

Practical Econometrics II/II

Andrius Buteikis, andrius.buteikis@mif.vu.lt
<http://web.vu.lt/mif/a.buteikis/>

Course Overview

Location & website

The course is taught at Vilnius university Faculty of Mathematics and Informatics.

All the announcements, grading, lectures, examples & tasks will be available at [the course website](#).

Aim of the course

This course provides the foundations of the theory and methods used in analyzing, modelling and forecasting time series data in econometrics.

Course Outline

Below are the main topics that will be covered in this course.

1. Statistical data and their models
2. Stationary time series: WN process, AR, MA and ARCH models
3. Time series with trend and seasonality components
4. Time series with unit root
5. Regressions with time lags
6. Regressions with time series variables
7. Multivariate models: Granger causality, VAR and VECM models
8. Endogeneity problem
9. Simultaneous equations
10. Panel data models

Grade assessment

The final grade (100 Points) will be comprised of the following parts:

1. Midterm 1 = 30 % (with Python, duration ~ 3 hours)
2. Midterm 2 = 30 % (with R, duration ~ 3 hours)
3. Final Exam = 40 % (most likely with R, duration ~ 3 - 4 hours)

You need to have a total of at least ~45 points to pass (equivalent to a grade of $4.5 \approx 5$).

Each of these will be comprised of a number of tasks and will be carried out primarily using statistical software. In addition you will be required to specify a matrix, formula, hypothesis and so on in a mathematical notation via LaTeX and provide explanations.

Nevertheless, during each of the midterms and the final exam, at least one task will require R and at least one task will require Python, **so make sure that you have installed the required software beforehand!**

The total installation process can take between 20 minutes to over hour, depending on your PC specifications.

Midterm Dates (preliminary)

- ▶ Midterm 1: either 2019-03-12, or 2019-03-19
- ▶ Midterm 2: around 2019-04-23

While it should already be obvious:

- ▶ You are responsible to making sure that the software on your PC is working.
- ▶ **Do NOT install** any new versions of R and/or Python a few days before the examinations.
- ▶ **Do NOT update** any packages/libraries a few days before the examinations.
- ▶ If you have any problems with the software - ask for help during the lectures, i.e. **before** the examination dates.

Midterm and Exam Submission

Will be carried out primarily using statistical software. Any resulting models or transformations will need to be provided (if required by the exam tasks) along with any other results in a *report-style* document.

File types that should be submitted (for each task separately):

- ▶ *.ipynb (or, *.Rmd) of your code and explanations;
- ▶ *.html file of the **compiled code** (i.e. do not simply rename your raw code file to an .html file as it will not work).

Submit the completed tasks to: andrius.buteikis@mif.vu.lt

Email subject will depend of the exam:

- ▶ [PE_2] Midterm_1
- ▶ [PE_2] Midterm_2
- ▶ [PE_2] Final_Exam

Submissions should be made from your university email!

If a task (or any methods used within) requires random number generation, make sure to include a seed, so the results can be reproduced:

► **R:**

```
set.seed(student_code)
```

► **Python:**

```
import numpy as np  
#  
np.random.seed(student_code)
```

where `student_code` is your unique student code.

Preparation

During the midterm and final exams (which are **open book**) you can use the following:

- ▶ Any notes and literature - both online and offline;
- ▶ **Formula templates** - sometimes a task will require you to write down an equation or a matrix notation - having formula templates will save you some time;
- ▶ **Code templates or examples** - from existing notes, prepared, etc.;

In practice, an empirical application does not consist of just the code - you need to be able to understand the results - whether they **do** or **do not** support any preliminary assumptions about the data and variables of interest.

Grading

During the midterms and the final exam the following should be included in your submitted files:

- ▶ **Your code** used for the solutions to the tasks and their parts;
- ▶ **The output from your code in .html format**, which provides answers to the questions of the tasks. **Compiling the code to an .html format is the basic requirement to show that your code is working. Failure to provide the CORRECTLY COMPILED file will automatically result in a 0.**
- ▶ **Comment lines inside the code** - are generally ignored when grading, i.e. comments like the following

```
x = seq(1:10)
# This is a comment
y = x^2
# Another comment about the result
```

are skipped.

- ▶ **Your clear explanations/formulas on the results from your code;** Examine the template files on the course website. **Your explanations and formulas need to be provided as formatted text, NOT comments in the code. Failure to do so, or not providing explanations will reduce the points for the task by 80%.**

The last part is especially necessary - if no commentary is provided, no conclusions can be drawn, whether the task was understood, or simply copy-pasted from somewhere. Likewise if the code does not match the comments and written formulas.

Since this is the second semester of the Practical Econometrics course, you are already familiar with R, Python, along with formatting formulas and inserting text blocks alongside the code in JupyterLab.

Extra Task

An extra task worth 5 points (roughly 0.5 to final grade) will be specified in 1 - 2 weeks time.

Software

Some useful links:

- ▶ [R Cheatsheets](#) - useful summaries for various tasks, like data transformation, R Markdown, Data Visualization, etc.
- ▶ ['ctv' package](#) - a useful package for downloading and updating many R packages at once (e.g. packages covering *Econometrics*).
- ▶ [R and Python Cheatsheets](#) - usefull summaries for various packages in Python.
- ▶ [R](#) along with [RStudio](#)
- ▶ [Anaconda](#) - an open source distribution of Python (similarly to RStudio + R), which simplifies package management and deployment.

Old and New Software

To avoid any problems with incompatible packages and older R or Python version, as well as any problems between R and JupyterLab:

1. Uninstall Anaconda, RStudio and R. Follow the guide in the notes to remove old R packages.
2. Install a new version of R.
3. Install a new version of RStudio.
4. Open RStudio and:
 - ▶ Install tidyverse, Rcpp and ctv packages.
 - ▶ From ctv install the Econometrics, Finance, TimeSeries, Graphics and NumericalMathematics Topics.
 - ▶ You can install additional Topics, though the total amount of packages might considerably slow down the help functionality in RStudio.
5. Open cmd and use `'jupyter kernelspec uninstall ir'` to remove the existing R kernel.
6. Install a new version of Anaconda and **update all of its libraries**, as per the guide in the notes.
7. Install Git.
8. Install the IRkernel kernel in order to add R to JupyterLab.
9. Create a shortcut to launch JupyterLab from your desktop (this is faster than launching the Anaconda Navigator).

Core Literature

- ▶ R. Lapinskas, *Practical Econometrics II. Time Series Analysis (Lecture Notes)*, 2016
- ▶ R. Leipus, *Ekonometrija II*, 2016

Supplementary Literature

- ▶ P. J. Brockwell, R. A. Davis, *Introduction to Time Series and Forecasting*, 3rd ed., Springer, 2016
- ▶ R. S. Tsay, *An Introduction to Analysis of Financial Data with R*, 1st ed., Wiley, 2012