Python data science

Hamish Gibbs

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.

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`

R equivalents

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

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

- 10 minutes to pandas
  - *This is a detailed tutorial with everything in `pandas` you will need to know for tomorrow!*
- Core concepts:
  - Creating a DataFrame
  - Selection
  - Missing data
  - Grouping

Tutorial #2: plotting in pandas with matplotlib

- Chart visualization
  - *This picks up where the previous tutorial leaves off.*
- Core concepts:
  - Different types of visualizations
  - Plot formatting
  - Subplots
- **Good question:** This is a `pandas` tutorials. Where is `matplotlib` in all of this?

Tutorial #3: sklearn

- sklearn - Getting Started
  - *A very basic introduction to the way `sklearn` models work.*
- Core concepts:
  - Fitting a model to data

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

  - Making predictions with a model

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

  - Model evaluation

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

Tutorial #4: pandas / matplotlib

- Linear model: from regression to sparsity
- Core concepts:
  - Linear regression
    - I recommend focusing on the (small) linear regression section.
    - *Tip: print out each variable and try to understand how data flows through the model. What format is it in? How could you format a different dataset in the same way?*

Extra

- Start working with the dataset we will use tomorrow.
  - For help reading the data: see here.
- 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?*