Econometric Principles and Data Analysis

Module Introduction and Overview

Contents

1 Introduction to the Module

This module provides an introduction to econometric methods. In brief, the module examines how we can start from relationships suggested by financial and economic theory, formulate those relationships in mathematical and statistical models, estimate those models using sample data, and make statements based on the parameters of the estimated models. The module examines the assumptions that are necessary for the estimators to have desirable properties, and the assumptions necessary for us to make statistical inference based on the estimated models. In addition, the module explores what happens when these assumptions are not satisfied, and what we can do in these circumstances. The module concludes with an examination of model selection.

2 The Module Authors

The module, and its more advanced sequel, *Econometric Analysis and Application*s, were designed and written by **Dr Graham Smith**, who was a Senior Lecturer in the Department of Economics, SOAS, where he taught econometrics to MSc students and carried out research on empirical finance. His main research interests focused on emerging stock markets. He has published extensively in international refereed journals. His recent research demonstrates that stock market efficiency is determined by market size, liquidity and the quality of markets.

The module has been revised by **Dr Jonathan Simms**, who is a tutor for CeFiMS, and has taught at University of Manchester, University of Durham and University of London. He has contributed to development of various CeFiMS modules including *Econometric Analysis and Applications*; *Financial Econometrics*; *Risk Management: Principles and Applications*; *Financial Engineering*; *Public Financial Management: Reporting and Audit*; *Introduction to Law and to Finance*; *Banking Strategy*; *Corporate and Investment Banking*; and *Introduction to Valuation*.

3 Study Resources

This study guide is your central learning resource; it structures your learning unit by unit. Each unit should be studied within a week. The module units are designed in the expectation that studying the unit and the associated readings in the key text, and completing the exercises, will require 15 to 20 hours during the week.

Key text

In addition to the module units you must read the assigned sections from the key text:

Jeffrey M Wooldridge (2020) *Introductory Econometrics*. 7th Edition. Boston MA: Cengage.

We have used this key text because it provides an excellent introduction to econometric theory and techniques, combining a good level of technical detail and intuitive explanation. Part 1 of the key text covers fundamental econometric methods suitable for cross-section and time series data. Part 2 examines issues that are specifically related to time series models. Part 3 considers more advanced methods. In this module you will be reading mainly from Parts 1 and 2. Wooldridge presents examples from a diverse range of topics including finance, economics and business. The examples and exercises in the module units are drawn entirely from finance. In each module unit there is a section, called Study Guide, which leads you through the relevant parts of the key text, and helps you to read and understand the analysis presented there. If, while studying this module, you find you need some revision in basic probability and statistics, you may find it useful to look at parts of Appendices A to C in the key text, which cover some of the basic mathematical tools, probability and probability distributions, and statistical inference.

Software

R

This module will use R. This is a widely used programming environment for data analysis and graphics. You will use this software to do the exercises in the units, and also the data analysis part of your assignments. The results presented in the units are also from R.

R is free software, released under the GNU General Public License. Instructions for downloading R, and a brief introduction on how to use it, are provided below.

The best advice on using R is to stay focused on the subject that is being studied in each unit, and to do the exercises for the unit; this will reinforce your understanding and also develop your confidence in using data and R.

The units include all of the R commands that are required to complete the examples and exercises. However, as you become more confident in using R, or if you are already familiar with R, you may find that you develop your own R commands.

Exercises

There are exercises in every unit. These require you to work with R and data files, available from the VLE in the module area for this study session, to do your own econometric analysis. It is very important that you attempt these exercises, and do not just look at the Answers at the end of the units. Your understanding of the material you have studied in the unit will be greatly improved if you do the exercises yourself. You will also develop better understanding and confidence in using R.

We hope that you enjoy this module.

4 Module Overview

The paragraphs following the list of topics presented in the units provide brief descriptions of the unit content. They are intended as an introduction and overview of the module. More complete, detailed explanation, analysis and discussion are provided in the units themselves, and in the module key text. So, don't worry if you do not understand everything in this short introduction.

Unit 1 An Introduction to Econometrics and Regression Analysis

1.1 What is Econometrics?
1.2 How to Use the Module Units
1.3 Ideas – The Concept of Regression
1.4 Study Guide
1.5 An Example – Efficiency in the Foreign Exchange Market
1.6 Conclusion
1.7 Working with R
1.8 Exercises
1.9 Answers to Exercises

Unit 2 The Classical Linear Regression Model

2.1 Ideas and Issues
2.2 Study Guide
2.3 Example – the Single-Index Model (SIM)
2.4 Conclusion
2.5 Exercises
2.6 Answers to Exercises
Appendix 2.1: Derivation of OLS estimators

Unit 3 The Multiple Regression Model

3.1 Ideas and Issues
3.2 Study Guide
3.3 Example – A Multi-index Model
3.4 Conclusion
3.5 Exercises
3.6 Answers to Exercises

Unit 4 Hypothesis Testing

4.1 Ideas and Issues
4.2 Study Guide
4.3 Example 1 – The Capital Asset Pricing Model
4.4 Example 2 – A Multi-index Model
4.5 Conclusion
4.6 Exercises
4.7 Answers to Exercises

Unit 5 Heteroscedasticity

5.1 Ideas and Issues
5.2 Study Guide

Unit 1 provides an introduction to econometrics and *regression analysis*. By regression we mean an equation that captures the mathematical relationship between the variables, and also the imperfect nature of that relationship. The unit introduces the stages of an econometric investigation:

- statement of the theory
- collection of data
- mathematical model of the theory (an exact relationship between variables)
- econometric model of the theory (a stochastic model of the relationship between variables)
- parameter estimation
- checking for model adequacy
- tests of hypotheses
- prediction.

Unit 1 also provides guidance on how to use the study materials. In addition, it provides a brief revision of how to calculate financial rates of return.

Each unit includes a worked example. (In Unit 1, the example concerns the relation between spot and forward exchange rates.) All of the units also contain exercises for you to do in order to develop your own understanding and confidence, from a wide range of econometric studies. Data for the exercises are provided. The data used in the examples are also provided so that you can replicate the results presented in the unit (replicating the results in the example is presented as an exercise).

Answers for the exercises are provided at the end of each unit, but you should look at the answers only after you have done the exercises yourself!

Data on the stock price of Delta Airlines Inc. and the New York Stock Exchange Composite Index are introduced in the exercises in Unit 1. This data set is used in a number of units throughout the module, in the worked examples or the exercises. By applying different econometric tools with the same data set, it is hoped you will develop a rounded view of how the methods you will learn relate to each other. A variety of other models and data sets are also used.

Unit 2 presents the *classical linear regression model*. It explains the method of 'ordinary least squares' (OLS) and how that can be used to estimate the unknown parameters of a regression equation using sample data. In this unit we are concerned with models containing two variables; we are trying to discover how one variable – the explanatory variable – explains another variable – the dependent variable – and estimate the parameters in that relationship.

We then need to ask whether we can make statements about the true, unknown, parameters of the model, based on our estimated values. To do this we need to make a number of assumptions. These assumptions, if satisfied, ensure that the estimators we use have desirable properties (in brief and oversimplified terms: the estimators are accurate and efficient). If the assumptions are satisfied, we can also make predictions about the unknown model parameters, and we can specify, precisely, how confident we are about those predictions. Unit 2 also explains goodness of fit: how closely our estimated model fits our sample data. These ideas are demonstrated using the single-index market model applied to Delta Airlines Inc., and the British retailer Marks & Spencer.

Unit 3 extends the analysis to the multiple regression model; these are regression models in which one variable is explained by two or more variables. The unit examines the assumptions necessary to estimate and make predictions with such models. The unit asks what happens if, in a multiple regression model, there is a relationship between any of the explanatory variables, in addition to the relationships we hope to discover between the explanatory variables and the dependent variable (this is called *multicollinearity*). The techniques of multiple regression are demonstrated with an example of a multi-index model.

Unit 4 explores how to test *hypotheses*. Based on our estimated model coefficients, can we answer questions of the form:

- Is the true, unknown coefficient negative, zero, or positive?
- Does it take a particular value?
- Is there actually a relationship between the variables?

Unit 3 uses the capital asset pricing model (CAPM) and a multi-index model to demonstrate hypothesis testing. So, for example, we might be concerned with how we can test whether the stock we are interested in is defensive or aggressive; is the company beta less than one or greater than one? The efficiency of foreign exchange markets is also examined.

Units 5, 6 and 7 are concerned with what happens if a number of the assumptions of the classical linear regression model are not satisfied. What are the consequences for the properties of the ordinary least squares estimators, and can we still make predictions about the unknown model parameters based on our estimated model?

Unit 5 is concerned with *heteroscedasticity*. What is that? Here is a very brief and simplified explanation; a more detailed and precise explanation is provided in Unit 5. Unit 1 explains how we can specify a mathematical relationship between variables. The actual relationship between variables is not exact, and we attempt to capture this by including an error or disturbance term in the regression equation. One of the assumptions we make is that the variance of the disturbance term – how much it varies about its mean value – is constant for all observations. This is the assumption of *homoscedasticity*, and is explained in Unit 2.

In some econometric studies this assumption may not be satisfied. Consider a cross-section study of commission rates for different brokerage companies. The disturbance term also attempts to capture those influences on commission rates that we have not included in our model. Is it likely that the variance of this disturbance term will be constant for all brokerage companies? If the variance of the disturbance term is not constant, we say there is heteroscedasticity. Unit 5 examines the consequences of heteroscedasticity:

- What are the effects on the properties of OLS estimators, and can we still make predictions based on our estimated model?

The unit examines how heteroscedasticity can be identified, and how we can deal with it, either by transforming the model or by using a different estimation method. If we know what form the heteroscedasticity takes, we can use the method of weighted least squares. Heteroscedasticity is demonstrated with a study of price-earnings ratios estimated for a cross-section of companies.

Unit 6 is concerned with *autocorrelation*. Again, here is a very simple and brief explanation; a more precise and formal explanation is provided in Unit 6. Consider again the disturbance term that we include in our regression equation. The disturbance term reflects the stochastic nature of the relationship between variables, and also attempts to capture the elements

that we have not included in the model. Another assumption we make about the disturbance term is that the disturbance terms for different observations (*eg* if using annual data, last year and this year, or if using daily data, yesterday and today) are not related.

This is the assumption of *noncorrelated disturbances*, and is explained in Unit 2. If the disturbances for different observations are related, we say that the disturbance term is serially correlated or 'autocorrelated'. For example, an economic or financial shock in one month may have persistent effects in following months, and if the model does not explicitly include such persistence effects, the disturbance terms in different months will be correlated. Unit 6 examines the implications of autocorrelation for the properties of OLS estimators, and also the consequences for prediction based on OLS estimators. It also shows how to identify autocorrelation using plots and more formal tests, and what can be done to take account of autocorrelation, including changing the method of estimation. The effects of autocorrelated disturbances are demonstrated with the single-index market model for Delta Airlines, and a model of spot and forward exchange rates.

Unit 7 is concerned with the assumption of normality. In order to make predictions about the true, unknown model parameters, based on our estimated values, we need to assume that the disturbance terms are distributed normally – that is, they follow a normal distribution. You are probably already familiar with the normal distribution from your other studies. It is a probability distribution with known properties, which allows us to make statements concerning the unknown model parameters with a particular degree of confidence – for example, we can reject a hypothesis about a parameter with a 5 per cent chance of being wrong, or we can be 95 per cent confident that an unknown parameter takes a value within a certain range of values.

If the disturbance terms are not normally distributed, we are unable to make such predictions, and it also has consequences for the properties of the OLS estimators. Unit 7 explains the effects of having disturbances that are not distributed normally, the tests to detect *nonnormal disturbances*, and what can be done about nonnormal disturbances. This includes the use of *dummy variables* to take account of outliers (data points which are very different from the rest of the sample). These methods are demonstrated with two examples: stock market returns and the single-index model for Marks & Spencer. The exercises include consideration of the SIM for Delta Airlines and for Bank of America.

Unit 8 is concerned with *model selection*. One of the assumptions we make is that the model we estimate is correctly specified: the regression equation includes all relevant variables, and the functional form of the relationship is specified correctly – variables are included correctly as levels, or their logged values are included, or perhaps squared values of the variables are included. If the model is not correctly specified, this has consequences for

the properties of the OLS estimators and for prediction based on those estimators. In particular, Unit 8 examines the consequences of omitting a relevant explanatory variable, including an irrelevant explanatory variable, and using the wrong functional form.

The unit explains methods to identify misspecified equations. These include tests specifically designed to identify misspecified models. In addition, evidence of heteroscedasticity, autocorrelated errors, or nonnormal errors, may be a further sign that a model is not correctly specified. Unit 8 also shows how we can decide between different specifications of a particular economic relationship. It demonstrates model selection using the Delta Airlines data set, and also the SIM for IBM stock. Finally, Unit 8 includes a summary of the module, to help with your revision for the final examination.

More advanced topics in econometrics are studied in the module *Econometric Analysis & Applications*. These include more use of dummy variables; dynamic models: lags and expectations; simultaneous equation models; time series analysis: stationarity and nonstationarity; and forecasting.

5 Learning Outcomes

After studying this module you will be able to:

- explain the principles of regression analysis
- discuss the assumptions of the classical normal linear regression model
- explain the method of ordinary least squares
- produce and interpret plots of data
- estimate a regression equation, and interpret the results, for bivariate (two-variable) regression models and multiple regression models
- assess the consequences of multicollinearity
- test hypotheses concerning model parameters
- discuss the consequences of heteroscedasticity for the properties of OLS estimators
- assess the methods used to identify heteroscedasticity, and the various techniques to deal with heteroscedasticity
- discuss the consequences of autocorrelated disturbances for the properties of OLS estimators
- outline and discuss the methods used to identify autocorrelated disturbances, and what can be done about it
- assess the consequences of disturbance terms not being normally distributed, tests for nonnormal disturbances, and methods to deal with nonnormal disturbances
- examine the consequences of specifying equations incorrectly
- discuss the tests used to identify correct model specification, and statistical criteria for choosing between models.

6 R

> R is an implementation of the object-oriented mathematical programming language S. It is developed by statisticians around the world and is free software, released under the GNU General Public License. Syntactically and functionally it is very similar (if not identical) to S+, the popular statistics package.
>
> R is much more flexible than most software used by econometricians because it is a modern mathematical programming language, not just a program that does regressions and tests. The S language is the de facto standard for statistical science. Since most users have a statistical background, the jargon used by R experts sometimes differs from what an econometrician (especially a beginning econometrician) may expect. Code written for R can be run on many computational platforms with or without a graphical user interface, and R comes standard with some of the most flexible and powerful graphics routines available anywhere. And of course, R is completely free for any use.
>
> (extracted from Grant V. Farnsworth , Econometrics in R, 19 March 2005)

Because the R software is a programming language and not just an econometrics program, some of the functions we will be interested in are available through libraries (sometimes called packages) obtained from the R website http://www.r-project.org/.

The data files required are available on the Virtual Learning Environment (VLE).

Starting up

The R software must be installed on your system. If it is not, follow the installation instructions appropriate to the operating system (OS). Installation is especially straightforward for Windows users.

To install R

Go to the website of R http://www.r-project.org/.

There, choose your preferred CRAN mirror (*eg* http://cran.ma.imperial.ac.uk/) and click on the link referring to your OS in the box "Download and Install R". Note that these units are written with the version "Windows", but Mac OS and Linux versions are also downloadable. (CRAN mirrors are provided in locations around the world to reduce download time and internet traffic.)

On the next page, click on "base" and you will be redirected to a page where there will be a link to download the latest version of R. At the top will be written something like "Download R for Windows". Click on this link. Updated versions of R are made available over time, but the commands explained in the module files should be similar.

Save the file "R-3.6.0-win.exe" (or whatever is the latest version of the file) in any location, for example, on your Desktop. Double click on this file to start the installation, and follow the instructions. You will be asked to choose the place

where you want the file to be, and you can also choose to have a shortcut to R in the Start Menu and a Desktop icon. Choose the standard installation.

To get started with R

To open R, double click on the desktop icon or on the shortcut. The Command Window opens.

To use R, you can either type a command in the Command Window, or run a pre-written program.

1. To type a command in the command window

   You just type it on the line starting with **>** and press the key Enter on your keyboard. Note that R is case sensitive. In the unit file, if it says "type in" or "can be performed with", this means "Type in the following command in the command window and then press Enter".

   The units include all of the R commands required to work on the examples and exercises. To save time, you can copy and paste the required coding from the units into R. Note that if multiple lines of code are pasted into R in one go, they will be executed in sequence immediately, without the need to press Enter.

2. Every time you see the sign **>**, it means that R is ready for a new command. When you do not see it or when you see the sign **+**, it means that R is still working on a previous command or is stuck because your command has not been properly written (for example, a bracket may be missing at the end of the previous command). In this case, press the red button STOP in the toolbar to stop the process.

3. To run a pre-written program

   In the Menu bar go to File, then choose "New script". A new Editor Script Window opens.

   You can type all commands you wish in this file, one command per line. To run them on R, select with the mouse the commands you wish to run, and then simultaneously press the keys CTRL and the key R (CTRL + R) (or alternatively go to Edit in the Menu bar and choose "Run line or selection").

   Save your program by choosing "Save as" in File in the Menu bar (the Menu bar specific to the Editor, that is to say, the one that is available when you are in the Editor Window, and not in the Command Window). All script files have the extension ".R". Then you can close it and open it from the File menu in the Menu bar.

   In your program file you can write things that are not commands (for example, notes or explanations); but to indicate to R that these are not commands, put the symbol # in front of each non-command line.

To get help at any time, go to Help in the Menu bar. Alternatively, there are several methods of obtaining help in R. You may type in alternatively the following commands:

```
?qt
help(qt)
help.start()
help.search("covariance")
```

The last command is for a search on the term covariance, for example. Please note the use of the quotation marks " ". If you choose to develop R commands in Word, for example, and then copy the commands to R, please make sure you use the " " symbols, and not the Word symbols “ ”.

Preceding a command with a question mark, or writing a command as an argument in help(), gives a description of its usage and functionality. The help.start() function brings up a menu of help options, and help.search() searches the help files for the word or phrase given as an argument. Many times, though, the best help available can be found by a search online. The help tools above only search through the R functions that belong to packages already installed on your computer. However, often users have the following type of question: “Does R have a function to do …”. Users do not know if functionality exists because the corresponding package is not installed on their computer, and therefore resort to the R website. To search the R website for functions and references, use:

```
RSiteSearch("scatter plot")
```

This command example searches the R website for ‘scatter plot’ functions.

The results from the search should appear in your web browser.

By default, R has a couple of excellent free manuals in PDF format. If you are not already familiar with using R you are advised to read *An Introduction to R*. To access the manual, click Help | Manuals and the list of available documents will be shown.

To quit R, in the Menu bar go to File, then choose “Exit”. You are asked “Save workspace image”? Click on No.

In case R is installed in another language, to change the language of the Menu bar into English, in the Menu bar choose Edit (second from the left in the Menu bar) and then choose the last option “Preferences”. A new window opens. In the top right of it, there is an empty box for “Language for menus and messages”. Type in this box “English” and click OK. You will have to exit R and start it again.

You may want to specify a working directory for your R session, so that R chooses data files (and scripts) directly from this directory; then there is no need to specify the full path to access the files. The command getwd() returns an absolute filename representing the current working directory of the R process. The command setwd() helps you specify the working directory. For example, to set the working directory to “C:\Documents and Settings\my files\R SOAS”, you would type in:

```
setwd("C:/Documents and Settings/my files/R SOAS")
```

This avoids the need to put the command scripts and the data files within the R directory on your computer. **Note that, in this command, the slash / is used**.

Alternatively, you can set the working directory using File (on the Menu bar), then Change dir… and then browse through the folders on your device to select the folder you want to use as the working directory for that session.

To save your work, go to File | Save Workspace… Provide a filename for the workspace, and it will be saved with the extension .RData. The file will be saved in your current working directory, but you can browse to another location if you prefer.

A saved workspace can be loaded with File | Load Workspace… R will display files with the .Rdata extension in the current working directory, but again, you can browse to other locations.

Every time you start R, before running any command or program, it is best to clean up any object that may remain in memory.

During an R session, objects are created and stored by name. The command objects(), or the command ls(), can be used to display the names of the objects which are currently stored within R. The command rm() removes a data object whose name should be given as an argument.

To remove all objects, type in:

```
rm(list=ls())
```

Alternatively, use Misc | Remove all objects.

Removing all objects is useful if, for example, you work on one exercise and dataset, then you want to work on another data set and exercise in the same R session. If you do not remove the first set of objects before starting on the second question, your workspace will contain objects you created for both questions, which could be confusing. Remember to save a workspace before removing all the objects!

Installing and loading packages

The basic setup for R provides considerable statistical and mathematical functionality.

However, you will also need to use some of the more specialised econometric functions that have been written for R, and made available in packages. The module units will indicate when you need to use a particular function, and the unit will also indicate which package is required.

To use a function from a package, the relevant package must be installed on your machine. To install a package, go to Packages | Install package(s)… You will be asked to select a CRAN mirror for this session. Once the mirror has been selected, you can choose from the list of available packages to install. Some of the packages (for example, dynlm) take a few minutes to install, but installation of a package is a one-off procedure, and once the package is installed, you will not need to install it again.

To use a package in an R session, *the package must be loaded in that session*. Loading a package takes considerably less time than installing it. To load a package, go to Packages | Load package… You will see a list of packages that you have previously installed, and you can select the package you want to load.

To summarise, to access a package it must first be installed on your device. And to use the package in an R session, the package must be loaded in that session.

If you try to use a function but you have not loaded the relevant package in your R session, you will get an error message saying R could not find the function.

Reading data from text files and creating zoo objects

The data for the examples and exercises in the module units is provided in tab-delimited text files. R can read data from files in other formats, but tab-delimited text files are very simple, they can be created and opened in many applications, and they are robust to updates.

To work on the data in R, it must first be read from the text file into what is called a data frame, using a read.table command. A data frame is a matrix or table that contains the data in R. Once the data is in a data frame, it is then possible to use R to perform various operations, such as plotting the data, and estimating models.

However, many models in finance specifically focus on the time element of the observations. Furthermore, financial data can have irregular dates. For example, if you are using daily data, there may be no observations for week-ends and bank holidays. For these reasons we have used 'zoo' objects to handle the data in R, rather than data tables or regularly dated time series objects. The zoo package provides an infrastructure for regular and irregular time series (Z's Ordered Observations – hence 'zoo'), and is maintained by Achim Zeileis (Zeileis and Grothendieck, 2005). The zoo objects are indexed by the dates of the observations, which allows us to perform operations that specifically relate to the timing of the data.

The commands for reading the data from the text files and creating the zoo objects are provided in the units.

R outputs

The outputs that you obtain can be viewed in R, and can also be copied to word processing applications (for your assignments). The tables of outputs in the module units were obtained in R and then produced with the star-gazer package (Hlavac, 2018). Advice on how to use stargazer is also provided in the units.