FIA_STATS.XLA

Statistics Functions Add-In for MS Excel

Developer:
Stefan Zeugner
+49 211 826 11436
stefan.zeugner@westlb.de

WestLB Research

Developed for:
Fixed Income Analysis
WestLB AG
Düsseldorf 2005

HIGHLIGHTS

-FIA Scatter Plot

Select two columns and press the "FIA Scatter Plot" Button to create a two-dimensional scatter chart with point colours passing from grey (uppermost cells of columns) to blue (lowest cells). Helps to evaluate a relationship between two data series over time.

-Statistics Worksheet Functions

Most of the formulas used are array formulas. Array formulas may contain several cells with differing results. To display all results of an array formula: select a range with as many cells as provided the formula output. Then enter the formula – after completing the formula and its parameters hold the SHIFT and CTRL buttons and press ENTER.

FiaCorrMat (inputrange)

Select an array of data (inputrange) in FiaCorrMat. The functions displays the correlation matrix between its columns.

FiaCovMat (inputrange)

Select an array of data (inputrange) in FiaCovMat. The functions displays the covariance matrix between its columns.

FiaHisto(InputArray)

Use FiaHisto to perform a frequency table of InputArray, which provides the base for a histogram chart. If InputArray has two columns, FiaHisto performs a two-dimension frequency table.

FiaLinEst (Y_Dependent, X_Regressors, Optional Const)

Use FiaLinEst to regress a vector (Y_dependent) on an array of explanatory variables (X_Regressors). Returns a row array with regression coefficients, t-statistics, R2, AIC, etc. Set Const=TRUE if you want to include a constant into the regression.

FiaLinEstParaName (Y_Dependent, X_Regressors, Optional Const)

Use FiaLinEstParaName with the same specification as in FiaLinEst, to get a clue which number returned by FiaLinEst means what.

FIAisNormal (InputData)

States TRUE if InputData (a single column data vector) is normally distributed, FALSE if not.

FiaUroot (InputSeries, Optional Reverse)

States TRUE if InputSeries (a single column data vector) is "integrated" (i.e. behaving like a e.g. a random walk), FALSE otherwise. Using integrated data in regression analysis may lead to wrong conclusions. Set Reverse=TRUE if your time series is in descending order.

FiaPCA (InputArray)

Based on the covariance matrix of InputArray (a range with data series in columns). FiaPCA returns the Eigenvalues and corresponding Eigenvectors (Principal Components, Factors), ordered by importance, i.e. Eigenvalues.

FiaPCtime (InputArray)

Based on the covariance matrix of InputArray (a range with data series in columns). Returns time series with the principal components' behaviour over time, ordered by importance of the component.

FiaLinestPCA (Y_Dependent, X_Regressors)

Uses Principal Components to calculate adjusted regression of Y_Dependent on explanatory variables in the columns of X_Regressors. Recommend if columns of X_Regressors are highly correlated, i.e. "collinear". Constant is automatically included.

FiaLinestPCAParaName (Y_Dependent, X_Regressors)

Use FiaLinEstPCAParaName with the same specification as in FiaLinEstPCA, to get a clue which number returned by FiaLinEstPCA means what.

-Worksheet Function Operators

FiaReduceFrq (InputRange, Frequency, Optional DateRange)

Convert daily data to weekly (monthly) averages in weekly (monthly) frequency by passing a range of time series, "W" ("M") as Frequency and providing a column containing time series dates in DateRange. Or simply pass InputRange, and an integer number for Frequency (e.g. 5) to obtain e.g. 5-day averages.

FiaFilter (inputrange)

Inputrange is range of time series (data series in columns) to be corrected for outliers in erroneous data. Replaces the outliers with interpolated values.

FiaDiff (InputRange)

Calculates the differences (returns) of the time series in InputRange (series in columns) FiaDiff(x)t=xt-xt-1.

FiaMergeV (Input1, Input2, Optional Input3, Optional Input4, Optional Input5, Optional Input6,...)

Creates a single array out of ranges input1, input2, etc. Returns array containing columns of input1 at the left, then columns of input2 and so forth

FiaMergeH (Input1, Input2, Optional Input3, Optional Input4, Optional Input5, Optional Input6,...)

Creates a unified array out of ranges input1, input2, etc. Returns array containing rows of input1, then rows of input2 and so forth

FiaGetCol (InputArray, Col1, Optional Col2)

Extracts one or more columns out of a matrix: Extracts either column number Col1, or the columns from number Col1 to number Col2.

FiaGetRow (InputArray, Row1, Optional Row2)

Extracts one or more rows out of a matrix: Extracts either row number Row1, or the rows from number Row1 to number Row2.

FiaMMult (Input1,Input2)

FiaMMult enables matrix multiplication just as quickly as the Excel function MMULT does. However FiaMMult is not subject to MMULT's maximum number of columns constraint.

Ctrl+Shift+V

Paste Special Values you copied before.

-Export Data to EViews

Select data and press the EViews button. Then specify whether your data is undated or daily/weekly/monthly/annual. In the latter case, please provide a start date. Select names for your data series and press the "Go EViews" Button. Your data will be extracted to a new EViews workfile.

Info

This Excel Add-In was created by WestLB AG to the best of its knowledge and in all conscience. WestLB takes no responsibility whatsoever for calculation results and functionality related to the Add-In code. The add-in may be distributed and used freely.

FIA_STATS means Fixed Income Analysis Statistics: It was developed for facilitating data work in the Fixed Income Analysis Department of WestLB Research GmbH. Most functions help to get a quick grasp of data daily coming along in Excel spreadsheets. Those functions, however, are not up to the performance of a real statistic application. Neither party involved in its development does assume any responsibility arising from the use or malfunction of FIA_STATS.

In case of questions please contact the developer:

Stefan Zeugner
+49 211 826 11436
stefan.zeugner@westlb.de
Fixed Income Analysis, Financial Markets Research
WestLB
Herzogstr. 15
D-40217 Düsseldorf

Installation

Before installation set your Macro Security to low: Open Excel → Select "Tools" on the menu bar → Select "Options" → Select the Tab "Security" → Click "Macro Security" → Select "low" → Press "OK" and "OK". Then open FIA_STATS_SETUP.XLS in Excel

Restraints

Language settings and Conventions

FIA_STATS was optimised for Excel XP (English language settings) on a Windows XP machine. It was found to work properly with (English) Excel 97 on Windows NT.

In particular, there may be problems associated with the German/Continental decimal separator "," instead of "." as well as with Continental date conventions: In this case we recommend switching to the English standard by choosing the menu "Tools" -> "Options" and then selecting the Tab "International" (German Version: "Werkzeuge" -> "Optionen" -> "International").

Associated Files

In order to provide function help, the installation routine places a fia_stats.dll in the location of fia_stats.xla. For transfer to EViews, an EViews program was positioned in the same place. In case of problems, please uninstall FIA_STATS with fia_stats_setup.xls, and reinstall it again.

VBA Programming

Most of FIA_STATS VBA Functions are normally accessible for programming, if you reference the FIA_STATS.xla in your workbook under "Tools" → "References". Caution: All arrays in FIA_STATS are two-dimensional with base 1.

For programming purposes, FIA_STATS.xla also contains the function…

FiaSortArray(inputRng As Variant, Optional Reverse As Boolean = False)

…this sorts a two-dimensional array (base 1) in the code rather than on the worksheet.

Index

FiaLinEst

☞In Depth

Returns an array containing the coefficients and statistics obtained by "least squares" regression to calculate a straight line that best fits your data. Because this function returns an array of values, it must be entered as an array formula.

The equation for the line is y=b0 + b1*x1+…+bi*xi+…

What is this for?

Least squares estimation is widely used to compile models of financial data. Adding the coefficients multiplied by the respective explanatory series yields an estimate of the dependent series. Keep in mind that FiaLinEst is only apt to model linear data, not more complicated relationships.

Syntax

FiaLinEst(**Y_Dependent**,**X_Regressors**,Optional Constant,Optional Stats,Optional HACcov,Optional tStats)

Y_Dependent:

Dependent series (a vector, i.e. a single column filled with values) whereupon regression parameters are fitted.

X_Regressors:

An array containing columns with independent variables: 1st column=x1 2nd column =x2, …
X_Regressors and Y_Dependent have to have same number of observations (=number of rows)

Optional Constant:

TRUE if constant coefficient b0 should be estimated, FALSE if no constant term to be included. Default is FALSE

Optional Stats:

TRUE if statistics like R2 to be displayed by FiaLinEst, FALSE if only coefficients b0, b1, b2, … to be returned. Default is TRUE

Optional HacCov:

Standard errors are the basis for t-statistics which help to determine the significance of parameters.

With financial data standard errors are often biased due to time-dependent volatility ("heteroskedasticity") and relatedness of data to prior developments ("auto-correlation") – which turn leads to wrong conclusions on parameter significance.

Set ***HACcov***=TRUE if FiaLinEst should adjust for these annoyances by computing HAC ("heteroskedasticity and auto-regression consistent" or "Newey-West") t-stats and standard errors. HACcov=FALSE returns "normal" standard errors and t-statistics as with Excel worksheet function LINEST. Default is TRUE

Optional tStats:

Set TRUE if you want t-Statistics returned, FALSE if standard errors should be presented in their place. Default is TRUE

Output:

Output is presented in a single row with k*2 + 12 data members, where k is the number of columns in X_Regressors (plus 1 if ***Const*** is set to TRUE).

To ease identification, use the Function ☞FiaLinEstParaName to depict the identifiers of data members. The first, left cell contains the constant b0 (if ***Const***=TRUE), then coefficients b1, b2, b3… up to the number of columns in X_Regressors.

To the right, the corresponding t-statistics are depicted: first the t-stat of the constant (if ***Const***=TRUE) then t-stat for b1, t-stat for b2, … Rule of thumb: If a t-stat is >1.96 then the corresponding coefficient is significantly different from zero.

If ***tStat***=FALSE then standard errors are presented instead of t-statistics.
To the right of t-stats, regression performance indicators are presented in the following order:

Indicators	Description
AdjR2	R-squared adjusted for some flaws due to sample size. We recommend to prefer it vs. R2. Ranges between small negative values and 1. The nearer to 1 the better the fit.
R2	"R-squared" / "Coefficient of determination": Ranges between small negative values and 1. The nearer to 1 the better the fit.
SE of Reg	"Standard error of regression": Estimated "standard deviation" of the regression errors. The smaller it is, the better the fit. Should at least be smaller than StDev Dep
StDev Dep	Standard deviation of Y_Dependent. Used to compare with SE of Reg
F-Prob	The probability for the whole FiaLinEst estimation to be worthless (i.e. not explaining anything)
DW-Stat	"Durbin-Watson Statistic": test for integratedness of regression residuals. Ideally close to 2. If lower than R2, serious hints at "spurious regression".
AIC	"Akaike Information Criterion". The lower (i.e. the "more negative"), the better the fit. Considered a "better" indicator than R2.

Moreover, FiaLinEst returns structural regression statistics of minor importance (☞See "In Depth").

Output if ***Stats*** is set to TRUE

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	b0	b1	…	tStat0	tStat1	…	AdjR2	R2	SE of Reg	StDev Dep	F-Prob	DW-Stat	AIC

Additional output of minor Importance

	N	O	P	Q	R
1	LogIf	F-Stat	Df	SS Est	SS Resid

🏍 To increase calculation speed, set ***Stats*** to FALSE if you do not need t-stats etc.; If the data is not subject to changing volatility and auto-regressive components, set ***HACcov*** to FALSE

FiaLinEstParaName

Returns an array naming the results of FiaLinEst. Because this function returns an array of values, it must be entered as an array formula.

Syntax

FiaLinEstParaName(**Y_Dependent**,**X_Regressors**,Optional Constant,Optional Stats,Optional HACcov,Optional tStats)

Use the same settings as in FiaLinEst and FiaLinEstParaName will provide the parameter identification in the same order.

Output

If Stats is set to FALSE, FiaLinEstParaName will only return the names of coefficients. If it is set to TRUE (default mode) then FiaLinEstParaName will provide the following output:

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	b0	b1	…	t-Stat 0	t-Stat 1	…	AdjR2	R2	SE of Reg	StDev Dep	F-Prob	DW-Stat	AIC

	…	N	O	P	Q	R
1	…	LogIf	F-Stat	Df	SS Est	SS Resid

If ***tStats*** (default=TRUE) is set to FALSE, "St.Err" will be displayed instead of "t-Stat" since the corresponding setting in FiaLinEst will return standard errors instead of t-statistics.

If ***HACcov*** (default=TRUE) is set to TRUE, then FiaLinEstParaName will add "(hac)" to "t-Stat" (or "St.Err") to indicate that the t-statistic (standard error) was calculated in "heteroskedasticity and autocorrelation consistent" manner.

FiaCorrMat

Returns an array displaying the correlation matrix between the data series (columns) of the input range passed. Because this function returns an array of values, it must be entered as an array formula.

Syntax

FiaCorrMat(**Inputrange**,Optional Difference,Optional Horizontal)

Inputrange:
An array containing the data series to be calculated. FiaCorrMat considers each column as one data series.

Optional Difference:
Set TRUE if FiaCorrMat should display the correlations between first differences (changes) in data series. Default=FALSE.

Optional Horizontal:
Set TRUE if data series in Inputrange are organized in rows rather than columns. Default=FALSE.

Output

If Inputrange contains k columns, FiaCorrMat returns a k x k matrix displaying the correlations between column 1 (X1) and column 1 in the left upper cell (cell(1,1)) between column 1 (X1) and column 2 (X2) in cell(1,2) and in cell (2,1) and so forth.

	A	B	C	…
1	1	Corr(X1,X2)	Corr(X1,X3)	…
2	Corr(X1,X2)	1	Corr(X2,X3)	…
3	Corr(X1,X3)	Corr(X2,X3)	1	…
…	…	…	…	…

If InputRange contains just a single row then FiaCorrMat returns the #VALUE error.

FiaCovMat

Returns an array displaying the covariance matrix between the data series (columns) of the input range passed. Because this function returns an array of values, it must be entered as an array formula.

Syntax

FiaCovMat(**Inputrange**,Optional Difference,Optional Horizontal)

Inputrange:
An Array containing the data series to be calculated. FiaCorrMat considers each column as one data series.

Optional Difference:
Set TRUE if FiaCorrMat should display the covariance between first differences (changes) in data series. Default=FALSE.

Optional Horizontal:
Set TRUE if data series in Inputrange are organized in rows rather than columns. Default=FALSE.

Output

If Inputrange contains k columns, FiaCovMat returns a k x k matrix displaying the individual covariances between column 1 (X1) and column 1 in the left upper cell (cell(1,1)) between column 1 (X1) and column 2 (X2) in cell(1,2) and in cell (2,1) and so forth.

	A	B	C	…
1	1	Corr(X1,X2)	Corr(X1,X3)	…
2	Corr(X1,X2)	1	Corr(X2,X3)	…
3	Corr(X1,X3)	Corr(X2,X3)	1	…
…	…	…	…	…

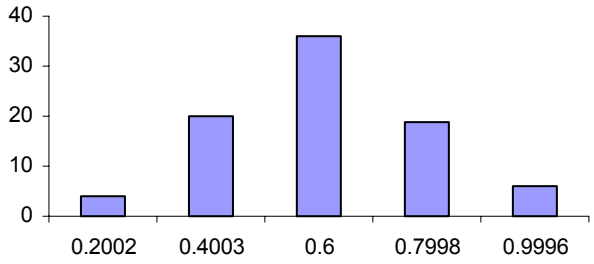
If InputRange contains just a single row then FiaCovMat returns the #VALUE error.

FiaHisto

Returns an array containing a frequency table usable for histograms for a data range provided. If the data range has two columns, then a two-dimensional cross-tabulation of frequency will be performed.

What is this for?

You may want two display a histogram representing the distribution of data. The chart below provides an example for such a histogram. The bar to the left shows that 4 observations in a data sample exhibit values between 0 and 0.2002, 20 observations exhibit values between 0.2002 and 0.4003, and so on. FiaHisto provides the basic table for such a histogram, as to be outlined below.



Syntax

FiaHisto (**InputArray,** Optional NbSteps, Optional MinBound, Optional StepSize, Optional MinBound2, Optional StepSize2)

InputArray:

An array containing the data series to be considered, with data series in columns.

Optional NbSteps:

Number of intervals (per dimension) into which data is to be classified. Default: Automatic selection (with 10 intervals at4 maximum). Is only taken into consideration if MinBound and StepSize (or MinBound2 and StepSize2 for the second dimension) are not set.

Optional MinBound:

If you want to define the exact properties of your intervals yourself, then set MinBound to the value where intervals start from. If you set MinBound, StepSize has also to be set (see below). For instance if you want you intervals to range from -3 to -2, -2 to -1, -1 to 0 and so forth, set MinBound to -3 and StepSize to 1. If MinBound and StepSize are not set, FiaHisto relies on NbSteps (respectively its default value) to perform data clustering.

Optional StepSize:

If you want to define the exact properties of your intervals yourself, then set StepSize to the size of intervals. If you set StepSize, MinBound has also to be set (see above). Example: see above. If MinBound and StepSize are not set, FiaHisto relies on NbSteps (respectively its default value) to perform data clustering.

Optional MinBound2:

If you perform a two-dimensional frequency cross-table and want to define the exact properties of the second dimension's intervals yourself, then set MinBound2 to the value where intervals start from. If you set MinBound2, StepSize2 has also to be set (see below). Example: see MinBound. If MinBound and StepSize are not set, FiaHisto relies on NbSteps (respectively its default value) to perform data clustering.

Optional StepSize2:

If you perform a two-dimensional frequency cross-table and want to define the exact properties of the second dimension's intervals yourself, then set set StepSize to the size of this dimension's

intervals. If you set StepSize2 , MinBound2 has also to be set (see above). Example: see MinBound. If MinBound and StepSize are not set, FiaHisto relies on NbSteps (respectively its default value) to perform data clustering.

Output

If InputArray contains 1 column then FiaHisto performs a one-dimensional frequency table on the data in the column.

The figures displayed in the "Frequency" columns show the number of observations which are lower than or equal to the upper interval bound row of the "1st Col =<" but larger than the interval bound provided before. For instance, in the example below, "40" is the number of observations in "C3:C219" that are lower than or equal to 0.4002 and larger than 0.2002.

Example: Return for FiaHisto(C3:C219,5), where C3:C219 is a column of numbers.

	A	B
1	1st Col =<	Frequency
2	0.2002	4
3	0.4002	20
4	0.6000	36
5	0.7998	19
6	0.9996	6

Example: Return for FiaHisto(C3:D219,,0,0.2,0,0.2), where C3:D219 is two columns of numbers.

	A	B	C	D	E	F
1	1st Col (vert) =< / 2nd Col (hor) =<	0.2	0.4	0.6	0.8	1
2	0.2	6	14	6	9	10
3	0.4	6	6	20	3	5
4	0.6	6	8	6	5	8
5	0.8	13	7	9	8	12
6	1	14	3	8	14	11

FIAisNormal

☞In Depth

Returns an array stating TRUE if a data series can be considered as normally distributed, and FALSE otherwise. You may use this function just for TRUE or FALSE information as a single cell worksheet function.

Moreover it returns a 3 x 1 array of values depicting more detailed statistics. If these statistics are to be displayed, FIAisNormal must be entered as an array formula.

What is this for?

Many financial applications require input data to be normally distributed (e.g. least squares estimation, or VaR). FIAisNormal tests for this property and moreover provides a "Jarque-Bera probability" indicating how far the input data is from normal distribution.

Syntax

FIAisNormal(**InputData**,Optional ProbVal,Optional k)

InputData:

An array containing the data to be calculated. FIAisNormal assesses the distribution of the entire set of InputData.

Optional ProbVal:

Set the significance level (between 0 and 1), i.e. the minimum probability of making an error in considering InputData as not normally distributed. Default=0.05.

Optional k:

Leave ***k*** at its default value 0. See ☞ In Depth for further discussion.

Output

FIAisNormal performs a "Jarque-Bera" (JB) test on the distribution of InputData. The lower the calculated JB statistic the more likely InputData is normally distributed.

JB probability is calculated out of JB statistic and exhibits the probability to make an error if InputData is not normally distributed. The higher this probability, the more likely InputData follows normal distribution.

If JB probability is larger than the optional input ***ProbVal*** (default=0.05) then FIAisNormal considers InputData as normally distributed and returns TRUE in the uppermost cell. If JB probability is smaller than ProbVal then InputData is considered to be not normally distributed and FIAisNormal returns FALSE.

	A
1	TRUE or FALSE
2	JB statistic
3	JB probability

If InputData contains less than 4 values, then FIAisNormal returns #DIV/0. If InputData's standard deviation is 0, then FIAisNormal returns #VALUE.

FiaURoot

☞In Depth

Returns an array stating TRUE if a time series can be considered as integrated, and FALSE otherwise. You may use this function just for TRUE or FALSE information as a single cell worksheet function.

Moreover it returns a 1 x 7 array of values depicting more detailed statistics. If these statistics are to be displayed, FiaURoot must be entered as an array formula.

What is this for?

If your time series exhibits trends and behaves like a random walk series (e.g. a stock index) then it is not normally distributed in an apparently random order. This may lead to wrong conclusions in correlation and regression analysis.

One is "spurious regression": For instance the total African population with the Dow from 1950 to 2002 exhibit a correlation coefficient of 0.85 and regressing the Dow on African population yields an R-squared of 0.72. These figures are due to trends in both series. If you take the first differences (i.e. the changes of the series instead of their levels), the regression returns an R-squared of 0.05.

Thus if FiaURoot states TRUE, the data series examined is subject to trends and you are advised to take its changes (see operator ☞FiaDiff for that purpose) for correlation and regression analysis. A DW-stat in ☞FiaLinEst is below 1 hints as well at this problem. Perform FiaURoot to be sure.

Syntax

FiaURoot(**InputSeries**,Optional Reverse,Optional iPVal,Optional ilags,Optional Urootmode)

InputSeries:

A vector (one single column) containing a time series. If you enter a range containing more than one column, FiaURoot only considers the first column of the range.

Optional Reverse:

Set to TRUE if your time series is in descending order, i.e. the most current value is in the uppermost cell. Default=FALSE, i.e. FiaURoot assumes time series to be in ascending order.

Optional iPVal:

Criterion for strictness of FiaURoot, Default=5. Set to 1 if you want FiaURoot to be more strict, or set to 10 if you prefer a more lenient approach.

Optional iLag:

We recommend leaving ***iLag*** empty, FiaURoot then decides automatically on this parameter.

Optional URootMode:

Leave empty. See ☞ "In Depth" for more details.

Output

FiaURoot performs a "Unit Root" test (more specifically an "Augmented Dickey-Fuller test") returning a 7-cell row array. The first cell is TRUE if InputSeries is integrated. Regarding the meaning of the figures in the remaining 6 cells, please consider ☞ "In Depth".

🏍 To increase calculation speed, set ***iLag*** to an integer value. We recommend the nearest integer to the third root of the number of observations in InputSeries.

FiaPCA

☞In Depth

Returns an array containing the eigenvalues and eigenvectors of the covariance matrix of the data series range you enter. Because this function returns an array of values, it must be entered as an array formula.

What is this for?

With principal components analysis, one first draws the eigenvalues of a symmetric matrix (in this case, the covariance matrix of the input data range). Each eigenvalue is assigned a corresponding eigenvector. Dividing an eigenvalue by the number of columns (in the covariance matrix) yields a percentage depicting how much of the matrix variance can be explained by the corresponding eigenvector. This method enables to reduce a set of data series to a few factors (principal components) explaining nearly the total of movements in the data set. The principal components are compiled by using eigenvectors and input data.

Syntax

FiaPCA(**Input_array**,Optional Corr,Optional OutputLine)

Input_array:

A range containing the input data series in columns. If Input_array consists of only one column, FiaPCA returns #VALUE.

Optional Corr:

Set TRUE if you want the principal components to be drawn from Input_array's correlation matrix rather than its covariance matrix. Default=FALSE.

Optional OutputLine:

If FALSE, then FiaPCA returns a (k+3) x k array (where k is the number of columns in Input_array). If TRUE, FiaPCA returns the same results in a single row (see below). Default=FALSE.

Output

FiaPCA calculates the covariance (correlation) matrix of input_array and draws the eigenvalues and eigenvectors. If k is the number of columns in Input_array, then k single eigenvalues are drawn along with k vertical eigenvectors, each containing k elements.

If ***OutputLine***=FALSE (default) then the output is structured as follows:

	A	B	…
1	"EigVal1"	"EigVal2"	…
2	1st eigenvalue	2nd eigenvalue	…
3	"EigVec 1"	"EigVec 2"	…
4	1st element of EigVec1	1st element of EigVec2	…
5	2nd element of EigVec1	2nd element of EigVec2	…
6	3rd element of EigVec1	3rd element of EigVec2	…
…	…	…	…

If OutputLine=TRUE then the same results are returned in a single line (as below):

	A	B	C	D	E	F	…
1	"EigVal1"	1st eigenvalue	"EigVal2"	2nd eigenvalue	"EigVec1"	1st element of EigVec1	…

FiaPCtime

☞In Depth

Returns an array containing the principal components series derived from the covariance matrix of the data series range you enter. Because this function returns an array of values, it must be entered as an array formula.

What is this for?

Combining eigenvectors (see ☞ FiaPCA) with the input data for principal component analysis re-centres the data series entered. Explaining the behaviour of the most important principal components (those with the largest eigenvalues) already explains a very large part of total data variation.

Syntax

FiaPCtime(**Input_array**,Optional Corr)

Input_array:

A range containing the input data series in columns. If ***Input_array*** consists of only one column FiaPCtime returns #VALUE.

Optional Corr:

Set TRUE if you want the principal components to be drawn from ***Input_array***'s correlation matrix rather than its covariance matrix. Default=FALSE.

Output

FiaPCtime calculates the covariance (correlation) matrix of ***Input_array*** and draws the eigenvalues and eigenvectors, ordered by importance (i.e. by eigenvalues). It multiplies the eigenvector matrix with ***Input_array*** to obtain a range of the same size as ***Input_array***. (I.e. there are as many principal components as there are columns in ***Input_array***). The left column is the 1st (i.e. the most important) principal component PC1, the next column is the 2nd (i.e. the second-most important) principal component PC2, and so on. The individual PCs contain observations (at time/date) t. PC1, for instance, is a vector with elements {PC1(1), PC1(2) …,PC1(t),…PC1(n)}.

	A	B	…
1	PC1(1)	PC2(1)	…
2	PC1(2)	PC2(2)	…
…	…	…	…
n	PC1(n)	PC2(n)	…

FiaLinEstPCA

☞In Depth

As in FiaLinEst, this function returns an array containing the coefficients and statistics obtained by "least squares" regression to calculate a straight line that best fits your data. In addition, FiaLinEstPCA adjusts coefficients for the correlation of explanatory variables. Because this function returns an array of values, it must be entered as an array formula.

What is it for?

The use of highly correlated ("collinear") explanatory variables in least squares estimation may bias coefficients and t-stats and may thus lead to wrong conclusions about parameter impact and model structure. FiaLinEstPCA takes advantage of principal components analysis to correct coefficients and t-stats for these effects.

If explanatory variables are uncorrelated, the results are basically the same as in FiaLinEst. For calculation purposes, a constant is automatically included among regression factors.

Syntax

FiaLinEstPCA(**Y_Dependent**,**X_Regressors**,Optional Stats,Optional tStats)

Y_Dependent:

Dependent data series (a vector, i.e. a single column) whereupon regression parameters are fitted.

X_Regressors:

An array containing columns with independent variables: 1st column=x1, 2nd column =x2, …

X_Regressors and Y_Dependent ought to have same number of observations (=number of rows). X_Regressors ought to have at least two columns.

Optional Stats:

TRUE if statistics like R^2 to be displayed by ☞FiaLinEst, FALSE if only coefficients b0, b1, b2, … to be returned. Default is TRUE

Optional tStats:

Set TRUE if you want t-Statistics to be returned, FALSE if standard errors should be presented in their place. Default is TRUE

Output

To ease identification, please use the Function ☞FiaLinEstPCAParaName to depict the identifiers of data members. Output is presented in a single row with k*2 + 12 data members, where k is the number of columns in X_Regressors plus one. Since FiaLinEstPCA is of no use with only one data series, it returns the error #USE >1 K!#, if X_Regressors contains only a single column.

The first, left cell contains the constant b0, then coefficients b1, b2, b3… up to the number of columns in X_Regressors.

To the right, the corresponding t-statistics are depicted: the t-stat of the constant cannot be calculated due to the structure of FiaLinEstPCA calculus. Then t-stat for b1, t-stat for b2, … are depicted. Rule of thumb: If a t-stat is >1.96 then the corresponding coefficient is significantly different from zero. If ***tStats***=FALSE then standard errors are presented instead of t-statistics.

If Stats is set to TRUE, then regression performance indicators are presented t o the right of t-stats in the following order:

Indicators	Description
AdjR2	R-squared adjusted for some flaws due to sample size. We recommend to prefer it vs. R2. Ranges between small negative values and 1. The nearer to 1 the better the fit.
R2	"R-squared" / "Coefficient of determination": Ranges between small negative

	values and 1. The nearer to 1 the better the fit.
SE of Reg	"Standard error of regression": Estimated "standard deviation" of the regression errors. The smaller it is, the better the fit. Should at least be smaller than StDev Dep
StDev Dep	Standard deviation of Y_Dependent. Used to compare with SE of Reg
F-Prob	The probability for the whole FiaLinEstPCA estimation to be worthless (i.e. not explaining anything)
DW-Stat	"Durbin-Watson Statistic": test for integratedness of regression residuals. Ideally close to 2. If lower than R2, serious hints at "spurious regression".
AIC	"Akaike Information Criterion". The lower (i.e. the "more negative"), the better the fit. Considered a "better" indicator than R2.

Moreover, FiaLinEstPCA returns structural regression statistics of minor importance (See ☞ In Depth).

Output if ***Stats*** is set to TRUE

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	b0	b1	…	tStat0	tStat1	…	AdjR2	R2	SE of Reg	StDev Dep	F-Prob	DW-Stat	AIC

Additional output of minor Importance

	N	O	P	Q	R
1	LogIf	F-Stat	Df	SS Est	SS Resid

🏍 To increase calculation speed, set Stats to FALSE, if you do not need t-stats etc.,

FiaLinEstPCAParaName

Returns an array naming the results of FiaLinEstPCA. Because this function returns an array of values, it must be entered as an array formula.

Syntax

FiaLinEstParaName(**Y_Dependent**,**X_Regressors**,Optional Stats,Optional tStats)

Use the same settings as in FiaLinEstPCA and FiaLinEstPCAParaName will provide parameter identifications in the same order as parameters are placed in FiaLinEstPCA.

Output

If ***Stats*** is set to FALSE, FiaLinEstParaName will only return the names of coefficients. If it is set to TRUE (default mode) then FiaLinEstParaName will provide the following output:

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	b0	b1	…	t-Stat 0	t-Stat 1	…	AdjR2	R2	SE of Reg	StDev Dep	F-Prob	DW-Stat	AIC

	…	N	O	P	Q	R
1	…	LogIf	F-Stat	Df	SS Est	SS Resid

If ***tStats*** (default=TRUE) is set to FALSE, "St.Err" will be displayed instead of "t-Stat" since the corresponding setting in FiaLinEst will return standard errors instead of t-statistics.

FiaMergeV

Returns an array combining the arrays passed to the function column by column. Because this function returns an array of values, it must be entered as an array formula.

What is it for?

Many Excel and FIA Stats array functions require input to be passed as a matrix. In FiaLinEst, for instance you cannot denote separate columns as explanatory variables – you are obliged to pass one array containing all the variables in columns. FiaMergeV allows to combine separate columns to a matrix – particularly for use in other functions: E.g. FiaLinEst (D3:D50, FiaMergeV(E3:E50, H3:H50))

Syntax

FiaMergeV(**Input1**,**Input2**,Optional Input3,Optional Input4,Optional Input5,…)

Input1, Input2, Input3,…:
Inputs are worksheet ranges with preferably the same number of rows. At least two input ranges have to be provided.

Output

An Example: Consider the two ranges A1:B3 and E1:E3

	A	B
1	1	4
2	2	5
3	3	6

…

E
A
B
C

`FiaMergeV (A1:B3, E1:E3)` returns a combined array:

	A	B	C
1	1	4	A
2	2	5	B
3	3	6	C

Empty cells in input ranges passed to FiaMergeV will be returned as 0. If input X contains more rows than the others, FiaMergeV will return an array with X's number of rows. The columns originating from other inputs will display the value 0, where no information is available.

FiaMergeH

Returns an array combining the arrays passed to the function row by row. Because this function returns an array of values, it must be entered as an array formula.

What is it for?

Many Excel and FIA Stats array functions require input to be passed as a matrix. In FiaCorrMat, for instance you cannot denote separate columns as explanatory variables – you are obliged to pass one array containing all the variables in columns. FiaMergeV allows to combine separate columns to a matrix – particularly for use in other functions:
E.g. `FiaCorrMat(FiaMergeV(B3:E4, C8:F8),,TRUE)`

Syntax

FiaMergeH(**Input1**,**Input2**,Optional Input3,Optional Input4,Optional Input5,…)

Input1, Input2, Input3,…:
Inputs are worksheet ranges with preferably the same number of columns. At least two input ranges have to be provided.

Output

An Example: Consider the two ranges A1:C2 and A6:C6

	A	B	E
1	1	3	5
2	2	4	6

…

6	A	B	C

`FiaMergeV (A1:C2, A6:C6)` returns a combined array:

	A	B	C
1	1	2	4
2	2	3	5
3	A	B	C

Empty cells in input ranges passed to FiaMergeH will be returned as 0. If input X contains more columns than the others, FiaMergeV will return an array with X's number of columns. The rows originating from other inputs will display the value 0 where no information is available.

FiaDiff

Returns an array containing the first differences (changes) of the input time series. Time series have to be in ascending order – if you want to calculate differences from reverse/descending time series, use – FiaDiff or FiaRDiff. Because this function returns an array of values, it must be entered as an array formula.

What is it for?

The application of least squares estimation, or even correlation analysis require input time series not to exhibit trends. If input data follows time trends, this may lead to wrong conclusions about parameter significance, e.g. in "spurious regression" (see ☞ FiaURoot). By taking the first differences of a series, this trend problem reduces considerably. First differences $\Delta y(t)$ of y(t) are defined as $\Delta y(t)=y(t) - y(t-q)$ where t is a time index and q is the "lag". In most applications, q=1 – i.e. with daily data y(t) the first differences $\Delta y(t)$ would depict the day-to-day changes.

In this manner you could use FiaDiff as an operator:
e.g. `FiaLinEst( FiaDiff(E3:E50), FiaDiff(B3:C50))`

Syntax

FiaDiff(Inputrange,Optional Lag,Optional Horizontal)

Inputrange:

A range with time series in columns. Time series ought to be in ascending order (i.e. the most current observation is at the bottom of the table).

Optional Lag:

Lag to construct the difference $\Delta y(t)=y(t) - y(t-q)$. If y(t) is e.g. daily data then the lag 1 would imply day-to-day changes, if y(t) is monthly data then the lag 12 would imply year-over-year changes. Default=1.

Optional Horizontal:

Set TRUE if data series in ***Inputrange*** are organized in rows rather than columns. Default=FALSE.

Output

FiaDiff considers each column of ***Inputrange*** as a time series in ascending order with the oldest observation in the uppermost cell and the most current observation in the lowest cell. The output range contains one row less than the input range and displays the differences $\Delta y(t)$ in the same column order as the input series.

FiaFilter

☞In Depth

Returns an array containing the input time series filtered for outliers. Because this function returns an array of values, it must be entered as an array formula.

What is it for

Historical time series data is frequently subject to wrong entries: e.g. a Bund 10yr yield of 1 590 000. FiaFilter searches for such outliers and replaces them by linear interpolation.

Syntax

FiaFilter(Inputrange, Optional MaxOutliers, Optional MultDev, Optional FilterSteps)

Inputrange:

A range with time series in columns.

Optional MaxOutliers

An integer determining the maximum number of outliers to be corrected (Default=10). If you want outlier identification to be stricter than by default, reduce this number.

Optional MultDev

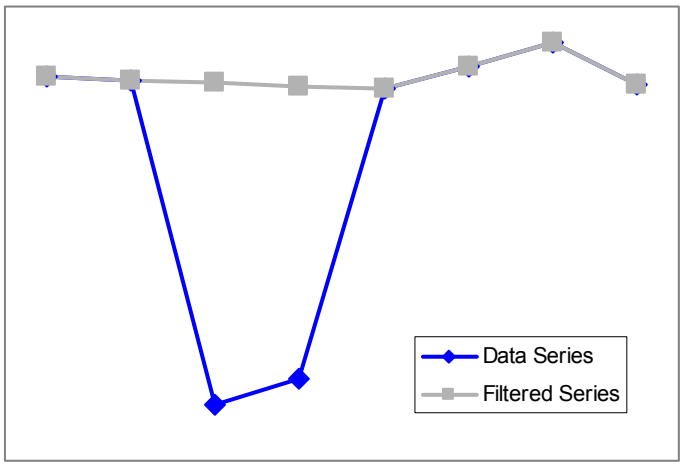
A positive number determining the threshold of identification (Default=3). If you want to increase the strictness of FiaFilter raise MultDev.

Optional FilterSteps

Leave empty, or see ☞ "In Depth" for more details.

Output

FiaFilter returns the input time series, but replaces outliers with interpolated values, i.e. by drawing a line from the last "normal" observation to the next "normal" observation (as depicted below).



FiaReduceFrq

Returns an array containing the input time series filtered for outliers in erroneous data. Because this function returns an array of values, it must be entered as an array formula.

What is it for?

You may want to reduce the frequency of the data you have: E.g. aggregate daily data to monthly data, or quarterly to annual data. FiaReduceFrq converts e.g. 5 daily observations within one week to one weekly observation.

Syntax

FiaReduceFrq(InputRange, **Frequency**, Optional DateRange, Optional ConversionMode)

InputRange:
A range with time series in columns.

Frequency
An Integer determining the length of the data cluster you want to agglomerate. For converting e.g. weekly to biweekly data, set Frequency to 2. To agglomerate to weekly data set Frequency to "W", for conversion to monthly set it to "M". You may also combine integers with "W" or "M", e.g. "2M". (Mind to type the quotation marks)

Optional DateRange
A Frequency set to "W" or "M" requires a date range denominating the days of observation to be passed. ***DateRange*** should have the same number of rows as InputRange.

Optional ConversionMode
By Default, ***ConversionMode*** is set to 2, which leads FiaReduceFrq to calculate averages. With ***ConversionMode***=1, FiaReduceFrq takes the last value of each agglomeration period, while with ***ConversionMode***=0 takes the first value. ***ConversionMode***=3 compiles the sums of agglomerations periods.

Output

FiaReduceFrq partitions ***InputRange*** into agglomeration periods you specify in Frequency and takes the average (or first/last observation or the sum) of these periods. I.e. if you pass the frequency 5, the first observation in the output of FiaReduceFrq will contain the average of the first 5 observations, the second cell will exhibit the average from the 6th to the 10th observation and so on. E.g. if you want to aggregate monthly data to quarterly data, set ***Frequency*** to 3.

You may evenly set Frequency to "W" and provide a column of dates in ***DateRange***. Then FiaReduceFrq will take all observations belonging to the same week (from MON to SUN) and compute the weekly averages. If Frequency is "M", all observations within the same month are aggregated to one observation. You may combine an integer with "M" or "W". If, for instance, you want to construct quarterly data out of some higher frequency data, you may set Frequency to "3M".

An Example: Suppose you use the daily data below.

You may use `FrqReduceFrq( B1:B10, 5 )` which in steps of five calculates the averages of the two five-day periods in the range B1:B10.

To convert from daily to weekly data you may evenly use `FrqReduceFrq(B1:B10,"W",A1:A10)`.

	A	B
1	7/5/04	101
2	7/6/04	101.5
3	7/7/04	102
4	7/8/04	102.5
5	7/9/04	103

… Converts to

D
102
104.5

6	7/12/04	103.5
7	7/13/04	104
8	7/14/04	104.5
9	7/15/04	105
10	7/16/04	101

🚲 To increase calculation speed, try to use integers in ***Frequency*** instead of "W" or "M". For instance, if you use daily data and want to convert it to weekly data, download a data set "including non-active weekdays" and set ***Frequency*** to 5 instead of "W".

FiaGetCol

Extracts one or more columns out of an array

What is it for?

FiaGetCol helps if you want for instance just two display the second column of the output of something like `FiaPCA(A2:C160)`, and you are not interested in the other columns. Then type: `FiaGetCol(FiaPCA(A2:C160),2)`

Syntax

FiaGetCol(InputArray, **Col1**, Optional Col2)

InputArray:
A matrix of values (numbers).

Col1:
The column number of the column to be extracted.

Optional Col2:
The end number of the columns you want to extract – if you want to extract several columns. For instance, If you want to extract the array from the second to the fourth column, set Col1 to 2 and Col2 to 4. For extracting just the second column, set Col1 to 2 and leave Col2 empty.

Output

FiaGetCol returns an array of values, therefore use CTRL+SHIFT+ENTER instead of just pressing ENTER when you calculate the function.

Tip: You may nest FiaGetRow and FiaGetCol, for instance if you intend to display just in the elements in the second and the third row of the third column in e.g. a FiaPCA output. Then type: `FiaGetCol(FiaGetRow(FiaPCA(A2:C160),2,4),3)`

FiaGetRow

Extracts one or more rows out of an array

What is it for?

FiaGetRow helps if you want for instance just two display the third row of the output of something like `FiaPCA(A2:C160)`, and you are not interested in the other rows. Then type:
`FiaGetRow(FiaPCA(A2:C160),3)`

Syntax

FiaGetCol(InputArray, **Col1**, Optional Col2)

InputArray:
A matrix of values (numbers).

Row1:
The row number of the row to be extracted.

Optional Row2:
The end number of the rows you want to extract – if you want to extract several rows. For instance, If you intend to extract the array from the second to the fourth row, set Row1 to 2 and Row2 to 4. For extracting just the second row, set Row1 to 2 and leave Row2 empty.

Output

FiaGetRow returns an array of values, therefore use CTRL+SHIFT+ENTER instead of just pressing ENTER when you calculate the function.

Tip: You may nest FiaGetRow and FiaGetCol, for instance if you intend to display just in the elements in the second and the third row of the third column in e.g. a FiaPCA output. Then type:
`FiaGetCol(FiaGetRow(FiaPCA(A2:C160),2,4),3)`

FiaMMult

Matrix-multiplies two arrays without the number of columns constraints imposed by Excel.

What is it for?

The Excel MMult function only allows matrix multiplying where the output does not contain more than 72 columns. FiaMMult overcomes this restraint, by using the quickness of MMult for smaller matrices, and by calculating the output for larger matrices.

Syntax

FiaMMult(Input1, **Input2**)

Input1:
A matrix of values (numbers).

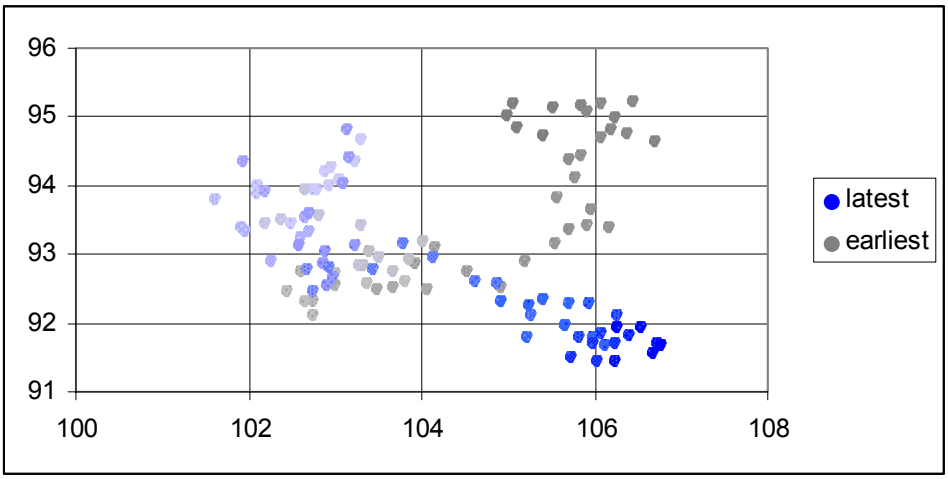
Input2:
A matrix of values (numbers), having the same number of rows as Input1 has columns.

Output

FiaGetRow returns an array of values, therefore use CTRL+SHIFT+ENTER instead of just pressing ENTER when you calculate the function.

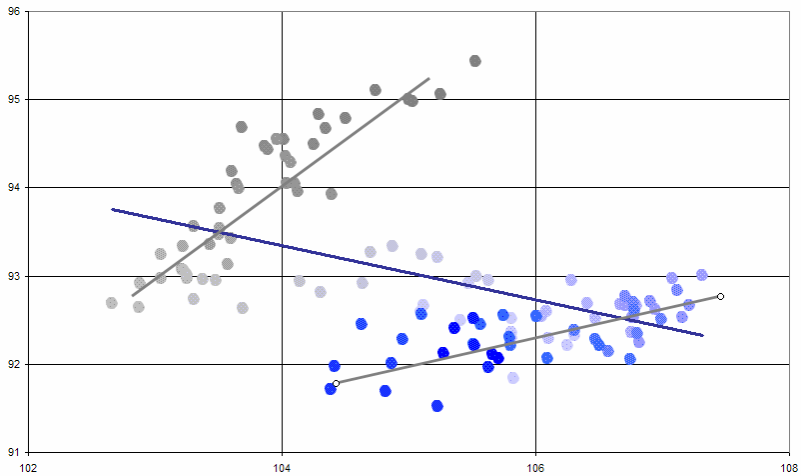
FIA Scatter Plot

FIA Scatter Plot creates a chart not feasible with plain Excel charts. It produces a scatter plot displaying two time series just as the ordinary XY (Scatter) chart type of Excel. However in FIA Scatter Plot the colour of points varies according to time, as in the example below:



What is it for?

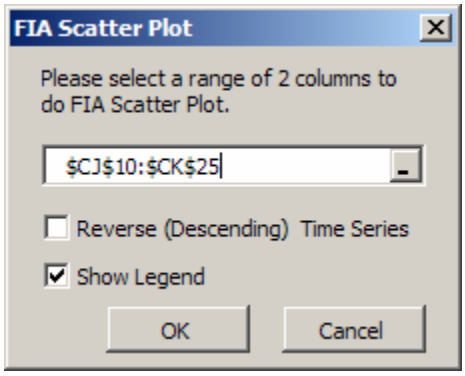
Scatter plots are frequently used to look at the relationship between two data series, for instance to identify the parameter b of a model y= a + b x where x and y are data series. However this relationship may change over time, which is difficult to identify with the classic XY (Scatter) chart of Excel. FIA Scatter Plot renders the identification of such time dependencies much quicker. Thus it is about for the same purpose as FiaURoot was intended for: To distinguish white noise from "spurious" relationships.



By regarding the points of the example scatter plot above as uniform many would assume a Least Squares trend line through all points, just as the bold blue line above. But the facts are different. Through time, the data points evolved from grey to blue. In the beginning their relationship was like the upper grey line and later on like the lower grey line. Thus the colours facilitate the recognition of specification errors such as the blue trend line.

Operation

Press the button in the menu bar to start FIA Scatter Plot. The following Input box will appear:



Select the range with the data for your Scatter Plot with your mouse. It has to consist of two columns: Either you select it like in the example above (e.g. `$A$1:$B$30`) or you select one column, press the CTRL button and select another column of equal length (in this case the range will appear like `$A$1:$A$30,$B$5:$B$35`).

Tip: Select your two columns before activating FIA Scatter Plot, and the range will already appear "filled-in". Thus you only have to click OK.

Show Legend

Leave "Show Legend" checked if you prefer a legend like in the first graph to be displayed (depicting the colours for "earliest" and "latest"). Unselect if you do not want a legend (like for instance in the second graph).

Reverse (Descending) Time Series

Check "Reverse (Descending) Time Series" for ensuring that the colours follow the right direction: Normally, the uppermost cell (i.e. the earliest observation) in your range is dark grey and the farthest below is shown in intense blue (i.e. the most recent observation). If you check "Reverse (Descending) Time Series", FIA Scatter Plot will assume the time series to start from below.

Output

Output will be the chart as depicted above. You may change some of its Options just as with a normal Excel chart. However, we were not capable to implement the whole chart as a true Excel chart: It is first created as a normal Excel bubble chart, and then the colour procedure is performed as a VBA macro. Several of the charts features, notably the colours, need this macro. If you change the cart properties, the macro is not re-performed, and a change of several features may therefore lead to in-attractive results.

Therefore: If you want to change the data source of your FIA Scatter Plot, we recommend you to delete it and then press the button once again with your new data source.

Caution: Since FIA Scatter Plot uses the normal Excel Bubble Chart, the bubble sizes have to be defined somewhere. If input data comprises less than 120 rows, these bubble sizes are included as a string (➔ the procedure will not bother you). However, if the data range has more than 120 rows, you will be asked to a column full of value 1 somewhere in your workbook.

Exporting Excel Data to EViews

By pressing the button, an extracting form (as depicted below) appears and enables you to extract vertically oriented data series to EViews. The data provided to FIA STATS will be opened in a new workfile in a new instance of EViews.

Data Range

Refer to the Excel range you want to extract in "Data Range" (by pressing Ctrl you are able to select distinct columns).

Set Frequency

Then set the frequency of your data set. EViews accepts 6 types of frequency: "undated/irregular" and five time series frequencies. If one of the latter is selected, it is required to provide a start date in order that EViews can label the observations.

Date of First observation

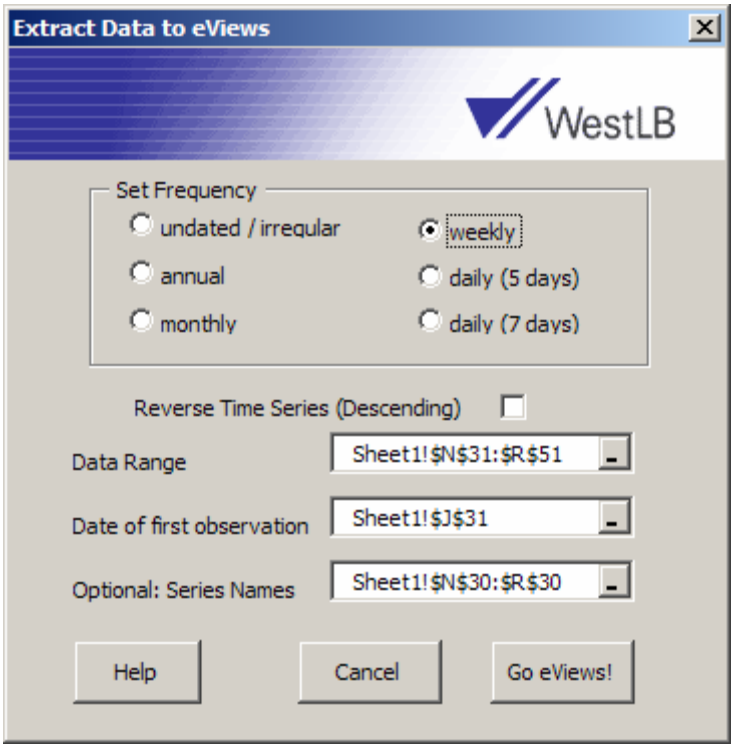
Provide the start date of your data sample, if you selected a frequency other than "undated / irregular". If the frequency is monthly, weekly or daily, please take care that the start date is provided as an Excel date, not as a string.

Optional: Series Names

You may select a row of strings describing the variables. The data series in EViews will be named in the order of the cells provided in "Series Names". Since EViews does not support special characters as series names, FIA STATS corrects for familiar signs such as spaces, etc. Bear in mind, that the procedure may not display names fully if there are strange characters provided.

By Pressing the "Go eViews!" Button, your data set is extracted to EViews. Non-number values are replaced by "NA".

Note: The extracting form automatically takes the range you selected before starting the extracting procedure. If the first cell contains a string, FIA STATS assumes the first row to contain series names.



In Depth – Technical Description of Advanced Functions

The following pages explain several advanced functions more closely. In particular the item "Reasoning" explains the logics behind procedures in order to facilitate understanding of advanced parameters.

Index

In Depth – FiaLinEst

Syntax

FiaLinEst(**Y_Dependent**,**X_Regressors**,Optional Constant,Optional Stats,Optional HACcov,Optional tStats)

Y_Dependent:

Dependent single-column array whereupon regression parameters are fitted.

X_Regressors:

An array containing columns with independent variables: 1st column=x1 2nd column =x2, …
X_Regressors and Y_Dependent have to have same number of observations (=number of rows)

Optional Constant:

TRUE if constant coefficient b0 should be estimated, FALSE if no constant term to be included. Default is FALSE

Optional Stats:

If ***Stats***=TRUE, then regression statistics like R2 are returned, if Stats=FALSE, then those statistics will not be calculated (saves time). Default is TRUE

Optional HacCov:

If ***HACcov***=TRUE then FiaLinEst "heteroskedasticity and auto-regression consistent" ("Newey-West") standard errors respectively t-stats. ***HACcov***=FALSE returns "normal" standard errors and t-statistics as with Excel worksheet function LINEST. Default is TRUE

Optional tStats:

If ***tStats***=TRUE then t-stats are returned, if ***tStats***=FALSE then standard errors are returned in the place of t-stats. Default is TRUE

Output:

Output is presented in a single row with k*2 + 12 data members, where k is the number of columns in X_Regressors (plus 1 if Const is set to TRUE).

To ease identification use the Function ☞FiaLinEstParaName to depict the identifiers of data members.

The first, left cell contains the constant b0 (if Const=TRUE), then coefficients b1, b2, b3… up to the number of columns in X_Regressors.

To the right, the corresponding t-statistics are depicted: first the t-stat of the constant (if Const=TRUE) then t-stat for b1, t-stat for b2, etc.

If ***tStats***=FALSE then standard errors are presented instead of t-statistics.

To the right of t-stats, regression performance indicators are presented in the following order:

Indicators	Description
AdjR2	R-squared adjusted for some flaws due to sample size. We recommend to prefer it vs. R2. Ranges between small negative values and 1. The nearer to 1 the better the fit.
R2	"R-squared" / "Coefficient of determination": Ranges between small negative values and 1. The nearer to 1 the better the fit.
SE of Reg	"Standard error of regression": Estimated "standard deviation" of the regression errors. The smaller it is, the better the fit. Should at least be smaller than StDev Dep
StDev Dep	Standard deviation of Y_Dependent. Used to compare with SE of Reg
F-Prob	The probability for the whole FiaLinEst estimation to be worthless (i.e. not explaining anything)

DW-Stat	"Durbin-Watson Statistic": test for integratedness of regression residuals. Ideally close to 2. If lower than R2, serious hints at "spurious regression".
AIC	"Akaike Information Criterion". The lower (i.e. the "more negative"), the better the fit. Considered a "better" indicator than R2.
loglf	Log-Likelihood
F-Stat	F-Statistic for entire regression
Df	Degrees of freedom (observations - parameters)
SS Est	ESS Sum of Squares of fitted values minus their average
SS Resid	RSS Sum of Squares of estimation errors

Reasoning

The parameter and standard error calculation is standard for ordinary least squares estimation (compare Glatzer/Hackl 2000, chapters 2 to 3). F-statistic and prob value are calculated as in Glatzer/Hackl (2000, chapter 4), maximum likelihood and AIC as in EViews (2001).

Newey-West heteroscedasticity and auto-correlation consistent standard errors are an extension of White's heteroscedasticity-consistent standard errors and are calculated as in the formula provided by EViews (2001, p. 278).

The Durbin Watson Statistic is calculated as follows:

$$DW = \frac{\sum_{t=2}^{n}(e_t - e_{t-1})^2}{\sum_{t=1}^{n} e_t^2}$$ where DW is the Durbin-Watson Statistic, t is the observation index (i.e. time), e_t is the estmation error at observation t, n the total number of observations. If errors are not auto-correlated, DW should take the value of 2.

The log-likelihood ℓ (or loglf) is an important non-classic measure of goodness-of-fit. It is calculated as follows:

$$\ell = -\frac{n}{2}\left(1 + \ln(2\pi) + \ln\left(RSS/n\right)\right)$$

…where n is the number of observations and RSS is the residual (estimation error) sum of squares

The AIC, or Akaike Information Criterion draws on the properties of log-likelihood, and represents a statistic comparable for different estimations of the same dependent variable. It basically takes the log-likelihood and "punishes" for the number of parameters (=k in the formula)

$$AIC = -2\,\ell/n + 2\,k/n$$

The lowest AIC hints at the "best" model: It is particularly preferable to R^2 when lagged terms of the dependent variable are included.

In Depth – FIAisNormal

Syntax

FIAisNormal(**InputData**,Optional ProbVal,Optional k)

InputData:

An array containing the data to be calculated. FIAisNormal assesses the distribution of the entire set of ***InputData***.

Optional ProbVal:

Set the significance level (between 0 and 1), i.e. the minimum probability of committing an error in considering ***InputData*** as not normally distributed. Default=0.05.

Optional k:

Number of parameters used to create the series to be tested.

Output

FIAisNormal performs a "Jarque-Bera" (JB) test on the distribution of InputData. The lower the calculated JB statistic the more likely ***InputData*** is normally distributed.

JB probability is calculated out of JB statistic and exhibits the probability to commit an error if InputData is not normally distributed. The higher this probability, the more likely ***InputData*** follows normal distribution.

If JB probability is larger than the optional input ***ProbVal*** (default=0.05) then FIAisNormal considers ***InputData*** as normally distributed and returns TRUE in the uppermost cell. If JB probability is smaller than ***ProbVal*** then ***InputData*** is considered to be not normally distributed and FIAisNormal returns FALSE.

	A
1	TRUE or FALSE
2	JB statistic
3	JB probability

If ***InputData*** contains less than 4 values, then FIAisNormal returns #DIV/0. If InputData's standard deviation is 0, then FIAisNormal returns #VALUE.

Reasoning

FIA_Stats uses the JB statistic formula as provided by EViews (2001, p. 153)

$$JB = \frac{N-k}{6}\left(S^2 + \frac{(K-3)^2}{4}\right)$$

where N is the number of observations, k is the number of estimation coefficients to estimate the input series, S is the skewness of the input series, and K is its kurtosis. The resulting JB statistic is distributed as X^2 with 2 degrees of freedom. A JB statistic too high leads to the rejection of the null hypothesis of a normally distributed series.

FIAisNormal uses Excel functions for skewness, kurtosis and X^2-prob-value computation.

In Depth - FiaURoot

Unit Root test (Augmented Dickey-Fuller test) for Input series

Syntax

FiaURoot(**InputSeries**,Optional Reverse,Optional IPVal,Optional Ilags,Optional Urootmode)

InputSeries:
A vector (one single column) containing a time series. If you enter a range containing more than one column, FiaURoot only considers the first column of the range.

Optional Reverse:
Set to TRUE if your time series is in descending order, i.e. the most current value is in the uppermost cell. Default=FALSE, i.e. FiaURoot assumes time series to be in ascending order.

Optional iPVal:
Significance level criteria for the rejection of null-hypothesis that input series is integrated. Choose 10 for a significance level of 0.10, 5 for 0.05, and 1 for 0.01. Default=5.

Optional iLag:
Manual lag length selection; if not set, then FIAisNormal performs automatic lag length selection. (For an explanation of lag length settings, see below)

Optional URootMode:

Determines whether a constant with/without a trend is added to the Unit root equation: 0 for no constant or trend, 1 for a constant, 2 for a constant and a trend. Default=1.

Output

FiaURoot performs an "Augmented Dickey-Fuller test", returning a 7-cell row array.

	A	B	C	D	
1	TRUE or FALSE	"t-stat"	unit root t-statistic value	IPVal "% critical value"	…

	E	F	G
…	Critical barrier for null-hypothesis rejection at significance value IPVal	"chosen lag"	Lag length chosen automatically or manually

Reasoning

A unit root test basically tests whether a time series is integrated, i.e. whether it tends to explode into one direction for a certain period (as opposed to a stationary, random series). The following outline of the test type "Augmented Dickey-Fuller Test" is rather brief, for more information see Glatzer/Hackl (2000, p. 112) or EViews (2001) – in fact FiaURoot performs the same operation as EViews' ADF test does.

$$y_t = \alpha y_{t-1} + \varepsilon_t$$

If α>1 then the series y is supposed to be integrated. The equation is re-transformed into

$$\Delta y_t = y_{t-1} - y_t = (\alpha - 1) y_{t-1} + \varepsilon_t = \gamma y_{t-1} + \varepsilon_t$$

In order to correct for auto-correlation, lagged Δy_t are included in the regression.

$$\Delta y_t = \gamma y_{t-1} + \sum_p \Delta y_{t-p} + \varepsilon_t$$

The number of lags p (parameter ***iLag***) is either decided on by the user – the integer closest to the number of observations to the power of $^1/_3$ is frequently chosen. If this option is left empty by the user (or if the number chosen is too large), then FIA_STATS automatically selects the lag length corresponding to minimal AIC. The lag chosen manually or automatically is reported in the 7th cell of FiaURoot output.

Moreover, a constant and/or a trend are frequently added to the structure (parameter ***URootMode***). An equation with constant is default in FIA_STATS.

Thus if $\gamma<0$ then y is stationary. However, to decide whether γ is significantly negative, normal t-statistics are not sufficient due to specifics of the unit root test. MacKinnon provides test statistics that are consistently stricter and based on Monte Carlo simulation. FIA_STATS incorporates MacKinnon statistics for the 1% 5% and 10% significance level (to be set by parameter ***iPVal***) and a certain number of observations. The corresponding "critical barrier" is reported in the 5th cell of FiaURoot output.

The coefficient γ, divided by its standard error ("t-stat"), reported in the 3rd cell of FiaURoot output) is compared to this critical barrier. If it is lower, then γ is significantly below zero and FiaURoot returns TRUE in its output's 1st cell.

In Depth – FiaPCA

Eigenvalue and eigenvector calculation for the covariance/correlation matrix of two or more data series

Syntax

FiaPCA(**Input_array**,Optional Corr,Optional OutputLine)

Input_array:

A range containing the input data series in columns. If ***Input_array*** consists of only one column, FiaPCA returns #VALUE.

Optional Corr:

Set TRUE if you want the principal components to be drawn from ***Input_array***'s correlation matrix rather than its covariance matrix. Default=FALSE.

Optional OutputLine:

If FALSE, then FiaPCA returns a (k+3) x k array (where k is the number of columns in ***Input_array***). If TRUE, FiaPCa returns the same results in a single row (see below). Default=FALSE.

Output

FiaPCA calculates the covariance (correlation) matrix of ***Input_array*** and draws the eigenvalues and eigenvectors. If k is the number of columns in ***Input_array***, then k single eigenvalues are drawn along with k vertical eigenvectors, each containing k elements.

If ***OutputLine***=FALSE (default) then the output is structured as follows:

	A	B	…
1	"EigVal1"	"EigVal2"	…
2	1st eigenvalue	2nd eigenvalue	…
3	"EigVec 1"	"EigVec 2"	…
4	1st element of EigVec1	1st element of EigVec2	…
5	2nd element of EigVec1	2nd element of EigVec2	…
6	3rd element of EigVec1	3rd element of EigVec2	…
…	…	…	…

If ***OutputLine***=TRUE then the same results are returned in a single line (as below):

	A	B	C	D	E	F	…
1	"EigVal1"	1st eigenvalue	"EigVal2"	2nd eigenvalue	"EigVec1"	1st element of EigVec1	…

Reasoning

FiaPCA (for Fixed Income Analysis: Principal Component Analysis) first calculates the covariance or correlation matrix for ***Input_array***. Subsequent orthogonalisation returns eigenvalues and corresponding eigenvectors which are ordered by largest eigenvalue in the output (from left to right). The orthogonalisation procedure used is "Jacobi", based on a code by Sermier (2003). This method only works properly with symmetric matrices (as is the case with covariance/correlation matrices). In order to perform eigenvalue/eigenvector computation on any symmetric matrix, use the functions FiaEigVal and FiaEigVec, which return eigenvalues and eigenvectors in similar, but random order. For more profound explanation of the mathematical procedures please refer to (☞ In Depth – FiaPCtime).

In Depth – FiaPCtime

Returns an array containing the principal components series derived from the covariance matrix of the data series range you enter.

Syntax

FiaPCtime(**Input_array**,Optional Corr)

Input_array:
A range containing the input data series in columns. If ***Input_array*** consists of only one column FiaPCtime returns #VALUE.

Optional Corr:
Set TRUE if you want the principal components to be drawn from ***Input_array***'s correlation matrix rather than its covariance matrix. Default=FALSE.

Output

FiaPCA calculates the covariance (correlation) matrix of ***Input_array*** and draws the eigenvalues and eigenvectors, ordered by importance (i.e. by eigenvalues). It multiplies the eigenvector matrix with ***Input_array*** to obtain a range of the same size as ***Input_array***. (I.e. there are as many principal components as there are columns in ***Input_array***). The left column is the 1st (i.e. the most important) principal component P_1, the next column is the 2nd (i.e. the second-most important) principal component P_2, and so on. The individual Ps contain observations (at time/date) t. P_1, for instance, is a vector with elements {$P_1(1)$, $P_1(2)$ …, $P_1(t)$,… $P_1(n)$}. FiaPCtime returns the matrix P comprising all vectors P_i by columns.

	A	B	…
1	PC1(1)	PC2(1)	…
2	PC1(2)	PC2(2)	…
…	…	…	…
N	PC1(n)	PC2(n)	…

Reasoning

Principal components analysis (PCA) in the financial markets is commonly applied to the covariance or correlation matrix of data series (mainly time series). Based on the eigenvalues and eigenvectors drawn from this symmetric matrix, PCA is able to replicate a data series' sample behaviour in an orthogonalised fashion. I.e. through PCA, respectively the function FiaPCtime, a sample **P** with an equal number of data series is produced which are perfectly orthogonal (i.e. not correlated to each other), and which are attached to eigenvalues. The series $\mathbf{P_1}$ (the first column of FiaPCtime output) corresponding to the largest eigenvalue captures the maximum amount of variance possible from the entire original data sample. The principal component $\mathbf{P_2}$ explains the maximum variance possible of the part of the sample that is not explained by $\mathbf{P_1}$ and so forth.

The mathematical background is as follows:

A multivariate data sample has a covariance (correlation) matrix $\mathbf{\Sigma}$, defined as

$$\mathbf{\Sigma} = \frac{\mathbf{X'X}}{n}$$

…if the data series in **X** have mean zero; respectively a covariance (correlation) as in FiaCovMat (FiaCorrMat). $\mathbf{\Sigma}$ is a k x k matrix, where **X** is the matrix of original data series in columns, k is the number of columns in **X**, and n is the number of rows.

There exists a matrix of eigenvectors that produces the behaviour requested in the first paragraph of "Reasoning". Denote by **W** a k x k matrix of factor weights (orthogonal eigenvectors), and by $\mathbf{\Lambda}$ a k x k diagonal matrix of eigenvalues of $\mathbf{\Sigma}$. Then a **W** and a $\mathbf{\Lambda}$ can be found such that…

$$\mathbf{\Sigma W} = \mathbf{W\Lambda}$$

Define the i-th principal component of **X** as…

$$\mathbf{P_i} = \mathbf{X w_i}$$

…where $\mathbf{w_i}$ is the i-th (eigen-)vector of **W**. Since its columns are orthogonal and normalized to one, $\mathbf{W'W=I}$ and $\mathbf{W^{-1}=W'}$.

In practice the index 1 is attributed to the eigenvector corresponding to the largest eigenvalue, the index 2 to second largest and son on. So the matrix of Principal Components **P** is defined as follows:

$$\mathbf{P} = \mathbf{XW}$$

The matrix **P** contains the principal components $\mathbf{P_i}$ by column; all of these components have mean 0, variance 1 and are orthogonal (uncorrelated) to each other. It is important that the columns in the matrix X used have a mean of zero, therefore the X used in calculating FiaPCtime is not the original data matrix but a "center " one – i.e. the elements of X consist of the elements of the original data matrix Y minus the column averages (arithmetic means).

$$\mathbf{X} = \mathbf{Y} - \overline{\mathbf{Y}} \quad \text{where} \quad \overline{\mathbf{Y}} = \left\{ \mathbf{y_{t,i}} - \overline{\mathbf{y}}_\mathbf{i} \right\}$$

Therefore the covariance matrix is **P'P** reduces to $\mathbf{\Lambda}$, a diagonal variance matrix.

$$\mathbf{P'P} = \mathbf{W'X'XW} = \mathbf{W'W\Lambda\Lambda W'W} = \mathbf{\Lambda} \qquad \text{since} \quad \mathbf{W'W} = \mathbf{I}$$

This implies that the variance of each principal component $\mathbf{P_i}$ is identical to its corresponding eigenvalue.

The output by FiaPCtime is exactly **P**.

In Depth – FiaLinEstPCA

FiaLinEstPCA regresses a dependent data series on (at least two) explanatory series. In contrast to ☞FiaLinEst, it adjusts coefficients and t-statistics for multicollinearity among explanatory factors.

Syntax

FiaLinEstPCA(**Y_Dependent**,**X_Regressors**,Optional Stats,Optional tStats)

Y_Dependent:
Dependent data series (a vector, i.e. a single column) whereupon regression parameters are fitted.

X_Regressors:
An array containing columns with independent variables: 1st column=x1, 2nd column =x2, …
X_Regressors and Y_Dependent ought to have same number of observations (=number of rows). X_Regressors ought to have at least two columns.

Optional Stats:
TRUE if statistics like R^2 to be displayed by FiaLinEst, FALSE if only coefficients b0, b1, b2, … to be returned. Default is TRUE

Optional tStats:
Set TRUE if you want t-Statistics to be returned, FALSE if standard errors should be presented in their place. Default is TRUE

Output

To ease identification use the Function FiaLinEstPCAParaName to depict the identifiers of data members. Output is presented in a single row with k*2 + 12 data members, where k is the number of columns in X_Regressors plus one. Since FiaLinEstPCA is of no use with only one data series, it returns the error #USE >1 K!#, if X_Regressors contains only a single column.
The first, left cell contains the constant b0, then coefficients b1, b2, b3… up to the number of columns in X_Regressors.

To the right, the corresponding t-statistics are depicted: the t-stat of the constant cannot be calculated due to the structure of FiaLinEstPCA calculus. Then t-stat for b1, t-stat for b2, … are depicted. If tStat=FALSE then standard errors are presented instead of t-statistics.
If Stats is set to TRUE, then regression performance indicators are presented t o the right of t-stats in the following order:

Indicators	Description
AdjR2	R-squared adjusted for some flaws due to sample size. We recommend to prefer it vs. R2. Ranges between small negative values and 1. The nearer to 1 the better the fit.
R2	"R-squared" / "Coefficient of determination": Ranges between small negative values and 1. The nearer to 1 the better the fit.
SE of Reg	"Standard error of regression": Estimated "standard deviation" of the regression errors. The smaller it is, the better the fit. Should at least be smaller than StDev Dep
StDev Dep	Standard deviation of Y_Dependent. Used to compare with SE of Reg
F-Prob	The probability for the whole FiaLinEst estimation to be worthless (i.e. not explaining anything)
DW-Stat	"Durbin-Watson Statistic": test for integratedness of regression residuals. Ideally close to 2. If lower than R2, serious hints at "spurious regression".
AIC	"Akaike Information Criterion". The lower (i.e. the "more negative"), the better the fit. Considered a "better" indicator than R2.
loglf	Log-Likelihood
F-Stat	F-Statistic for entire regression

Df	Degrees of freedom (observations - parameters)
SS Est	ESS Sum of Squares of fitted values minus their average
SS Resid	RSS Sum of Squares of estimation errors

Moreover, FiaLinEstPCA returns structural regression statistics of minor importance (See ☞ "In Depth").

Output if Stats is set to TRUE

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	b0	b1	…	tStat0	tStat1	…	AdjR2	R2	SE of Reg	StDev Dep	F-Prob	DW-Stat	AIC

Additional output of minor Importance

	N	O	P	Q	R
1	LogIf	F-Stat	Df	SS Est	SS Resid

Reasoning

The rationale for the following procedure stems from Alexander (2001, pp. 172-174): It draws on the properties of PCA (principal components analysis) to correct least squares estimators distorted by multicollinearity of exogenous variables. Multicollinearity causes serious bias in multivariate regression coefficients (compare Glatzer/Hackl 2000): check the correlation matrix of your exogenous factors with FiaCorrMat – high correlation coefficients between exogenous series indicate multicollinearity. The problem may be faced by several means, one of it being the adjustment of coefficients through PCA.

The procedure followed by FiaLinEstPCa can broadly be divided into 6 steps:

Step 1: Center the exogenous data matrix **X**

$$x^*_{t,i} = \frac{x_{t,i} - \bar{x}_i}{\sigma_i} \qquad \mathrm{X} = \left\{ \mathrm{x}_{\mathrm{t,i}} \right\}$$

where X is a matrix whose columns are data series – thus with elements $x_{t,i}$. The resulting matrix **X*** with elements $x^*_{t,i}$ contains thus the same data columns, but each with mean normalized to 0 and standard deviation 1.

Step 2: Draw principal components from **X***

$$\mathbf{X}^* = \mathbf{P}\mathbf{W}^*$$

…where matrix **P** is the principal components over time (just as in ☞FiaPCtime) and **W** the corresponding matrix of factor weights (or eigenvectors).

Step 3: Regress the dependent y on the principal components over time P

$$\mathbf{b} = \left(\mathbf{P}'\mathbf{P}\right)^{-1}\mathbf{P}'\mathbf{y} = \mathbf{\Lambda}^{-1}\mathbf{P}'\mathbf{y}$$

Regressing the dependent data series vector **y** on the principal components is straightforward akin to the familiar ordinary least squares formula. The covariance matrix **P 'P** is equal to the diagonalised eigenvalue matrix $\mathbf{\Lambda}$.

Since $\mathbf{y} = \mathbf{a} + \mathbf{P}\mathbf{b} + \boldsymbol{\varepsilon}$ and $\mathbf{P} = \mathbf{X}^*\mathbf{W}$ then

$$\mathbf{y} = \mathbf{a} + \mathbf{X}^*\mathbf{b}^* + \boldsymbol{\varepsilon} \qquad \text{where } \mathbf{b}^* = \mathbf{W}\mathbf{b}$$

…**a** is the constant vector, i.e. a vector uniformly containing elements a_t

Step 4: Transform into the original model structure

$\mathbf{y} = \mathbf{c} + \mathbf{X}\mathbf{d} + \boldsymbol{\varepsilon}$ where $\mathbf{d} = \boldsymbol{\Sigma}\mathbf{W}\mathbf{b}$ and $\boldsymbol{\Sigma}$ is a diagonal matrix with i-th main diagonal elements $1/\sigma_i$ and **c** is the constant vector in the original setting, i.e. $\mathbf{c} = \mathbf{a} - \{\mathbf{d}'\bar{\mathbf{x}}\}$ where **a** is the PCA regression vector, **d** is the newly obtained coefficient vector and $\bar{\mathbf{x}}$ is the vector containing the means of **X** (means per column). The new structure with c and d provides the model structure applicable to original data **X** with PCA-adjusted coefficients.

Step 5: Compile inverted variance matrix of b V(b)

$$\mathbf{V(b)} = RSS/(n-k)\,(\mathbf{P}'\mathbf{P})^{-1} = RSS/(n-k)\,\boldsymbol{\Lambda}^{-1}$$

The inverted estimator variance matrix of **b** is the inverted covariance matrix of **P**, multiplied by error variance (i.e. residual sum of squares divided by the number of observations minus the number of regression parameters k).

Step 6: Compile inverted variance matrix of **d V(d)** in order to compute standard errors and t-stats

$$\mathbf{V(d)} = \mathbf{W}'\boldsymbol{\Sigma}\mathbf{V(b)}\boldsymbol{\Sigma}\mathbf{W}$$

The inverted estimator variance of matrix adjusted coefficients **d** is a **V(b)** transformed from orthogonalised to "original" via **W** and adjusted for data series variances through **Σ**.

Out from **V(d)**, it is straight-forward to compute coefficient standard errors from the square roots of **V(d)**'s main diagonal elements (a corresponding t-stat is a coefficient divided by its standard error). It has to be noted that the constant c is not included in this procedure, therefore FiaLinEstPCA delivers no standard errors/t-stats for the constant.

Caution: The variance matrix transformation procedure deviates from Alexander (2001), since there is a calculation/spelling error in her book (p.173).

In Depth – FiaFilter

Searches a time series for outliers and corrects outliers through linear interpolation. The results are very responsive to parameter settings; therefore please examine the syntax and "Reasoning" closely.

Syntax

FiaFilter(Inputrange, Optional MaxOutliers, Optional MultDev, Optional FilterSteps)

Inputrange:
A range with time series in columns.

Optional MaxOutliers
An integer determining the maximum number of outliers to be corrected (Default=10). If you want outlier identification to be stricter than by default, reduce this number.

Optional MultDev
A positive number determining the threshold of identification (Default=3). If you want to increase the strictness of ***FiaFilter*** raise ***MultDev***.

Optional FilterSteps
Maximum number of observations an "outlier" may last.

Output

FiaFilter returns the input time series, but replaces outliers with interpolated values. For the identification of outliers, please refer to "Reasoning" below.

Reasoning

FiaFilter assumes the array passed to consist of time series organized in columns. For each column, it repeats the procedure as outlined below:

Step 1: transform time series into series of first differences

In order to search for outliers, FiaFilter looks for deviations in the first differences $\Delta y(t)=y(t)-y(t-1)$

Thus a time series like the chart on the left (below) transforms into a series depicted to the right.

As depicted in the chart to the left, the outlier in the right chart has transformed into a "start outlier" and a "counter outlier" denoting the end of the outlier period.

Step 2: Compile standard deviation

Calculate the standard deviation for the entire time series

Step 3: Compare observations to standard deviation bounds

For each observation, check if its value deviates more from the mean than ***MultDev*** times the series' standard deviation.

$$\text{If} \quad \left|x_t - \bar{x}\right| > \text{MultDev}\,\sigma_t$$

Step 4: Search for subsequent "counter-outlier"

If there is an outlier as defined in Step 2, there has to be a "counter-outlier" soon after, as depicted in Step 1. If the "counter-outlier" occurs less than ***FilterSteps*** observations after the "start outlier" then the period between both (comprising start and end) is classified as "outlier", to be replaced by linear interpolation.

Step 5: Replace outlier period with linearly interpolated values

FiaFilter replaces the observations in the outlier period by linear interpolation between the observation before the "start outlier" and the one after the "counter-outlier". This is akin to drawing a line between the latter two points in the original data series (not in the one consisting of first differences).

Step 6: Adjust to maximum number of outliers

If FiaFilter has found more than ***MaxOutliers*** "outlier periods" in the entire column/time series, it raises ***MultDev*** and starts the procedure form the beginning in order to reduce the number of outliers to the most severe one – i.e. FiaFilter never corrects more outlier periods than the number provided by ***MaxOutliers***.

The parameter ***MaxOutliers*** thus represents an easy-to-use option to adjust FiaFilter strictness e.g. for volatile time series. For the more advanced user, however, we recommend using ***MultDev*** in combination with ***FilterSteps*** to fine-tune the function.

References

Literature:

EViews (2001): EViews 4.0 User's Guide; Quantitative Micro Software LLC, Irvine CA.

Alexander C. (2001): Market Models; John Wiley & Sons, Ltd., Chichester.

Glatzer E. and P. Hackl (2000): Ökonometrie; 6. Auflage, Wirtschaftsuniversität Wien, Vienna,

Programming References:

Longre L. (2003): Funcustomize.xla
http://longre.free.fr/english/

Estima (2004): Formula from MacKinnon code at
http://www.estima.com/procs/DFUNIT.SRC

Sermier (2003): Jacobi orthogonalisation procedure for VBA
http://www-irma.u-strasbg.fr/~mmaumy/enseignement/Esiea/vp1.bas