E 4160: Econometrics–Modelling and Systems
Estimation
Computer Class
Ragnar Nymoen

Department of Economics, University of Oslo

29 August 2011
Aims and purpose

- Use computer to learn about
  - econometric theory and
  - applied econometrics
- The first aspect is complementary and supplementary to the lectures in E4160
- The second is integrated with the seminars
  - We will work together in the computer class
  - To each seminar, some of the exercises will be applied modelling tasks, and you will hold a computer presentation of your solution proposal at the seminar
  - The other type of seminar exercise will be more theoretical and algebraic
As noted by Greene, many powerful program packages and program languages are in use in econometric research and in applied projects:

- EViews
- Gauss
- LIMDEP
- MATLAB
- NLOGIT (LIMDEP)
- RATS
- SAS
- Shazam
- Stata
- TSP
Introduction

Econometric software menu

Review of elementary econometrics using OxMetrics

- MicroFit
- **OxMetrics** which includes PcGive
- ForecastPro
- Troll

and many more (I am sure)

The above are licensed commercial products, though developed from research projects...

New development, free econometric software:

- **R**
- **Gret**
- **Scilab**
What to look for

- 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

- Menus and batch language and programming capability
  - Menus are good for getting started, and for demonstrations
  - Batch language is important for efficient work (once you are "expert")
  - and for documentation
  - and communication (e.g., between supervisor and yourself)
What to look for II

- Econometric programs are like (specialized) tools: They are designed to do specific tasks effectively.
- So unless you are very specialized yourself, you will probably end up using more than one program
Topics for first computer class

- Loading data and exploring data properties in OxMetrics
  - Note 1 to computer class: Use of the natural logarithm
- Simple regression and mis-specification testing
  - Note 2 to Computer class: Standard misspecification tests.
- Regression and mis-specification testing with the use of the batch language
- Stability of regression models
Regression with experimental and non-experimental data

Consider the modelling task with experimental/lab data:

\[ Y_i = g(X_i) + \nu_i. \]  \hspace{1cm} (1)

result \hspace{0.5cm} input \hspace{0.5cm} shock

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

\[ Y_i = f(X_i) + \varepsilon_i \]  \hspace{1cm} (2)

observed \hspace{0.5cm} explained \hspace{0.5cm} remainder

Clearly, we know much less about the match between \( f(x_i) \) and \( Y_i \) in this case, so everything that we “do” with \( Y_i \) and \( f(X_i) \) is reflected in the remainder \( \varepsilon_i \):

\[ \varepsilon_i = Y_i - f(X_i) \]  \hspace{1cm} (3)
Regression with experimental and non-experimental data II

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

Disturbances $\varepsilon_i$ are:

<table>
<thead>
<tr>
<th>$X_i$</th>
<th>heteroscedastic</th>
<th>$\hat{\beta}_1$</th>
<th>$\text{Var}(\hat{\beta}_1)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>exogenous</td>
<td>unbiased</td>
<td>consistent</td>
<td>wrong</td>
</tr>
<tr>
<td>predetermined</td>
<td>unbiased</td>
<td>consistent</td>
<td>wrong</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>