ECON 4160: Econometrics–Modelling and Systems Estimation:

Computer Class

Ragnar Nymoen, August 29th, 2011

Department of Economics, University of Oslo

Revised by André K. Anundsen, 17th August, 2012
Practical information

Who am I? → André K. Anundsen (PhD-student)
Email: a.k.anundsen@econ.uio.no
Office: 1143
Responsible for the CPU class + the seminar series
Aims and purpose I

- Use computer program to learn about:
  1. Econometric theory
     - Complementary and supplementary to the lectures
  2. Applied econometrics
     - Integrated with the seminar series
Aims and purpose II

- In the CPU class, we will demonstrate the implications of the theoretical results established in the lectures through Monte Carlo simulations, and we will apply the methods you learn to real-world data!

- To each seminar, you will be given a set of exercises. The exercises are of two types:
  - Applied modeling tasks, where you use your (theoretical) econometric skills in combination with the skills you acquire in the computer class to analyze real-world data. You will then hold a live computer presentation of your solution proposal at the seminar
  - The other type of seminar exercises will be more theoretical and algebraic, establishing results that are extremely important to have in mind when analyzing a data set
As noted by Greene, many powerful program packages and program languages are in use in econometric research and in applied projects:

- EVViews
- Gauss
- LIMDEP
- MATLAB
- NLOGIT (LIMDEP)
- RATS (including CATS in RATS)
- SAS
- Shazam
- Stata
- TSP
MicroFit

OxMetrics, which includes PcGive

ForecastPro

Troll

Time Series Modelling (TSM)

and probably many more
The above programs are licensed commercial products, though developed from research projects...

Also exist “freeware” and “open source” programs:

- $R$
- $Gretl$
- $Scilab$
How do I choose which one to use? I

- Ease and accuracy of data loading, data storage and result reporting
  - The data is often your most valuable asset, so:
    - Getting the data into the program is an essential step!
    - Basic reporting is very similar between programs, but not standard! Reporting for typesetting (e.g., \LaTeX) is still rare. PcGive has some, Stata is better
How do I choose which one to use? II

- Menus, batch language and programming capability
  - Menus are good for getting started, and for demonstrations
  - Batch language is important for
    1. Efficient work (once you become an “expert”)
    2. Documentation (colleagues, yourself and journals!)
    3. Communication (e.g., between supervisor and yourself)

- Programming capability can be important to increase flexibility (can’t always do “everything” using menus and batch language)
How do I choose which one to use? III

- The purpose of your project!
  - Econometric programs are like (specialized) tools: They are designed to do specific tasks efficiently
  - Each program has its strengths and weaknesses
  - Unless you are very specialized yourself, you will probably end up using more than one program
Why do we use OxMetrics/PcGive in this course?

- OxMetrics is a powerful package including an option to estimate all models we consider in this course (particularly good at simultaneous equation systems! Give = Generalized Instrumental Variable Estimation)
  - and... the names of the models in the program are close to the names you will see in Erik’s notes
- Fairly easy to do Monte Carlo simulations demonstrating how e.g. heteroskedasticity, autocorrelation and endogeneity affects estimated coefficients when we use OLS
- Finally, it is relatively easy to use and provides a lot of output that are essential to any econometric analysis!
When you’re unsure about something (or just want to explore the options) in PcGive, go here:

http://pcgive.com/pcgive/index.html

Or use the help menu in the program (we’ll see later)
Topics for first computer class

- Loading the data into the program and get to “know” the data (always start by looking at graphs!)
- Variable transformations
  - *Note 1 to computer class: Use of the natural logarithm*
- Simple regression and mis-specification testing
  - *Note 2 to Computer class: Standard mis-specification tests*
- Regression and mis-specification testing with the use of the batch language
- Stability of regression models
- At last (if time) you will have the opportunity to use your newly acquired skills to look at US housing price data from 1890 to today!
Regression with experimental and non-experimental data

Consider the modelling task with experimental/lab data:

\[ Y_i = g(X_i) + v_i \]

and compare with the situation with non-experimental, real-world data:

\[ Y_i = f(X_i) + \varepsilon_i \]
Clearly, we know much less about the match between $f(X_i)$ and $Y_i$ in the non-experimental case: We simply can’t control the input and then study the output. Often, we may not even know all variables that are included in the vector $X_i$! Thus, our choice of functional form, $f(\cdot)$, and explanatory variables, $X_i$, will be reflected in the remainder $\varepsilon_i$:

$$\varepsilon_i = Y_i - f(X_i)$$

(3)

However, we will follow custom and refer to $\varepsilon_i$ as the disturbance and the estimated counterpart $\hat{\varepsilon}_i$ as the residual.
We will then need to keep in mind that the assumptions that we make about the disturbances, e.g., the "classical assumptions" are only tentative, and that we need to test that they are valid after estimation.

This is called residual mis-specification testing.
Residual mis-specification overview

<table>
<thead>
<tr>
<th>$X_i$</th>
<th>Disturbances $\varepsilon_i$ are:</th>
<th>heteroscedastic</th>
<th>autocorrelated</th>
</tr>
</thead>
<tbody>
<tr>
<td>exogenous</td>
<td>$\hat{\beta}_1$</td>
<td>unbiased</td>
<td>consistent</td>
</tr>
<tr>
<td></td>
<td>$\hat{\beta}_1$</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$\hat{\beta}_1$</td>
<td>unbiased</td>
<td>consistent</td>
</tr>
<tr>
<td></td>
<td>$\hat{\beta}_1$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>predetermined</td>
<td>unbiased</td>
<td>consistent</td>
<td>consistent</td>
</tr>
<tr>
<td></td>
<td>$\hat{\beta}_1$</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>unbiased</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>biased</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$\hat{\beta}_1$</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$\hat{\beta}_1$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Disclaimer: This material is from the Econometrics course at the Department of Economics, University of Oslo. It is designed for educational purposes and may not reflect the official course content or assignments. Use it at your own risk.
Test battery in PcGive I: Non-normality

**Normality**, Jarque and Bera (1980): (Note: Small sample correction in Give)

Test the joint hypothesis of no skewness and no excess kurtosis (3rd and 4th moment of the normal distr.):

\[ JB = \frac{n}{6} \left( \text{Skewness}^2 + \frac{1}{4} \text{Excess kurtosis}^2 \right) \] (4)

The sample skewness and excess kurtosis (skewness = 0 and kurtosis = 3 for normal distr. is defined as follows):

\[
\text{Skewness} = \frac{\frac{1}{n} \sum_{i=1}^{n} (\hat{\varepsilon}_i - \bar{\hat{\varepsilon}})^3}{\left( \frac{1}{n} \sum_{i=1}^{n} (\hat{\varepsilon}_i - \bar{\hat{\varepsilon}})^2 \right)^{\frac{3}{2}}} = \frac{\frac{1}{n} \sum_{i=1}^{n} \hat{\varepsilon}_i^3}{\left( \frac{1}{n} \sum_{i=1}^{n} \hat{\varepsilon}_i^2 \right)^{\frac{3}{2}}} \] (5)

\[
\text{Excess kurtosis} = \frac{\frac{1}{n} \sum_{i=1}^{n} (\hat{\varepsilon}_i - \bar{\hat{\varepsilon}})^4}{\left( \frac{1}{n} \sum_{i=1}^{n} (\hat{\varepsilon}_i - \bar{\hat{\varepsilon}})^2 \right)^{\frac{4}{2}}} = \frac{\frac{1}{n} \sum_{i=1}^{n} \hat{\varepsilon}_i^4}{\left( \frac{1}{n} \sum_{i=1}^{n} \hat{\varepsilon}_i^2 \right)^{\frac{4}{2}}} \] (6)

Null is normality.
Test battery in PcGive II: Heteroskedasticity

**Heteroskedasticity**, White (1980):

Auxiliary regression:

\[ \hat{\varepsilon}_i^2 = \beta_0 + \beta_1 X_i + \beta_2 X_i^2 + u_i \]  

Test that \( \beta_1 = \beta_2 = 0 \) against non-zero using an F-test.

**Hetero X** (two or more regressors), White (1980)

Auxiliary regression:

\[ \hat{\varepsilon}_i^2 = \beta_0 + \beta_1 X_i + \beta_2 Z_i + \beta_3 X_i Z_i + \beta_4 X_i^2 + \beta_5 Z_i^2 + u_i \]  

Test that \( \beta_1 = \beta_2 = \beta_3 = \beta_4 = \beta_5 = 0 \) against non-zero using an F-test.

In both cases the null is homoskedasticity.
Test battery in PcGive III: Autocorrelation

**Autocorrelation**, Godfrey (1978):
Auxiliary regression (note, $i$ is now time unit! Call it $t$):

$$
\hat{\epsilon}_t = \beta_0 + \sum_{i=1}^{p} \beta_i \hat{\epsilon}_{t-i} + \beta_p X_t + u_t
$$

(9)

A test for $p^{th}$ order autocorrelation is then to test that
$\beta_1 = \beta_2 = \cdots = \beta_p = 0$ against non-zero using an F-test.
Null is no autocorrelation.
Test battery in PcGive IV: Autoregressive conditional heteroskedasticity

ARCH, Engle (1982):
Auxiliary regression:

\[ \hat{\varepsilon}_t = \beta_0 + \sum_{i=1}^{p} \beta_i \hat{\varepsilon}_{t-i}^2 + u_t \]  

(10)

A test for \( p^{th} \) order ARCH is then to test that \( \beta_1 = \beta_2 = \cdots = \beta_p = 0 \) against non-zero using an F-test. Null is no ARCH.
Test battery in PcGive V: Regression Specification Test

RESET, Ramsey (1969):
Auxiliary regression (Note 2,3 in PcGive means squares and cubes!):

\[ \hat{\varepsilon}_t = \beta_0 + \beta_1 X_t + \beta_2 \hat{Y}_t + \beta_3 \hat{Y}_t^2 + u_t \]  

(11)

A test for \( p^{th} \) order ARCH is then to test that \( \beta_1 = \beta_2 = 0 \) against non-zero using an F-test.
Null is no specification error.