Digitise, Optimise, Visualise:
Digital Bootcamp

Peter Gruber, USI Lugano
peter.gruber@usi.ch

Instructor

Peter Gruber has PhDs in financial economics and particle physics. He is a Senior Scientist and Lecturer at the Università della Svizzera italiana (USI) in Lugano. Before this, he has done research in neutrino physics at CERN. His current research interests are asset pricing, blockchain economics and data science with non-traditional data sets. Peter has been teaching how to solve economics and finance problems with the computer since 2005.

Prerequisites

There are no prerequisites with respect to programming, but students are expected to have completed the online introduction to Python on datacamp.com before the start of the course. Students should bring their laptops with Python and Jupyter installed to class.

Learning goals

The goal of this course is to introduce students to new and useful tools for problem solving, especially with respect to mathematics, computation and data science. Great emphasis will be put on implementation details ("making it work") and soft skills ("organising the work"). After this course, students should be able to

- Follow the subsequent parts of the summer school.
- Use Python efficiently and competently.
- Set up their tool chain for reproducible empirical work.
- Create engaging, yet statistically correct data visualisations.

Course organisation and didactics

The course is organised in blocks of usually 4 hours. In each block, we will develop the theoretical foundations for a new topic, discuss implementation in Python and work through programming examples. Daily problem sets – partly in class, partly after class – allow students to immediately test and apply their new capabilities.

Teaching material

A free access to the datacamp courses for Python will be provided. All slides and code examples will be made available to the students.

Detailed Contents

Linear algebra primer

- Scipy – linear algebra in Python
- Vectors, matrices, vector spaces
- Linear models in matrix notation
- Matrix algebra: rank, inverse, determinant, trace, norms
- Solving matrix equations
- A little tensor analysis: matrix derivatives and an alternative OLS derivation

Numerics

- Numpy – numerics in Python
- Floating point representation, precision, errors (analytic and numeric)
- Condition of a problem

Programming and Algorithms

- Setup: Python, Jupyter and required modules
- A Recap of basic Python
- From theory to program: translating algorithms into Python
- Good and bad algorithms (convergence, precision)
- Standard algorithms in linear algebra (Gauss, LU, ... – time permitting)
- Inside Python modules: setting up and using CVXOPT, CVXPY and CPLEX

Data

- Pandas – managing data in Python
- Data toolchain: extract, transform, load, merge, filter
- Random variables, data types, relationships
- A few useful datasets: stocks, interest rates, exchange rates, macro economics

Visualisation

- Plotly – advanced plotting in Python
- Vis foundations: perception, design, grammar of graphics
- Plotting concepts I: functions, inequalities, levels and contours
- Plotting concepts II: Vectors and planes, projections, matrices and linear maps
- Plotting data: statistical and exploratory plots, scatterplots and convex hulls
- Animated and interactive visualisations using Bokeh (time permitting)

Grading

Grading will be based on an individual solution to a take-home exam.