Practical Econometrics I / 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 methodology used in analyzing economic and financial data, model creation, estimation and forecasting using statistical and econometrical methods. Theoretical background and proofs are presented with focus on their effects and meaning in empirical applications.

Empirical applications consist of: data generation and analysis as well as empirical (i.e. real-life) data examples.

Aim of the course (2)

Note that traditionally example data is usually used to **highlight specific methodology applications** and it is usually not an example of a **complete econometric analysis routine** from start-to-finish - some results are ignored, often times only the output is explained without much focus on the result implication for the data analysis as a whole.

In this course, we will additionally focus on a "complete" analysis example (in terms of the scope of this course) on at least one dataset - if some methods cannot be applied to the same data - a new dataset will be examined. The aim is to examine not only **what** do the various methods show, but also **how** the resulting implications of these methods on the analysis of our data, and even cases **when these methods do not work**.

Course Outline

The course can be divided into three main parts:

1. Univariate Regression (-> `Midterm 1`)
2. Multivariable Regression (-> `Midterm 2`)
3. Discrete response and *other* models (-> `Final Exam`)

The first part focuses on the general concepts of econometrics starting from one of the simplest models. Model results are examined in different ways, various tests are carried out and conclusions presented for the examined empirical data. The first part is concluded with the first midterm.

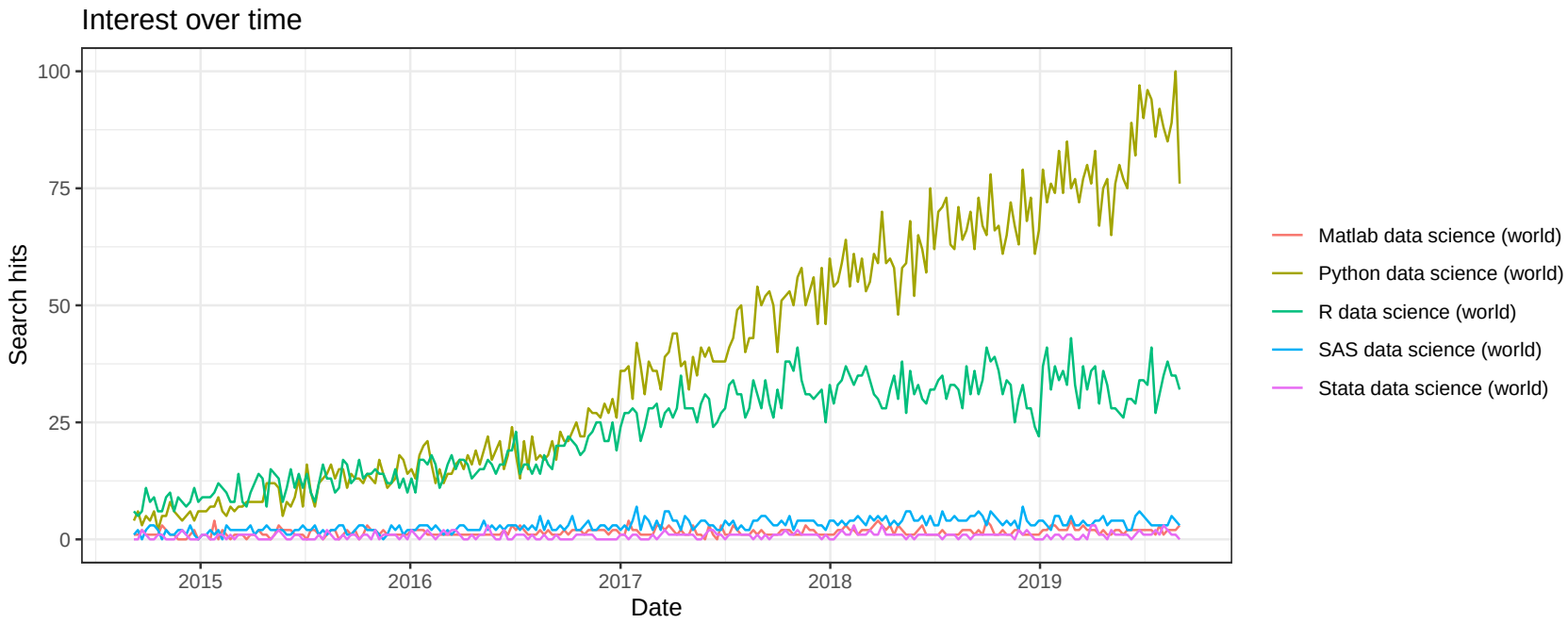
The second part **expands** on the methodology in the first part. More advanced models and more complex variable relationships are examined. Finally, most frequently encoutnered modelling problems are presented with various ways to try to account for them. The second part is concluded with the second midterm.

The final, third, part examines a more general class of models that deal with binary response data. Most of the core ideas of the **previous chapters** remain relevant. New methods are presented, which are tailored for these specific models. The third part is concluded with the final exam.

Software

Two programming languages will be used throughout this course:

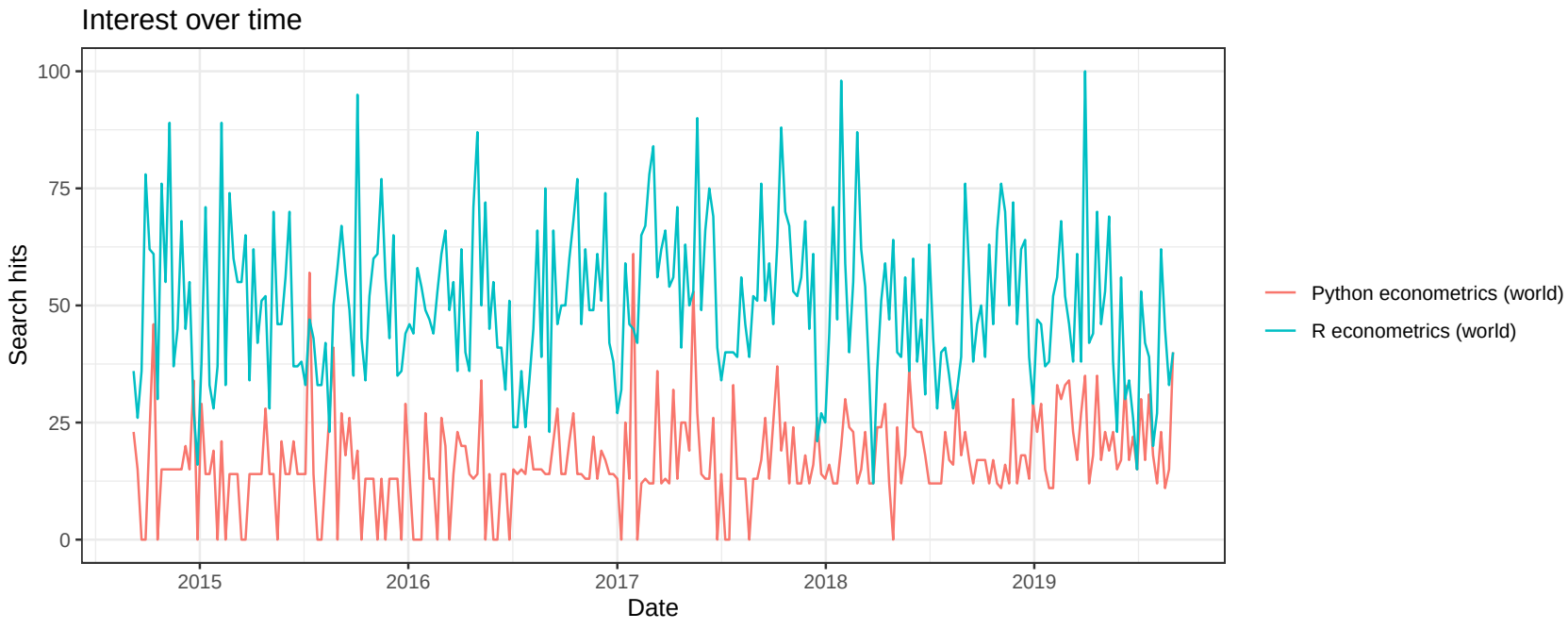
- `R` and `Python`



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?

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`: $\sim$ 2019-10-17
- `Midterm 2`: $\sim$ 2019-11-28
- `Exam`: January of 2020.

Important:

- You are responsible for making sure that the software (`R`, `Python`, their packages, etc.) on your laptop 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.

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_1] Midterm_1`
- `[PE_1] Midterm_2`
- `[PE_1] 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` 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.