# Practical Econometrics I / II

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

# Course Information

## Website

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

Practical Econometrics courses are divided into two parts (semesters). The aim of these courses is to provide the foundations of the methodology used in analysing economic and financial data, model creation, estimation and forecasting using statistical and econometrical methods. Theoretical background and proofs are presented in order to highlight their effects in empirical applications.
Empirical applications consist of: data generation and analysis as well as real-life data examples. Note that the example data is usually used to *highlight specific methodology applications* and it is usually not an example of a *complete econometrics analysis routine* from start-to-finish.

# Course Outline

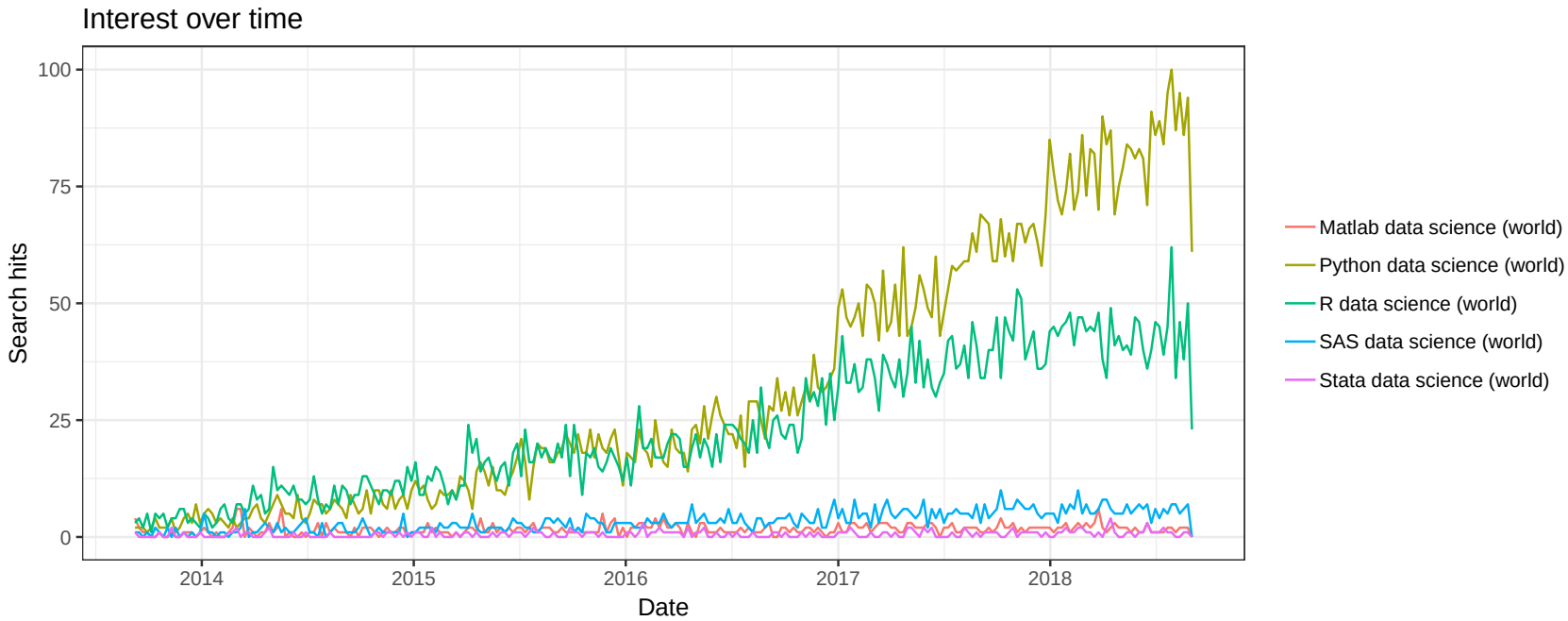
The course can be divided into three main parts:

1. Univariate Regression
2. Multivariate Regression
3. Discrete response and *other* models

# 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. will be provided in a separate file.

# 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 2 - 3 tasks and will be carried out primarily using statistical software. The tasks will specify whether `R`, or `Python` will be necessary.

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.

# 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!**

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.

# Theoretical and Empirical requirements

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.

During the midterm and final exams the following will be required in your submitted files:

- **Your code** used for the solutions to the tasks and their parts;
- **The output from your code**, which provides answers to the questions of the tasks;
- **Your comments/explanations/formulas on the results from your code**;

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.