

Computational Analysis of Big Data

Week 1

Mostly an introduction, but also:

Coding with data in Python

Course overview

Course overview

Sessions

1. Coding with data in Python
2. A Data Scientist's most fundamental tools
3. Getting data—scraping and APIs
4. Machine learning 1
5. Machine learning 2
6. Networks
7. Natural language processing
8. Crunching Big Data with MapReduce
9. Ethical and legal considerations in Big Data
10. Lab work on project report
11. Lab work on project report
12. Project presentations

Course overview

Sessions

- 1. Coding with data in Python**
2. A Data Scientist's most fundamental tools
3. Getting data—scraping and APIs
4. Machine learning 1
5. Machine learning 2
6. Networks
7. Natural language processing
8. Crunching Big Data with MapReduce
9. Ethical and legal considerations in Big Data
10. Lab work on project report
11. Lab work on project report
12. Project presentations

```

212 def randomize_by_edge_swaps(G, num_iterations):
213     """Randomizes the graph by swapping edges in such a way that
214     preserves the degree distribution of the original graph.
215
216     Source: https://gist.github.com/gotgenes/2770023
217     """
218     newgraph = G.copy()
219     edge_list = newgraph.edges()
220     num_edges = len(edge_list)
221     total_iterations = num_edges * num_iterations
222
223     for i in xrange(total_iterations):
224         rand_index1 = int(round(random.random() * (num_edges - 1)))
225         rand_index2 = int(round(random.random() * (num_edges - 1)))
226         original_edge1 = edge_list[rand_index1]
227         original_edge2 = edge_list[rand_index2]
228         head1, tail1 = original_edge1
229         head2, tail2 = original_edge2
230
231         # Flip a coin to see if we should swap head1 and tail1 for
232         # the connections
233         if random.random() >= 0.5:
234             head1, tail1 = tail1, head1
235
236         if head1 == tail2 or head2 == tail1:
237             continue
238
239         if newgraph.has_edge(head1, tail2) or newgraph.has_edge(
240             head2, tail1):
241             continue
242
243         # Succeeded checks, perform the swap
244         original_edge1_data = newgraph[head1][tail1]
245         original_edge2_data = newgraph[head2][tail2]
246
247         newgraph.remove_edges_from((original_edge1, original_edge2))
248
249         new_edge1 = (head1, tail2, original_edge1_data)
250         new_edge2 = (head2, tail1, original_edge2_data)
251         newgraph.add_edges_from((new_edge1, new_edge2))
252
253         # Now update the entries at the indices randomly selected
254         edge_list[rand_index1] = (head1, tail2)
255         edge_list[rand_index2] = (head2, tail1)
256
257     assert len(newgraph.edges()) == num_edges
258

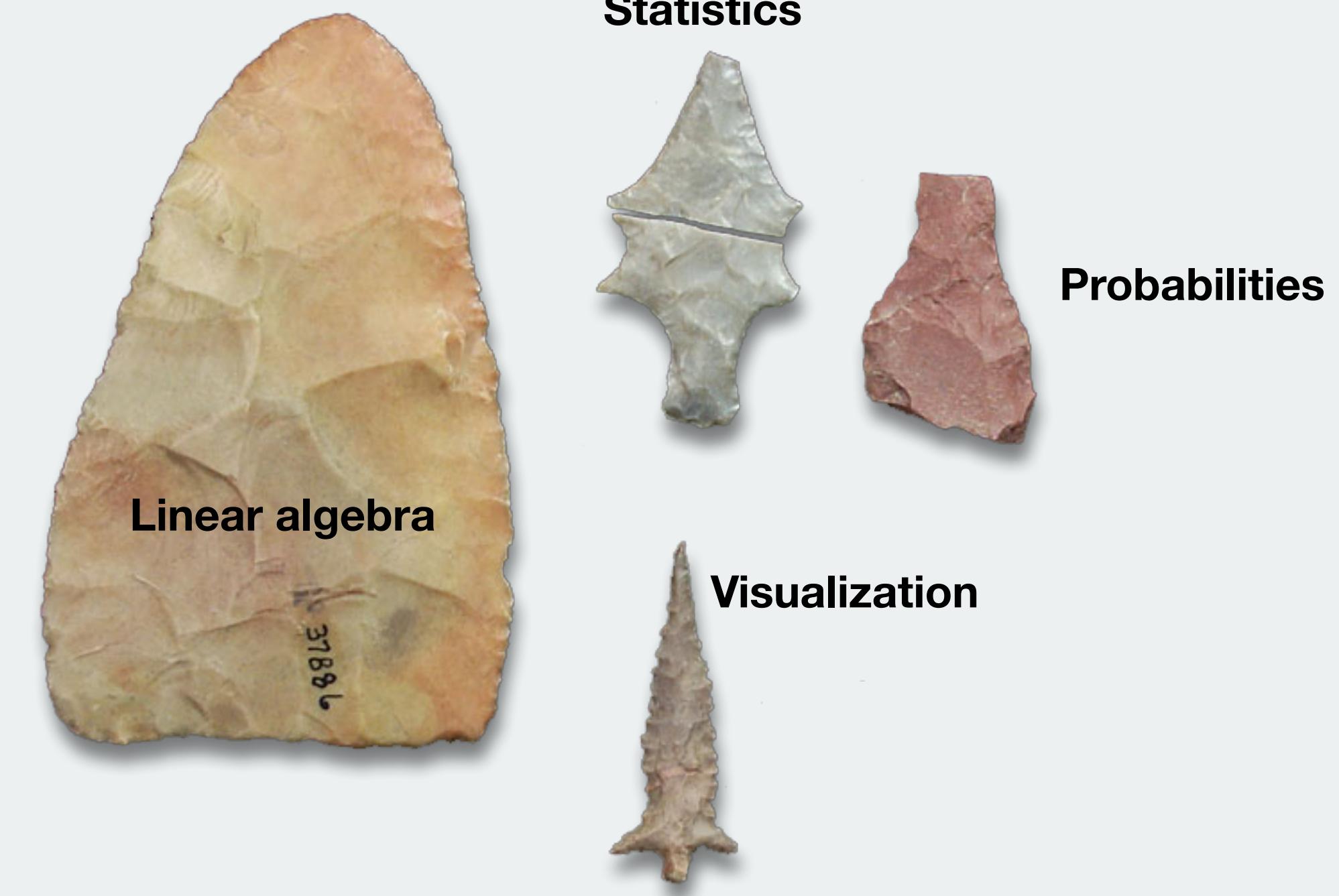
```

Aim: Get comfortable with basic Python

Course overview

Sessions

1. Coding with data in Python
2. **A Data Scientist's most fundamental tools**
3. Getting data—scraping and APIs
4. Machine learning 1
5. Machine learning 2
6. Networks
7. Natural language processing
8. Crunching Big Data with MapReduce
9. Ethical and legal considerations in Big Data
10. Lab work on project report
11. Lab work on project report
12. Project presentations

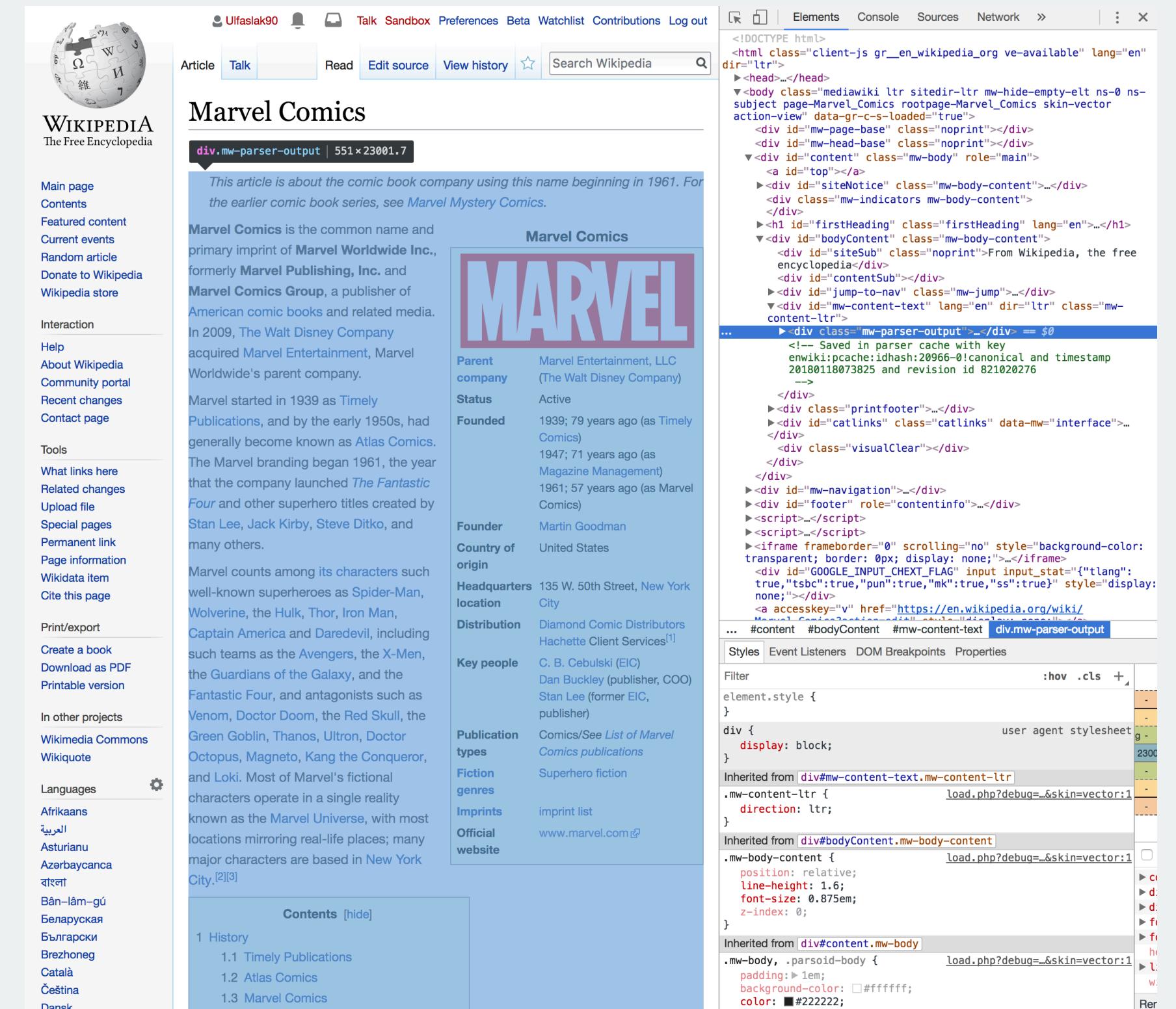


Aim: Refresh math skills and learn how to use them in Python

Course overview

Sessions

1. Coding with data in Python
2. A Data Scientist's most fundamental tools
3. Getting data—scraping and APIs
4. Machine learning 1
5. Machine learning 2
6. Networks
7. Natural language processing
8. Crunching Big Data with MapReduce
9. Ethical and legal considerations in Big Data
10. Lab work on project report
11. Lab work on project report
12. Project presentations

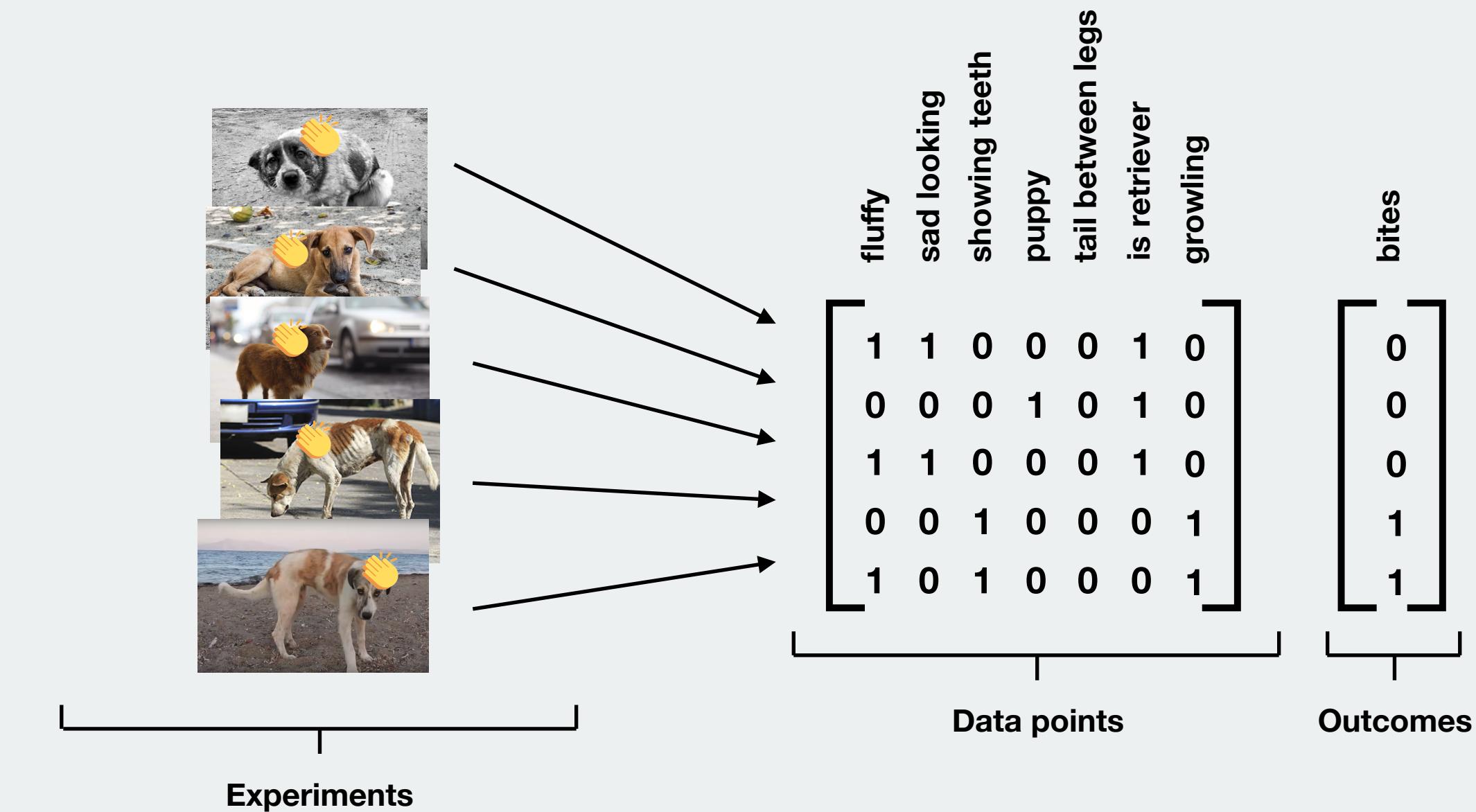


Aim: Learn how to get data from web and get some from the Wiki API

Course overview

Sessions

1. Coding with data in Python
2. A Data Scientist's most fundamental tools
3. Getting data—scraping and APIs
- 4. Machine learning 1**
- 5. Machine learning 2**
6. Networks
7. Natural language processing
8. Crunching Big Data with MapReduce
9. Ethical and legal considerations in Big Data
10. Lab work on project report
11. Lab work on project report
12. Project presentations

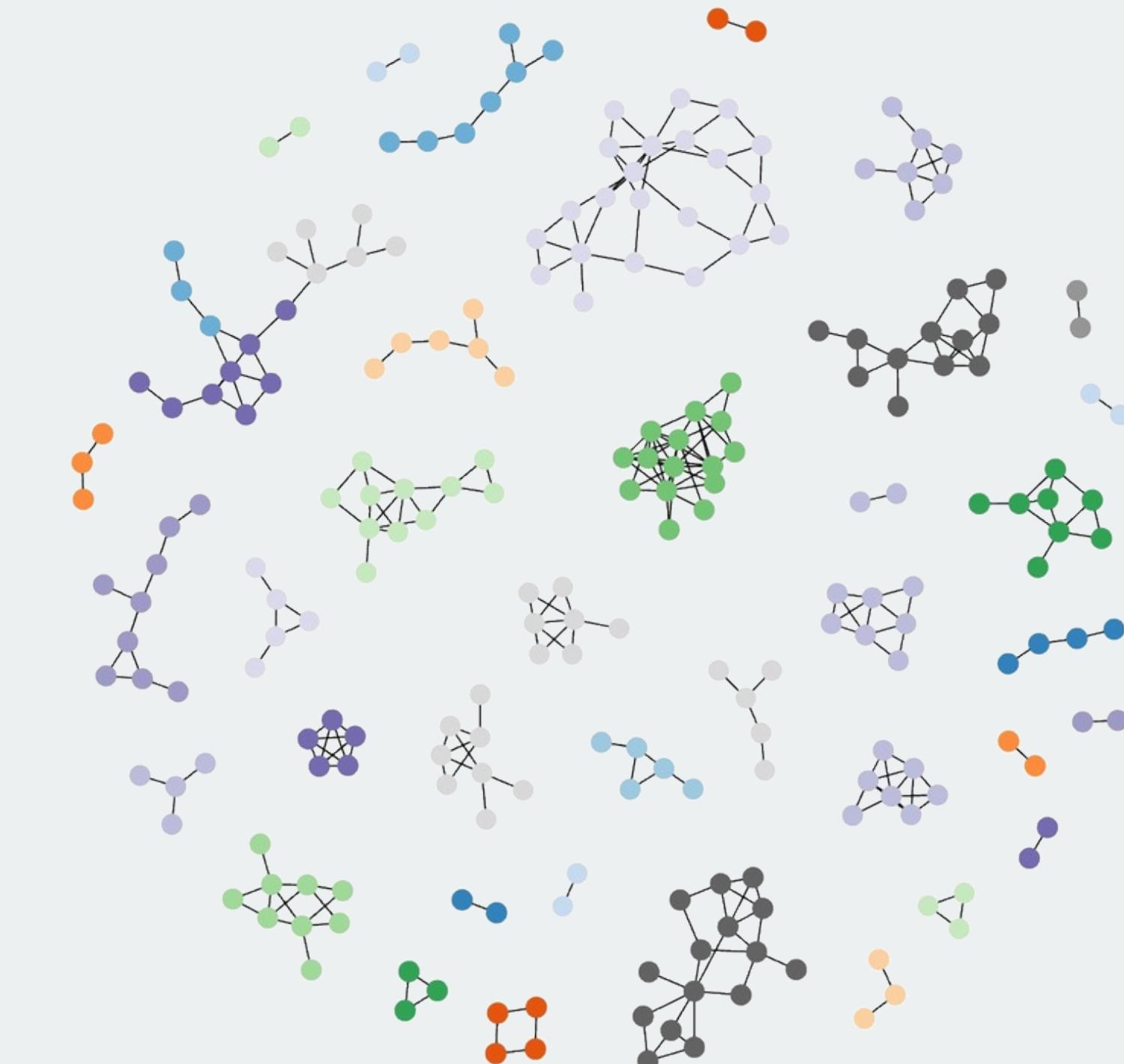


Aim: Understand the paradigm, learn how (some of) it works

Course overview

Sessions

1. Coding with data in Python
2. A Data Scientist's most fundamental tools
3. Getting data—scraping and APIs
4. Machine learning 1
5. Machine learning 2
- 6. Networks**
7. Natural language processing
8. Crunching Big Data with MapReduce
9. Ethical and legal considerations in Big Data
10. Lab work on project report
11. Lab work on project report
12. Project presentations



Aim: Learn how to describe and visualize complex data as a network

Course overview

Sessions

1. Coding with data in Python
2. A Data Scientist's most fundamental tools
3. Getting data—scraping and APIs
4. Machine learning 1
5. Machine learning 2
6. Networks
7. **Natural language processing**
8. Crunching Big Data with MapReduce
9. Ethical and legal considerations in Big Data
10. Lab work on project report
11. Lab work on project report
12. Project presentations

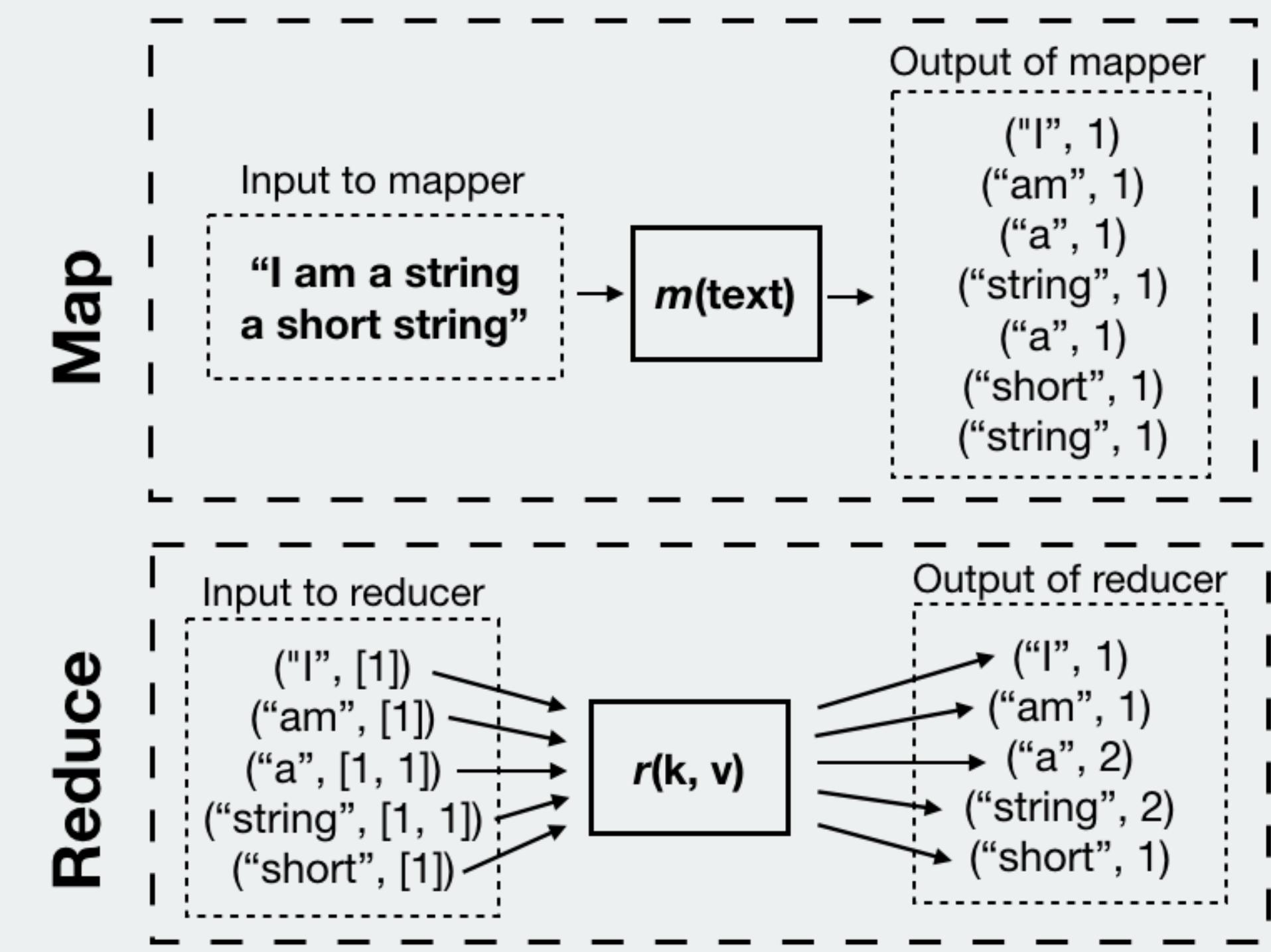


Aim: Get introduced to the (huge) field of NLP, and learn a few skills

Course overview

Sessions

1. Coding with data in Python
2. A Data Scientist's most fundamental tools
3. Getting data—scraping and APIs
4. Machine learning 1
5. Machine learning 2
6. Networks
7. Natural language processing
- 8. Crunching Big Data with MapReduce**
9. Ethical and legal considerations in Big Data
10. Lab work on project report
11. Lab work on project report
12. Project presentations

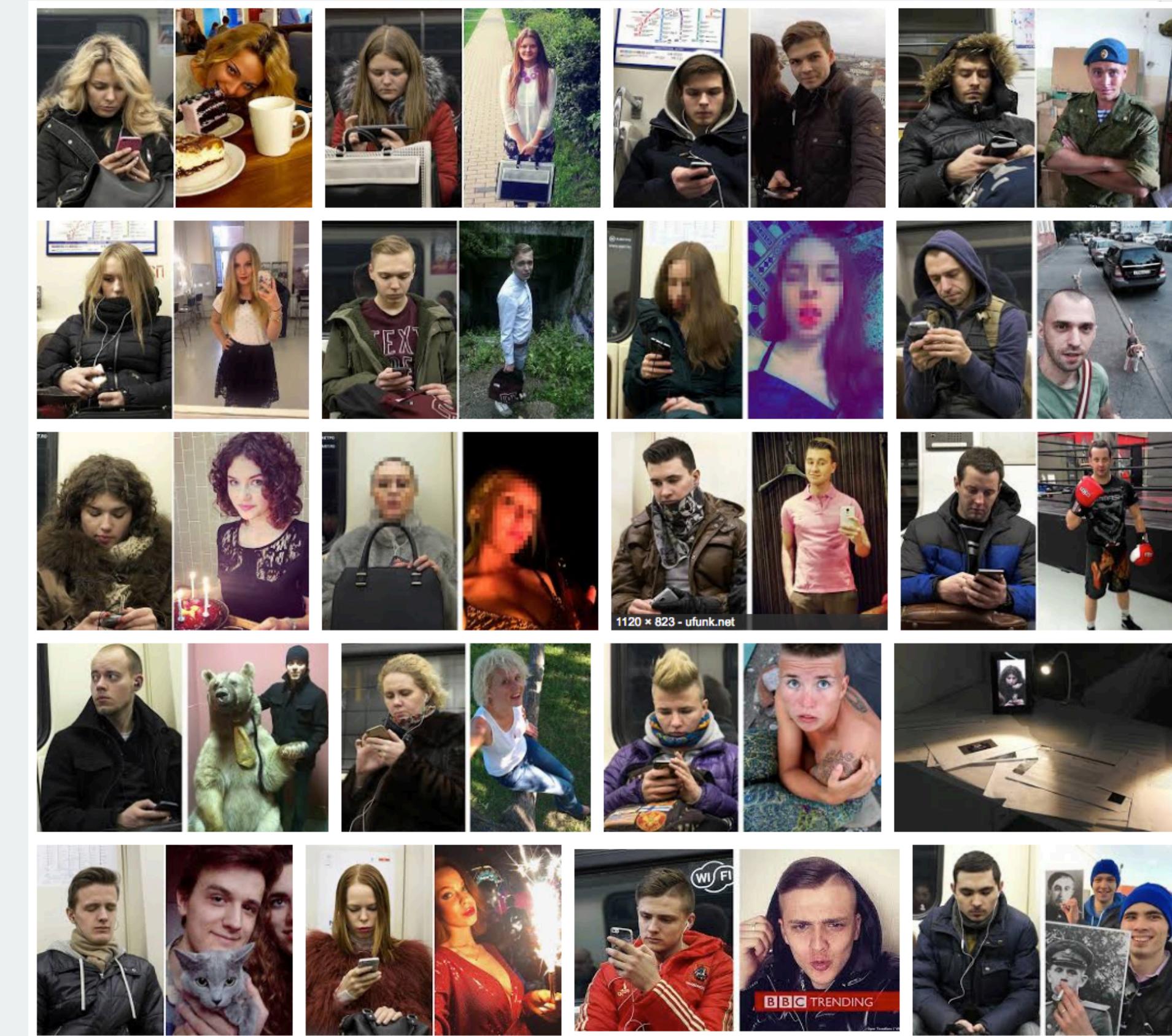


Aim: Process a massive dataset with an awesome method

Course overview

Sessions

1. Coding with data in Python
2. A Data Scientist's most fundamental tools
3. Getting data—scraping and APIs
4. Machine learning 1
5. Machine learning 2
6. Networks
7. Natural language processing
8. Crunching Big Data with MapReduce
9. **Ethical and legal considerations in Big Data**
10. Lab work on project report
11. Lab work on project report
12. Project presentations

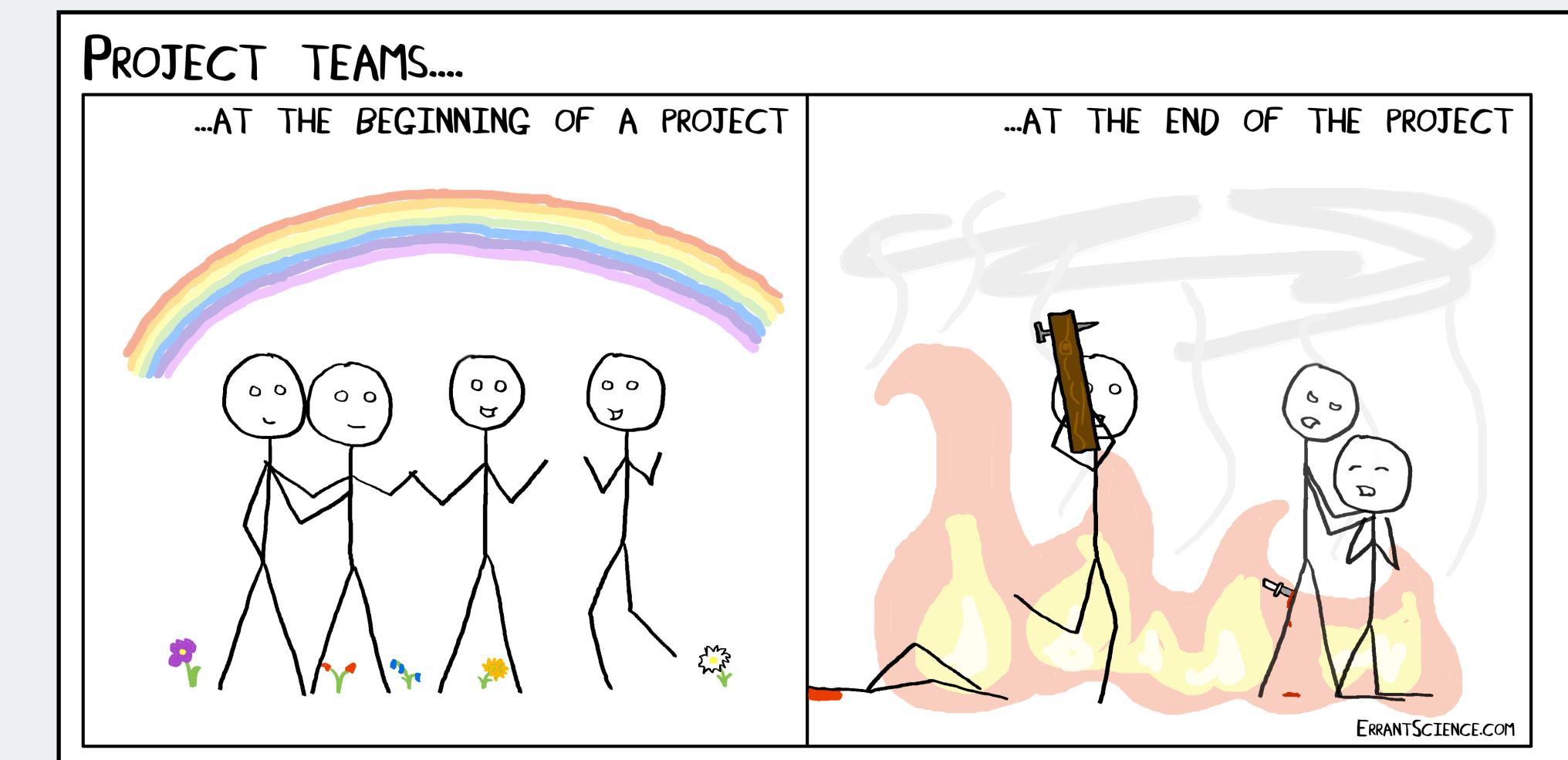


Aim: Reflect on some of the issues that powerful algorithms produce

Course overview

Sessions

1. Coding with data in Python
2. A Data Scientist's most fundamental tools
3. Getting data—scraping and APIs
4. Machine learning 1
5. Machine learning 2
6. Networks
7. Natural language processing
8. Crunching Big Data with MapReduce
9. Ethical and legal considerations in Big Data
- 10. Lab work on project report**
- 11. Lab work on project report**
- 12. Project presentations**



Aim: Synthesize all the thing you have learned
and make a project for your portfolio

What will you learn?

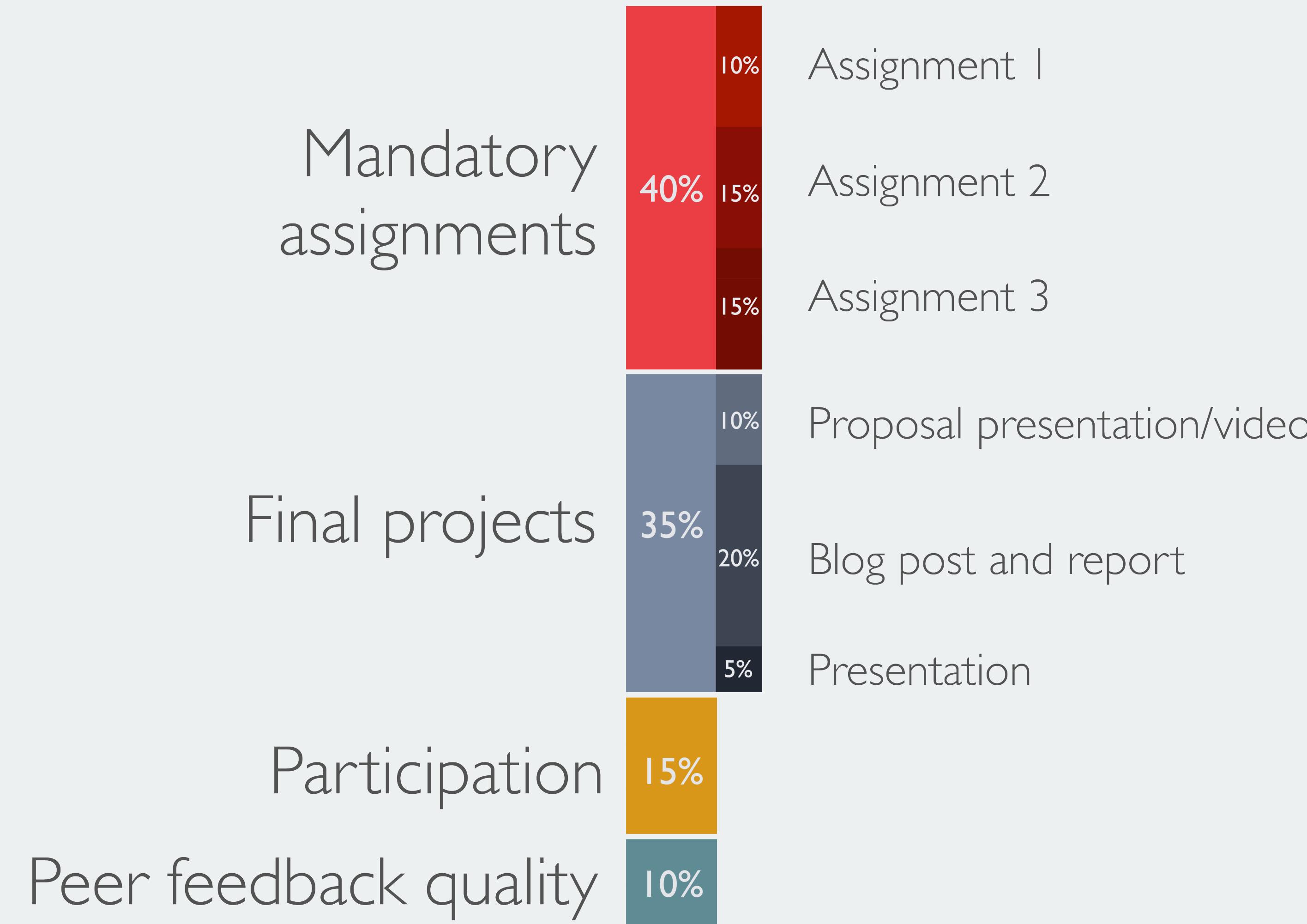
Knowledge and competences

- Get to know the landscape of problems and tools in Data Science
- Learn what Big Data is in this context
- Know where to start when you want to analyze something that requires lots of data

Concrete skills

- Juggle data in Python
- Predict outcomes from input data (machine learning)
- Visualize complex data
- Analyze Big Data using parallel methods (MapReduce)

How will you be graded?



How assignments work

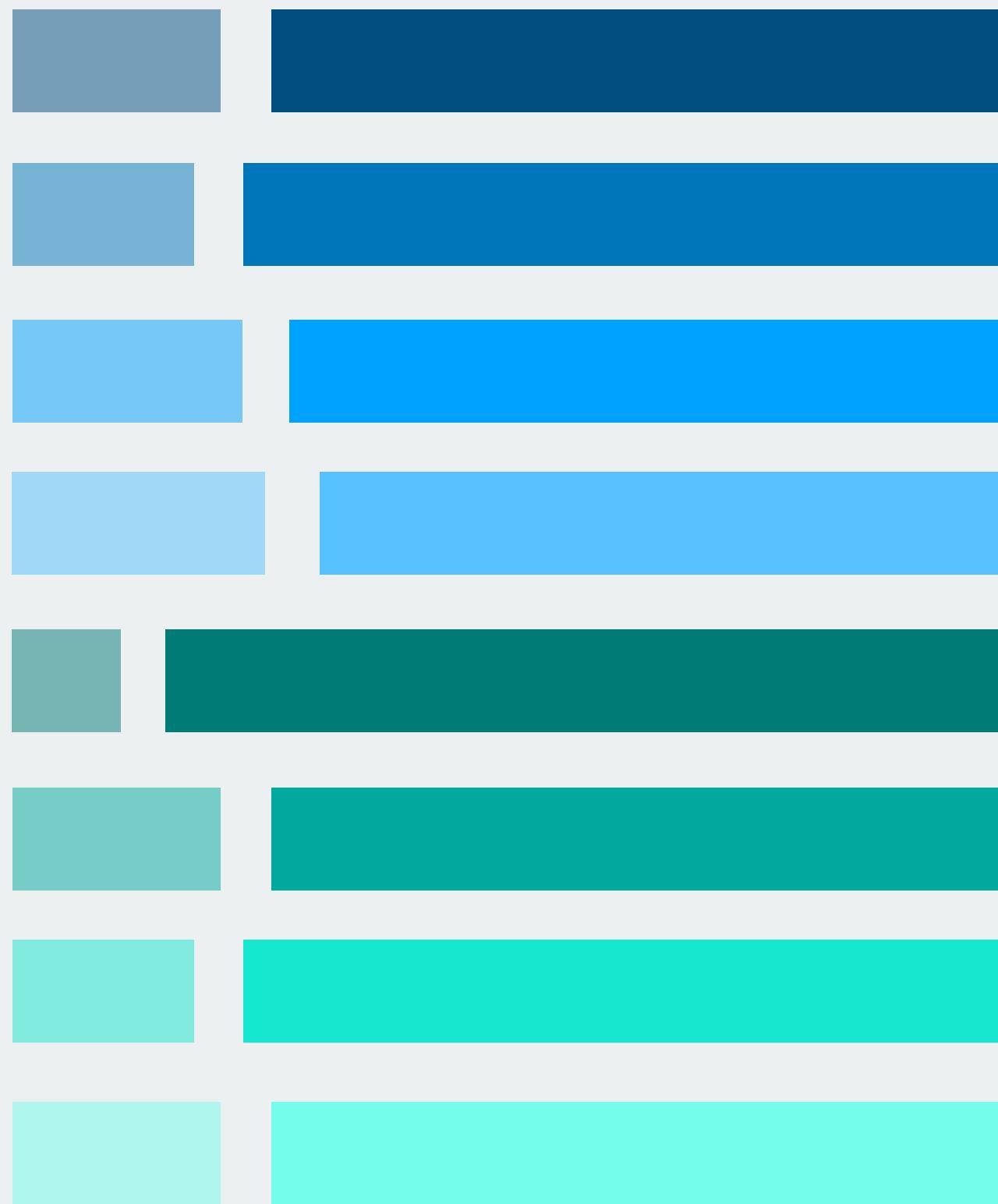
Sessions

Talk

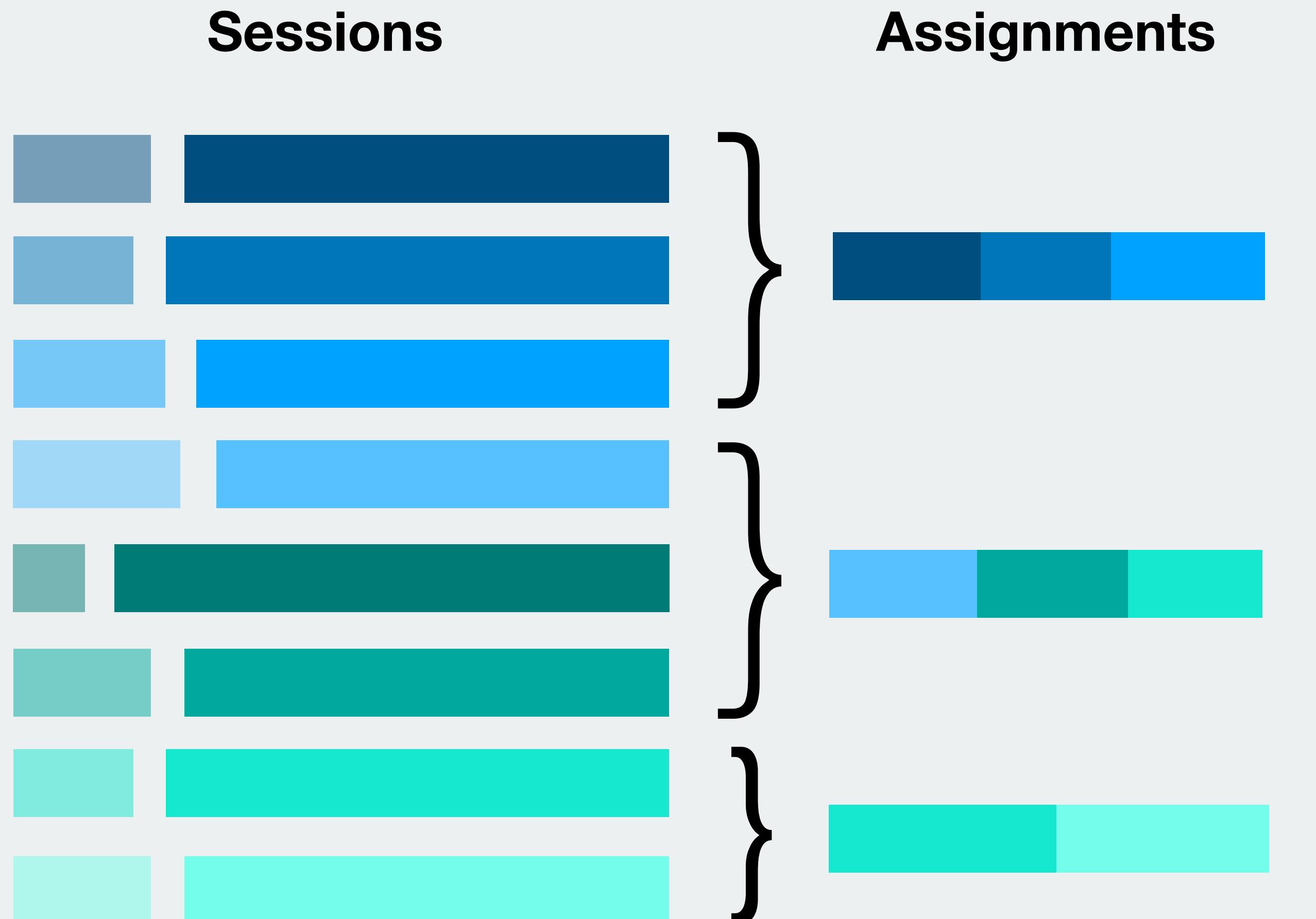
Solve exercises

How assignments work

Sessions



How assignments work



How assignments work



Timeline

- Assignment 1
 - *Posted:* Tuesday September 17 after class
 - *Handin due:* Monday September 23 23:59
 - *Peergrading due:* Friday September 27 23:59
 - *Peergrading feedback due:* .. Sunday September 29 23:59
- Assignment 2
 - *Posted:* Tuesday October 15 after class
 - *Handin due:* Monday October 21 23:59
 - *Peergrading due:* Friday October 25 23:59
 - *Peergrading feedback due:* .. Sunday October 27 23:59
- Assignment 3
 - *Posted:* Tuesday November 5 after class
 - *Handin due:* Monday November 11 23:59
 - *Peergrading due:* Friday November 15 23:59
 - *Peergrading feedback due:* .. Sunday November 17 23:59

The final project

I validate your
project idea

You present a
proposal video

Deliverable

You deliver a blog
post and your code

Deliverable

You give a
presentation

Previous students projects

"What makes us happy?"

"The secrets hidden in your Venmo and Instagram data"

"Pizza Makes the World go Round"

How to do well in this course

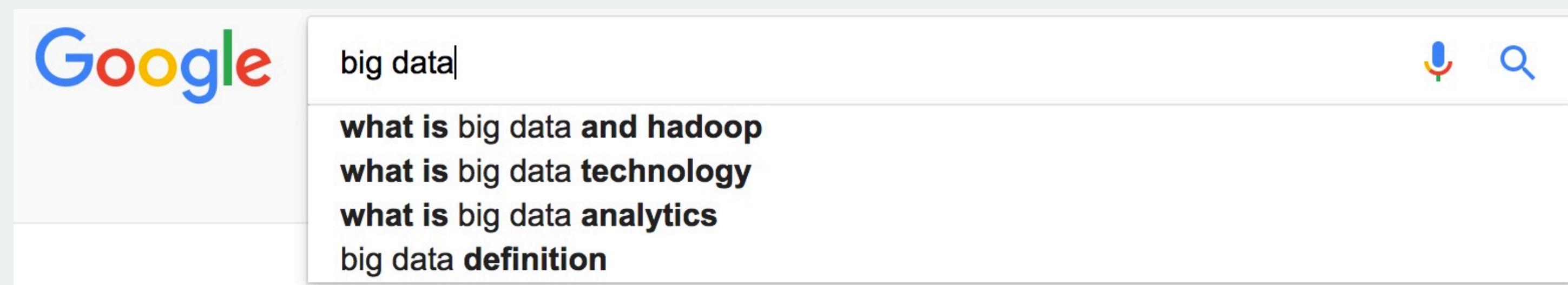
Best strategy:

1. Complete the *preparation goals* for each session (see wiki)
2. Be inquisitive. Ask lots of questions to your neighbors and me, and up your googling-game

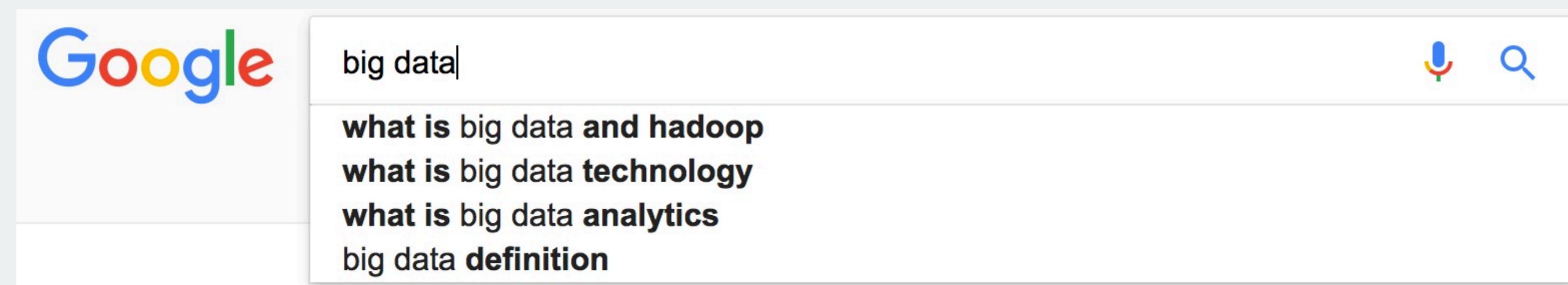
Everything else is on the wiki on Canvas

What is Big Data?

What is Big Data?



What is Big Data?



<http://www.internetlivestats.com/>

What is Big Data?

big data

noun COMPUTING

extremely large data sets that may be analysed computationally to reveal patterns, trends, and associations, especially relating to human behaviour and interactions.
"much IT investment is going towards managing and maintaining big data"

 Translations, word origin, and more definitions

What is Big Data?

big data

noun COMPUTING

extremely large data sets that may be analysed computationally to reveal patterns, trends, and associations, especially relating to human behaviour and interactions.
"much IT investment is going towards managing and maintaining big data"

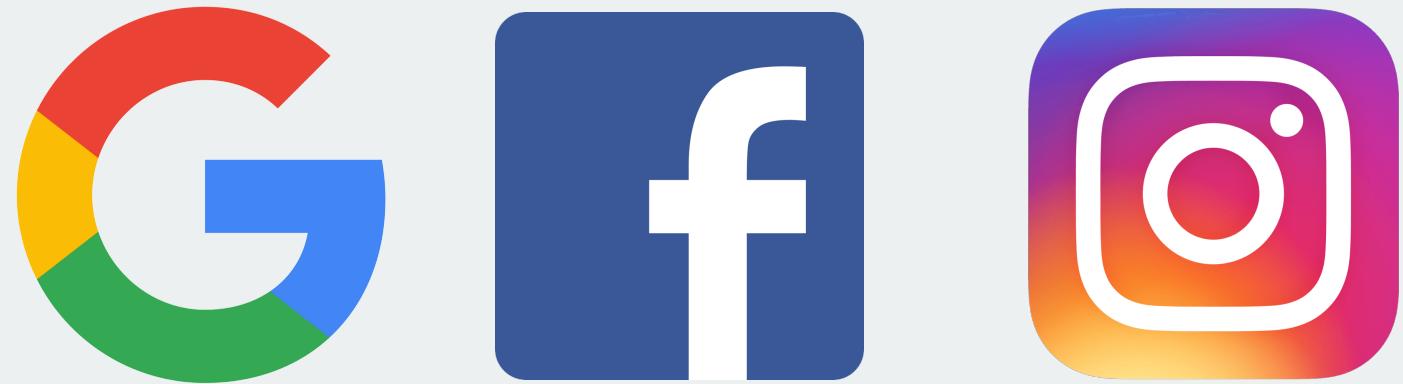
 Translations, word origin, and more definitions

In this course: **Datasets with potential to reveal interesting insight**

Where and for what is Big Data being used?

Where and for what is Big Data being used?

Ads



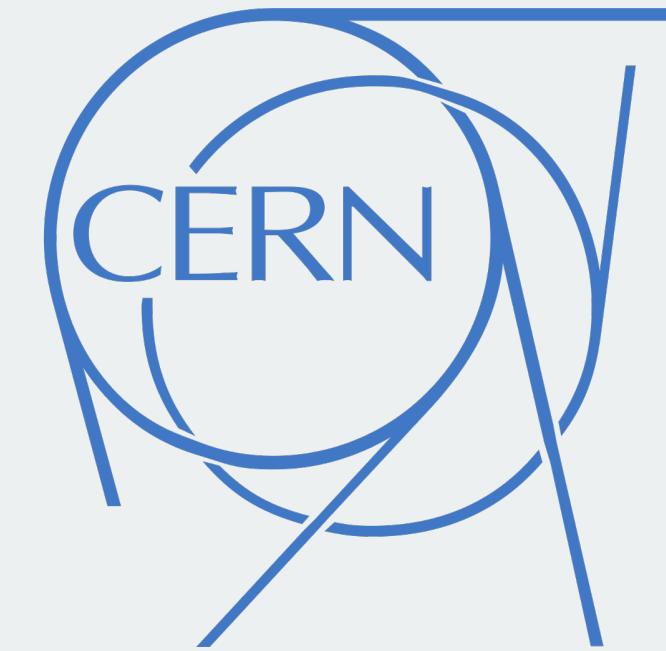
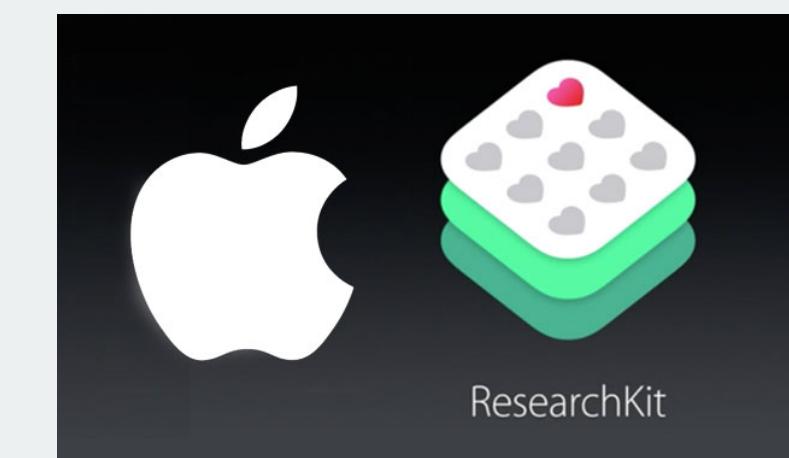
 **theTradeDesk[®]**
Omnicom WPP

Ted Talk: “Is Big Data Killing Creativity?”

User customization

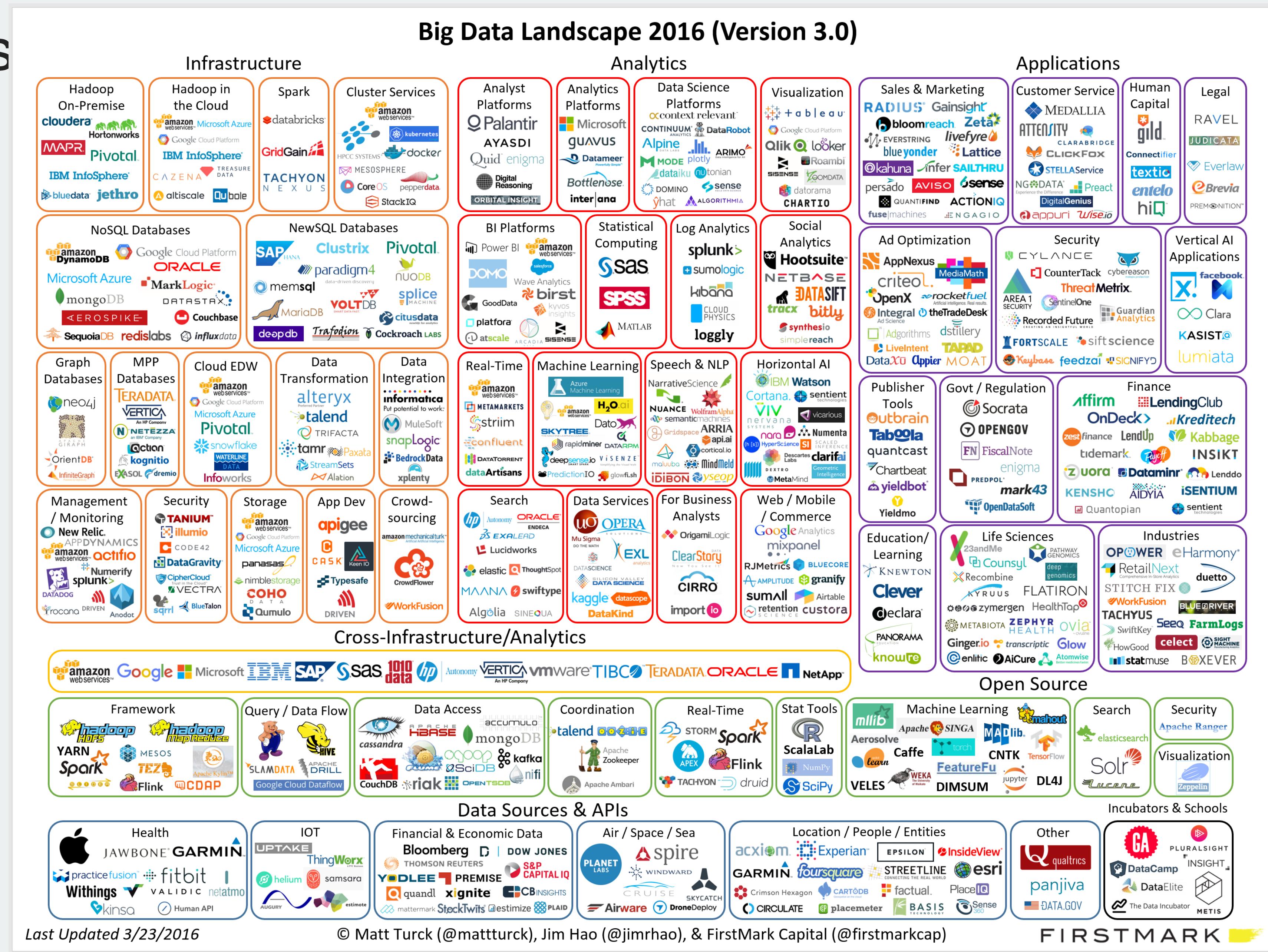


Research



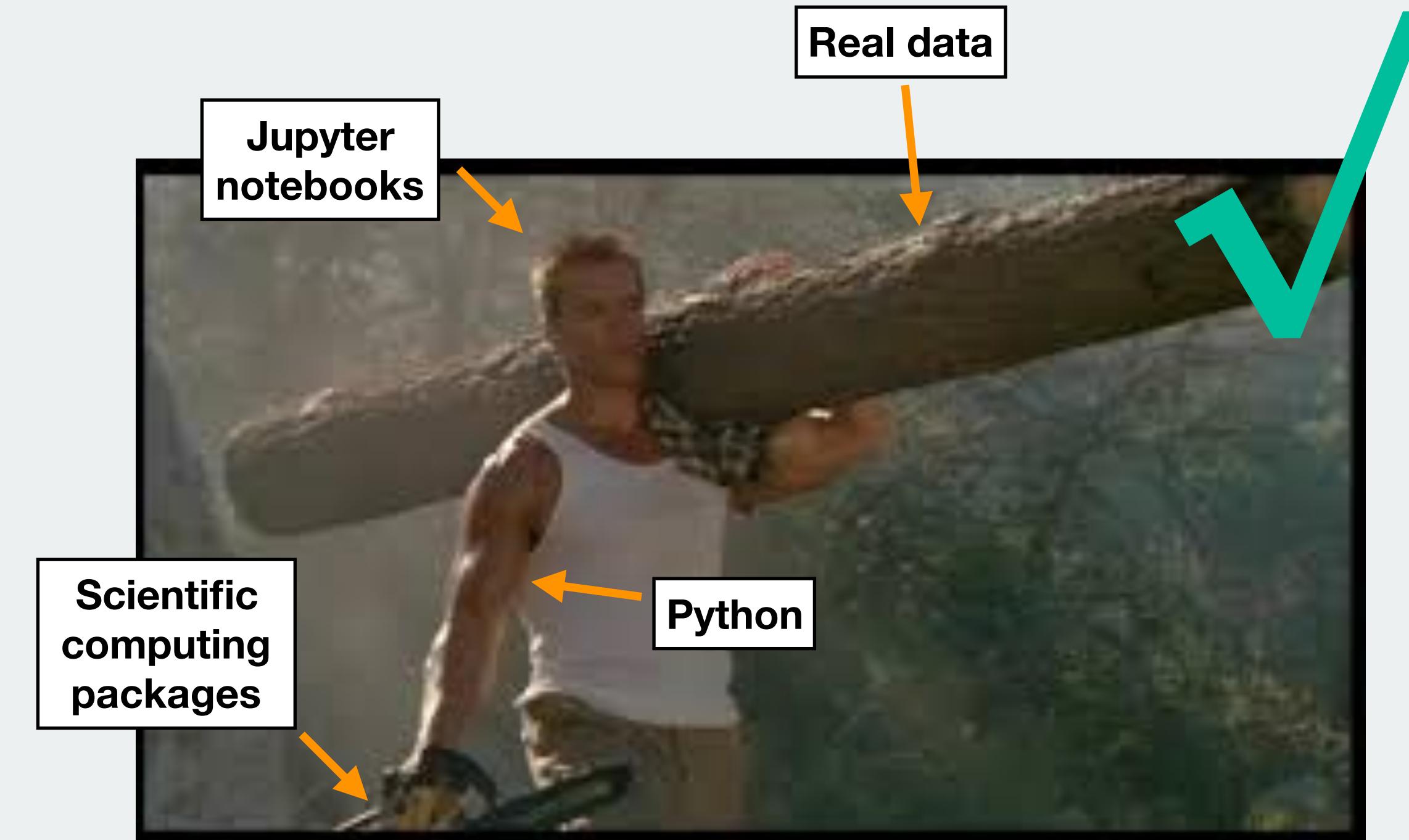
How does one work with Big Data?

How does



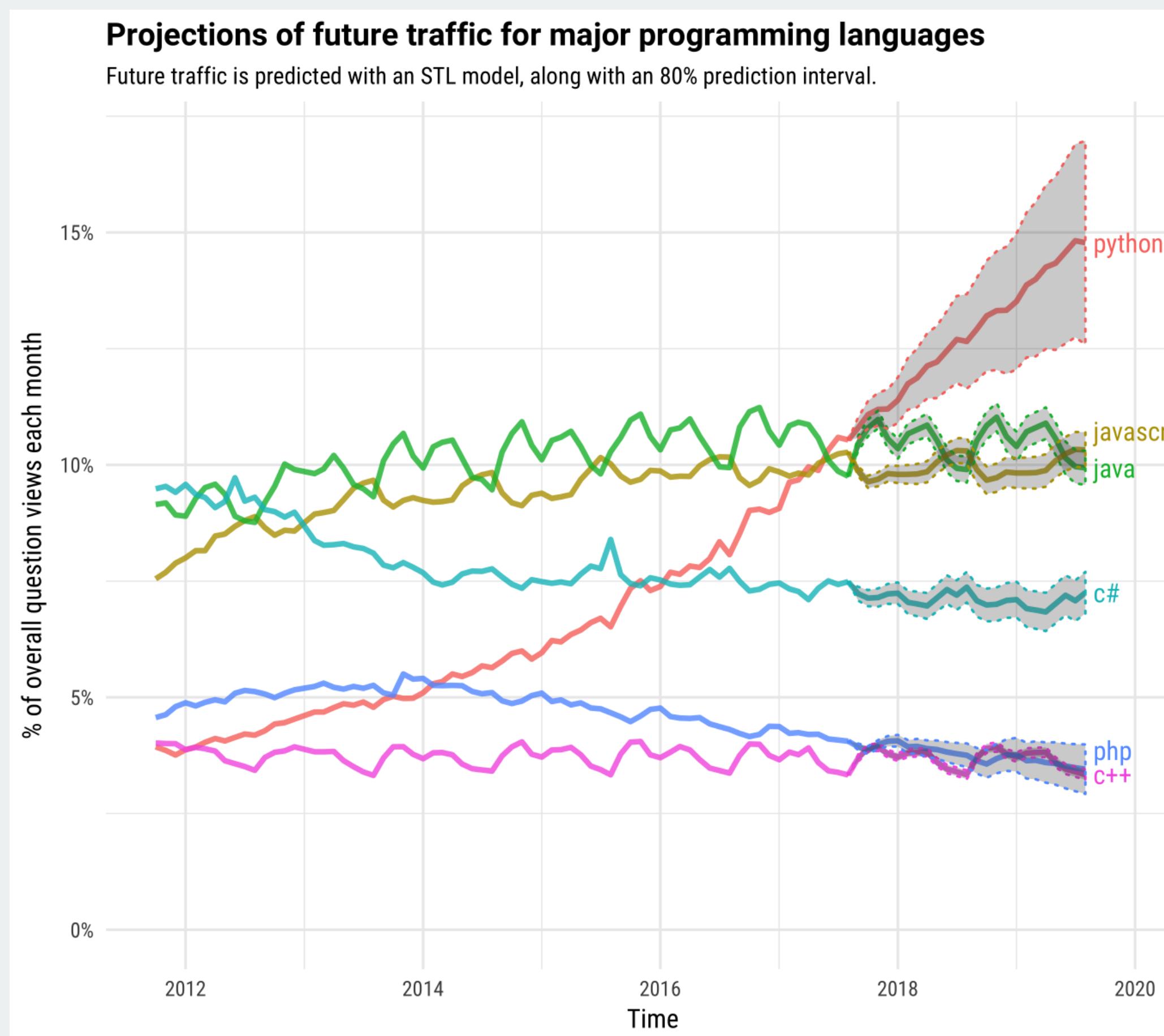
How does one work with Big Data?

In this course: Everything from scratch



What is Python?

What is Python?

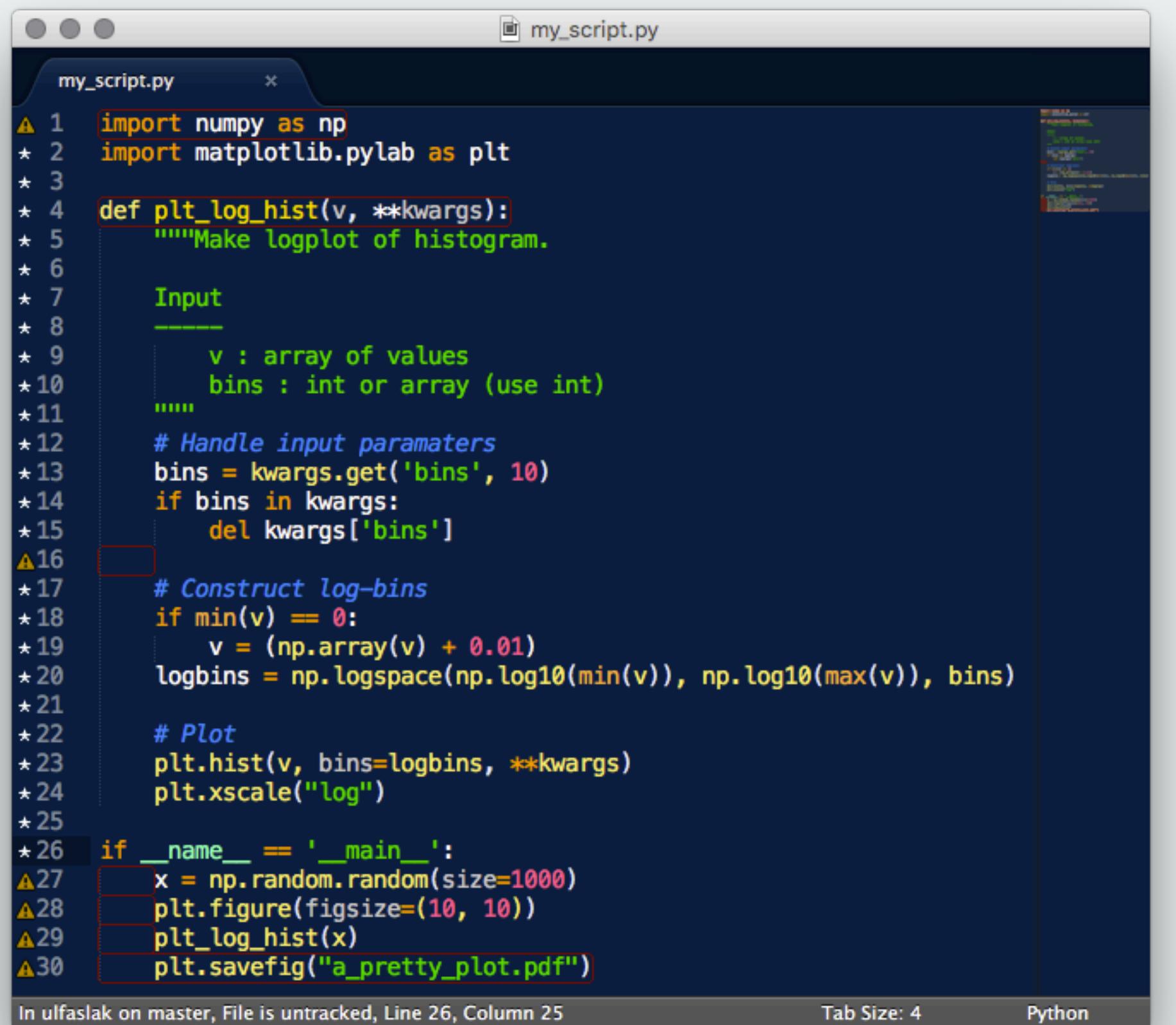


<https://stackoverflow.blog/2017/09/06/incredible-growth-python/>

What is Python?

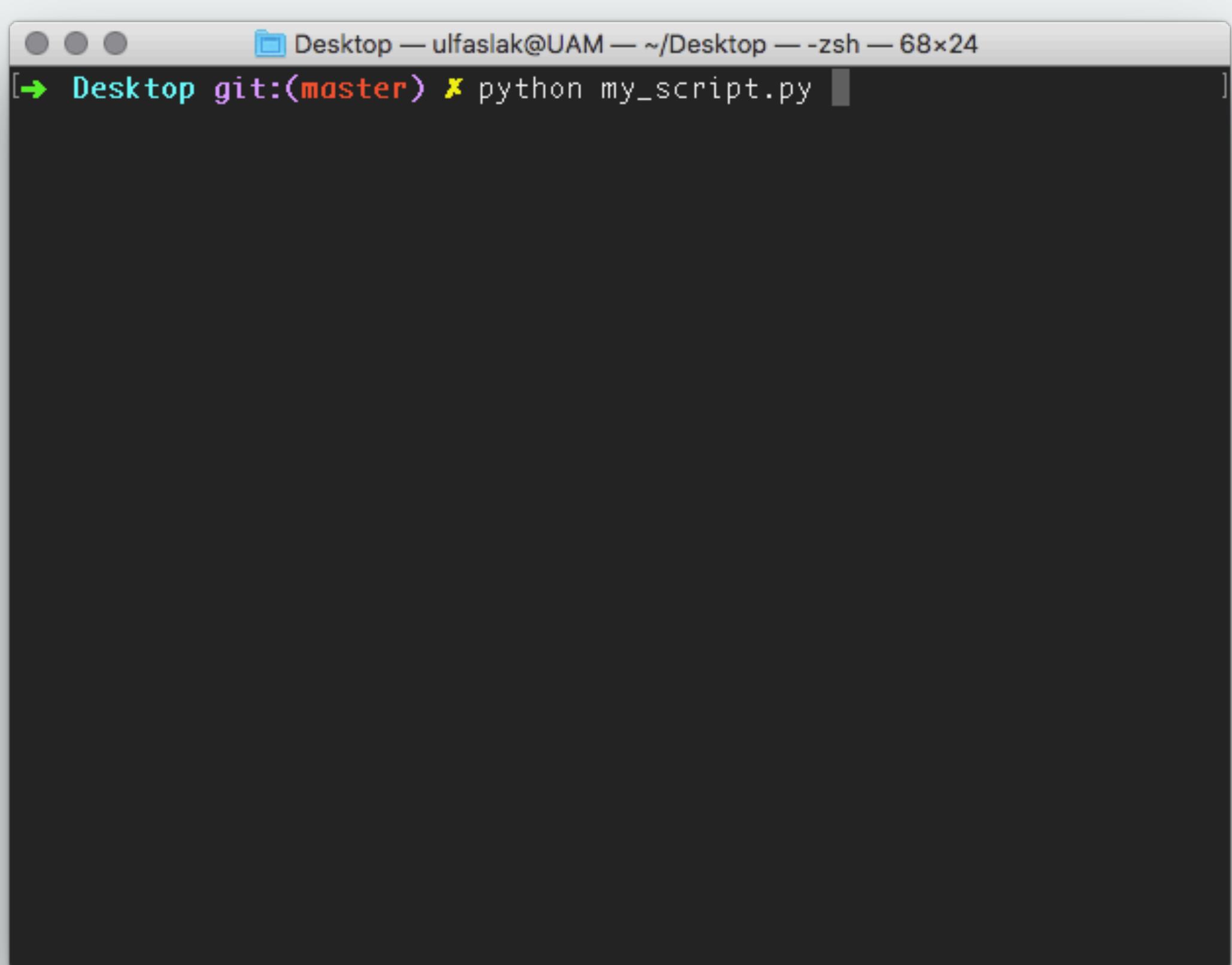
- A programming language: Interpreted, high-level, general-purpose. Created by Guido van Rossum.
- High and growing popularity in both industry and research.
- Simple syntax, easy to learn
- Supports multiple paradigms: object-oriented, functional, imperative, procedural.
- Fast: bindings to C
- Open source. Has packages for almost everything.

Using Python with scripts



```
my_script.py
my_script.py  *
1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 def plt_log_hist(v, **kwargs):
5     """Make logplot of histogram.
6
7     Input
8     -----
9         v : array of values
10        bins : int or array (use int)
11    """
12    # Handle input parameters
13    bins = kwargs.get('bins', 10)
14    if bins in kwargs:
15        del kwargs['bins']
16
17    # Construct log-bins
18    if min(v) == 0:
19        v = (np.array(v) + 0.01)
20    logbins = np.logspace(np.log10(min(v)), np.log10(max(v)), bins)
21
22    # Plot
23    plt.hist(v, bins=logbins, **kwargs)
24    plt.xscale("log")
25
26 if __name__ == '__main__':
27     x = np.random.random(size=1000)
28     plt.figure(figsize=(10, 10))
29     plt_log_hist(x)
30     plt.savefig("a_pretty_plot.pdf")
```

In ulfaslak on master, File is untracked, Line 26, Column 25 Tab Size: 4 Python



```
[Desktop git:(master)* python my_script.py]
[Desktop git:(master)* python my_script.py]
```

Using Python with Jupyter notebooks

The screenshot shows a Jupyter Notebook interface running in a web browser. The notebook has two cells:

In [32]:

```
#matplotlib inline
import numpy as np
import matplotlib.pylab as plt

def plt_log_hist(v, **kwargs):
    """Make logplot of histogram.

    Input
    -----
    v : array of values
    bins : int or array (use int)
    """
    # Handle input parameters
    bins = kwargs.get('bins', 10)
    if bins in kwargs:
        del kwargs['bins']

    # Construct log-bins
    if min(v) == 0:
        v = (np.array(v) + 0.01)
    logbins = np.logspace(np.log10(min(v)), np.log10(max(v)), bins)

    # Plot
    plt.hist(v, bins=logbins, **kwargs)
    plt.xscale("log")
```

Last executed 2018-01-18 12:24:53 in 11ms

In [33]:

```
x = np.random.random(size=1000)
plt.figure(figsize=(5, 5))
plt_log_hist(x)
plt.show()
```

Last executed 2018-01-18 12:24:56 in 693ms

A histogram plot is displayed below the second cell, showing a distribution of values on a logarithmic x-axis ranging from 10^{-3} to 10^0 . The y-axis ranges from 0 to 600. The distribution is highly right-skewed, with most values between 10^{-1} and 10^0 .

Let's get started!

1. If you haven't already: download and install Anaconda (Python 3.7 version) <https://www.continuum.io/downloads>.
2. Make a folder for this course. Open a terminal (Mac) or a console (PC) and navigate to that folder.
3. Run the following command in your terminal/console: git clone [https://github.com/ulfaslak/
caobd_f19](https://github.com/ulfaslak/caobd_f19)
4. In your terminal/console, navigate into the exercises folder inside of the newly created **caobd_f19** directory, and run the following command: jupyter notebook exercises_week1.ipynb.