

Python data science

Hamish Gibbs

Recap

- Great job!
- We only have 4 days to go from introductory to advanced Python concepts.
 - Plus: programming tools like VSCode and [git](#)!
- Classes and functions can be abstract, but they are the building blocks of what we will do today.
 - Hopefully today is more familiar to people who have used R!

Today: Python data science

- Introduction to Python data science tools.
- Introduction to a basic data science workflow.
- This afternoon: collaborating on a data science project.

Tomorrow

- Data science “challenge”
- *Predicting the nightly price of AirBnBs in London*
- See the Guidelines [here](#).

Data science

- Definition of data science:
 - *“Extracting meaningful insights from data.”*
- *Meaningful* is important.
 - Use the tools of programming / statistics to create meaning from your data.
- Usually, there is no “right” answer, just “better” and “worse” answers.
 - You exercise a lot of judgement.

Data science workflow

- Data science is **not** just machine learning.
 - Most data science work is:
 - Data preparation
 - Data transformation
 - Method selection
 - *Statistics / machine learning*
 - Communicating results

Python data science tools

- Today, we will learn about the most popular Python data science “stack”
 - Data preparation / data transformation
 - `pandas`, `numpy`
 - Statistics / machine learning
 - `sklearn`
 - Communicating results
 - `matplotlib`

Python data science tools

- Tomorrow, we will use this “stack” to do our data science project
- Exploratory analysis, data transformation
 - `pandas`
- Regression model fitting and evaluation
 - `sklearn`
- Visualize results
 - `pandas, matplotlib`

R equivalents

- Python libraries mostly have their R equivalents:
 - `pandas` : `dplyr`
 - `matplotlib` : `ggplot2`
 - `sklearn` : `caret`?
- See what you prefer, I use both languages!

Diving deeper

- Python has many other options for data science tools.
- Alternatives to `pandas`:
 - `polars` (*Like Python's version of `data.table`*)
 - `dask`
- Alternatives to `sklearn`:
 - ...?
- Alternatives to `matplotlib`:
 - `seaborn`
 - `plotnine` (*R users might like this one!*)

Tutorial #1: pandas and matplotlib

- [pandas-cookbook: Selecting data \(Chapter 2\)](#)
- Core concepts:
 - Reading data from a `.csv` file
 - Inspecting a dataset
 - Selecting data

Tutorial #2: pandas and matplotlib

- [pandas-cookbook: More selecting data \(Chapter 3\)](#)
- Core concepts:
 - Selection by multiple columns
 - The role of [numpy](#) in [pandas](#)
 - Basic plotting ([matplotlib](#) in [pandas](#))

Data: Tutorials 1 and 2

- Tutorials #1 and #2 come from the [pandas-cookbook](#).
- Go to the [/data](#) folder in the GitHub repository (link above).
- Download the [311-service-requests.csv](#) file and store it on your computer.

Tutorial #3: sklearn

- [sklearn - Getting Started](#)

- *Note: just work up to the “Model Evaluation” section.*

- Core concepts:

- Fitting a model to data

```
1 clf.fit(X, y)
```

- Making predictions with a model

```
1 clf.predict(X)
```

- Model evaluation

```
1 result = cross_validate(lr, X, y)
```

Packages: Tutorials 1, 2, and 3

- Install the required packages using your terminal in VSCode

```
1 pip install pandas matplotlib sklearn
```

- Trouble installing? Tell me!

Extra

- [pandas-cookbook: String operations \(Chapter 6\)](#)
 - This tutorial is about extracting information from text in [pandas](#).
 - *Hint:* Some of the most interesting information in tomorrow's dataset might be in string variables.

Extra

- Start working with the [dataset](#) we will use tomorrow.
- Work on a few of the items in the challenge:
 - *What variables are in the dataset?*
 - *What are the data types of the variables?*
 - *Is there any missing data?*