

Part 2: Lecture 1

TECH2: Introduction to Programming, Data, and Information Technology

Richard Foltyn

NHH Norwegian School of Economics

September 24, 2025

Contents

- 1 Python ecosystem
- 2 Outline of part 2
- 3 Software & tools
- 4 Goals for today
- 5 Additional resources

About me

- Undergraduate studies in software engineering (& economics), PhD in Economics
- Research fields: Quantitative Macroeconomics & Household Finance
- 20+ years of programming experience:
 - Previously (and mostly forgotten): C/C++, Visual Basic, Java, Java Script, PHP, Perl, SQL, Matlab, R
 - These days: Python, Fortran, Unix shell scripts, Stata

Contact

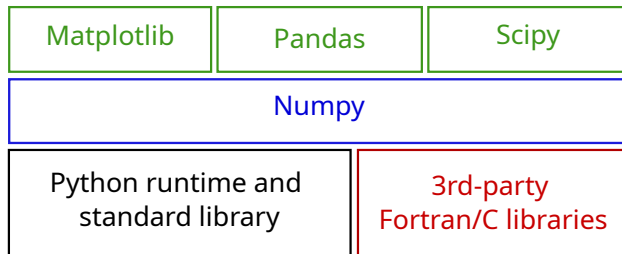
- Email: richard.foltyn@nhh.no
- Office: D231 (SAM, 2nd floor in the new building)

PYTHON ECOSYSTEM

Python software stack

How things fit together

- “Python” is the language & standard library supported by the [Python Software Foundation](#)
- For numerical applications, we need additional 3rd-party packages such as [NumPy](#), [SciPy](#), etc.
- For working with data, we use additional packages such as [Pandas](#)



Python software stack for quantitative work

Used in this course

- **Python** language, runtime and standard libraries (“Python”)
- **NumPy**: implements n -dimensional arrays, linear algebra routines, random number generators
- **Matplotlib**: High-level plotting routines for visualization
- **Pandas**: Containers to handle heterogeneous data & routines for data analysis
- **Jupyter notebooks**: Interactive documents that contain code, figures, and text

Not used in this course (but useful for future work)

- **SciPy**: Optimization routines, sparse matrices, integration, interpolation, linear algebra, statistics
- **statsmodels**: Estimation of (mostly linear) econometric & statistical models
- **scikit-learn**: Routines used for machine learning (Ridge regression, Lasso, elastic net, etc.)

OUTLINE OF PART 2

Outline of part 2

Week 39: Version control & Visual Studio Code

- Working with the git version control system
- Working with the Visual Studio Code editor
- Application: Write and benchmark `argmax()` function

Week 40: Introduction to pandas

- Creating/importing data
- Data cleaning
- Summary statistics
- Indexing
- Time series data

Outline of part 2 (continued)

Week 41: Plotting

- Creating line plots, bar charts, etc.
- Plotting data with pandas

Week 42: Data wrangling I

- Aggregation and reduction
- Transformations
- Resampling of time series data

Week 43: Data wrangling II

- Concatenating data
- Merging & joining data

TECH2 is the foundation for other courses in BEDS

Semester 2

1 TECH3: Applied Statistics

- Uses Python as programming language for statistical analysis

Semester 3

2 TECH4: Mathematics for Decisions

- Python optimizers and solvers

Semester 5

3 TECH5: Programming and Application Design

- Object-oriented programming with Python

4 TECH6: Data Science and Econometrics

- AI tools to translate Python to R

+ any other task that requires you to work with data

TOOLS:
GIT, GITHUB, AND VS CODE

Goal: learn to use industry-standard tools for programming in Python

- Python distribution: Anaconda
- Version control: git
- Code hosting: GitHub
- Editor: Visual Studio Code

Why git? (and GitHub)

- Because everyone uses it: almost completely replaced all other version control systems over the last 19 years

Examples:

- Python: <https://github.com/python/cpython>
- NumPy: <https://github.com/numpy/numpy>
- SciPy: <https://github.com/scipy/scipy>
- Pandas: <https://github.com/pandas-dev/pandas>
- Matplotlib: <https://github.com/matplotlib/matplotlib>
- PyTorch (Meta's ML library): <https://github.com/pytorch/pytorch>
- TensorFlow (Google's ML library): <https://github.com/tensorflow/tensorflow>
- Keeps history of **your** code changes (and can restore previous versions)
- Keeps history of **other's** code changes
- Allows for decentralized coding in teams
- Allows synchronizing of code across devices

Why GitHub?

- Everyone uses it!
- Alternatives (less popular):
 - [GitLab](#)
 - [BitBucket](#)
- Offers many other services besides version control (issue tracking, Wiki, etc.)
- Register for free at <https://github.com/signup>

Why Visual Studio Code?

- Has become the most widely used editor for most languages (see [StackOverflow Developer Survey 2024](#))
- Free & open source
- Good support for almost any programming language and file format (e.g., Jupyter Notebooks) via extensions
- Natively supports git & GitHub (unlike older editors)
- Alternative: PyCharm by JetBrains (free community edition is available, free professional edition for students)
- Note: [Visual Studio Code](#) is completely independent of [Visual Studio](#), a commercial IDE from Microsoft for Windows development

VS Code is the most popular editor

“Which development environments did you use regularly over the past year?”

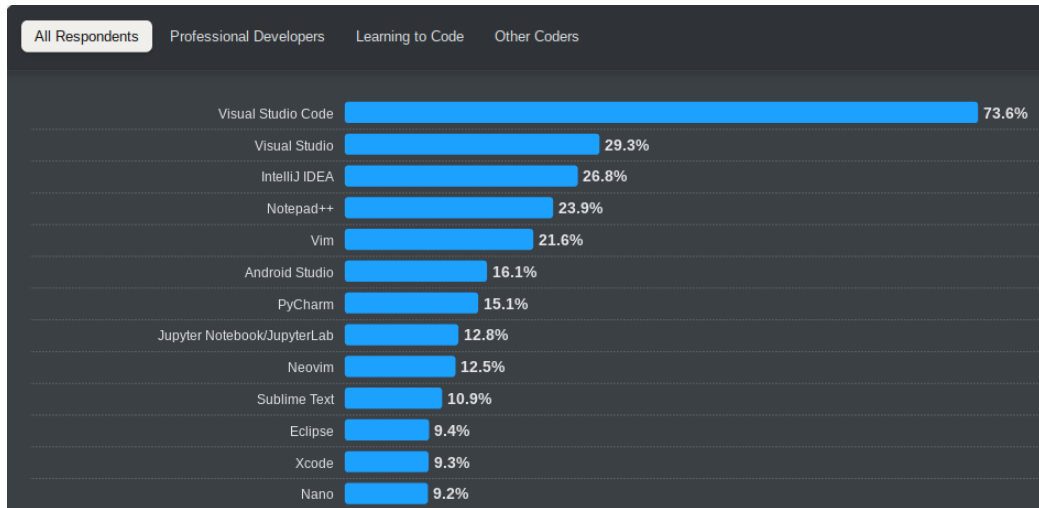


Figure 1: Source: [StackOverflow Developer Survey 2024](#)

GOALS FOR TODAY

Goals for today

git & GitHub

- 1 Create your own fork of repository at <https://github.com/richardfoltyn/TECH2-H25>
- 2 Create local clone of your fork on your computer

Visual Studio Code

- 3 Open repository in VS Code and familiarize yourself with the environment
- 4 Explore using VS Code by implementing a function `argmax()` and benchmarking it against NumPy's implementation.

This is a good opportunity to practice working with NumPy!

- 5 Integrate git into your programming workflow: add commits as you finish individual tasks

ADDITIONAL RESOURCES

Additional resources — Videos

Introduction to the command line / terminal:

- Absolute BEGINNER Guide to the **Mac OS** Terminal [17 min]
<https://youtu.be/aKRYQsKR46I>
- Git Bash - Simplest command line program for **Windows** [7 min]
<https://youtu.be/yoZ910JQzrg>

Introduction to using git

- Git for dummies [20 min] <https://youtu.be/mJ-qvsxPHpY>
- Git and GitHub Tutorial for Beginners [46 min] <https://youtu.be/tRZGeaHPoaw>
- Git Essentials in VS Code [30 min] <https://youtu.be/twsYxYaQikI>
Focuses on interacting with git and GitHub through VS Code