Open in Colab

Lecture 2: Setup

Coding: Introduction to Python - Applied Economics

Laboratory: Coding - Accounting, Finance and Business Consulting

The Tools of Data Analysis in Python

We will start presenting the coding tools we will use in the course and how to set them up

- Main tools: Python, Jupyter Notebook
- Auxiliary tools: Anaconda, Colab, VLEM

- Focus on the role of each tool
- Learn syntax to perform basic actions

Python

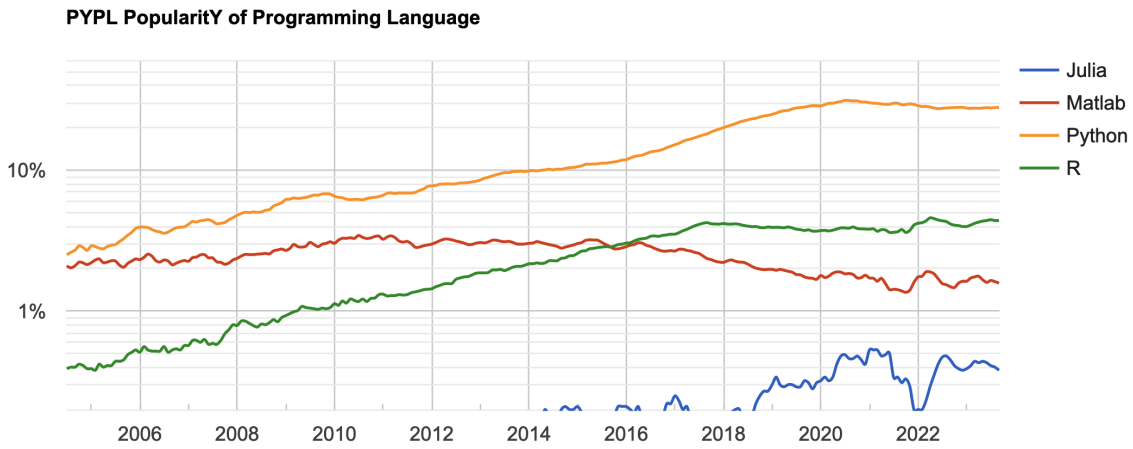
Python is a general-purpose programming language. Created in 80s by Guido van Rossum, it is now one of the most popular languages in the world.

- Not the fastest but easy to use and very versatile
  - Can be used for web development, videogames, edit videos/music, create Large Language Models (LLM) such as Chat GPT, manage networks, etc.
  - Used and supported extensively by Internet services and high-tech companies: Google, Netflix, Meta, Amazon, Reddit

Python

- Python versatility and huge community led to the development of endless extensions
  - Crucial role in today's data analyis landscape thanks to many data analysis tools (to make it easy to write/debug/share of analysis) and libraries (to clean/analyze/visualize data)
- Python developers are in huge demand in the job market
  - Knowing python can improve your efficiency and efficacy in any job that involve a computer
  - Most of the Deep Learning and LLM tools that revolutionized the data analysis environment in later years are developed in Python

Python popularity



Based on tutorial searches on Google (more info here)

Tools for Data Analysis - Python

Python: Provides a command-line Python interpreter

- Takes your code and translates it to the processor so that can be executed
- Its usual inputs are scripts that perform a set of pre-defined actions
- Not well suited to perform data anlysis that involves continuous exploration
  - Read the data, analyze it to understand some aspect, visualize the result, perform sub-analysis on the result or on the riginal data based on what has been leart from the results and so on

Tools for Data Analysis - Jupiter

IPython/Jupyter Notebooks: Extend the Python interpreter making it more usable for scientific computing

- Provides and interactive command-line terminal for Python
- Allows read-eval-print loop (REPL) to ease the interact with the data
- In 2015, the IPython became Jupyter Notebook that provides a browser-based control panel
- While Jupyter notebooks are run in a browser, they connect to a computational engine (the kernel)
  - can be used not only with Python but also with R, Julia, SageMath
- Offers a more convenient text editor, include formatted text, static and dynamic visualizations, mathematical equations, JavaScript widgets, and much more.
- Eases data analysis reproducibility by facilitating other people to execute the code on their own systems
- This presentation and the course materials are Jupyter notebooks

Tools for Data Analysis - IDEs

IDEs (Integrated Development Environment): are applications making programming easier by combining a suite of tools to develop code: code highlighting, support for writing (AI-completion), rendering, debugging, etc.

- We will have a look at Microsoft Visual Studio code (VS code) because it is integrated with Copilot, an AI-tool to write code

Using AI in this course

I expect you to use AI (e.g., ChatGPT, Bing, Copilot)

Learning to use AI is an emerging skill that can greatly improve your learning and your outputs

During the course I will show you how to use AI for some use cases

While you are invited to use AI as a learning aid, be mindful or not rely on it mindlessly: ***during the exam you will not have access to it***

Using AI in this course

Also, always be aware of the limits of AI:

- If you provide minimum-effort prompts, you will get low-quality results
  - Refining the prompts to get good outcomes will take work
- AI has a tendency to allucinate, don't trust anything it says and double-check
  - If you learn something wrong its your grade getting affected
  - **It works best for topics one understands**
    - If you do not know how to code, using AI for coding will make you lose a lot of time

Be thoughtful about when this tool is useful. Don't use it if it isn't appropriate for the case or circumstance.

ChatGPT, Bing and Copilot

- **ChatGPT** and **Bing** are great aid for learning how to code
  - You can ask for explanations on specific topics, i.e., loops, pandas, data import
  - You can ask about your specific problem, e.g. how can I find if a number is odd using python?,
  - You can ask it to explain some code, with details for each step/operation

- **Copilot** is a Chat-GPT Large Language Model trained on StackOverflow questions and answers
  - It integrates with VSCode and GitHub, if we will be able I will show how to set that up
  - Copilot can be very useful when coding because it takes as prompt the code and the comments you are writing and suggests the next lines
    - You need to have in mind the steps ahead, know what can be done and using what tools
    - You need to **always** check if the code is indeed doing what you want

Setting up your coding environment

We will see three ways to set up your coding environment:

- A local installation on your laptop
- Using the VLEM of the dSEA
- Using Google Colab

Set up the coding environment on your laptop

We will see how to do so using ***Anaconda***, a famous python and R distribution for data science

- Need to install all the software on your laptop
  - **PROs** No internet needed to run the code, no constraints on what to install, all your files readily available, all the computational power of your laptop
  - **CONs** Installation requires a bit of work, some knowledge of how a computer works and can entail unexpected problems
- Probably the best option
  - Is the most flexible and powerful, opens the door to many other tools
  - You will develop a bit of practice on some valuable computer science skills

Use an environment in the cloud

- Rely on a remote server that has ***some*** of the needed software already installed
  - **PROs** Avoid all the installation difficulties and start to code "right away"
  - **CONs** Requires an internet connection, less flexible
- We will see two options:
  - Aula Virtuale **VLEM** is a platform developed by the dSEA to support the teaching of programming and data analysis
  - **Google Colab** is a free service offered by Google that provides a Jupyter notebook environment in the cloud

Use the coding environment on the VLEM

The dSEA provides a Virtual Learning Environment that allows you to run Jupyter notebooks on a remote server

Our IT has set up a cloud instance with all the needed tools for this course

- The instance you should use is the VLEM-BASE
- Information on the VLEM is available here (in Italian, but google translate works well)
- To access the VLEM is necessary to book a slot

- **PROs** No need to install ***anything*** (is it a pro if your aim is learning?)
- **CONs** Need an internet connection, need to book a slot, limited time (30 minutes), pretty slow, cannot store anything and any modification to the environment is lost

Use a coding environment on Google Colab

There are many options for coding in Python on the cloud

- Google Google Colab is one of the best
- Offers a free tier with your Google account (that also include GPUs) and its paid tier has an extremely rich offer (GPU for AI)
- Google Colab instances come with only a minimal set of pre-installed packages
- Google instances are temporary, if you stay idle for too much or you close your window your instance gets deleted
    - But you can connect it to your Google Drive to save your notebooks and coding environment

Use a coding environment on Google Colab

- The introductory notebooks include a “Launch notebook” button that will directly run them in Colab
    - Then you will need to upload them or store them on your Gdrive
- **PROs** Always ready with one click, possibility of storing the notebooks and personalizing the environment, a framework used by professionals (great for your experience)
- **CONs** Need an internet connection, need some set up to have a semi-permanent environment (but we will see it together), the interface is a bit different from the standard Jupyter notebook

Setting up Python and Jupyter on your computer using Anaconda

Anaconda is a distribution (a collection of software that is pre-configured to work together) for Python and R that is focused on data science

- Simplifies deployment and is focused on scientific computing and data analysis
- Has a graphical interface to manage libraries and environments
- Also installs Jupyter, so that you edit notebooks
- Provides many other tools for data science
- Has a comprehensive and detailed documentation, a lively community plus a massive knwoledge base on Stackoverflow

Installing Anaconda

We will now see how to install Anaconda on your computer following the installation docs

Command Line Interface and Graphical Interface

Anaconda can be used through the command line interface (CLI) or the graphical interface (GUI)

- Each have their advantages and disadvantages
  - The CLI is more powerful and more flexible
  - The GUI is easier to use and more intuitive
- For this course the GUI is enough, but start getting your head around the CLI, we will need to use it later on

Anaconda Navigator

Anaconda Navigator is the GUI of Anaconda

- It allows you to manage your environments and libraries

To get started with Navigator we will use this anaconda doc

Setting up a python environment

An environment is the context in which a Python program runs, it specifies:

- What is the interpreter to be used
  - In this case the default version of Python
- Defines its fundamental settings
  - Where the libraries are stored

Setting up the environment for this course: codepy

To set up the environment to be used in this course

1. Open Anaconda Navigator
2. Click on the Environments tab on the left shoulder
3. Click on 'create', name the new environment `codepy` select Python and the latest version
4. Click on 'create' and wait for the environment to be created (it may take a while, it is downloading and installing a lot of stuff)
5. Click on the 'Home' tab on the left shoulder
6. Select the `codepy` environment from the drop down menu
7. Click on 'install' on the Jupyter Notebook tile
8. Click on 'install' on the VS Code tile (we will use it later on)

Congrats! You have set up your first Anaconda Python environment

Creating a Jupyter notebook

To create your first Jupyter Notebook

1. Open Anaconda Navigator
2. Go to the home tab
3. Select the `codepy` environment
4. Click on 'launch' on the Jupyter Notebook tile

Creating a Jupyter notebook

5. A new tab will open in your browser listing the files in your home directory
6. It is conveninet to store all your notebooks in a single directory
    - Click on 'New' on the top right corner and create a new directory
    - Select the "Untitled folder" just created and click the rename button on the top left
    - Rename it `codepy`
    - Click on the `codepy` directory
7. Click on 'New' on the top right corner and select "codepy" from the drop down menu
8. You can now start editing your first notebook

Install a package in Anaconda

1. Open Navigator and select the environment tab
2. Select the `codepy` environment
3. Click on the dropdown menu and select not installed
4. Search for the library you are interested in, let's say `rise`
5. Click on the checkbox on the left of the `rise` package and apply at the bottom of the window

6. Now if you change the dropdown menu to installed you will see the `rise` package listed
7. You can now import the package in your notebook and use all its functionalities

Install the packages needed for this course in Anaconda

The packages needed for this course are:

- `nbconvert` - To transform the notebooks in slides
- `rise` - To display the slides directly from the notebook (not available on the VLEM and on Colab)
- `matplotlib` - To plot data
- `seaborn` - To plot data
- `statsmodels` - To perform statistical analysis
- `scikit-learn` - To perform machine learning
- `arrow` - To manage dates

install them in the codepy environment following the steps above

Setting up the environment in Colab

If you are going to use Colab in this course (even just on occasion), it is worth to set up a semi-permanent environment

While the set-up of Anaconda is a standard procedure and there is official documentation, what we are doing is something a bit more custom, so we will make follow this tutorial.

Homework

Next week we will start coding, so you need to set up your environment

My Earnest advice is to do all three set-ups

- Each has its pros and cons
  - You will be able to choose the one you like the most
  - You will be able to use the one that is more convenient for the task at hand
  - Some things we will see are only available in one of the three
- You will learn a lot by doing so
  - On topics that are crucial for your future career
  - On tools that can widen your horizon and make you more efficient

Notes

Book references:

Ch.1 of Python for Data Analysis

Preface of Python Data Science Handbook

Office Hours:

Tuesday

15:00 - 17:00

Dsea via del Santo 33, first office on the left entering the Levi Cases inner garden

Author: Duccio Gamannossi degl'Innocenti

Web: https://www.dgdi.me/

Mail: duccio.gamannossi@unipd.it