is a logical value that specifies whether to use the U.S. or European method in the calculation.

EDATE[#] This function is used to return 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.

Syntax: EDATE (start_date, months)

Start_date is a date that represents the start date. Dates should be entered by using the DATE function, or as results of other formulas/functions. Problems can occur if dates are entered as text.

Months is the number of months before or after start_date. A positive value for months yields a future date; a negative value yields a past date.

EOMONTH[#] This function is used to return 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.

Syntax: EOMONTH (start_date, months)

Start_date is a date that represents the starting date. Dates should be entered by using the DATE function, or as results of other formulas/functions. Problems can occur if dates are entered as text.

Months is the number of months before or after start_date. A positive value for months yields a future date; a negative value yields a past date. If months is not an integer, it is truncated.

HOUR This function is used to return 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.).

Syntax: HOUR (serial_number)

Serial_number is the time that contains the hour you want to find. Times may be entered as text strings within quotation marks (for example, "6:45 PM"), as decimal numbers (for example, 0.78125, represents 6:45 PM), or as results of other formulas or functions (for ex., TIMEVALUE ("6:45 PM")).

MINUTE This function is used to return the minutes of a time value. The minute is given as an integer, ranging from 0 to 59.

Syntax: MINUTE (serial_number)

Serial_number is the time that contains the minute you want to find. Times may be entered as text strings within quotation marks (for example, "6:45 PM"), as decimal numbers (for example, 0.78125, represents 6:45 PM), or as results of other formulas or functions (for ex., TIMEVALUE ("6:45 PM")).

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

Syntax: MONTH (serial_number)

Serial_number is the date of the month you are trying to find. Dates should be entered by using the DATE function, or as results of other formulas or functions.

NETWORKDAYS This function is used to return the no. 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.

Syntax: NETWORKDAYS (start_date, end_date, holidays)

Important Dates should be entered by using DATE function/results of other formulas functions.

Start_date is a date that represents the start date.

End_date is a date that represents the end date.

Holidays is an optional range of one or more dates to exclude from the working calendar, such as state and federal holidays and floating holidays. The list can be either a range of cells that contains the dates or an array constant of the serial numbers that represent the dates.

NOW This function is used to return the serial number of the current date and time. If the cell format was General before the function was entered, the result is formatted as a date.

Syntax: NOW ()

SECOND This function is used to return the seconds of a time value. The second is given as an integer in the range 0 (zero) to 59.

Syntax: SECOND (serial_number)

Serial_number is the time that contains the seconds you want to find. Times may be entered as text strings within quotation marks (for example, "6:45 PM"), as decimal numbers (for example, 0.78125, represents 6:45 PM), or as results of other formulas/functions (for ex., TIMEVALUE ("6:45 PM")).

TIME This function is used to return the decimal number for a particular time. If the cell format was General before the function was entered, the result is formatted as a date.

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

Syntax: TIME (hour, minute, second)

Hour is a number from 0 (zero) to 32767 representing the hour. Any value greater than 23 will be divided by 24 and the remainder will be treated as the hour value. For example, TIME(27,0,0) = TIME(3,0,0) = .125 or 3:00 AM.

Minute is a number from 0 to 32767 representing the minute. Any value greater than 59 will be converted to hours and minutes. For example, TIME(0,750,0) = TIME(12,30,0) = .520833 or 12:30 PM.

Second is a number from 0 to 32767 representing the second. Any value greater than 59 will be converted to hours, minutes, and seconds. For example, TIME(0,0,2000) = TIME(0,33,22) = .023148 or 12:33:20 AM

TIMEVALUE This function is used to return 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 AM) to 23:59:59 (11:59:59 P.M.).

Syntax: TIMEVALUE (time_text)

Time_text is a text string that represents a time in any one of the Microsoft Excel time formats; for example, "6:45 PM" and "18:45" text strings within quotation marks that represent time.

TODAY This function is used to return the serial number of the current date. The serial number is the date-time code used by Microsoft Excel for date and time calculations. If the cell format was General before the function was entered, the result is formatted as a date.

Syntax: TODAY ()

WEEKDAY This function is used to return 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.

Syntax: WEEKDAY (serial_number, return_type)

Serial_number is a sequential number that represents the date of the day you are trying to find. Dates should be entered by using DATE function, or as results of other formulas/functions. For ex., use DATE(2008,5,23) for the 23rd day of May, 2008. Problems can occur if dates entered as text.

Return_type is a number that determines the type of return value.

WEEKNUM This function is used to return a number that indicates where the week falls numerically within a year.

Syntax: WEEKNUM (serial_num, return_type)

Serial_num is a date within the week. Dates should be entered by using the DATE function, or as results of other formulas or functions. Problems can occur if dates are entered as text.

Return_type is a number that determines on which day the week begins. The default is 1.

WORKDAY This function is used to return 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.

Syntax: WORKDAY (start_date, days, holidays)

Important Dates should be entered by using the DATE function, or as results of other formulas or functions. Problems can occur if dates are entered as text.

Start_date is a date that represents the start date.

Days are the number of non-weekend and non-holiday days before or after start_date. A positive value for days yields a future date; a negative value yields a past date.

Holidays is an optional list of one or more dates to exclude from the working calendar, such as state and federal holidays and floating holidays. The list can be either a range of cells that contain the dates or an array constant of the serial numbers that represent the dates.

YEAR This function is used to return the year corresponding to a date. The year is returned as an integer in the range 1900-9999.

Syntax: YEAR (serial_number)

Serial_number is the date of the year you want to find. Dates should be entered by using the DATE function, or as results of other formulas/functions. Problems can occur if dates are entered as text.

YEARFRAC[#] This function is used to return calculate 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.

Syntax: YEARFRAC (start_date, end_date, basis)

Start_date is a date that represents the start date.

End_date is a date that represents the end date.

Basis is the type of day count basis to use.

The End