Aarhus School of Business

Handelshøjskolen i Århus

Manual

EViews 4.0

By

Christian Nagstrup

Description

Basic insight into EViews

IT-Department

Ver. 021120

TABLE OF CONTENTS

Introduction

EViews is an econometrics package, which provides data analysis, regression and forecasting tools. EViews can be useful for many different analyses, but this introduction will focus on financial analysis. Once you get familiar with EViews, the program is very user-friendly. Unlike Microsoft Excel, however, EViews is not a spreadsheet program, and this can cause a few problems in the beginning. In the next section we will look at how to avoid such problems. At the same time it is worth mentioning that EViews has an excellent help function, where you can seek additional assistance.

1 EViews Files and Data

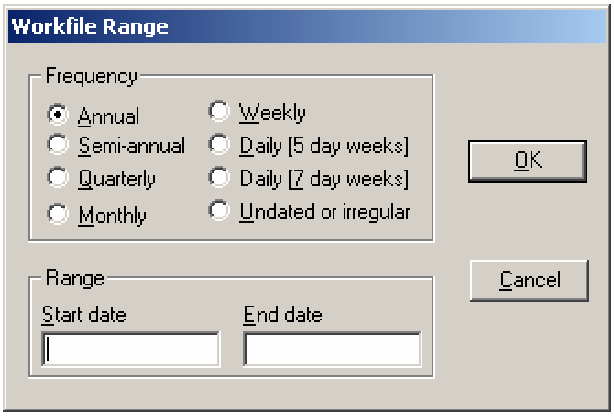
The following sections describe how to create a new workfile and import data into EViews. Different ways of handling the data in the workfile are shown as well.

1.1 Creating a workfile

Before you carry out any analysis you must first create a so-called workfile, which must be of the exact size and type as the data you want to work with. After the workfile is created EViews will let you import files from Excel, Lotus, ASCII (text files) etc. Data from SAS or SPSS cannot be directly imported to EViews. Instead the data must be saved as a text file before importing[1]. Entering data directly into EViews can be done, but is generally easier to do this in Excel first.

To create a workfile click **File → New → Workfile** and the following dialog box will appear:

[1] The SPSS manual includes a description of this procedure. For SAS you can refer to the student instructors at the IT-Department.

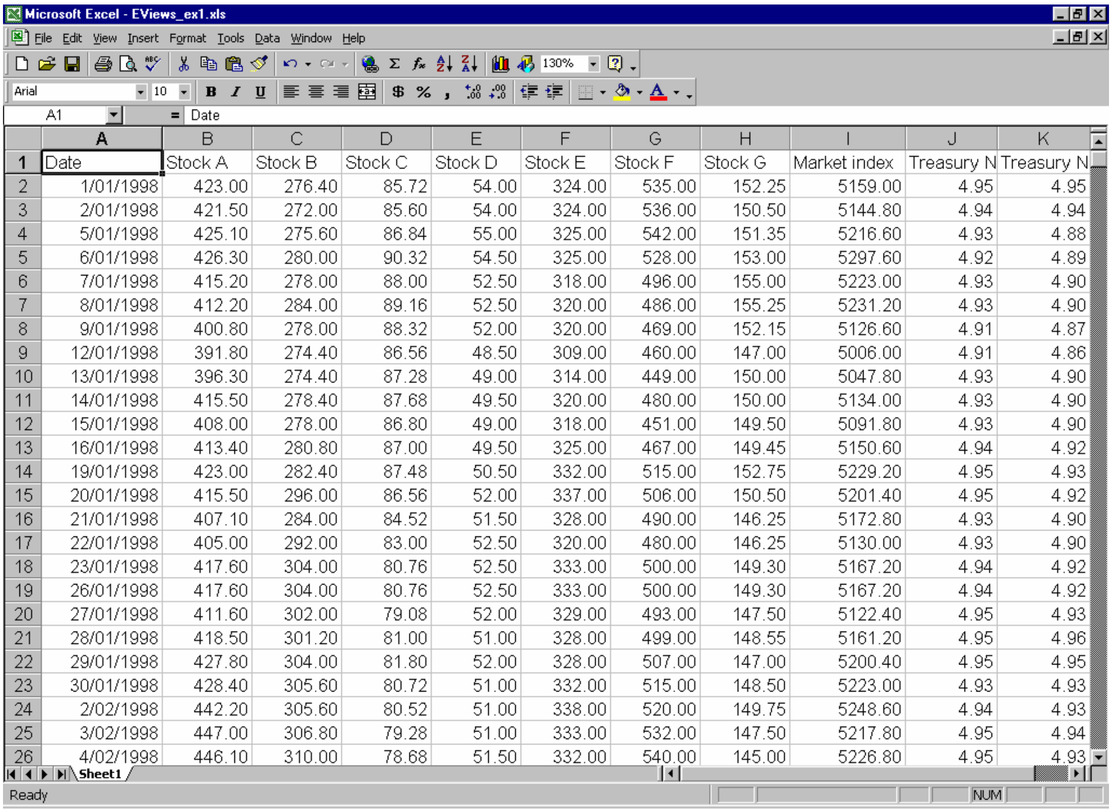


If you are working with time series data you need to know the sampling period (daily, monthly, quarterly etc.) as well as the starting and finishing date for the data. If you are working with cross-sectional data you need the number of observations. So in the latter case you should choose ***Undated or irregular*** and enter the start observation and the end observation in the appropriate textboxes. We will now look at an example where time series data is imported from an Excel file using the import function[2]. It can be done by copy-and-paste as well, which is described in EViews' help function.

1.2 Importing time series data from Excel

The file we would like to import (***X:\Eviews\Eviews_ex1.xls***) can be viewed below.

[2] The procedure for importing text files is very similar and will therefore not be demonstrated here.



	A	B	C	D	E	F	G	H	I	J	K
1	Date	Stock A	Stock B	Stock C	Stock D	Stock E	Stock F	Stock G	Market index	Treasury N	Treasury N
2	1/01/1998	423.00	276.40	85.72	54.00	324.00	535.00	152.25	5159.00	4.95	4.95
3	2/01/1998	421.50	272.00	85.60	54.00	324.00	536.00	150.50	5144.80	4.94	4.94
4	5/01/1998	425.10	275.60	86.84	55.00	325.00	542.00	151.35	5216.60	4.93	4.88
5	6/01/1998	426.30	280.00	90.32	54.50	325.00	528.00	153.00	5297.60	4.92	4.89
6	7/01/1998	415.20	278.00	88.00	52.50	318.00	496.00	155.00	5223.00	4.93	4.90
7	8/01/1998	412.20	284.00	89.16	52.50	320.00	486.00	155.25	5231.20	4.93	4.90
8	9/01/1998	400.80	278.00	88.32	52.00	320.00	469.00	152.15	5126.60	4.91	4.87
9	12/01/1998	391.80	274.40	86.56	48.50	309.00	460.00	147.00	5006.00	4.91	4.86
10	13/01/1998	396.30	274.40	87.28	49.00	314.00	449.00	150.00	5047.80	4.93	4.90
11	14/01/1998	415.50	278.40	87.68	49.50	320.00	480.00	150.00	5134.00	4.93	4.90
12	15/01/1998	408.00	278.00	86.80	49.00	318.00	451.00	149.50	5091.80	4.93	4.90
13	16/01/1998	413.40	280.80	87.00	49.50	325.00	467.00	149.45	5150.60	4.94	4.92
14	19/01/1998	423.00	282.40	87.48	50.50	332.00	515.00	152.75	5229.20	4.95	4.93
15	20/01/1998	415.50	296.00	86.56	52.00	337.00	506.00	150.50	5201.40	4.95	4.92
16	21/01/1998	407.10	284.00	84.52	51.50	328.00	490.00	146.25	5172.80	4.93	4.90
17	22/01/1998	405.00	292.00	83.00	52.50	320.00	480.00	146.25	5130.00	4.93	4.90
18	23/01/1998	417.60	304.00	80.76	52.50	333.00	500.00	149.30	5167.20	4.94	4.92
19	26/01/1998	417.60	304.00	80.76	52.50	333.00	500.00	149.30	5167.20	4.94	4.92
20	27/01/1998	411.60	302.00	79.08	52.00	329.00	493.00	147.50	5122.40	4.95	4.93
21	28/01/1998	418.50	301.20	81.00	51.00	328.00	499.00	148.55	5161.20	4.95	4.96
22	29/01/1998	427.80	304.00	81.80	52.00	328.00	507.00	147.00	5200.40	4.95	4.95
23	30/01/1998	428.40	305.60	80.72	51.00	332.00	515.00	148.50	5223.00	4.93	4.93
24	2/02/1998	442.20	305.60	80.52	51.00	338.00	520.00	149.75	5248.60	4.94	4.93
25	3/02/1998	447.00	306.80	79.28	51.00	333.00	532.00	147.50	5217.80	4.95	4.94
26	4/02/1998	446.10	310.00	78.68	51.50	332.00	540.00	145.00	5226.80	4.95	4.93

The following 5-step procedure for importing *time series data* can be used:[3]

1) Examine the contents of the file in Excel and note

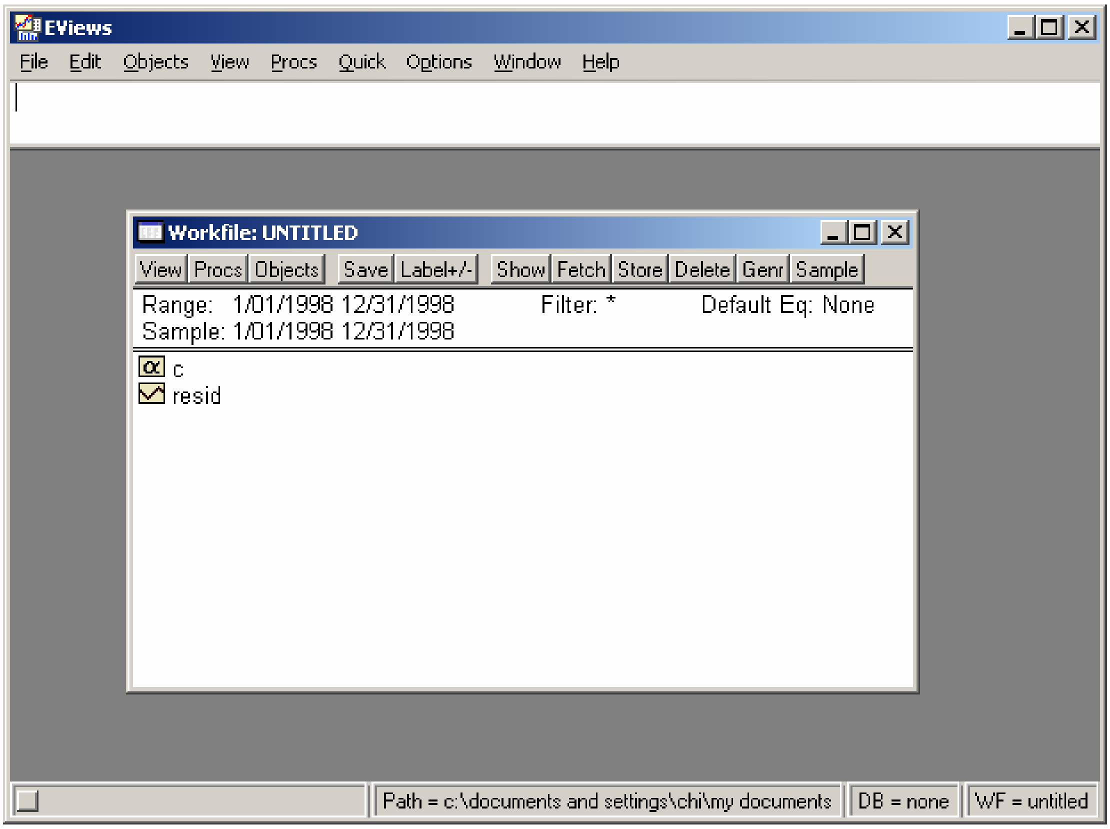
- the start and end date of the observations
- the cell where the data starts (usually B2)
- the names of the variables and the order in which they appear
- the sheet name

This example has daily (5 day weeks) data with a start date of 01-01-1998 and an end date of 31-12-1998. The data starts at B2 in the sheet called *Sheet1*. There are 11 variables. Some of them have very long names so it will be a good idea to change these to make them easier to work with later.

2) Create a new workfile as described above.

[3] For cross-sectional data you note the number of observations in the dataset instead of the start and end dates. This should be the only difference.

You should choose ***Daily [5 day weeks*]** and enter *01/01/1998* as the start date and *12/31/1998* as the end date. Be aware of the American format, MM/DD/YYYY. In the case of quarterly data you would enter *1975:3* e.g. for the third quarter of 1975. Monthly data follows the same pattern, i.e. *1975:1* means January 1975 and *1975:3* means March 1975. After clicking ***OK*** you should end up with the workfile shown below.



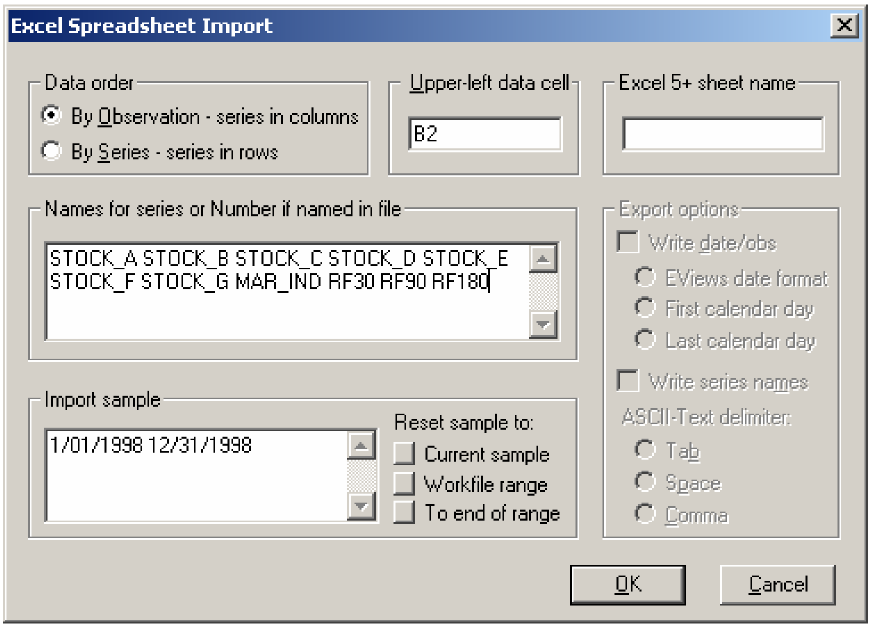
The range as well as the sample is the period between 01-01-1998 and 31-12-1998. There are always two different series, *C* and *RESID*, as default. *C* is the column that will contain the coefficients from the last regression equation you have estimated. *RESID* is the column that will contain the residuals from your last estimated model.

3) Click **Procs → Import → Read Text-Lotus-Excel**. In the dialog box for ***Open*** choose the Excel format and browse for the file. Select the file and click ***Open***.

NB!! Remember to close the file in Excel before you try to import it to EViews. Otherwise there will be an error message.

4) A dialog box now appears in which it is very crucial to enter the correct information. Any mistakes could result in an incomplete or even wrong dataset. This is where our former check-up of the Excel file becomes very important.

In this example the dialog box should be filled out as follows:

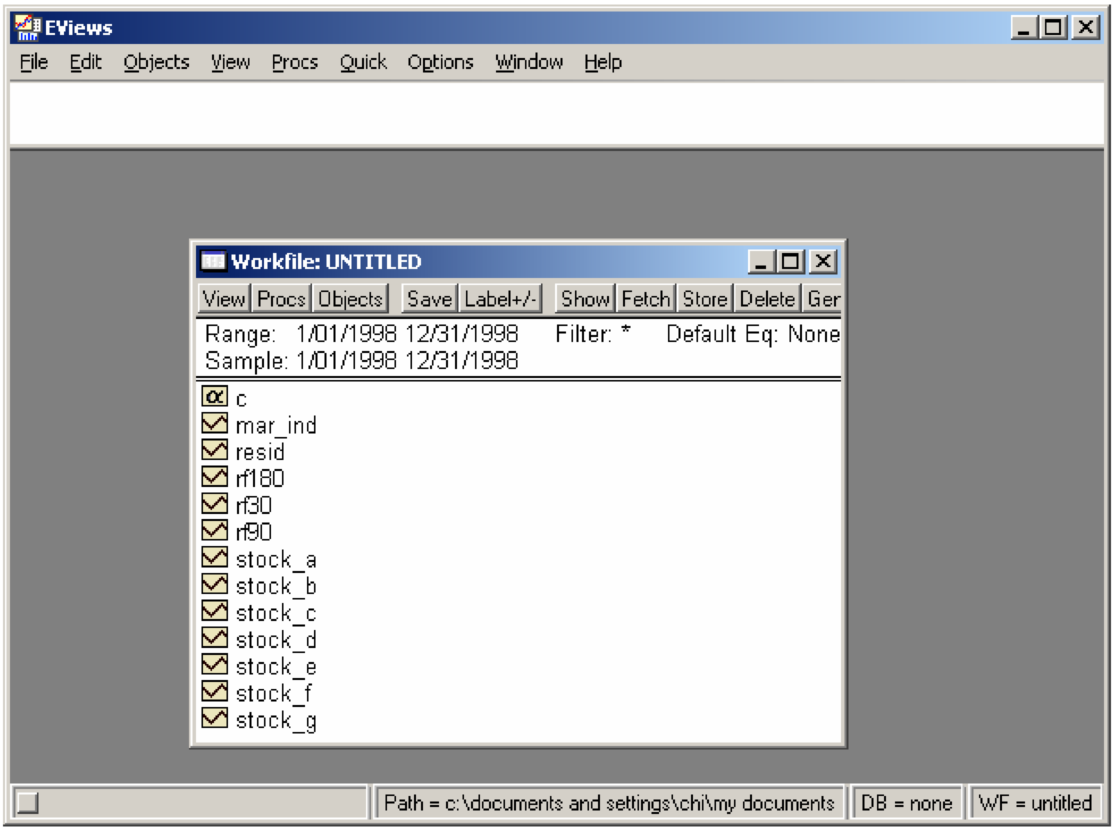


The order of data is (as in most cases) ***By Observation – series in columns***. Upper-left data cell is B2 and the sheet name is *Sheet1*. If there is only a single sheet (as in this example) it is not necessary to enter the name of it.

The names of the series/variables have been changed (notice that no spaces are allowed in the names) in order to make them easier to work with. However, if you would like to import the names that are given in Excel you simply enter the number of series (in this case 11). These names can then be changed in EViews using the ***Rename*** function. However, using this method can cause problems if for example the names start with a number and are very similar (e.g. names such as *7 DAY RETURN*, *30 DAY RETURN* etc.).

The sample to import is taken from the workfile. Here it is possible to exclude periods, which can be useful in case you would like to get rid of any outliers.

The workfile that you should have by now is shown below.



It contains a list of the 11 imported variables in alphabetical order as well as the two columns for the estimated coefficients and the residuals. It is always a good idea to check if the first and the last imported variables have been correctly imported[4]. In this example you do this by double clicking *STOCK_A* and *RF180* and compare these series with the Excel file.

Another useful approach is to open the two variables as a group. You do this by:

- clicking the variable *STOCK_A*
- holding down the [Ctrl] key and clicking the variable *RF180*
- clicking **View → Open as One Window → Open Group**

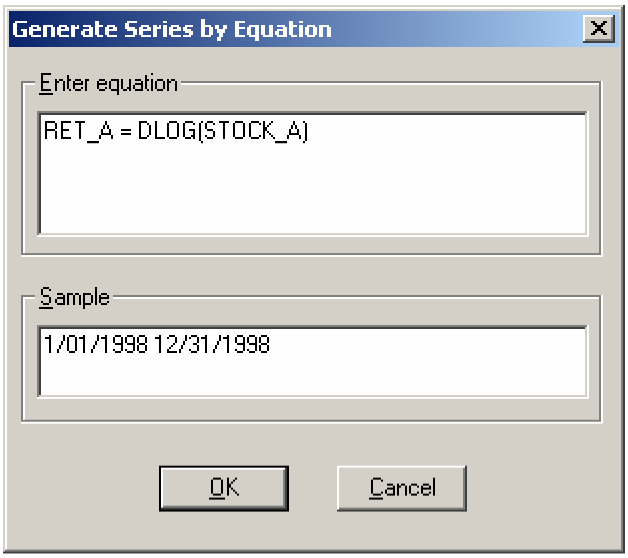
  <u>or</u> simply just right clicking or double clicking on either of the selected series and then clicking ***Open Group***

[4] Common errors include rows with "NA" and numbers that are too high or low (if the decimal separator has been misinterpreted – for instance 43500 instead of 435,00).

5) If you are certain that you have imported the data into the EViews workfile correctly you should now save this workfile by clicking **File → Save As.** The workfile will be saved in EViews' own '.wf1' format. A saved workfile can be opened later by selecting **File → Open File → Workfile** from the main menu.

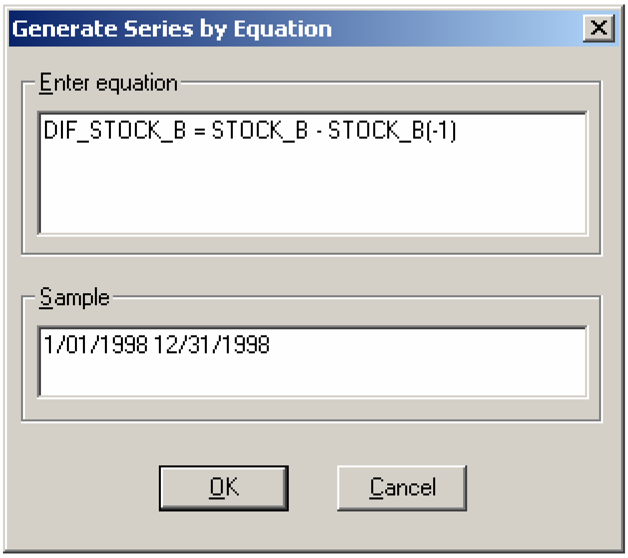
1.3 Transforming the data

It is often useful to transform existing variables for various purposes. This can be done in EViews using the ***[Genr]*** button in the top right hand corner of your workfile. Say you would like to find the return on *STOCK_A* in the ***X:\Eviews\Eviews_ex1.wf1*** workfile. The continuous return can quickly be calculated by means of the DLOG function. This finds the difference between the log of the current price and the log of the previous price; $\ln(P_t) - \ln(P_{t-1})$. Simply click the ***[Genr]*** button and enter the equation below followed by ***OK***.



This will create the variable *RET_A* and include it in the workfile. You can view the returns by double clicking the variable.

Besides DLOG there are naturally a number of other mathematical functions as well as simple addition, subtraction, division and multiplication available. A useful one worth mentioning here as well is the lag function. Lagging a variable is simply done by entering the name of the followed by *(-k)*, where k is the number of lags. So if you would like to calculate the first difference for the variable *STOCK_B* for example, you create the following variable:



If you need the lagged variable in a regression model for example, you could simply enter the variable *STOCK_B(-1)* in your equation.

1.4 Creating seasonal dummy variables

We will demonstrate how to create such dummy variables by assuming that we have a workfile containing quarterly data for the period 1975:1 (first quarter of 1975) to 2000:4 (fourth quarter of 2000). To create a dummy variable (named *Q4*) with 1's in the fourth quarter of each year and 0's in other quarters, you click the ***[Genr]*** button in your workfile and enter the following equation:

Q4 = @SEAS(4)

If you are working with monthly data, the number in the bracket refers to months instead. So if you for example would like to test for the 'January effect' on the stock market you could create a dummy variable (named *JAN*), which has a value of 1 in January each year and 0 in the remaining months. *JAN* is created by entering the following equation:

JAN = @SEAS(1)

If you wish to create a dummy variable to model the impact of the stock market crash of October 1987 in your fictive workfile with monthly data, you can follow this quick procedure: Click the ***[Genr]*** button in your workfile and enter the equation *DUM1987 = 0*. When the new dummy variable is created you click the ***[Genr]*** button again. This time you change the sample from [1975:1 2000:4] to [1987:10 1987:10] (i.e. October 1987 only) and enter the equation *DUM1987 = 1*. The variable DUM1987 now contains a value of 1 in October 1987 and 0 values in all other months.

1.5 Copying output

Any graph or equation output can easily be copied into a Word document for example. To copy a table, simply select the area you want to copy and click **Edit → Copy**. A dialog box should appear, where you would usually select the first option: ***Formatted – copy numbers as they appear in table***. Then you go to Word and paste the selected area and change the size of the output until it suits your document.

To copy a graph, click on it and a blue border should appear, then click **Edit → Copy**. In the appearing dialog box, click ***Copy to Clipboard*** and then paste into Word. Again the size can be adjusted to a suitable size.

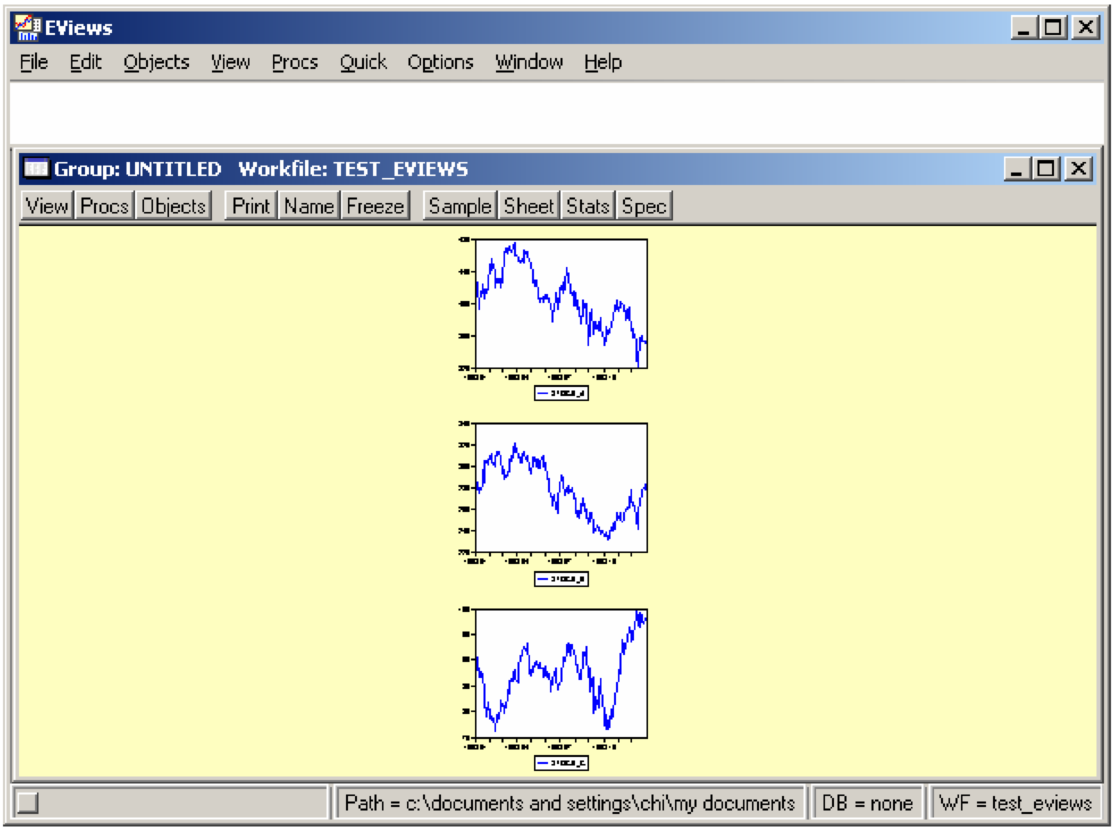
2 Examining the data

EViews can be used for examining the data in a variety of ways. In the following a few will be demonstrated.

2.1 Displaying line graphs

If you want to select a few variables and display a line graph of each of the series, you can follow this example based on the workfile ***X:\Eviews\Eviews_ex1.wf1***.

In this example we want to view the first three time series in the workfile, *STOCK_A, STOCK_B and STOCK_C*. The procedure is to highlight the three variables (using the mouse and the [Ctrl] key) followed by a double or right click. Then you click ***Open Group*** and click the ***[View]*** button in the appearing spreadsheet. From this menu you click **Multiple Graphs → Line** and the three line graphs depicted below appear. As you can see there are other choices of graphs as well. In general, clicking the ***[View]*** button mentioned above offers you many options of viewing your selected data.

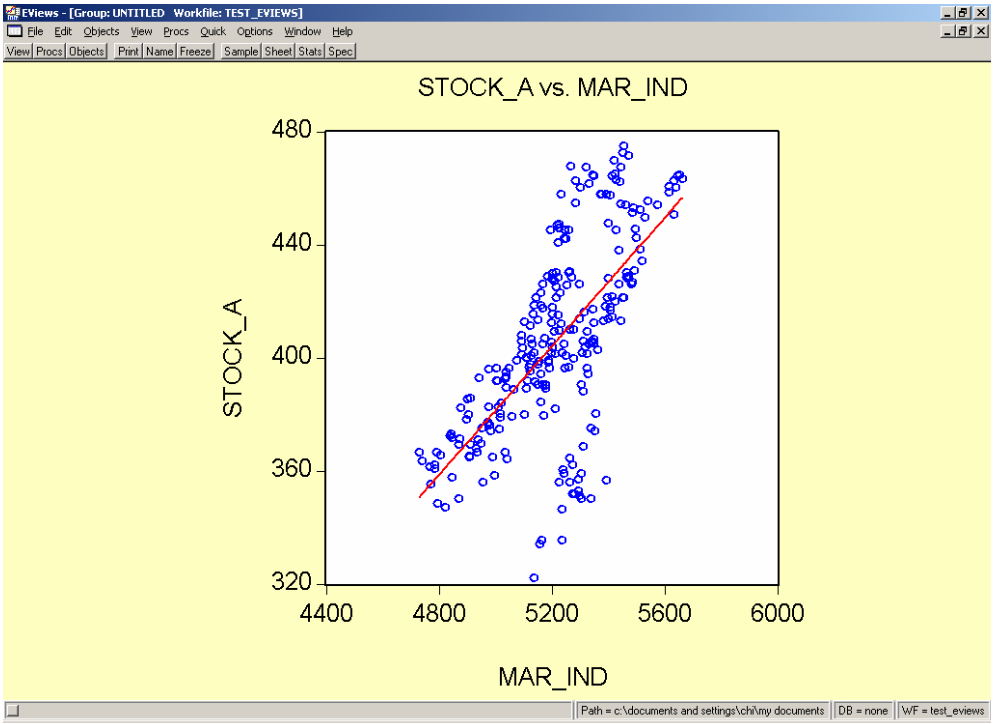


If you want to save the output in your workfile for later use, you first click the ***[Freeze]*** button. In the new window which appears you click the ***[Name]*** button. In the dialog box you enter a name for the output and click ***OK***. Now the output appears with a graph icon in your workfile.

2.2 Drawing a scatter plot

The procedure is similar to the one just mentioned above, but when using scatter plots in connection with a regression model it is now important to highlight the independent variable first, and then hold the [Ctrl] key and click the dependent variable. This secures that a regression line will be correctly drawn.

We can show an example from our workfile (***X:\Eviews\Eviews_ex1.wf1***) where *STOCK_A* is plotted against *MAR_IND*. In a regression model we want *STOCK_A* to be the dependent variable and *MAR_IND* to be the dependent variable. Consequently we first highlight *MAR_IND* followed by the other variable. Double click and choose ***Open Group*** in order to open a spreadsheet with the two variables. Click the ***[View]*** button and choose **Graph → Scatter → Scatter with Regression**. A dialog box will now appear. For default values just click ***OK*** and the scatter plot shown below should appear. If you only need a simple scatter plot just choose **Simple Scatter** instead. To save the output in your workfile use the same procedure as described in section 2.1.



2.3 Obtaining descriptive statistics and histograms

You can obtain a histogram and the descriptive statistics of a series by double clicking the series in the workfile. In the appearing spreadsheet you click the ***[View]*** button and choose **Descriptive Statistics → Histogram and Stats.**

If you want to obtain descriptive statistics for several series at a time instead, you highlight the relevant series (using the mouse and the [Ctrl] key), double or right click and choose ***Open Group***. In the appearing spreadsheet you click the ***[View]*** button and choose **Descriptive Statistics → Individual Samples.** This procedure will not give you the histograms, however.

Again the procedure from section 2.1 can be used if you wish to save the output in your workfile.

2.4 Displaying correlation and covariance matrices

The easiest way to display correlation and covariance matrices is to highlight the relevant series (using the mouse and the [Ctrl] key) and then click **Quick → Group Statistics → Correlations** (or **Covariances** if you want a covariance matrix). This creates a new group and produces a *common sample* correlation/covariance matrix. If a *pair wise* correlation/covariance matrix is more suitable (see the box below), this is produced by clicking the ***[View]*** button and choosing **Correlations** (or **Covariances**) **→ Pairwise Samples**.

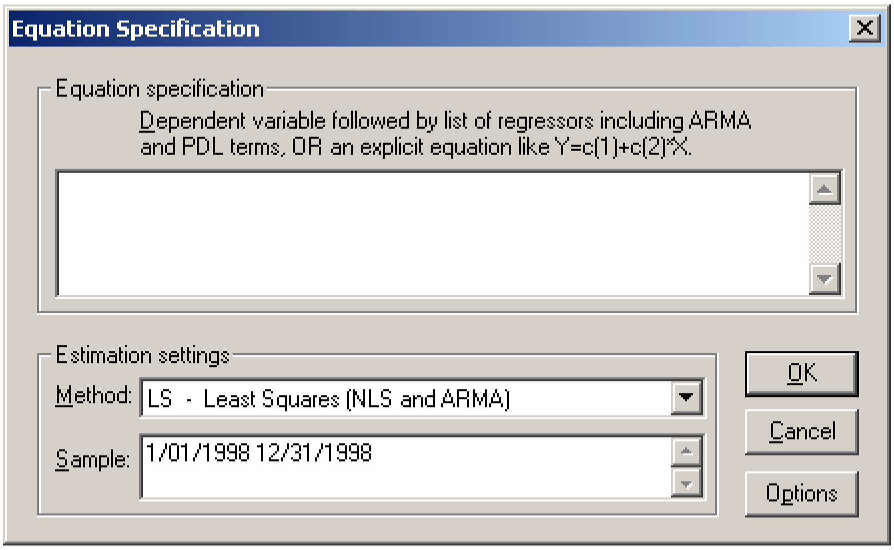
> **NB!!** If you have a matrix with several series/variables you should be aware of any missing variables. If one of the series has missing variables, then <u>all</u> the calculated cross-correlations will be different from obtaining *pair wise* correlations instead!!!

Section 2.1 describes how to save your output in the workfile.

3 Estimating equations

In the following we will demonstrate how you estimate a regression model in EViews. The example is based on the workfile ***X:\Eviews\Eviews_ex1.wf1***.

When you have opened your workfile you click on the ***[Objects]*** button. Select **New Object → Equation** and the following dialog box should appear.



Alternatively, you could have clicked **Quick** → **Estimate Equation**.

Say we want to estimate the regression equation that is illustrated in the scatter plot from section 2.2, i.e. we want *STOCK_A* as the dependent variable and *MAR_IND* as the independent variable.

You can enter the model in two ways (of which the first is probably the quickest):

1) List

First you list the dependent variable followed by *C* for the intercept term and then the independent variable(s). There must be a single space between each variable. In this example we only have a single independent variable, so we will enter the following simple regression into the first window:

STOCK_A C MAR_IND

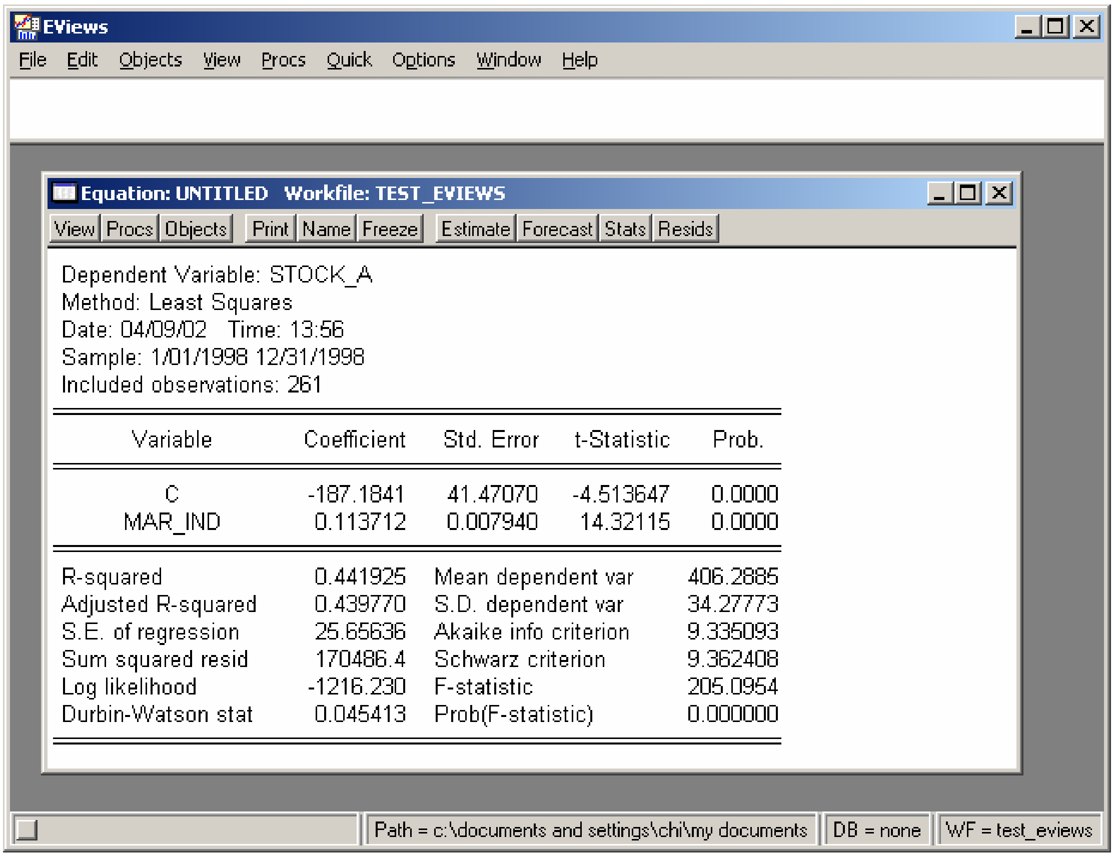
2) Formula

In the same window you type the model as a true equation, where each coefficient is written as *C(k)* with $k = 1, 2, 3, \ldots$. In our simple regression example we would enter:

STOCK_A = C(1) C(2)*MAR_IND

When you have entered your preferred equation you select the estimation method (default is Least Squares) and your sample (default is the entire sample). The ***[Options]*** button gives you the opportunity to change the estimation procedure further. For example you can choose the White correction or the Newey-West correction if the dataset has problems with heteroscedasticity. To do this you simply check the ***[Heteroscedasticity]*** button and select either ***White*** or ***Newey-West***. The help function in EViews provides a more detailed discussion of these corrections.

In this example we will not make any corrections, so we just click ***OK*** and get the following output.



Dependent Variable: STOCK_A
Method: Least Squares
Date: 04/09/02 Time: 13:56
Sample: 1/01/1998 12/31/1998
Included observations: 261

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	-187.1841	41.47070	-4.513647	0.0000
MAR_IND	0.113712	0.007940	14.32115	0.0000

R-squared	0.441925	Mean dependent var	406.2885
Adjusted R-squared	0.439770	S.D. dependent var	34.27773
S.E. of regression	25.65636	Akaike info criterion	9.335093
Sum squared resid	170486.4	Schwarz criterion	9.362408
Log likelihood	-1216.230	F-statistic	205.0954
Durbin-Watson stat	0.045413	Prob(F-statistic)	0.000000

This output is obtained by using the **List** method and is slightly different from the output resulting from the **Formula** method. However, the interpretation is exactly the same.

If you want to save the regression model that you have estimated in your workfile you simply click the ***[Name]*** button above the output. Enter a name and click ***OK***.

It is also possible to re-estimate an equation by clicking the ***[Estimate]*** button. The previous *Equation Specification* dialog box re-appears and you can make any preferred changes. For example you can include more independent variables or exclude any outliers in the *Sample* window.

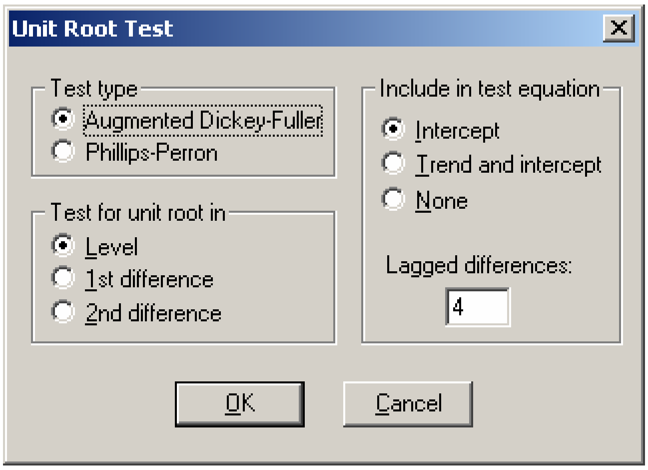
4 Testing for unit roots

Testing for unit roots is useful when you want to know whether a time series is stationary or not. A time series is stationary if and only if $|\rho| < 1$ in the following regression model:

$$X_t = \mu + \rho X_{t-1} + \varepsilon_t$$

Therefore, you have to test whether or not $\rho = 1$. If $\rho \geq 1$ then ρ does not follow a standard distribution and consequently the usual t-test is not valid. The ρ-coefficient will therefore be tested against the so-called Dickey-Fuller distribution, which is used to test for stationary time series.

This test is quickly done in EViews by double clicking the relevant time series in order to go to the spreadsheet view. Here you click the ***[View]*** button and select **Unit root test**. The following dialog box should appear.



In the dialog box you first select the test type. In this example we will use ***Augmented Dickey-Fuller***[5]. Next you select whether you want to test for unit root in:

Level (X_t)
1st difference (X_t-X_{t-1}=ΔX_t)
or ***2nd difference*** (ΔX_t-ΔX_{t-1}).

Furthermore, you are asked whether you want to include other exogenous variables in the test equation. You can choose between the following:

1. $\Delta X_t = \mu + \gamma X_{t-1} + \sum_{i=1}^{k} \delta_i \Delta X_{t-i} + \varepsilon_t$ **(Intercept)**

2. $\Delta X_t = \mu + \gamma X_{t-1} + \beta T + \sum_{i=1}^{k} \delta_i \Delta X_{t-i} + \varepsilon_t$ **(Trend and intercept)**

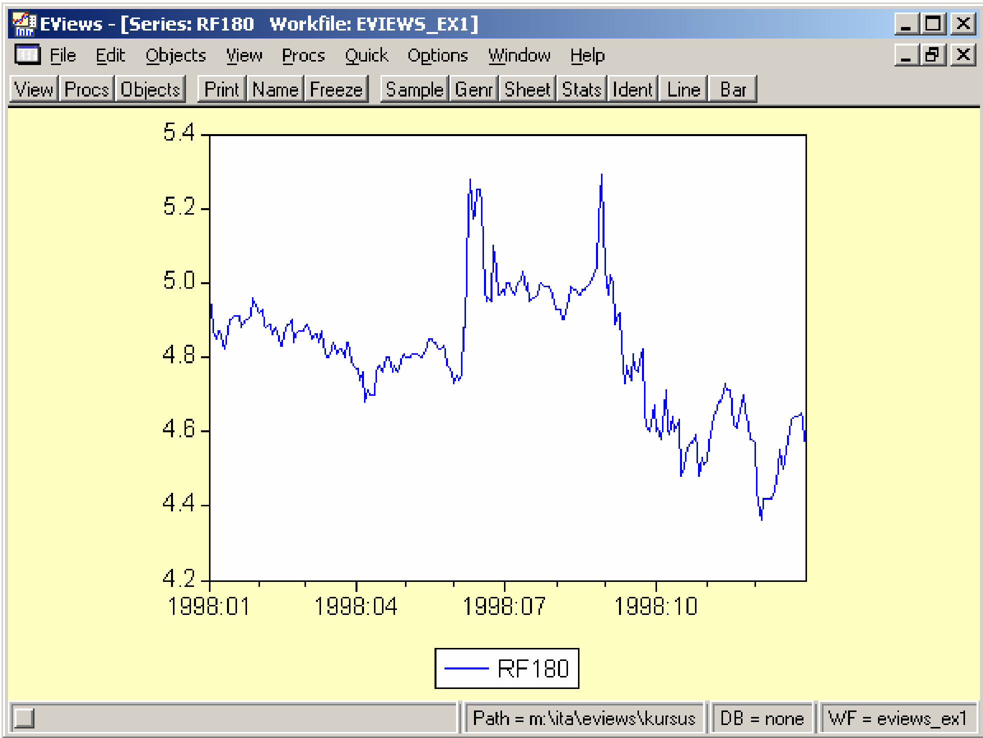
3. $\Delta X_t = \gamma X_{t-1} + \sum_{i=1}^{k} \delta_i \Delta X_{t-i} + \varepsilon_t$ **(None)**

As you can see, a number of lagged first differences of the test variable are included in the test regression. The exact number is specified in the *Lagged differences* window.

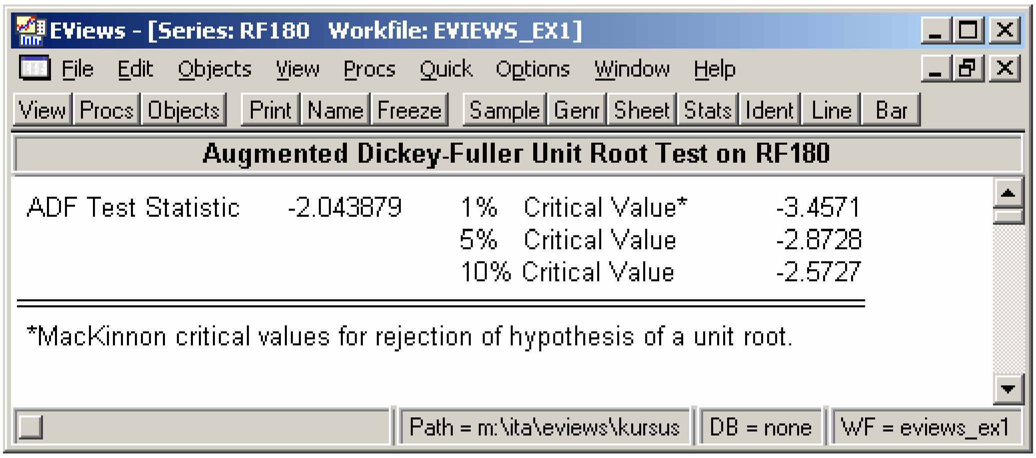
[5] For a short description of the theory behind the Augmented Dickey-Fuller and Phillips_Perron tests you can refer to the help function in EViews, which provides references to more detailed sources as well.

You can decide which of the models to estimate from looking at a plot of ΔX_t. From this plot you estimate if the time series has a mean different from zero (model 1), if there is a trend in the time series (model 2), or if the time series has a mean around zero (model 3).

In the output below you see a plot of the short interest rate in Denmark, *KORT_DK* (The plot is obtained by double clicking the variable, clicking the ***[View]*** button and selecting **Graph** → **Line**).



We will test to see whether the time series is stationary around a mean different from zero, i.e. model 1 above. You can see the output from this test below.



You see the ADF test statistic to the left (-2.043879) and the critical values to the right. The hypotheses are the following:

$H_0 : \rho = 1$ (non-stationary time series)

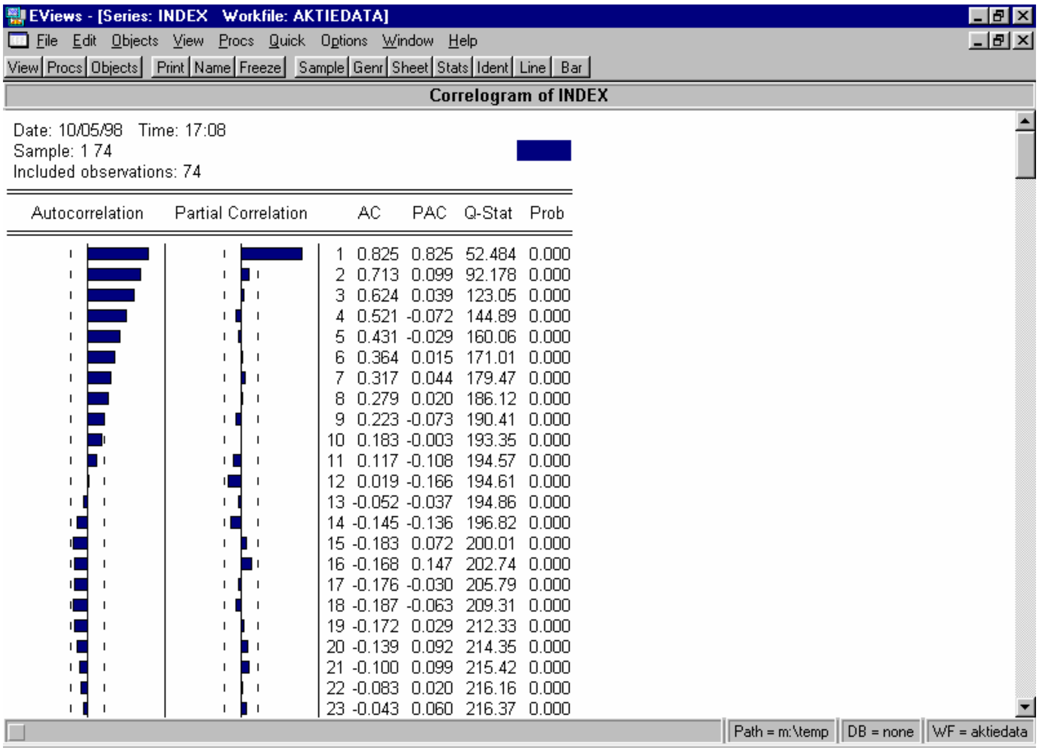
$H_1 : \rho < 1$ (stationary time series)

The null hypothesis of a unit root is rejected against the one-sided alternative if the test statistic is less than (lies to the left of) the critical value[6]. In this example, the test fails to reject the null hypothesis of a unit root in the time series at all three levels of significance. Therefore, it is not possible to reject that the time series is non-stationary.

5 ARIMA identification and estimation

The identification of an ARIMA model is done by examining a correlogram. In EViews you obtain a correlogram for at variable by double clicking the variable to open the spreadsheet view. Here you click the ***[View]*** button and choose **Correlogram**. In the appearing dialog box you choose between level, first difference, or second difference and then you enter the desired number of lags to include. An example of a correlogram for the variable *INDEX* is shown below.

[6] If $\rho > 1$ the series is said to be explosive (and consequently does not make much economic sense). Therefore it is not included in the alternative hypothesis.



	AC	PAC	Q-Stat	Prob
1	0.825	0.825	52.484	0.000
2	0.713	0.099	92.178	0.000
3	0.624	0.039	123.05	0.000
4	0.521	-0.072	144.89	0.000
5	0.431	-0.029	160.06	0.000
6	0.364	0.015	171.01	0.000
7	0.317	0.044	179.47	0.000
8	0.279	0.020	186.12	0.000
9	0.223	-0.073	190.41	0.000
10	0.183	-0.003	193.35	0.000
11	0.117	-0.108	194.57	0.000
12	0.019	-0.166	194.61	0.000
13	-0.052	-0.037	194.86	0.000
14	-0.145	-0.136	196.82	0.000
15	-0.183	0.072	200.01	0.000
16	-0.168	0.147	202.74	0.000
17	-0.176	-0.030	205.79	0.000
18	-0.187	-0.063	209.31	0.000
19	-0.172	0.029	212.33	0.000
20	-0.139	0.092	214.35	0.000
21	-0.100	0.099	215.42	0.000
22	-0.083	0.020	216.16	0.000
23	-0.043	0.060	216.37	0.000

An ARIMA estimation consists of four steps, or the so-called Box-Jenkins methodology for ARIMA models (Madsen (1992), pp. 101-115 & Maddala (1992), pp. 542-543)[7]:

1) Identification
2) Estimation
3) Diagnosis
4) Forecast

In step 1 you compare the estimated autocorrelations for the residuals with the theoretic autocorrelations, which normally arise in a given ARIMA model. If there are any autocorrelations between the residuals, it is possible to set up an ARIMA model. Examining the nature of the correlation between the residuals is useful in determining p and q in the AR (autoregressive) and MA (moving average) components.

[7] Madsen, Henning (1992), *"Forecasting Economic Time Series An Introduction"*, Institut for Informationsbehandling, Handelshøjskolen i Århus.
Maddala, G. S. (1992), *"Introduction to Econometrics"*, 2nd edition, Prentice Hall.

From the correlogram above you see the plots of the serial autocorrelations (SACF) and the partial autocorrelations (PACF). SACF is geometrically declining, but it is not a long, linearly declining sequence. PACF is curtailed after lag 1. From the table below you can see which ARIMA model fits this description.

Identification of ARIMA models using SACF and PACF

Process	SACF	PACF
AR (p)	Geometrically declining	Curtailed after lag p
MA (q)	Curtailed after lag q	Geometrically declining
ARMA (p,q)	Declining after lag q	Declining after lag p
If SACF shows a linearly declining sequence you should consider differencing the series.		

Source: (Madsen (1992), p. 103)

According to the table, this example can be considered an AR(1) model. The SACF and PACF characteristics do not suggest including a MA component or differencing the series.

In step 2 you should estimate the identified ARIMA model. If an ARIMA model includes a MA component you cannot use OLS estimation, as the model is non-linear. Instead, a non-linear least squares estimation must be carried out. In this example there is no MA component and common OLS estimation is therefore correct.
The model for an autoregressive process of order 1 can be written as:

(a) $X_t = \alpha + \varphi_1 X_{t-1} + \varepsilon_t$

Model (a) can be transformed into:

(b) $X_t - \mu_0 = \mu + \varphi_1 (X_{t-1} - \mu_0) + \varepsilon_t$

Model (b) suggests that the time series is fluctuating around a mean. EViews allows you to parameterise such a model yourself by using the command line interface (see next section).

6 Using EViews commands

So far we have only looked at the Windows interface of EViews. However, EViews provides you with a command line interface as well. This is useful if you want to create macros or work with matrices for example. Commands may be used interactively, or executed in batch mode. Virtually every operation that can be accomplished using menus may also be entered into the command window of EViews or placed in programs for batch processing. For specific commands you can refer to the EViews help function.

6.1 Interactive Use

To work interactively, you should type a command into the command window, then press ENTER to execute the command immediately. If you enter an incomplete command, EViews will open a dialog box prompting you for additional information.

The command window is located just below the main menu bar at the top of the EViews window. The blinking vertical insertion bar at the left end of the command window indicates that EViews is expecting a command.

A command that you enter in the window will be executed as soon as you press *ENTER*. The insertion point need not be at the end of the command line when you press *ENTER*. EViews will execute the entire line containing the insertion point.

As you enter commands, EViews will create a list in the command window. You can scroll up to an earlier command, edit it, and hit *ENTER*. The modified command will be executed again. You may also use standard Windows copy-and-paste between the command window and any other window. The contents of the command area may also be saved directly into a text file for later use. To do this you first make certain that the command window is active by clicking anywhere in the window, and then select **File→ Save As…** from the main menu.

6.2 Batch Program Use

You can assemble a number of commands into a program, and then execute the commands in batch mode. Each command in the program will be executed in the order that it appears in the program. Using batch programs allows you to make use of macro processing for example.

One way to create a program file is to select **File → New → Program**. EViews will open an untitled program window into which you may enter your commands. You can save the program by clicking on the ***[Save]*** or ***[SaveAs]*** button, browsing to the desired directory, and entering a file name. The file name you enter will be given the extension '.prg'.

Alternatively, you can use any (ASCII) editor to create a program file containing your commands. The commands in this program may then be executed in EViews.

7 Working with matrices

EViews provides you with tools for working directly with data contained in matrices and vectors. To make use of these tools it is required that you are familiar with the use of EViews commands, which was discussed in section 7.

The following six objects (all called matrix objects) can be created and manipulated using the matrix command language:

- **Coef:** column vector of coefficients to be used by Equation, System, Pool, and Sspace objects
- **Matrix:** two-dimensional array
- **Rowvector:** row vector
- **Scalar:** scalar
- **Sym:** symmetric matrix (stored in lower triangular form)
- **Vector:** column vector

7.1 Declararing matrix objects

All matrix objects must be *declared* prior to use. This declaration consists of the object keyword followed either by size information (in parentheses) and the name to be given to the object, or by the name and an assignment statement (e.g. *matrix_a = matrix_z*).

The various matrix objects require different sizing information. A matrix requires the number of rows and the number of columns. A sym requires that you specify a single number representing both the number of rows and the number of columns. A vector, rowvector, or coef declaration requires the number of elements. Of course a scalar requires no size information. If no size information is provided, EViews will assume that there is only one element in the object. If no assignment statement is included in the declaration, EViews will initialize all values to zero.

Example:
Look at the following commands:

```
matrix(3,10) mat_a
sym(9) sym_x
vector(11) vec_y
rowvector(5) rvec_z
```

These four commands create a 3x10 matrix called *mat_a*, a symmetric 9x9 matrix *sym_x*, an 11x1 column vector *vec_y*, and a 1x5 row vector *rvec_z*. All of these objects are initialized to zero, since they have been given no assignment statements.

To change the size of a matrix object, you can repeat the declaration statement with the new size information. Furthermore, if you use an assignment statement with an existing matrix object, the target will be resized as necessary. Consider this example.

Example:
Writing the commands

sym(10) big_z
matrix mat_a
matrix(10,2) mat_a
mat_a = big_z10

will first declare *mat_a* to be a matrix with a single element, and then redeclare *mat_a* to be a 10x2 matrix. The assignment statement in the last line will resize *mat_a* so that it holds the contents of the 10x10 symmetric matrix *big_z*.

7.2 Assigning Matrix Values

There are three ways to assign values to the elements of a matrix. In the following only one of them will be demonstrated; the method of fill assignment. This should be sufficient for most operations, but you can refer to the EViews help function if you would like to know about the other two methods.

The procedure involves assigning a list of numbers to each element of the matrix in the specified order. By default, the procedure fills the matrix column by column, but you may override this behavior. You should enter the name of the matrix object, followed by a period, the ***fill*** keyword, and then a comma delimited list of values.

Example:
The commands

vector(3) v
v.fill 0.1, 0.2, 0.3
matrix(2,4) x
x.fill 1, 2, 3, 4, 5, 6, 7, 8

create the matrix objects

$$V = \begin{pmatrix} 0.1 \\ 0.2 \\ 0.3 \end{pmatrix} \quad \text{and} \quad X = \begin{pmatrix} 1 & 3 & 5 & 7 \\ 2 & 4 & 6 & 8 \end{pmatrix}$$

If we replace the last line with

x.fill(b=r) 1,2,3,4,5,6,7,8 (default is b=c, where c is colums) then X is given by

$$X = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \end{pmatrix}$$

In some situations, you may wish to repeat the assignment over a list of values. You may use the ***l*** option to fill the matrix by repeatedly looping through the listed numbers until the matrix elements are exhausted. Thus,

Example:
The commands

matrix(3,3) y
y.fill(l) 1, 0, -1

will create the matrix

$$Y = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{pmatrix}$$

7.3 Matrix Operators

EViews provides standard mathematical operators for matrix objects. With the exception of comparison operators[8], these will be discussed below.

Negation

The unary minus changes the sign of every element of a matrix object, yielding a matrix or vector of the same dimension. If you for example want to change the sign of the elements from a matrix, *pos_x*, you create the new matrix, *neg_x*, by typing:

matrix neg_x = -pos_x

[8] Comparison operators are: =, >, >=, <, <=, <>. You can refer to the EViews help function for details.

Addition and subtraction

You can add or subtract two matrix objects of the same type and size by using the procedure in the following example.

Example:
If we consider the two matrices, *a* and *b*, created by the commands

matrix(3,4) a
matrix(3,4) b

then we simply add them by creating a new matrix called *sum*:

matrix sum = a+b

If you want to subtract the two matrices instead you just create another matrix:

matrix diff = a–b

As you can see we have called this new matrix *diff*.

You can add a square matrix and a sym of the same dimension. The upper triangle of the sym is taken to be equal to the lower triangle. Adding a scalar to a matrix object adds the scalar value to each element of the matrix or vector object. Similarly, subtracting a scalar object from a matrix object subtracts the scalar value from every element of the matrix object.

Multiplication

You can multiply two matrix objects if the number of columns of the first matrix is equal to the number of rows of the second matrix.

Example:

matrix(5,9) a
matrix(9,22) b
*matrix prod = a*b*

The two matrices, *a* and *b*, are multiplied and the result is contained in the new matrix, *prod*, which will have 5 rows and 22 columns.

You use the same procedure when multiplying vectors, rowvectors, scalars and combinations of all matrix objects. Another example can be considered.

Example:

rowvector(4) rv_a
rv_a.fill 2,3,0,1
matrix(4,4) mat_z
mat_z.fill 2,1,0,0,2,1,2,1,1,0,0,2,0,1,3,1
*rowvector rv_res = rv_a*mat_z*

Here the rowvector, *rv_a*, is multiplied with the matrix, *mat_z*. This should result in the following rowvector:

$$\text{RV\_RES} = \begin{pmatrix} 7 & 8 & 4 & 4 \end{pmatrix}$$

Division

You can divide a matrix object by a scalar. Consider the following example.

Example:

m*atrix(2,3) orig*
orig.fill 2,4,8,18,6,4
matrix div = orig/2

Each element of the matrix, *orig*, will be divided by 2. Thus we get the following matrix:

$$DIV = \begin{pmatrix} 1 & 4 & 3 \\ 2 & 9 & 2 \end{pmatrix}$$

7.4 Matrix commands and functions

EViews provides a number of commands and functions that allow you to work with the contents of your matrix objects. Each is well described in the EViews help function, and we will therefore only demonstrate a few basic ones in the following.

7.4.1 Transposing a matrix object

The syntax *@transpose(x)* forms the transpose of a matrix object, *x*.

Example:
The following syntax

```
matrix(2,3) mat_a
mat_a.fill 4,2,5,7,0,4
matrix tr_a = @transpose(mat_a)
```

first creates the matrix, *mat_a*, and returns the transpose of *mat_a* in a new matrix:

$$TR\_A = \begin{pmatrix} 4 & 2 \\ 5 & 7 \\ 0 & 4 \end{pmatrix}$$

7.4.2 Calculating the rank

The syntax *@rank(x, n)* returns the rank of a matrix object, *x*. The rank is calculated by counting the number of singular values of the matrix which are smaller in absolute value than the tolerance, which is given by the argument *n*. If n is not provided, EViews uses the value given by the largest dimension of the matrix multiplied by the norm of the matrix multiplied by machine epsilon (the smallest representable number).

Example:

```
matrix(2,2) mat_x
mat_x.fill 1,2,1,2
scalar rank_a = @rank(mat_x)
```

Here the rank of *mat_x* is calculated and returned as a scalar, *rank_a*. You can view the value of this scalar by double clicking it in your workfile. The value should appear in the bottom left corner, and in this case it should equal 1.

Other basic commands include:

- @trace (returning the trace of a square matrix or sym)

- @det (returning the determinant of a square matrix or sym)
- @inverse (returning the inverse of a square matrix or sym)
- @identity (creates an identity matrix)
- etc.