Practical Econometrics II / II

(with R and Python)

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

Course Information

Website

- Practical Econometrics courses are divided into two parts (semesters).
- The course is taught at Vilnius University, Faculty of Mathematics and Informatics.
- All course information - announcements, grading, lectures, examples, etc. will be available at the course website.

Aim of the course (1)

The aim of these courses is to provide the **foundations** of the theory and methods used in analyzing, modelling and forecasting time series data in econometrics.

Data examples consist of: data generation and analysis as well as some empirical (i.e. real-life) data examples. When dealing with time series data, most of the challenges discussed throughout the course can be present in a single time series, so the focus is on data simulation to highlight different model properties.

Aim of the course (2)

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

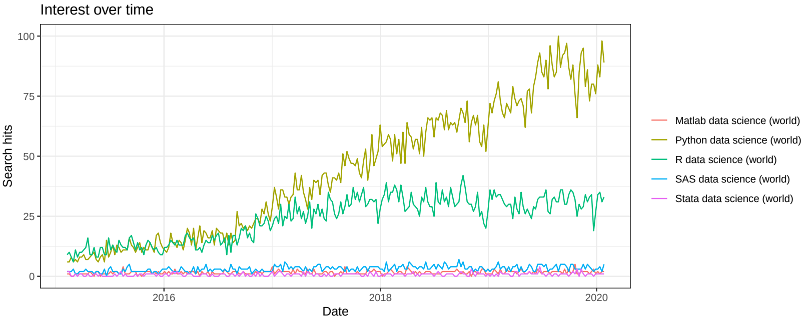
1. Statistical data and their models
2. Stationary time series: WN process, AR, MA and ARCH models
3. Time series with trend and seasonality components
4. Time series with unit root
5. Regressions with time lags
6. Regressions with time series variables
7. Multivariate models: Granger causality, VAR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

Topics marked with a '*' might only be briefly presented due to time constraints.

Software

Two programming languages will be used throughout this course:

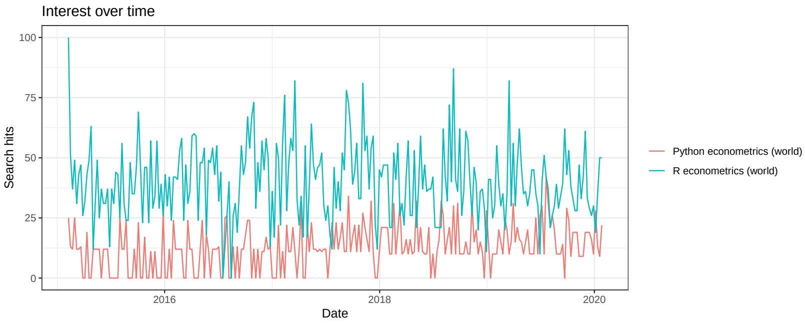
- R and Python



The popularity of both R and Python has increased immensely throughout the last 5 years.

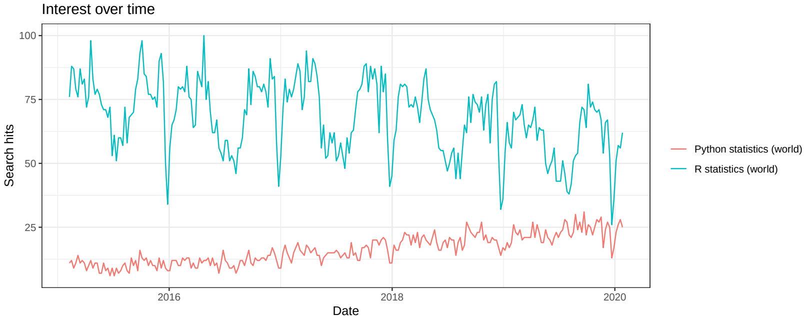
Details on installation, setup, introductory examples, etc. are provided in the lecture notes.

```r
library(gtrendsR)
library(reshape2)
library(ggplot2)
# Google Trends Query
lang_trend <- gtrends(c("R econometrics",
                        "Python econometrics"))
plot(lang_trend)
```



Will Python be a contender to R in econometric analysis?

```r
# Google Trends Query
lang_trend <- gtrends(c("R statistics",
                        "Python statistics"))
plot(lang_trend)
```



No matter how you look at it - the popularity of `Python` is increasing over time.

Course Outline

The course can be divided into three main parts:

1. Stationary time series, trend and seasonality (-> `Midterm 1`)
2. Time series with unit root and time series variables (-> `Midterm 2`)
3. VAR, VECM and Panel data models (-> `Final Exam`)

The first part focuses on the general concepts by examining a popular subset of time series models. These models are then extended to account for the effect of time itself - by examining *trend* and *seasonality*. The first part is concluded with the first midterm.

The second part **expands** on the methodology in the first part. More advanced models are examined. Finally, instead of analysing a single series, we consider the possibility of additional time series variables. The second part is concluded with the second midterm.

The final, third, part examines a more general class of models by examining a **multivariate** model of different variables, allowing them to depend on the past values of each other. Finally, in addition to the time dimension, we examine an *additional* **cross-sectional** (e.g. person, firm, country, et.c) dimension, which leads to panel data models. The third part is concluded with the final exam.

Grade Assessment

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

1. `Midterm 1` $=$ 30 % (duration ~ 3 hours)
2. `Midterm 2` $=$ 30 % (duration ~ 3 hours)
3. `Final Exam` $=$ 40 % (duration ~ 3 - 4 hours)

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

Make sure that you have installed the required software beforehand! Past student experience shows that simply installing the software does not work always work, due to different windows versions, package updates and so on.

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

Examination Dates (preliminary)

- `Midterm 1`: ∼ 2020-03-17
- `Midterm 2`: ∼ 2020-04-28
- `Exam`: June of 2020 (possible early exam session sometime in the interval of May 18 - 23)

Important:

- Bring some paper and a pen - it will be easier to derive/rewrite some models by hand before writing them in the solutions file.
- You are responsible for making sure that the software on your laptop is working (`R`, `Python`, their packages, etc.).
- **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, or email, but do it **before** the examination dates.

Submission for Grading

File name should be selected for each task separately
`Name_Surname_Task_1`, `Name_Surname_Task_2`, etc.

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!

Again, renaming the code file will not magically transform it into a `.html` file - it mus be compiled!

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**:

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

- **Python**:

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

where `student_code` is your unique student code. The start of the student code depends on the year and study programme, with the last 4 digits being unique to each student:

- `171xxxx` for VU Econometrics students, who enrolled in 2017;
- `181xxxx` for VU Econometrics students, who enrolled in 2018;
- `190xxxx`, or `200xxxx` for ERASMUS students;

Note: replace `xxxx` with the rest of your student code.

Preparing for the Midterm(-s) and the Exam

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.

Extra Task

An extra task worth 5 points (roughly 0.5 to final grade) will be specified in $\sim$2 weeks time. The task will be individual-based.

Old and New Software (**Recommendation**)

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

1. Open `cmd` and use '`jupyter kernelspec uninstall ir`' to remove the existing `R` kernel.
2. Uninstall `Anaconda`, `RStudio` and `R`. Follow the guide in the notes to remove old `R` packages.
3. Install a new version of `R`.
4. Install a new version of `RStudio`.
5. 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`.
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).