SPIDERFINANCIAL
www.spiderfinancial.com

Phone: 1-888-427-9486
1-312-324-0367
Fax: 1-312-238-9092
info@spiderfinancial.com

NumXL 1.63 (SHAMROCK) Functions

In this page, we maintain a comprehensive list of NumXL time series functions that are available in Microsoft Excel. Each function can be used directly in a worksheet formula, or called from a VBA macro.

Descriptive Stats (24)

Descriptive statistics is the discipline of quantitatively describing the main features of a collection of data. The main aim is to summarize a data set, rather than use the data to learn about the population that the data are thought to represent.

Function	Description
EWMA	Calculates the estimated value of the exponential-weighted volatility (EWV)
XCF	Calculates the cross-correlation function between two time series
EWXCF	Computes the correlation factor using the exponential-weighted correlation function (i.e. using the exponential-weighted covariance (EWCOV) and volatility (EWMA/EWV) method)
ACF	Calculates the sample autocorrelation function (ACF) of a stationary time series
ACFCI	Calculates the confidence interval limits (upper/lower) for the autocorrelation function
PACF	Calculates the sample partial autocorrelation function (PACF)
PACFCI	Returns the confidence interval limits (upper/lower) for the partial autocorrelation function (PACF)
GINI	Returns the sample Gini coefficient
HURST	Calculates the Hurst exponent for time series with more than 128 observations
LRVar	Returns the long-run variance using a Bartlett kernel with window size k
MAD	Returns the sample median of absolute deviation (MAD)
MD	Returns the mean difference of the input data series
RMD	Returns the mean difference of the input data series
SAE	Calculates the sum of absolute errors (SAE) between the forecast and the eventual outcomes
MAE	Calculates the mean absolute error function for the forecast and the eventual outcomes
MAPE	Calculates the mean absolute percentage error (Deviation) function for the forecast and the eventual outcomes
RMSD	Calculates the root mean squared deviations (aka roor mean squared error (RMSE)) function
RMSE	Calculates the root mean squared error (aka root mean squared deviation (RMSD)) function
SSE	Calculates the sum of the squared errors of the prediction function
IQR	Returns the inter-quartile range (IQR), also called the mid-spread or middle fifty.
Quantile	Returns the sample p-quantile of the non-missing observations (i.e. divides the sample data into equal parts determined by the percentage p)
RMS	Returns the sample root mean square (RMS)

GED_XKURT	Calculates the excess kurtosis of the generalized error distribution (GED)
TDIST_XKURT	Calculates the excess kurtosis of the Student's t-Distribution

Histogram (5)

In statistics, a histogram is a graphical representation showing a visual impression of the distribution of data. It is an estimate of the probability distribution of a continuous variable.

Function	Description
HISTBINS	Returns the number of histogram bins using a given heuristic/method
HISTBIN	Returns the upper/lower limit or center value of the k-th histogram bin
NxHistogram	Calculates the histogram or cumulative histogram function for a given bin
EDF	Calculates the density or cumulative probability of the empirical sample distribution (EDF)
KDE	Calculates the kernel density estimation (PDE) of the sample data

Statistical tests (13)

Statistical/Hypothesis testing is a common method of drawing inferences about a population based on statistical evidence from a sample.

Function	Description
TEST_MEAN	Calculates the p-value of the statistical test for the population mean
TEST_STDEV	Calculates the p-value of the statistical test for the population standard deviation
TEST_SKEW	Calculates the p-value of the statistical test for the population skew (i.e. 3rd moment)
TEST_XKURT	Calculates the p-value of the statistical test for the population excess kurtosis (4th moment)
CollinearityTest	Returns the p-value of the multi-collinearity test
ChowTest	Returns the p-value of the regression stability test (i.e. whether the coefficients in two linear regressions on different data sets are equal)
NormalityTest	Returns the p-value of the normality test (i.e. whether a data set is well-modeled by a normal distribution)
WNTest	Computes the p-value of the statistical portmanteau test (i.e. whether any of a group of autocorrelations of a time series is different from zero)
ACFTest	Calculates the p-value of the statistical test for the population autocorrelation function
ARCHTest	Calculates the p-value of the ARCH effect test (i.e. the white-noise test for the squared time series)
XCFTest	Calculates the test stats, p-value or critical value of the correlation significance test.
ADFTest	Returns the test statistics (e.g. p-value, etc.) of the Augmented Dickey-Fuller (ADF) test, which tests for a unit root in the time series sample
JohansenTest	Returns the test statistics (e.g. p-value, etc.) of the Johansen test for cointegration

Data transformation (15)

In statistics, data transformation refers to the application of a deterministic mathematical function to each point in a data set.

Function	Description
DETREND	Detrends a time series using a regression of y against a polynomial time trend of order p
BoxCox	Returns the Box-Cox transformation of the input data point(s)
LAG	Returns an array of cells for the backward shifted, back shifted or lagged time series
DIFF	Returns an array of cells for the differenced time series (i.e. $(1-L^s)^d$)
INTG	Integrates the differenced time series and recovers the original data (inverse of DIFF)
TSADD	Returns an array of cells for the sum of two time series
TSSUB	Returns an array of the difference between two time series
TSSCALE	Returns an array of cells for the scaled time series
REVERSE	Returns the time-reversed order time series (i.e. the first observation is swapped with the last observation, etc.) for both missing and non-missing values.
LOGIT	Computes the logit transformation, including its inverse
PROBIT	Computes the probit transformation, including its inverse
CLOGLOG	Computes the complementary log-log transformation, including its inverse
RMNA	Returns an array of cells of a time series after removing all missing values
SUBNA	Returns an array of cells of a time series after substituting all missing values with the mean/median
NxTranspose	Converts a vertical range of columns into rows range or vice versa

Time series smoothing (6)

The term "smoothing" is often used to refer to techniques that can be applied to time series data in order to produce smoothed (less noisy or slower moving) data for presentation, or to make out-of-sample forecasts.

Function	Description
WMA	Returns the weighted moving (rolling/running) average using the previous m data points
SESMTH	Returns the (Brown's) simple exponential (EMA) smoothing estimate of the value of X at time t+m (based on the raw data up to time t)
DESMTH	Returns the (Holt-Winters's) double exponential smoothing estimate of the value of X at time T+m
LESMTH	Returns the (Brown's) Linear exponential smoothing estimate of the value of X at time T+m (based on the raw data up to time t)
TESMTH	Returns the (Winters's) triple exponential smoothing estimate of the value of X at time T+m
NxTrend	Returns values along a trend curve (e.g. linear, quadratic, exponential, etc.) at time T+m

Spectral Analysis (5)

In statistics, spectral analysis is a procedure that decomposes a time series into a spectrum of cycles of different lengths. Spectral analysis is also known as frequency domain analysis.

Function	Description
DFT	Calculates the discrete fast Fourier transformation for amplitude and phase
IDFT	Calculates the inverse discrete fast Fourier transformation, recovering the time series
NxHP	Calculates the Hodrick-Prescott (HP) filtered data
NxBK	Calculates the Baxter-King (BK) filtered data
NxConv	Calculates the convolution between two time series

Date & Calendar (8)

The Date and Holiday Calendar functionality were introduced starting in version 1.56 (Zebra) to help you identify any holidays, trading days, and weekdays bias effects that often arise in time series analysis.

Function	Description
NxAdjust	Examines whether the given date falls on a weekend or a holiday (i.e. non-working day), and returns the nearest working business day using a Business Day Convention (BDC)
NxEDATE	Returns the serial number of the date after a given period (e.g. 1-week, 3-months, etc.)
NxIsWorkday	Examines a given date for weekends and holidays (non-working days), and returns FALSE if it falls on a non-working day; otherwise it returns TRUE
NxNetWorkdays	Returns the number of whole working days between two dates (inclusive). Working days exclude weekends and any date identified as a holiday
NxNWKDY	Returns the serial number of the n-th weekday in a month. If the target weekday falls on a holiday, users can adjust the day to the next or prior workday
NxWeekday	Returns the day of the week corresponding to a date. The day is given as an integer, ranging from 1 (Sunday) to 7 (Saturday), by default.
NxWKDYOrder	Returns the order of the weekday in the month for a given date. This is the inverse operator of NxNWKDAY
NxWorkday	Returns the serial date number that represents the date that falls after the start date by a given number of working days

Weekend (5)

As of Excel 2007, Microsoft supports different weekend occurrences in the international version of the date functions (e.g. WORKDAY.INTL). The weekend conventions are defined by either a number or a 7-characters long string (code).

Function	Description
NxIsWeekend	Returns TRUE if the reference date falls on a weekend
NxWKNDate	Returns the serial date number that corresponds to the first (last) day in the next (last) weekend
NxWKNDNo	Returns the weekend-number (1-7 or 11-17) for a given country or a weekend code
NxWKNDStr	Returns the seven-character string code for a weekend-number (1-7, 11-17)
NxWKNDUR	Returns the duration (in calendar days) of the (long) weekend that a given day falls on

Holiday (6)

A holiday is a day designated as having special significance for individuals, governments or religious groups. Typically, a holiday does not necessarily exclude doing normal work but for our purposes, NumXL assumes all supported holidays (e.g. National Holidays) exclude normal work.

Function	Description
NxIsHoliday	Returns TRUE if the reference date falls on a holiday
NxHolidays	Returns an array of the matching supported holiday codes
NxFindHLDY	Returns the holiday code that falls on the given date
NxHLDYDate	Returns the date serial number that represents the holiday in the given year
NxHLDYDates	Returns an array of serial date numbers that represent observed holidays between the two given dates
NxHLDYName	Returns the full name of the holiday that corresponds to the given short code

Calendar (6)

For financial time analysis, a calendar is basically a definition of a list of observed holidays and a weekend day’s convention.

Function	Description
NxCalendars	Returns an array of names and codes for the supported calendars
NxCALHolidays	Returns an array of the holidays' names and codes as defined for the given calendar
NxCALName	Returns the calendar name and description, given the calendar's short code (e.g., "US" will return "US Government Holidays")
NxCALWKND	Returns the weekend number of a 7 character string, calendar code, or currency. Each character represents a day of the week, beginning with Monday

ARMA Model (17)

Given a time series of data X_t, the ARMA model is a tool for understanding and, perhaps, predicting future values in this series. The model consists of two parts, an autoregressive (AR) part and a moving average (MA) part. The model is usually then referred to as the ARMA(p,q) model where p is the order of the autoregressive part and q is the order of the moving average part (as defined below).

Function	Description
ARMA	Returns a unique string to designate the specified ARMA model
ARMA_MEAN (*deprecated*)	Returns an array of cells for the fitted values of the conditional mean. ***Note:*** *function is deprecated as of version 1.63: use ARMA_FIT function instead.*
ARMA_VOL (*deprecated*)	Returns an array of cells for the fitted (in-sample) conditional volatility/standard deviation ***Note:*** *function is deprecated as of version 1.63: use ARMA_FIT function*

	instead.
ARMA_GUESS **(*deprecated*)**	Returns an array of cells for the initial/quick guess of the model's parameters ***Note:*** *function is deprecated as of version 1.63: use ARMA_PARAM function instead.*
ARMA_GOF	Computes the goodness of fit measure (e.g. log-likelihood function (LLF), AIC, etc.) of the estimated ARIMA model
ARMA_LLF **(*deprecated*)**	Computes the log-likelihood function (LLF) of the estimated ARMA model. ***Note:*** *function is deprecated as of version 1.63: use ARMA_GOF function instead.*
ARMA_AIC **(*deprecated*)**	Calculates the Akaike's information criterion (AIC) of the given estimated ARMA model (with corrections to small sample sizes). ***Note:*** *function is deprecated as of version 1.63: use ARMA_GOF function instead.*
ARMA_CHECK	Examines the model's parameters for stability constraints (e.g. stationary, invertibility, causality, etc.).
ARMA_CALIBRATE **(*deprecated*)**	Computes the maximum likelihood estimate (MLE) of the model's parameters. ***Note:*** *function is deprecated as of version 1.63: use ARMA_PARAM function instead.*
ARMA_ERRORS **(*deprecated*)**	Returns an array of cells for the estimated error/standard deviation of the model's parameters ***Note:*** *function is deprecated as of version 1.63: use ARMA_PARAM function instead.*
ARMA_RESID **(*deprecated*)**	Returns an array of cells for the standardized residuals of a given ARMA model. ***Note:*** *function is deprecated as of version 1.63: use ARMA_FIT function instead.*
ARMA_FIT	Returns an array of cells for the in-sample model fitted values of the conditional mean, volatility or residuals
ARMA_FORE	Calculates the out-of-sample conditional mean forecast
ARMA_FORESD **(*deprecated*)**	Calculates the estimated error/standard deviation of the conditional mean forecast. ***Note:*** *function is deprecated as of version 1.63: use ARMA_FORE function instead.*
ARMA_FORECI **(*deprecated*)**	Returns the confidence interval limits of the conditional mean forecast. ***Note:*** *function is deprecated as of version 1.63: use ARMA_FORE function instead.*

ARMA_PARAM	Returns an array of cells for the quick guess, optimal (calibrated) or std. errors of the values of the model's parameters
ARMA_SIM	Calculates the out-of-sample simulated values

ARMAX

In principle, an ARMAX model is a linear regression model that uses an ARMA[i]-type process (i.e. w_t)to model residuals.

Function	Description
ARMAX	Returns a unique string to designate the specified ARMAX model
ARMAX_GOF	Computes the goodness of fit measure (e.g. log-likelihood function (LLF), AIC, etc.) of the estimated ARIMA model
ARMAX_FIT	Returns an array of cells for the in-sample model fitted values of the conditional mean, volatility or residuals
ARMAX_CHECK	Examines the model's parameters for stability constraints (e.g. stationarity, invertibility, causality, etc.)
ARMAX_PARAM	Returns an array of cells for the quick guess, optimal (calibrated) or std. errors of the values of model's parameters
ARMAX_FORE	Calculates the out-of-sample conditional forecast (i.e. mean, error and confidence interval)
ARMAX_SIM	Calculates the out-of-sample simulated values

ARIMA Model

The ARIMA model is an extension of the ARMA model that applies to non-stationary time series (the kind of time series with one or more integrated unit-roots).By definition, the auto-regressive integrated moving average (ARIMA) process is an ARMA process for the differenced time series.

Function	Description
ARIMA	Returns a unique string to designate the specified ARIMA model
ARIMA_CHECK	Examines the model's parameters for stability constraints (e.g. stationarity, invertibility, causality, etc.)
ARIMA_FIT	Returns an array of cells for the in-sample model fitted values of the conditional mean, volatility or residuals
ARIMA_FORE	Calculates the out-of-sample conditional forecast (i.e. mean, error, and confidence interval)
ARIMA_GOF	Computes the goodness of fit measure (e.g. log-likelihood function (LLF), AIC, etc.) of the estimated ARIMA model
ARIMA_PARAM	Returns an array of cells for the quick guess, optimal (calibrated) or std. errors of the values of model's parameters
ARIMA_SIM	Calculates the out-of-sample simulated values

SARIMA

The SARIMA model is an extension of the ARIMA model, often used when we suspect a model may have a seasonal effect. By definition, the seasonal auto-regressive integrated moving average - SARIMA(p,d,q)(P,D,Q)s - process is a multiplicative of two ARMA processes of the differenced time series.

Function	Description
SARIMA	Returns a unique string identifier to designate the specified SARIMA model
SARIMA_CHECK	Examines the model's parameters for stability constraints (e.g. stationarity, invertibility, causality, etc.)
SARIMA_FIT	Returns an array of cells for the in-sample model fitted values of the conditional mean, volatility or residuals
SARIMA_FORE	Calculates the out-of-sample conditional forecast (i.e. mean, error, and confidence interval)
SARIMA_GOF	Computes the goodness of fit measure (e.g. log-likelihood function (LLF), AIC, etc.) of the estimated SARIMA, model.
SARIMA_PARAM	Returns an array of cells for the quick guess, optimal (calibrated) or std. errors of the values of model's parameters
SARIMA_SIM	Calculates the out-of-sample simulated values

SARIMAX

In principle, an SARIMAX model is a linear regression model that uses a SARIMA-type process (i.e. w_t) This model is useful in cases we suspect that residuals may exhibit a seasonal trend or pattern.

Function	Description
SARIMAX	Returns a unique string identifier to designate the specified SARIMAX model
SARIMAX_CHECK	Examines the model's parameters for stability constraints (e.g. stationary, invertibility, causality, etc.)
SARIMAX_FIT	Returns an array of cells for the in-sample model fitted values of the conditional mean, volatility or residuals
SARIMAX_FORE	Calculates the out-of-sample conditional forecast (i.e. mean, error and confidence interval)
SARIMAX_GOF	Computes the goodness of fit measure (e.g. log-likelihood function (LLF), AIC, etc.) of the estimated SARIMA model.
SARIMAX_PARAM	Returns an array of cells for the quick guess, optimal (calibrated) or std. errors of the values of model's parameters
SARIMAX_SIM	Calculates the out-of-sample simulated values

AIRLINE Model (8)

The airline model is a special, but commonly used, case of the multiplicative ARIMA model.

Function	Description
AIRLINE	Returns a unique string to designate the specified AIRLINE model
AIRLINE_AIC **(*deprecated*)**	Calculates the Akaike's information criterion (AIC) of the given airline model (with corrections to small sample sizes). ***Note:*** *function is deprecated as of version 1.63: use AIRLINE_GOF function instead.*
AIRLINE_LLF **(*deprecated*)**	Calculates the log-likelihood function (LLF) of the given airline model. ***Note:*** *function is deprecated as of version 1.63: use AIRLINE_GOF function instead.*
AIRLINE_CHECK	Examines the model's parameters for stability constraints (e.g. stationary, etc.)
AIRLINE_RESID **(*deprecated*)**	Returns an array of cells for the standardized residuals of a given airline model ***Note:*** *function is deprecated as of version 1.63: use AIRLINE_GOF function instead.*
AIRLINE_MEAN **(*deprecated*)**	Returns an array of cells for the fitted values of the conditional mean ***Note:*** *function is deprecated as of version 1.63: use AIRLINE_GOF function instead.*
AIRLINE_FIT	Returns an array of cells for the fitted values of the conditional mean
AIRLINE_FORE	Calculates the out-of-sample conditional mean forecast
AIRLINE_FORESD **(*deprecated*)**	Calculates the estimated error/standard deviation of the conditional mean forecast. ***Note:*** *function is deprecated as of version 1.63: use AIRLINE_GOF function instead.*
AIRLINE_FORECI **(*deprecated*)**	Returns the confidence interval limits of the conditional mean forecast. ***Note:*** *function is deprecated as of version 1.63: use AIRLINE_GOF function instead.*
AIRLINE_GOF	Computes the goodness of fit measure (e.g. log-likelihood function (LLF), AIC, etc.) of the estimated AIRLINE model.
AIRLINE_SIM	Calculates the out-of-sample simulated values
AIRLINE_PARAM	Returns an array of cells for the quick guess, optimal (calibrated) or std. errors of the values of model's parameters

X-12-ARIMA Model (4)

NumXL supports the X-12-ARIMA method. X-12-ARIMA is the software produced, distributed and maintained by the United States Census Bureau.

Function	Description
X12ARIMA	Returns a unique string to designate the specified X12-ARIMA model
X12APROP	Returns the current option status (e.g. transformation function, outliers) of a given X-12-ARIMA model
X12ACOMP	Returns the value of an X-12-ARIMA model output component (e.g. trend, seasonal or irregular)
X12AFORE	Returns the forecast value and/or confidence interval limits for the X-12-ARIMA model

GARCH Model (13)

If an autoregressive moving average model (ARMA model) is assumed for the error variance, the model is a generalized autoregressive conditional heteroskedasticity (GARCH, Bollerslev(1986)) model.

Function	Description
GARCH	Returns a unique string to designate the specified GARCH model
GARCH_VOL	Returns an array of the fitted (in-sample) conditional volatilities/standard deviations (sigma)
GARCH_GUESS	Returns the initial guess of a given model's parameters
GARCH_LLF	Computes the log-likelihood function for the fitted model
GARCH_AIC	Calculates the Akaike's information criterion (AIC) of a given estimated GARCH model (with corrections to small sample sizes)
GARCH_CHECK	Examines the model's parameters for stability constraints (e.g. stationary, positive variance, etc.)
GARCH_VL	Calculates the long-run average volatility for the given GARCH model
GARCH_CALIBRATE	Computes the maximum likelihood estimated (MLE) model parameters
GARCH_ERRORS	Returns an array of cells for the estimated error/standard deviation of a given model's parameters
GARCH_RESID	Returns an array of the standardized residuals for the fitted GARCH model
GARCH_FORE	Calculates the out-of-sample conditional mean forecast
GARCH_FORESD	Calculates the estimated error/standard deviation of the conditional mean forecast
GARCH_FORECI	Returns the confidence interval limits of the conditional mean forecast
GARCH_SIM	Returns an array of cells for the model simulated values

EGARCH Model (13)

The exponential general autoregressive conditional heteroskedastic (E-GARCH) model by Nelson (1991) is another form of the GARCH model, formally called EGARCH(p,q).

Function	Description
EGARCH	Returns a unique string to designate the specified EGARCH model
EGARCH_VOL	Returns an array of the fitted (in-sample) conditional volatilities/standard deviations
EGARCH_GUESS	Returns the initial guess of a given model's parameters
EGARCH_LLF	Computes the log-likelihood function for the fitted model
EGARCH_AIC	Calculates the Akaike's information criterion (AIC) of a given estimated EGARCH model (with corrections for small sample sizes)
EGARCH_CHECK	Examines the model's parameters for stability constraints (e.g. stationary, positive variance, etc.)
EGARCH_VL	Calculates the long-run average volatility for the given EGARCH model
EGARCH_CALIBRATE	Computes the maximum likelihood estimated (MLE) model parameters
EGARCH_ERRORS	Returns an array of cells for the estimated error/standard deviation of a given model's parameters

EGARCH_RESID	Returns an array of the standardized residuals for the fitted EGARCH model
EGARCH_FORE	Calculates the out-of-sample conditional mean forecast
EGARCH_FORESD	Calculates the estimated error/standard deviation of the conditional mean forecast
EGARCH_FORECI	Returns the confidence interval limits of the conditional mean forecast
EGARCH_SIM	Calculates the out-of-sample conditional mean forecast.

GARCH-M Model (14)

In finance, the return of a security may depend on its volatility (risk). To model such phenomena, the GARCH-in-mean (GARCH-M) model adds a heteroskedasticity term into the mean equation.

Function	Description
GARCHM	Returns a unique string to designate the specified EGARCH model
GARCHM_MEAN	Returns an array of the fitted (in-sample) conditional mean values
GARCHM_VOL	Returns an array of the fitted (in-sample) conditional volatilities/standard deviations
GARCHM_GUESS	Returns the initial guess of a given model's parameters
GARCHM_LLF	Computes the log-likelihood function for the fitted model
GARCHM_AIC	Calculates the Akaike's information criterion (AIC) of a given estimated GARCH-M model (with corrections for small sample sizes)
GARCHM_CHECK	Examines the model's parameters for stability constraints (e.g. stationary, positive variance, etc.)
GARCHM_VL	Calculates the long-run average volatility for the given GARCH-M model
GARCHM_CALIBRATE	Computes the maximum likelihood estimated (MLE) model parameters
GARCHM_ERRORS	Returns an array of cells for the estimated error/standard deviation of a given model's parameters
GARCHM_RESID	Returns an array of the standardized residuals for the fitted GARCH-M model
GARCHM_FORE	Calculates the out-of-sample conditional mean forecast
GARCHM_FORESD	Calculates the estimated error/standard deviation of the conditional mean forecast
GARCHM_FORECI	Returns the confidence interval limits of the conditional mean forecast
GARCHM_SIM	Returns an array of cells for the model simulated values

COMBO Model (12)

The advanced models framework addresses the need to build a combo model. A combo model has two components: (1) conditional mean model (e.g. ARMA), and a conditional variance model (e.g. GARCH).

Function	Description
TSM_LLF	Computes the log-likelihood function for the fitted model
TSM_AIC	Calculates the Akaike's information criterion (AIC) of a given estimated mixed model (with corrections for small sample sizes)
TSM_CHECK	Examines the model's parameters for stability constraints (e.g. stationary, positive

	variance, etc.)
TSM_MEAN	Returns an array of the fitted (in-sample) conditional mean values
TSM_VOL	Returns an array of the fitted (in-sample) conditional volatilities/standard deviations
TSM_RESID	Returns an array of the standardized residuals for the fitted mixed-model
TSM_CALIBRATE	Computes the maximum likelihood estimated (MLE) model parameters
TSM_ERRORS	Returns an array of cells for the estimated error/standard deviation of a given model's parameters
TSM_FORE	Calculates the out-of-sample conditional mean forecast
TSM_FORESD	Calculates the estimated error/standard deviation of the conditional mean forecast
TSM_FORECI	Returns the confidence interval limits of the conditional mean forecast
MIXED_MODEL	Returns an array of cells for the packed form of the mixed model (i.e. conditional mean and conditional volatility model components)

Generalized Linear Model (14)

The generalized linear model (GLM) is a flexible generalization of an ordinary least squares regression. The GLM generalizes linear regression by allowing the linear model to be related to the response variable (i.e. $\mathbf{Y}$) via a link function (i.e. $\mathbf{g(.)}$)and by allowing the magnitude of the variance of each measurement to be a function of its predicted value.

Function	Description
GLM	Returns an array of cells for the packed form of a given GLM model
GLM_AIC	Calculates the Akaike's information criterion (AIC) of the GLM model (with corrections to small sample sizes)
GLM_LLF	Computes the log-likelihood function (LLF) of the GLM model
GLM_RSQ	Calculates the plain coefficient of determination (R-Squared), or the adjusted R-Squared (adjusts for the number of explanatory terms in a model)
GLM_CHECK	Examines the model's parameters for constraints (e.g. positive variance, etc.)
GLM_GUESS	Returns an array of cells for the initial (non-optimal) guess of the model's parameters
GLM_CALIBRATE	Computes the maximum likelihood estimate (MLE) of the model's parameters
GLM_ERRORS	Returns an array of cells for the estimated error/standard deviation of the model's parameters
GLM_FORE	Calculates the expected response (i.e. mean) value; given the GLM model and the values of the explanatory variables
GLM_FORESD	Calculates the standard deviation (sigma of the error terms (epsilon)) of the GLM model; given the values of explanatory variables
GLM_FORECI	Calculates the standard deviation (sigma of the error terms (epsilon)) of the GLM model; given the values of explanatory variables
GLM_MEAN	Calculates the expected response (i.e. mean) value; given the GLM model and the values of the explanatory variables
GLM_VOL	Calculates the standard deviation (sigma) of the error terms (epsilon) in the given GLM model
GLM_RESID	Returns the standardized residuals/errors of a given GLM

Simple Linear Regression Model (5)

Simple linear regression (SLR) is the least squares estimator of a linear regression model with a single explanatory variable.

Function	Description
SLR_PARAM	Calculates the OLS regression coefficients values
SLR_FITTED	Returns an array of cells for the fitted values of the conditional mean (or residuals)
SLR_FORE	Calculates the forecast value, error and confidence interval for regression model
SLR_ANOVA	Calculates the regression model analysis of the variance (ANOVA) values
SLR_GOF	Calculates a measure for the goodness of fit (e.g. R^2).

Multiple Linear Regression Model (7)

Multiple linear regression (SLR) is the least squares estimator of a linear regression model with a multiple explanatory variables.

Function	Description
MLR_PARAM	Calculates the regression coefficients values for a given input variable
MLR_FITTED	Returns an array of cells for the fitted values of the conditional mean (or residuals)
MLR_FORE	Calculates the forecast value, error and confidence interval for regression model
MLR_ANOVA	Calculates the regression model analysis of the variance (ANOVA) values
MLR_GOF	Calculates a measure for the goodness of fit (e.g. R^2).
MLR_PRFTest	Calculates the p-value and related statistics of the partial f-test (used for testing the inclusion/exclusion variables).
MLR_STEPWISE	Returns a list of the selected variables after performing the stepwise regression

Principal Component Analysis (9)

Principal component analysis (PCA) is a mathematical procedure that uses an orthogonal linear transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components.

Function	Description
PCA_COMP	Returns an array of cells for the i-th principal component (or residuals)
PCA_VAR	Returns an array of cells for the i-th principal component (or residuals)
PCR_PARAM	Calculates the regression coefficients values for a given input variable
PCR_FITTED	Returns an array of cells for the fitted values of the conditional mean (or residuals)
PCR_FORE	Calculates the forecast value, error and confidence interval for regression model
PCR_ANOVA	Calculates the regression model (of principal components) analysis of variance (ANOVA) values
PCR_GOF	Calculates a measure for the goodness of fit (e.g. R square, adjusted R square, MSRE, LLF, AIC, etc.)
PCR_PRFTest	Calculates the p-value and related statistics of the partial f-test for PCR (used for testing the inclusion/exclusion variables.
PCR_STEPWISE	Returns a list of the selected variables after performing the stepwise regression

Portfolio Analysis (3)

NumXL also features a number of functions to facilitate the multi-assets portfolio calculations

Function	Description
PORT_RET	Calculates the portfolio equivalent return
PORT_VOL	Calculates the portfolio equivalent volatility
PORT_COVAR	Calculates the covariance between two portfolios weighting

Utilities (9)

NumXL also features a number of functions that do not fit into our main function categories. These "orphan" functions, including interpolation, extrapolation, and excess kurtosis of a GED/Student’s t distribution, are grouped together in the "utilities" category.

Function	Description
INTERPOLATE	Returns an array of cells for the interpolated function value(s)
HASNA	Checks whether the input array contains any missing values and returns TRUE or FALSE
NxTokenize	Returns the n-th token/sub-string in a string after splitting it using a given delimiter
NxReplace	Returns the modified string after performing match/replace on the given string
NxMatch	Returns TRUE if the string matches the regular expression expressed
NxRegress	Calculates the value of the regression function for an intermediate x-value
MV_VARS	Calculates the number of variables in a given data set.
MV_OBS	Calculates the number of non-missing observations in a data set (X and Y).
NUMXL_INFO	Returns version and license information for the local NumXL installation