Part 2: Lecture 2

TECH2: Introduction to Programming, Data, and Information Technology

Richard Foltyn

Norwegian School of Economics (NHH)

October 9, 2024

Contents of lecture 2

Visual Studio Code

Visual Studio Code

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 Spyder and older editors)

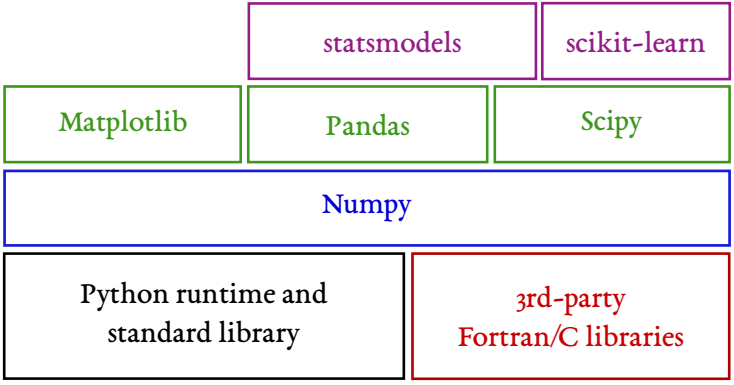
- Alternative: PyCharm by JetBrains (free community edition is available, free professional edition for students)

- Note: Visual Studio Code completely independent of Visual Studio, a commercial IDE from Microsoft for Windows development

Python ecosystem

Python software stack

How things fit together



Python software stack (used in this course)

Core libraries for quantitative work

- Python language, runtime and standard libraries ("Python")
- NumPy: implements n-dimensional arrays, linear algebra routines, random number generators
- SymPy: library for symbolic mathematics
- Matplotlib: High-level plotting routines for visualisation
- Pandas: Containers to handle heterogeneous data & routines for data analysis
- SciPy: Optimization routines, sparse matrices, integration, interpolation, linear algebra, statistics

Python software stack (**not** covered in this course)

Econometrics & Machine learning

- scikit-learn: routines used for machine learning (Ridge regression, Lasso, elastic net, etc.)
- statsmodels: routines for estimating many (linear) models
- TensorFlow: ML library maintained by Google with Python API
- JAX: Low-level API for automatic differentiation and accelerated linear algebra used to build ML models, developed by Google
- PyTorch: Python interface to ML libraries originally developed by Facebook

Frameworks to speed things up

- Numba: compiles Python code to machine code using LLVM
- Cython: converts pseudo-Python to C code (advanced, don't use this)

More on git

Telling git to ignore files

- Often your code or operating system generates temporary files and should not be stored in the repository.
- For example, when running Python you often see the folder __pycache__:

```
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        __pycache__/
        example2.py

nothing added to commit but untracked files present (use "git add" to track)
```

- We never want such cache and temporary files to end up in our repository
- We can tell git to ignore them by creating a file called `.gitignore` in the root folder of the repository
- Each line in this file specifies a pattern or file name to ignore:

```
__pycache__/
```

Additional Resources

Video tutorials

Introduction to VS Code & git

- Git Essentials in VS Code [30 min] https://youtu.be/twsYxYaQikI
  Focuses on interacting with git and GitHub through VS Code