Getting Started With Python

A *How-To* Guide for Social Scientists

Andreas Küpfer
Technical University of Darmstadt

✉ *andreas.kuepfer@tu-darmstadt.de*
🌐 *andreaskuepfer.github.io*
🐦 *@ankuepfer*

Ruben Bach
University of Mannheim

✉ *r.bach@uni-mannheim.de*
🐦 *@rub3n_luc*

Social Science Data Lab
MZES, University of Mannheim
February 22, 2023

Outline

1. Recap: Day 1
2. Dataset & Methods for Today
3. Outlook

Recap: Day 1

Recap: Day 1

Python is better suited as R when …

- Working with computer scientists
- Using state-of-the-art machine learning, deep learning, natural language processing
- Preparing for a data science job outside of academia
- General purpose programming

Full disclosure: If your work is focused on statistical inference and explanation (coefficients, statistical tests, …) and some ML and NLP, you'll probably be better-off with R. For prediction-focused ML, deep learning, NLP and heavy data science, you'll probably want to consider using python at some point.

Jupyter Notebook

- Web-based interactive computing environments
- Combine markdown with code, interactive cells, lots of exporting and publishing options
- Can run bash commands from within notebook using ”!”

Example: List installed packages using bash

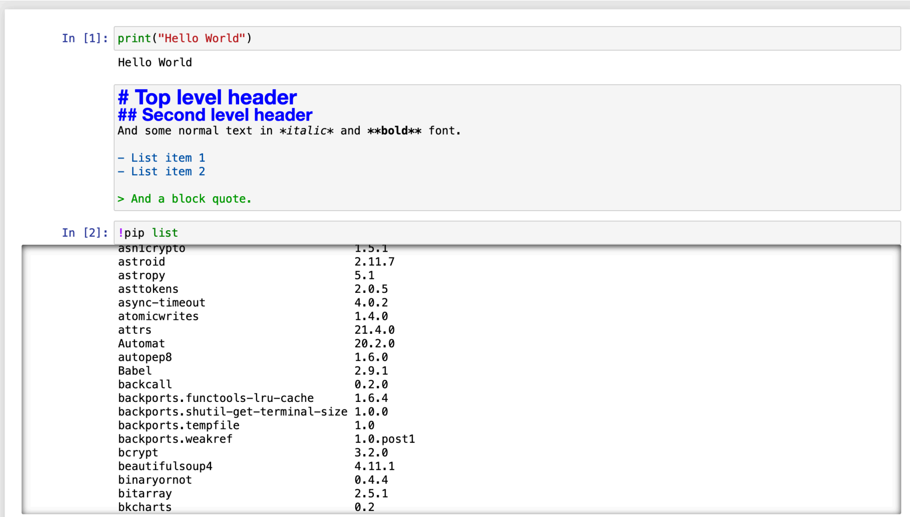
```
!pip list
```

- Magics (% and %%) give additional cell functionality

Example: Include HTML content in notebook

```
%%HTML
```

Jupyter Notebook



```
In [1]: print("Hello World")
```

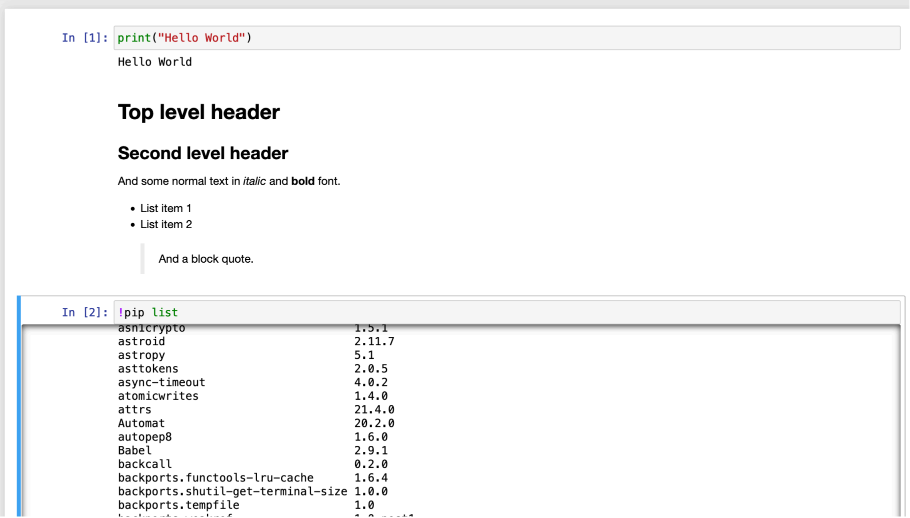
```
Hello World
```

```
# Top level header
## Second level header
And some normal text in *italic* and **bold** font.

- List item 1
- List item 2

> And a block quote.
```

```
In [2]: !pip list
```

```
asn1crypto                         1.5.1
astroid                            2.11.7
astropy                            5.1
asttokens                          2.0.5
async-timeout                      4.0.2
atomicwrites                       1.4.0
attrs                              21.4.0
Automat                            20.2.0
autopep8                           1.6.0
Babel                              2.9.1
backcall                           0.2.0
backports.functools-lru-cache      1.6.4
backports.shutil-get-terminal-size 1.0.0
backports.tempfile                 1.0
backports.weakref                  1.0.post1
bcrypt                             3.2.0
beautifulsoup4                     4.11.1
binaryornot                        0.4.4
bitarray                           2.5.1
bkcharts                           0.2
```

Jupyter Notebook



```
In [1]: print("Hello World")
```

```
Hello World
```

Top level header

Second level header

And some normal text in *italic* and **bold** font.

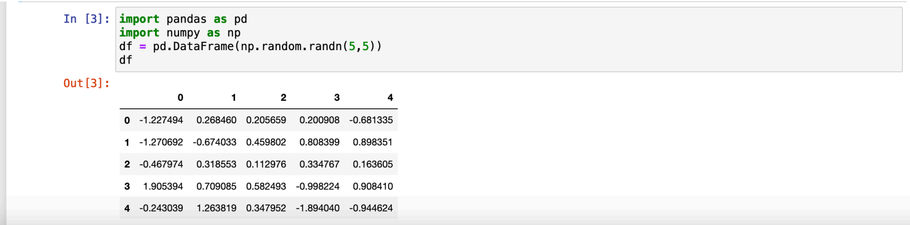
- List item 1
- List item 2

> And a block quote.

```
In [2]: !pip list
```

```
asn1crypto                        1.5.1
astroid                           2.11.7
astropy                           5.1
asttokens                         2.0.5
async-timeout                     4.0.2
atomicwrites                      1.4.0
attrs                             21.4.0
Automat                           20.2.0
autopep8                          1.6.0
Babel                             2.9.1
backcall                          0.2.0
backports.functools-lru-cache     1.6.4
backports.shutil-get-terminal-size 1.0.0
backports.tempfile                1.0
backports.weakref                 1.0.post1
```

Jupyter Notebook

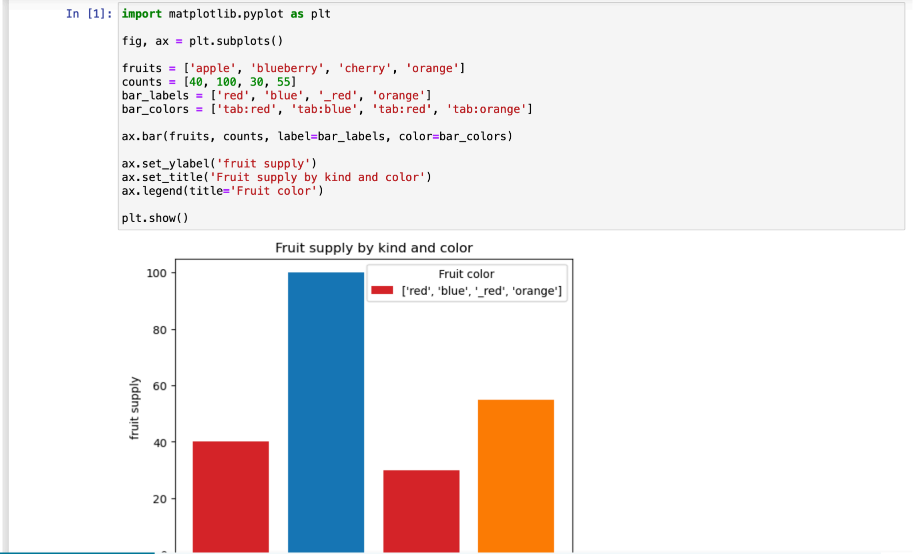


```
In [3]: import pandas as pd
        import numpy as np
        df = pd.DataFrame(np.random.randn(5,5))
        df
```

Out[3]:

	0	1	2	3	4
0	-1.227494	0.268460	0.205659	0.200908	-0.681335
1	-1.270692	-0.674033	0.459802	0.808399	0.898351
2	-0.467974	0.318553	0.112976	0.334767	0.163605
3	1.905394	0.709085	0.582493	-0.998224	0.908410
4	-0.243039	1.263819	0.347952	-1.894040	-0.944624

Jupyter Notebook



In [1]:

```
import matplotlib.pyplot as plt

fig, ax = plt.subplots()

fruits = ['apple', 'blueberry', 'cherry', 'orange']
counts = [40, 100, 30, 55]
bar_labels = ['red', 'blue', '_red', 'orange']
bar_colors = ['tab:red', 'tab:blue', 'tab:red', 'tab:orange']

ax.bar(fruits, counts, label=bar_labels, color=bar_colors)

ax.set_ylabel('fruit supply')
ax.set_title('Fruit supply by kind and color')
ax.legend(title='Fruit color')

plt.show()
```

Fruit supply by kind and color
Fruit color
['red', 'blue', '_red', 'orange']
fruit supply
100
80
60
40
20
0

Google Colaboratory

”Colab”

- Web-based interactive computing environments
- Google account required (to edit code)
- Runs on Google hardware (including free GPU use)
- Powerful hardware available per premium plans
- Markdown-based text cells and code cells, magic commands, bash commands (”!”), R kernels available, LaTeX, ...
- Load data from your Google Drive
- Pay attention when sharing colab notebooks

Google Colaboratory

”Colab” – Downsides

- Lose data and results when runtime stops
- Uploaded files removed when session restarted, no persistent storage
- Sensitive data? Sensitive code?

Dataset & Methods for Today

Dataset & Methods

Wage Dataset 1985 (Current Population Survey)

- Gathered from OpenML Datasets
- 534 persons
- 11 (numerical/categorical) features

Methods: Logistic Regression & Random Forest

- Training and hyperparameter optimization on 80% of the data
- Evaluation on 20% of the data
- Extraction of feature importance

Workshop Material

- No local Python installation required to participate, you can follow along within the Notebook
- Viewing is possible without a Google Account, for editing code you need to sign in
- Link to Google Colab Workshop Notebook

Outlook

There is much more to explore...

- A useful pandas cheatsheet of the package can be found here: Pandas Cheatsheet
- How to choose the right ML estimator? Some guidance can be found here
- Deep Learning: TensorFlow or YouTube Series
- Beyond Machine Learning in Python: from Plotly to sophisticated analysis based on extremely large (unstructured) data sources (e.g. with Dask or PySpark)