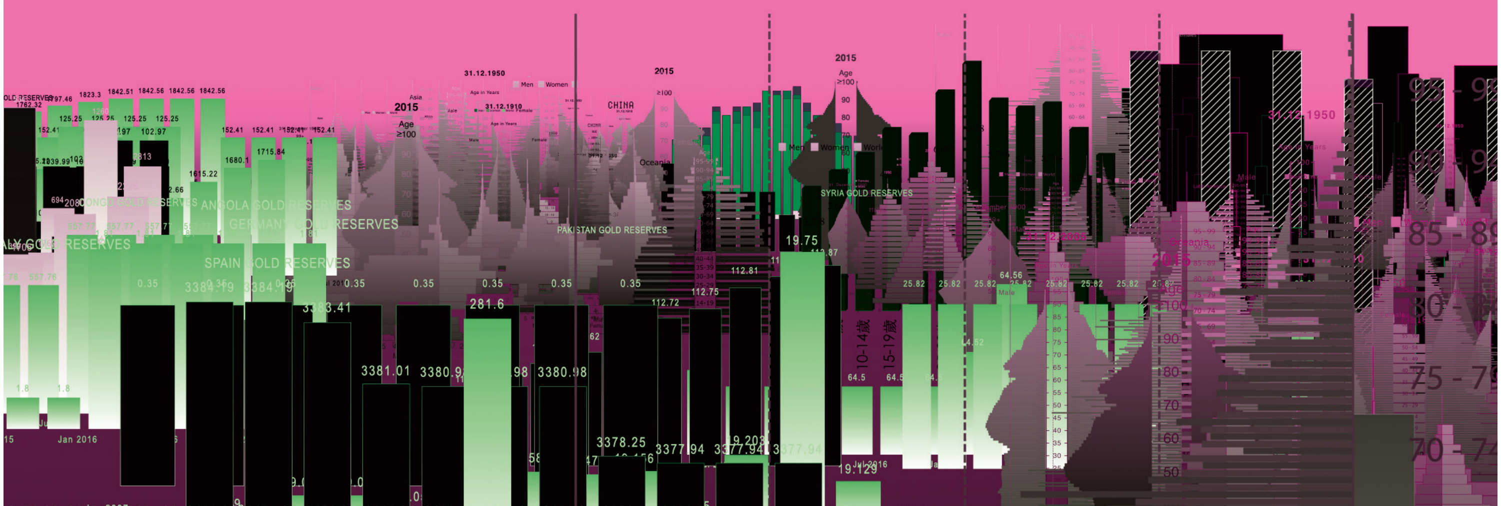
CHRIS BROOKS
ROBERT WICHMANN



EVIEWS GUIDE TO ACCOMPANY

INTRODUCTORY ECONOMETRICS FOR FINANCE

4TH EDITION

This guide draws on material from 'Introductory Econometrics for Finance', published by Cambridge University Press, © Chris Brooks (2019). The Guide is intended to be used alongside the book, and page numbers from the book are given after each section and subsection heading.

The authors accept no responsibility for the persistence or accuracy of URLs for external or third-party internet websites referred to in this work, and nor do we guarantee that any content on such web sites is, or will remain, accurate or appropriate.

Contents

List of Figures

List of Tables

1 An Introduction to EViews

The number of packages available for econometric modelling is large, and over time, all packages have improved in breadth of available techniques, and have also converged in terms of what is available in each package. The programs can usefully be categorised according to whether they are fully interactive (menu-driven), command-driven (so that the user has to write mini-programs) or somewhere in between. Menu-driven packages, which are usually based on a standard Microsoft Windows graphical user interface, are almost certainly the easiest for novices to get started with, for they require little knowledge of the structure of the package, and the menus can usually be negotiated simply. EViews is a package that falls into this category.

On the other hand, some such packages are often the least flexible, since the menus of available options are fixed by the developers, and hence if one wishes to build something slightly more complex or just different, then one is forced to consider alternatives. EViews, however, has a command-based programming language as well as a click-and-point interface so that it offers flexibility as well as user-friendliness. As for previous editions of this book, sample instructions and output for the EViews package will be given.

EViews is a simple to use, interactive econometrics software package providing the tools most frequently used in practical econometrics. It is built around the concept of objects with each object having its own window, its own menu, its own procedure and its own view of the data. Using menus, it is easy to change between displays of a spreadsheet, line and bar graphs, regression results, etc. One of the most important features of EViews that makes it useful for model-building is the wealth of diagnostic (misspecification) tests, that are automatically computed, making it possible to test whether the model is econometrically valid or not. You work your way through EViews using a combination of windows, buttons, menus and sub-menus. A good way of familiarising yourself with EViews is to learn about its main menus and their relationships through the examples given in this and subsequent sections.

This section assumes that readers have obtained a licensed copy of EViews 10 (the latest version available at the time of writing), and have successfully loaded it onto an available computer. There now follows a description of the EViews package, together with instructions to achieve standard tasks and sample output. Any instructions that must be entered or icons to be clicked are illustrated throughout this book by **bold-faced type**. The objective of the treatment in this and subsequent sections is not to demonstrate the full functionality of the package, but rather to get readers started quickly and to explain how the techniques are implemented and how the results may be interpreted. For further details, readers should consult the software manuals in the first instance, which are now available electronically with the software as well as in hard copy.[1] Note that EViews is not case-sensitive, so that it does not matter whether commands are entered as lower-case or CAPITAL letters.

[1] A student lite edition of EViews 10 is available for free, but with restrictions on the number of observations and objects that can be included in each saved workfile.

2 Data Management

2.1 Importing Data

To load EViews from Windows, click the **Start** button, then **All Programs**, **EViews10** and finally, **EViews10** again. EViews provides support to read from or write to various file types, including 'ASCII' (text) files, Microsoft Excel '.XLS' and '.XLSX' files (reading from any named sheet in the Excel workbook), Lotus '.WKS1' and '.WKS3' files. It is usually easiest to work directly with Excel files, and this will be the case throughout this guide.

The first step when the EViews software is opened is to create a *workfile* that will hold the data. To do this, select **New** from the File menu. Then choose **Workfile**. The 'Workfile Create' window in Figure 1 will be displayed.

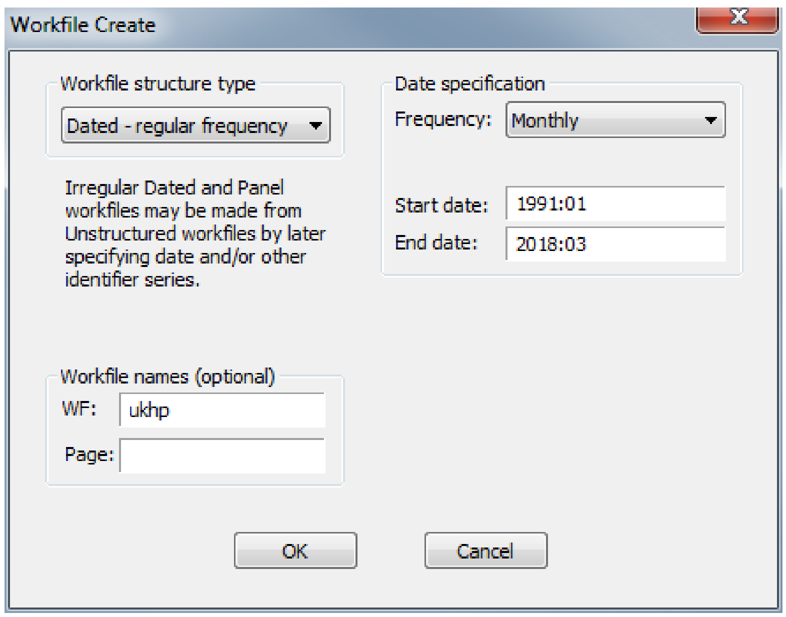


Figure 1: Creating a Workfile

We are going to use as an example a time series of UK average house price data obtained from Nationwide, which comprises 269 monthly observations from January 1991 to March 2018.

Under 'Workfile structure type', keep the default option, **Dated – regular frequency**. Then, under 'Date specification', choose **Monthly**. Note the format of date entry for monthly and quarterly data: YYYY:M and YYYY:Q, respectively. For daily data, a US date format must usually be used depending on how EViews has been set up: MM/DD/YYYY (e.g., 03/01/1999 would be 1st March 1999, not 3rd January). Caution therefore needs to be exercised here to ensure that the date format used is the correct one. Type the start and end dates for the sample into the boxes: **1991:01** and **2018:03**, respectively. Then click **OK**. The workfile will now have been created.

Note that two pairs of dates are displayed, 'Range' and 'Sample': the first one is the range of dates contained in the workfile and the second one (which is the same as above in this case) is for the current workfile sample. Two objects are also displayed: C (which is a vector that will eventually contain the parameters of any estimated models) and RESID (a residuals series, which will currently be empty). All EViews workfiles will contain these two objects, which are created automatically.

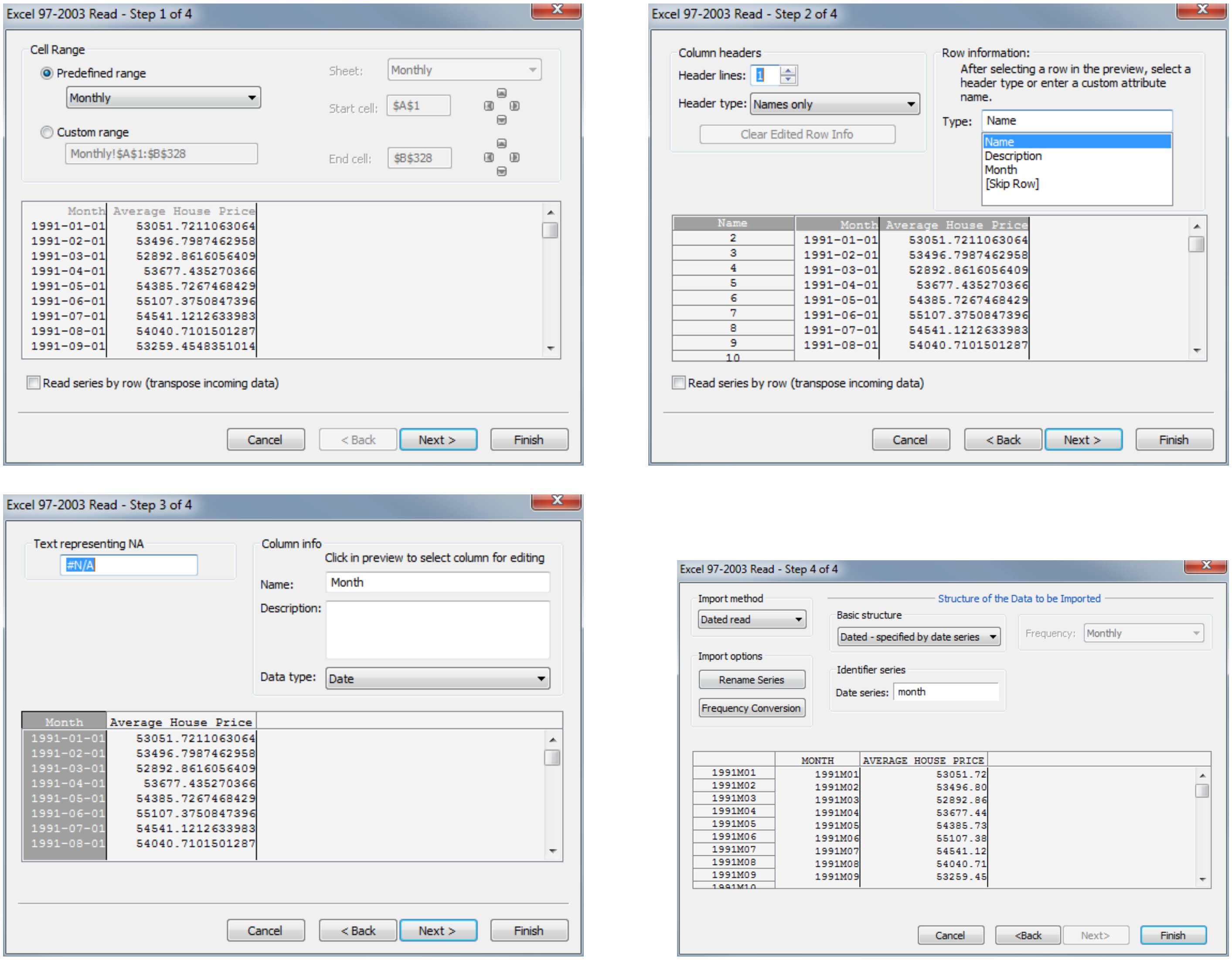


Figure 2: Importing Excel Data into the Workfile

Now that the workfile has been set up, we can import the data from the Excel file UKHP.XLS. So from the File menu, select **Import** and **Import from File**. You will then be prompted to select the directory and file name. Once you have found the directory where the file is stored, enter **UKHP.XLS** in the 'file name' box and click **Open**. You are then faced with a series of four screens where it is possible to modify the way that the data are imported. Most of the time it is not necessary to change any of the default options as EViews peeks inside the data file and identifies the structure of the data, whether there is a header row containing the names of the series, etc. The four screens are shown in Figure 2. In the fourth screen, click **Rename Series** and in the box that appears, type **AVERAGE_HOUSE_PRICE HP** and this will change the name of the series to 'HP', which is a bit easier to deal with!

Click **Finish** and the series will be imported.[2] The series will appear as a new icon in the workfile window, as in Figure 3. Note that EViews has sensibly not imported the column of dates as if it were an additional variable.

[2]After you click **Finish**, you will be asked whether to link the imported series to the actual file 'UKHP.xls'. This might be useful, if you want to change the source file throughout. In our case it is not necessary, so we click **No**.

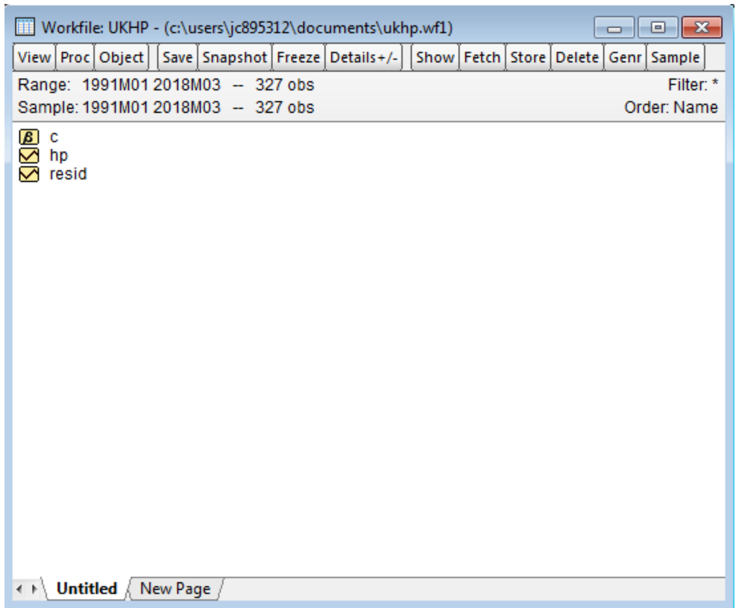


Figure 3: Workfile Containing Loaded Data

Double click on the new hp icon that has appeared, and this will open up a spreadsheet window within EViews containing the monthly house price values. Make sure that the data file has been correctly imported by checking a few observations at random.

The next step is to save the workfile: click on the **Save As** button from the **File** menu. A save dialog box will open, prompting you for a workfile name and location. You should enter XX (where XX is your chosen name for the file), then click **OK**. EViews will save the workfile in the specified directory with the name XX.wf1. I have called my file 'ukhp.wf1' You will also be prompted to select whether the data in the file should be saved in 'single precision' or 'double precision'. The latter is preferable for obvious reasons unless the file is likely to be very large because of the quantity of variables and observations it contains (single precision will require less space) so just click **OK**. The saved workfile can be opened later by selecting File/Open/EViews Workfile... from the menu bar.

Variables of interest can be created in EViews by selecting the *Genr* button from the workfile toolbar and typing in the relevant formulae. Suppose, for example, we have a time series called Z. The latter can be modified in the following ways so as to create variables A, B, C, etc. The mathematical background and simple explanations of these transformations, including powers, logarithms and exponents, will be discussed in detail in the following section. Some common operations and functions are summarised in Table 1.

Table 1: Commonly Used Functions and Operations

Syntax	Description	Syntax	Description
Z/2	divide Z by 2	Z(-1)	first lag of Z
Z*2	multiply Z by 2	Z(-n)	n^{th} lag of Z
Z^2	square of Z	d(Z)	first difference of Z (equivalent to $Z - Z(-1)$)
sqr(Z)	square root of Z	d(Z,n)	n^{th} difference of Z (equivalent to $Z - Z(-n)$)
log(Z)	natural logarithm of Z	dlog(Z)	first difference of the log of Z
exp(Z)	exponential function of Z	abs(Z)	absolute value of Z

Other functions that can be used in the formulae include: *abs*, *sin*, *cos*, etc. Notice that no special instruction is necessary; simply type 'new variable = function of old variable(s)'. The variables will be displayed in the same workfile window as the original (imported) series.

In this case, it is of interest to calculate simple percentage changes in the series. Click **Genr** and type **DHP = 100∗(HP-HP(-1))/HP(-1)**. It is important to note that this new series, DHP, will be a series of monthly percentage changes and will not be annualised.

2.2 Summary Statistics

Descriptive summary statistics of a series can be obtained by selecting **Quick/Series Statistics/Histogram and Stats** and typing in the name of the variable (**DHP**). The view in Figure 4 will be displayed in the window.

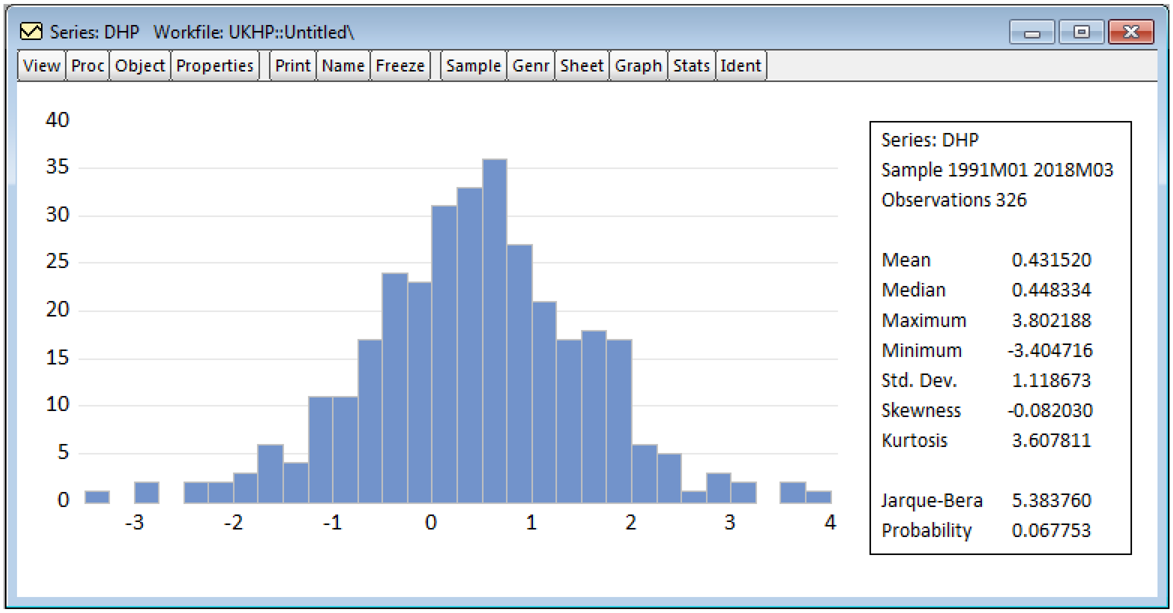


Figure 4: Histogram and Summary Statistics for a Series

As can be seen, the histogram suggests that the series has a slightly longer upper tail than lower tail (note the x-axis scale) and is centred slightly above zero. Summary statistics including the mean, maximum and minimum, standard deviation, higher moments and a test for whether the series is normally distributed, are all presented. Interpreting these will be discussed in subsequent sections. Other useful statistics and transformations can be obtained by selecting the command *Quick/Series Statistics*, but these are also covered later in this guide.

EViews supports a wide range of graph types including line graphs, bar graphs, pie charts, mixed line–bar graphs, high–low graphs and scatter plots. A variety of options permits the user to select the line types, colour, border characteristics, headings, shading and scaling, including logarithmic scale and dual scale graphs. Legends are automatically created (although they can be removed if desired), and customised graphs can be incorporated into other Windows applications using copy-and-paste, or by exporting as Windows metafiles.

From the main menu, select **Quick/Graph** and type in the name of the series that you want to plot (**HP** to plot the level of house prices) and click **OK**. You will be prompted with the 'Graph Options' window where you choose the type of graph that you want (line, bar, scatter or pie charts, etc.) and also control the layout and style of the graph (e.g., whether you want a legend, axis labels, etc.). Choosing a line and symbol graph would produce Figure 5.

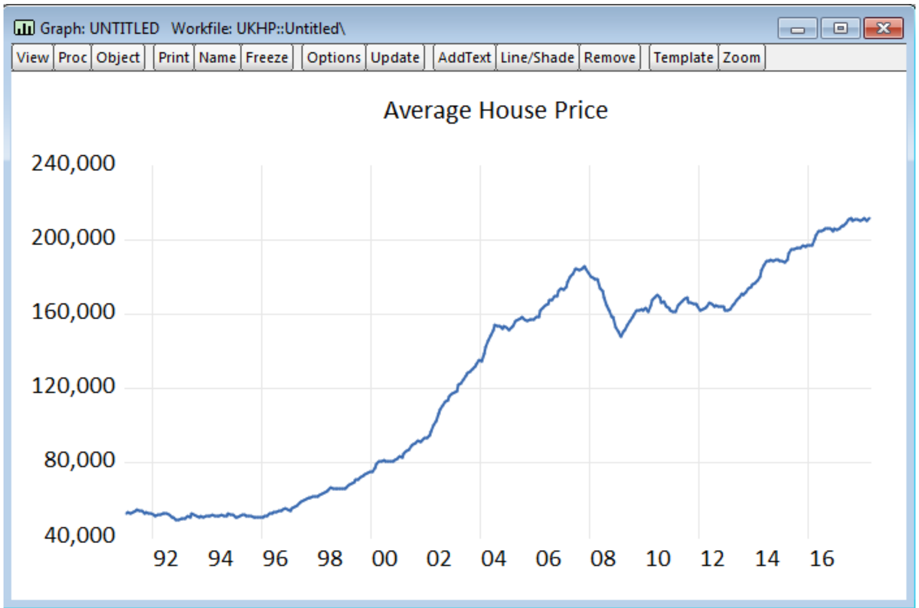


Figure 5: A Line Graph

It is always useful to plot any series you are working with to get a feel for the basic features of the data. It is clear that in this case house prices appreciated quickly to reach a peak in October 2007 before falling sharply until early 2009, after which they increased again. It is possible to identify any value on the chart and its timing by simply hovering the mouse over it. Double-clicking on the graph will revert back to the Graph Options menu.

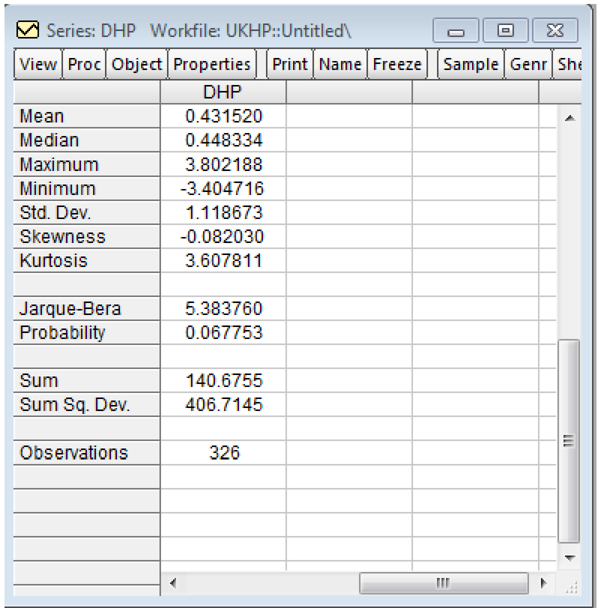
As an exercise, try **plotting the DHP** series – you will see that the volatility of percentage change series makes their graphs much harder to interpret, even though they are usually the form of the data that we work with in econometrics.

Results can be printed at any point by selecting the *Print* button on the object window toolbar. The whole current window contents will be printed. Graphs can be copied into the clipboard if desired by right clicking on the graph and choosing *Copy to clipboard*.

Data generated in EViews can be exported to other Windows applications, e.g., Microsoft Excel. From the main menu, select *File/Save As*. After specifying the data type as *Excel 97-2003 file* and providing a name for the exported file, the next window will ask you to select all the series that you want to export, together with the sample period.

Assuming that the workfile has been saved after the importation of the data set (as mentioned above), additional work can be saved by just selecting *Save* from the *File* menu. The workfile will be saved including all objects in it – data, graphs, equations, etc. *so long as they have been given a title*. Any untitled objects will be lost upon exiting the program.

We will now re-use the summary statistics of the returns (the percentage changes of the house prices). So **re-open the house price EViews workfile** and **double click on the DHP series** to bring up the spreadsheet view. Then **click View/Descriptive Statistics & Tests/Stats Table** to see Figure 6 containing some simple summary statistics. We can see that the mean house price return is around 0.43% per month while the median is slightly larger at 0.45%. The highest monthly price increase was 3.66%, while the biggest fall was 3.80%. The standard deviation is 1.12%, which is quite small compared with stocks (see the next section) and reflects the smoothness of house prices over time. The series has a negative skew so it has a slightly longer lower tail than the upper tail. The series is also leptokurtic and so has fatter tails than a normal distribution with the same mean and variance; there are a total of 326 return observations. EViews also tells us the result of a Jarque–Bera test on whether the series shows significant departures from normality. With a p-value of 0.0678, in this case the null hypothesis of a zero skewness and zero kurtosis is rejected at the 10% level.



Series: DHP Workfile: UKHP::Untitled\

View | Proc | Object | Properties | Print | Name | Freeze | Sample | Genr | She

	DHP
Mean	0.431520
Median	0.448334
Maximum	3.802188
Minimum	-3.404716
Std. Dev.	1.118673
Skewness	-0.082030
Kurtosis	3.607811
Jarque-Bera	5.383760
Probability	0.067753
Sum	140.6755
Sum Sq. Dev.	406.7145
Observations	326

Figure 6: Sample Summary Statistics

If we wanted to calculate less well known statistics like the interquartile range, we need to type the commands directly in the *Command window*. The interquartile range can be computed as the difference between the 0.75 and 0.25 quantile. We can store this value into a scalar **IQR** by typing into the command window

scalar IQR = @quantile(dhp,0.75) - @quantile(dhp,0.25)

and a new variable that only contains one number will appear in the workfile. Double clicking on the new scalar **IQR** will tell us the interquartile range is equal to 1.4089. Using the functions in Table 1 we could also compute other values like this.

3 Linear Regression – Estimation of an Optimal Hedge Ratio

Reading: Brooks (2019, Section 3.3)

This section shows how to run a bivariate regression using EViews. The example considers the situation where an investor wishes to hedge a long position in the S&P500 (or its constituent stocks) using a short position in futures contracts. Many academic studies assume that the objective of hedging is to minimise the variance of the hedged portfolio returns. If this is the case, then the appropriate hedge ratio (the number of units of the futures asset to sell per unit of the spot asset held) will be the slope estimate (i.e., $\hat{\beta}$) in a regression where the dependent variable is a time series of spot returns and the independent variable is a time series of futures returns.

This regression will be run using the file 'SandPhedge.xls', which contains monthly returns for the S&P500 index (in column 2) and S&P500 futures (in column 3). As described in section 2, the first step is to open an appropriately dimensioned workfile. **Open EViews** and click on **File/New/Workfile**; choose **Dated regular frequency** and **Monthly** frequency data. The start date is **2002:02** and the end date is **2013:04**. Then import the Excel file by clicking **File/Import** and **Import from file**. As for the previous example in section 2, the first column contains only dates which we do not need to read in so click **Next** twice. You will then be prompted with another screen as shown in Figure 7 that invites you to decide how to deal with the dates – it is possible either to read the dates from the file or to use the date range specified when the workfile was set up. Since there are no missing data points in this case the two would give the same outcome so just click on **Finish**. The two imported series will now appear as objects in the workfile (the column of dates has not been imported) and can be verified by checking a couple of entries at random against the original Excel file.

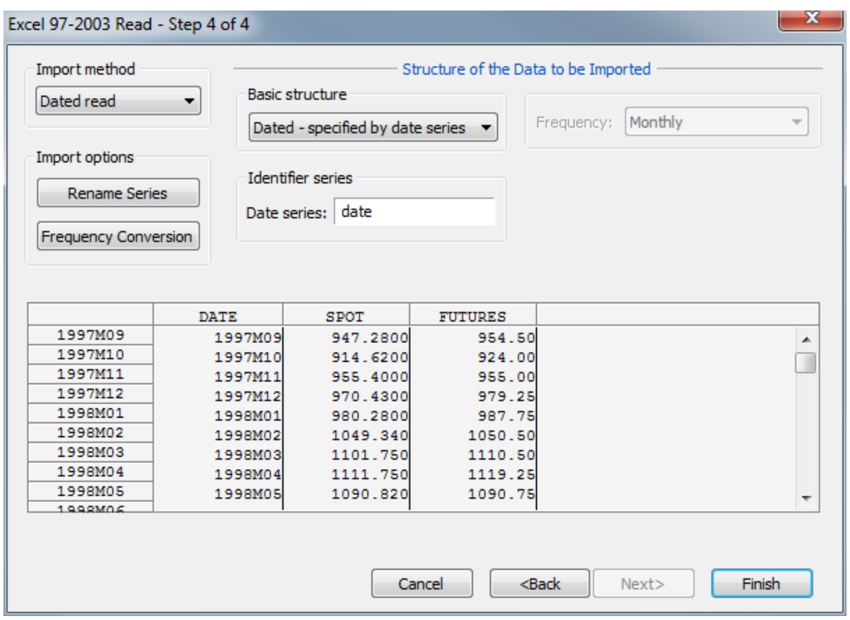


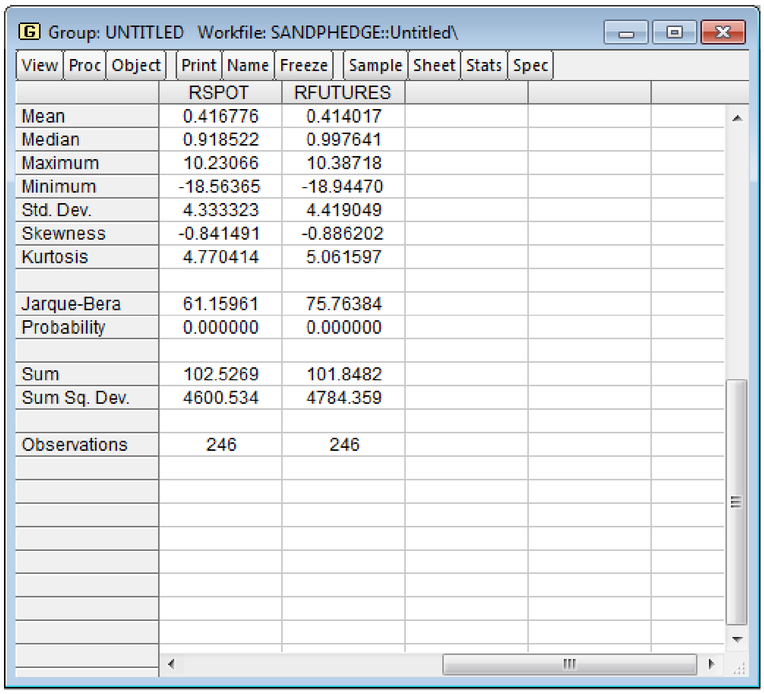
Figure 7: How to Deal with Dated Observations

The first step in the analysis is to transform the levels of the two series into percentage returns. It is common in academic research to use continuously compounded returns rather than simple returns. To achieve this (i.e., to produce continuously compounded returns), click on **Genr** and in the 'Enter Equation' dialog box, enter **rfutures=100*dlog(futures)**. Then click **Genr** again and do the same for the spot series: **rspot=100*dlog(spot)**.[3] Do not forget to **Save the workfile** – call it 'Sandphedge'

[3]Note that we use the function **dlog** here as it creates the first difference of the logarithm. This is equivalent to typing

and EViews will add the suffix '.wf1' to denote that it is an EViews workfile. Continue to re-save it at regular intervals to ensure that no work is lost.

Before proceeding to estimate the regression, now that we have imported more than one series, we can examine a number of descriptive statistics together and measures of association between the series. For example, click **Quick** and **Group Statistics**. From there, you will see that it is possible to calculate the covariances or correlations between series and a number of other measures that will be discussed later in this guide. For now, click on **Descriptive Statistics** and **Common Sample**.[4] In the dialog box that appears, type **rspot rfutures** and click **OK**. Some summary statistics for the spot and futures are presented, as displayed in Figure 8, and these are quite similar across the two series, as one would expect.



	RSPOT	RFUTURES
Mean	0.416776	0.414017
Median	0.918522	0.997641
Maximum	10.23066	10.38718
Minimum	-18.56365	-18.94470
Std. Dev.	4.333323	4.419049
Skewness	-0.841491	-0.886202
Kurtosis	4.770414	5.061597
Jarque-Bera	61.15961	75.76384
Probability	0.000000	0.000000
Sum	102.5269	101.8482
Sum Sq. Dev.	4600.534	4784.359
Observations	246	246

Figure 8: Summary Statistics for Spot and Futures Returns

Note that the number of observations has reduced from 247 for the levels of the series to 246 when we computed the returns (as one observation is 'lost' in constructing the $t-1$ value of the prices in the returns formula). If you want to save the summary statistics, you must name them by clicking **Name** and then choose a name, e.g., **Descstats**. The default name is 'group01', which could have also been used. Click **OK**.

We can now proceed to estimate the regression. There are several ways to do this, but the easiest is to select **Quick** and then **Estimate Equation**. You will be presented with a dialog box, which, when it has been completed, will look like Figure 9, left panel.

rspot = 100*(log(spot)-log(spot(-1))) and respectively for futures.

[4]'Common sample' will use only the part of the sample that is available for all the series selected, whereas 'Individual sample' will use all available observations for each individual series. In this case, the number of observations is the same for both series and so identical results would be observed for both options.

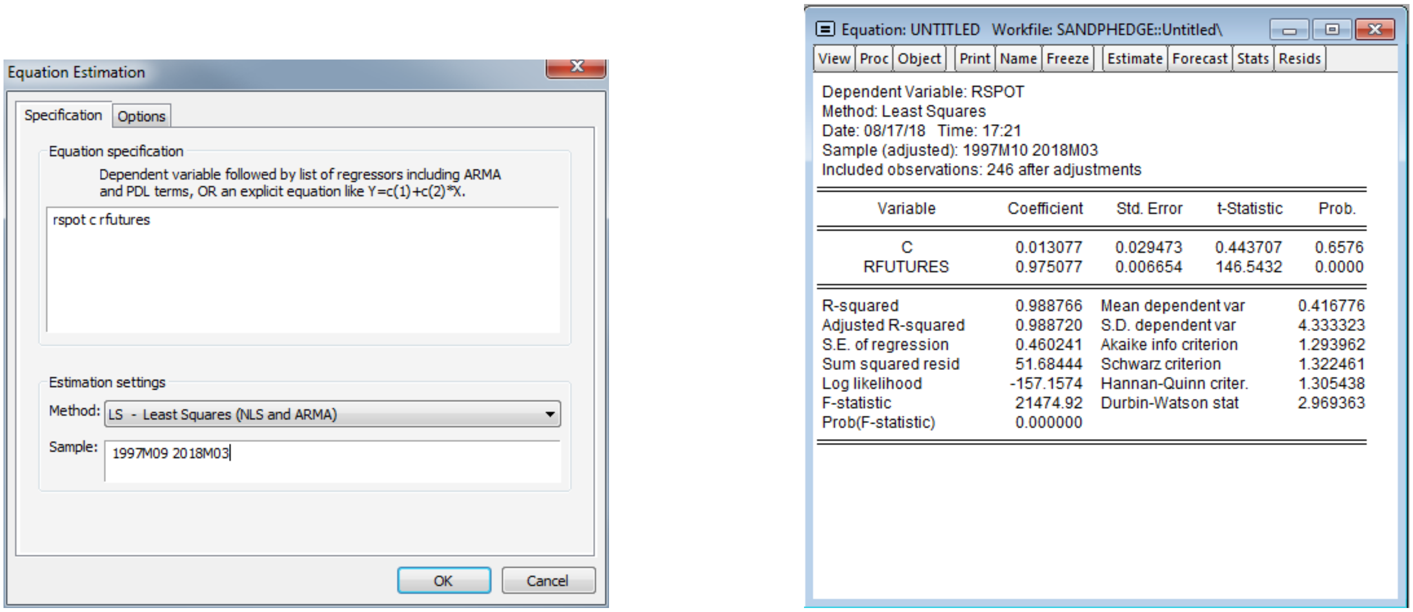


Figure 9: Equation Estimation Window

In the 'Equation Specification' window, you insert the list of variables to be used, with the dependent variable (y) first, and including a constant (c), so type **rspot c rfutures**. Note that it would have been possible to write this in an equation format as rspot = c(1) + c(2)*rfutures, but this is more cumbersome.

In the 'Estimation settings' box, the default estimation method is OLS and the default sample is the whole sample, and these need not be modified. Click **OK** and the regression results will appear, as in Figure 9, right panel.

The parameter estimates for the intercept ($\hat{\alpha}$) and slope ($\hat{\beta}$) are 0.013 and 0.975, respectively. Name the regression results **returnreg**, and it will now appear as a new object in the list. A large number of other statistics are also presented in the regression output – the purpose and interpretation of these is discussed extensively in Brooks (2019, Chapters 3 and 4).

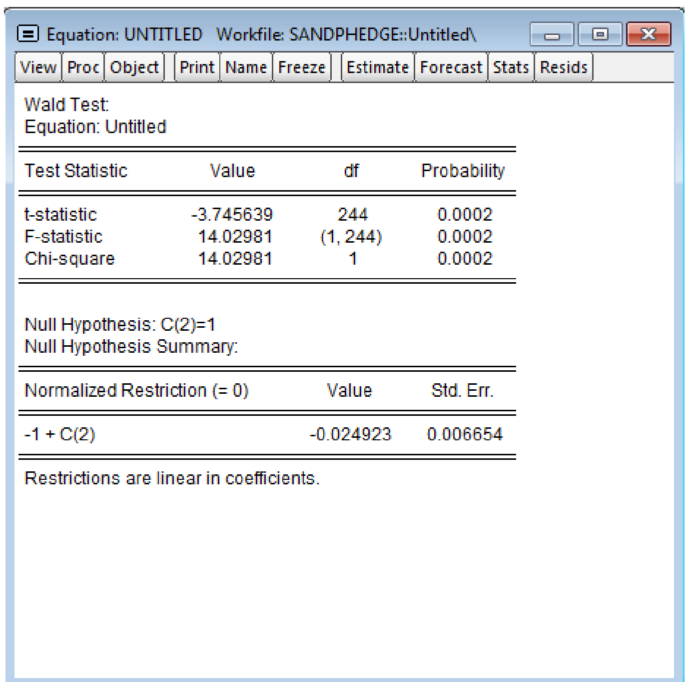
Now estimate a regression for the levels of the series rather than the returns (i.e., run a regression of spot on a constant and futures) and examine the parameter estimates. The return regression slope parameter estimated above measures the optimal hedge ratio and also measures the short run relationship between the two series. By contrast, the slope parameter in a regression using the raw spot and futures indices (or the log of the spot series and the log of the futures series) can be interpreted as measuring the long run relationship between them. For now, click **Quick/Estimate Equation** and enter the variables **spot c futures** in the Equation Specification dialog box, click **OK**, then **name the regression results 'levelreg'**. The intercept estimate ($\hat{\alpha}$) in this regression is –2.838 and the slope estimate ($\hat{\beta}$) is 1.002. The intercept can be considered to approximate the cost of carry, while as expected, the long-term relationship between spot and futures prices is almost 1:1. Finally, click the **Save** button to save the whole workfile.

4 Hypothesis Testing – Example 1: Hedging Revisited

Reading: Brooks (2019, Sections 3.8 and 3.9.

Reload the 'sandphedge.wf1' EViews work file that was created above. If we re-examine the results table from the returns regression (Figure 9, right panel), it can be seen that as well as the parameter estimates, EViews automatically calculates the standard errors, the t-ratios and the p-values associated with a two-sided test of the null hypothesis that the true value of a parameter is zero.

The third column presents the t-ratios, which are the test statistics for testing the null hypothesis that the true values of these parameters are zero against a two sided alternative – i.e., these statistics test $\mathrm{H}_0 : \alpha = 0$ versus $\mathrm{H}_1 : \alpha \neq 0$ in the first row of numbers and $\mathrm{H}_0 : \beta = 0$ versus $\mathrm{H}_1 : \beta \neq 0$ in the second. The fact that the first of these test statistics is very small is indicative that the corresponding null hypotheses is likely not to be rejected but it probably will be rejected for the slope. This suggestion is confirmed by the p-values given in the final column. The intercept p-value is considerably larger than 0.1, indicating that the corresponding test statistic is not even significant at the 10% level; for the slope coefficient, however, it is zero to four decimal places so the null hypothesis is decisively rejected.



Wald Test:
Equation: Untitled

Test Statistic	Value	df	Probability
t-statistic	-3.745639	244	0.0002
F-statistic	14.02981	(1, 244)	0.0002
Chi-square	14.02981	1	0.0002

Null Hypothesis: C(2)=1
Null Hypothesis Summary:

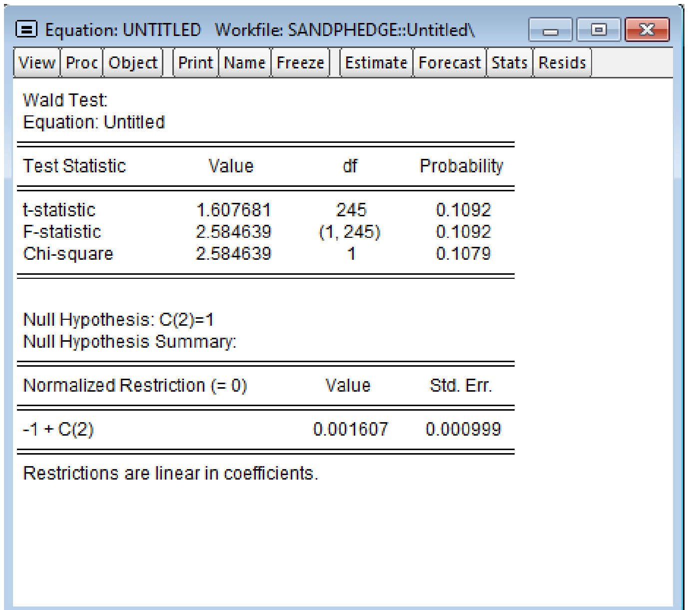
Normalized Restriction (= 0)	Value	Std. Err.
-1 + C(2)	-0.024923	0.006654

Restrictions are linear in coefficients.

Figure 10: Wald Test on the Slope Coefficient

Suppose now that we wanted to test the null hypothesis that $H_0 : \beta = 1$ rather than $H_0 : \beta = 0$. We could test this, or any other hypothesis about the coefficients, by hand, using the information we already have. But it is easier to let EViews do the work by typing **View** and then **Coefficient Diagnostics/Wald Test Coefficient Restrictions . . .** . EViews defines all of the parameters in a vector C, so that C(1) will be the intercept and C(2) will be the slope. Type **C(2)=1** and click **OK**. The window shown in Figure 10 should come up. Note that using this software, it is possible to test multiple hypotheses, which will be discussed in section 6 of this guide, and also non-linear restrictions, which cannot be tested using the standard procedure for inference described above.

The test is performed in three different ways, but results suggest that the null hypothesis should clearly be rejected as the p-value for the test is smaller than 0.01 in each case. Note that, since we are testing a single restriction, the t and F and Chi-squared versions of the test will give the same conclusions. EViews also reports the 'normalised restriction', although this can be ignored for the time being since it merely reports the regression slope parameter (in a different form) and its standard error.


Equation: UNTITLED Workfile: SANDPHEDGE::Untitled\

View | Proc | Object | Print | Name | Freeze | Estimate | Forecast | Stats | Resids

Wald Test:
Equation: Untitled

Test Statistic	Value	df	Probability
t-statistic	1.607681	245	0.1092
F-statistic	2.584639	(1, 245)	0.1092
Chi-square	2.584639	1	0.1079

Null Hypothesis: C(2)=1
Null Hypothesis Summary:

Normalized Restriction (= 0)	Value	Std. Err.
-1 + C(2)	0.001607	0.000999

Restrictions are linear in coefficients.

Figure 11: Wald Test on the Slope Coefficient for the Levels Regression

Now go back to the regression in levels (i.e., with the raw prices rather than the returns) and test the null hypothesis that $\beta = 1$ in this regression. You should find in this case that the null hypothesis is not rejected as p-values are slightly above 0.1 (Figure 11).

5 Hypothesis Testing – Example 2: The CAPM

Reading: Brooks (2019, Sections 3.10 and 3.11)

This exercise will estimate and test some hypotheses about the CAPM beta for several US stocks. First, **open a new workfile** to accommodate monthly data commencing in January 2002 and ending in February 2018. Note that it is standard to employ five years of monthly data for estimating betas but let us use all of the observations (over ten years) for now. Then **import the Excel file 'capm.xls'.** The file is organised by observation and contains six columns of numbers plus the dates in the first column – you should be able to just click through with the default options. The monthly stock prices of four companies (Ford, General Electric, Microsoft and Oracle) will appear as objects, along with index values for the S&P500 ('sandp') and three-month US-Treasury bills ('ustb3m'). **Save the EViews workfile as 'capm.wf1'**.

In order to estimate a CAPM equation for the Ford stock, for example, we need to first transform the price series into returns and then the excess returns over the risk free rate. To transform the series, click on the Generate button (**Genr**) in the workfile window. In the new window, type

RSANDP=100*LOG(SANDP/SANDP(−1))

This will create a new series named RSANDP that will contain the returns of the S&P500. The operator (−1) is used to instruct EViews to use the one-period lagged observation of the series. To estimate percentage returns on the Ford stock, press the **Genr** button again and type

RFORD=100*LOG(FORD/FORD(−1))

This will yield a new series named RFORD that will contain the returns of the Ford stock. If, in the transformation, the new series is given the same name as the old series, then the old series will be overwritten. Note that the returns for the S&P index could have been constructed using a simpler command in the 'Genr' window such as **RSANDP=100*DLOG(SANDP)** as we used previously, but it is instructive to see how the 'dlog' formula is working. Before we can transform the returns into excess returns, we need to be slightly careful because the stock returns are monthly but the Treasury bill yields are annualised. We could run the whole analysis using monthly data or using annualised data and it should not matter which we use, but the two series must be measured consistently. So, to turn the T-bill yields into monthly figures and to write over the original series, press the **Genr** button again and type

USTB3M=USTB3M/12

Now, to compute the excess returns, click **Genr** again and type

ERSANDP=RSANDP-USTB3M

where 'ERSANDP' will be used to denote the excess returns, so that the original raw returns series will remain in the workfile. The Ford returns can similarly be transformed into a set of excess returns.

Now that the excess returns have been obtained for the two series, before running the regression, plot the data to examine visually whether the series appear to move together. To do this, create a new object by clicking on the **Object/New Object** menu on the menu bar. Select **Graph**, provide a name (call the graph **Graph1**) and then in the new window provide the names of the series to plot. In this new window, type

ERSANDP ERFORD

Then click **OK** and Figure 12 will appear. It is evident that the Ford series is far more volatile than the index as a whole, especially during the 2008–9 period, although on average the two series seem to move in the same direction at most points in time.

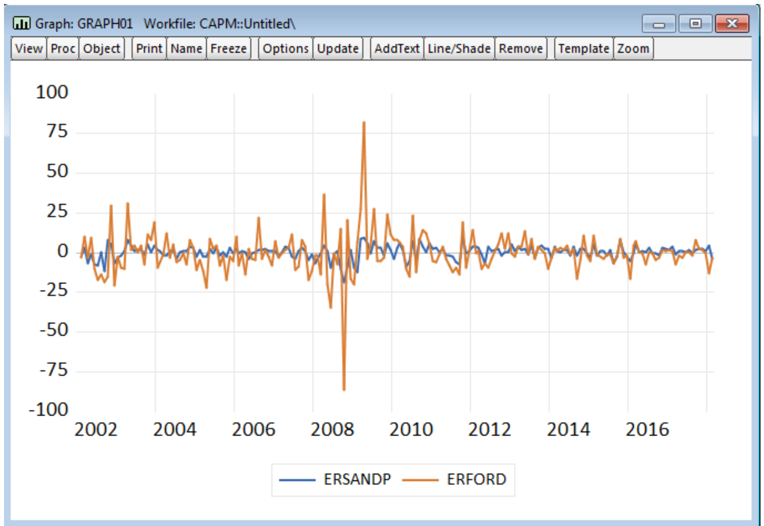


Figure 12: Plot of Excess Returns for the S&P500 and Ford

This is a time series plot of the two variables, but a scatter plot may be more informative. To examine a scatter plot, Click **Options**, choose the **Graph Type** tab, then select **Scatter** from the list and click **OK**. There appears to be a weak positive association between **ERSANDP** and **ERFORD**. Close the window of the graph and return to the workfile window.

To estimate the CAPM equation, click on **Object/New Object ...** . In the new window, select **Equation** (the first option in the list) and name the object **CAPM**. Click on **OK**. In the window, specify the regression equation. The regression equation takes the form

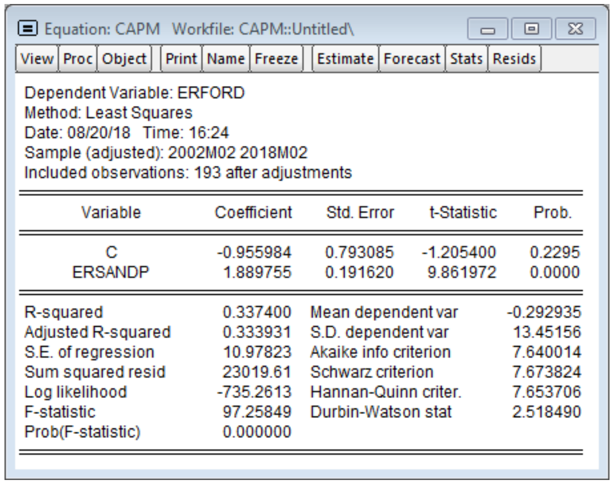
$$(R_{Ford} - r_f)_t = \alpha + \beta(R_M - r_f)_t + u_t \tag{1}$$

Since the data have already been transformed to obtain the excess returns, in order to specify this regression equation, in the equation window type

ERFORD C ERSANDP

To use all the observations in the sample and to estimate the regression using LS – Least Squares (NLS and ARMA), click on **OK**. The results screen appears as in Figure 13. Make sure that you **save the Workfile** again to include the transformed series and regression results.

Take a couple of minutes to examine the results of the regression. What is the slope coefficient estimate, and what does it signify? Is this coefficient statistically significant? The beta coefficient (the slope coefficient) estimate is 1.889. The p-value of the t-ratio is 0.0000, signifying that the excess return on the market proxy has highly significant explanatory power for the variability of the excess returns of Ford stock. What is the interpretation of the intercept estimate? Is it statistically significant?


Equation: CAPM Workfile: CAPM::Untitled\

View | Proc | Object | Print | Name | Freeze | Estimate | Forecast | Stats | Resids

Dependent Variable: ERFORD
Method: Least Squares
Date: 08/20/18 Time: 16:24
Sample (adjusted): 2002M02 2018M02
Included observations: 193 after adjustments

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	-0.955984	0.793085	-1.205400	0.2295
ERSANDP	1.889755	0.191620	9.861972	0.0000

R-squared	0.337400	Mean dependent var	-0.292935
Adjusted R-squared	0.333931	S.D. dependent var	13.45156
S.E. of regression	10.97823	Akaike info criterion	7.640014
Sum squared resid	23019.61	Schwarz criterion	7.673824
Log likelihood	-735.2613	Hannan-Quinn criter.	7.653706
F-statistic	97.25849	Durbin-Watson stat	2.518490
Prob(F-statistic)	0.000000		

Figure 13: Regression Results for the CAPM Regression

In fact, there is a considerably quicker method for using transformed variables in regression equations, and that is to write the transformation directly into the equation window. In the CAPM example above, this could be done by typing

(100*DLOG(FORD))-(USTB3M/12) C (100*DLOG(SANDP))-(USTB3M/12)

into the equation window. As well as being quicker, an advantage of this approach is that the output will show more clearly the regression that has actually been conducted, so that any errors in making the transformations can be seen more clearly.

How could the hypothesis that the value of the population coefficient is equal to 1 be tested? The answer is to click on **View/Coefficient Diagnostics/Wald Test/Coefficient Restrictions...** and then in the box that appears, **Type C(2)=1**. The conclusion here is that the null hypothesis that the CAPM beta of Ford stock is 1 is convincingly rejected and hence the estimated beta of 1.889 is significantly different from 1.[5] The following sections will now re-examine the CAPM model as an illustration of how to conduct multiple hypothesis tests using EViews.

[5]This is hardly surprising given the distance between 1 and 1.889. However, it is sometimes the case, especially if the sample size is quite small and this leads to large standard errors, that many different hypotheses will all result in non-rejection – for example, both $H_0 : \beta = 0$ and $H_0 : \beta = 1$ not rejected.

6 Sample Output for Multiple Hypothesis Tests

Reading: Brooks (2019, Section 4.4)

Reload the 'capm.wk1' workfile constructed in the previous section. Double clicking the object 'capm' reopens the regression results from Figure 13. If we examine the regression F-test, this also shows that the regression slope coefficient is very significantly different from zero, which in this case is exactly the same result as the t-test for the beta coefficient (since there is only one slope coefficient). Thus, in this instance, the F-test statistic is equal to the square of the slope t-ratio.

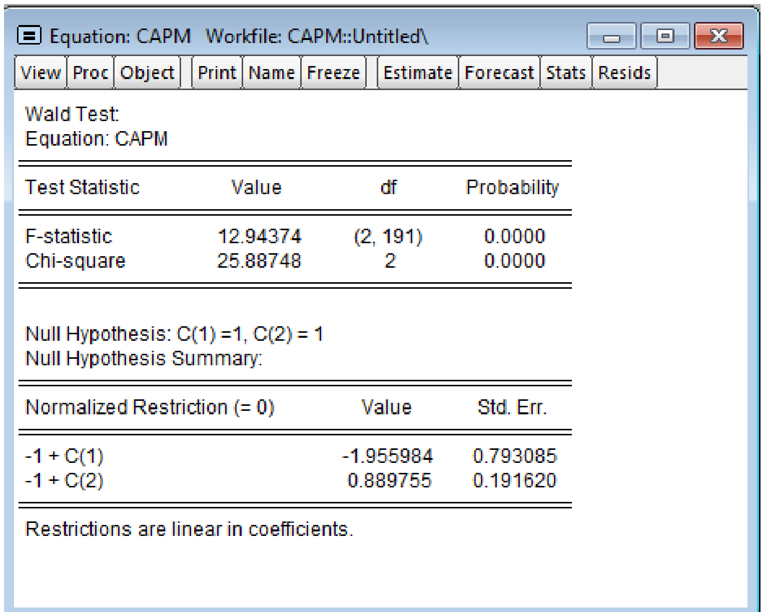


Figure 14: Results of a F-test on Two Coefficients

Now suppose that we wish to conduct a joint test that both the intercept and slope parameters are 1. We would perform this test exactly as for a test involving only one coefficient. Select **View/Coefficient Diagnostics/Wald Test – Coefficient Restrictions...** and then in the box that appears, type **C(1)=1, C(2)=1**. The results are displayed as in Figure 14. There are two versions of the test given: an F-version and a χ^2-version. The F-version is adjusted for small sample bias and should be used when the regression is estimated using a small sample. Both statistics asymptotically yield the same result, and in this case the p-values are very similar. The conclusion is that the joint null hypothesis, $\mathrm{H}_0 : \alpha = 1$ and $\beta = 1$, is strongly rejected.

7 Multiple Regression Using an APT-Style Model

Reading: Brooks (2019, Section 4.4)

In the spirit of arbitrage pricing theory (APT), the following example will examine regressions that seek to determine whether the monthly returns on Microsoft stock can be explained by reference to unexpected changes in a set of macroeconomic and financial variables. **Open a new EViews workfile** to store the data. There are 326 monthly observations in the file **'macro.xls'**, starting in March 1986 and ending in March 2018. There are thirteen series in total plus a column of dates. The series in the Excel file are the Microsoft stock price, the S&P500 index value, the consumer price index, an industrial production index, Treasury bill yields for the following maturities: three months, six months, one year, three years, five years and ten years, a measure of 'narrow' money supply, a consumer credit series, and a 'credit spread' series. The latter is defined as the difference in annualised average yields between a portfolio of bonds rated AAA and a portfolio of bonds rated BAA.

Import the data from the Excel file and save the resulting workfile as **'macro.wf1'**. The first stage is to generate a set of changes or *differences* for each of the variables, since the APT posits that the stock returns can be explained by reference to the *unexpected changes* in the macroeconomic variables rather than their levels. The unexpected value of a variable can be defined as the difference between the actual (realised) value of the variable and its expected value. The question then arises about how we believe that investors might have formed their expectations, and while there are many ways to construct measures of expectations, the easiest is to assume that investors have naive expectations that the next period value of the variable is equal to the current value. This being the case, the entire change in the variable from one period to the next is the unexpected change (because investors are assumed to expect no change).[6]

Transforming the variables can be done as described above. Press **Genr** and then enter the following in the **'Enter equation'** box:

series dspread = d(bminusa)

Repeat theses steps to conduct all of the following transformations

dcredit = d(ccdredit)

dprod = d(indpro)

rmsoft = 100*dlog(microsoft)

rsandp = 100*dlog(sandp)

dmoney = d(m1supply)

inflation = 100*dlog(cpi)

term = ustb10y - ustb3m

and then click **OK**. Next, we need to apply further transformations to some of the transformed series, so **repeat the above steps** to generate

dinflation = d(inflation)

[6] It is an interesting question as to whether the differences should be taken on the levels of the variables or their logarithms. If the former, we have absolute changes in the variables, whereas the latter would lead to proportionate changes. The choice between the two is essentially an empirical one, and this example assumes that the former is chosen, apart from for the stock price series themselves and the consumer price series.

mustb3m=ustb3m/12

rterm = d(term)

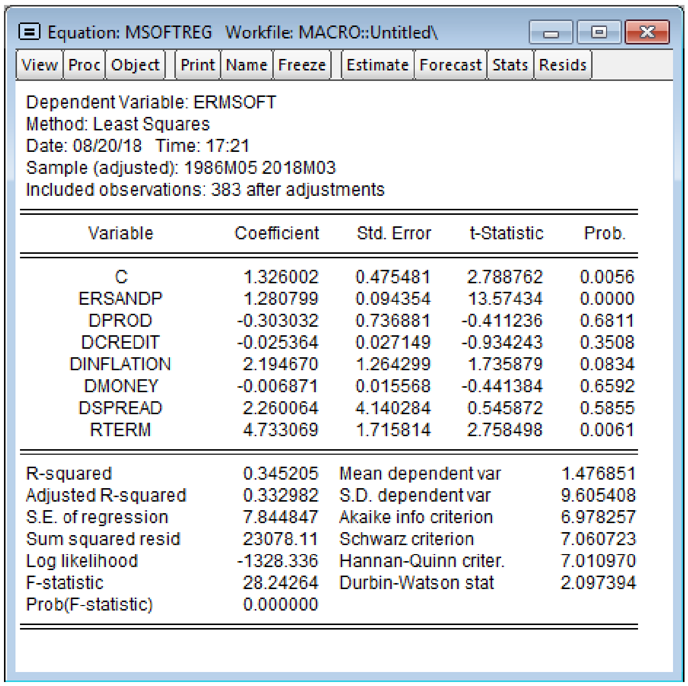
ermsoft=rmsoft-mustb3m

ersandp=rsandp-mustb3m

The final two of these calculate excess returns for the stock and for the index. We can now run the regression. So click **Object/New Object/Equation** and name the object **msoftreg**. **Type the following variables** in the Equation specification window

ermsoft c ersandp dprod dcredit dinflation dmoney dspread rterm

and use **Least Squares** over the whole sample period. The table of results will appear as in Figure 15.



Equation: MSOFTREG Workfile: MACRO::Untitled\

View | Proc | Object | Print | Name | Freeze | Estimate | Forecast | Stats | Resids

Dependent Variable: ERMSOFT
Method: Least Squares
Date: 08/20/18 Time: 17:21
Sample (adjusted): 1986M05 2018M03
Included observations: 383 after adjustments

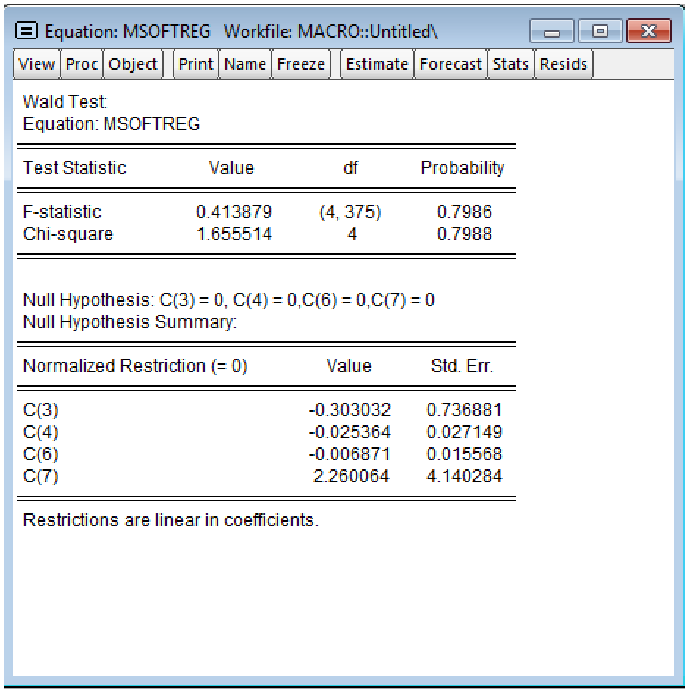
Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	1.326002	0.475481	2.788762	0.0056
ERSANDP	1.280799	0.094354	13.57434	0.0000
DPROD	-0.303032	0.736881	-0.411236	0.6811
DCREDIT	-0.025364	0.027149	-0.934243	0.3508
DINFLATION	2.194670	1.264299	1.735879	0.0834
DMONEY	-0.006871	0.015568	-0.441384	0.6592
DSPREAD	2.260064	4.140284	0.545872	0.5855
RTERM	4.733069	1.715814	2.758498	0.0061

R-squared	0.345205	Mean dependent var	1.476851
Adjusted R-squared	0.332982	S.D. dependent var	9.605408
S.E. of regression	7.844847	Akaike info criterion	6.978257
Sum squared resid	23078.11	Schwarz criterion	7.060723
Log likelihood	-1328.336	Hannan-Quinn criter.	7.010970
F-statistic	28.24264	Durbin-Watson stat	2.097394
Prob(F-statistic)	0.000000		

Figure 15: Results from a Multiple Linear Regression

Take a few minutes to examine the main regression results. Which of the variables has a statistically significant impact on the Microsoft excess returns? Using your knowledge of the effects of the financial and macroeconomic environment on stock returns, examine whether the coefficients have their expected signs and whether the sizes of the parameters are plausible.

The regression F-statistic takes a value 28.24. Remember that this tests the null hypothesis that all of the slope parameters are jointly zero. The p-value of zero attached to the test statistic shows that this null hypothesis should be rejected. However, there are a number of parameter estimates that are not significantly different from zero – specifically those on the dprod, dcredit, dmoney and dspread variables. Let us test the null hypothesis that the parameters on these four variables are jointly zero using an F-test. To test this, Click on **View/Coefficient Diagnostics/Wald Test–Coefficient Restrictions. . .** and in the box that appears type **C(3)=0, C(4)=0, C(6)=0, C(7)=0** and click **OK**. Figure 16 shows that the resulting F-test statistic follows an $F(4, 375)$ distribution as there are four restrictions, 375 usable observations and eight parameters to estimate in the unrestricted regression. The F-statistic value is 0.41 with p-value 0.7986, suggesting that the null hypothesis cannot be rejected. The parameters on 'rterm' and 'dinflation' are significant at the 10% level or better and so the parameters are not included in this F-test and the variable is retained.



Equation: MSOFTREG Workfile: MACRO::Untitled\

View | Proc | Object | Print | Name | Freeze | Estimate | Forecast | Stats | Resids

Wald Test:
Equation: MSOFTREG

Test Statistic	Value	df	Probability
F-statistic	0.413879	(4, 375)	0.7986
Chi-square	1.655514	4	0.7988

Null Hypothesis: C(3) = 0, C(4) = 0,C(6) = 0,C(7) = 0
Null Hypothesis Summary:

Normalized Restriction (= 0)	Value	Std. Err.
C(3)	-0.303032	0.736881
C(4)	-0.025364	0.027149
C(6)	-0.006871	0.015568
C(7)	2.260064	4.140284

Restrictions are linear in coefficients.

Figure 16: Wald Test on Coefficient Restrictions

7.1 Stepwise Regression

There is a procedure known as a *stepwise regression* that is available in EViews. Stepwise regression is an automatic variable selection procedure which chooses the jointly most 'important' (variously defined) explanatory variables from a set of candidate variables. There are a number of different stepwise regression procedures, but the simplest is the uni-directional forwards method. This starts with no variables in the regression (or only those variables that are always required by the researcher to be in the regression) and then it selects first the variable with the lowest p-value (largest t-ratio) if it were included, then the variable with the second lowest p-value conditional upon the first variable already being included, and so on. The procedure continues until the next lowest p-value relative to those already included variables is larger than some specified threshold value, then the selection stops, with no more variables being incorporated into the model.

To conduct a stepwise regression which will automatically select from among these variables the most important ones for explaining the variations in Microsoft stock returns, click **Object/New Object** and then keep the default option **Equation**. Name the equation **Msoftstepwise** and then in the 'Estimation settings/Method' box, change **LS – Least Squares (NLS and ARMA)** to **STEPLS/Stepwise Least Squares** and then in the top box that appears, 'Dependent variable followed by list of always included regressors', enter **ERMSOFT C**

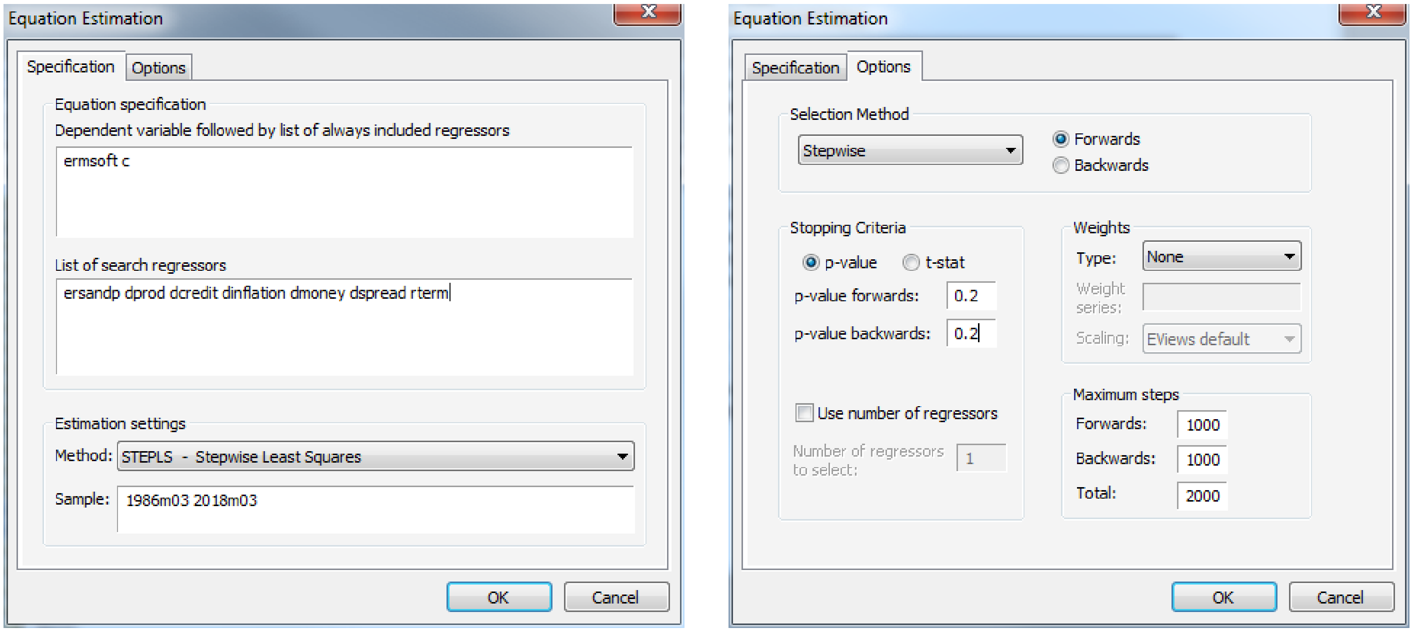


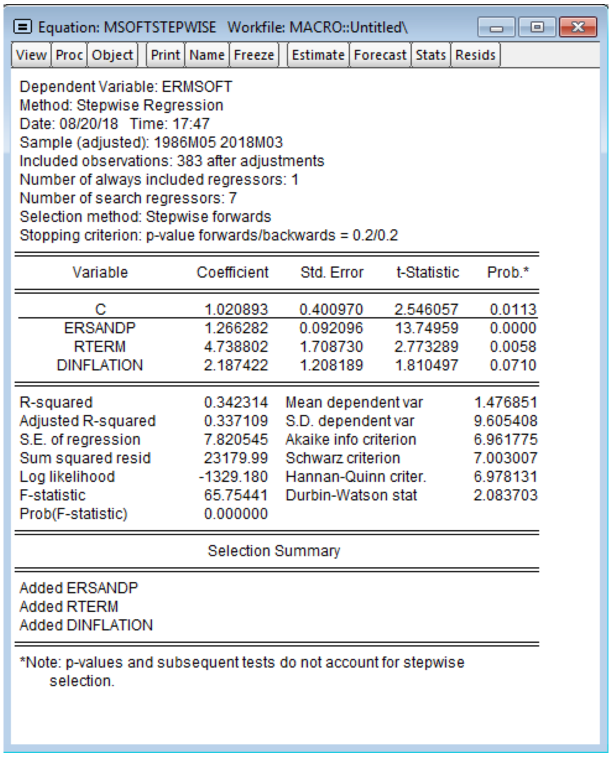
Figure 17: Stepwise Regression Windows

This shows that the dependent variable will be the excess returns on Microsoft stock and that an intercept will always be included in the regression. If the researcher had a strong prior view that a particular explanatory variable must always be included in the regression, it should be listed in this first box. In the second box, ‘List of search regressors’, type the list of all of the explanatory variables used above: **ersandp dprod dcredit dinflation dmoney dspread rterm**. The window will appear as in Figure 17, left panel.

Clicking on the ‘Options’ tab gives a number of ways to conduct the regression as shown in Figure 17, right panel. For example, ‘Forwards’ will start with the list of required regressors (the intercept only in this case) and will sequentially add to them, while ‘Backwards’ will start by including all of the variables and will sequentially delete variables from the regression. The default criterion is to include variables if the p-value is less than 0.5, but this seems high and could potentially result in the inclusion of some very insignificant variables, so **modify this to 0.2** and then click **OK** to see the results as in Figure 18.

As can be seen, the excess market return, the term structure, and unexpected inflation variables have been included, while the money supply, production, default spread and credit variables have been omitted.

Stepwise procedures have been strongly criticised by statistical purists. At the most basic level, they are sometimes argued to be no better than automated procedures for data mining, in particular if the list of potential candidate variables is long and results from a ‘fishing trip’ rather than a strong prior financial theory. More subtly, the iterative nature of the variable selection process implies that the size of the tests on parameters attached to variables in the final model will not be the nominal values (e.g., 5%) that would have applied had this model been the only one estimated. Thus the p-values for tests involving parameters in the final regression should really be modified to take into account that the model results from a sequential procedure, although they are usually not in statistical packages such as EViews.



Equation: MSOFTSTEPWISE Workfile: MACRO::Untitled\

View | Proc | Object | Print | Name | Freeze | Estimate | Forecast | Stats | Resids

Dependent Variable: ERMSOFT
Method: Stepwise Regression
Date: 08/20/18 Time: 17:47
Sample (adjusted): 1986M05 2018M03
Included observations: 383 after adjustments
Number of always included regressors: 1
Number of search regressors: 7
Selection method: Stepwise forwards
Stopping criterion: p-value forwards/backwards = 0.2/0.2

Variable	Coefficient	Std. Error	t-Statistic	Prob.*
C	1.020893	0.400970	2.546057	0.0113
ERSANDP	1.266282	0.092096	13.74959	0.0000
RTERM	4.738802	1.708730	2.773289	0.0058
DINFLATION	2.187422	1.208189	1.810497	0.0710

R-squared	0.342314	Mean dependent var	1.476851
Adjusted R-squared	0.337109	S.D. dependent var	9.605408
S.E. of regression	7.820545	Akaike info criterion	6.961775
Sum squared resid	23179.99	Schwarz criterion	7.003007
Log likelihood	-1329.180	Hannan-Quinn criter.	6.978131
F-statistic	65.75441	Durbin-Watson stat	2.083703
Prob(F-statistic)	0.000000		

Selection Summary

Added ERSANDP
Added RTERM
Added DINFLATION

*Note: p-values and subsequent tests do not account for stepwise selection.

Figure 18: Stepwise Regression Results

8 Quantile Regression

Reading: Brooks (2019, Section 4.10)

To illustrate how to run quantile regressions using EViews, we will now employ the simple CAPM beta estimation conducted above. So **Re-open the 'CAPM.wf1' workfile** constructed previously. Click on **Quick/Estimate Equation. . .** , change Method in Estimation settings to **QREG - Quantile regression (including LAD)** and Figure 19, left panel will appear. Write '**erford c ersandp**' in the Equation specification window. As usual, there is an Options tab that allows the user to control various aspects of the estimation technique. We will set **Coefficient Covariance** to **Ordinary (IID)** and the **Sparsity Estimation Method** as **Siddiqui (mean fitted)**and click **OK** and the quantile regression results for the median will appear (Figure 20, left panel). EViews will estimate the median (0.5 quantile) by default, but any value of τ between 0 and 1 can be chosen. Rather than estimate each quantile separately and obtain a full statistical output in each case, after estimating a single quantile, click **View/Quantile Process/Process Coefficients**.

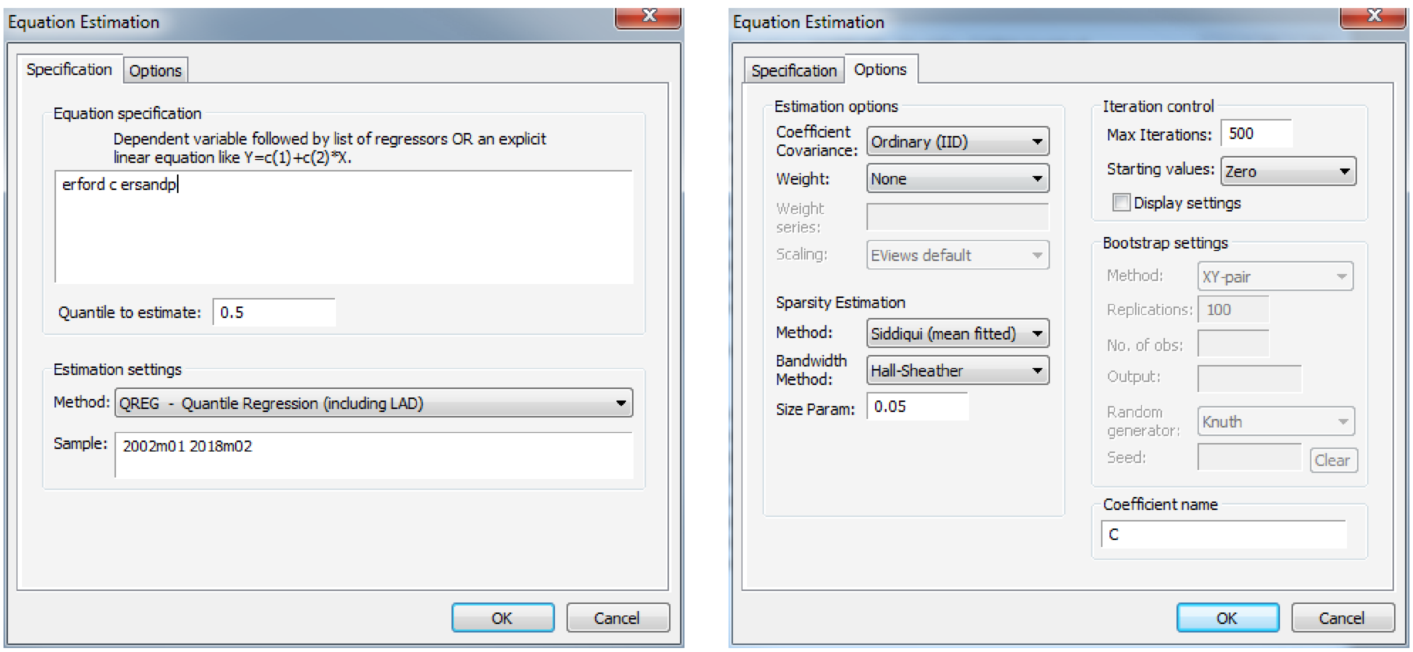


Figure 19: Quantile Regression Estimation Window

EViews will then open a window that permits the simultaneous estimation of a number of quantiles. We choose to estimate quantiles for the data split into ten segments. The quantile estimates can be displayed in a table or in a graph for all of the coefficients (the default) or for specific coefficients. Just click **OK** and results will appear as in Figure 20, right panel. For comparison, the OLS regression results are also presented in the left panel and these are identical to the quantile regression parameters at the 50th percentile.

Equation: UNTITLED Workfile: CAPM::Untitled\

View | Proc | Object | Print | Name | Freeze | Estimate | Forecast | Stats | Resids

Dependent Variable: ERFORD
Method: Quantile Regression (Median)
Date: 08/20/18 Time: 18:58
Sample (adjusted): 2002M02 2018M02
Included observations: 193 after adjustments
Ordinary (IID) Standard Errors & Covariance
Sparsity method: Siddiqui using fitted quantiles
Bandwidth method: Hall-Sheather, bw=0.16812
Estimation successfully identifies unique optimal solution

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	-1.489678	0.639382	-2.329871	0.0209
ERSANDP	1.438457	0.154484	9.311383	0.0000

Pseudo R-squared	0.172352	Mean dependent var	-0.292935
Adjusted R-squared	0.168019	S.D. dependent var	13.45156
S.E. of regression	11.15820	Objective	667.7793
Quantile dependent var	-1.061981	Restr. objective	806.8401
Sparsity	17.70122	Quasi-LR statistic	62.84799
Prob(Quasi-LR stat)	0.000000		

Equation: UNTITLED Workfile: CAPM::Untitled\

View | Proc | Object | Print | Name | Freeze | Estimate | Forecast | Stats | Resids

Quantile Process Estimates
Equation: UNTITLED
Specification: ERFORD C ERSANDP
Estimated equation quantile tau = 0.5
Number of process quantiles: 10
Display all coefficients

	Quantile	Coefficient	Std. Error	t-Statistic	Prob.
C	0.100	-11.92905	1.462810	-8.154885	0.0000
	0.200	-7.349600	0.830446	-8.850181	0.0000
	0.300	-4.878333	0.725336	-6.725617	0.0000
	0.400	-3.347896	0.674528	-4.963313	0.0000
	0.500	-1.489678	0.616447	-2.416555	0.0166
	0.600	-0.018844	0.616189	-0.030582	0.9756
	0.700	1.756783	0.673283	2.609281	0.0098
	0.800	4.001337	0.865894	4.621048	0.0000
	0.900	10.45627	2.094401	4.992488	0.0000
ERSANDP	0.100	2.340420	0.427487	5.474834	0.0000
	0.200	1.804414	0.290820	6.204581	0.0000
	0.300	1.659638	0.250799	6.617404	0.0000
	0.400	1.500533	0.225875	6.643206	0.0000
	0.500	1.438457	0.215489	6.675323	0.0000
	0.600	1.296706	0.218139	5.944403	0.0000
	0.700	1.528341	0.216316	7.065329	0.0000
	0.800	1.619863	0.227501	7.120239	0.0000
	0.900	1.847333	0.391589	4.717534	0.0000

Figure 20: OLS and Quantile Regression Results

The monotonic rise in the intercept coefficients as the quantiles increase is to be expected since the data on y have been arranged that way. But the slope estimates are very revealing – they show that the beta estimate is much higher in the lower tail than in the rest of the distribution of ordered data. Thus the relationship between the excess returns on Ford stock and those of the S&P500 is much stronger when Ford share prices are falling most sharply. This is worrying, for it shows that the 'tail systematic risk' of the stock is greater than for the distribution as a whole. This is related to the observation that when stock prices fall, they tend to all fall at the same time, and thus the benefits of diversification that would be expected from examining only a standard regression of y on x could be much overstated.

Several diagnostic and specification tests for quantile regressions may be computed, and one of particular interest is whether the coefficients for each quantile can be restricted to be the same. To compute this test following estimation of a quantile regression, click **View/Quantile Process/Slope Equality Test...** . Again, several options are possible. Run the test for **10 quantiles** and click **OK**. Output is then shown first as a test of whether the corresponding slope coefficients are identical, followed by a pairwise comparison of one quantile with the next one (e.g., 0.1 with 0.2). The results in this case show that only the first and second as well as sixth and seventh decile are significantly different at the 5% level. The majority of the statistics are insignificant, indicating that, despite the beta estimates differing across the quantiles by an economically large magnitude, they are not statistically significantly different. A further test can be conducted for whether the quantiles are symmetric – that is, the estimates for $\tau = 0.1$ and $\tau = 0.9$ are identical, for instance. If we run this test for the CAPM example here we would find that the null hypothesis is rejected at the 5% level (p-value 0.0414).

9 Calculating Principal Components

Reading: Brooks (2019, Appendix 4.2)

In order to calculate the principal components of a set of series with EViews, the first stage is to compile the series concerned into a group. **Create a new workfile** for a series of monthly observations from January 1982 to May 2018. **Import** the US Treasury bill and bond series of various maturities from the file 'fred.xls'. Save the workfile as **'fred.wf1'**. Select **Object/New Object** and change 'Equation' to 'Group' but do not name the object and click **OK**. When EViews prompts you to give a 'List of series, groups and/or series expressions', enter

gs3m gs6m gs1 gs3 gs5 gs10

and click **OK**. You will then see a spreadsheet containing all six of the series. Name the group **Interest** by clicking the **Name** tab. From within this window, click **View/Principal Components...**. Figure 21 will appear.

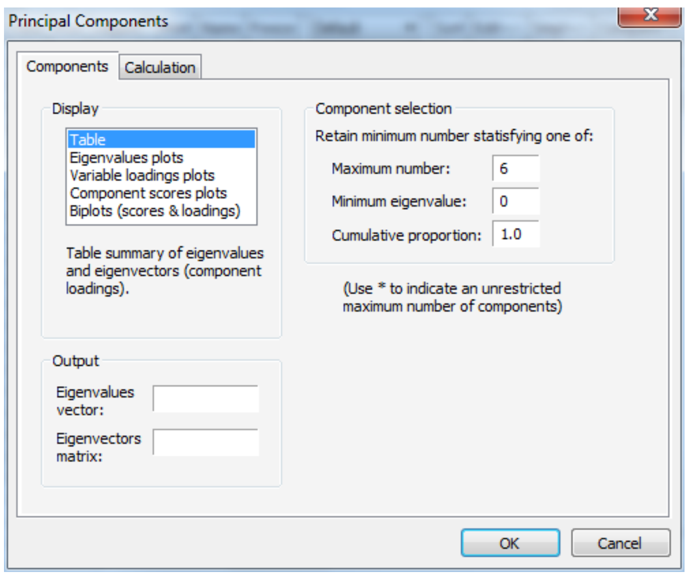


Figure 21: Conducting Principal Components Analysis

There are many features of principal components that can be examined, but for now keep the defaults and click **OK**. The results will appear as in Figure 22. It is evident that there is a great deal of common variation in the series, since the first principal component captures over 98% of the variation in the series and the first two components capture 99.9%. Consequently, if we wished, we could reduce the dimensionality of the system by using two components rather than the entire six interest rate series. Interestingly, the first component comprises almost exactly equal weights in all six series while the second component puts a large negative weight on the shortest yield and gradually increasing weights thereafter. This ties in with the common belief that the first component captures the level of interest rates, the second component captures the slope of the term structure (and the third component captures curvature in the yield curve). **Minimise the group** and you will see that the 'Interest' group has been added to the list of objects.

Group: INTEREST Workfile: FRED::Untitled\

View | Proc | Object | Print | Name | Freeze | Sample | Sheet | Stats | Spec

Principal Components Analysis
Date: 08/20/18 Time: 19:59
Sample: 1982M01 2018M05
Included observations: 437
Computed using: Ordinary correlations
Extracting 6 of 6 possible components

Eigenvalues: (Sum = 6, Average = 1)

Number	Value	Difference	Proportion	Cumulative Value	Cumulative Proportion
1	5.894011	5.794905	0.9823	5.894011	0.9823
2	0.099106	0.093914	0.0165	5.993117	0.9989
3	0.005191	0.003789	0.0009	5.998308	0.9997
4	0.001402	0.001201	0.0002	5.999710	1.0000
5	0.000201	0.000112	0.0000	5.999911	1.0000
6	8.88E-05	---	0.0000	6.000000	1.0000

Eigenvectors (loadings):

Variable	PC 1	PC 2	PC 3	PC 4	PC 5	PC 6
GS3M	0.407122	-0.463403	0.527801	-0.485797	0.278686	-0.165136
GS6M	0.408684	-0.393242	0.073690	0.294843	-0.581599	0.497712
GS1	0.410250	-0.264723	-0.340138	0.549916	0.248837	-0.530655
GS3	0.411254	0.120079	-0.544686	-0.290093	0.424330	0.505527
GS5	0.409037	0.366564	-0.211559	-0.412648	-0.554849	-0.418840
GS10	0.403091	0.641679	0.508784	0.346741	0.185049	0.111503

Ordinary correlations:

	GS3M	GS6M	GS1	GS3	GS5	GS10
GS3M	1.000000					
GS6M	0.998692	1.000000				
GS1	0.995302	0.998568	1.000000			
GS3	0.980045	0.985589	0.992006	1.000000		
GS5	0.964360	0.970793	0.979490	0.996542	1.000000	
GS10	0.938947	0.946275	0.957220	0.983144	0.994327	1.000000

Figure 22: Results of Principal Components Analysis

10 Diagnostic Testing

10.1 Testing for Heteroscedasticity

Reading: Brooks (2019, Section 5.4)

Re-open the **'macro.wf1'** workfile that was examined in section 7 and the regression (msoftreg) that included all the macroeconomic explanatory variables and make sure that the regression output window is open (showing the table of parameter estimates). First, plot the residuals by selecting **View/Actual, Fitted, Residuals/Residual Graph.** The graph should appear as in Figure 23. If the residuals of the regression have systematically changing variability over the sample, that is a sign of heteroscedasticity. In this case, it is hard to see any clear pattern (although it is interesting to note the considerable reduction in volatility post-2003), so we need to run the formal statistical test.

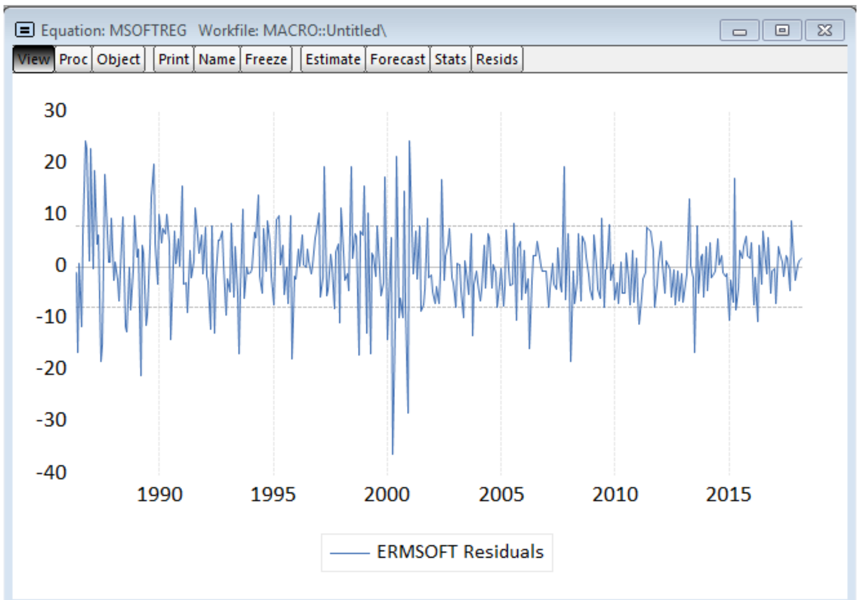
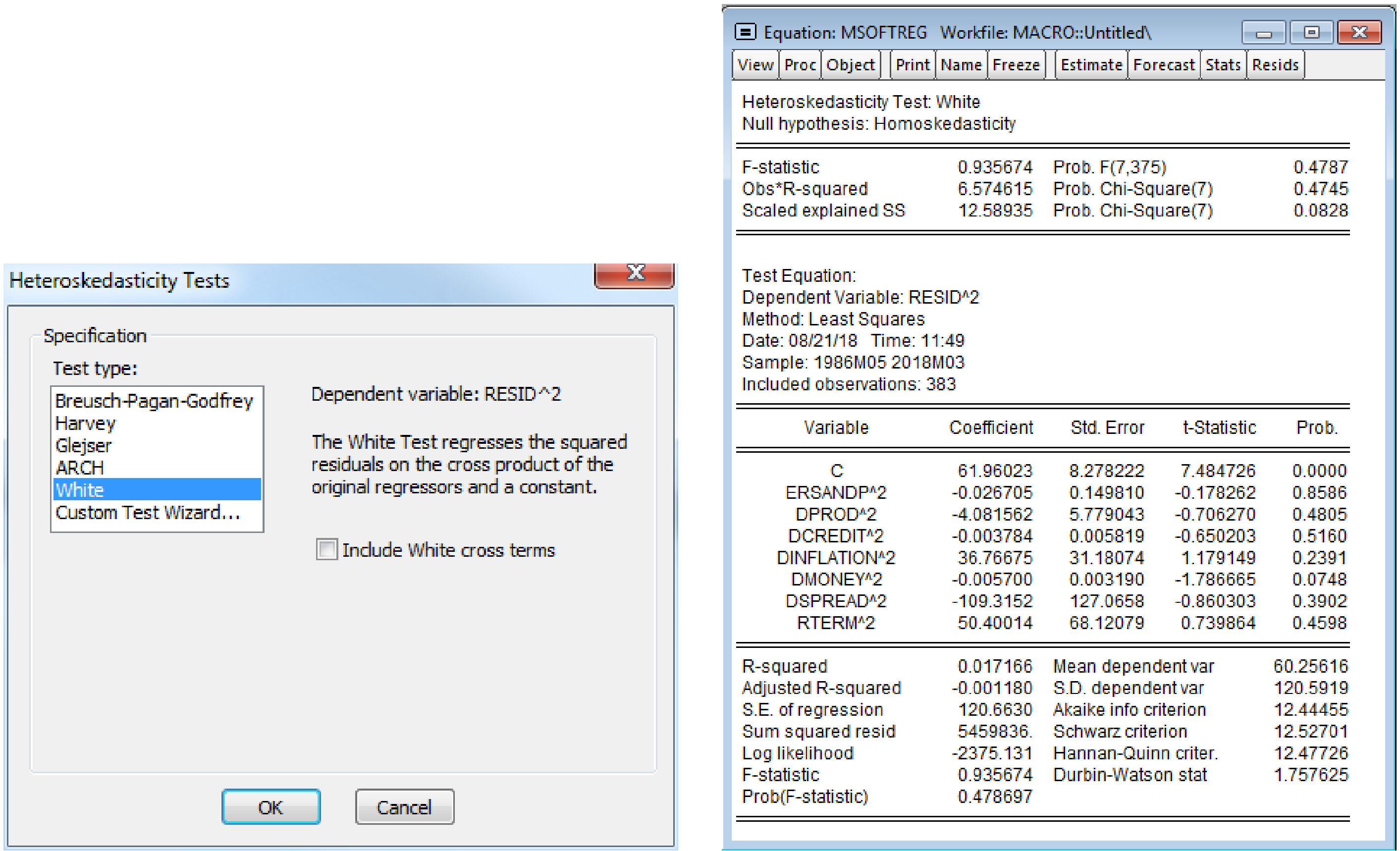


Figure 23: Residual Graph for Microsoft Regression

To test for heteroscedasticity using White's test, click on the **View** button in the regression window and select **Residual Diagnostics/Heteroscedasticity Tests. . .** . You will see a large number of different tests available, including the autoregressive conditional heteroscedasticity (ARCH) test that will be discussed in section 19 of this guide. For now, select the **White** specification. You can also select whether to include the cross-product terms or not (i.e., each variable multiplied by each other variable) or include only the squares of the variables in the auxiliary regression. Uncheck the '**Include White cross terms**' given the relatively large number of variables in this regression (Figure 24, left panel) and then click **OK**. The results of the test will appear as in Figure 24, right panel.



Heteroskedasticity Test: White
Null hypothesis: Homoskedasticity

F-statistic	0.935674	Prob. F(7,375)	0.4787
Obs*R-squared	6.574615	Prob. Chi-Square(7)	0.4745
Scaled explained SS	12.58935	Prob. Chi-Square(7)	0.0828

Test Equation:
Dependent Variable: RESID^2
Method: Least Squares
Date: 08/21/18 Time: 11:49
Sample: 1986M05 2018M03
Included observations: 383

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	61.96023	8.278222	7.484726	0.0000
ERSANDP^2	-0.026705	0.149810	-0.178262	0.8586
DPROD^2	-4.081562	5.779043	-0.706270	0.4805
DCREDIT^2	-0.003784	0.005819	-0.650203	0.5160
DINFLATION^2	36.76675	31.18074	1.179149	0.2391
DMONEY^2	-0.005700	0.003190	-1.786665	0.0748
DSPREAD^2	-109.3152	127.0658	-0.860303	0.3902
RTERM^2	50.40014	68.12079	0.739864	0.4598

R-squared	0.017166	Mean dependent var	60.25616
Adjusted R-squared	-0.001180	S.D. dependent var	120.5919
S.E. of regression	120.6630	Akaike info criterion	12.44455
Sum squared resid	5459836.	Schwarz criterion	12.52701
Log likelihood	-2375.131	Hannan-Quinn criter.	12.47726
F-statistic	0.935674	Durbin-Watson stat	1.757625
Prob(F-statistic)	0.478697		

Figure 24: Conducting a White Test

EViews presents three different types of tests for heteroscedasticity and then the auxiliary regression in the first results table displayed. The test statistics give us the information we need to determine whether the assumption of homoscedasticity is valid or not, but seeing the actual auxiliary regression in the second table can provide useful additional information on the source of the heteroscedasticity if any is found. In this case, both the F- and χ^2 ('LM') versions of the test statistic give the same conclusion that there is no evidence for the presence of heteroscedasticity, since the p-values are considerably in excess of 0.05. The third version of the test statistic, 'Scaled explained SS', which as the name suggests is based on a normalised version of the explained sum of squares from the auxiliary regression, suggests in this case that there is some limited evidence of heteroscedasticity (with the test result significant at the 10% level but not lower). Thus the conclusion of the test is slightly ambiguous but overall we would probably be satisfied that there is not a serious problem here.

10.2 Using White's Modified Standard Error Estimates

Reading: Brooks (2019, Subsection 5.4.3)

In order to estimate the regression with heteroscedasticity-robust standard errors in EViews, select this from the option button in the regression entry window. In other words, **close** the heteroscedasticity test window and **click** on the original 'msoftreg' regression results, then click on the **Estimate** button and in the Equation Estimation window, choose the **Options** tab and Figure 25 will appear.

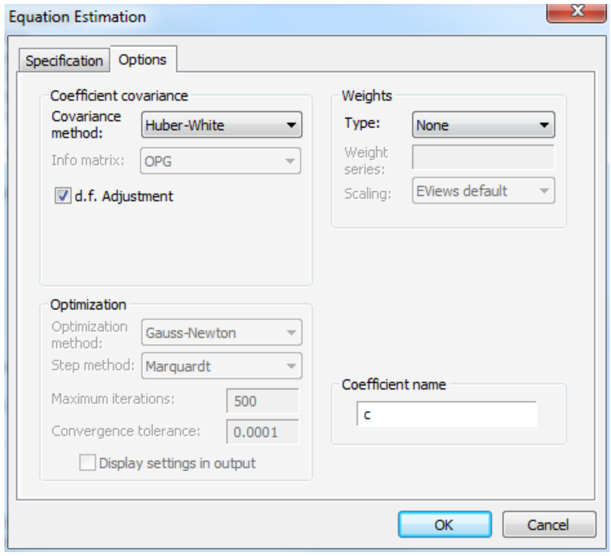


Figure 25: Implementing White's Modified Standard Errors

In the 'Coefficient covariance' box at the top left of the tab, change the option to **Huber-White** and click **OK**. Comparing the results of the regression using heteroscedasticity-robust standard errors with those using the ordinary standard errors, the changes in the significances of the parameters are only marginal. Of course, only the standard errors have changed and the parameter estimates have remained identical to those from before. The heteroscedasticity-consistent standard errors are smaller for all variables except 'rterm' and 'dinflation', resulting in the t-ratios growing in absolute value and the p-values being smaller. However, there are no changes in the conclusions as all variables show at the same significance level as previously.

While White's correction to standard errors for heteroscedasticity as discussed above does not require any user input, the Newey–West procedure requires the specification of a truncation lag length to determine the number of lagged residuals used to evaluate the autocorrelation. The statistical software EViews, for example, uses INTEGER$[4(T/100)^{2/9}]$. In EViews, the Newey–West procedure for estimating the standard errors is employed by invoking it from the same place as the White heteroscedasticity correction. That is, click the **Estimate** button and in the Equation Estimation window, choose the **Options** tab and then instead of checking the 'Huber-White' box, check **HAC (Newey-West)**. While this option is listed under 'Heteroskedasticity consistent coefficient variance', the Newey–West procedure in fact produces 'HAC' (Heteroscedasticity and Autocorrelation Consistent) standard errors that correct for both autocorrelation and heteroscedasticity that may be present.

10.3 Autocorrelation and Dynamic Models

Reading: Brooks (2019, Subsections 5.5.7–5.5.11)

In EViews, the lagged values of variables can be used as regressors or for other purposes by using the notation $x(-1)$ for a one-period lag, $x(-5)$ for a five-period lag, and so on, where x is the variable name. EViews will automatically adjust the sample period used for estimation to take into account the observations that are lost in constructing the lags. For example, if the regression contains five lags of the dependent variable, five observations will be lost and estimation will commence with observation six.

The DW statistic is calculated automatically, and was given in the general estimation output screens that result from estimating any regression model. To view the results screen again, click on the **View** button in the regression window and select **Estimation output**. For the Microsoft macroeconomic regression that included all of the explanatory variables, the value of the DW statistic was 2.097. What is the appropriate conclusion regarding the presence or otherwise of first order autocorrelation in this case?

The Breusch–Godfrey test can be conducted by selecting **View/Residual Diagnostics/Serial Correlation LM Test. . . .** In the new window, type again the number of lagged residuals you want to include in the test and click on **OK**. Assuming that you selected to employ ten lags in the test, the results would be as given in Figure 26.

Equation: MSOFTREG Workfile: MACRO::Untitled\

View | Proc | Object | Print | Name | Freeze | Estimate | Forecast | Stats | Resids

Breusch-Godfrey Serial Correlation LM Test:
Null hypothesis: No serial correlation at up to 10 lags

F-statistic	0.459982	Prob. F(10,365)	0.9150
Obs*R-squared	4.766591	Prob. Chi-Square(10)	0.9062

Test Equation:
Dependent Variable: RESID
Method: Least Squares
Date: 08/21/18 Time: 13:13
Sample: 1986M05 2018M03
Included observations: 383
Presample missing value lagged residuals set to zero.

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	0.024348	0.480649	0.050657	0.9596
ERSANDP	-0.022521	0.096472	-0.233443	0.8155
DPROD	-0.059639	0.753453	-0.079154	0.9370
DCREDIT	0.000611	0.027892	0.021900	0.9825
DINFLATION	-0.388421	1.311149	-0.296245	0.7672
DMONEY	-0.001645	0.015851	-0.103787	0.9174
DSPREAD	0.111915	4.223083	0.026501	0.9789
RTERM	-0.083128	1.746558	-0.047595	0.9621
RESID(-1)	-0.050280	0.052999	-0.948696	0.3434
RESID(-2)	-0.020780	0.052968	-0.392311	0.6951
RESID(-3)	0.053818	0.053956	0.997441	0.3192
RESID(-4)	-0.006274	0.053128	-0.118088	0.9061
RESID(-5)	-0.037044	0.052852	-0.700888	0.4838
RESID(-6)	-0.013066	0.052619	-0.248321	0.8040
RESID(-7)	0.040095	0.053089	0.755251	0.4506
RESID(-8)	-0.000329	0.052773	-0.006226	0.9950
RESID(-9)	-0.037694	0.052885	-0.712749	0.4765
RESID(-10)	0.035434	0.052861	0.670326	0.5031

R-squared	0.012445	Mean dependent var	-4.96E-16
Adjusted R-squared	-0.033550	S.D. dependent var	7.772638
S.E. of regression	7.901949	Akaike info criterion	7.017953
Sum squared resid	22790.89	Schwarz criterion	7.203500
Log likelihood	-1325.938	Hannan-Quinn criter.	7.091557
F-statistic	0.270578	Durbin-Watson stat	2.003414
Prob(F-statistic)	0.998571		

Figure 26: Results of a Breusch–Godfrey Test for Serial Correlation

In the upper part of the output, EViews offers two versions of the test – an F-version and a χ^2 version, while the lower part presents the estimates from the auxiliary regression. The conclusion from both versions of the test in this case is that the null hypothesis of no autocorrelation cannot be rejected. Does this agree with the DW test result? Overall, this is suggestive that no remedial action is necessary.

10.4 Testing for Non-Normality

Reading: Brooks (2019, Section 5.7)

By selecting **View/Residual Diagnostics/Histogram Normality Test** the Jarque–Bera normality tests results can be viewed. The statistic has a χ^2 distribution with two degrees of freedom under the null hypothesis of normally distributed errors. If the residuals are normally distributed, the histogram should be bell-shaped and the Bera–Jarque statistic would not be significant. This means that the p-value given at the bottom of the normality test screen should be bigger than 0.05 so as, to not reject the null of normality at the 5% level. In the example of the Microsoft regression, the screen would appear as in Figure 27.

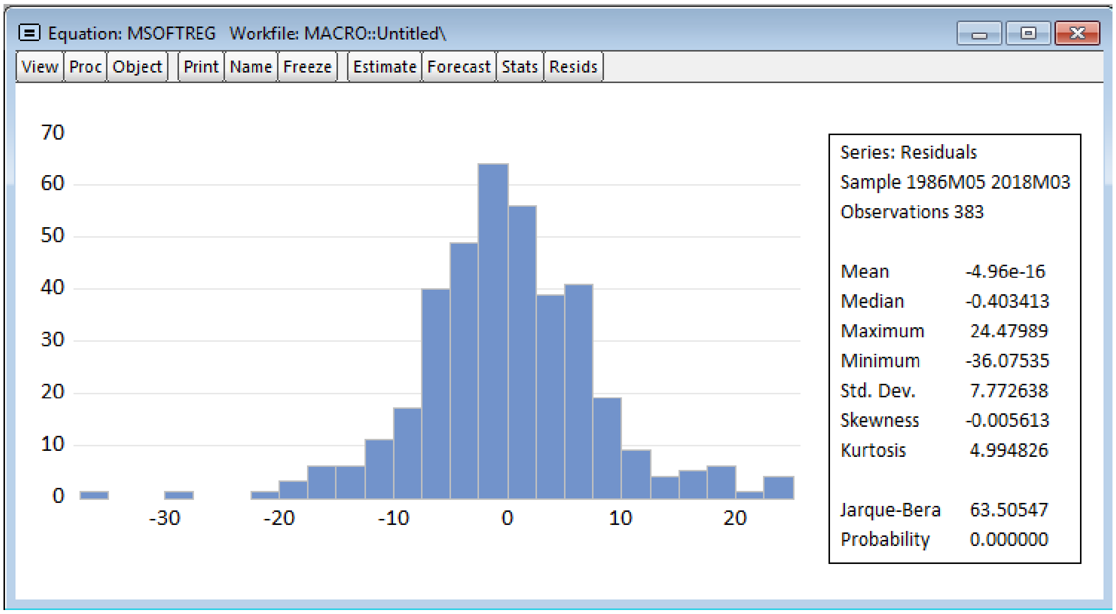


Figure 27: Results of Jarque–Bera Test (Lower Right Corner)

In this case, the residuals are very negatively skewed and are leptokurtic. Hence the null hypothesis for residual normality is rejected very strongly (the p-value for the JB test is zero to six decimal places), implying that the inferences we make about the coefficient estimates could be wrong, although the sample is probably large enough that we need be less concerned than we would be with a small sample. The non-normality in this case appears to have been caused by a small number of very large negative residuals representing monthly stock price falls of more than 25%.

10.5 Dummy Variable Construction and Application

Reading: Brooks (2019, Subsection 5.7.2)

As we saw from the plot of the distribution above, the non-normality in the residuals from the Microsoft regression appears to have been caused by a small number of outliers in the sample. Such events can be identified if they are present by plotting the actual values, the fitted values and the residuals of the regression. This can be achieved in EViews by selecting **View/Actual, Fitted, Residual/Actual, Fitted, Residual Graph**. The plot should look as in Figure 28.

From the graph, it can be seen that there are several large (negative) outliers, but the largest of all occur in 2000. All of the large outliers correspond to months where the actual return was much smaller (i.e., more negative) than the model would have predicted. Interestingly, the residual in October 1987 is not quite so prominent because even though the stock price fell, the market index value fell as well, so that the stock price fall was at least in part predicted (this can be seen by comparing the actual and fitted values during that month).

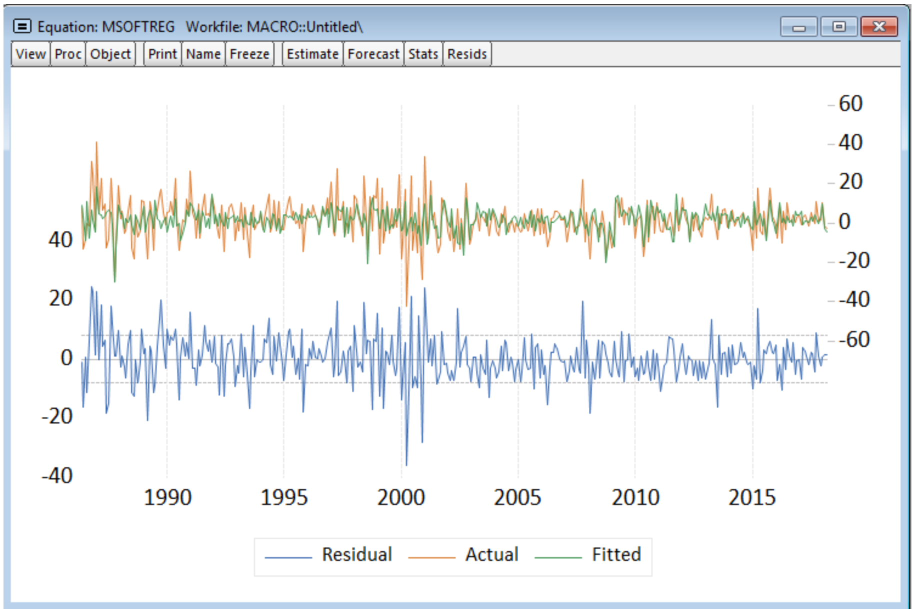


Figure 28: Regression Residuals, Actual Values and Fitted Series

In order to identify the exact dates that the biggest outliers were realised, we could use the shading option by right clicking on the graph and selecting the 'add lines & shading' option. But it is probably easier to just examine a table of values for the residuals, which can be achieved by selecting **View/Actual, Fitted, Residual/Actual, Fitted, Residual Table**. If we do this, it is evident that the two most extreme residuals were in April (−36.075) and December 2000 (−28.143).

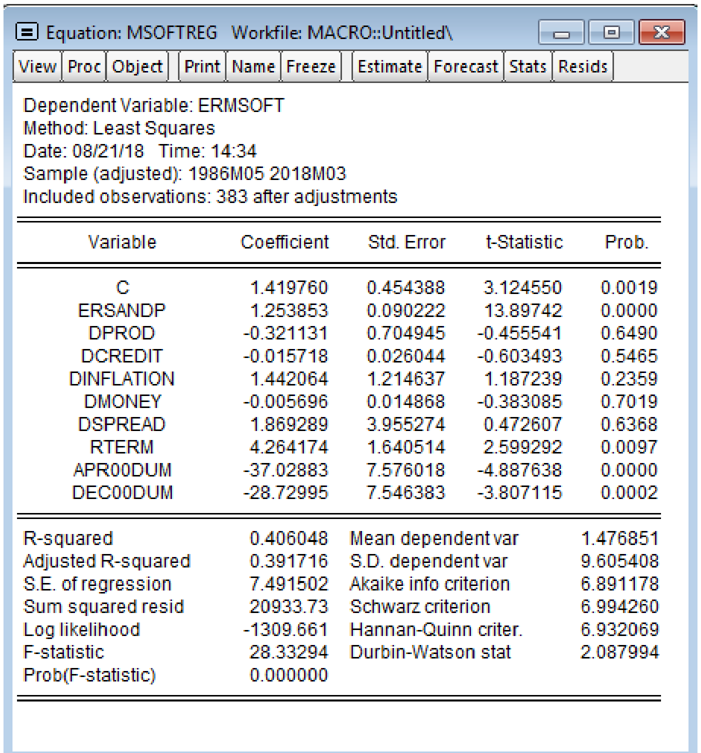
As stated above, one way to remove big outliers in the data is by using dummy variables. It would be tempting, but incorrect, to construct one dummy variable that takes the value 1 for both April 2000 and December 2000, but this would not have the desired effect of setting both residuals to zero. Instead, to remove two outliers requires us to construct two separate dummy variables. In order to create the April 2000 dummy first, we generate a series called 'APR00DUM' that will initially contain only zeros. **Generate this series** (hint: you can use 'Quick/Generate Series' and then type in the box 'APR00DUM = 0'. **Double click on the new object** to open the spreadsheet and **turn on the editing mode** by clicking 'Edit +/−' and input a single **1** in the cell that corresponds to April 2000. Leave all other cell entries as zeros). However, note that the dummy variables could be created also by typing **APR00DUM = @date = @dateval("2000m04")** into the **Generate** window.[7]

Once this dummy variable has been created, repeat the process above to **create another dummy variable** called 'DEC00DUM' that takes the value 1 in December 2000 and zero elsewhere and then **rerun the regression** including all the previous variables plus these two dummy variables. This can most easily be achieved by clicking on the **'Msoftreg' results object**, then the **Estimate** button and **adding the dummy variables** to the end of the variable list. The full list of variables is

ermsoft c ersandp dprod dcredit dinflation dmoney dspread rterm apr00dum dec00dum

and the results of this regression are as in Figure 29.

[7]The second approach makes use of the logical operation '=' and hence sets all values for which the date does not equal to April 2000 to 0.



Equation: MSOFTREG Workfile: MACRO::Untitled\

View | Proc | Object | Print | Name | Freeze | Estimate | Forecast | Stats | Resids

Dependent Variable: ERMSOFT
Method: Least Squares
Date: 08/21/18 Time: 14:34
Sample (adjusted): 1986M05 2018M03
Included observations: 383 after adjustments

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	1.419760	0.454388	3.124550	0.0019
ERSANDP	1.253853	0.090222	13.89742	0.0000
DPROD	-0.321131	0.704945	-0.455541	0.6490
DCREDIT	-0.015718	0.026044	-0.603493	0.5465
DINFLATION	1.442064	1.214637	1.187239	0.2359
DMONEY	-0.005696	0.014868	-0.383085	0.7019
DSPREAD	1.869289	3.955274	0.472607	0.6368
RTERM	4.264174	1.640514	2.599292	0.0097
APR00DUM	-37.02883	7.576018	-4.887638	0.0000
DEC00DUM	-28.72995	7.546383	-3.807115	0.0002

R-squared	0.406048	Mean dependent var	1.476851
Adjusted R-squared	0.391716	S.D. dependent var	9.605408
S.E. of regression	7.491502	Akaike info criterion	6.891178
Sum squared resid	20933.73	Schwarz criterion	6.994260
Log likelihood	-1309.661	Hannan-Quinn criter.	6.932069
F-statistic	28.33294	Durbin-Watson stat	2.087994
Prob(F-statistic)	0.000000		

Figure 29: Multiple Regression with Dummy Variables

Note that the dummy variable parameters are both highly significant and take approximately the values that the corresponding residuals would have taken if the dummy variables had not been included in the model.[8] By comparing the results with those of the regression above that excluded the dummy variables, it can be seen that the coefficient estimates on the remaining variables change quite a bit in this instance. The inflation parameter is now insignificant and the R^2 value has risen from 0.34 to 0.41 because of the perfect fit of the dummy variables to those two extreme outlying observations.

Finally, if we re-examine the normality test by clicking **View/Residual Diagnostics/Histogram - Normality Test**, we will see that while the skewness and kurtosis are both slightly closer to the values that they would take under normality, the Jarque–Bera test statistic still takes a value of 28.88 (compared with 63.51 previously).

We would thus conclude that the residuals are still a long way from following a normal distribution, and the distribution plot shows that there are still several more very large negative residuals. While it would be possible to continue to generate dummy variables, there is a limit to the extent to which it would be desirable to do so. With this particular regression, we are unlikely to be able to achieve a residual distribution that is close to normality without using an excessive number of dummy variables. As a rule of thumb, in a monthly sample with 381 observations, it is reasonable to include, perhaps, two or three dummy variables for outliers, but more would probably be excessive.

10.6 Multicollinearity

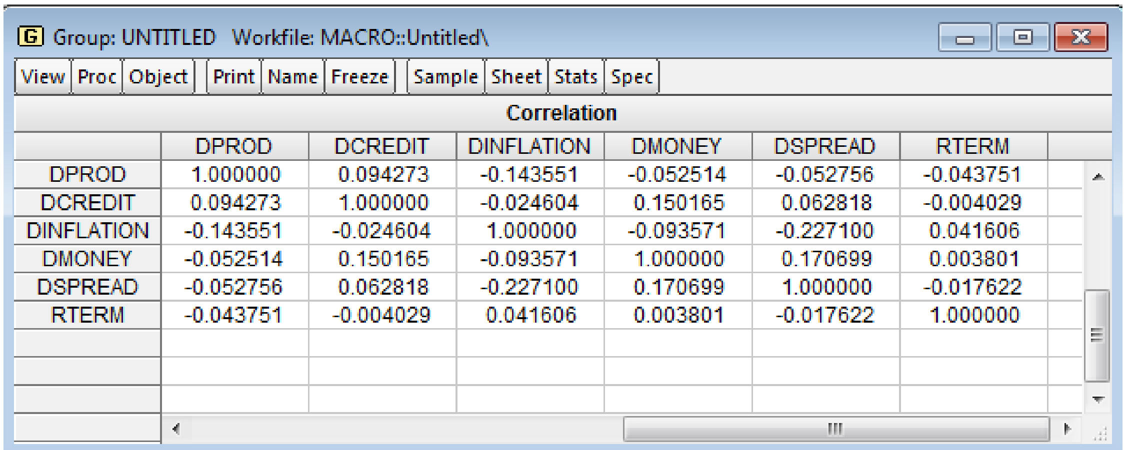
Reading: Brooks (2019, Section 5.8)

[8]Note the inexact correspondence between the values of the residuals and the values of the dummy variable parameters because two dummies are being used together; had we included only one dummy, the value of the dummy variable coefficient and that which the residual would have taken would be identical.

For the Microsoft stock return example given above, a correlation matrix for the macroeconomic independent variables can be constructed by clicking **Quick/Group Statistics/Correlations** and then entering the list of regressors (not including the regressand or the S&P returns) in the dialog box that appears:

dprod dcredit dinflation dmoney dspread rterm

A new window will be displayed that contains the correlation matrix of the series in a spreadsheet format as in Figure 30.



	DPROD	DCREDIT	DINFLATION	DMONEY	DSPREAD	RTERM
DPROD	1.000000	0.094273	-0.143551	-0.052514	-0.052756	-0.043751
DCREDIT	0.094273	1.000000	-0.024604	0.150165	0.062818	-0.004029
DINFLATION	-0.143551	-0.024604	1.000000	-0.093571	-0.227100	0.041606
DMONEY	-0.052514	0.150165	-0.093571	1.000000	0.170699	0.003801
DSPREAD	-0.052756	0.062818	-0.227100	0.170699	1.000000	-0.017622
RTERM	-0.043751	-0.004029	0.041606	0.003801	-0.017622	1.000000

Figure 30: Correlation Matrix of Macroeconomic Variables

Do the results indicate any significant correlations between the independent variables? In this particular case, the largest observed correlations (in absolute value) are 0.17 between the money supply and spread variables, and –0.23 between the spread and unexpected inflation. This is probably sufficiently small that it can reasonably be ignored.

10.7 The RESET Test for Functional Form

Reading: Brooks (2019, Section 5.9)

Using EViews, the Ramsey RESET test is found in the **View** menu of the regression window (for 'msoftreg') under **Stability Diagnostics/Ramsey RESET test...** . EViews will prompt you for the 'number of fitted terms', equivalent to the number of powers of the fitted value to be used in the regression; leave the default of **1** to consider only the square of the fitted values. The Ramsey RESET test for this regression is in effect testing whether the relationship between the Microsoft stock excess returns and the explanatory variables is linear or not. The results of this test for one fitted term are shown in Figure 31.

The test is presented in a t, $F-$ and χ^2 version in the first three rows, respectively, and it can be seen that there is no evidence for non-linearity in the regression equation (the p-values indicate that the test statistics are all insignificant). So it would be concluded that there is some support for the notion that the linear model for the Microsoft returns is appropriate.

Equation: MSOFTREG Workfile: MACRO::Untitled\

View | Proc | Object | Print | Name | Freeze | Estimate | Forecast | Stats | Resids

Ramsey RESET Test
Equation: MSOFTREG
Specification: ERMSOFT C ERSANDP DPROD DCREDIT DINFLATION
DMONEY DSPREAD RTERM APR00DUM DEC00DUM
Omitted Variables: Squares of fitted values

	Value	df	Probability
t-statistic	1.182333	372	0.2378
F-statistic	1.397912	(1, 372)	0.2378
Likelihood ratio	1.436551	1	0.2307

F-test summary:

	Sum of Sq.	df	Mean Squares
Test SSR	78.37086	1	78.37086
Restricted SSR	20933.73	373	56.12260
Unrestricted SSR	20855.36	372	56.06279

LR test summary:

	Value
Restricted LogL	-1309.661
Unrestricted LogL	-1308.942

Unrestricted Test Equation:
Dependent Variable: ERMSOFT
Method: Least Squares
Date: 08/21/18 Time: 15:02
Sample: 1986M05 2018M03
Included observations: 383

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	1.154215	0.506647	2.278144	0.0233
ERSANDP	1.267339	0.090892	13.94328	0.0000
DPROD	-0.362070	0.705420	-0.513268	0.6081
DCREDIT	-0.013409	0.026104	-0.513681	0.6078
DINFLATION	1.424977	1.214076	1.173713	0.2413
DMONEY	-0.005280	0.014864	-0.355229	0.7226
DSPREAD	1.156752	3.998839	0.289272	0.7725
RTERM	3.911671	1.666525	2.347202	0.0194
APR00DUM	-50.13489	13.42423	-3.734656	0.0002
DEC00DUM	-34.58034	9.020624	-3.833476	0.0001
FITTED^2	0.007379	0.006241	1.182333	0.2378

R-squared	0.408271	Mean dependent var	1.476851
Adjusted R-squared	0.392365	S.D. dependent var	9.605408
S.E. of regression	7.487509	Akaike info criterion	6.892650
Sum squared resid	20855.36	Schwarz criterion	7.006040
Log likelihood	-1308.942	Hannan-Quinn criter.	6.937630
F-statistic	25.66664	Durbin-Watson stat	2.094285
Prob(F-statistic)	0.000000		

Figure 31: Results of a Ramsey RESET Test

10.8 Stability Tests

Reading: Brooks (2019, Section 5.12)

In EViews, to access the Chow test, click on **View/Stability Diagnostics/Chow Breakpoint Test . . .** in the 'Msoftreg' regression window. In the new window that appears, enter the date at which it is believed that a breakpoint occurred. Input **1996:01** in the dialog box in Figure 32 to split the sample roughly in half. Note that it is not possible to conduct a Chow test or a parameter stability test when there are outlier dummy variables in the regression, so make sure that APR00DUM and DEC00DUM are omitted from the variable list. This occurs because when the sample is split into two parts, the dummy variable for one of the parts will have values of zero for all observations, which would thus cause perfect multicollinearity with the column of ones that is used for the constant term. So ensure that the Chow test is performed using the regression containing all of the explanatory variables except the dummies. By default, EViews allows the values of all the parameters to vary across the two subsamples in the unrestricted regressions although, if we wanted, we could force some of the parameters to be fixed across the two subsamples.

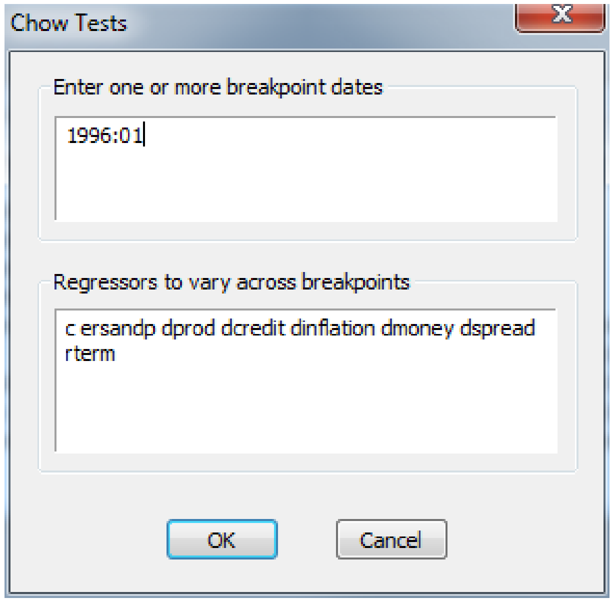


Figure 32: Conducting a Chow Test

EViews gives three versions of the test statistics, as shown in Figure 33. The first version of the test is the familiar F-test, which computes a restricted version and an unrestricted version of the auxiliary regression and 'compares' the residual sums of squares, while the second and third versions are based on χ^2 formulations. In this case, all three test statistics are very close to their critical values at the 5% level, suggesting that the null hypothesis of equal parameters across the two subsamples would be close to being rejected.

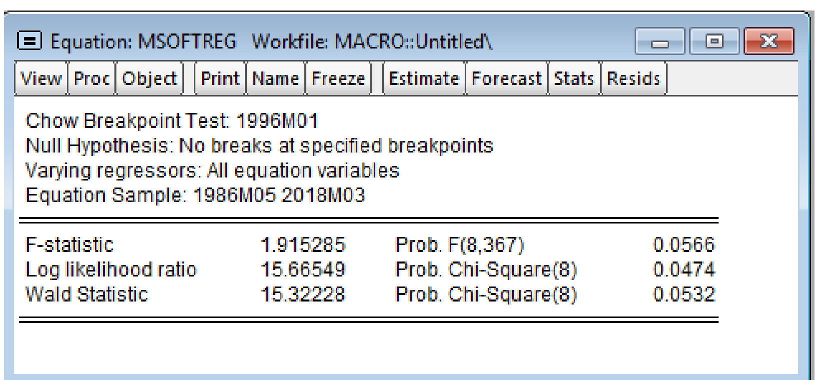


Figure 33: Results of a Chow Breakpoint Test

Note that the Chow forecast (i.e., the predictive failure) test could also be employed by clicking on the **View/Stability Diagnostics/Chow Forecast Test . . .** in the regression window. **Determine whether the model can predict the last four observations** by entering **2017:12** in the dialog box. The results of this test are given in Figure 34 (note that only the first two lines of results are presented since the remainder are not needed for interpretation).

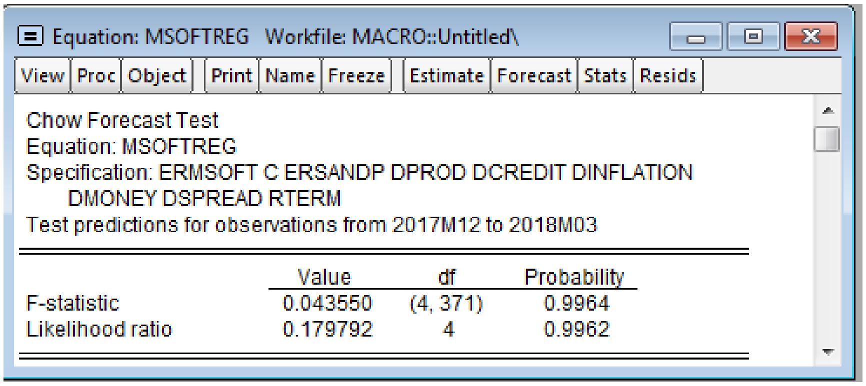


Figure 34: Results of a Chow Forecast Test

The results indicate that the model can indeed adequately predict the 2007 observations. Thus the conclusions from both forms of the test are that there is no evidence of parameter instability. However, the conclusion should really be that the parameters are stable *with respect to this particular break date*. It is important to note that, for the model to be deemed adequate, it needs to be stable with respect to any break dates that we may choose. A good way to test this is to use one of the tests based on recursive estimation.

Click on **View/Stability Diagnostics/Recursive Estimates (OLS Only) ...** . You will be presented with a menu as shown in Figure 35, containing a number of options including the CUSUM and CUSUMSQ tests described above and also the opportunity to plot the recursively estimated coefficients.

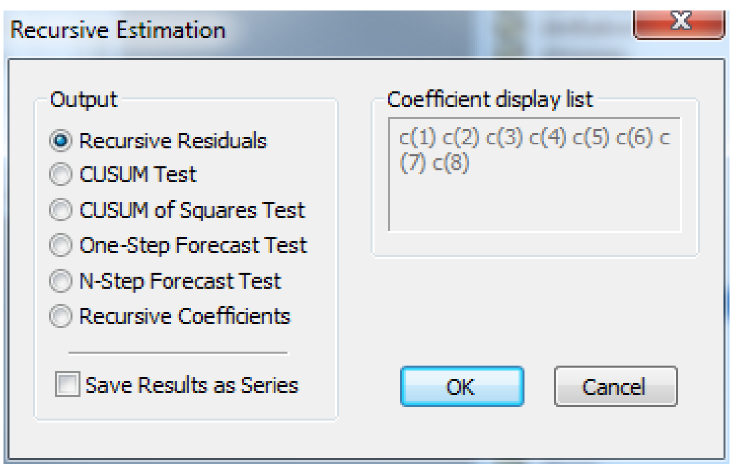


Figure 35: Calculating Recursive Estimates

First, check the box next to **Recursive coefficients** and then recursive estimates will be given for all those parameters listed in the 'Coefficient display list' box, which by default is all of them. Click **OK** and you will be presented with eight small figures, one for each parameter, showing the recursive estimates and ±2 standard error bands around them. As discussed above, it is bound to take some time for the coefficients to stabilise since the first few sets are estimated using such small samples. Given this, the parameter estimates in all cases are fairly stable over time. Now go back to **View/Stability Diagnostics/Recursive Estimates (OLS Only)...** and choose **CUSUM Test**. The resulting graph is in Figure 36, left panel.

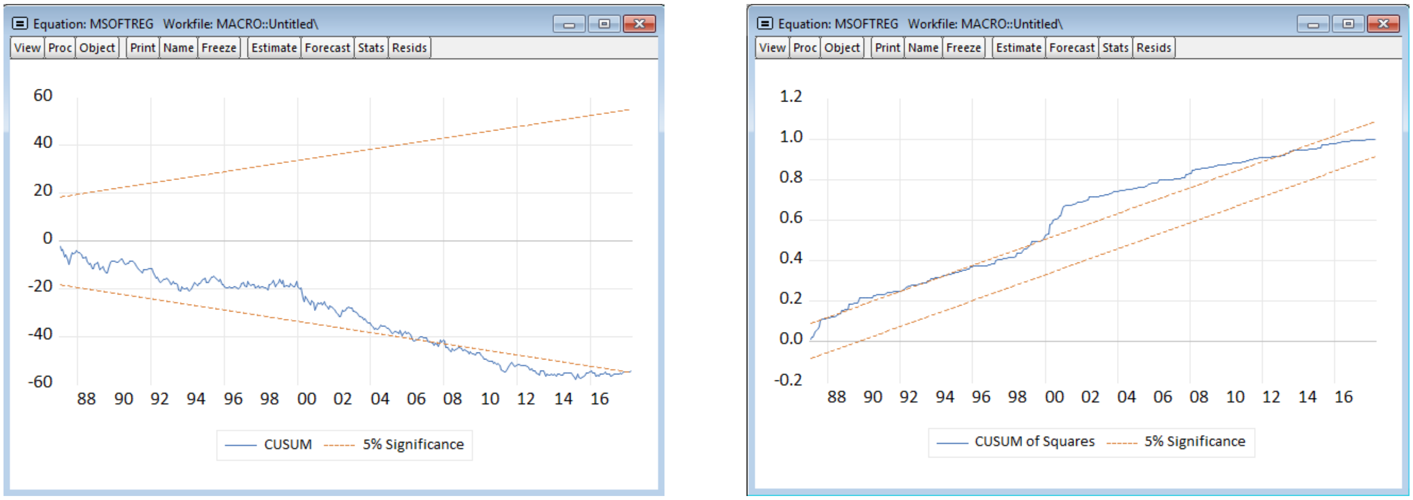


Figure 36: CUSUM and CUSUMSQ Tests for Parameter Stability

Since the line is close to the lower confidence band and even crosses it after 2008, the conclusion would be that the null hypothesis of stability is rejected after the financial crisis of 2008. **Now repeat the above but using the CUSUMSQ test rather than CUSUM**. You should end up with the window in Figure 36, right panel. This test suggests even more evidence of parameter instability, with the line being on or outside of one of the confidence bands for most of the sample period.

10.9 The Kalman Filter for Time-Varying Coefficients

Reading: Brooks (2019, Section 10.14)

Time-varying parameter models are an appealing extension of those with fixed parameters. In general, there are many situations where we might expect parameters to change (probably slowly) over time. If we wanted to allow for the coefficients in the regression to vary with time, instead of using rolling window regression, we can use a state space formulation of the problem:

$$R_t = \alpha + \beta_t X_t + u_t \tag{2}$$
$$\beta_t = \beta_{t+1} + \eta_t \tag{3}$$

To model the observation or measurement equation (2) and the state equation (3) in EViews, we need to create a State Space object. To do this, we re-open the **'macro.wf1'** workfile and click on **Object/New Object** and specify our new object as state space object (**SSpace**). Name this object **tv_beta** as in Figure 37 and click **OK**.

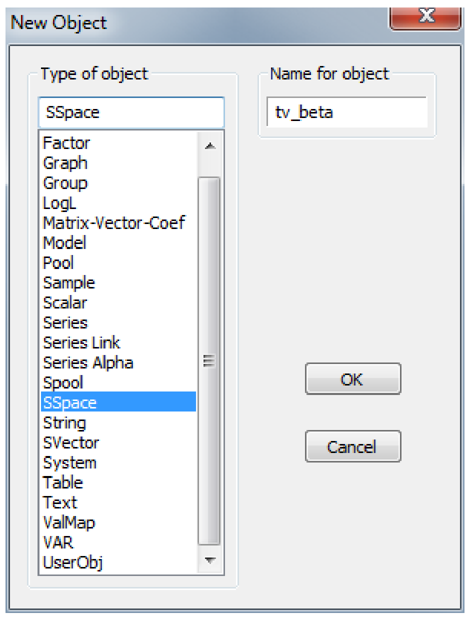


Figure 37: Creating a State Space Object

A new empty window will appear, where we can now specify our state space model. We click on **Proc/Define State Space...** to use the Auto-Specification window. In the tab **Basic Regression** (Figure 38, left panel) we specify the dependent variable as **ermsoft** and put the constant **c** into the box for regressors with fixed coefficients, because we do not want the coefficient for α in equation (2) to vary. We switch to the tab **Stochastic Regressors** (Figure 38, right panel). Since our state space equation (3) is essentially a random walk, we use the **ersandp** variable as the regressor in the box for Random walk coefficients. We leave the Variance Specification tab unchanged, but note that by changing the covariance structure we can allow for more flexibility in our estimation. Finally, we click **OK** and the state space formulation will appear as in Figure 39.

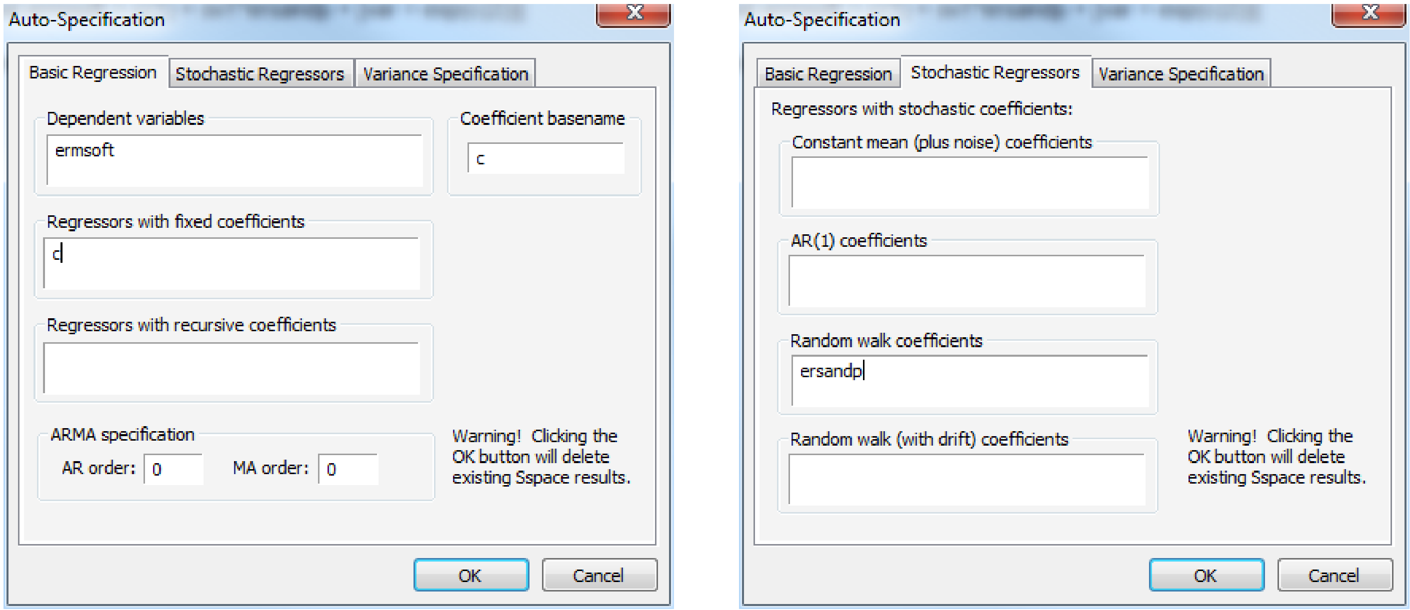


Figure 38: Specifying a State Space Model

The observation equation is preceded by **@signal** and contains the state variable **sv1** which corresponds to β_t in equation (2). Coefficients to be estimated are denoted by **c()**, which in this case are three coefficients: the intercept (α), **c(1)**; the variance of the measurement errors, **c(2)**; and the variance of random walk process, **c(3)**. The formulation **var = exp(c(2))** and **var = exp(c(3))** ensure that the variances are positive. The state equation is preceded by **@state** and contains the random walk as in equation (3).

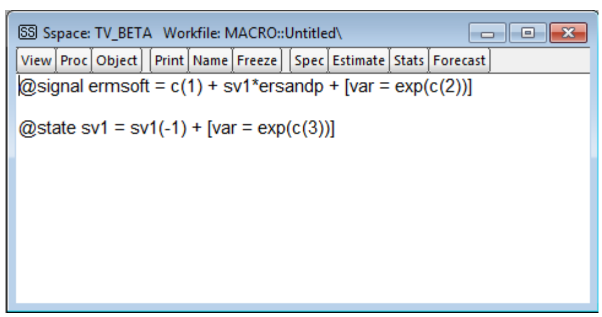
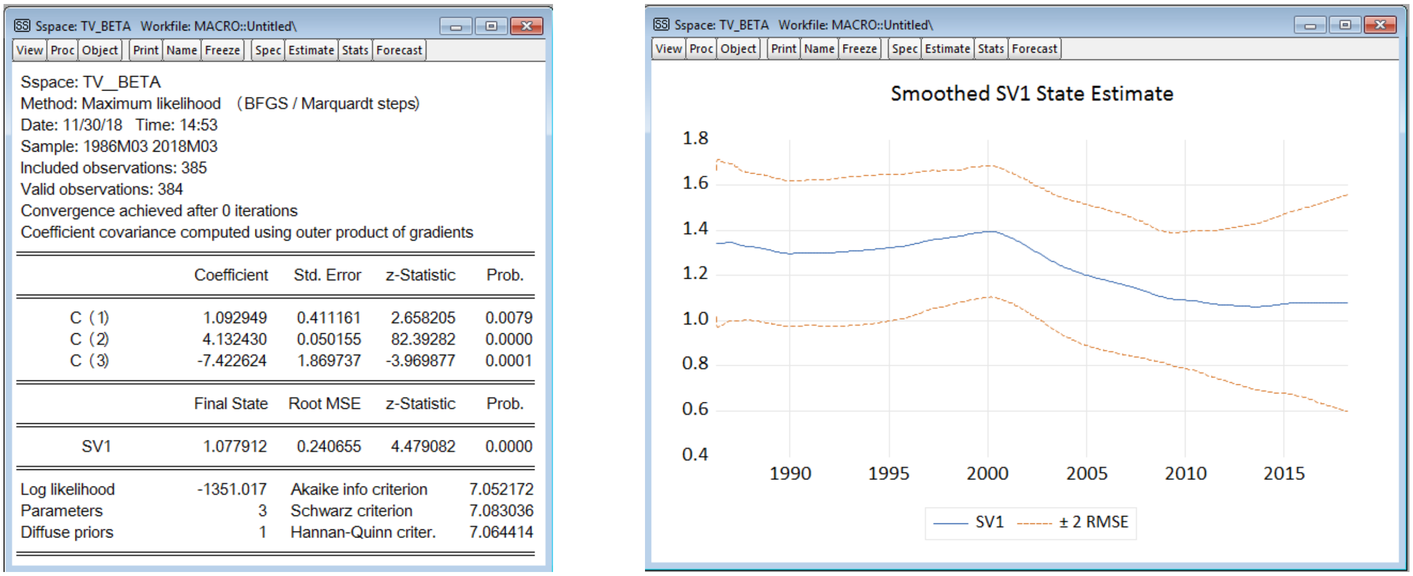


Figure 39: State Space Model Formulation in EViews

If we now click **Estimate** and **OK** without changing the default estimation settings, we obtain the coefficient estimates in Figure 40, left panel. Remember that **c(1)** will denote the α, while for β we do not obtain estimates as it is time-varying and results from the filtering process. However, if we click **View/State Views/Graph State Series...** we can choose to plot the smoothed state estimates (Figure 40, right panel). We observe that while β was rising to a peak of around 1.4 in 2000, it since fell to below 1.1, suggesting that the market risk of Microsoft stock has diminished over time.



Sspace: TV_BETA
Method: Maximum likelihood (BFGS / Marquardt steps)
Date: 11/30/18 Time: 14:53
Sample: 1986M03 2018M03
Included observations: 385
Valid observations: 384
Convergence achieved after 0 iterations
Coefficient covariance computed using outer product of gradients

	Coefficient	Std. Error	z-Statistic	Prob.
C (1)	1.092949	0.411161	2.658205	0.0079
C (2)	4.132430	0.050155	82.39282	0.0000
C (3)	-7.422624	1.869737	-3.969877	0.0001

	Final State	Root MSE	z-Statistic	Prob.
SV1	1.077912	0.240655	4.479082	0.0000

Log likelihood	-1351.017	Akaike info criterion	7.052172
Parameters	3	Schwarz criterion	7.083036
Diffuse priors	1	Hannan-Quinn criter.	7.064414

Figure 40: State Space Model Estimation Using the Kalman Filter and Graph for State Variable

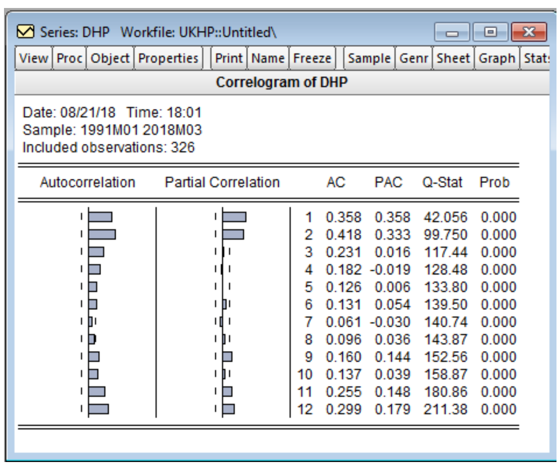
11 Constructing ARMA Models in EViews

Reading: Brooks (2019, sections 6.4–6.7)

This example uses the monthly UK house price series which was already incorporated in an EViews workfile in section 2. There were a total of 326 monthly observations running from February 1991 (recall that the January observation was 'lost' in constructing the lagged value) to March 2018 for the percentage change in house price series. The objective of this exercise is to build an ARMA model for the house price changes. Recall that there are three stages involved: identification, estimation and diagnostic checking. The first stage is carried out by looking at the autocorrelation and partial autocorrelation coefficients to identify any structure in the data.

11.1 Estimating Autocorrelation Coefficients

Double click on the **DHP series** and then click **View** and choose **Correlogram ...**. In the 'Correlogram Specification' window, choose **Level** (since the series we are investigating has already been transformed into percentage returns or percentage changes) and in the 'Lags to include' box, type **12**. Click on **OK**. The output, including relevant test statistics, is given in Figure 41.



	AC	PAC	Q-Stat	Prob
1	0.358	0.358	42.056	0.000
2	0.418	0.333	99.750	0.000
3	0.231	0.016	117.44	0.000
4	0.182	-0.019	128.48	0.000
5	0.126	0.006	133.80	0.000
6	0.131	0.054	139.50	0.000
7	0.061	-0.030	140.74	0.000
8	0.096	0.036	143.87	0.000
9	0.160	0.144	152.56	0.000
10	0.137	0.039	158.87	0.000
11	0.255	0.148	180.86	0.000
12	0.299	0.179	211.38	0.000

Figure 41: Estimating the Correlogram

It is clearly evident that the series is quite persistent given that it is already in percentage change form. The autocorrelation function dies away rather slowly. Only the first two partial autocorrelation coefficients appear strongly significant. The numerical values of the autocorrelation and partial autocorrelation coefficients at lags 1–12 are given in the second and third columns of the output, with the lag length given in the first column.

The penultimate column of output gives the statistic resulting from a Ljung–Box test with number of lags in the sum equal to the row number (i.e., the number in the third column). The test statistics will follow a $\chi^2(1)$ for the first row, a $\chi^2(2)$ for the second row, and so on. p-values associated with these test statistics are given in the last column.

Remember that, as a rule of thumb, a given autocorrelation coefficient is classed as significant if it is outside a $\pm 1.96 \times 1/(T)^{\frac{1}{2}}$ band, where T is the number of observations. In this case, it would imply that a correlation coefficient is classed as significant if it is bigger than approximately 0.11 or smaller than -0.11. The band is of course wider when the sampling frequency is monthly, as it is here, rather than daily where there would be more observations. It can be deduced that the first six autocorrelation coefficients (then nine through twelve) and the first two partial autocorrelation coefficients (then nine, eleven and twelve) are significant under this rule. Since the first acf coefficient is highly significant, the

Ljung–Box joint test statistic rejects the null hypothesis of no autocorrelation at the 1% level for all numbers of lags considered. It could be concluded that a mixed ARMA process could be appropriate, although it is hard to precisely determine the appropriate order given these results. In order to investigate this issue further, the information criteria are now employed.

11.2 Using Information Criteria to Decide on Model Orders

As demonstrated above, deciding on the appropriate model orders from autocorrelation functions could be very difficult in practice. An easier way is to choose the model order that minimises the value of an information criterion. An important point to note is that books and statistical packages often differ in their construction of the test statistic. For example, the formulae given in Brooks (2019, Chapter 6) for Akaike's and Schwarz's Information Criteria are

$$\text{AIC} = \ln(\hat{\sigma}^2) + \frac{2k}{T} \tag{4}$$

$$\text{SBIC} = \ln(\hat{\sigma}^2) + \frac{k}{T}\ln(T) \tag{5}$$

where $\hat{\sigma}^2$ is the estimator of the variance of regressions disturbances u_t, k is the number of parameters and T is the sample size. When using the criterion based on the estimated standard errors, the model with the lowest value of *AIC* and *SBIC* should be chosen. However, EViews uses a formulation of the test statistic derived from the log-likelihood function value based on a maximum likelihood estimation (see Chapter 9 of Brooks, 2019). The corresponding EViews formulae are

$$\text{AIC}_\ell = -\frac{2\ell}{T} + \frac{2k}{T} \tag{6}$$

$$\text{SBIC}_\ell = -\frac{2\ell}{T} + \frac{k}{T}\ln(T), \tag{7}$$

where $\ell = -\frac{T}{2}\left(1 + \ln(2\pi) + \ln\left(\frac{\hat{u}'\hat{u}}{T}\right)\right)$ and $\hat{u}$ are the residuals of the regression. Unfortunately, this modification is not benign, since it affects the relative strength of the penalty term compared with the error variance, sometimes leading different packages to select different model orders for the same data and criterion.

Suppose that it is thought that ARMA models from order (0,0) to (5,5) are plausible for the house price changes. This would entail considering thirty-six models (ARMA(0,0), ARMA(1,0), ..., ARMA(5,5)), i.e., from zero up to five lags in both the autoregressive and moving average terms.

In EViews, this can be done by separately estimating each of the models and noting down the value of the information criteria in each case.[9] This would be done in the following way. From the EViews main menu, click on **Quick** and choose **Estimate Equation ...** . EViews will open an Equation Specification window (Figure 42, left panel). In the Equation Specification editor, type, for example

dhp c ar(1) ma(1)

For the estimation settings, select **LS – Least Squares (NLS and ARMA)**, select the whole sample, and click **OK** – this will specify an ARMA(1,1). In the options tab (Figure 42, right panel), we can specify the standard errors, the optimisation method and the options on the algorithm. We untick the **d.f. Adjustment** box, other than this we keep the default, which uses Outer product gradient (OPG) standard errors and the Berndt–Hall–Hall–Hausman (BHHH) algorithm. The output is given in Figure 43.

[9] Alternatively, any reader who knows how to write programs in EViews could set up a structure to loop over the model orders and calculate all the values of the information criteria together. We will discuss programming in EViews later in this guide.

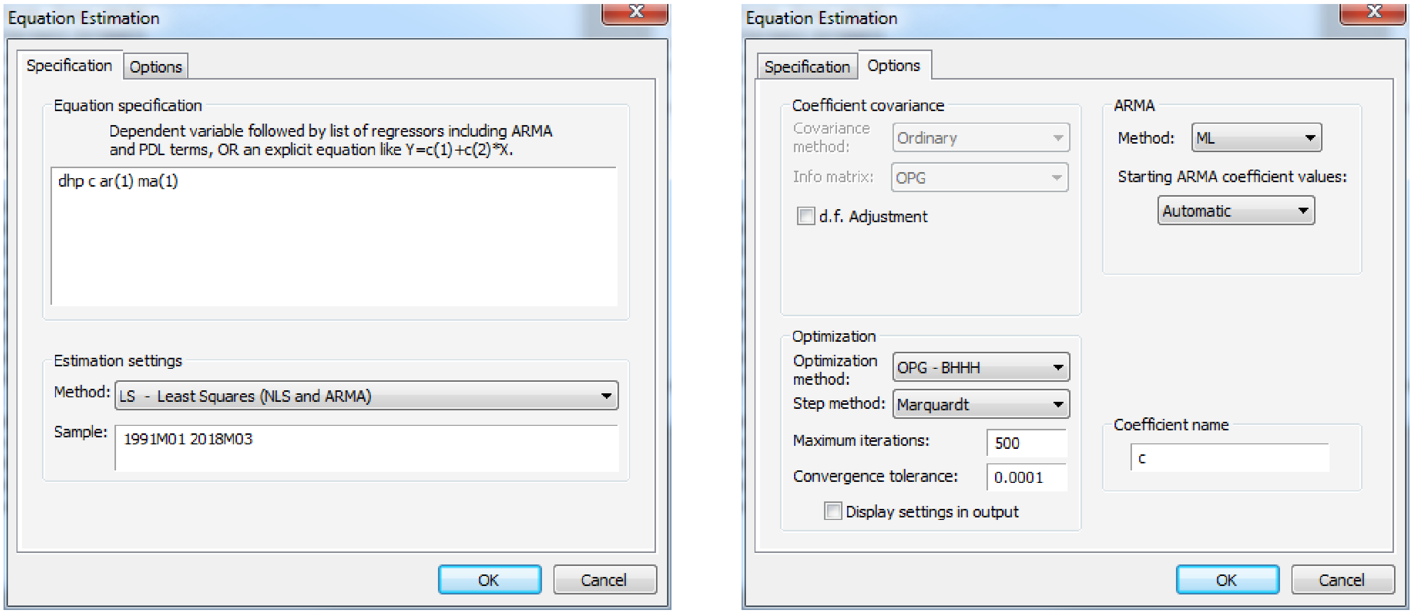
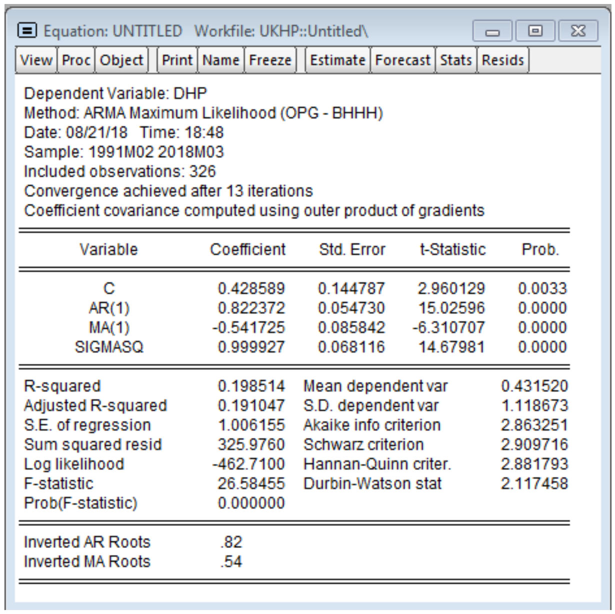


Figure 42: Estimating an ARMA(1,1) Model

In reality it is very difficult to interpret the parameter estimates in the sense of, for example, saying, 'a one unit increase in x leads to a β unit increase in y'. In part because the construction of ARMA models is not based on any economic or financial theory, it is often best not to even try to interpret the individual parameter estimates, but rather to examine the plausibility of the model as a whole and to determine whether it describes the data well and produces accurate forecasts (if this is the objective of the exercise, which it often is).

The inverses of the AR and MA roots of the characteristic equation are also shown. These can be used to check whether the process implied by the model is stationary and invertible. For the AR and MA parts of the process to be stationary and invertible, respectively, the inverted roots in each case must be smaller than one in absolute value, which they are in this case. Note also that the roots are identical to (absolute values of) the values of the parameter estimates in this case (since there is only one AR term and one MA term) – in general, this will not be the case when there are more lags. The header for the EViews output for ARMA models states the number of iterations that have been used in the model estimation process. This shows that, in fact, an iterative numerical optimisation procedure has been employed to estimate the coefficients.



Equation: UNTITLED Workfile: UKHP::Untitled\

View | Proc | Object | Print | Name | Freeze | Estimate | Forecast | Stats | Resids

Dependent Variable: DHP
Method: ARMA Maximum Likelihood (OPG - BHHH)
Date: 08/21/18 Time: 18:48
Sample: 1991M02 2018M03
Included observations: 326
Convergence achieved after 13 iterations
Coefficient covariance computed using outer product of gradients

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	0.428589	0.144787	2.960129	0.0033
AR(1)	0.822372	0.054730	15.02596	0.0000
MA(1)	-0.541725	0.085842	-6.310707	0.0000
SIGMASQ	0.999927	0.068116	14.67981	0.0000

R-squared	0.198514	Mean dependent var	0.431520
Adjusted R-squared	0.191047	S.D. dependent var	1.118673
S.E. of regression	1.006155	Akaike info criterion	2.863251
Sum squared resid	325.9760	Schwarz criterion	2.909716
Log likelihood	-462.7100	Hannan-Quinn criter.	2.881793
F-statistic	26.58455	Durbin-Watson stat	2.117458
Prob(F-statistic)	0.000000		

Inverted AR Roots	.82
Inverted MA Roots	.54

Figure 43: Regression Results for an ARMA(1,1) Model

Repeating these steps for the other ARMA models would give all of the required values for the information criteria. To give just one more example, in the case of an ARMA(5,5), the following would be typed in the Equation Specification editor box:

dhp c ar(1) ar(2) ar(3) ar(4) ar(5) ma(1) ma(2) ma(3) ma(4) ma(5)

Note that, in order to estimate an ARMA(5,5) model, it is necessary to write out the whole list of terms as above rather than to simply write, for example, 'dhp c ar(5) ma(5)', which would give a model with a fifth lag of the dependent variable and a fifth lag of the error term, but no other variables. Table 2 reports the values of all of the Akaike and Schwarz information criteria calculated using EViews.

Table 2: Information Criteria for ARMA(p,q) Models

			AIC			
p/q	0	1	2	3	4	5
0	3.065226	2.998166	2.869657	2.857222	2.853665	2.852609
1	2.941078	2.863251	2.839413	2.833708	2.835889	2.850636
2	2.829632	2.835607	2.841235	2.834579	2.840422	2.830027
3	2.835588	2.841592	2.847188	2.832082	2.838741	2.818035
4	2.841292	2.847419	2.826194	2.841803	2.844853	2.816236
5	2.847378	2.840240	**2.806240**	2.828046	2.831763	2.845552
			SBIC			
p/q	0	1	2	3	4	5
0	3.076842	3.033014	2.916122	2.915304	2.923363	2.933923
1	2.975927	2.909716	2.897495	2.903406	2.917203	2.943566
2	**2.876097**	2.893689	2.910933	2.915892	2.933352	2.934573
3	2.893669	2.911289	2.928502	2.925012	2.943287	2.934197
4	2.910989	2.928733	2.919124	2.946349	2.961015	2.944015
5	2.928692	2.933170	2.910787	2.944209	2.959542	2.984947

So which model actually minimises the two information criteria? In this case, the criteria choose different models: *AIC* selects an ARMA(5,2), while *SBIC* selects the smaller ARMA(2,0) model – i.e., an AR(2). These chosen models are highlighted in bold in Table 2. It will always be the case that *SBIC* selects a model that is at least as small (i.e., with fewer or the same number of parameters) as *AIC*, because the former criterion has a stricter penalty term. This means that *SBIC* penalises the incorporation of additional terms more heavily. Many different models provide almost identical values of the information criteria, suggesting that the chosen models do not provide particularly sharp characterisations of the data and that a number of other specifications would fit the data almost as well. Note that we could also have employed the Hannan–Quinn criterion and, as an exercise, you might determine the appropriate model order using that approach too.

12 Forecasting Using ARMA Models

Reading: Brooks (2019, Section 6.8)

Once a specific model order has been chosen and the model estimated for a particular set of data, it may be of interest to use the model to forecast future values of the series. Suppose that the AR(2) model selected for the house price percentage changes series were estimated using observations February 1991–December 2015, leaving 27 remaining observations to construct forecasts for and to test forecast accuracy (for the period January 2016–March 2018).

Once the required model has been estimated and EViews has opened a window displaying the output, click on the **Forecast** icon. In this instance, the sample range to forecast would be entered as 2016M01–2018M03. There are two methods available in EViews for constructing forecasts: dynamic and static. Select the option **Dynamic** to calculate multi-step forecasts starting from the first period in the forecast sample or **Static** to calculate a sequence of one-step-ahead forecasts, rolling the sample forwards one observation after each forecast. There is also a box that allows you to choose to use actual rather than forecasted values for lagged dependent variables for the out-of-sample observations. Figure 44 shows the window to enter these options while the outputs for the dynamic and static forecasts are given in Figure 45. By default, EViews will store the forecasts in a new series DHPF. If you examine this series you will see that all of the observations up to and including December 2015 are the same as the original series (since we did not forecast those data points) but the data points from January 2016 onwards represent the forecasts from the AR(2).

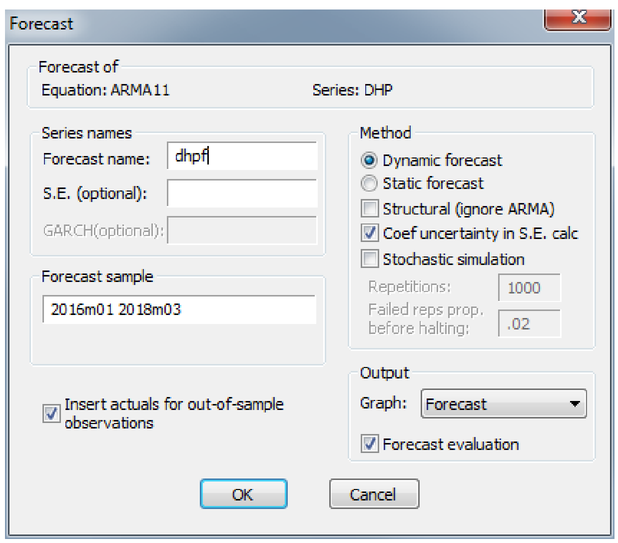


Figure 44: The Options Available When Producing Forecasts

The forecasts are plotted using the continuous line, while a confidence interval is given by the two dotted lines in each case. For the dynamic forecasts, it is clearly evident that the forecasts quickly converge upon the long-term unconditional mean value as the horizon increases. Of course, this does not occur with the series of one-step-ahead forecasts produced by the 'static' command. Several other useful measures concerning the forecast errors are displayed in the plot box, including the square root of the mean squared error (RMSE), the MAE, the MAPE and Theil's U-statistic. The MAPE for the dynamic and static forecasts for DHP are well over 100% in both cases, which can sometimes happen for the reasons outlined above. This indicates that the model forecasts are unable to account for much of the variability of the out-of-sample part of the data. This is to be expected as, forecasting changes in house prices, along with the changes in the prices of any other assets, is difficult!

Figure 45: Dynamic and Static Forecasts for the Percentage Changes in House Prices

EViews provides another piece of useful information – a decomposition of the forecast errors. The mean squared forecast error can be decomposed into a bias proportion, a variance proportion and a covariance proportion. The *bias component* measures the extent to which the mean of the forecasts is different to the mean of the actual data (i.e., whether the forecasts are biased). Similarly, the *variance component* measures the difference between the variation of the forecasts and the variation of the actual data, while the *covariance component* captures any remaining unsystematic part of the forecast errors. As one might have expected, the forecasts are not biased. Accurate forecasts would be unbiased and also have a small variance proportion, so that most of the forecast error should be attributable to the covariance (unsystematic or residual) component. A robust forecasting exercise would of course employ a longer out-of-sample period than the two years or so used here, would perhaps employ several competing models in parallel, and would also compare the accuracy of the predictions by examining the error measures given in the box after the forecast plots.

13 Exponential Smoothing Models

Reading: Brooks (2019, Section 6.9)

This class of models can be easily estimated in EViews by double clicking on the desired variable in the workfile, so that the spreadsheet for that variable appears, and selecting **Proc** on the button bar for that variable and then **Exponential Smoothing/Simple Exponential Smoothing. . .** . The screen with options will appear as in Figure 46.

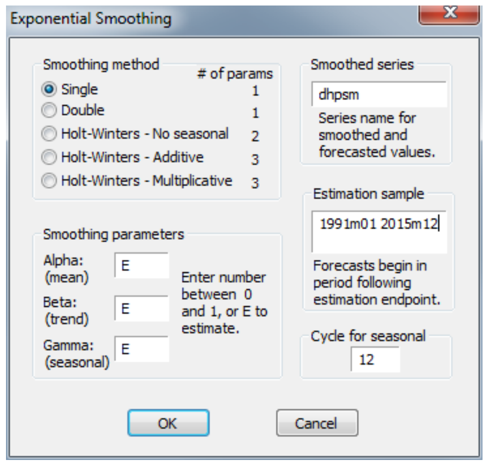


Figure 46: Estimating Exponential Smoothing Models

There is a variety of smoothing methods available, including single and double, or various methods to allow for seasonality and trends in the data. Select **Single** (exponential smoothing), which is the only smoothing method that has been discussed in this guide, and specify the estimation sample period as **1991M1–2015M12** to leave 27 observations for out-of-sample forecasting. Clicking **OK** will give the results in Figure 47.

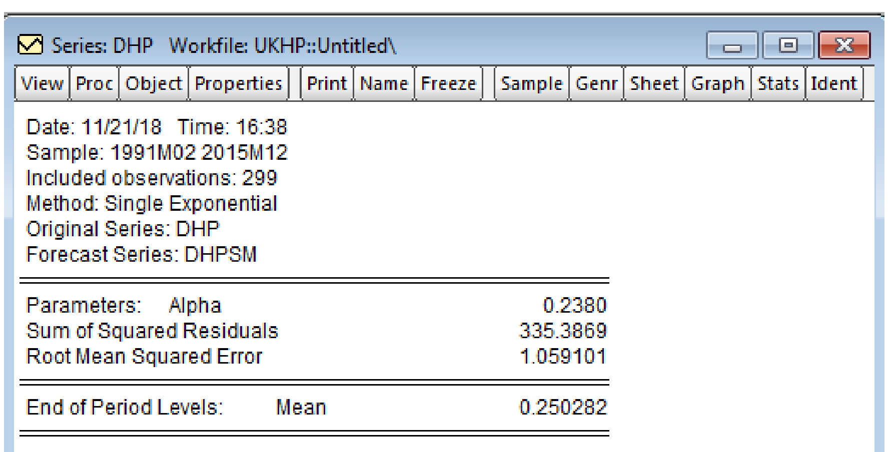


Figure 47: Exponential Smoothing Model Results

The output includes the value of the estimated smoothing coefficient (0.2380 in this case), together with the RSS for the in-sample estimation period and the RMSE for the 27 forecasts. The final in-sample smoothed value will be the forecast for those 27 observations (which in this case would be 0.250282). EViews has automatically saved the smoothed values (i.e., the model fitted values) and the forecasts in a series called 'DHPSM'.

14 Simultaneous Equations Modelling

Reading: Brooks (2019, Section 7.5–7.9)

What is the relationship between inflation and stock returns? Holding stocks is often thought to provide a good hedge against inflation, since the payments to equity holders are not fixed in nominal terms and represent a claim on real assets (unlike the coupons on bonds, for example). However, the majority of empirical studies that have investigated the sign of this relationship have found it to be negative. Various explanations of this puzzling empirical phenomenon have been proposed, including a link through real activity, so that real activity is negatively related to inflation but positively related to stock returns and therefore stock returns and inflation vary positively. Clearly, inflation and stock returns ought to be simultaneously related given that the rate of inflation will affect the discount rate applied to cashflows and therefore the value of equities, but the performance of the stock market may also affect consumer demand and therefore inflation through its impact on householder wealth (perceived or actual).[10]

This simple example uses the same macroeconomic data as used previously to estimate this relationship simultaneously. Suppose (without justification) that we wish to estimate the following model, which does not allow for dynamic effects or partial adjustments and does not distinguish between expected and unexpected inflation

$$inflation_t = \alpha_0 + \alpha_1 returns_t + \alpha_2 dcredit_t + \alpha_3 dprod_t + \alpha_4 dmoney + u_{1t}, \tag{8}$$

$$returns_t = \beta_0 + \beta_1 dprod_t + \beta_2 dspread_t + \beta_3 inflation_t + \beta_4 rterm_t + u_{2t}, \tag{9}$$

where 'returns' are stock returns and all of the other variables are defined as previously.

It is evident that there is feedback between the two equations since the *inflation* variable appears in the *returns* equation and vice versa. Are the equations identified? Since there are two equations, each will be identified if one variable is missing from that equation. Equation (8), the inflation equation, omits two variables. It does not contain the default spread or the term spread, and so is over-identified. Equation (9), the stock returns equation, omits two variables as well – the consumer credit and money supply variables – and so is over-identified too. Two-stage least squares (2SLS) is therefore the appropriate technique to use.

In EViews, to do this we need to specify a list of instruments, which would be all of the variables from the reduced form equation. In this case, the reduced form equations would be

$$inflation = f(constant, dprod, dspread, rterm, dcredit, rterm, dmoney) \tag{10}$$

$$returns = g(constant, dprod, dspread, rterm, dcredit, rterm, dmoney) \tag{11}$$

We can perform both stages of 2SLS in one go, but by default, EViews estimates each of the two equations in the system separately. To do this, click **Quick**, **Estimate Equation** and then select **TSLS – Two Stage Least Squares (TSNLS and ARMA)** from the list of estimation methods. Then fill in the dialog box as in the left panel of Figure 48 to estimate the inflation equation.

[10]Crucially, good econometric models are based on solid financial theory. This model is clearly not, but represents a simple way to illustrate the estimation and interpretation of simultaneous equations models using EViews with freely available data!

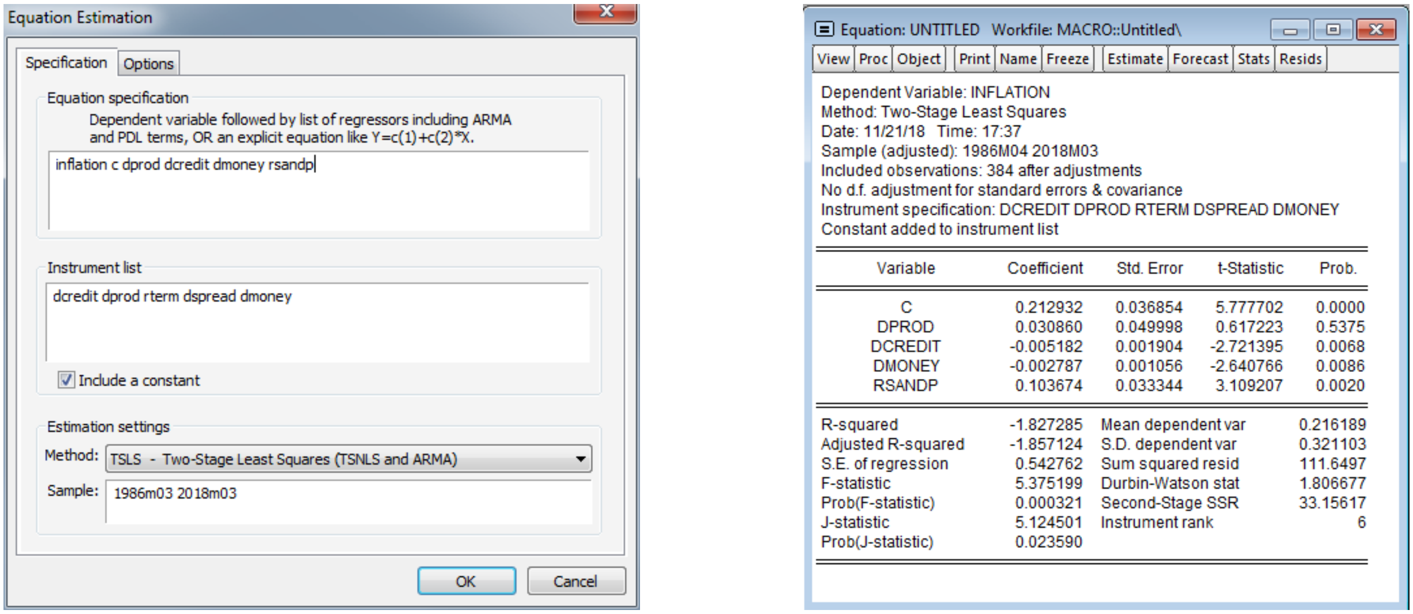


Figure 48: Estimating the Inflation Equation

Thus the format of writing out the variables in the first window is as usual, and the full structural equation for inflation as a dependent variable should be specified here. In the instrument list, include every variable from the reduced form equation, including the constant, and click **OK**. The results would then appear as in the right panel of Figure 48.

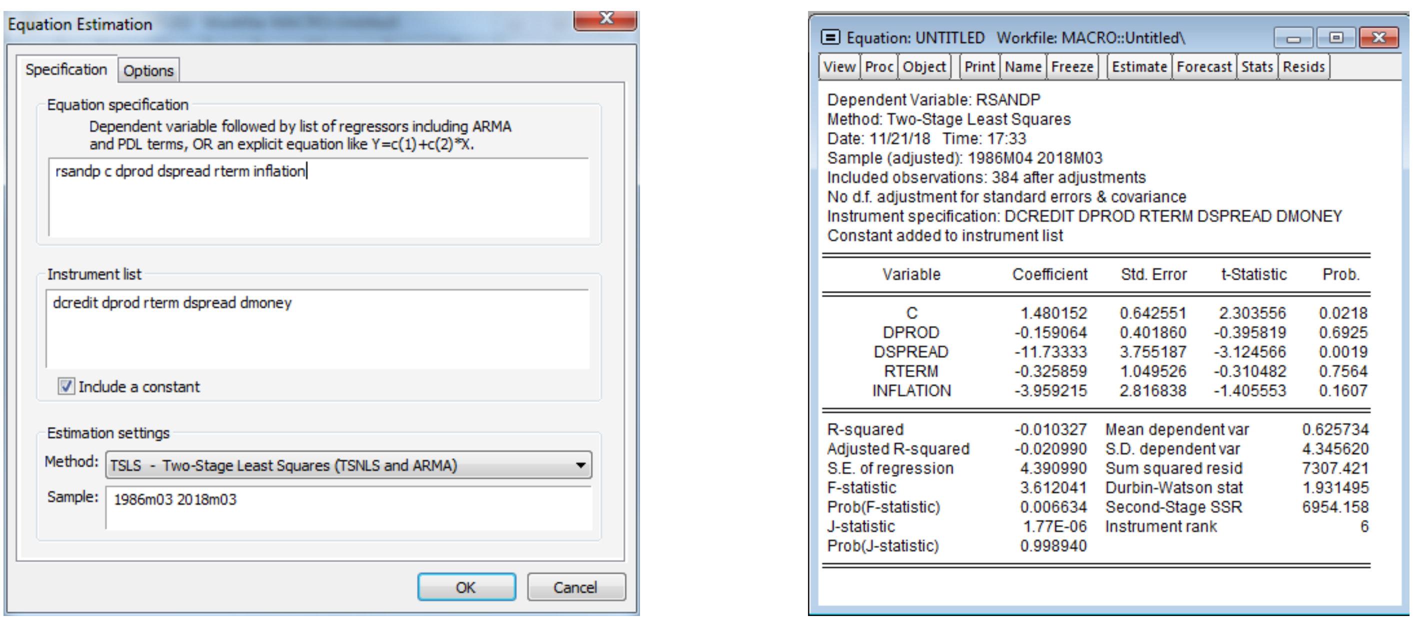


Figure 49: Estimating the Returns Equation

Similarly, the dialog box for the rsandp equation would be specified as in Figure 49, left panel and the output for the returns equation is as in Figure 49, right panel.

The results overall show that the stock index returns are a positive and significant determinant of inflation (changes in the money supply negatively affect inflation), while inflation has a negative effect on the stock market, albeit not significantly so. The R^2 and $\bar{R}^2$ values from the inflation equation are also negative, so should be interpreted with caution. As the EViews *User's Guide* warns, this can sometimes happen even when there is an intercept in the regression. The J-statistic is essentially a transformed version of the residual sum of squares that evaluates the model fit.

It may also be of relevance to conduct a Hausman test for the endogeneity of the inflation and stock return variables. To do this, **estimate the reduced form equations** and **save the residuals**. Then **create series of fitted values** by constructing new variables which are equal to the actual values minus the residuals. Call the fitted value series **inflation_fit** and **rsandp_fit**. Then **estimate the structural equations** (separately), adding the fitted values from the relevant reduced form equations. The two sets of variables (in EViews format, with the dependent variables first followed by the lists of independent variables) are as follows.
For the inflation equation:

inflation c dprod dcredit dmoney rsandp rsandp_fit

and for the stock return equation:

rsandp c dprod dspread rterm inflation inflation_fit

The conclusion is that the inflation fitted value term is not significant in the stock return equation and so inflation can be considered exogenous for stock returns. Thus it would be valid to simply estimate this equation (minus the fitted value term) on its own using OLS. But the fitted stock return term is significant in the inflation equation, suggesting that stock returns are endogenous.

15 The Generalised Method of Moments

Reading: Brooks (2019, Section 14.4)

Instead of the Two-Stage Least Squares Method, the instrumental variables approach can also be realised using the Generalised Method of Moments. We therefore follow the same steps as in the previous section, but in the specification window we choose the option **GMM – Generalized Method of Moments** as 'Method'. Now we can specify the equation and the instruments as before. Figure 50 shows the specifications for the inflation equation (8) in the left panel and analogous for the return equation (9) in the right panel. Note, that we also changed the **'Estimation weighting matrix'** drop down menu to **'White'**. This weighting matrix is optimal under the setting of heteroscedastic error terms.

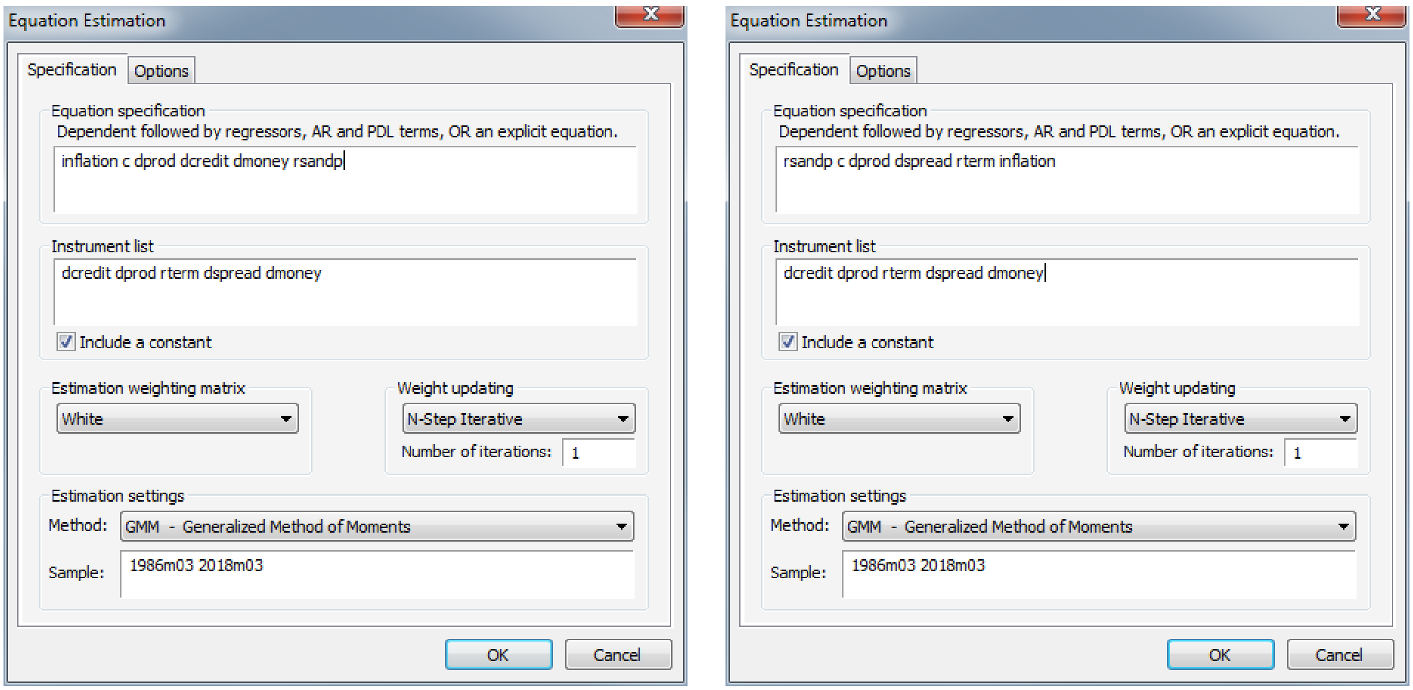
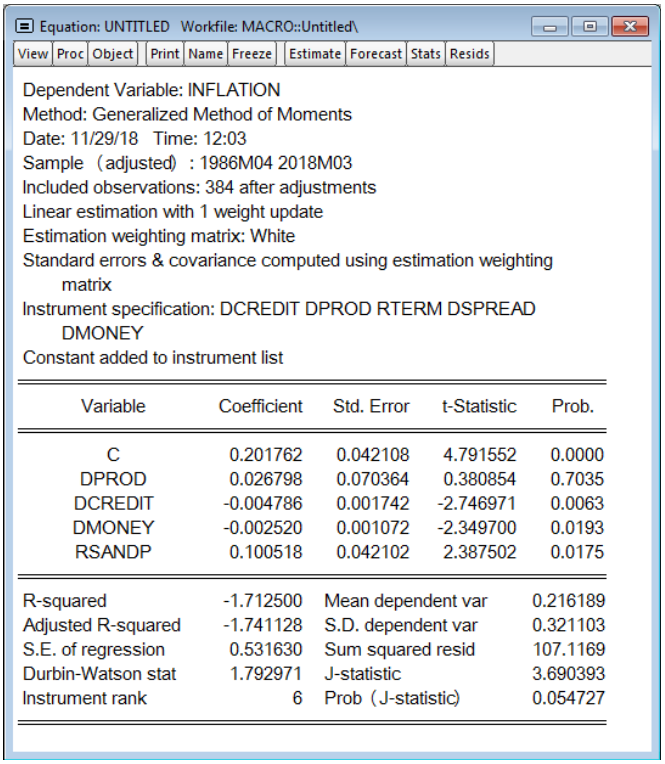


Figure 50: Specifying the Instrumental Variables Approach with the GMM Estimator

Clicking **OK** yields the results in Figure 51. Note that the results are qualitatively similar to those from the previous section, but show small differences. To obtain the exact results, we would need to specify the estimation weighting matrix as **'Two-Stage Least Squares'**. As such, the GMM method is only a specific way to estimate the same regression. Unfortunately, EViews does not provide a tool to specify the moment conditions separately, which would be necessary to use the GMM estimator for a different purpose.



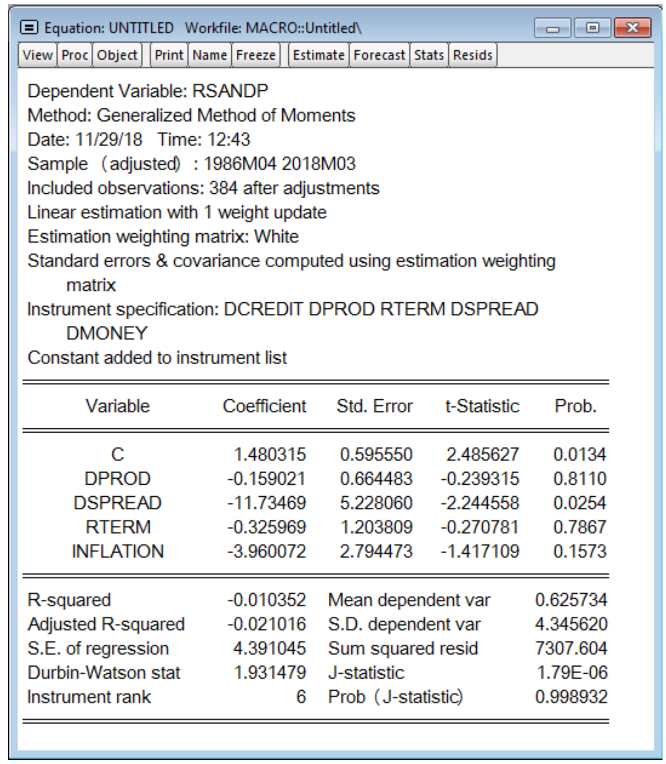
Equation: UNTITLED Workfile: MACRO::Untitled\

View | Proc | Object | Print | Name | Freeze | Estimate | Forecast | Stats | Resids

Dependent Variable: INFLATION
Method: Generalized Method of Moments
Date: 11/29/18 Time: 12:03
Sample (adjusted) : 1986M04 2018M03
Included observations: 384 after adjustments
Linear estimation with 1 weight update
Estimation weighting matrix: White
Standard errors & covariance computed using estimation weighting matrix
Instrument specification: DCREDIT DPROD RTERM DSPREAD DMONEY
Constant added to instrument list

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	0.201762	0.042108	4.791552	0.0000
DPROD	0.026798	0.070364	0.380854	0.7035
DCREDIT	-0.004786	0.001742	-2.746971	0.0063
DMONEY	-0.002520	0.001072	-2.349700	0.0193
RSANDP	0.100518	0.042102	2.387502	0.0175

R-squared	-1.712500	Mean dependent var	0.216189
Adjusted R-squared	-1.741128	S.D. dependent var	0.321103
S.E. of regression	0.531630	Sum squared resid	107.1169
Durbin-Watson stat	1.792971	J-statistic	3.690393
Instrument rank	6	Prob (J-statistic)	0.054727



Equation: UNTITLED Workfile: MACRO::Untitled\

View | Proc | Object | Print | Name | Freeze | Estimate | Forecast | Stats | Resids

Dependent Variable: RSANDP
Method: Generalized Method of Moments
Date: 11/29/18 Time: 12:43
Sample (adjusted) : 1986M04 2018M03
Included observations: 384 after adjustments
Linear estimation with 1 weight update
Estimation weighting matrix: White
Standard errors & covariance computed using estimation weighting matrix
Instrument specification: DCREDIT DPROD RTERM DSPREAD DMONEY
Constant added to instrument list

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	1.480315	0.595550	2.485627	0.0134
DPROD	-0.159021	0.664483	-0.239315	0.8110
DSPREAD	-11.73469	5.228060	-2.244558	0.0254
RTERM	-0.325969	1.203809	-0.270781	0.7867
INFLATION	-3.960072	2.794473	-1.417109	0.1573

R-squared	-0.010352	Mean dependent var	0.625734
Adjusted R-squared	-0.021016	S.D. dependent var	4.345620
S.E. of regression	4.391045	Sum squared resid	7307.604
Durbin-Watson stat	1.931479	J-statistic	1.79E-06
Instrument rank	6	Prob (J-statistic)	0.998932

Figure 51: Estimation Results from GMM Estimation for the Inflation and Returns Equations

16 Vector Autoregressive (VAR) Models

Reading: Brooks (2019, Section 7.10)

By way of illustration, a VAR is estimated in order to examine whether there are lead–lag relationships for the returns to three exchange rates against the US dollar – the Euro, the British pound and the Japanese yen. The data are daily and run from 14 December 1998 to 7 March 2018, giving a total of 7,142 observations. The data are contained in the Excel file 'currencies.xls'. First **Create a new workfile**, called 'currencies.wf1', and import the three currency series. Construct a set of continuously compounded percentage returns called **'reur', 'rgbp' and 'rjpy'.** VAR estimation in EViews can be accomplished by clicking on the **Quick** menu and then **Estimate VAR. . . .** The specification window in Figure 52 appears.

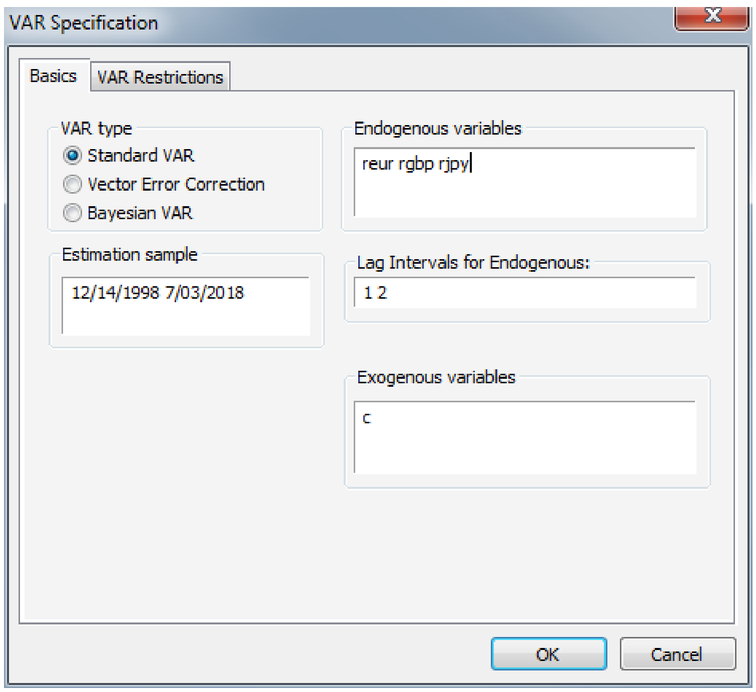


Figure 52: Specification Window for VAR Model Estimation

In the Endogenous variables box, type the three variable names, **reur rgbp rjpy**. In the Exogenous box, leave the default **'c'** and in the Lag Interval box, enter **1 2** to estimate a VAR(2), just as an example. The output appears as in Figure 53, with one column for each equation in the first and second panels, and a single column of statistics that describes the system as a whole in the third. So values of the information criteria are given separately for each equation in the second panel and jointly for the model as a whole in the third.

We will shortly discuss the interpretation of Figure 53, but the example so far has assumed that we know *the appropriate lag length* for the VAR. However, in practice, the first step in the construction of any VAR model, once the variables that will enter the VAR have been decided, will be to determine the appropriate lag length. This can be achieved in a variety of ways, but one of the easiest is to employ a multivariate information criterion.

Var: UNTITLED Workfile: CURRENCIES::Untitled\

View | Proc | Object | Print | Name | Freeze | Estimate | Forecast | Stats | Impulse | Resids

Vector Autoregression Estimates

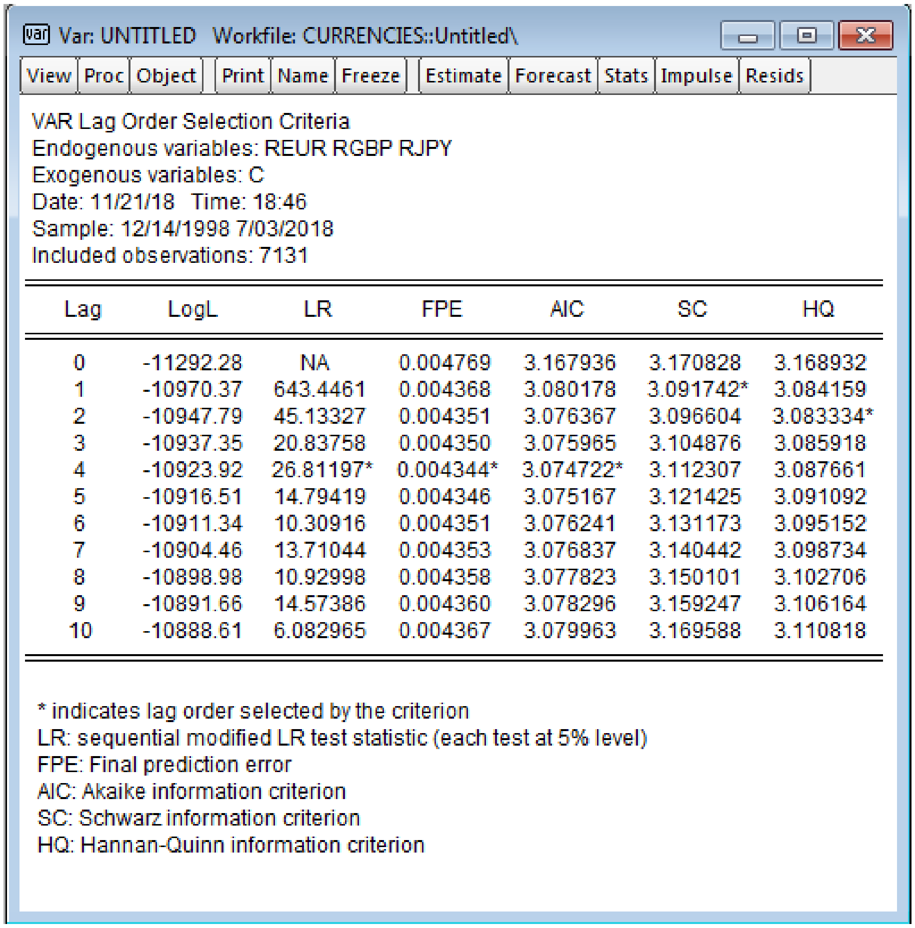
Vector Autoregression Estimates
Date: 11/21/18 Time: 18:36
Sample (adjusted): 12/17/1998 7/03/2018
Included observations: 7139 after adjustments
Standard errors in ( ) & t-statistics in [ ]

	REUR	RGBP	RJPY
REUR(-1)	0.147497	-0.025271	0.041061
	(0.01568)	(0.01406)	(0.01591)
	[9.40804]	[-1.79759]	[2.58120]
REUR(-2)	-0.011808	0.046927	-0.018892
	(0.01566)	(0.01404)	(0.01589)
	[-0.75390]	[3.34123]	[-1.18869]
RGBP(-1)	-0.018356	0.221362	-0.070846
	(0.01704)	(0.01528)	(0.01729)
	[-1.07742]	[14.4902]	[-4.09827]
RGBP(-2)	0.006623	-0.067794	0.024908
	(0.01703)	(0.01527)	(0.01728)
	[0.38885]	[-4.43899]	[1.44124]
RJPY(-1)	-0.007098	-0.039016	0.132457
	(0.01212)	(0.01087)	(0.01230)
	[-0.58563]	[-3.59003]	[10.7707]
RJPY(-2)	-0.005427	0.003287	0.014957
	(0.01212)	(0.01087)	(0.01230)
	[-0.44781]	[0.30245]	[1.21622]
C	0.000137	0.002826	-0.000413
	(0.00544)	(0.00488)	(0.00552)
	[0.02514]	[0.57884]	[-0.07469]
R-squared	0.018244	0.040828	0.020823
Adj. R-squared	0.017418	0.040021	0.019999
Sum sq. resids	1508.762	1213.124	1553.375
S.E. equation	0.459944	0.412427	0.466694
F-statistic	22.08921	50.59682	25.27779
Log likelihood	-4581.789	-3803.314	-4685.804
Akaike AIC	1.285555	1.067464	1.314695
Schwarz SC	1.292294	1.074204	1.321435
Mean dependent	0.000128	0.003377	-0.000660
S.D. dependent	0.464003	0.420936	0.471432

Determinant resid covariance (dof adj.)	0.004339
Determinant resid covariance	0.004326
Log likelihood	-10960.34
Akaike information criterion	3.076435
Schwarz criterion	3.096654
Number of coefficients	21

Figure 53: Results from VAR Model Estimation

In EViews, this can be done easily from the EViews VAR output we have by clicking **View/Lag Structure/Lag Length Criteria. . .** . You will be invited to specify the maximum number of lags to entertain including in the model, and for this example, arbitrarily select **10**. The output in Figure 54 would be observed.



Var: UNTITLED Workfile: CURRENCIES::Untitled\

View | Proc | Object | Print | Name | Freeze | Estimate | Forecast | Stats | Impulse | Resids

VAR Lag Order Selection Criteria
Endogenous variables: REUR RGBP RJPY
Exogenous variables: C
Date: 11/21/18 Time: 18:46
Sample: 12/14/1998 7/03/2018
Included observations: 7131

Lag	LogL	LR	FPE	AIC	SC	HQ
0	-11292.28	NA	0.004769	3.167936	3.170828	3.168932
1	-10970.37	643.4461	0.004368	3.080178	3.091742*	3.084159
2	-10947.79	45.13327	0.004351	3.076367	3.096604	3.083334*
3	-10937.35	20.83758	0.004350	3.075965	3.104876	3.085918
4	-10923.92	26.81197*	0.004344*	3.074722*	3.112307	3.087661
5	-10916.51	14.79419	0.004346	3.075167	3.121425	3.091092
6	-10911.34	10.30916	0.004351	3.076241	3.131173	3.095152
7	-10904.46	13.71044	0.004353	3.076837	3.140442	3.098734
8	-10898.98	10.92998	0.004358	3.077823	3.150101	3.102706
9	-10891.66	14.57386	0.004360	3.078296	3.159247	3.106164
10	-10888.61	6.082965	0.004367	3.079963	3.169588	3.110818

* indicates lag order selected by the criterion
LR: sequential modified LR test statistic (each test at 5% level)
FPE: Final prediction error
AIC: Akaike information criterion
SC: Schwarz information criterion
HQ: Hannan-Quinn information criterion

Figure 54: Information Criteria for VAR Lag Length Selection

EViews presents the values of various information criteria and other methods for determining the lag order. In this case, the Akaike (AIC) and Hannan–Quinn (HQ) criteria select a lag length of two and four as optimal, while the Schwarz criterion chooses a VAR(1). **Estimate a VAR(1)** and examine the results. Does the model look as if it fits the data well? Why or why not?

Next, run a Granger causality test by clicking **View/Lag Structure/ Granger Causality/Block Exogeneity Tests**. The table of statistics will appear immediately as in Figure 55.

The results show only modest evidence of lead-lag interactions between the series. Since we have estimated a tri-variate VAR, three panels are displayed, with one for each dependent variable in the system. There is causality from EUR to GBP and from JPY to GBP that is significant at the 5% level. We also find significant causality at the 5% level from EUR to JPY and GBP to JPY, but no causality from any of the currencies to EUR. These results might be interpreted as suggesting that information is incorporated slightly more quickly in the pound-dollar rate and yen-dollar rates than into the euro-dollar rate.

Var: UNTITLED Workfile: CURRENCIES::Untitled\

View | Proc | Object | Print | Name | Freeze | Estimate | Forecast | Stats | Impulse | Resids

VAR Granger Causality/Block Exogeneity Wald Tests
Date: 11/21/18 Time: 18:58
Sample: 12/14/1998 7/03/2018
Included observations: 7139

Dependent variable: REUR

Excluded	Chi-sq	df	Prob.
RGBP	1.186417	2	0.5526
RJPY	0.625955	2	0.7313
All	1.764027	4	0.7791

Dependent variable: RGBP

Excluded	Chi-sq	df	Prob.
REUR	12.88261	2	0.0016
RJPY	12.92263	2	0.0016
All	28.99111	4	0.0000

Dependent variable: RJPY

Excluded	Chi-sq	df	Prob.
REUR	7.319526	2	0.0257
RGBP	17.12019	2	0.0002
All	17.38201	4	0.0016

Figure 55: Results of Granger Causality Test

It is worth also noting that the term 'Granger causality' is something of a misnomer since a finding of 'causality' does not mean that movements in one variable physically cause movements in another. For example, in the above analysis, if movements in the euro–dollar market were found to Granger-cause movements in the pound–dollar market, this would not have meant that the pound–dollar rate changed as a direct result of, or because of, movements in the euro–dollar market. Rather, causality simply implies a *chronological ordering of movements in the series*. It could validly be stated that movements in the pound–dollar rate appear to lead those of the euro–dollar rate, and so on.

The EViews manual suggests that block F-test restrictions can be performed by estimating the VAR equations individually using OLS and then by using the **View** then **Lag Structure** then **Lag Exclusion Tests**. EViews tests for whether the parameters for a given lag of all the variables in a particular equation can be restricted to zero.

To obtain the impulse responses for the estimated model, simply click the **Impulse** on the button bar above the VAR object and a new dialogue box will appear as in Figure 56. By default, EViews will offer to estimate and plot all of the responses to separate shocks of all of the variables in the order that the variables were listed in the estimation window, using ten steps and confidence intervals generated using analytic formulae. If twenty steps ahead had been selected, with 'combined response graphs', you would see the graphs in the format in Figure 57, left panel. As one would expect given the parameter estimates and the Granger causality test results, only a few linkages between the series are established here. The responses to the shocks are very small, except for the response of a variable to its own shock, and they die down to almost nothing after the first lag. The only exceptions are that the pound (second panel in Figure 57, right panel) and the yen (third graph) both respond to shocks to the euro rate against the dollar.

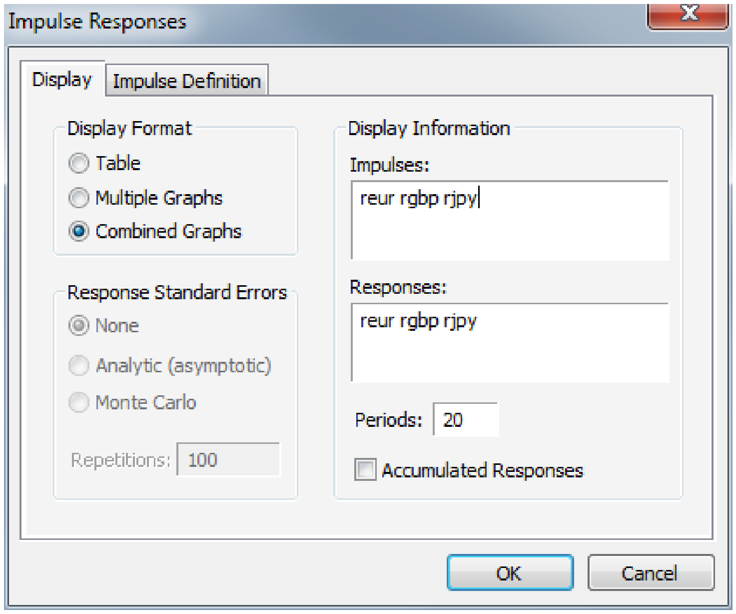


Figure 56: Constructing VAR Impulse Responses

Plots of the variance decompositions can also be generated by clicking on **View** and then **Variance Decomposition. . .** . A similar plot for the variance decompositions would appear as in Figure 57, right panel.

There is little again that can be seen from these variance decomposition graphs apart from the fact that the behaviour is observed to settle down to a steady state very quickly. Interestingly, while the percentage of the errors that is attributable to own shocks is 100% in the case of the euro rate, for the pound, the euro series explains around 40% of the variation in returns, and for the yen, the euro series explains around 7% of the variation and the pound 37%.

We should remember that the ordering of the variables has an effect on the impulse responses and variance decompositions, and when, as in this case, theory does not suggest an obvious ordering of the series, some sensitivity analysis should be undertaken. This can be achieved by clicking on the 'Impulse Definition' tab when the window that creates the impulses is open. A window entitled 'Ordering for Cholesky' should be apparent, and it would be possible to reverse the order of variables or to select any other order desired. For the variance decompositions, the 'Ordering for Cholesky' box is observed in the window for creating the decompositions without having to select another tab.

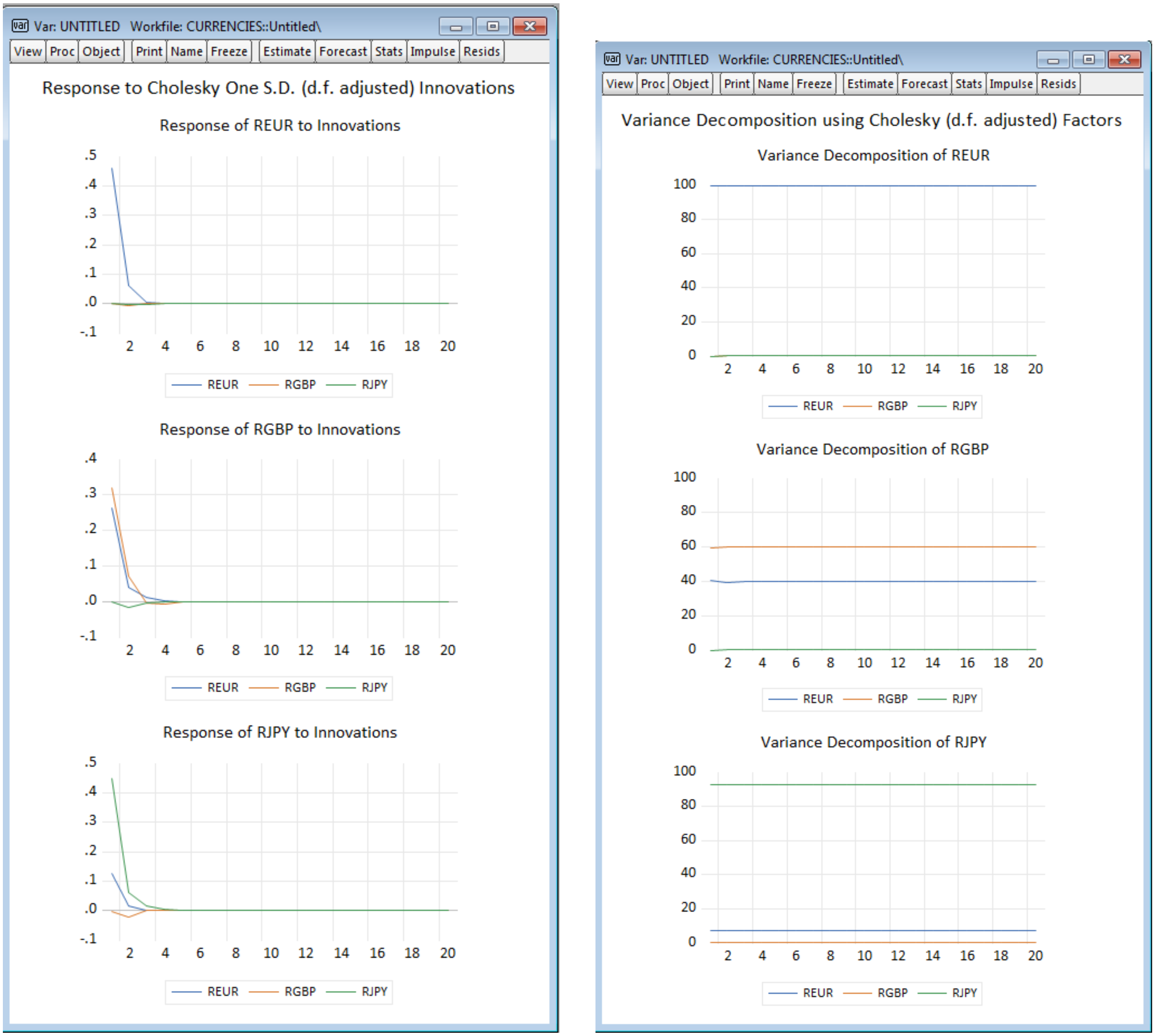


Figure 57: Impulse Response and Variance Decomposition Graphs

17 Testing for Unit Roots

Reading: Brooks (2019, Section 8.1)

This example uses the same data on UK house prices as employed in previous sections of this guide. Assuming that the data have been loaded, and the variables are defined as before, double click on the icon next to the name of the series that you want to perform the unit root test on, so that a spreadsheet appears containing the observations on that series. Open the raw house price series, 'hp' by clicking on the **hp** icon. Next, click on the **View** button on the button bar above the spreadsheet and then **Unit Root Test ...** . You will then be presented with a menu containing various options, as in Figure 58.

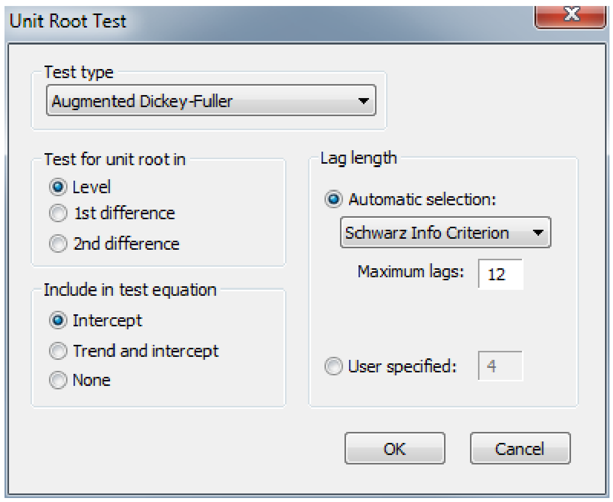
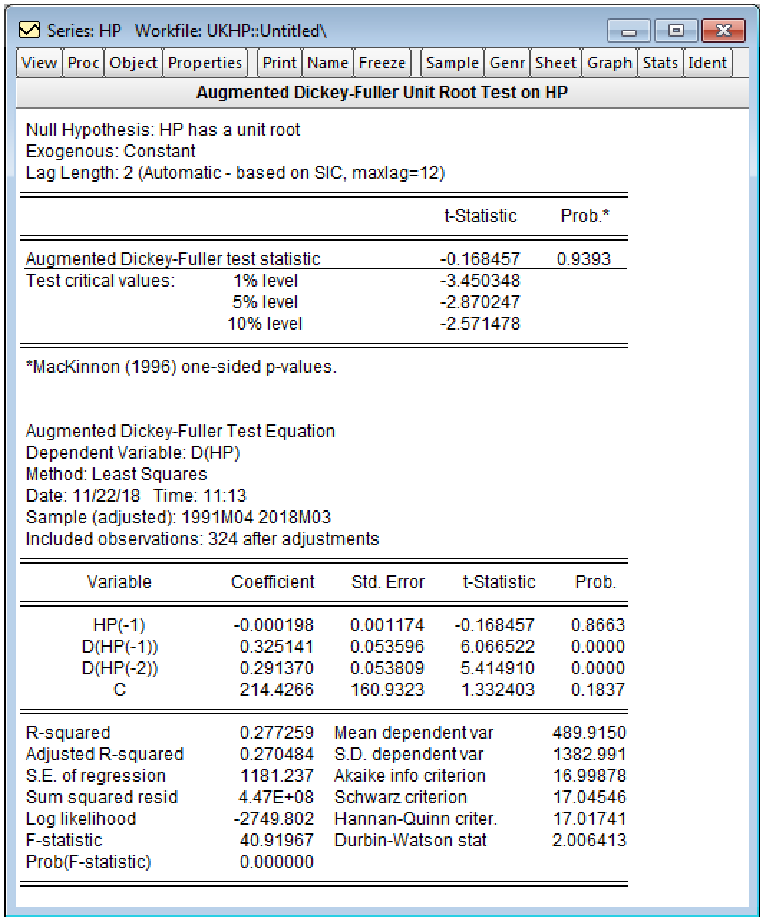


Figure 58: Conducting a Unit Root Test

This will perform an ADF test with up to twelve lags of the dependent variable in a regression equation on the raw data series with a constant but no trend in the test equation. EViews presents a large number of options here – for example, instead of the Dickey–Fuller series, we could run the Phillips–Perron or KPSS tests. Or, if we find that the levels of the series are non-stationary, we could repeat the analysis on the first differences directly from this menu rather than having to create the first differenced series separately. We can also choose between various methods for determining the optimum lag length in an augmented Dickey–Fuller test, with the Schwarz criterion being the default. The results for the raw house price series would appear as in Figure 59.

The value of the test statistic and the relevant critical values given the type of test equation (e.g., whether there is a constant and/or trend included) and sample size, are given in the first panel of the output above. The Schwarz criterion has in this case chosen to include two lags of the dependent variable in the test regression. Clearly, the test statistic is not more negative than the critical value, so the null hypothesis of a unit root in the house price series cannot be rejected. The remainder of the output presents the estimation results. Since one of the independent variables in this regression is non-stationary, it is not appropriate to examine the coefficient standard errors or their t-ratios in the test regression.



Series: HP Workfile: UKHP::Untitled\

View | Proc | Object | Properties | Print | Name | Freeze | Sample | Genr | Sheet | Graph | Stats | Ident

Augmented Dickey-Fuller Unit Root Test on HP

Null Hypothesis: HP has a unit root
Exogenous: Constant
Lag Length: 2 (Automatic - based on SIC, maxlag=12)

		t-Statistic	Prob.*
Augmented Dickey-Fuller test statistic		-0.168457	0.9393
Test critical values:	1% level	-3.450348	
	5% level	-2.870247	
	10% level	-2.571478	

*MacKinnon (1996) one-sided p-values.

Augmented Dickey-Fuller Test Equation
Dependent Variable: D(HP)
Method: Least Squares
Date: 11/22/18 Time: 11:13
Sample (adjusted): 1991M04 2018M03
Included observations: 324 after adjustments

Variable	Coefficient	Std. Error	t-Statistic	Prob.
HP(-1)	-0.000198	0.001174	-0.168457	0.8663
D(HP(-1))	0.325141	0.053596	6.066522	0.0000
D(HP(-2))	0.291370	0.053809	5.414910	0.0000
C	214.4266	160.9323	1.332403	0.1837

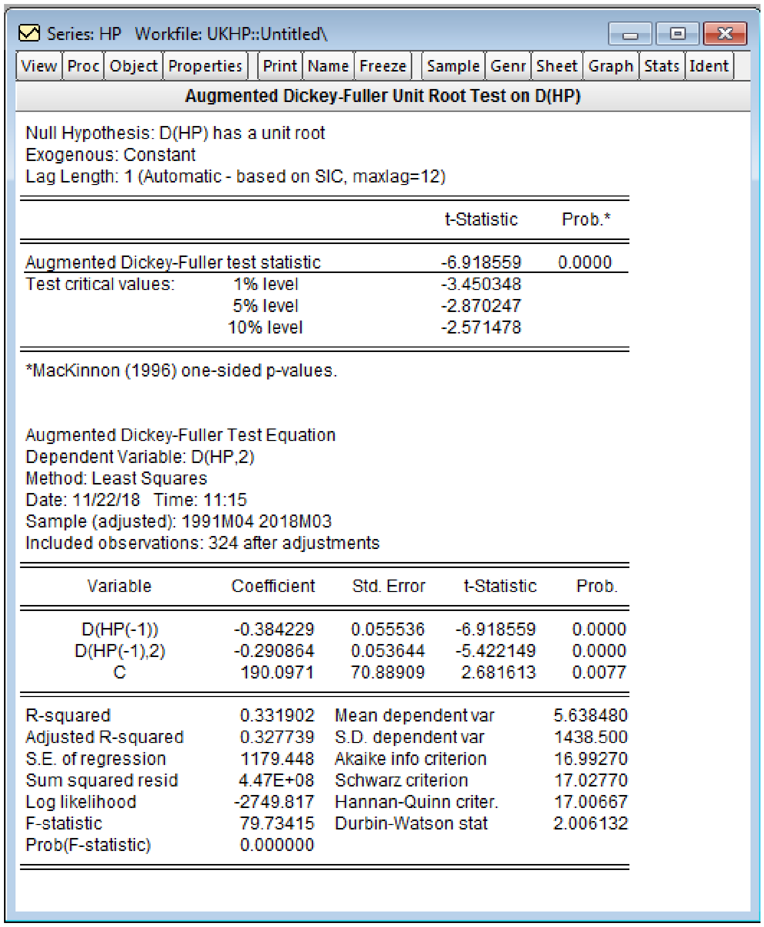
R-squared	0.277259	Mean dependent var	489.9150
Adjusted R-squared	0.270484	S.D. dependent var	1382.991
S.E. of regression	1181.237	Akaike info criterion	16.99878
Sum squared resid	4.47E+08	Schwarz criterion	17.04546
Log likelihood	-2749.802	Hannan-Quinn criter.	17.01741
F-statistic	40.91967	Durbin-Watson stat	2.006413
Prob(F-statistic)	0.000000		

Figure 59: Conducting a Unit Root Test on the Levels of House Prices

Now repeat all of the above steps for the **first difference of the house price series** (use the 'First Difference' option in the unit root testing window rather than using the level of the dhp series). The output would appear as in Figure 60.

In this case, as one would expect, the test statistic is more negative than the critical value and hence the null hypothesis of a unit root in the first differences is convincingly rejected. For completeness, run a unit root test on the **levels of the dhp series**, which are the percentage changes rather than the absolute differences in prices. You should find that these are also stationary.

Finally, run the KPSS test on the hp levels series by selecting it from the **'Test Type'** box in the unit root testing window. You should observe now that the test statistic exceeds the critical value, even at the 1% level, so that the null hypothesis of a *stationary series* is strongly rejected, thus confirming the result of the unit root test previously conducted on the same series.



Series: HP Workfile: UKHP::Untitled\

View | Proc | Object | Properties | Print | Name | Freeze | Sample | Genr | Sheet | Graph | Stats | Ident

Augmented Dickey-Fuller Unit Root Test on D(HP)

Null Hypothesis: D(HP) has a unit root
Exogenous: Constant
Lag Length: 1 (Automatic - based on SIC, maxlag=12)

		t-Statistic	Prob.*
Augmented Dickey-Fuller test statistic		-6.918559	0.0000
Test critical values:	1% level	-3.450348	
	5% level	-2.870247	
	10% level	-2.571478	

*MacKinnon (1996) one-sided p-values.

Augmented Dickey-Fuller Test Equation
Dependent Variable: D(HP,2)
Method: Least Squares
Date: 11/22/18 Time: 11:15
Sample (adjusted): 1991M04 2018M03
Included observations: 324 after adjustments

Variable	Coefficient	Std. Error	t-Statistic	Prob.
D(HP(-1))	-0.384229	0.055536	-6.918559	0.0000
D(HP(-1),2)	-0.290864	0.053644	-5.422149	0.0000
C	190.0971	70.88909	2.681613	0.0077

R-squared	0.331902	Mean dependent var	5.638480
Adjusted R-squared	0.327739	S.D. dependent var	1438.500
S.E. of regression	1179.448	Akaike info criterion	16.99270
Sum squared resid	4.47E+08	Schwarz criterion	17.02770
Log likelihood	-2749.817	Hannan-Quinn criter.	17.00667
F-statistic	79.73415	Durbin-Watson stat	2.006132
Prob(F-statistic)	0.000000		

Figure 60: Conducting a Unit Root Test on the First Differences of House Prices

18 Cointegration Tests and Modelling Cointegrated Systems

Reading: Brooks (2019, sections 8.3–8.11)

The S&P500 spot and futures series that were discussed in section 4 of this guide will now be examined for cointegration using EViews. If the two series are cointegrated, this means that the spot and futures prices have a long-term relationship, which prevents them from wandering apart without bound. To test for cointegration using the Engle–Granger approach, the residuals of a regression of the spot price on the futures price are examined.[11] **Create two new variables**, for the log of the spot series and the log of the futures series, and call them '**lspot**' and '**lfutures**', respectively. Then generate a new equation object and run the regression:

lspot c lfutures

Note again that it is not valid to examine anything other than the coefficient values in this regression. The residuals of this regression are found in the object called RESID. From viewing the regression results, **click View/Actual,Fitted,Residual/Actual,Fitted,Residual Graph**. You will see a plot of the levels of the residuals (blue line), which looks much more like a stationary series than the original spot series (the red line corresponding to the actual values of y,[12]) does. Note how close together the actual and fitted lines are – the two are virtually indistinguishable and hence the very small left-hand scale for the residuals. The plot should appear as in Figure 61.

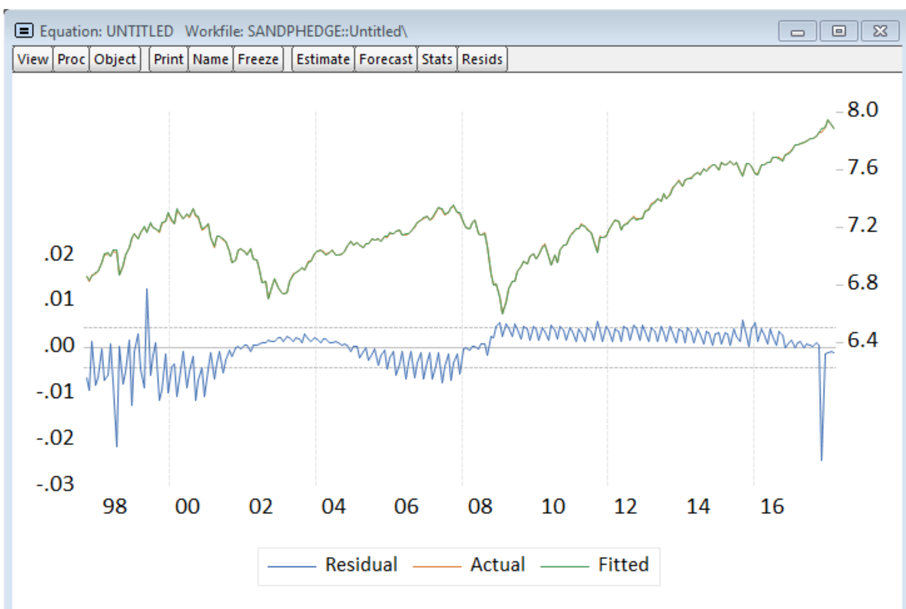


Figure 61: Actual, Fitted and Residual Plot

Generate a new series that will keep these residuals in an object for later use:

statsresid = resid

This is required since every time a regression is run, the RESID object is updated (overwritten) to contain the residuals of the most recently conducted regression. **Perform the ADF Test** on the

[11] Note that it is common to run a regression of the log of the spot price on the log of the futures rather than a regression in levels; the main reason for using logarithms is that the differences of the logs are returns, whereas this is not true for the levels.

[12] Although this is very hard to identify in the plot as it is hidden behind the fitted line!

residual series STATRESIDS. Assuming again that up to twelve lags are permitted, that the Schwarz criterion is used to select the optimal lag length, and that a constant but not a trend are employed in a regression on the levels of the series, the results are presented in Figure 62.

Series: STATRESIDS Workfile: SANDPHEDGE::Untitled\

View | Proc | Object | Properties | Print | Name | Freeze | Sample | Genr | Sheet | Graph | Stats | Ident

Augmented Dickey-Fuller Unit Root Test on STATRESIDS

Null Hypothesis: STATRESIDS has a unit root
Exogenous: Constant
Lag Length: 9 (Automatic - based on SIC, maxlag=12)

		t-Statistic	Prob.*
Augmented Dickey-Fuller test statistic		-2.031363	0.2733
Test critical values:	1% level	-3.457865	
	5% level	-2.873543	
	10% level	-2.573242	

*MacKinnon (1996) one-sided p-values.

Augmented Dickey-Fuller Test Equation
Dependent Variable: D(STATRESIDS)
Method: Least Squares
Date: 11/22/18 Time: 12:09
Sample (adjusted): 1998M07 2018M03
Included observations: 237 after adjustments

Variable	Coefficient	Std. Error	t-Statistic	Prob.
STATRESIDS(-1)	-0.107783	0.053060	-2.031363	0.0434
D(STATRESIDS(-1))	-0.690345	0.079167	-8.720085	0.0000
D(STATRESIDS(-2))	-0.539857	0.092388	-5.843396	0.0000
D(STATRESIDS(-3))	-0.312568	0.100686	-3.104380	0.0022
D(STATRESIDS(-4))	-0.220109	0.103969	-2.117059	0.0353
D(STATRESIDS(-5))	-0.165542	0.110329	-1.500431	0.1349
D(STATRESIDS(-6))	0.072182	0.116074	0.621862	0.5347
D(STATRESIDS(-7))	-0.000380	0.118992	-0.003194	0.9975
D(STATRESIDS(-8))	-0.116980	0.106569	-1.097695	0.2735
D(STATRESIDS(-9))	0.257583	0.081440	3.162850	0.0018
C	3.52E-05	0.000169	0.208125	0.8353

R-squared	0.700686	Mean dependent var	2.56E-05
Adjusted R-squared	0.687442	S.D. dependent var	0.004642
S.E. of regression	0.002595	Akaike info criterion	-9.024871
Sum squared resid	0.001522	Schwarz criterion	-8.863906
Log likelihood	1080.447	Hannan-Quinn criter.	-8.959992
F-statistic	52.90592	Durbin-Watson stat	1.869560
Prob(F-statistic)	0.000000		

Figure 62: Augmented Dickey–Fuller Test Results on the Residuals

Since the test statistic (-2.03) is above the critical values, even at the 10% level, the null hypothesis of a unit root in the test regression residuals cannot be rejected. We would thus conclude that the two series are not cointegrated. This means that the most appropriate form of model to estimate would be one containing only first differences of the variables as they have no long-run relationship.

If instead we had found the two series to be cointegrated, an error correction model (ECM) could have been estimated, as there would be a linear combination of the spot and futures prices that would be stationary. The ECM would be the appropriate model in that case rather than a model in pure first difference form because it would enable us to capture the long-run relationship between the series as well as the short-run one. We could estimate an error correction model by running the regression

rspot c rfutures statresids(–1)

However, if you estimate the model, the estimate on the error correction term is not really plausible and given that the two series are not cointegrated, a model of the form

rspot c rfutures rspot(–1) rfutures(–1)

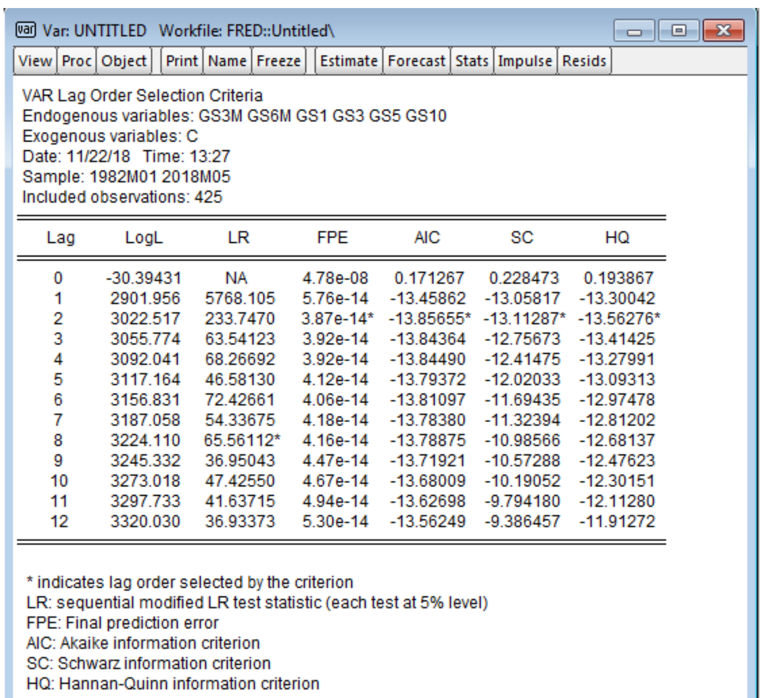
would be more appropriate. Note that we can either include or exclude the lagged terms and either form would be valid from the perspective that all of the elements in the equation are stationary.

Before moving on, we should note that this result is not an entirely stable one – for instance, if we run the regression containing no lags (i.e., the pure Dickey–Fuller test) or on a subsample of the data, we would find that the unit root null hypothesis should be rejected, indicating that the series are cointegrated. We thus need to be careful about drawing a firm conclusion in this case.

18.1 The Johansen Cointegration Test

Although the Engle–Granger approach is evidently very easy to use, as outlined above, one of its major drawbacks is that it can estimate only up to one cointegrating relationship between the variables. In the spot–futures example, there can be at most one cointegrating relationship since there are only two variables in the system. But in other situations, if there are more variables, there could potentially be more than one linearly independent cointegrating relationship. Thus, it is appropriate instead to examine the issue of cointegration within the Johansen VAR framework.

The application we will now examine centres on whether the yields on Treasury bills of different maturities are cointegrated. **Re-open the 'fred.wf1' workfile** that was used previously. There are six interest rate series corresponding to three and six months, and one, three, five and ten years. Each series has a name in the file starting with the letters 'gs'.


Var: UNTITLED Workfile: FRED::Untitled\

View | Proc | Object | Print | Name | Freeze | Estimate | Forecast | Stats | Impulse | Resids

VAR Lag Order Selection Criteria
Endogenous variables: GS3M GS6M GS1 GS3 GS5 GS10
Exogenous variables: C
Date: 11/22/18 Time: 13:27
Sample: 1982M01 2018M05
Included observations: 425

Lag	LogL	LR	FPE	AIC	SC	HQ
0	-30.39431	NA	4.78e-08	0.171267	0.228473	0.193867
1	2901.956	5768.105	5.76e-14	-13.45862	-13.05817	-13.30042
2	3022.517	233.7470	3.87e-14*	-13.85655*	-13.11287*	-13.56276*
3	3055.774	63.54123	3.92e-14	-13.84364	-12.75673	-13.41425
4	3092.041	68.26692	3.92e-14	-13.84490	-12.41475	-13.27991
5	3117.164	46.58130	4.12e-14	-13.79372	-12.02033	-13.09313
6	3156.831	72.42661	4.06e-14	-13.81097	-11.69435	-12.97478
7	3187.058	54.33675	4.18e-14	-13.78380	-11.32394	-12.81202
8	3224.110	65.56112*	4.16e-14	-13.78875	-10.98566	-12.68137
9	3245.332	36.95043	4.47e-14	-13.71921	-10.57288	-12.47623
10	3273.018	47.42550	4.67e-14	-13.68009	-10.19052	-12.30151
11	3297.733	41.63715	4.94e-14	-13.62698	-9.794180	-12.11280
12	3320.030	36.93373	5.30e-14	-13.56249	-9.386457	-11.91272

* indicates lag order selected by the criterion
LR: sequential modified LR test statistic (each test at 5% level)
FPE: Final prediction error
AIC: Akaike information criterion
SC: Schwarz information criterion
HQ: Hannan-Quinn information criterion

Figure 63: Information Criteria for VAR Lag Order Decision

The first step in any cointegration analysis is to ensure that the variables are all non-stationary in their levels form, so **confirm that this is the case** for each of the six series, by running a unit root test on each one.[13] Before we run the cointergration test, we also need to specify how many lags we want to include and therefore look at the information criteria.

[13]Running ADF tests for all the six series you should find that the hypothesis of a unit root is rejected at the 10% level for all series except GS10 (p-value=0.1023), which is close to the critical value.

From Figure 63 we conclude that we should include two lags. Next, to run the cointegration test, click **Quick/Group Statistics/Johansen Cointegration Test**. A box should then appear with the names of the six series in it or with the group **'interest'** (Figure 64, left panel) if you have already generated it. Click **OK**, and then the list of options will be seen (Figure 64, right panel).

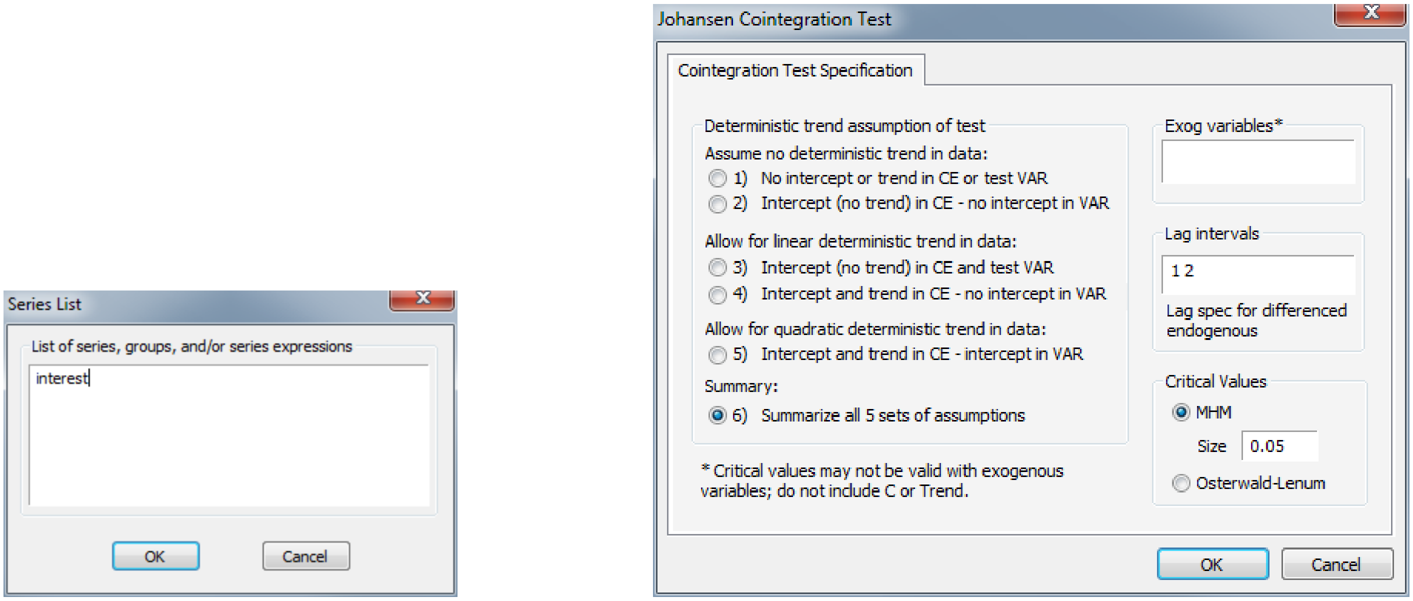


Figure 64: Running a Johansen Test for Cointegration

The differences between models 1 to 6 centre on whether an intercept or a trend or both are included in the potentially cointegrating relationship and/or the VAR. It is usually a good idea to examine the sensitivity of the result to the type of specification used, so select **Option 6** which will do this and click **OK**. The results appear as in Figure 65.

Group: INTEREST Workfile: FRED::Untitled\

View | Proc | Object | Print | Name | Freeze | Default | Sort | Edit+/- | Smpl+/- | Compare+/-

Johansen Cointegration Test Summary

Date: 11/22/18 Time: 13:37
Sample: 1982M01 2018M05
Included observations: 434
Series: GS3M GS6M GS1 GS3 GS5 GS10
Lags interval: 1 to 2

Selected (0.05 level*) Number of Cointegrating Relations by Model

Data Trend:	None	None	Linear	Linear	Quadratic
Test Type	No Intercept No Trend	Intercept No Trend	Intercept No Trend	Intercept Trend	Intercept Trend
Trace	4	3	4	4	6
Max-Eig	4	3	3	3	3

*Critical values based on MacKinnon-Haug-Michelis (1999)

Information Criteria by Rank and Model

Data Trend:	None	None	Linear	Linear	Quadratic
Rank or No. of CEs	No Intercept No Trend	Intercept No Trend	Intercept No Trend	Intercept Trend	Intercept Trend
	Log Likelihood by Rank (rows) and Model (columns)				
0	2831.463	2831.463	2833.281	2833.281	2834.624
1	2862.297	2863.838	2865.318	2865.319	2866.368
2	2886.025	2888.250	2889.652	2892.507	2893.460
3	2904.345	2909.266	2910.590	2914.367	2915.285
4	2913.692	2918.738	2920.014	2924.326	2925.001
5	2917.730	2923.142	2923.681	2933.626	2933.716
6	2919.792	2926.807	2926.807	2937.260	2937.260
	Akaike Information Criteria by Rank (rows) and Model (columns)				
0	-12.71642	-12.71642	-12.69715	-12.69715	-12.67569
1	-12.80321	-12.80570	-12.78948	-12.78488	-12.76667
2	-12.85726	-12.85830	-12.84632	-12.85026	-12.83622
3	-12.88638	-12.89524*	-12.88751	-12.89109	-12.88150
4	-12.87416	-12.87898	-12.87564	-12.87708	-12.87097
5	-12.83746	-12.83936	-12.83724	-12.86003	-12.85583
6	-12.79167	-12.79635	-12.79635	-12.81687	-12.81687
	Schwarz Criteria by Rank (rows) and Model (columns)				
0	-12.04071*	-12.04071*	-11.96513	-11.96513	-11.88735
1	-12.01488	-12.00799	-11.94484	-11.93086	-11.86572
2	-11.95631	-11.93858	-11.88907	-11.87424	-11.82265
3	-11.87282	-11.85351	-11.81763	-11.79306	-11.75531
4	-11.74797	-11.71525	-11.69314	-11.65704	-11.63216
5	-11.59866	-11.55363	-11.54212	-11.51799	-11.50441
6	-11.44024	-11.38861	-11.38861	-11.35282	-11.35282

Figure 65: Results of a Johansen Test for Cointegration

The findings across the six types of model and the type of test (the 'trace' or 'max' statistics) are a little mixed concerning the number of cointegrating vectors (the top panel), with both statistics always suggesting at least three cointegrating vectors but the trace approach selecting between three and six cointegrating vectors dependent on the specification of the VAR model.

The following three panels all provide information that could be used to determine the appropriate number of cointegrating relationships as well. AIC selects a cointegration rank of three for a model with only an intercept in the cointegrating relationship, while SBIC selects a model without cointegration or cointegration rank 0 within the same model specification. In order to see the estimated models, click **View/Cointegration Test/Johansen System Cointegration Test...** and select **Option 2** (Intercept (no trend) in CE - no intercept in VAR), keeping the 'Lag Intervals' at **1 2**, and clicking **OK**. EViews produces a very large quantity of output, as shown in Figure 66.

Var: UNTITLED Workfile: FRED::Untitled\

View | Proc | Object | Print | Name | Freeze | Estimate | Forecast | Stats | Impulse | Resids

Johansen Cointegration Test

Date: 11/22/18 Time: 13:56
Sample (adjusted): 1982M04 2018M05
Included observations: 434 after adjustments
Trend assumption: No deterministic trend (restricted constant)
Series: GS3M GS6M GS1 GS3 GS5 GS10
Lags interval (in first differences): 1 to 2

Unrestricted Cointegration Rank Test (Trace)

Hypothesized No. of CE(s)	Eigenvalue	Trace Statistic	0.05 Critical Value	Prob.**
None *	0.138598	190.6889	103.8473	0.0000
At most 1 *	0.106403	125.9387	76.97277	0.0000
At most 2 *	0.092306	77.11364	54.07904	0.0001
At most 3	0.042710	35.08159	35.19275	0.0514
At most 4	0.020091	16.13819	20.26184	0.1680
At most 5	0.016748	7.329973	9.164546	0.1101

Trace test indicates 3 cointegrating eqn(s) at the 0.05 level
* denotes rejection of the hypothesis at the 0.05 level
**MacKinnon-Haug-Michelis (1999) p-values

Unrestricted Cointegration Rank Test (Maximum Eigenvalue)

Hypothesized No. of CE(s)	Eigenvalue	Max-Eigen Statistic	0.05 Critical Value	Prob.**
None *	0.138598	64.75018	40.95680	0.0000
At most 1 *	0.106403	48.82505	34.80587	0.0006
At most 2 *	0.092306	42.03205	28.58808	0.0006
At most 3	0.042710	18.94340	22.29962	0.1379
At most 4	0.020091	8.808215	15.89210	0.4547
At most 5	0.016748	7.329973	9.164546	0.1101

Max-eigenvalue test indicates 3 cointegrating eqn(s) at the 0.05 level
* denotes rejection of the hypothesis at the 0.05 level
**MacKinnon-Haug-Michelis (1999) p-values

Unrestricted Cointegrating Coefficients (normalized by b'*S11*b=I):

GS3M	GS6M	GS1	GS3	GS5	GS10	C
10.16782	-24.23283	18.92667	-10.11937	7.708560	-2.381371	0.998322
4.953569	-1.547047	-4.080362	2.726714	-3.772767	2.047175	-0.814054
2.877557	-1.764397	-7.760092	17.60629	-13.83242	2.560537	1.219307
0.468814	-3.551838	3.587013	7.101529	-13.05669	5.509828	-0.433873
-2.164809	5.287028	-3.211911	-1.398573	3.332477	-2.159889	1.744552
1.222864	-2.131817	1.109851	-0.249089	2.719789	-2.682795	1.606642

Unrestricted Adjustment Coefficients (alpha):

D(GS3M)	0.020525	-0.034340	0.033499	0.012503	0.012806	-0.016044
D(GS6M)	0.035794	-0.011907	0.036845	0.007500	0.009609	-0.018213
D(GS1)	0.029964	0.005383	0.032584	0.004404	0.012971	-0.021349
D(GS3)	0.041468	0.023942	0.012894	0.012754	0.019104	-0.019667
D(GS5)	0.038113	0.033705	0.015563	0.018294	0.019997	-0.013989
D(GS10)	0.038410	0.033721	0.017848	0.012873	0.022563	-0.006294

1 Cointegrating Equation(s): Log likelihood 2863.838

Normalized cointegrating coefficients (standard error in parentheses)

GS3M	GS6M	GS1	GS3	GS5	GS10	C
1.000000	-2.383287	1.861428	-0.995235	0.758133	-0.234207	0.098184
	(0.12435)	(0.20018)	(0.25767)	(0.25473)	(0.09234)	(0.04630)

Adjustment coefficients (standard error in parentheses)

D(GS3M)	0.208690
	(0.11414)
D(GS6M)	0.363946
	(0.11050)
D(GS1)	0.304669
	(0.11451)
D(GS3)	0.421637
	(0.12433)
D(GS5)	0.387524
	(0.12314)
D(GS10)	0.390542
	(0.11610)

2 Cointegrating Equation(s): Log likelihood 2888.250

Normalized cointegrating coefficients (standard error in parentheses)

GS3M	GS6M	GS1	GS3	GS5	GS10	C
1.000000	0.000000	-1.228651	0.783549	-0.990810	0.510915	-0.203926
		(0.19658)	(0.60833)	(0.65364)	(0.24598)	(0.12665)
0.000000	1.000000	-1.296562	0.746358	-0.733837	0.312644	-0.126762
		(0.08930)	(0.27634)	(0.29692)	(0.11174)	(0.05753)

Adjustment coefficients (standard error in parentheses)

D(GS3M)	0.038586	-0.444242
	(0.12554)	(0.26953)
D(GS6M)	0.304963	-0.848967
	(0.12274)	(0.26351)
D(GS1)	0.331335	-0.734442
	(0.12734)	(0.27340)
D(GS3)	0.540235	-1.041921
	(0.13767)	(0.29556)
D(GS5)	0.554482	-0.975724
	(0.13572)	(0.29137)
D(GS10)	0.557583	-0.982942
	(0.12780)	(0.27438)

3 Cointegrating Equation(s): Log likelihood 2909.266

Normalized cointegrating coefficients (standard error in parentheses)

GS3M	GS6M	GS1	GS3	GS5	GS10	C
1.000000	0.000000	0.000000	-2.361258	1.325297	0.201125	-0.502485
			(0.42818)	(0.76777)	(0.37561)	(0.20258)
0.000000	1.000000	0.000000	-2.572274	1.710289	-0.014268	-0.441824
			(0.32661)	(0.58564)	(0.28650)	(0.15452)
0.000000	0.000000	1.000000	-2.559562	1.885082	-0.252138	-0.242998
			(0.20358)	(0.36503)	(0.17858)	(0.09632)

Adjustment coefficients (standard error in parentheses)

D(GS3M)	0.134982	-0.503348	0.268622
	(0.12813)	(0.26730)	(0.22901)
D(GS6M)	0.410988	-0.913977	0.440122
	(0.12490)	(0.26056)	(0.22324)
D(GS1)	0.425098	-0.791933	0.292299
	(0.13009)	(0.27138)	(0.23250)
D(GS3)	0.577338	-1.064670	0.587097
	(0.14186)	(0.29594)	(0.25355)
D(GS5)	0.599265	-1.003183	0.463052
	(0.13976)	(0.29155)	(0.24979)
D(GS10)	0.608942	-1.014433	0.450866
	(0.13148)	(0.27429)	(0.23500)

4 Cointegrating Equation(s): Log likelihood 2918.738

Normalized cointegrating coefficients (standard error in parentheses)

GS3M	GS6M	GS1	GS3	GS5	GS10	C
1.000000	0.000000	0.000000	0.000000	-2.784501	1.994736	-0.758812
				(0.27908)	(0.30164)	(0.33373)
0.000000	1.000000	0.000000	0.000000	-2.766784	1.939630	-0.721058
				(0.26816)	(0.28984)	(0.32067)
0.000000	0.000000	1.000000	0.000000	-2.569866	1.692105	-0.520852
				(0.23942)	(0.25878)	(0.28630)
0.000000	0.000000	0.000000	1.000000	-1.740512	0.759600	-0.108555
				(0.08537)	(0.09227)	(0.10208)

Adjustment coefficients (standard error in parentheses)

D(GS3M)	0.140844	-0.547757	0.313470	0.377257
	(0.12804)	(0.26971)	(0.23201)	(0.23772)
D(GS6M)	0.414504	-0.940615	0.467024	0.307290
	(0.12493)	(0.26317)	(0.22638)	(0.23195)
D(GS1)	0.427162	-0.807574	0.308096	0.316418
	(0.13017)	(0.27420)	(0.23587)	(0.24167)
D(GS3)	0.583318	-1.109971	0.632846	-0.036757
	(0.14179)	(0.29869)	(0.25694)	(0.26325)
D(GS5)	0.607842	-1.068161	0.528674	0.110147
	(0.13948)	(0.29382)	(0.25275)	(0.25897)
D(GS10)	0.614977	-1.060157	0.497043	0.108930
	(0.13139)	(0.27677)	(0.23808)	(0.24393)

5 Cointegrating Equation(s): Log likelihood 2923.142

Normalized cointegrating coefficients (standard error in parentheses)

GS3M	GS6M	GS1	GS3	GS5	GS10	C
1.000000	0.000000	0.000000	0.000000	0.000000	-1.584446	3.917127
					(0.26846)	(1.73006)
0.000000	1.000000	0.000000	0.000000	0.000000	-1.616778	3.925128
					(0.26680)	(1.71936)
0.000000	0.000000	1.000000	0.000000	0.000000	-1.611187	3.794656
					(0.24814)	(1.59914)
0.000000	0.000000	0.000000	1.000000	0.000000	-1.477645	2.814240
					(0.16713)	(1.07704)
0.000000	0.000000	0.000000	0.000000	1.000000	-1.285394	1.679273
					(0.09541)	(0.61487)

Adjustment coefficients (standard error in parentheses)

D(GS3M)	0.113122	-0.480052	0.272339	0.359347	-0.296177
	(0.13001)	(0.27542)	(0.23429)	(0.23783)	(0.23128)
D(GS6M)	0.393703	-0.889814	0.436162	0.293852	-0.254719
	(0.12694)	(0.26892)	(0.22876)	(0.23221)	(0.22582)
D(GS1)	0.399082	-0.738995	0.266433	0.298277	-0.254318
	(0.13217)	(0.28001)	(0.23819)	(0.24178)	(0.23513)
D(GS3)	0.541961	-1.008967	0.571485	-0.063476	-0.051887
	(0.14378)	(0.30460)	(0.25911)	(0.26302)	(0.25579)
D(GS5)	0.564552	-0.962436	0.464444	0.082180	-0.220859
	(0.14139)	(0.29953)	(0.25479)	(0.25864)	(0.25152)
D(GS10)	0.566132	-0.940864	0.424572	0.077373	-0.170916
	(0.13298)	(0.28173)	(0.23965)	(0.24327)	(0.23658)

Figure 66: Results of a Johansen Test for Cointegration for Option 2

The first two panels Figure 66 show the results for the λ_{trace} and $\lambda_{\max}$ statistics, respectively. The second column in each case presents the ordered eigenvalues, the third column the test statistics, the fourth column the critical values and the final column the p-values. Examining the trace test, if we look at the first row after the headers, the statistic of 190.69 considerably exceeds the critical value (of 103.85) and so the null of no cointegrating vectors is rejected. If we then move to the next row, the test statistic (125.94) again exceeds the critical value so that the null of at most one cointegrating vector is also rejected. This continues, and we also reject the null of at most two cointegrating vectors, but we stop at the next row, where we do not reject the null hypothesis of at most three cointegrating vectors at the 5% level, and this is the conclusion. The *max* test, shown in the second panel, confirms this result.

The unrestricted coefficient values are the estimated values of coefficients in the cointegrating vector, and these are presented in the third panel. However, it is sometimes useful to normalise the coefficient values to set the coefficient value on one of them to unity, as would be the case in the cointegrating regression under the Engle–Granger approach. The normalisation will be done by EViews with respect to the first variable given in the variable list (i.e., whichever variable you listed first in the system will by default be given a coefficient of 1 in the normalised cointegrating vector). The first panel after the unrestricted cointegration coefficients and adjustments presents the estimates if there were only one cointegrating vector, which has been normalised so that the coefficient on the three-month bond yield is unity. The adjustment coefficients, or loadings in each regression (the 'amount of the cointegrating vector' in each equation), are also given in this panel. In the next panel, the same format is used (i.e., the normalised cointegrating vectors are presented and then the adjustment parameters) but under the assumption that there are two cointegrating vectors, and this proceeds until the situation where there are five cointegrating vectors, the maximum number possible for a system containing six variables.

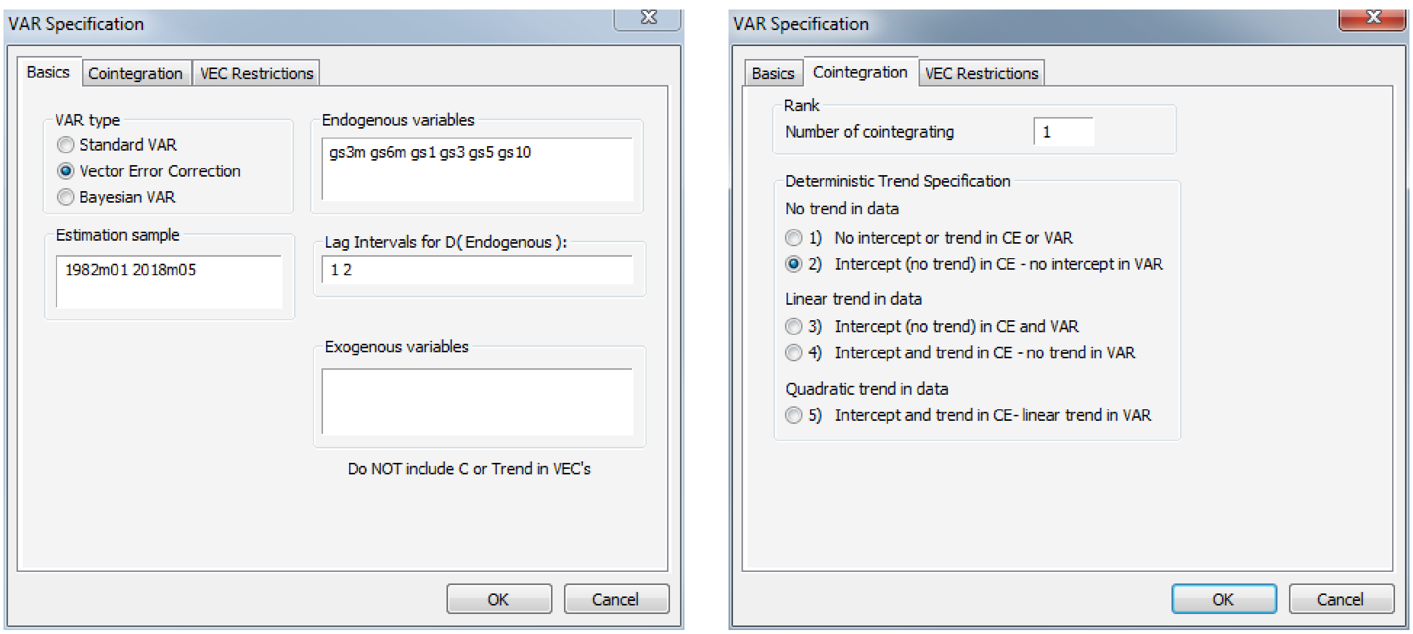


Figure 67: Specification of a VEC Model

In order to see the whole VECM model, select **Proc/Make Vector Autoregression . . . .** Starting on the default 'Basics' tab, in 'VAR type', select **Vector Error Correction**, and in the 'Lag Intervals for D(Endogenous):' box, type **1 2** as in Figure 67, left panel. Then click on the **Cointegration tab** (Figure 67, right panel) and leave the default as 1 cointegrating vector for simplicity in the 'Rank' box and option 2 to have an intercept but no trend in the cointegrating equation and the VAR. When **OK** is clicked, the output for the entire VECM will be seen.

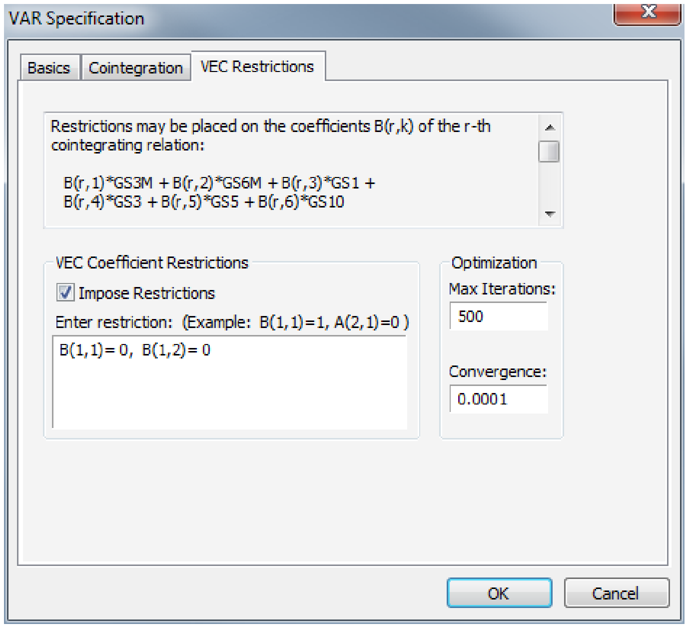


Figure 68: Specification of Restrictions to a VEC Model

It is sometimes of interest to test hypotheses about either the parameters in the cointegrating vector or their loadings in the VECM. To do this from the 'Vector Error Correction Estimates' screen, click the **Estimate** button and click on the **VEC Restrictions** tab. In EViews, restrictions concerning the cointegrating relationships embodied in β are denoted by $B(i,j)$, where $B(i,j)$ represents the jth coefficient in the ith cointegrating relationship (Figure 68).

In this case, we are allowing for only one cointegrating relationship, so suppose that we want to test the hypothesis that the three-month and six-month yields do not appear in the cointegrating equation. We could test this by specifying the restriction that their parameters are zero, which in EViews terminology would be achieved by writing **B(1,1) = 0, B(1,2) = 0** in the 'VEC Coefficient Restrictions' box and clicking **OK**. EViews will then show the value of the test statistic, followed by the restricted cointegrating vector and the VECM. To preserve space, only the test statistic and restricted cointegrating vector are shown in the output (Figure 69).

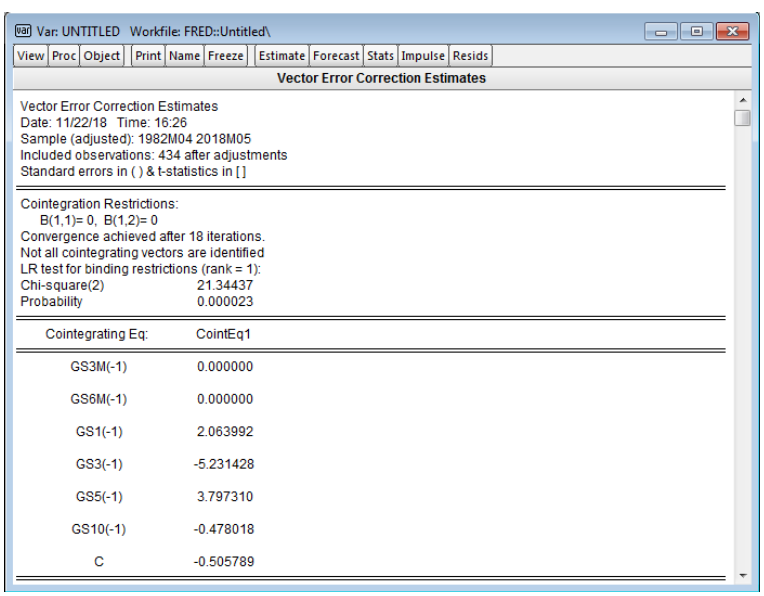


Figure 69: Restricted Cointegrating Vector

There are two restrictions, so that the test statistic follows a χ^2 distribution with two degrees of freedom. Here, the p-value for the test is 0.000023, and so the restrictions are not supported by the data at the 1% level and we would conclude that the cointegrating relationship does need to include the short end of the yield curve. When performing hypothesis tests concerning the adjustment coefficients (i.e., the loadings in each equation), the restrictions are denoted by $A(i, j)$, which is the coefficient on the cointegrating vector for the ith variable in the jth cointegrating relation. For example, $A(2, 1) = 0$ would test the null that the equation for the second variable in the order that they were listed in the original specification (GS6M in this case) does not include the first cointegrating vector, and so on. Examining some restrictions of this type are beyond the scope of this guide.

19 Volatility Modelling

19.1 Testing for 'ARCH Effects' in Exchange Rate Returns

Reading: Brooks (2019, Section 9.7)

Before estimating a GARCH-type model, it is sensible first to compute the Engle (1982) test for ARCH effects to make sure that this class of models is appropriate for the data. This exercise (and the remaining exercises of this section), will employ returns on the daily exchange rates (the file is **'currencies.wf1'**) where there are 7,141 observations. Models of this kind are inevitably more data intensive than those based on simple linear regressions, and hence, everything else being equal, they work better when the data are sampled daily rather than at a lower frequency.

A test for the presence of ARCH in the residuals is calculated by regressing the squared residuals on a constant and p lags, where p is set by the user. As an example, assume that p is set to 5. The first step is to estimate a linear model so that the residuals can be tested for ARCH. From the main menu, select **Quick** and then select **Estimate Equation**. In the Equation Specification Editor, input

rgbp c ar(1) ma(1)

which will estimate an ARMA(1,1) for the pound–dollar returns.[14] Select the **Least Squares (NLS and ARMA)** procedure to estimate the model, using the whole sample period and press the **OK** button (Figure 70).

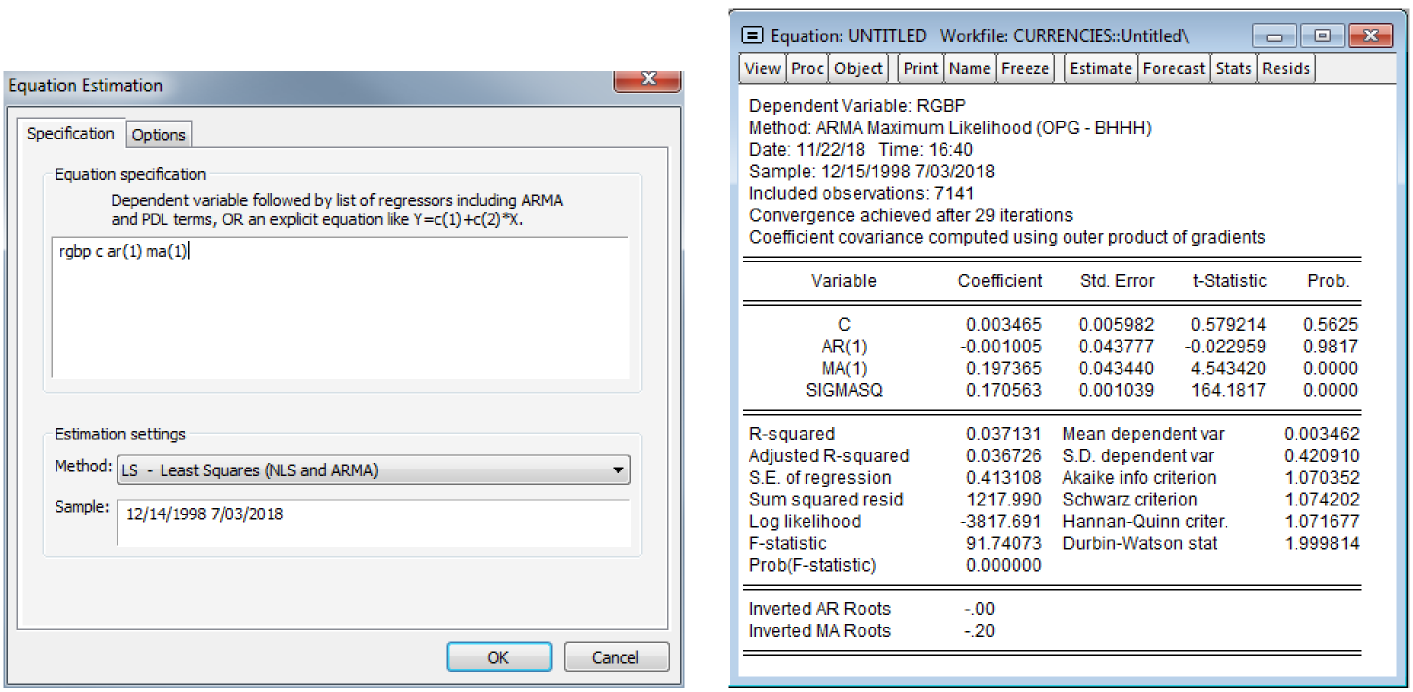


Figure 70: ARMA(1,1) Model for the British Pound Exchange Rate

The next step is to click on **View** from the Equation Window and to select **Residual Diagnostics** and then **Heteroskedasticity Tests. . . .** In the 'Test type' box, choose **ARCH** and the number of lags to include is **5**, and press **OK**. The output below shows the Engle test results. Both the F-version

[14] Note that the (1,1) order has been chosen entirely arbitrarily at this stage. However, it is important to give some thought to the type and order of model used even if it is not of direct interest in the problem at hand (which will later be termed the 'conditional mean' equation), since the variance is measured around the mean and therefore any mis-specification in the mean is likely to lead to a mis-specified variance.

and the LM-statistic are very significant, suggesting the presence of ARCH in the pound–dollar returns (Figure 71).

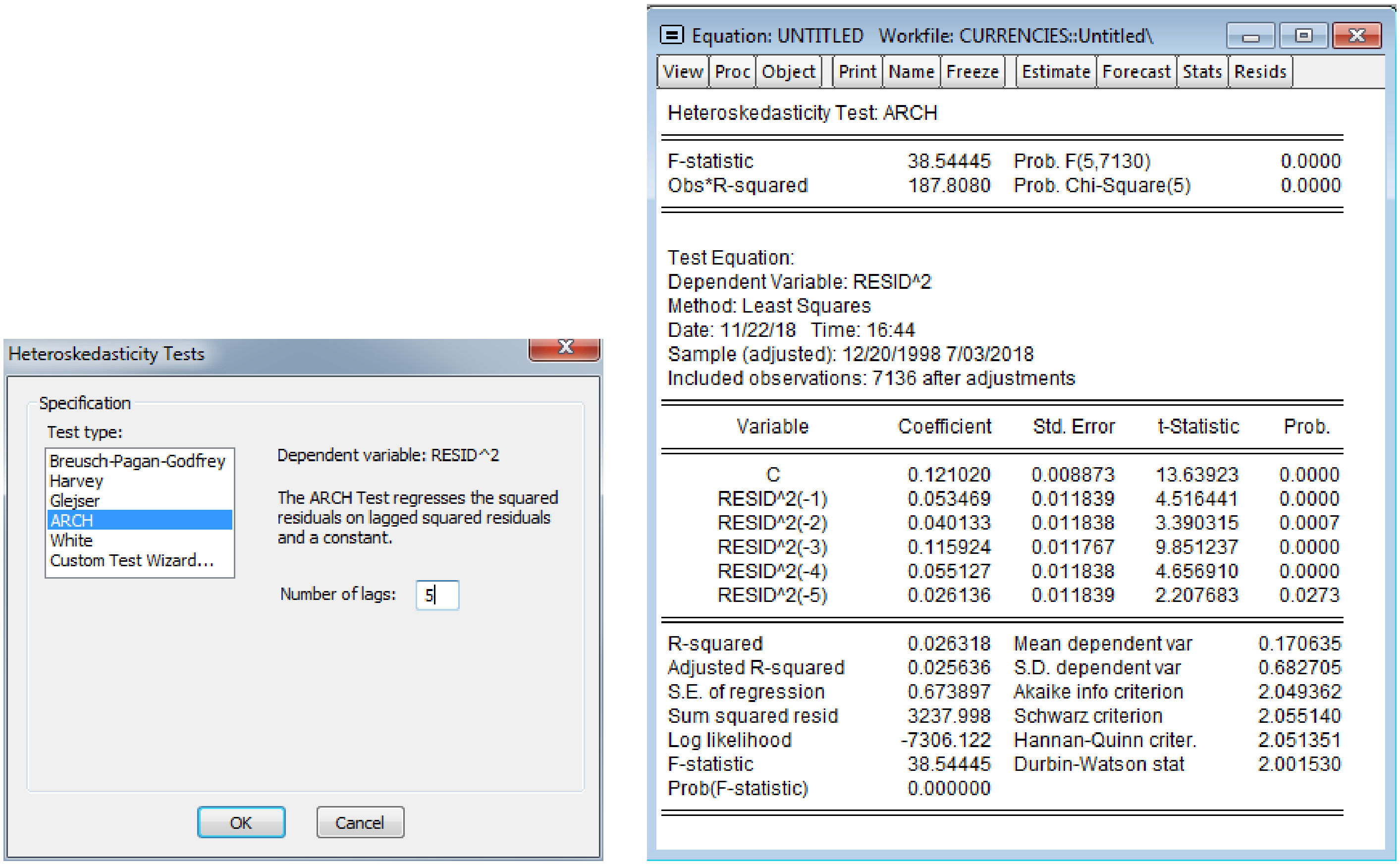


Figure 71: An ARCH Test

19.2 Estimating GARCH Models

Reading: Brooks (2019, Section 9.9)

To estimate a GARCH-type model, open the equation specification dialog box by selecting **Quick-/Estimate Equation** or by selecting **Object/New Object/Equation. . .** . Select **ARCH** from the 'Estimation Settings Method' selection box. The window in Figure 72, left panel will open. It is necessary to specify both the mean and the variance equations, as well as the estimation technique and sample.

The specification of the mean equation should be entered in the dependent variable edit box. Enter the specification by listing the dependent variable followed by the regressors. The constant term 'C' should also be included. If your specification includes an ARCH-M term (see later in this section), you should click on the appropriate button in the upper RHS of the dialog box to select the conditional standard deviation, the conditional variance, or the log of the conditional variance.

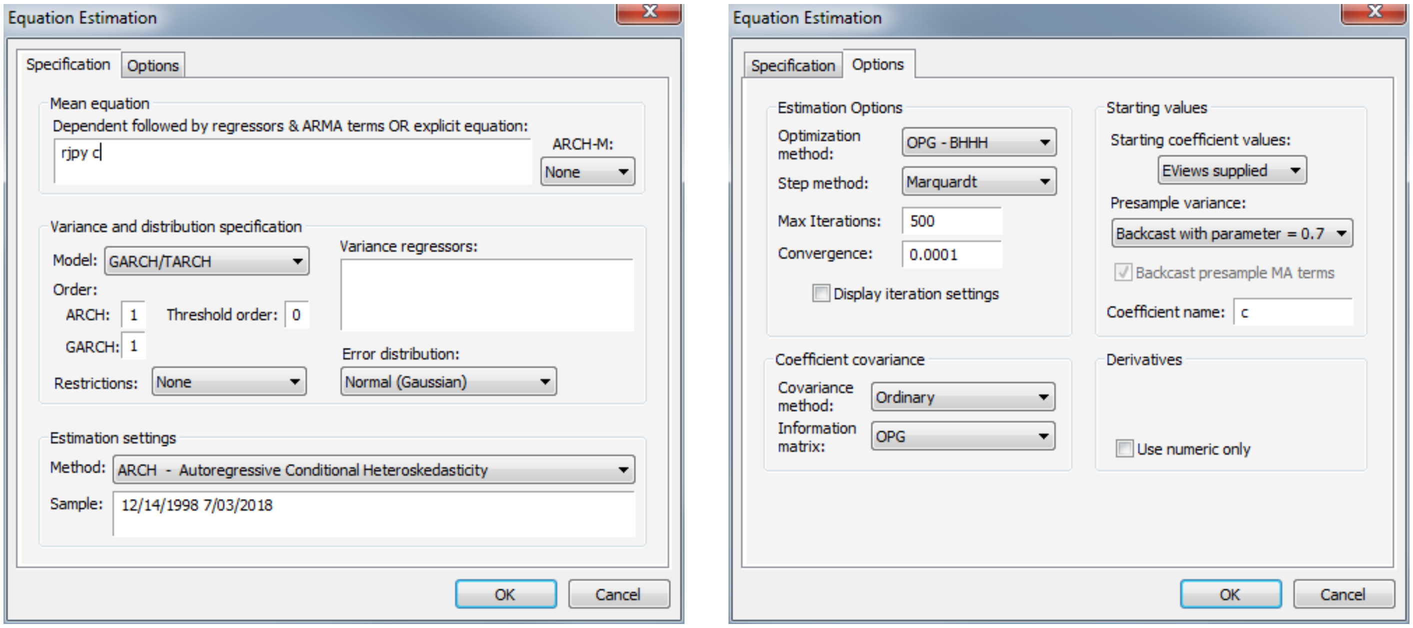


Figure 72: Specification Window for GARCH Model Estimation

The edit box labelled 'Variance regressors' is where variables that are to be included in the variance specification should be listed. Note that EViews will always include a constant in the conditional variance, so that it is not necessary to add 'C' to the variance regressor list. Similarly, it is not necessary to include the ARCH or GARCH terms in this box as they will be dealt with in other parts of the dialog box. Instead, enter here any exogenous variables or dummies that you wish to include in the conditional variance equation, or (as is usually the case), just leave this box blank.

Under the 'Variance and distribution Specification' label, choose the number of ARCH and GARCH terms. The default is to estimate with one ARCH and one GARCH term (i.e., one lag of the squared errors and one lag of the conditional variance, respectively). To estimate the standard GARCH model, leave the default 'GARCH/TARCH'. The other entries in this box describe more complicated variants of the standard GARCH specification, which are described in later subsections of this section.

EViews provides a number of optional estimation settings. Clicking on the Options tab gives the options in the right panel of Figure 72 to be filled out as required. The Heteroskedasticity Consistent Covariance option is used to compute the QML covariances and standard errors using the methods described by Bollerslev and Wooldridge (1992). This option should be used if you suspect that the residuals are not conditionally normally distributed. Note that the parameter estimates will be (virtually) unchanged if this option is selected; only the estimated covariance matrix will be altered.

The log-likelihood functions for ARCH models are often not well behaved so that convergence may not be achieved with the default estimation settings. It is possible in EViews to select the iterative algorithm (Marquardt, BHHH/Gauss Newton), to change starting values, to increase the maximum number of iterations or to adjust the convergence criteria. For example, if convergence is not achieved, or implausible parameter estimates are obtained, it is sensible to re-do the estimation using a different set of starting values and/or a different optimisation algorithm. Once the model has been estimated, EViews provides a variety of information and procedures for inference and diagnostic checking (Figure 73).

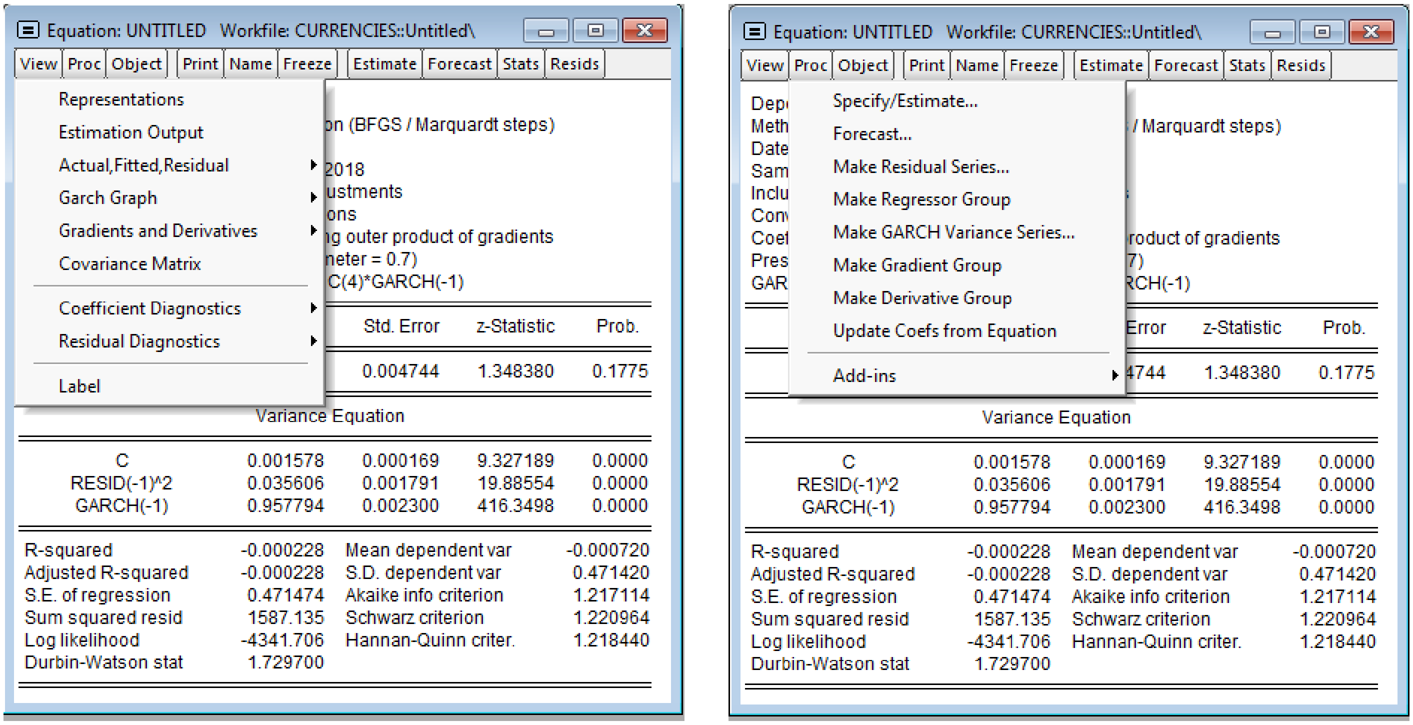
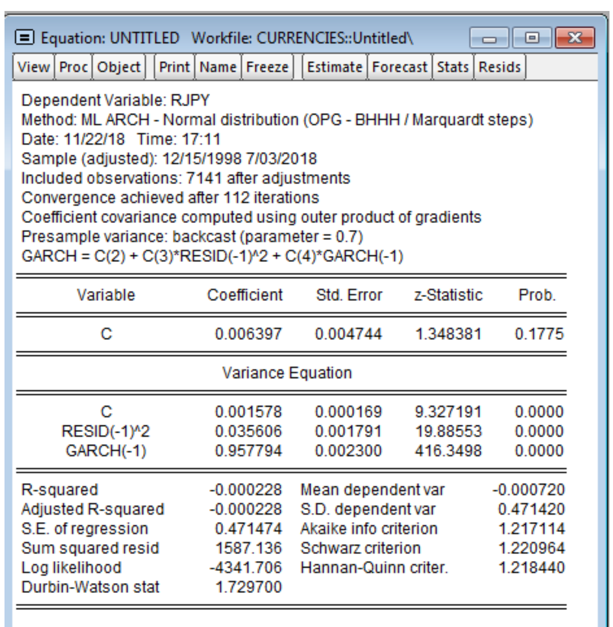


Figure 73: Additional Options for GARCH Models

For example, the **View** tab (Figure 73, left panel) offers you the options to display the residuals in various forms, such as table, graphs and standardised residuals. For example, you can plot the one-step ahead standard deviation, σ_t, or the conditional variance, σ_t^2, for each observation in the sample. You can examine the covariance matrix, run coefficient tests and residual diagnosis tests.

Further, by pressing the **Proc** tab (Figure 73, right panel) you can save the residuals from the mean and variance equations as well as produce forecasts. Estimating the GARCH(1,1) model for the yen–dollar ('rjpy') series using the instructions as listed above, and the default settings elsewhere would yield the results in Figure 74.


Equation: UNTITLED Workfile: CURRENCIES::Untitled\

View Proc Object Print Name Freeze Estimate Forecast Stats Resids

Dependent Variable: RJPY
Method: ML ARCH - Normal distribution (OPG - BHHH / Marquardt steps)
Date: 11/22/18 Time: 17:11
Sample (adjusted): 12/15/1998 7/03/2018
Included observations: 7141 after adjustments
Convergence achieved after 112 iterations
Coefficient covariance computed using outer product of gradients
Presample variance: backcast (parameter = 0.7)
GARCH = C(2) + C(3)*RESID(-1)^2 + C(4)*GARCH(-1)

Variable	Coefficient	Std. Error	z-Statistic	Prob.
C	0.006397	0.004744	1.348381	0.1775
		Variance Equation		
C	0.001578	0.000169	9.327191	0.0000
RESID(-1)^2	0.035606	0.001791	19.88553	0.0000
GARCH(-1)	0.957794	0.002300	416.3498	0.0000

R-squared	-0.000228	Mean dependent var	-0.000720
Adjusted R-squared	-0.000228	S.D. dependent var	0.471420
S.E. of regression	0.471474	Akaike info criterion	1.217114
Sum squared resid	1587.136	Schwarz criterion	1.220964
Log likelihood	-4341.706	Hannan-Quinn criter.	1.218440
Durbin-Watson stat	1.729700		

Figure 74: Results of GARCH Model Estimation

The coefficients on both the lagged squared residual and lagged conditional variance terms in the conditional variance equation are highly statistically significant. Also, as is typical of GARCH model estimates for financial asset returns data, the sum of the coefficients on the lagged squared error and lagged conditional variance is very close to unity (approximately 0.99). This implies that shocks to the conditional variance will be highly persistent. This can be seen by considering the equations for forecasting future values of the conditional variance using a GARCH model given in Brooks (2019, chapter 9). A large sum of these coefficients will imply that a large positive or a large negative return will lead future forecasts of the variance to be high for a protracted period. The individual conditional variance coefficients are also as one would expect. The variance intercept term 'C' is very small, and the 'ARCH parameter' is around 0.035 while the coefficient on the lagged conditional variance ('GARCH') is larger at 0.958.

19.3 GJR and EGARCH

Reading: Brooks (2019, Sections 9.10–9.13)

The main menu screen for GARCH estimation demonstrates that a number of variants on the standard GARCH model are available. Arguably the most important of these are asymmetric models, such as the TGARCH ('threshold' GARCH), which is also known as the GJR model, and the EGARCH model. To estimate a GJR model in EViews, from the GARCH model equation specification screen (Figure 75, left panel), change the 'Threshold Order' number from 0 to 1. The results should be as presented in Figure 75, right panel.

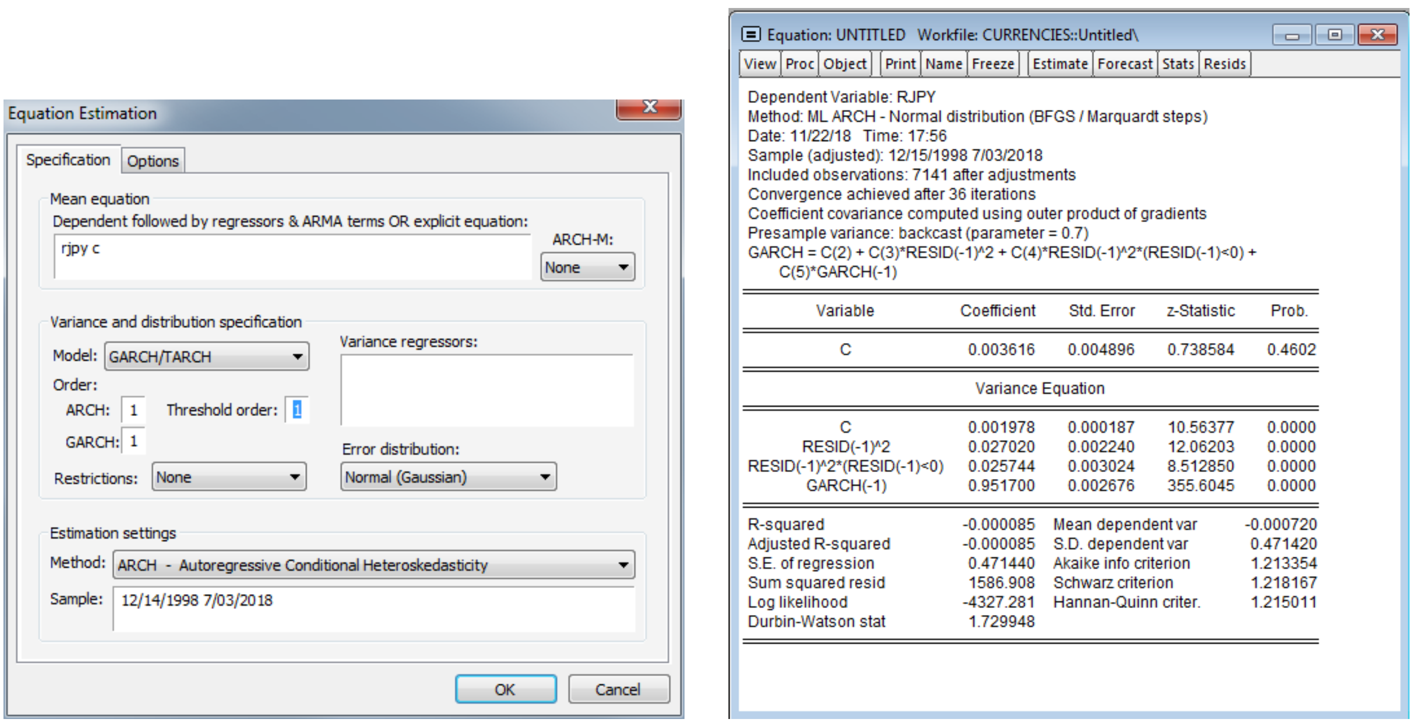


Figure 75: Threshold GARCH Model Estimation

To estimate an EGARCH model, we need to also **change the 'GARCH/TARCH' model estimation default to 'EGARCH'** as in Figure 76, left panel. Coefficient estimates for this specification using the daily Japanese Yen–US Dollar returns data is given in Figure 76, right panel. For the GJR specification, the asymmetry term ('(RESID(–1)ˆ2*RESID(–1)<0)') is positive and highly significant, while for the EGARCH model, the estimate on 'RESID(–1)/SQRT(GARCH(–1))' is highly significant but has a negative sign. So for the GJR model, the estimate indicates that negative shocks imply a higher next period conditional variance than positive shocks of the same sign, whereas the opposite is true for the EGARCH. Clearly, then, we must exercise caution when interpreting the estimates from GARCH-type models since the optimisation routine converged to an optimum in both cases and the estimates appear to be otherwise entirely plausible.

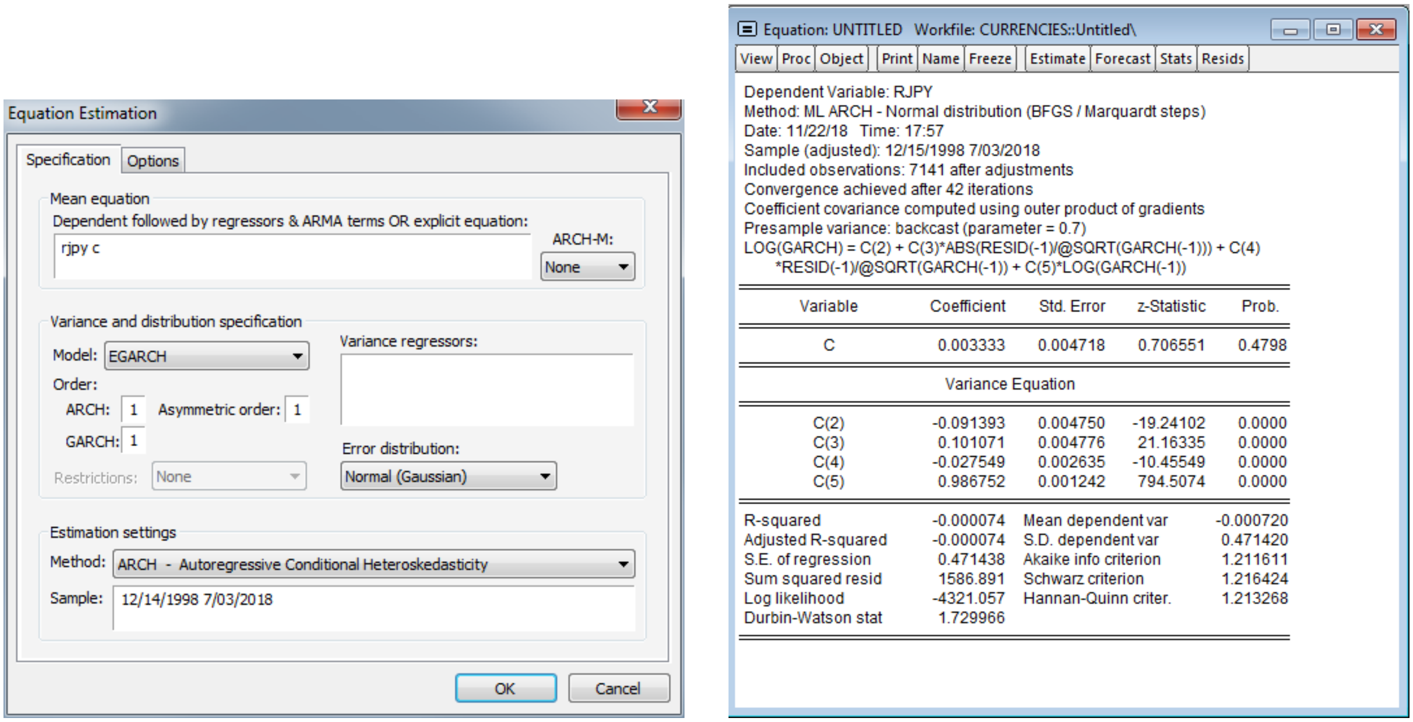


Figure 76: EGARCH Model Estimation

The sign of the EGARCH asymmetry term (C(4) in the EViews output) is also negative and highly significant as for the GJR model, and is as would have been expected in the case of the application of a GARCH model to a set of stock returns. But, arguably, neither the *leverage effect* or *volatility feedback* explanations for asymmetries in the context of stocks apply here. For a positive return shock, this implies more yen per dollar and therefore a strengthening dollar and a weakening yen. Thus the EGARCH results suggest that a strengthening dollar (weakening yen) leads to higher next period volatility than when the yen strengthens by the same amount (vice versa for the GJR).

19.4 GARCH-M Estimation

Reading: Brooks (2019, Section 9.15)

The GARCH-M model with the conditional variance term in the mean with no asymmetries, can be specified as shown in Figure 77, left panel. In this case, the estimated parameter on the mean equation has a negative sign but is not statistically significant (Figure 77, right panel). We would thus conclude that for these currency returns, there is no feedback from the conditional variance to the conditional mean.

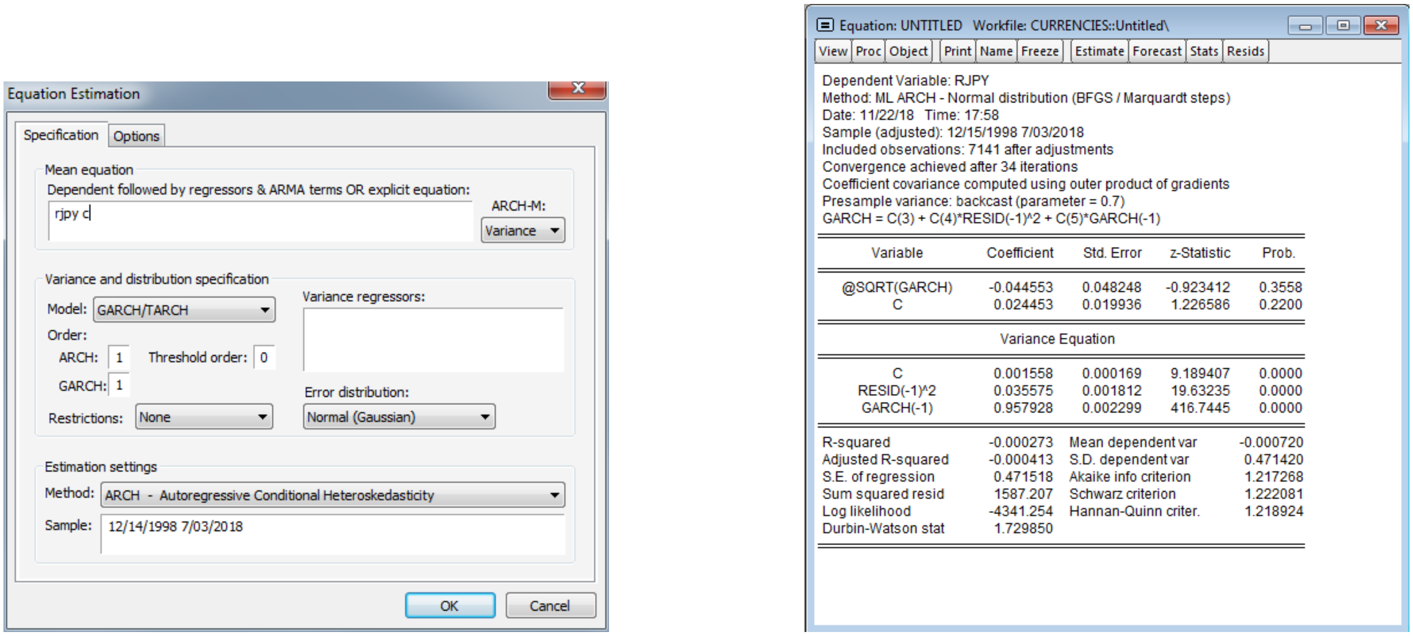


Figure 77: GARCH-in-Mean Model Estimation

19.5 Forecasting from GARCH Models

Reading: Brooks (2019, Section 9.18)

Forecasts from any of the GARCH models that can be estimated using EViews are obtained by using only a subsample of available data for model estimation, and then by clicking on the 'Forecast' button that appears after estimation of the required model has been completed. Suppose, for example, we stopped the estimation of the GARCH(1,1) model (with no asymmetries and no GARCH-in-mean term) for the Japanese yen returns at 2 August 2016 so as to keep the last two years of data for forecasting (i.e., the 'Forecast sample' is 8/03/2016 7/03/2018). Then click on the **Forecast** tab above the estimation results and the dialog box in Figure 78 will appear.

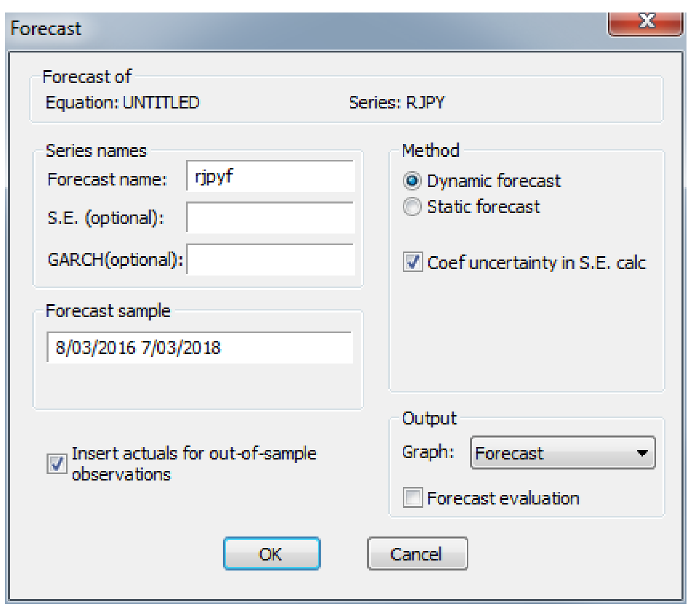


Figure 78: Forecasting from GARCH Models

Again, several options are available, including providing a name for the conditional mean and for the conditional variance forecasts, or whether to produce static (a series of rolling single-step-ahead) or dynamic (multiple-step-ahead) forecasts. The dynamic and static forecast plots that would be produced are given in Figure 79. Note that we have disabled the forecast evaluation to only show the graphs.

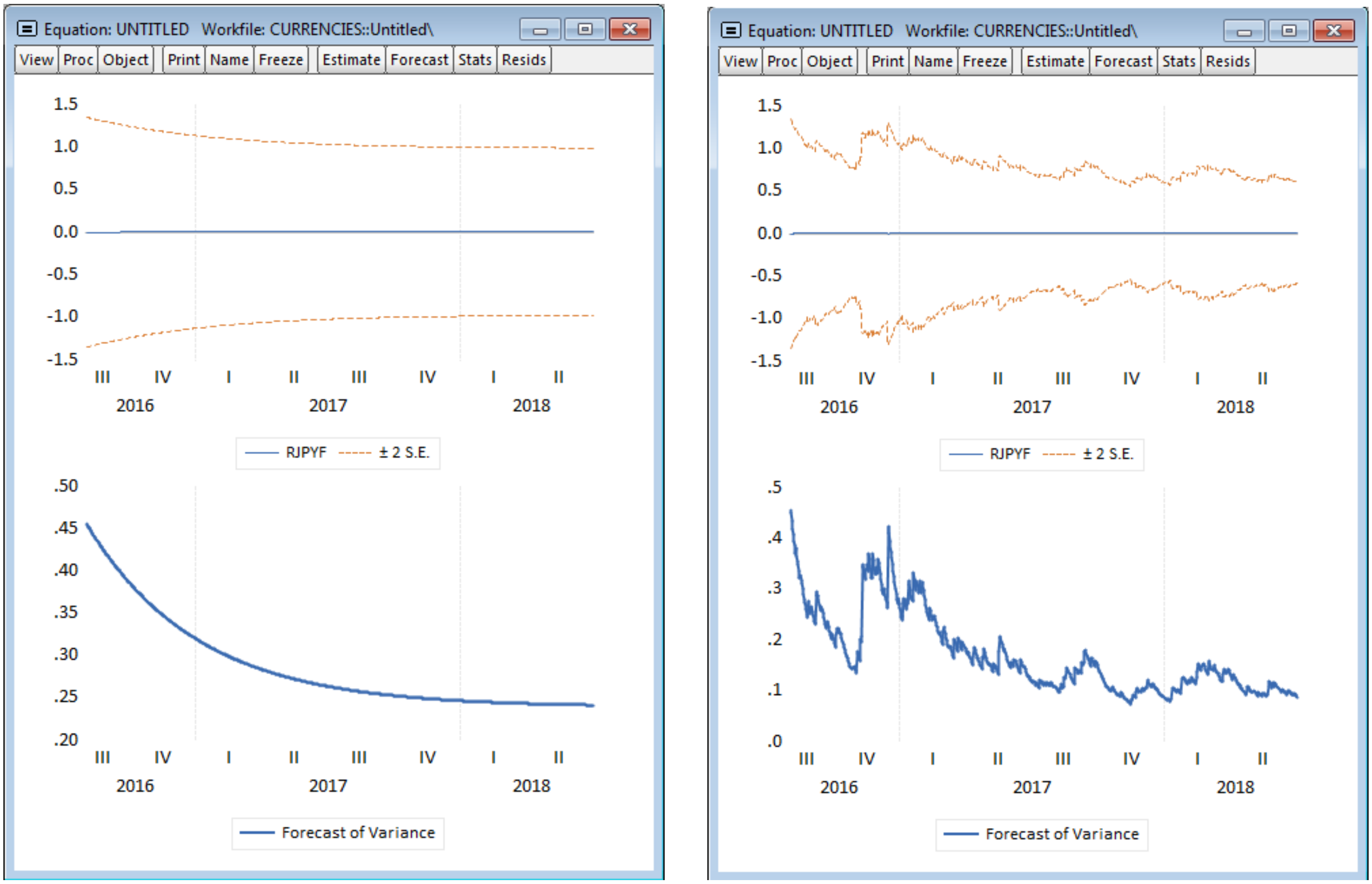


Figure 79: Dynamic (left panel) and Static (right panel) Forecast Using GARCH-in-Mean Model

The dynamic forecasts show a completely flat forecast structure for the mean (since the conditional mean equation includes only a constant term), while at the end of the in-sample estimation period the value of the conditional variance was at a historically high level relative to its unconditional average. Therefore, the forecasts converge upon their long-term mean value from above as the forecast horizon increases. Notice also that there are no ±2-standard error band confidence intervals for the conditional variance forecasts; to compute these would require some kind of estimate of the variance of variance, which is beyond the capability of the built-in functions of the EViews software. The conditional variance forecasts provide the basis for the standard error bands that are given by the dotted red lines around the conditional mean forecast. Because the conditional variance forecasts rise gradually as the forecast horizon increases, so the standard error bands widen slightly.

It is evident that the variance forecasts has a spike in the last quarter of 2016. Since these are a series of rolling one-step ahead forecasts for the conditional variance, they show much more volatility than for the dynamic forecasts. This volatility also results in more variability in the standard error bars around the conditional mean forecasts. Note that while the forecasts are updated daily based on new information that feeds into the forecasts, the parameter estimates themselves are not updated. Thus, towards the end of the sample the forecasts are based on estimates almost two years old. If we wanted to update the model estimates as we rolled through the sample, we would need to write some code to do this within a loop – it would also run much more slowly as we would be estimating a lot of GARCH models rather than one. See later sections of this guide for a discussion of how to construct loops in EViews. Predictions can be similarly produced for any member of the GARCH family that is estimable with the software.

19.6 Estimation of Multivariate GARCH Models

Reading: Brooks (2019, Sections 9.20 and 9.21)

To estimate such a model, first you need to create a system that contains the variables to be used. **Highlight the three variables 'reur', 'rgbp', and 'rjpy'** and then **right click** the mouse. Choose **Open/as System. . .**, as in Figure 80, left panel.

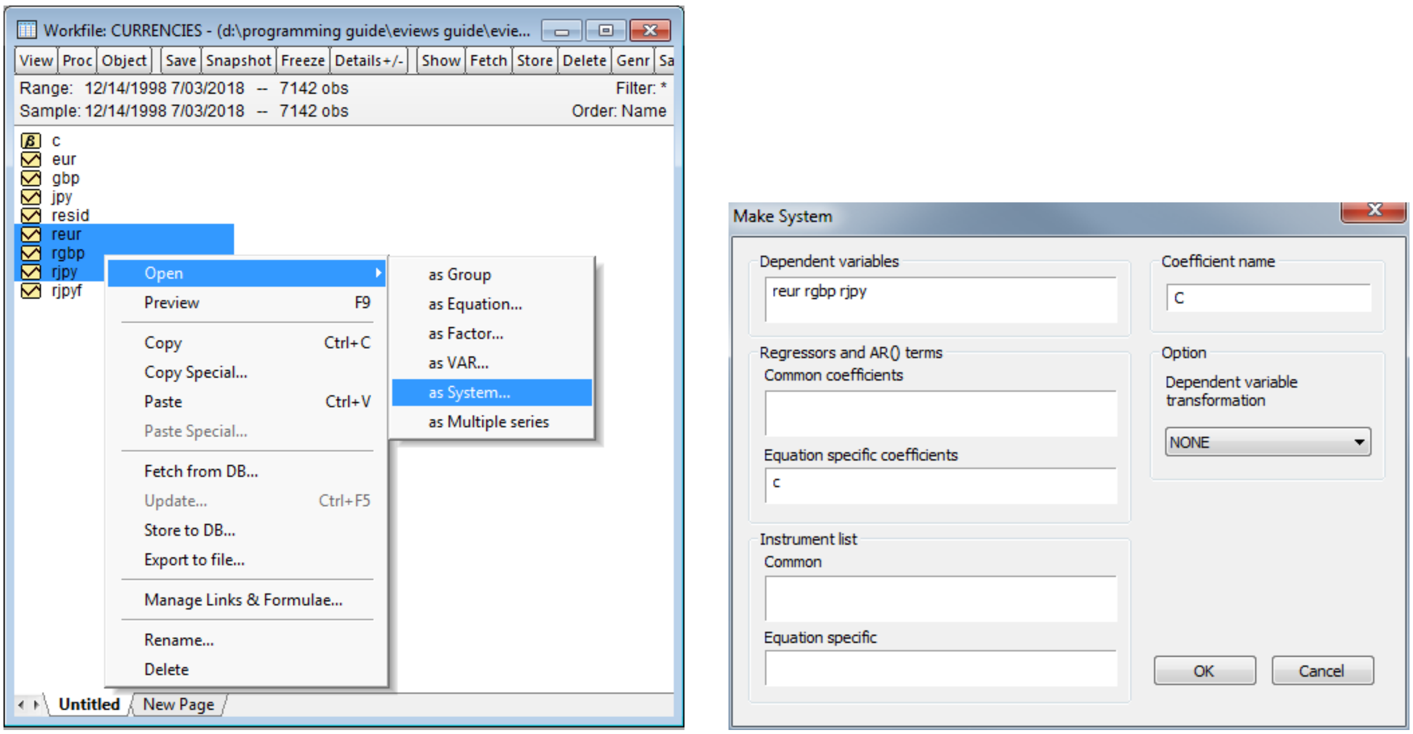


Figure 80: Creating a System

Since no explanatory variables will be used in the conditional mean equation, all of the default choices can be retained, so just click **OK** in Figure 80, right panel. A system box containing the three equations with just intercepts will be seen. Then click **Proc/Estimate . . .** for the 'System Estimation' window. Change the 'Estimation method' to **ARCH – Conditional Heteroscedasticity** and Figure 81 will appear.

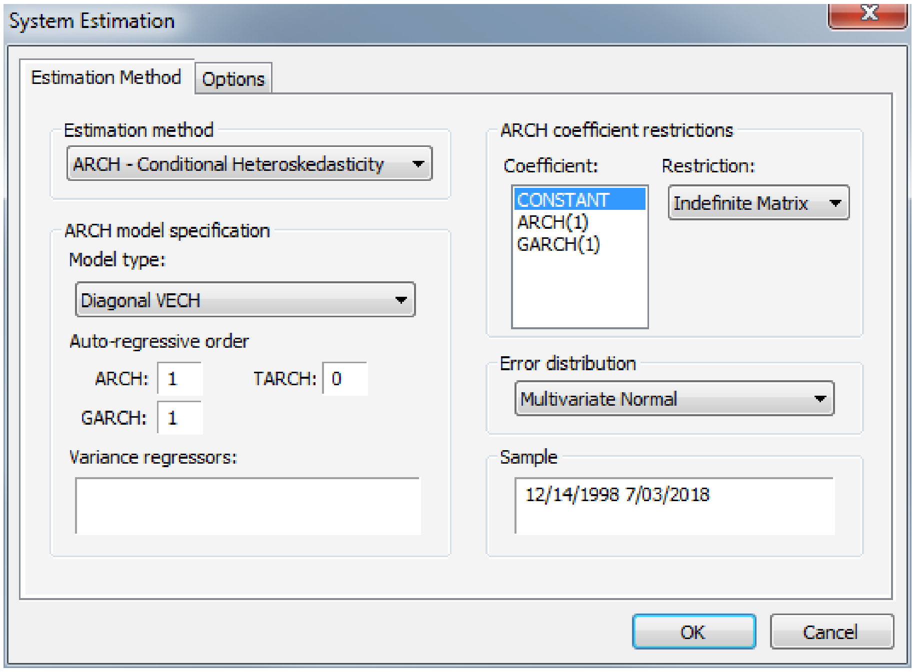


Figure 81: Estimation of a Multivariate GARCH Model

EViews permits the estimation of three important classes of multivariate GARCH models: the diagonal VECH, the constant conditional correlation and the diagonal BEKK. For the error distribution, either a multivariate normal or a multivariate Student's t can be used. Additional exogenous variables can be incorporated into the variance equation, and asymmetries can be allowed for. Clicking on the Options tab allows the user to modify the settings used in the optimisation, which can be useful in case there are problems with the model estimation such as non-convergence or convergence to implausible parameter estimates. Leaving all of these options as the defaults and clicking **OK** would yield the results in Figure 82.[15]

[15] The complexity of this model causes it to take longer to estimate than any of the univariate GARCH or other models examined previously.

System: UNTITLED Workfile: CURRENCIES::Untitled\

View | Proc | Object | Print | Name | Freeze | InsertTxt | Estimate | Spec | Stats | Resids

System: UNTITLED
Estimation Method: ARCH Maximum Likelihood (BFGS / Marquardt steps)
Covariance specification: Diagonal VECH
Date: 11/23/18 Time: 17:20
Sample: 12/15/1998 7/03/2018
Included observations: 7141
Total system (balanced) observations 21423
Presample covariance: backcast (parameter =0.7)
Convergence achieved after 86 iterations
Coefficient covariance computed using outer product of gradients

	Coefficient	Std. Error	z-Statistic	Prob.
C(1)	-0.004569	0.004171	-1.095380	0.2734
C(2)	-0.003079	0.004044	-0.761426	0.4464
C(3)	0.008863	0.004531	1.955966	0.0505
Variance Equation Coefficients				
C(4)	0.000634	7.10E-05	8.930006	0.0000
C(5)	0.000554	6.71E-05	8.253473	0.0000
C(6)	0.000901	0.000112	8.069672	0.0000
C(7)	0.000905	9.77E-05	9.255196	0.0000
C(8)	0.000625	9.26E-05	6.749794	0.0000
C(9)	0.002076	0.000173	11.98240	0.0000
C(10)	0.025843	0.001202	21.49392	0.0000
C(11)	0.024016	0.000985	24.37400	0.0000
C(12)	0.035451	0.001735	20.42995	0.0000
C(13)	0.030738	0.000879	34.98305	0.0000
C(14)	0.031306	0.001859	16.83957	0.0000
C(15)	0.041416	0.001892	21.89230	0.0000
C(16)	0.971588	0.001215	799.5553	0.0000
C(17)	0.971432	0.001295	750.1084	0.0000
C(18)	0.953094	0.002563	371.8625	0.0000
C(19)	0.964568	0.001212	795.6693	0.0000
C(20)	0.955868	0.003016	316.9050	0.0000
C(21)	0.950407	0.002340	406.1777	0.0000

Log likelihood	-8969.496	Schwarz criterion	2.538207
Avg. log likelihood	-0.418685	Hannan-Quinn criter.	2.524952
Akaike info criterion	2.517994		

Equation: REUR = C(1)

R-squared	-0.000105	Mean dependent var	0.000190
Adjusted R-squared	-0.000105	S.D. dependent var	0.463953
S.E. of regression	0.463977	Sum squared resid	1537.062
Durbin-Watson stat	1.732085		

Equation: RGBP = C(2)

R-squared	-0.000242	Mean dependent var	0.003462
Adjusted R-squared	-0.000242	S.D. dependent var	0.420910
S.E. of regression	0.420961	Sum squared resid	1265.264
Durbin-Watson stat	1.620005		

Equation: RJPY = C(3)

R-squared	-0.000413	Mean dependent var	-0.000720
Adjusted R-squared	-0.000413	S.D. dependent var	0.471420
S.E. of regression	0.471518	Sum squared resid	1587.430
Durbin-Watson stat	1.729379		

Covariance specification: Diagonal VECH
GARCH = M + A1.*RESID(-1)*RESID(-1)' + B1.*GARCH(-1)
M is an indefinite matrix
A1 is an indefinite matrix*
B1 is an indefinite matrix*

Transformed Variance Coefficients

	Coefficient	Std. Error	z-Statistic	Prob.
M(1,1)	0.000634	7.10E-05	8.930006	0.0000
M(1,2)	0.000554	6.71E-05	8.253473	0.0000
M(1,3)	0.000901	0.000112	8.069672	0.0000
M(2,2)	0.000905	9.77E-05	9.255196	0.0000
M(2,3)	0.000625	9.26E-05	6.749794	0.0000
M(3,3)	0.002076	0.000173	11.98240	0.0000
A1(1,1)	0.025843	0.001202	21.49392	0.0000
A1(1,2)	0.024016	0.000985	24.37400	0.0000
A1(1,3)	0.035451	0.001735	20.42995	0.0000
A1(2,2)	0.030738	0.000879	34.98305	0.0000
A1(2,3)	0.031306	0.001859	16.83957	0.0000
A1(3,3)	0.041416	0.001892	21.89230	0.0000
B1(1,1)	0.971588	0.001215	799.5553	0.0000
B1(1,2)	0.971432	0.001295	750.1084	0.0000
B1(1,3)	0.953094	0.002563	371.8625	0.0000
B1(2,2)	0.964568	0.001212	795.6693	0.0000
B1(2,3)	0.955868	0.003016	316.9050	0.0000
B1(3,3)	0.950407	0.002340	406.1777	0.0000

* Coefficient matrix is not PSD.

Figure 82: Multivariate GARCH Model Results

The first panel of the table in Figure 82 presents the conditional mean estimates; in this example, only intercepts were used in the mean equations. The next panel shows the variance equation coefficients, followed by some measures of goodness of fit for the model as a whole and then for each individual mean equation. The final panel presents the transformed variance coefficients, which in this case are identical to the panel of variance coefficients since no transformation is conducted with normal errors (these would only be different if a Student's t specification were used). It is evident that the parameter estimates are all both plausible and statistically significant. A number of useful further steps can be conducted once the model has been estimated, all of which are available by clicking the 'View' button. For example, we can plot the series of residuals, or estimate the correlations between them. Or by clicking on 'Conditional variance', we can list or plot the values of the conditional variances and covariances or correlations over time. We can also test for autocorrelation and normality of the errors.

20 Modelling Seasonality in Financial Data

20.1 Dummy Variables for Seasonality

Reading: Brooks (2019, Section 10.3)

The most commonly observed calendar effect in monthly data is the *January effect*. In order to examine whether there is indeed a January effect in a monthly time series regression, a dummy variable is created that takes the value 1 only in the months of January. This is most easily achieved by generating a new series via **Quick/Generate Series...** and specifying the new series as

jandum = @month=1

This command will create a variable called JANDUM equal to 1, if the month is January and 0 otherwise. In this way, dummy variables can be created without manually changing the entries.

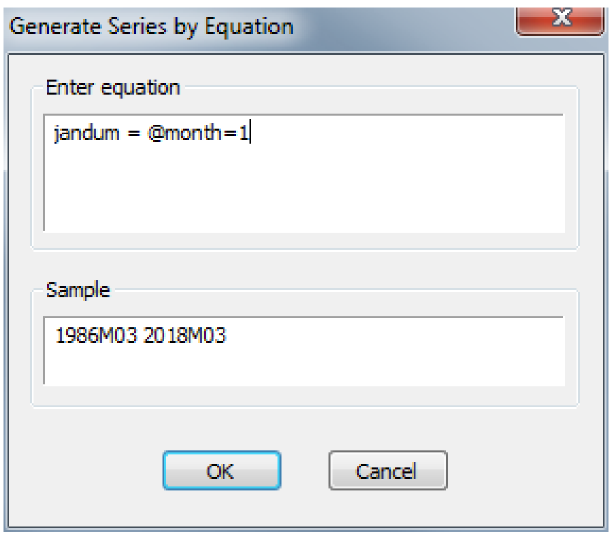
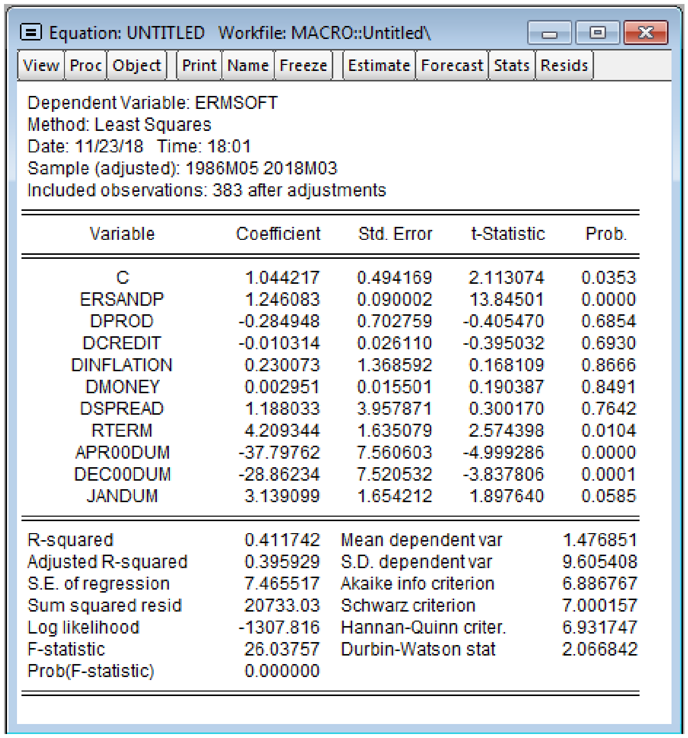


Figure 83: Create a Seasonal Dummy Variable

Returning to the Microsoft stock price example in the **'macro.wf1'** workfile, **create this variable** using the methodology described above as in Figure 83, and run the regression again including the new dummy variable for January and for April and December 2000. The results of this regression can be found in are in Figure 84.

As can be seen, the dummy is statistically significant at the 10% level, and it has the expected positive sign. The coefficient value of 3.139, suggests that on average and holding everything else equal, Microsoft stock returns are around 3.1% higher in January than the average for other months of the year.



Equation: UNTITLED Workfile: MACRO::Untitled\

View | Proc | Object | Print | Name | Freeze | Estimate | Forecast | Stats | Resids

Dependent Variable: ERMSOFT
Method: Least Squares
Date: 11/23/18 Time: 18:01
Sample (adjusted): 1986M05 2018M03
Included observations: 383 after adjustments

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	1.044217	0.494169	2.113074	0.0353
ERSANDP	1.246083	0.090002	13.84501	0.0000
DPROD	-0.284948	0.702759	-0.405470	0.6854
DCREDIT	-0.010314	0.026110	-0.395032	0.6930
DINFLATION	0.230073	1.368592	0.168109	0.8666
DMONEY	0.002951	0.015501	0.190387	0.8491
DSPREAD	1.188033	3.957871	0.300170	0.7642
RTERM	4.209344	1.635079	2.574398	0.0104
APR00DUM	-37.79762	7.560603	-4.999286	0.0000
DEC00DUM	-28.86234	7.520532	-3.837806	0.0001
JANDUM	3.139099	1.654212	1.897640	0.0585

R-squared	0.411742	Mean dependent var	1.476851
Adjusted R-squared	0.395929	S.D. dependent var	9.605408
S.E. of regression	7.465517	Akaike info criterion	6.886767
Sum squared resid	20733.03	Schwarz criterion	7.000157
Log likelihood	-1307.816	Hannan-Quinn criter.	6.931747
F-statistic	26.03757	Durbin-Watson stat	2.066842
Prob(F-statistic)	0.000000		

Figure 84: Linear Regression Including a Monthly Seasonal Dummy Variable

20.2 Estimating Markov Switching Models

Reading: Brooks (2019, Section 10.7)

Markov switching models can be estimated easily in recent versions of EViews. The example we will consider relates to the changes in house prices series used previously. So **Re-open the 'UKHP.wf1' file**, click **Quick/Estimate Equation** and then under 'Estimation Settings, Method', **Change LS Least squares (NLS and ARMA)** to the option, **SWITCHREG - Switching Regression** and complete the dialog box as in Figure 85, left panel.

The first box will include the dependent variable followed by a list of regressors that are allowed to vary across regimes. To estimate a simple switching model with just a varying intercept in each state, include only the constant. Any variables whose associated parameters should not be allowed to vary across regimes should be listed in the second box. To allow the variances to be different across the regimes, **tick the 'Regime specific error variances' box**. We could choose more regimes but for now **select '2'**. As usual, there is an 'Options' tab which allows the user to specify how the estimation and computation of standard errors is conducted. However, this can be left at the default options so click **OK** and the results will appear as Figure 85, right panel.

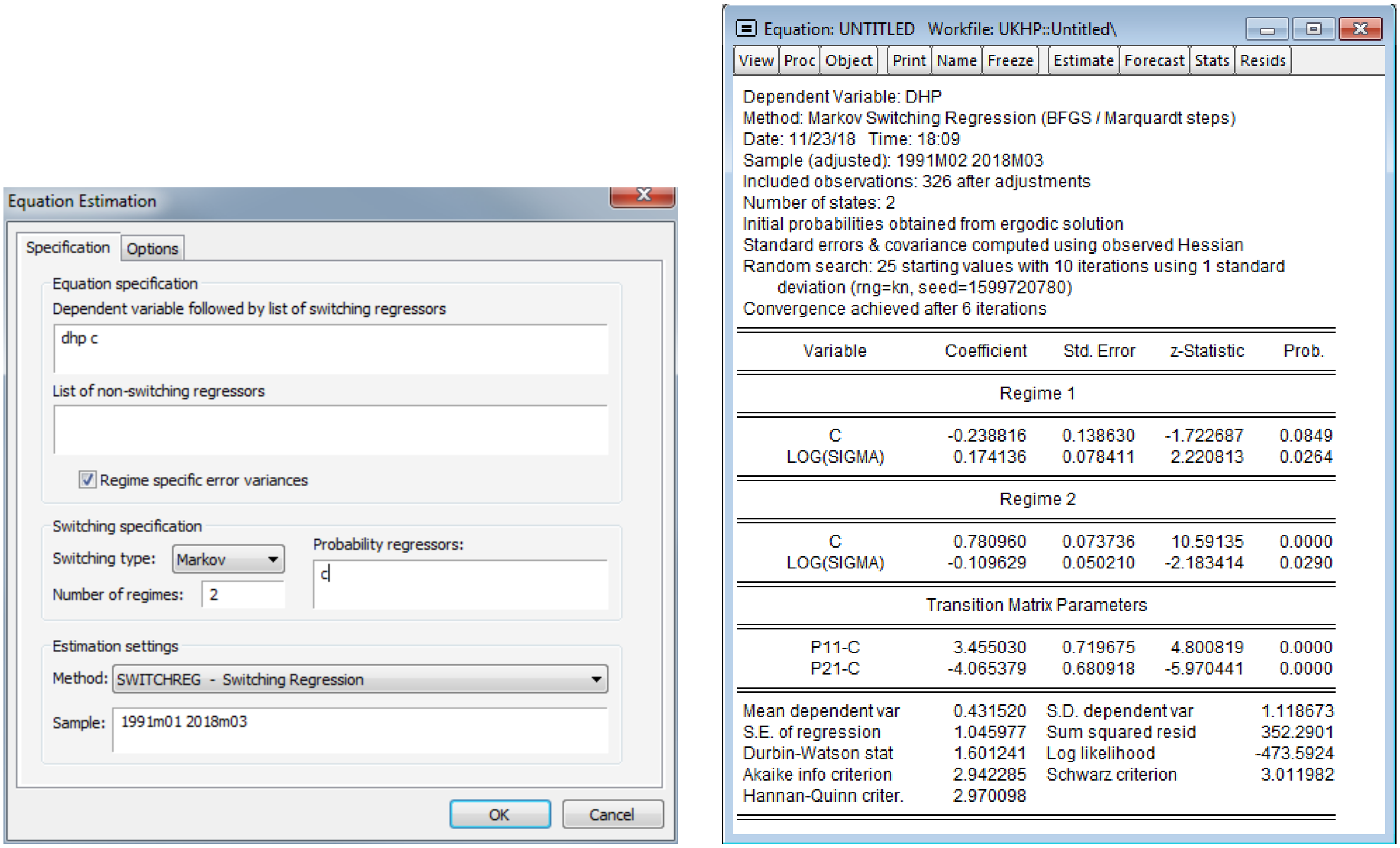


Figure 85: Constructing a Markov Switching Model

Two distinct regimes have been identified: regime 1 with a negative mean return (corresponding to a price fall of 0.24% per month) and a relatively high volatility, whereas regime 2 has a high average price increase of 0.78% per month and a much lower standard deviation.

To see the transition probability matrix, click **View/Regime Results/Transition Results...** and then select **Summary** and click **OK**. The regimes are fairly stable, with probabilities of around 97% of remaining in a given regime in the next period. The average durations are 59 months for regime 1 and 33 months for regime 2, which is again indicative of the stability of the regimes.

To examine the fitted states over time, select View/Regime Results/Regime Probabilities. It is then possible to choose the one-step ahead probabilities, or the filtered or smoothed probabilities. The smoothed probabilities are estimated using the entire sample whereas the filtered probabilities use a recursive approach using only the information available at time t to compute the probability of being in each regime at time t. Selecting the smoothed probabilities in a single graph gives the plot in Figure 86.

The first thing to note is that of course the two values always sum to one. Examining how the graphs move over time, the probability of being in regime 2 was close to zero until the mid-1990s, corresponding to a period of low or negative house price growth. The behaviour then changed and the probability of being in the low and negative growth state (regime 1) fell to zero and the housing market enjoyed a period of good performance until around 2006 when the regimes became less stable until early 2013 when the market again appeared to have turned to regime 2. There is evidence that the market state is again changing to low/negative growth at the end of the sample period (March 2018).

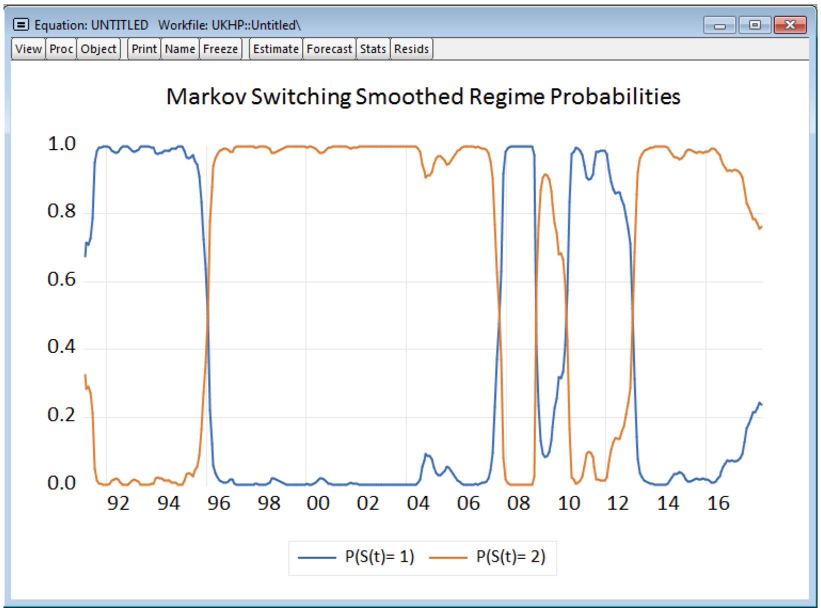


Figure 86: Smoothed Probabilities of Being in Regimes 1 and 2

21 Panel Data

Reading: Brooks (2019, Chapter 11)

The estimation of panel models, both fixed and random effects, is very easy with EViews; the harder part is organising the data so that the software can recognise that you have a panel of data and can apply the techniques accordingly. Therefore it is recommended to organise the data beforehand, such that it is in a long format containing all firm year observations in one table.

The application to be considered here is that of a variant on an early test of the CAPM due to Fama and MacBeth (1973). Their test involves a 2-step estimation procedure: first, the betas are estimated in separate time-series regressions for each firm, and second, for each separate point in time, a cross-sectional regression of the excess returns on the betas is conducted

$$R_{it} - R_{ft} = \lambda_0 + \lambda_m \beta_{Pi} + u_i \tag{12}$$

where the dependent variable, $R_{it} - R_{ft}$, is the excess return of the stock i at time t and the independent variable is the estimated beta for the portfolio (P) that the stock has been allocated to. The betas of the firms themselves are not used on the RHS, but rather the betas of portfolios formed on the basis of firm size. If the CAPM holds, then λ_0 should not be significantly different from zero and λ_m should approximate the (time average) equity market risk premium, $R_m - R_f$. Fama and MacBeth (1973) proposed estimating this second stage (cross-sectional) regression separately for each time period, and then taking the average of the parameter estimates to conduct hypothesis tests. However, one could also achieve a similar objective using a panel approach.

We will use an example in the spirit of Fama–MacBeth comprising the annual returns and 'second pass betas' for eleven years on 2,500 UK firms.[16] To import the excel file 'panel.xls' follow the four steps in Figure 87.

The settings in steps 1 to 3 do not need to be changed. In step 4 (Figure 87, lower right panel) we specify the Basis structure as **'Dated Panel'**, we can then define the column **firm_ident** as the Cross section ID series and the column **'year'** as the Date series. By clicking **OK**, the panel is set up and ready for use. Don't forget to save the workfile before closing EViews though.

[16] Computation by Keith Anderson and the author. There would be some significant limitations of this analysis if it purported to be a piece of original research, but the range of freely available panel datasets is severely limited and so hopefully it will suffice as an example of how to estimate panel models with EViews. No doubt readers, with access to a wider range of data, will be able to think of much better applications.

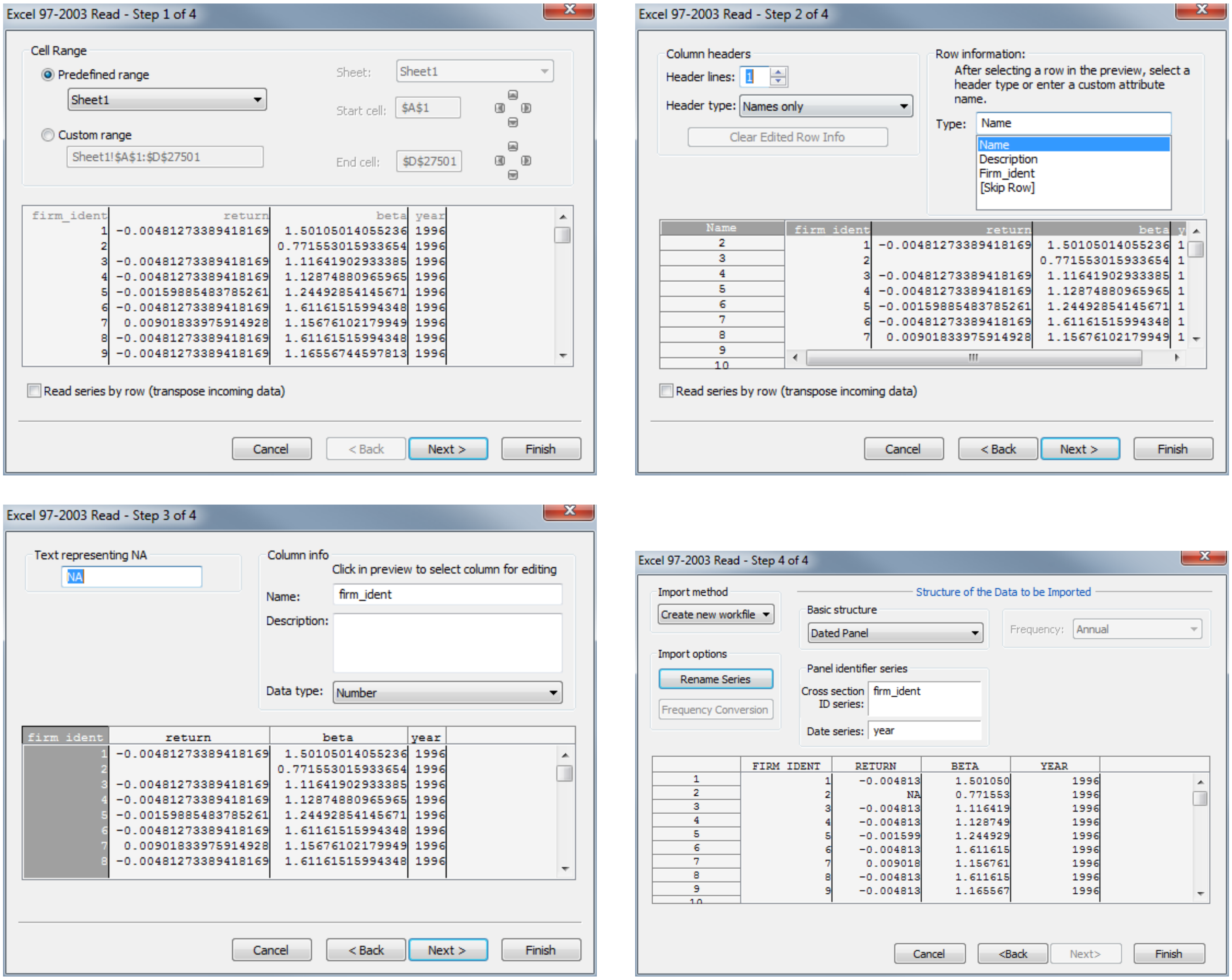
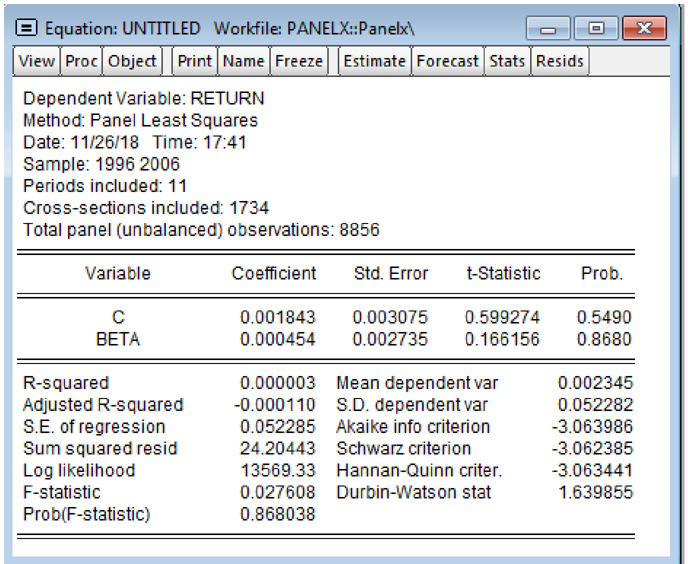


Figure 87: Importing the Data and Creating a Panel Dataset

To estimate panel regressions, click **Quick/Estimate Equation. . .** and then the Equation Estimation window will open. For the variables, enter **return c beta** in the Equation Specification window. If you click on the **Panel Options** tab, you will see that a number of options specific to panel data models are available. The most important of these is the first box, where either fixed or random effects can be chosen. The default is for neither, which would effectively imply a simple pooled regression, so **estimate a model with neither fixed nor random effects** first. The results would be as in Figure 88.



Equation: UNTITLED Workfile: PANELX::Panelx\

View | Proc | Object | Print | Name | Freeze | Estimate | Forecast | Stats | Resids

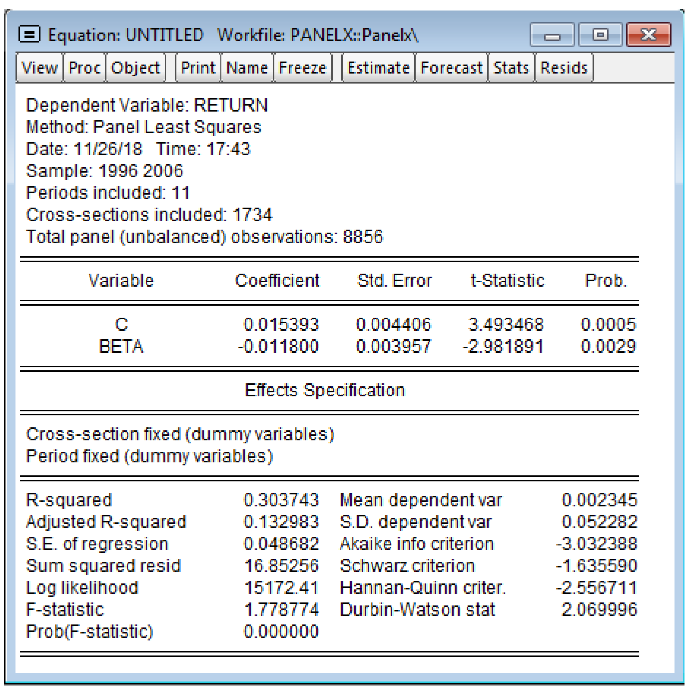
Dependent Variable: RETURN
Method: Panel Least Squares
Date: 11/26/18 Time: 17:41
Sample: 1996 2006
Periods included: 11
Cross-sections included: 1734
Total panel (unbalanced) observations: 8856

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	0.001843	0.003075	0.599274	0.5490
BETA	0.000454	0.002735	0.166156	0.8680

R-squared	0.000003	Mean dependent var	0.002345
Adjusted R-squared	-0.000110	S.D. dependent var	0.052282
S.E. of regression	0.052285	Akaike info criterion	-3.063986
Sum squared resid	24.20443	Schwarz criterion	-3.062385
Log likelihood	13569.33	Hannan-Quinn criter.	-3.063441
F-statistic	0.027608	Durbin-Watson stat	1.639855
Prob(F-statistic)	0.868038		

Figure 88: Pooled Regression Results Using No Fixed Effects

We can see that neither the intercept nor the slope is statistically significant. The returns in this regression are in proportion terms rather than percentages, so the slope estimate of 0.000454 corresponds to a risk premium of 0.0454% per year, whereas the (unweighted average) excess return for all firms in the sample is around -2% per year. But this pooled regression assumes that the intercepts are the same for each firm and for each year. This may be an inappropriate assumption, and we could instead estimate a model with firm-fixed and time-fixed effects, which will allow for latent firm-specific and time-specific heterogeneity, respectively, as shown in Figure 89.



Equation: UNTITLED Workfile: PANELX::Panelx\

View | Proc | Object | Print | Name | Freeze | Estimate | Forecast | Stats | Resids

Dependent Variable: RETURN
Method: Panel Least Squares
Date: 11/26/18 Time: 17:43
Sample: 1996 2006
Periods included: 11
Cross-sections included: 1734
Total panel (unbalanced) observations: 8856

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	0.015393	0.004406	3.493468	0.0005
BETA	-0.011800	0.003957	-2.981891	0.0029

Effects Specification

Cross-section fixed (dummy variables)
Period fixed (dummy variables)

R-squared	0.303743	Mean dependent var	0.002345
Adjusted R-squared	0.132983	S.D. dependent var	0.052282
S.E. of regression	0.048682	Akaike info criterion	-3.032388
Sum squared resid	16.85256	Schwarz criterion	-1.635590
Log likelihood	15172.41	Hannan-Quinn criter.	-2.556711
F-statistic	1.778774	Durbin-Watson stat	2.069996
Prob(F-statistic)	0.000000		

Figure 89: Panel Regression Results Using Year and Firm Fixed Effects

We can see that the estimate on the beta parameter is now negative and statistically significant, while the intercept is positive and statistically significant. If we wish to see the fixed effects (i.e., to see the values of the dummy variables for each firm and for each point in time), we could click on View/Fixed/Random

Effects and then either Cross-Section Effects or Period Effects (the latter are what EViews calls time-fixed effects).

Next, it is worth determining whether the fixed effects are necessary or not by running a redundant fixed effects test. To do this, click **View/Fixed/Random Effects Testing/Redundant Fixed Effects – Likelihood Ratio Test**. The output n Figure 90 will be seen.

Equation: UNTITLED Workfile: PANELX::Panelx\

View | Proc | Object | Print | Name | Freeze | Estimate | Forecast | Stats | Resids

Redundant Fixed Effects Tests
Equation: Untitled
Test cross-section and period fixed effects

Effects Test	Statistic	d.f.	Prob.
Cross-section F	1.412240	(1733,7111)	0.0000
Cross-section Chi-square	2619.415700	1733	0.0000
Period F	63.169151	(10,7111)	0.0000
Period Chi-square	753.703045	10	0.0000
Cross-Section/Period F	1.779777	(1743,7111)	0.0000
Cross-Section/Period Chi-square	3206.166621	1743	0.0000

Figure 90: Likelihood Ratio Test for Redundant Fixed Effects

Note that EViews will also present the results for a restricted model where only cross-sectional fixed effects and no period fixed effects are allowed for, and then a restricted model where only period fixed effects are allowed for.[17] Interestingly, the cross-sectional only fixed effects model parameters are not qualitatively different from those of the initial pooled regression, so it is the period fixed effects that make a difference. Three different redundant fixed effects tests are employed, each in both χ^2 and F-test versions, for: (1) restricting the cross-section fixed effects to zero; (2) restricting the period fixed effects to zero; and (3) restricting both types of fixed effects to zero. In all three cases, the p-values associated with the test statistics are zero to four decimal places, indicating that the restrictions are not supported by the data and that a pooled sample could not be employed.

Next, estimate a **random effects** model by selecting this from the panel estimation option tab. As for fixed effects, the random effects could be along either the cross-sectional or period dimensions, but select random effects **for the firms** (i.e., cross-sectional) but not **over time**. The results are observed as in Figure 91.

The slope estimate is again of a different order of magnitude compared with both the pooled and the fixed effects regressions. It is of interest to determine whether the random effects model passes the Hausman test for the random effects being uncorrelated with the explanatory variables. To do this, click **View/Fixed/Random Effects Testing/Correlated Random Effects – Hausman Test**. The results in Figure 92 are observed, with only the top panel that reports the Hausman test results being displayed here.

[17] These models are not shown to preserve space.

Equation: UNTITLED Workfile: PANELX::Panelx\

View | Proc | Object | Print | Name | Freeze | Estimate | Forecast | Stats | Resids

Dependent Variable: RETURN
Method: Panel EGLS (Cross-section random effects)
Date: 11/26/18 Time: 17:51
Sample: 1996 2006
Periods included: 11
Cross-sections included: 1734
Total panel (unbalanced) observations: 8856
Swamy and Arora estimator of component variances

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	0.003281	0.003267	1.004366	0.3152
BETA	-0.001499	0.002894	-0.518160	0.6044

Effects Specification

	S.D.	Rho
Cross-section random	0.012366	0.0560
Idiosyncratic random	0.050763	0.9440

Weighted Statistics

R-squared	0.000030	Mean dependent var	0.001663
Adjusted R-squared	-0.000083	S.D. dependent var	0.051095
S.E. of regression	0.051106	Sum squared resid	23.12475
F-statistic	0.264896	Durbin-Watson stat	1.716476
Prob(F-statistic)	0.606789		

Unweighted Statistics

R-squared	-0.000245	Mean dependent var	0.002345
Sum squared resid	24.21044	Durbin-Watson stat	1.639503

Figure 91: Panel Regression with Random Firm Effects and No Year Effects

The p-value for the test is less than 1%, indicating that the random effects model is not appropriate and that the fixed effects specification is to be preferred.

Equation: UNTITLED Workfile: PANELX::Panelx\

View | Proc | Object | Print | Name | Freeze | Estimate | Forecast | Stats | Resids

Correlated Random Effects - Hausman Test
Equation: Untitled
Test cross-section random effects

Test Summary	Chi-Sq. Statistic	Chi-Sq. d.f.	Prob.
Cross-section random	12.633579	1	0.0004

Figure 92: Hausman Test for Correlated Random Effects

21.1 Testing for Unit Roots and Cointegration in Panels

Reading: Brooks (2019, Section 11.8)

EViews provides a range of tests for unit roots within a panel structure, but all are based on the assumption of cross-sectional independence. Given that all of the approaches can be simultaneously employed at the click of a mouse, it seems preferable to do so in order to evaluate the sensitivity of the findings to the methodology employed. This illustration will use the Treasury bill/bond yields in

'fred.wf1' so **re-open this workfile**. We have already created a group to conduct the Johansen tests (call this 'interest'). If you did not name and save the group in the workfile, you will need to create a group again containing the yields on the Treasury instruments for all maturities: 3-months, 6-months, 1-year, 3-years, 5-years and 10-years. You could do this by highlighting the six series, then select Object/New Object/Group. The six series will already be included in the box and you can simply name the group and save the workfile.

Before running any panel unit root or cointegration tests, it is useful to start by examining the results of individual unit root tests on each series, so **run an augmented Dickey–Fuller test** on the levels of each yield series using a regression with an intercept but no deterministic trend, and use SIC to select the lag lengths in each case. You should find that for all series except GS10 the null hypothesis is rejected at the 10% level.

As we know from the discussion above, unit root tests have low power in the presence of small samples, and so the panel unit root tests may provide different results. To run these in EViews is easy, **double click on the group** that you created so that the spreadsheet view containing the six series appears. Then **click View/Unit Root Test...** and Figure 93 will appear.

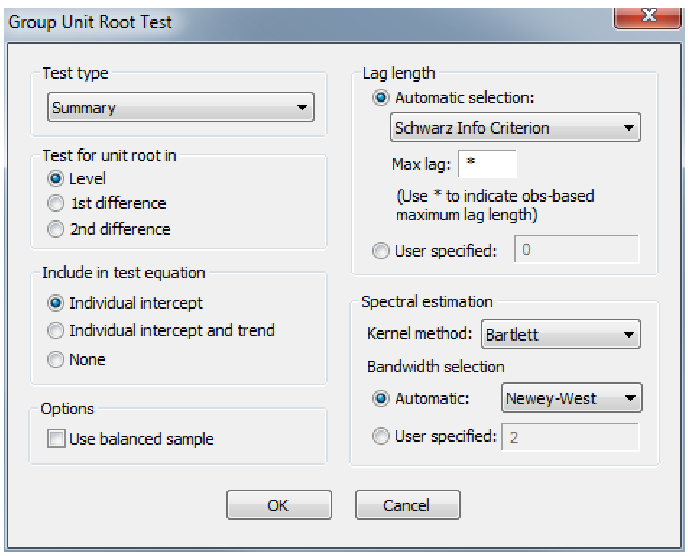
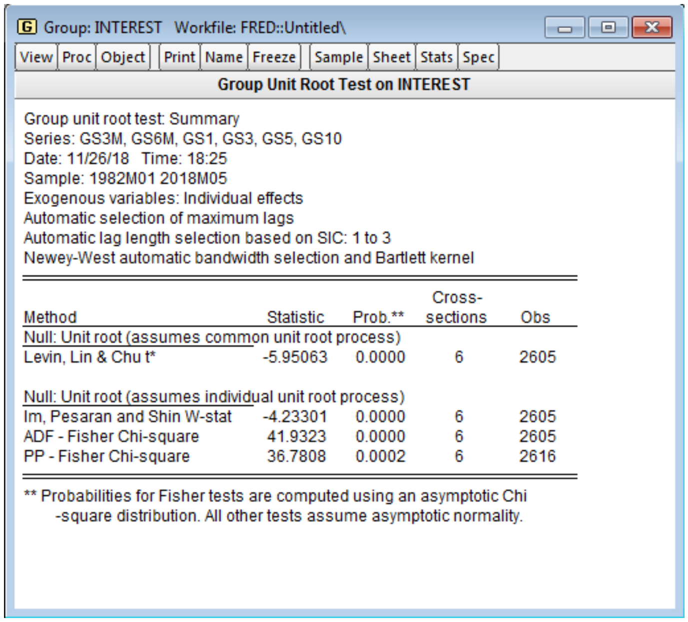


Figure 93: Conducting a Unit Root Test on a Group

The default options can be retained, and include printing a summary of the results from a range of panel unit root tests. Changing the test type box will enable the selection of a specific type of test, and in that case the results will be shown with more detail including the test regression. If we simply examine the summary results for now, just click **OK**, and we will see the results in Figure 94.



G Group: INTEREST Workfile: FRED::Untitled\

View | Proc | Object | Print | Name | Freeze | Sample | Sheet | Stats | Spec

Group Unit Root Test on INTEREST

Group unit root test: Summary
Series: GS3M, GS6M, GS1, GS3, GS5, GS10
Date: 11/26/18 Time: 18:25
Sample: 1982M01 2018M05
Exogenous variables: Individual effects
Automatic selection of maximum lags
Automatic lag length selection based on SIC: 1 to 3
Newey-West automatic bandwidth selection and Bartlett kernel

Method	Statistic	Prob.**	Cross-sections	Obs
Null: Unit root (assumes common unit root process)				
Levin, Lin & Chu t*	-5.95063	0.0000	6	2605
Null: Unit root (assumes individual unit root process)				
Im, Pesaran and Shin W-stat	-4.23301	0.0000	6	2605
ADF - Fisher Chi-square	41.9323	0.0000	6	2605
PP - Fisher Chi-square	36.7808	0.0002	6	2616

** Probabilities for Fisher tests are computed using an asymptotic Chi
-square distribution. All other tests assume asymptotic normality.

Figure 94: Results of a Unit Root Test on Interest Rates

Three lags were chosen on the basis of SIC for the ADF test including an intercept but no trend. Several tests are presented – first, the LLC test that assumes a common ρ for each series. The test statistic is -5.95 with p-value 0.0000 and thus the unit root null is rejected. Second, three tests that permit separate values of ρ for each series are presented. These are the IPS test and then two variants of the Fisher test proposed by Maddala and Wu (1999) and Choi (2001) – one for the ADF test and one for the Phillips–Perron test. In all cases, the test statistics are well above the critical values, indicating that the series contain no unit roots. Thus the conclusions from the panel unit root test are the same as those of the individual ones – in this case, using the panel did not make any difference, perhaps because $N = 6$ is quite small, while $T = 437$ for each series is quite large, and so the additional benefits from using a panel are minimal. If we wanted to run a panel cointegration test, this could be done simply by selecting **View/Cointegration Test** from the group spreadsheet view. Then either a Johansen-based system approach or a single equation approach can be chosen.

22 Limited Dependent Variable Models

Reading: Brooks (2019, Chapter 12)

Estimating limited dependent variable models in EViews is very simple. The example that will be considered here concerns whether it is possible to determine the factors that affect the likelihood that a student will fail his/her MSc. The data comprise a sample from the actual records of failure rates for five years of MSc students in finance at the ICMA Centre, University of Reading, contained in the spreadsheet **'MSc_fail.xls'**. While the values in the spreadsheet are all genuine, only a sample of 100 students is included for each of five years who completed (or not as the case may be!) their degrees in the years 2003 to 2007 inclusive. Therefore, the data should not be used to infer actual failure rates on these programmes. The idea for this example is taken from a study by Heslop and Varotto (2007) which seeks to propose an approach to preventing systematic biases in admissions decisions.[18]

The objective here is to analyse the factors that affect the probability of failure of the MSc. The dependent variable ('fail') is binary and takes the value 1 if that particular candidate failed at the first attempt in terms of his/her overall grade and 0 elsewhere. Therefore, a model that is suitable for limited dependent variables is required, such as a logit or probit.

The other information in the spreadsheet that will be used includes the age of the student, a dummy variable taking the value 1 if the student is female, a dummy variable taking the value 1 if the student has work experience, a dummy variable taking the value 1 if the student's first language is English, a country code variable that takes values from 1 to 10,[19] a dummy variable that takes the value 1 if the student already has a postgraduate degree, a dummy variable that takes the value 1 if the student achieved an A-grade at the undergraduate level (i.e., a first-class honours degree or equivalent), and a dummy variable that takes the value 1 if the undergraduate grade was less than a B-grade (i.e., the student received the equivalent of a lower second-class degree). The B-grade (or upper second-class degree) is the omitted dummy variable and this will then become the reference point against which the other grades are compared – see subsection 10.5. The reason why these variables ought to be useful predictors of the probability of failure should be fairly obvious and is therefore not discussed. To allow for differences in examination rules and in average student quality across the five-year period, year dummies for 2004, 2005, 2006 and 2007 are created and thus the year 2003 dummy will be omitted from the regression model.

First, **open a new workfile** that can accept 'unstructured/undated' series of length 500 observations and then **import the 13 variables**. The data are **organised by observation** and start in cell **A2**. The country code variable will require further processing before it can be used but the others are already in the appropriate format so, to begin, suppose that we estimate an LPM of fail on a constant, age, English, female and work experience. This would be achieved simply by running a linear regression in the usual way. While this model has a number of very undesirable features, as discussed above, it would nonetheless provide a useful benchmark with which to compare the more appropriate models estimated below.

[18]Note that since this guide uses only a subset of their sample and variables in the analysis, the results presented below may differ from theirs. Since the number of fails is relatively small, I deliberately retained as many fail observations in the sample as possible, which will bias the estimated failure rate upwards relative to the true rate.

[19]The exact identities of the countries involved are not revealed in order to avoid any embarrassment for students from countries with high relative failure rates, except that Country 8 is the UK!

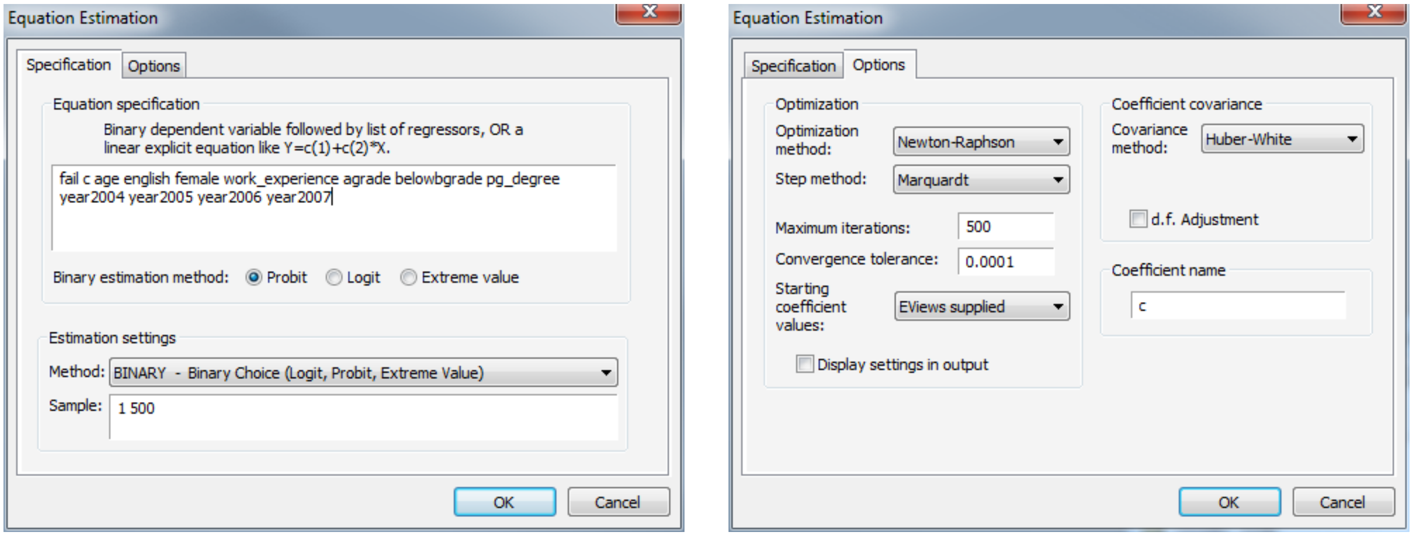


Figure 95: Probit Model Estimation

Next, estimate a probit model and a logit model using the same dependent and independent variables as above. Choose **Quick** and then **Equation Estimation**. Then type the dependent variable followed by the explanatory variables

FAIL C AGE ENGLISH FEMALE WORK_EXPERIENCE AGRADE BELOWBGRADE PG_DEGREE YEAR2004 YEAR2005 YEAR2006 YEAR2007

and then in the second window, marked 'Estimation settings', select **BINARY – Binary Choice (Logit, Probit, Extreme Value)** with the whole sample 1 500. The screen will appear as in Figure 95, left panel.

You can then choose either the probit or logit approach. Note that EViews also provides support for truncated and censored variable models and for multiple choice models, and these can be selected from the drop-down menu by choosing the appropriate method under 'estimation settings'. Suppose that here we wish to choose a probit model (the default). Click on the **Options** tab at the top of the window and this enables you to select **Robust Covariances** and **Huber/White**. This option will ensure that the standard error estimates are robust to heteroscedasticity (see Figure 95, right panel).

There are other options to change the optimisation method and convergence criterion. We do not need to make any modifications from the default here, so click **OK** and the results will appear. **Freeze and name this table** and then, for completeness, **estimate a logit model**. The results that you should obtain for the probit model are in Figure 96.

Equation: UNTITLED Workfile: MSC_FAIL::Msc_fail\

View | Proc | Object | Print | Name | Freeze | Estimate | Forecast | Stats | Resids

Dependent Variable: FAIL
Method: ML - Binary Probit (Newton-Raphson / Marquardt steps)
Date: 11/26/18 Time: 18:56
Sample: 1 500
Included observations: 500
Convergence achieved after 6 iterations
Coefficient covariance computed using the Huber-White method

Variable	Coefficient	Std. Error	z-Statistic	Prob.
C	-1.287210	0.609503	-2.111901	0.0347
AGE	0.005677	0.022559	0.251648	0.8013
ENGLISH	-0.093792	0.156226	-0.600362	0.5483
FEMALE	-0.194107	0.186201	-1.042460	0.2972
WORK_EXPERIENCE	-0.318247	0.151333	-2.102956	0.0355
AGRADE	-0.538814	0.231148	-2.331038	0.0198
BELOWBGRADE	0.341803	0.219301	1.558601	0.1191
PG_DEGREE	0.132957	0.225925	0.588502	0.5562
YEAR2004	0.349663	0.241450	1.448181	0.1476
YEAR2005	-0.108330	0.268527	-0.403422	0.6866
YEAR2006	0.673612	0.238536	2.823944	0.0047
YEAR2007	0.433785	0.247930	1.749630	0.0802

McFadden R-squared	0.088870	Mean dependent var	0.134000
S.D. dependent var	0.340993	S.E. of regression	0.333221
Akaike info criterion	0.765825	Sum squared resid	54.18582
Schwarz criterion	0.866976	Log likelihood	-179.4563
Hannan-Quinn criter.	0.805517	Deviance	358.9127
Restr. deviance	393.9204	Restr. log likelihood	-196.9602
LR statistic	35.00773	Avg. log likelihood	-0.358913
Prob(LR statistic)	0.000247		

Obs with Dep=0	433	Total obs	500
Obs with Dep=1	67		

Figure 96: Probit Regression Results

As can be seen, the pseudo-R^2 values are quite small at just below 9%, although this is often the case for limited dependent variable models. Only the work experience and A-grade variables and two of the year dummies have parameters that are statistically significant, and the Below B-grade dummy is almost significant at the 10% level in the probit specification (although less so in the logit). As the final two rows of the tables note, the proportion of fails in this sample is quite small, which makes it harder to fit a good model than if the proportions of passes and fails had been more evenly balanced. Various goodness of fit statistics can be examined by (from the logit or probit estimation output window) clicking **View/Goodness-of-fit Test (Hosmer-Lemeshow)**. A further check on model adequacy is to produce a set of 'in-sample forecasts' – in other words, to construct the fitted values. To do this, click on the **Forecast** tab after estimating the probit model and then **uncheck the forecast evaluation box** in the 'Output' window as the evaluation is not relevant in this case. All other options can be left as the default settings and then the plot of the fitted values will be shown as in Figure 97.

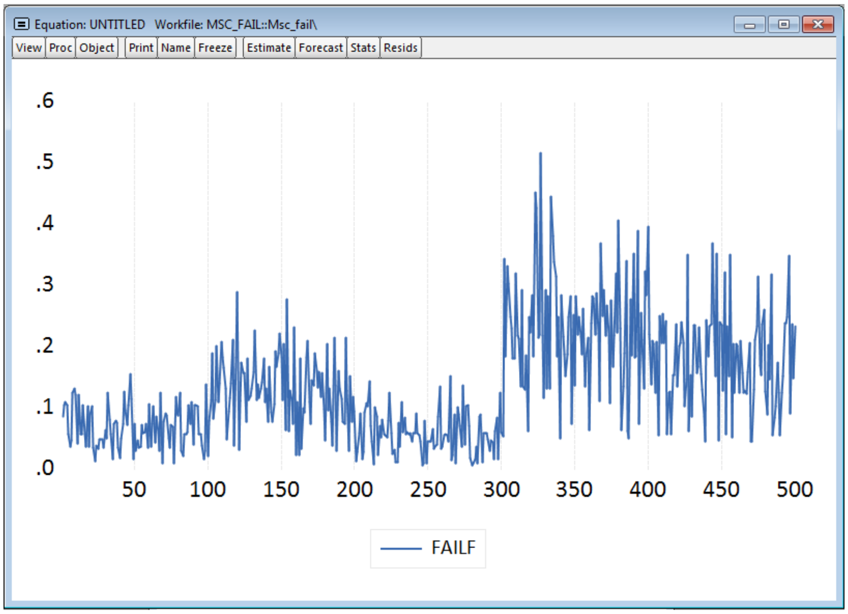


Figure 97: Fitted Values from the Failure Probit Regression

The unconditional probability of failure for the sample of students we have is only 13.4% (i.e., only 67 out of 500 failed), so an observation should be classified as correctly fitted if either $y_i = 1$ and $\hat{y}_i > 0.134$ or $y_i = 0$ and $\hat{y}_i < 0.134$. The easiest way to evaluate the model in EViews is to click **View/Actual,Fitted,Residual Table** from the logit or probit output screen. Then from this information we can identify that of the 67 students that failed, the model correctly predicted 46 of them to fail (and it also incorrectly predicted that 21 would pass). Of the 433 students who passed, the model incorrectly predicted 155 to fail and correctly predicted the remaining 278 to pass. EViews can construct an 'expectation-prediction classification table' automatically by clicking on **View/Expectation-Prediction Evaluation** and then entering the unconditional probability of failure as the cut-off when prompted (**0.134**). Overall, we could consider this a reasonable set of (in sample) predictions with 64.8% of the total predictions correct, comprising 64.2% of the passes correctly predicted as passes and 68.66% of the fails correctly predicted as fails.

It is important to note that, as discussed above, we cannot interpret the parameter estimates in the usual way. In order to be able to do this, we need to calculate the marginal effects. Unfortunately, EViews does not do this automatically, so the procedure is probably best achieved in a spreadsheet using the approach described in box 12.1 of Brooks (2019) for the logit model and analogously for the probit model. If we did this, we would end up with the statistics displayed in Table 3, which are interestingly quite similar in value to those obtained from the linear probability model.

Table 3: Marginal Effects for Logit and Probit Models for Probability of MSc Failure

Parameter	Logit	Probit
C	−0.2433	−0.1646
AGE	0.0012	0.0007
ENGLISH	−0.0178	−0.0120
FEMALE	−0.0360	−0.0248
WORK_EXPERIENCE	−0.0613	−0.0407
AGRADE	−0.1170	−0.0689
BELOWBGRADE	0.0606	0.0437
PGWORK_DEGREE	0.0229	0.0170
YEAR2004	0.0704	0.0447
YEAR2005	−0.0198	−0.0139
YEAR2006	0.1344	0.0862
YEAR2007	0.0917	0.0555

This table presents us with values that can be intuitively interpreted in terms of how the variables affect the probability of failure. For example, an age parameter value of 0.0012 implies that an increase in the age of the student by one year would increase the probability of failure by 0.12%, holding everything else equal, while a female student is around 2.5–3% (depending on the model) less likely than a male student with otherwise identical characteristics to fail. Having an A-grade (first class) in the bachelors degree makes a candidate either 6.89% or 12.7% (depending on the model) less likely to fail than an otherwise identical student with a B-grade (upper second-class degree). Since the year 2003 dummy has been omitted from the equations, this becomes the reference point. So students were more likely in 2004, 2006 and 2007, but less likely in 2005, to fail the MSc than in 2003. Finally, we should note that not all of the corresponding parameter estimates were statistically significant.

23 Simulations Methods

23.1 Deriving a Set of Dickey–Fuller Critical Values

Reading: Brooks (2019, Sections 13.1–13.7

Some EViews code for conducting such a simulation is given in Figure 98. The objective is to develop a set of critical values for Dickey–Fuller test regressions. The simulation framework considers sample sizes of 1,000, 500 and 100 observations. For each of these sample sizes, regressions with no constant or trend, a constant but no trend, and a constant and trend are conducted. 50,000 replications are used in each case, and the critical values for a one-sided test at the 1%, 5% and 10% levels are determined. The code can be found pre-written in a file entitled **'dickey_fuller.prg'** and is displayed in Figure 98.

Program: DICKEY_FULLER - (d:\programming guide\eviews guide\eviews_fil...

Run | Print | Save | SaveAs | Snapshot | Cut | Copy | Paste | InsertTxt | Find | Replace | Wrap+/- | LineNum+/-

```
' New workfile created called DF_CV, undated with 50000 observations
workfile DF_CV U 50000
rndseed 12345
series T_NONE
series T_CONST
series T_TREND
series Y
!NREPS = 50000
!NOBS = 1000
for !REPC=1 to !NREPS
   Y(1) = 0
   smpl @first+1 !NOBS+200
   Y = Y(-1)+nrnd
   series DY = Y - Y(-1)
   smpl @first+200 !NOBS+200
   equation eq.ls DY Y(-1) 'AR  model without constant or trend
   T_NONE(!REPC)=@tstats(1)
   equation eq.ls DY C Y(-1) 'AR model with constant
   T_CONST(!REPC)=@tstats(2)
   equation eq.ls DY C @trend Y(-1) 'AR model with constant and trend
   T_TREND(!REPC)=@tstats(3)
next
smpl @first !NREPS
scalar Q_NONE1=@quantile(T_NONE,0.01)
scalar Q_NONE5=@quantile(T_NONE,0.05)
scalar Q_NONE10=@quantile(T_NONE,0.1)
scalar Q_CONST1=@quantile(T_CONST,0.01)
scalar Q_CONST5=@quantile(T_CONST,0.05)
scalar Q_CONST10=@quantile(T_CONST,0.1)
scalar Q_TREND1=@quantile(t_TREND,0.01)
scalar Q_TREND5=@quantile(T_TREND,0.05)
scalar Q_TREND10=@quantile(T_TREND,0.1)
```

Figure 98: Program Code to Simulate ADF Critical Values

EViews programs are simply sets of instructions saved as plain text, so that they can be written from within EViews, or using a word processor or text editor. EViews program files must have a '.PRG' suffix. There are several ways to run the programs once written, but probably the simplest is to write all of the instructions first, and to save them. Then open the EViews software and choose **File, Open and Programs. . .** , and when prompted select the directory and file for the instructions. The program containing the instructions will then appear on the screen. To run the program, click on the **Run** button. EViews will then open a dialog box with several options, including whether to run the program in 'Verbose' or 'Quiet' mode. Choose Verbose mode to see the instruction line that is being run at each point in its execution (i.e., the screen is continually updated). This is useful for debugging programs or for running short programs. Choose Quiet to run the program without updating the screen display as it is running, which will make it execute (considerably) more quickly. The screen would appear as in Figure 99, left panel. Then click **OK** and off it goes! After some time the program should finish and the workfile in Figure 99, right panel, has been created.

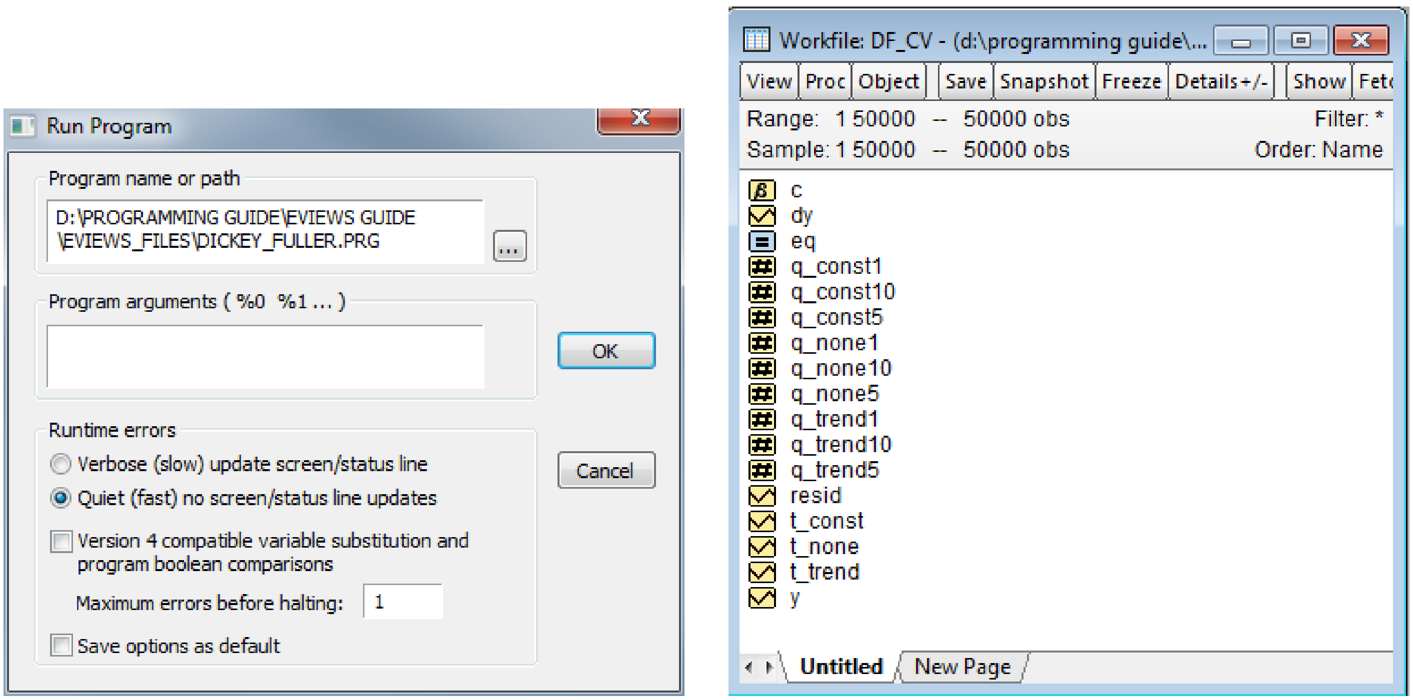


Figure 99: Window to Run a Program and Results

Although there are probably more efficient ways to structure the program than that given above, this sample code has been written in a style to make it easy to follow. The program would be run in the way described above. That is, it would be opened from within EViews, and then the **Run** button would be pressed and the mode of execution (Verbose or Quiet) chosen. Also remember that EViews is not case-sensitive, therefore the upper and lower case letters are only for better readability. In general, we tried to use lower case letter for commands and upper case letters for variables.

A first point to note is that comment lines are denoted by a $'$ symbol in EViews. The first line of code, **'workfile DF_CV U 50000'** will set up a new EViews workfile called **df_cv.wf1**, which will be undated **(U)** and will contain series of length 50,000. This step is required for EViews to have a place to put the output series since no other workfile will be opened by this program! In situations where the program requires an already existing workfile containing data to be opened, this line would not be necessary since any new results and objects created would be appended to the original workfile. **rndseed 12345** sets the random number seed that will be used to start the random draws.

'series T_NONE' creates a new series **T_NONE** that will be filled with NA elements. The series **T_NONE**, **T_CONST** and **T_TREND**, will hold the Dickey–Fuller test statistics for each replication, for the three cases (no constant or trend, constant but no trend, constant and trend, respectively). **!NREPS=50000** and **!NOBS=1000** set the number of replications that will be used to 50,000 and the number of observations to be used in each time series to 1,000. The exclamation marks enable the scalars to be used without previously having to define them using the **scalar** instruction. Of course, these values can be changed as desired. Loops in EViews are defined as **for** at the start and **next** at the end, in a similar way to visual basic code. Thus **for !REPC=1 to !NREPS** starts the main replications loop, which will run from **1** to **NREPS**.

```
Y(1) = 0
smpl @first+1 !NOBS + 200
Y = Y(-1) + nrnd
```

The three lines above set the first observation of the series **Y** to zero (line 11) then change the sample considered sample to the observations 2 to !NOBS+200 (1200) (so **@first** is EViews method of denoting the first observation in the series, and the final observation is denoted by, you guessed it, **@last**). This enables the program to generate 200 additional startup observations. It is very easy in EViews to construct a series following a random walk process, and this is done by the third of the above lines. The current value of **Y** is set to the previous value plus a standard normal random draw (**nrnd**). In EViews, draws can be taken from a wide array of distributions (see the User Guide). In line 14 **series DY = Y - Y(−1)** creates a new series called **DY** that contains the first difference of **Y**.

```
smpl @first+200 !NOBS+200
equation eq.ls DY Y(−1)
```

The first of the two lines above sets the sample to run from observation 201 to observation 1200, thus dropping the 200 startup observations. The following line actually conducts an OLS estimation (**'.ls'**), in the process creating an equation object called **eq**. The dependent variable is **DY** and the independent variable is the lagged value of Y, **Y(−1)**.

Following the equation estimation, several new quantities will have been created. These quantities are denoted by a **'@'** in EViews. So the line **'T_NONE(!REPC)=@tstats(1)'** will take the *t*-ratio of the coefficient on the first (and in this case only) independent variable, and will place it in the **!REPC** row of the series **T_NONE**. Similarly, the *t*-ratios on the lagged value of Y will be placed in **T_CONST** and **T_TREND** for the regressions with constant and constant and trend, respectively. Finally, **next** will finish the replications loop and **smpl @first !NREPS** will set the sample to run from 1 to 50,000, and the 1%, 5%, and 10% critical values for the no constant or trend case will then be stored in the scalars **Q** with respective indices. The **'SCALAR Q_NONE1=@quantile(T_NONE,0.01)'** instruction will take the 1% quantile from the series **T_NONE**, which avoids sorting the series, and store it in the scalar **Q_NONE** . The critical values obtained by running the above instructions are summarised in Table 4.

Table 4: Simulated Critical Values for an ADF Test

	1%	5%	10%
No Constant or Trend (**NONE**)	−2.57	−1.94	−1.62
Constant but No Trend (**CONST**)	−3.42	−2.65	−2.56
Constant and Trend (**TREND**)	−3.95	−3.40	−3.12

This is to be expected, for the use of 50,000 replications should ensure that an approximation to the asymptotic behaviour is obtained. For example, the 5% critical value for a test regression with no constant or trend and 500 observations is −1.945 in this simulation, and −1.95 in Fuller (1976). Although the Dickey–Fuller simulation was unnecessary in the sense that the critical values for the resulting test statistics are already well known and documented, a very similar procedure could be adopted for a variety of problems. For example, a similar approach could be used for constructing critical values or for evaluating the performance of statistical tests in various situations.

23.2 Pricing Asian Options

Reading: Brooks (2019, Section 13.8)

A sample of EViews code for determining the value of an Asian option is given below. The example is in the context of an arithmetic Asian option on the FTSE 100, and two simulations will be undertaken with different strike prices (one that is out of the money forward and one that is in the money forward). In each case, the life of the option is six months, with daily averaging commencing immediately, and the option value is given for both calls and puts in terms of index points. The parameters are given as follows, with dividend yield and risk-free rates expressed as percentages:

Simulation 1:	Strike = 6500	**Simulation 2:**	Strike = 5500
	Spot FTSE = 6,289.70		Spot FTSE = 6,289.70
	Risk-free rate = 6.24		Risk-free rate = 6.24
	Dividend yield = 2.42		Dividend yield = 2.42
	Implied Volatility = 26.52		Implied Volatility = 34.33

Any other programming language or statistical package would be equally applicable, since all that is required is a Gaussian random number generator, the ability to store in arrays and to loop. Since no actual estimation is performed, differences between packages are likely to be negligible. All experiments are based on 25,000 replications and their antithetic variates (total: 50,000 sets of draws) to reduce Monte Carlo sampling error. Some sample code for pricing an Asian option for normally distributed errors using EViews can be found in the file **'asian_pricing.prg'** and is given in Figure 100.

Many parts of the program above use identical instructions to those given for the DF critical value simulation, and so annotation will now focus on the construction of the program and on previously unseen commands. The first block of commands set up a new workfile called **'asian_pricing'** that will hold all of the objects and output. Then the following lines specify the parameters for the simulation of the path of the price of the underlying asset (the drift, the implied volatility, etc.). **'!DT=!TTM/!N'** splits the time to maturity (0.5 years) into N discrete time periods. Since daily averaging is required, it is easiest to set N=125 (the approximate number of trading days in half a year), so that each time period **DT** represents one day. The model assumes under a risk-neutral measure that the underlying asset price follows a geometric Brownian motion, which is given by

$$dS = (\mathrm{rf} - \mathrm{dy})Sdt + \sigma dz \tag{13}$$

where dz is the increment of a Brownian motion. Further details of this continuous time representation of the movement of the underlying asset over time are beyond the scope of this guide. A treatment of this and many other useful option pricing formulae and computer code are given in Haug (1998), and an accessible discussion is given in Hull (2018). The discrete time approximation to this for a time step of one can be written as

$$S_t = S_{t-1} \exp\left[\left(\mathrm{rf} - \mathrm{dy} - \frac{1}{2}\sigma^2\right) dt + \sigma\sqrt{dt}u_t\right] \tag{14}$$

where u_t is a white noise error process. The following instructions set up the arrays for the underlying spot price (called **'SPOT'**), and for the discounted values of the put (**'APVAL'**) and call (**'ACVAL'**). Note that by default, arrays of the length given by the 'workfile' definition statement (50000) will be created.

```
Program: ASIAN_PRICING - (d:\programming guide\eviews guide\eviews_files\asian_pr...
Run | Print | Save | SaveAs | Snapshot | Cut | Copy | Paste | InsertTxt | Find | Replace | Wrap+/- | LineNum+/- | Encrypt

' New workfile created called ASIAN_PRICING, undated with 50000 observations
workfile ASIAN_PRICING U 50000
rndseed 12345
!N = 125
!TTM = 0.5
!NREPS = 50000
!IV = 0.2652
!RF = 0.0624
!DY = 0.0242
!DT = !TTM / !N
!DRIFT = (!RF-!DY-(!IV^2/2.0))*!DT
!VSQRDT = !IV*(!DT^0.5)
!K = 6500
!S0 = 6289.7
series APVAL
series ACVAL
series SPOT
series RANDS
for !REPC=1 to !NREPS step 2
   RANDS = NRND 'Generating a series of random numbers
   SPOT(1)=!S0*exp(!DRIFT+!VSQRDT*RANDS(1)) 'Calculating spot price series
   smpl 2 !N
   SPOT = SPOT(-1)*exp(!DRIFT+!VSQRDT*RANDS)
   smpl @first !N
   !AV=@MEAN(SPOT)
   ACVAL(!REPC) = (!AV>!K)*(!AV-!K)*exp(-!RF*!TTM) 'Computing call prices
   APVAL(!REPC)=(!AV<!K)*(!K-!AV)*exp(-!RF*!TTM) 'Computing put prices
   RANDS = -RANDS 'Using the negative as second observation
   SPOT(1)=!S0*exp(!DRIFT+!VSQRDT*RANDS(1))
   smpl 2 !N
   SPOT = SPOT(-1)*exp(!DRIFT+!VSQRDT*RANDS)
   smpl @first !N
   !AV=@mean(SPOT)
   ACVAL(!REPC+1)=(!AV>!K)*(!AV-!K)*exp(-!RF*!TTM) 'Computing call prices
   APVAL(!REPC+1)=(!AV<!K)*(!K-!AV)*exp(-!RF*!TTM) 'Computing put prices
NEXT
smpl @first !NREPS
scalar CALLPRICE = @mean(ACVAL)
scalar PUTPRICE=@mean(APVAL)
```

Figure 100: EViews Program Code to Price an Asian Option

The command **'for !REPC=1 to !NREPS step 2'** starts the main for loop for the simulation, looping up to the number of replications, in steps of 2. The loop ends at **NEXT**. Steps of 2 are used because antithetic variates are also used for each replication, which will create another simulated path for the underlying asset prices and option value.

The random $\mathcal{N}(0, 1)$ draws are made, which are then constructed into a series of future prices of the underlying asset for the next 125 days. **'!AV=@mean(SPOT)'** will compute the average price of the underlying over the lifetime of the option (125 days). The following two statements construct the terminal payoffs for the call and the put options, respectively. For the call, **'ACVAL'** is set to the average underlying price less the strike price if the average is greater than the strike (i.e., if the option expires in the money), and zero otherwise. Vice versa for the put. The payoff at expiry is discounted back to the present using the risk-free rate, and placed in the REPC row of the **'ACVAL'** or **'APVAL'** array for the calls and puts, respectively. This is achieved by multiplying the call (put) value with the binary expression **!AV>!K** (**!AV<!K**).

The process then repeats using the antithetic variates, constructed using **'RANDS = -RANDS'**. The call and put present values for these paths are put in the even rows of **'ACVAL'** and **'APVAL'**.

This completes one cycle of the REPC loop, which starts again with **!REPC=3**, then 5, 7, 9, ..., 49999. The result will be two arrays **'ACVAL'** and **'APVAL'**, which will contain 50,000 rows comprising the present value of the call and put option for each simulated path. The option prices would then simply be given by the averages over the 50,000 replications.

Note that both call values and put values can be calculated easily from a given simulation, since the most computationally expensive step is in deriving the path of simulated prices for the underlying asset. The results are given in Table 5, along with the values derived from an analytical approximation to the option price, derived by Levy, and estimated using VBA code in Haug (1998) (pp. 97–100).

The main difference between the way that the simulation is conducted here and the method used for EViews simulation of the Dickey–Fuller critical values is that, here, the random numbers are generated by opening a new series called **'RANDS'** and filling it with the random number draws. The reason that this must be done is so that the negatives of the elements of **RANDS** can later be taken to form the antithetic variates. Finally, for each replication, the call prices and put prices are computed. Outside the replications loop, the options prices are the averages of these discounted prices across the 50,000 replications.

The workfile **'asian_pricing'** will contain quite a few objects by the end of the simulation, including the scalars **CALLPRICE** and **PUTPRICE**, which will be the call and put prices. Also, the series **ACVAL** and **APVAL** will contain the current value of the option for each of the 50,000 simulated paths. Having the whole series across all replications can be useful for constructing standard errors, and for checking that the program appears to have been working correctly.

Table 5: Simulated Asian Option Prices

Simulation 1: Strike = 6500, IV = 26.52%		**Simulation 2:** Strike = 5500, IV = 34.33%	
Call	Price	**Call**	Price
Analytical Approximation	203.45	Analytical Approximation	888.55
Monte Carlo Normal	203.69	Monte Carlo Normal	885.11
Put	Price	**Put**	Price
Analytical Approximation	348.70	Analytical Approximation	64.52
Monte Carlo Normal	349.01	Monte Carlo Normal	61.51

In both cases, the simulated options prices are quite close to the analytical approximations, although the Monte Carlo seems to overvalue the out-of-the-money call and to undervalue the out-of-the-money put. Some of the errors in the simulated prices relative to the analytical approximation may result from the use of a discrete-time averaging process using only 125 points.

24 Value-at-Risk

24.1 Extreme Value Theory

Reading: Brooks (2019, Section 14.3)

In this section, we are interested in extreme and rare events such as stock price crashes. Therefore, we will look into the left tail of the distribution of returns. We will use the data set provided in **'sp500.wf1'** which contains daily prices of the S&P500 index from January 1950 until July 2018. First, from the price series we generate log returns by typing

ret = log(sp500) - log(sp500(–1))

into the equation window that pops up after clicking on **'Generate'**. Alternatively, we can also type the code into the command line as in Figure 101 and press **'Enter'**.

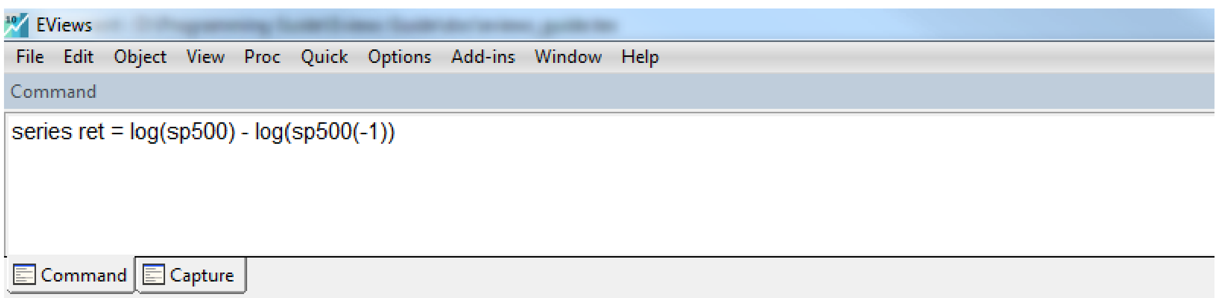


Figure 101: Using the Command Line to Generate a Series

To compute the Value-at-Risk (VaR), we only need to look at the quantiles of the distribution. However, it is important to specify which distribution we will assume for the returns. If we look at the historical distribution, we can compute the 1% quantile by typing

scalar q1 = @quantile(ret,0.01),

which will store the quantile in a new scalar named **q1**. Checking the value, we find it to be –0.026, which means that the maximum daily loss that we can expect in 100 days is 2.6%. Since this estimate is derived directly from the historical data, we will also refer to it as the historical VaR.

If we were to assume that the returns follow a normal distribution, we can compute the 1%-VaR from the mean (μ) and standard deviation (σ). The α-percentile of a normal distribution is then given as

$$VaR^{\alpha}_{\text{normal}} = \mu - \Phi^{-1}(1-\alpha)\sigma \tag{15}$$

In EViews, we can compute this by typing

scalar q1_norm = @mean(ret) - @qnorm(0.99)*@stdev(ret)

and store it in a new scalar **q1_norm**. We see that the value $VaR^{1\%}_{\text{normal}} = -0.022$ is smaller in absolute value than the one obtained form the historical distribution. Hence, assuming a normal distribution would lead to the assumption that the maximal loss to expect is only 2.2%.

24.2 The Hill Estimator for Extreme Value Distributions

In the next step, we use the tail of the distribution to estimate the VaR and we will implement the Hill estimator of the shape parameter ξ of an extreme value distribution. The necessary code can be found in the file **'hill.prg'** as presented in Figure 102.

Program: HILL - (d:\programming guide\eviews guide\eviews_files\hill.prg)

Run | Print | Save | SaveAs | Snapshot | Cut | Copy | Paste | InsertTxt | Find | Replace | Wrap+/- | LineNum+/-

```
1  'Program to compute the Hill estimator in EViews
2  load "D:\Programming Guide\Eviews Guide\eviews_files\sp500.wf1"
3  scalar alpha = 0.01
4  scalar U = -0.025
5  series y = -ret*(ret<U)
6  sort(d) y
7  !N_U = 197
8  series xi
9  series var
10 for !k=2 to !N_U
11     smpl 1 !k
12     !y_k = y(!k) 'save the k-th largest value
13     series s = log(y)-log(!y_k)
14     xi(!k) = @sum(s)/(!k-1) 'compute the hill estimator
15     var(!k) = -y(!k)*(17247*alpha/!N_U)^(-xi(!k))
16 next
```

Figure 102: Program Code to Implement the Hill Estimator

In the first lines we load the workfile **'sp500.wf1'** which we used before and define the confidence level **alpha** and the threshold **U**. In line 5 we generate the data series $\tilde{y}$, which comprises the absolute value of all returns below the threshold U. The command **sort(d) y** will sort **y** in descending order. We also save the length of this data set in the variable **!N_U=197**.

As the Hill estimator is dependent on how many of the observations below the threshold are used (k), we create two series **xi** and **var** to store the estimates ξ_k and VaR_k for all possible values of k. The code for this procedure spans from line 10 to 16, while it is specifically line 14, where we compute the hill estimator as in Brooks (2019):

$$\hat{\xi}_k = \frac{1}{k-1} \sum_{i=1}^{k-1} \log(\tilde{y}_{(i)}) - \log(\tilde{y}_{(k)}) \tag{16}$$

We compute Equation (16) in four steps. First, we set the sample to the first k observations with the command **smpl 1 !k**. Second, we save the k^{th} largest value from the **!k** observations. Remember that the series is sorted in descending order, so the k^{th} observation is also the k^{th} largest. Now we calculate the log differences of $\tilde{y}$ to the k^{th} biggest value $\tilde{y}_{(k)}$ and save it into the variable **s** (line 13). Lastly, we compute the Hill estimator in line 16 according to Equation (16). Finally, we compute the VaR following Brooks (2019, Chapter 14) as

$$\text{VaR}_{hill,k} = \tilde{y}_{(k)} \left[\frac{N\alpha}{N_U}\right]^{-\hat{\xi}_k} \tag{17}$$

If we double click the **var** variable in the resulting workfile and then click **View/Graph** without changing the default settings we will obtain the Hill plot in Figure 103. We can see that the estimated VaR is below the historical value of 2.6% for any choice of k.

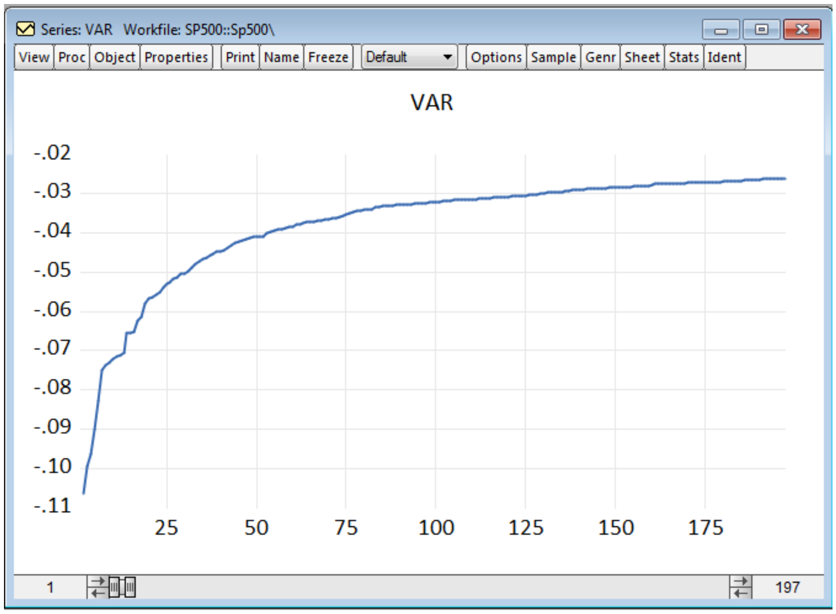


Figure 103: Hill Plot for Value-at-Risk

24.3 VaR Estimation Using Bootstrapping

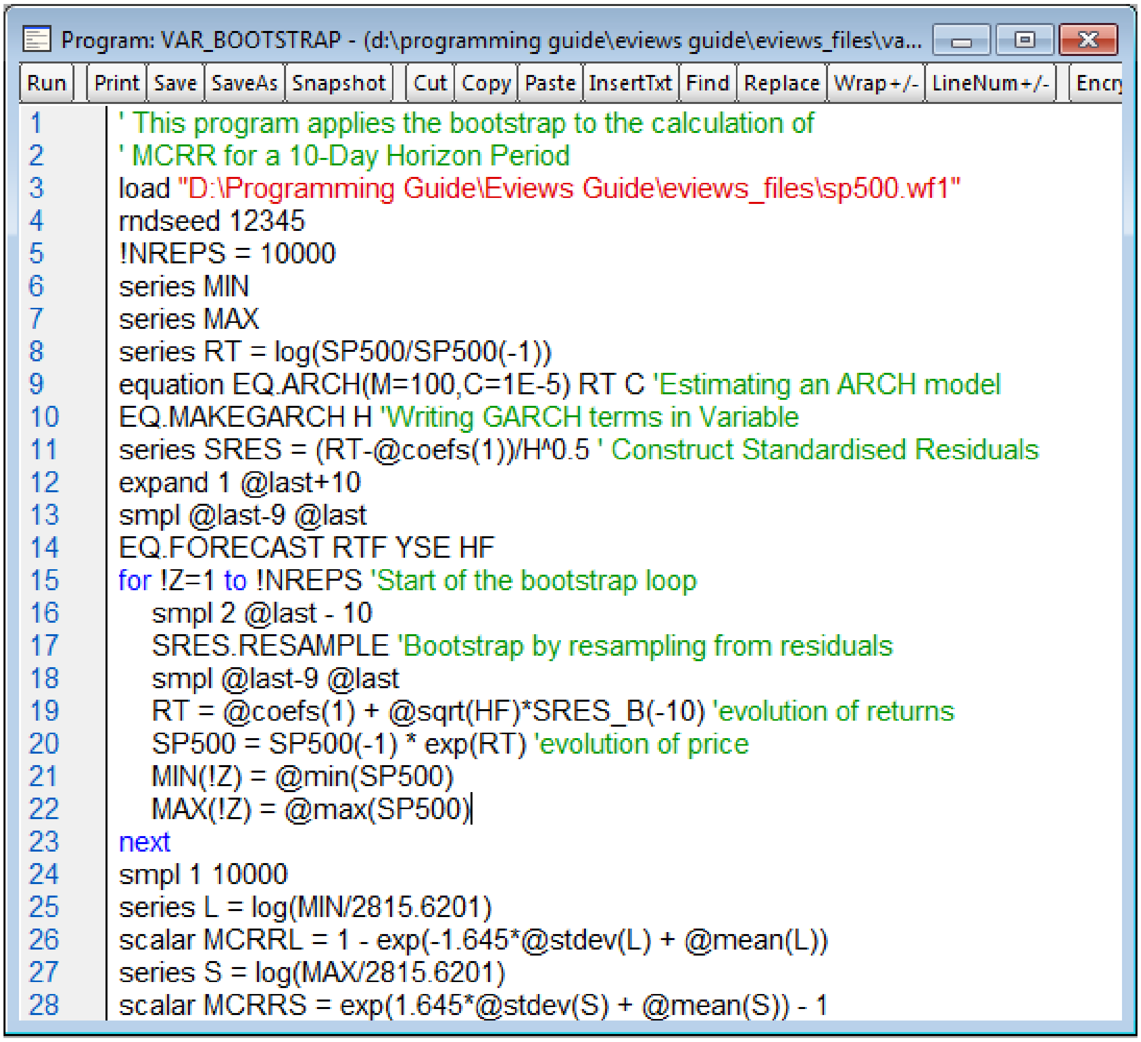
Reading: Brooks (2019, Section 13.9)

The following EViews code can be used to calculate the minimum capital risk requirement (MCRR) for a ten-day holding period (the length that regulators require banks to employ) using daily S&P500 data, which is found in the file **'sp500.wf1'**. The code is presented in Figure 104, followed by an annotated copy of some of the key lines.

Again, annotation of the code above will concentrate on commands that have not been discussed previously. The series **'MIN'** and **'MAX'** are set up to hold the minimal and maximal prices, respectively. Then **'equation EQ.ARCH(M=100,C=1E-5) RT C'** estimates an ARCH model, denoting the equation object created by **'EQ'**, and allowing the process to perform up to 100 iterations with a convergence criterion of 10^{-5}, with the dependent variable **RT** (which is the returns series) and the conditional mean equation containing a constant only. The line **'EQ.MAKEGARCH H'** will generate a series of fitted conditional variance values, denoted by H. The line **series SRES = (RT-@coefs(1))/H^0.5** will construct a set of standardised residuals.

The next step is to forecast the conditional variances for the ten observations 17248 to 17257. The **'expand 1 @last+10'** instruction will increase the size of the arrays in the workfile by 10 from the original length of the S&P500 series (17,247 observations), such that we can then set the sample to these last observation by **smpl @last-9 @last**. Using the command **'EQ.FORECAST RTF YSE HF'** we construct forecasts of the conditional mean (placed into RTF), the conditional standard deviation (YSE) and the conditional variance (HF), respectively.

Next follows the core of the program, which is the bootstrap loop. The number of replications **'!NREPS'** has been defined as 10,000. The instructions **SRES.RESAMPLE** resamples series, which is then placed in **SRES_B**. The future paths of the series over the ten-day holding period are then constructed, and the maximum and minimum price achieved over that period (observations 17248 to 17257) are saved in the arrays **MAX** and **MIN**, respectively. Finally, **next** finishes the bootstrapping loop.




```
1  ' This program applies the bootstrap to the calculation of
2  ' MCRR for a 10-Day Horizon Period
3  load "D:\Programming Guide\Eviews Guide\eviews_files\sp500.wf1"
4  rndseed 12345
5  !NREPS = 10000
6  series MIN
7  series MAX
8  series RT = log(SP500/SP500(-1))
9  equation EQ.ARCH(M=100,C=1E-5) RT C 'Estimating an ARCH model
10 EQ.MAKEGARCH H 'Writing GARCH terms in Variable
11 series SRES = (RT-@coefs(1))/H^0.5 ' Construct Standardised Residuals
12 expand 1 @last+10
13 smpl @last-9 @last
14 EQ.FORECAST RTF YSE HF
15 for !Z=1 to !NREPS 'Start of the bootstrap loop
16    smpl 2 @last - 10
17    SRES.RESAMPLE 'Bootstrap by resampling from residuals
18    smpl @last-9 @last
19    RT = @coefs(1) + @sqrt(HF)*SRES_B(-10) 'evolution of returns
20    SP500 = SP500(-1) * exp(RT) 'evolution of price
21    MIN(!Z) = @min(SP500)
22    MAX(!Z) = @max(SP500)
23 next
24 smpl 1 10000
25 series L = log(MIN/2815.6201)
26 scalar MCRRL = 1 - exp(-1.645*@stdev(L) + @mean(L))
27 series S = log(MAX/2815.6201)
28 scalar MCRRS = exp(1.645*@stdev(S) + @mean(S)) - 1
```

Figure 104: Program Code to Calculate Value-at-Risk

The following **smpl** instruction is necessary to reset the sample period used to cover only the entries from 1 to 10,000 (i.e., to incorporate all of the 10,000 bootstrap replications). By default, if this statement was not included, EViews would have continued to use the most recent sample statement, conducting analysis using only observations 17248 to 17257.

The following block of two commands (line 25 and 26) generates the MCRR for the long position. The first stage is to construct the log returns for the maximum loss over the ten-day holding period. Notice that the command will automatically do this calculation for every element of the 'MIN' array – i.e., for all 10,000 replications. In order to use information from all of the replications, and under the assumption that the **L** statistic is normally distributed across the replications, the **MCRR** can be calculated using the command given (rather than using the fifth percentile of the empirical distribution).

The following two lines then repeat the above procedure, but replacing the **'MIN'** array with **'MAX'** to calculate the **MCRR** for a short position. The results that would be generated by running the above program are approximately:

MCRRL = 0.03081874 for the long position, and

MCRRS = 0.03577174 for the short position.[20]

[20] Differences to approaches using other software can arise from the estimation of the ARCH model in line 9.

These figures represent the minimum capital risk requirement for a long and short position, respectively, as a percentage of the initial value of the position for 95% coverage over a ten-day horizon. This means that, for example, approximately 3.1% of the value of a long position held as liquid capital will be sufficient to cover losses on 95% of days if the position is held for ten days. The required capital to cover 95% of losses over a ten-day holding period for a short position in the S&P500 index would be around 3.6%. This is as one would expect since the index had a positive drift over the sample period. Therefore, the index returns are not symmetric about zero, as positive returns are slightly more likely than negative returns. Higher capital requirements are thus necessary for a short position since a loss is more likely than for a long position of the same magnitude.

25 The Fama–MacBeth Procedure

Reading: Brooks (2019, Section 14.2)

In this section we will perform the two-stage procedure by Fama and MacBeth (1973). The Fama–MacBeth procedure as well as related asset pricing tests are described in Chapter 14 of Brooks (2019). There is nothing particularly complex about the two-stage procedure – it only involves two sets of standard linear regressions. The hard part is really in collecting and organising the data. If we wished to do a more sophisticated study – for example, using a bootstrapping procedure or using the Shanken (1992) correction, this would require more analysis then is conducted in the illustration below. However, hopefully the EViews code and the explanations will be sufficient to demonstrate how to apply the procedures to any set of data.

The example employed here is taken from the study by Gregory et al. (2013) that examines the performance of several different variants of the Fama–French and Carhart models using the Fama–MacBeth methodology in the UK following several earlier studies showing that these approaches appear to work far less well for the UK than the US. The data required are provided by Gregory et al. (2013) on their web site.[21] Note that their data have been refined and further cleaned since their paper was written (i.e., the web site data are not identical to those used in the paper) and as a result the parameter estimates presented here deviate slightly from theirs. However, given that the motivation for this exercise is to demonstrate how the Fama–MacBeth approach can be used in EViews, this difference should not be consequential. The two data files used are **'monthlyfactors.csv'** and **'vw_sizebm_25groups.csv'**. The former file includes the time series of returns on all of the factors (SMB, HML, UMD, RMRF, the return on the market portfolio (RM) and the return on the risk-free asset (RF)), while the latter includes the time series of returns on 25 value-weighted portfolios formed from a large universe of stocks, two-way sorted according to their sizes and book-to-market ratios.

The first step in this analysis for conducting the Fama and French (1992) or Carhart (1997) procedures using the methodology developed by Fama and MacBeth is to create a new EViews workfile which I have called **'ff_data.wf1'** and to import the two .csv files into it. The data in both cases run from October 1980 to December 2017, making a total of 447 data points. In order to obtain results as close as possible to those of the original paper, when running the regressions, you can restrict the period to October 1980 to December 2010 (363 data points). We then need to set up a program file along the lines of those set up in the previous section – I have called mine **'fama_macbeth.prg'**. The full code to run the tests is shown in Figure 105.

We can think of this program as comprising several sections. The first step is to transform all of the raw portfolio returns into excess returns which are required to compute the betas in the first stage of Fama–MacBeth. This is fairly simple to do and writes over the original series with their excess return counterparts. We create the group **PF** containing all portfolios and then subtract the risk-free return from every variable in this group. This is done in line 5, **RF** by executing **PF{!i})=PF{!i})-RF**, within the loop over all variables in group **PF**.

[21] http://business-school.exeter.ac.uk/research/centres/xfi/famafrench/files/

Program: FAMA_MACBETH - (d:\programming guide\eviews guide\eviews_files\fama_macbeth.prg)

Run | Print | Save | SaveAs | Snapshot | Cut | Copy | Paste | InsertTxt | Find | Replace | Wrap+/- | LineNum+/- | Encrypt

```
2   load "D:\Programming Guide\Eviews Guide\eviews_files\ff_data.wf1" ' Read data from workfile
3   group PF SL S2 S3 S4 SH S2L S22 S23 S24 S2H M3L M32 M33 M34 M3H B4L B42 B43 B44 B4H BL B2 B3 B4 BH
4   for !i=1 to PF.@count ' Compute excess returns
5       PF({!i})=PF({!i}) -RF
6   next
7   !NOBS=447 'Define the number of time series observations (363 for Gregory et al.)
8   series BETA_C
9   series BETA_RMRF
10  series BETA_UMD
11  series BETA_HML
12  series BETA_SMB
13  for !i=1 to PF.@count 'Create series and run first stage time series regressions
14      equation EQ_TS.LS PF({!i}) C RMRF UMD HML SMB 'time series regression
15      BETA_C(!i)=@coefs(1)
16      BETA_RMRF(!i)=@coefs(2)
17      BETA_UMD(!i)=@coefs(3)
18      BETA_HML(!i)=@coefs(4)
19      BETA_SMB(!i)=@coefs(5)
20  next
21  for !k=1 to !nobs 'Transpose the data set into to variables for every time observation
22      series T{!k}
23      for !i=1 to PF.@count
24          T{!k}(!i)=PF({!i})(!k)
25      next
26  next
27  series LAMBDA_C
28  series LAMBDA_RMRF
29  series LAMBDA_UMD
30  series LAMBDA_HML
31  series LAMBDA_SMB
32  series RSQUARED
33  for !j = 1 to !NOBS 'Second stage cross sectional regressions
34      equation EQ_CS.LS T{!j} C BETA_RMRF BETA_UMD BETA_HML BETA_SMB 'cross-sectional regression
35      LAMBDA_C(!j)=@coefs(1)
36      LAMBDA_RMRF(!j)=@coefs(2)
37      LAMBDA_UMD(!j)=@coefs(3)
38      LAMBDA_HML(!j)=@coefs(4)
39      LAMBDA_SMB(!j)=@coefs(5)
40      RSQUARED(!j)=@r2
41  next
42  scalar LAMBDA_C_MEAN         =@mean(LAMBDA_C)*100
43  scalar LAMBDA_C_TRATIO       =@sqrt(!NOBS)*@mean(LAMBDA_C)/@stdev(LAMBDA_C)
44  scalar LAMBDA_RMRF_MEAN      =@mean(LAMBDA_RMRF)*100
45  scalar LAMBDA_RMRF_TRATIO    =@sqrt(!NOBS)*@mean(LAMBDA_RMRF)/@stdev(LAMBDA_RMRF)
46  scalar LAMBDA_UMD_MEAN       =@mean(LAMBDA_UMD)*100
47  scalar LAMBDA_UMD_TRATIO     =@sqrt(!NOBS)*@mean(LAMBDA_UMD)/@stdev(LAMBDA_UMD)
48  scalar LAMBDA_HML_MEAN       =@mean(LAMBDA_HML)*100
49  scalar LAMBDA_HML_TRATIO     =@sqrt(!NOBS)*@mean(LAMBDA_HML)/@stdev(LAMBDA_HML)
50  scalar LAMBDA_SMB_MEAN       =@mean(LAMBDA_SMB)*100
51  scalar LAMBDA_SMB_TRATIO     =@sqrt(!NOBS)*@mean(LAMBDA_SMB)/@stdev(LAMBDA_SMB)
52  scalar RSQUARED_MEAN         =@mean(RSQUARED)
```

Figure 105: Program Code to Run Fama–MacBeth Regressions

Line 7 **!NOBS=447** ensures that the whole sample period is used. Change this to **!NOBS=363** for comparability with the paper by Gregory et al. (2013). The next stage involves creating the series for the betas. These are set up as series since there will be one entry for each regression. Now, we can run the time-series regressions for each portfolio and store the betas in the appropriate series (lines 13 to 20).

The letter **i** is an index which is defined in the loop to start with a value of 1 (the statement **!i=1**) and then as each regression is run, the value of **i** is increased by 1. So, the loop starts off with **i=1** and the regression will be run with the series **T{1}** as the dependent variable. Then the intercept (i.e., the alpha) from this regression will be placed as the first entry in **BETA_C** (i.e., it will be BETA_C(1)), the parameter estimate on the RMRF term will be placed in **BETA_RMRF(1)** and so on. Then the value of **!i** will be increased by 1 to 2, and the second regression will have dependent variable **S2**. Its intercept estimate will be placed in **BETA_C(2)**, the slope estimate on **RMRF** will be placed in **BETA_RMRF(2)** and so on. This will continue until the final regression is run on the twenty-fifth series, which will be **BH**, with its intercept estimate being placed in **BETA_C(25)**. We should thus note that while these **BETA_** series were set up with the total number of observations in the workfile (i.e., they will have 447 rows), only the first twenty-five of those rows will be filled and the remainder will contain NA.

After this we need to estimate a set of cross-sectional regressions. This presents a problem because the data can only be organised in one way or the other in EViews. So the following three lines 21 to 26 rearrange the dataset. For every time observation we create a series **T{!k}** (line 22). Using a second loop we write the 25 portfolio returns into the first 25 entries of this new series by typing **T{!k}(!i)=PF({!i})(!k)**. You can see here, how i and k switch positions like rows and columns in a matrix. Hence, we obtain a set of 447 variables, **T{1}**, **T{2}**, ..., **T{447}**, each containing only 25 cross-sectional entries.

So now we have run the first step of the Fama–MacBeth methodology – we have estimated all of the betas, also known as the factor exposures. The slope parameter estimates for the regression of a given portfolio will show how sensitive the returns on that portfolio are to the factors and the intercepts will be Jensen's alpha estimates. These intercept estimates in **BETA_C** should be comparable to those in the second panel of Table 6 in Gregory et al. (2013) – their column headed '*Simple 4F*', if you used the subsample with only 363 observations. Since the parameter estimates in all of their tables are expressed as percentages, we need to multiply all of the figures given from the EViews output by 100 to make them on the same scale. If the 4-factor model is a good one, we should find that all of these alphas are statistically insignificant. We could test this individually if we wished by adding an additional line of code in the loop to save the t-ratios in the regressions (something like **BETA_T_C(!i)=@tstats(2)** should do it). It would also be possible to test the joint null hypothesis that all of the alphas are jointly zero using a test developed by Gibbons et al. (1989) – the GRS test, but this is beyond the scope of this guide.

Now we set up the risk price variables λ_C, λ_{RMRF}, λ_{UMD}, λ_{HML} and λ_{SMB}. Further, we store the R^2 in the variable **RSQUARED**. We run the second stage regression (lines 33 to 41). The second stage of Fama–MacBeth is to run a separate cross-sectional regression for each point in time. An easy way to do this is to, effectively, rearrange the data so that each column (while still in a time series workfile) is a set of cross-sectional data. So the loop over **!j** takes the observations in the twenty-five portfolios and arranges them cross-sectionally. Thus **T{1}** will contain twenty-five data points (one for each portfolio) – all the observations for the first month, October 1980; **T{2}** will contain all twenty-five observations for the portfolios in the second month, November 1980; ...; **T{447}** will contain all twenty-five portfolio observations for December 2017.

Note that the regressions are run from 1 to **!NOBS**, which was defined as 447 for all the data available up to December 2017 or 363 as in the sample of Gregory et al. (2013) to run to December 2010. Again, it is more efficient to run these in a loop (since there will be 447 (363) of them!) rather than individually. The **!j** index will loop over each of the months to produce a set of parameter estimates (lambdas) for each one, each time running a regression on the corresponding parameter estimates from the first stage.

Thus the first regression will be of **T{1}** on a constant, **BETA_RMRF**, **BETA_UMD**, **BETA_HML**, and **BETA_SMB**, with the estimates being put in new series as before. **LAMBDA_C** will contain all of the intercepts from the second stage regressions, **LAMBDA_RMRF** will contain all of the parameter estimates on the market risk premium betas and so on. We also collect the R^2 for each regression as it is of interest to examine the cross-sectional average.

The final stage is to estimate the averages and standard errors of these estimates. The mean is calculated simply using the **@mean** command, and the standard deviation is calculated using **@stdev**. So **LAMBDA_C_MEAN** will contain the mean of the cross-sectional intercept estimates, and the corresponding t-ratio will be in **LAMBDA_C_TRATIO** and so on.

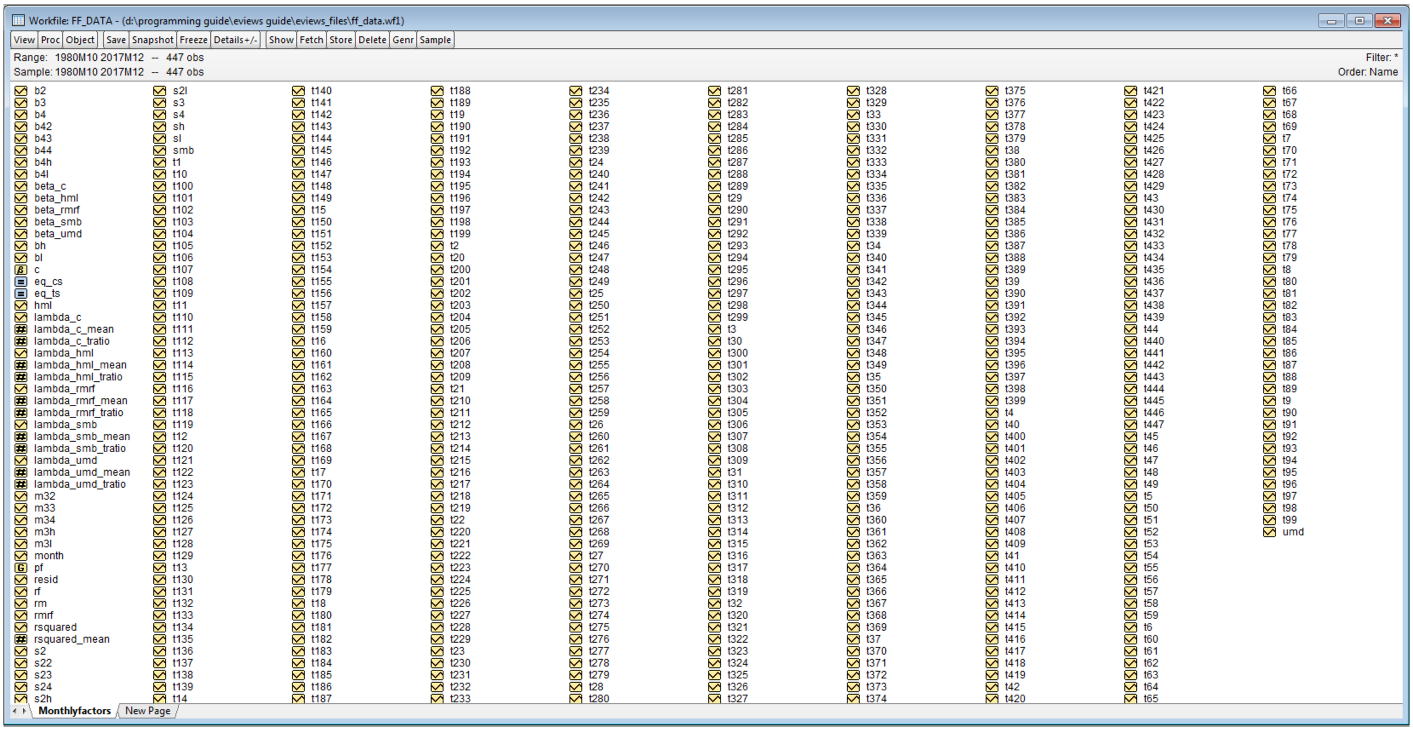


Figure 106: Workfile After Running Code for Fama–MacBeth Regressions

Once the program is run the workfile should appear as in Figure 106, we can double click on each of these objects to examine the contents. The lambda parameter estimates should be comparable with the results in the column headed '*Simple 4F Single*' from Panel A of Table 9 in Gregory et al. (2013). Note that they use γ to denote the parameters which have been called λ in this text. The parameter estimates obtained from this simulation and their corresponding t-ratios are given in Table 6. Note that the latter use the whole sample of 447 observations and do not use the Shanken correction as Gregory et al. (2013) do. These parameter estimates are the prices of risk for each of the factors (again, the coefficients from EViews need to be multiplied by 100 to turn them into percentages), and interestingly only the price of risk for value is significantly different from zero. While Gregory et al. (2013) additionally conduct a range of closely related but more sophisticated tests, their conclusion that further research is required to discover more convincing asset pricing model for the UK is identical to this one using the standard approach.

Table 6: Results from Fama–MacBeth Procedure

Parameter	Estimate	t-ratio
λ_C	0.73	2.24
λ_{HML}	0.30	1.87
λ_{RMRF}	-0.13	-0.32
λ_{SMB}	0.13	0.90
λ_{UMD}	0.02	0.03
Average R^2	0.3787	

References

Bollerslev, T. and Wooldridge, J. M. (1992). Quasi-maximum likelihood estimation and inference in dynamic models with time-varying covariances. *Econometric Reviews*, 11(2):143–172.

Brooks, C. (2019). *Introductory Econometrics for Finance*. Cambridge University Press, Cambridge, UK, 4th edition.

Carhart, M. M. (1997). On persistence in mutual fund performance. *Journal of Finance*, 52(1):57–82.

Choi, I. (2001). Unit root tests for panel data. *Journal of International Money and Finance*, 20(2):249–272.

Engle, R. F. (1982). Autoregressive conditional heteroscedasticity with estimates of the variance of united kingdom inflation. *Econometrica*, 50(4):987–1007.

Fama, E. F. and French, K. R. (1992). The cross-section of expected stock returns. *Journal of Finance*, 47(2):427–465.

Fama, E. F. and MacBeth, J. D. (1973). Risk, return, and equilibrium: Empirical tests. *Journal of Political Economy*, 81(3):607–636.

Fuller, W. A. (1976). *Introduction to Statistical Time Series*, volume 428. John Wiley and Sons.

Gibbons, M. R., Ross, S. A., and Shanken, J. (1989). A test of the efficiency of a given portfolio. *Econometrica*, 57(5):1121–1152.

Gregory, A., Tharyan, R., and Christidis, A. (2013). Constructing and testing alternative versions of the fama–french and carhart models in the uk. *Journal of Business Finance and Accounting*, 40(1-2):172–214.

Haug, E. G. (1998). *The Complete Guide to Option Pricing Formulas*. McGraw-Hill New York.

Heslop, S. and Varotto, S. (2007). Admissions of international graduate students: Art or science? a business school experience. *ICMA Centre Discussion Papers in Finance*, 8.

Hull, J. C. (2018). *Options, Futures, and Other Derivatives*. Pearson, 10th edition.

Maddala, G. S. and Wu, S. (1999). A comparative study of unit root tests with panel data and a new simple test. *Oxford Bulletin of Economics and statistics*, 61(S1):631–652.

Shanken, J. (1992). On the estimation of beta-pricing models. *Review of Financial Studies*, 5(1):1–33.