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

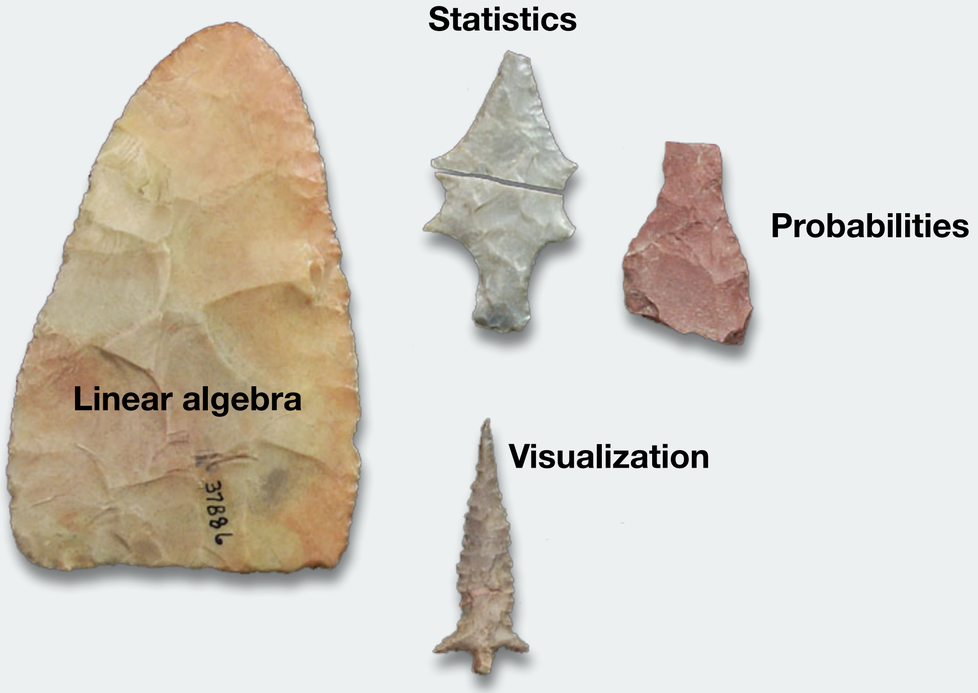
```python
def randomize_by_edge_swaps(G, num_iterations):
    """Randomizes the graph by swapping edges in such a way that
    preserves the degree distribution of the original graph.

    Source: https://gist.github.com/gotgenes/2770023
    """
    newgraph = G.copy()
    edge_list = newgraph.edges()
    num_edges = len(edge_list)
    total_iterations = num_edges * num_iterations

    for i in xrange(total_iterations):
        rand_index1 = int(round(random.random() * (num_edges - 1)))
        rand_index2 = int(round(random.random() * (num_edges - 1)))
        original_edge1 = edge_list[rand_index1]
        original_edge2 = edge_list[rand_index2]
        head1, tail1 = original_edge1
        head2, tail2 = original_edge2

        # Flip a coin to see if we should swap head1 and tail1 for
        # the connections
        if random.random() >= 0.5:
            head1, tail1 = tail1, head1

        if head1 == tail2 or head2 == tail1:
            continue

        if newgraph.has_edge(head1, tail2) or newgraph.has_edge(
                head2, tail1):
            continue

        # Suceeded checks, perform the swap
        original_edge1_data = newgraph[head1][tail1]
        original_edge2_data = newgraph[head2][tail2]

        newgraph.remove_edges_from((original_edge1, original_edge2))

        new_edge1 = (head1, tail2, original_edge1_data)
        new_edge2 = (head2, tail1, original_edge2_data)
        newgraph.add_edges_from((new_edge1, new_edge2))

        # Now update the entries at the indices randomly selected
        edge_list[rand_index1] = (head1, tail2)
        edge_list[rand_index2] = (head2, tail1)

    assert len(newgraph.edges()) == num_edges
    return newgraph
```

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

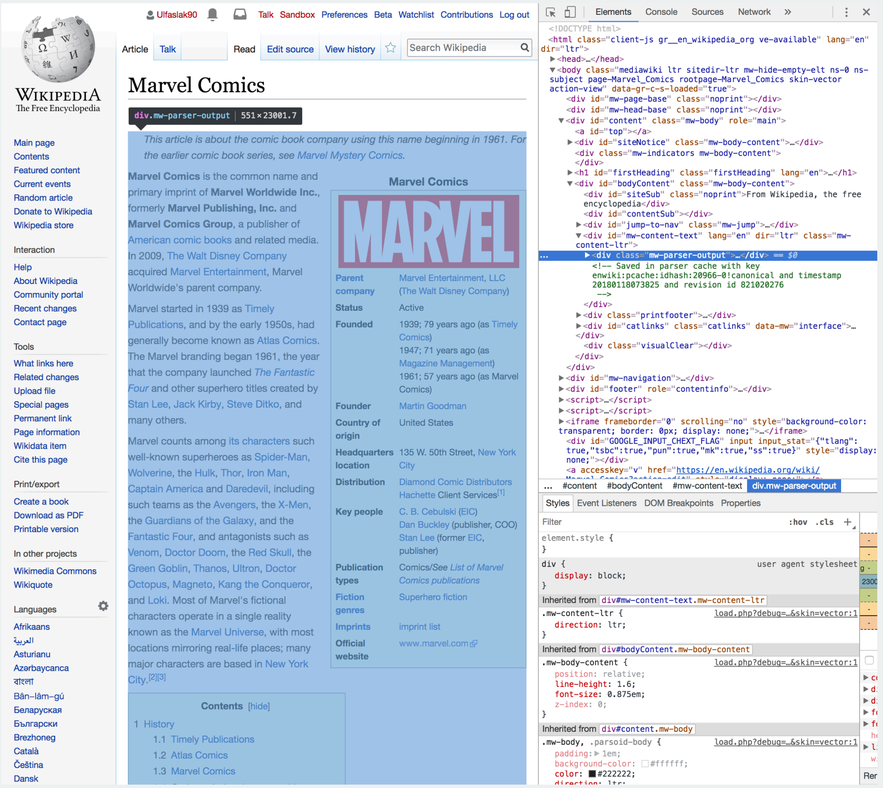


Aim: Refresh math skills and learn how to use them in Python

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

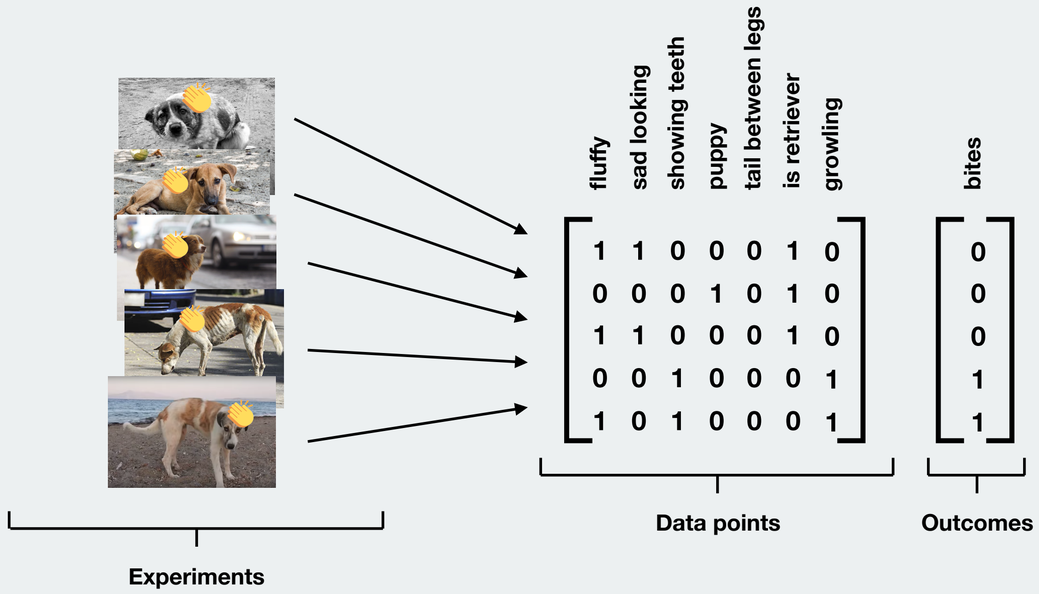


Aim: Learn how to get data from web and get some from the Wiki API

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

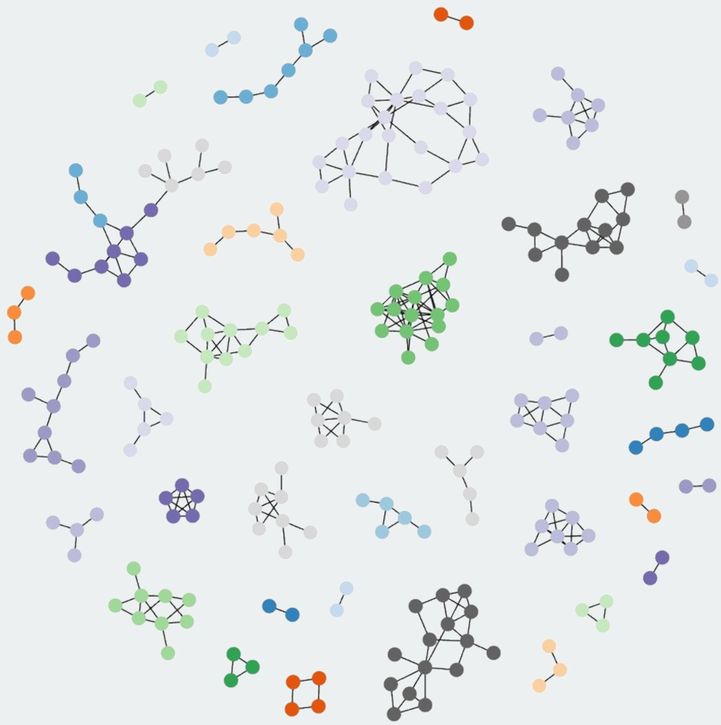


Aim: Understand the paradigm, learn how (some of) it works

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

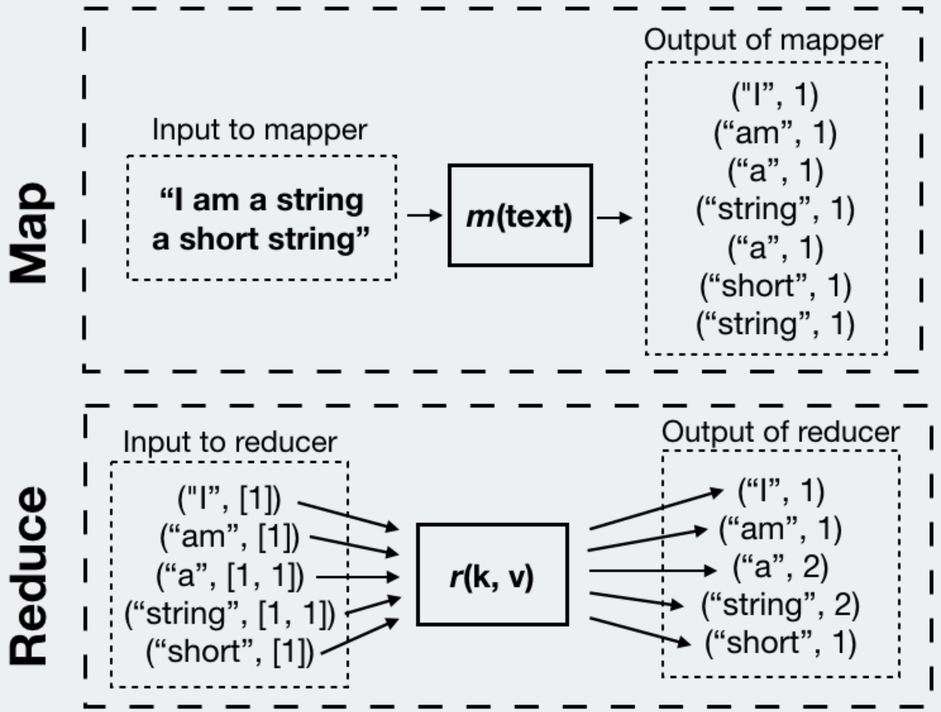
1. Coding with data in Python
2. A Data Scientist's most fundamental tools
3. Getting data—scraping and APIs
4. Machine learning 1
5. Machine learning 2
6. Networks
7. **Natural language processing**
8. Crunching Big Data with MapReduce
9. Ethical and legal considerations in Big Data
10. Lab work on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

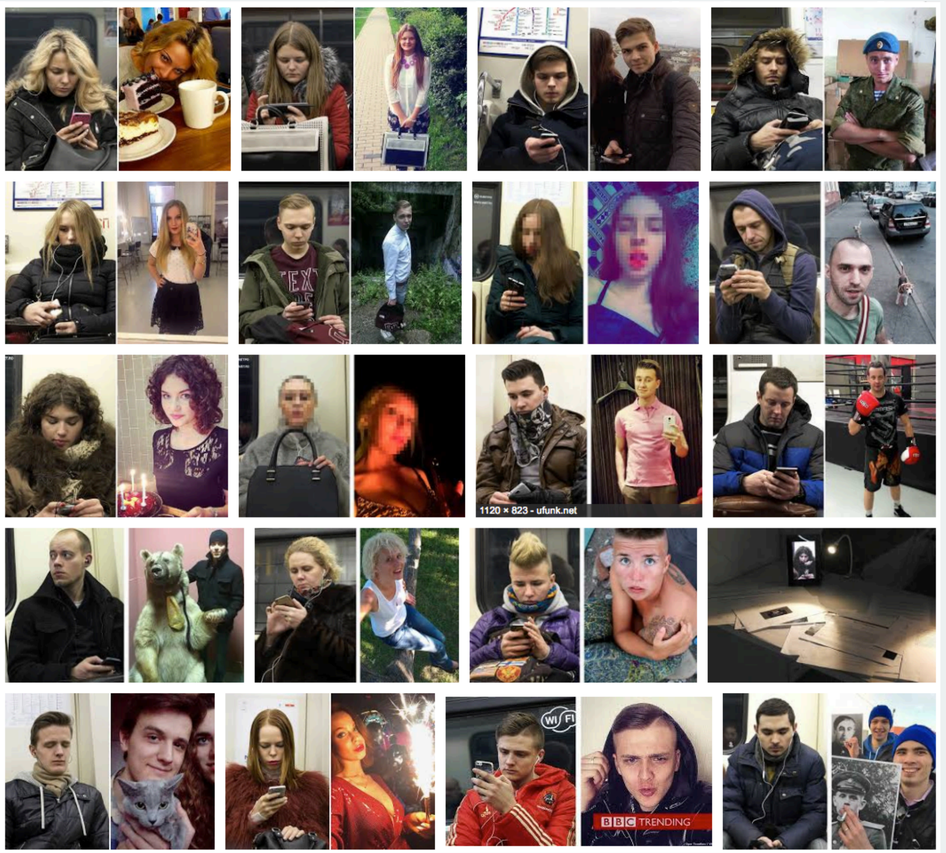


Aim: Process a massive dataset with an awesome method

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

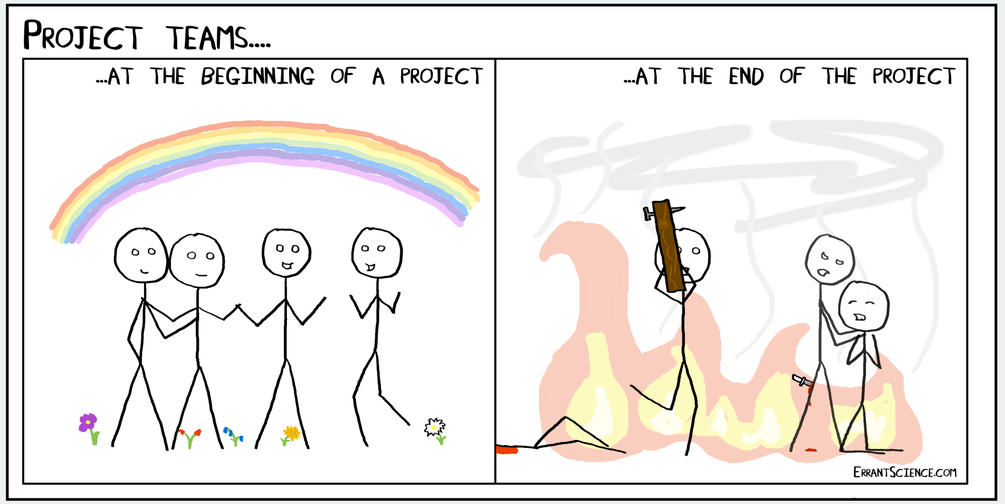


Aim: Reflect on some of the issues that powerful algorithms produce

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?

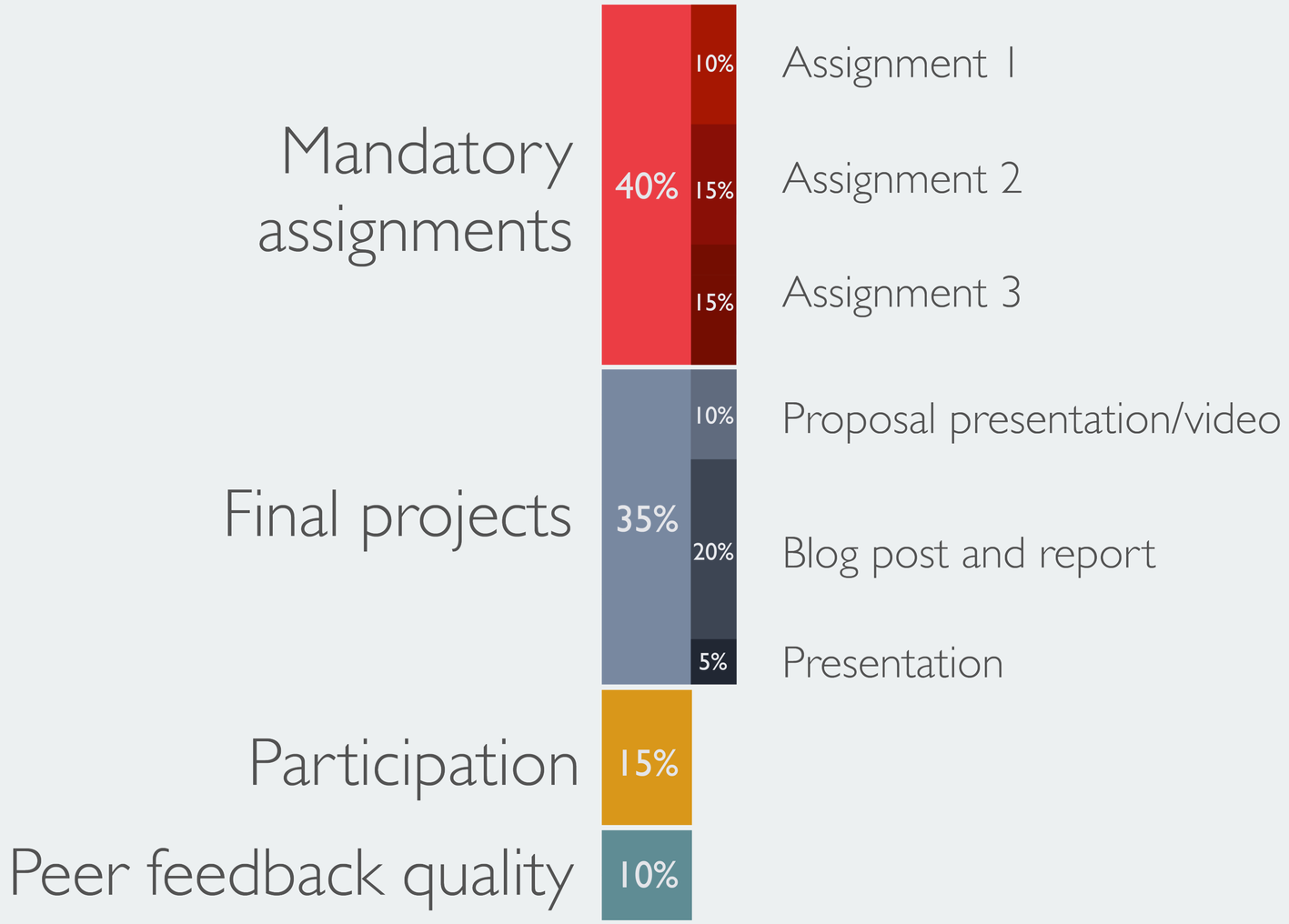
Knowledge and competences

- 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

- Practical data science (data munging, analysis, modeling)
- Predict outcomes from input data (machine learning)
- Visualize data
- Analyze massive data using state-of-the-art methods (MapReduce)

How will you be graded?

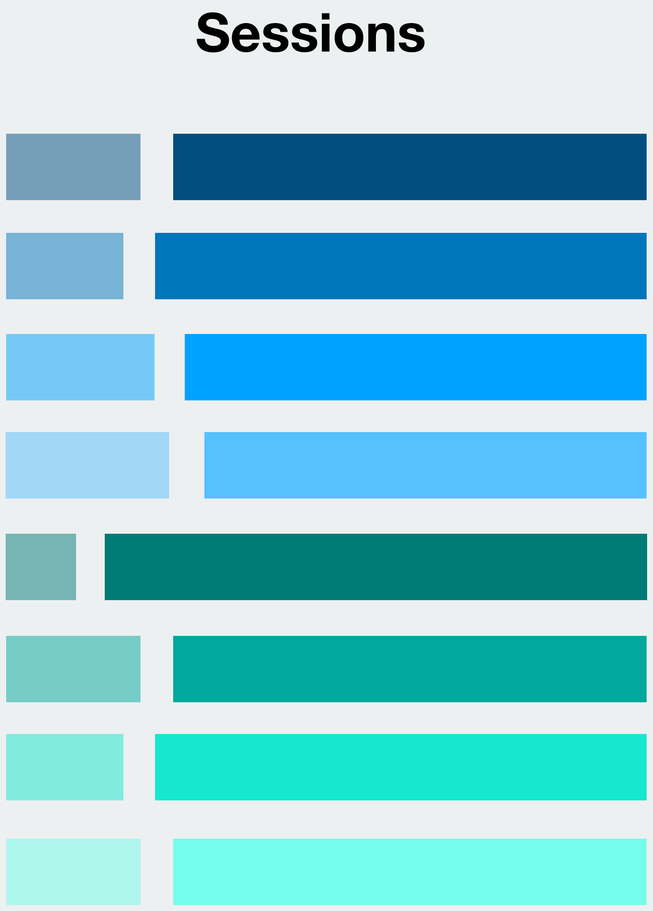


How assignments work

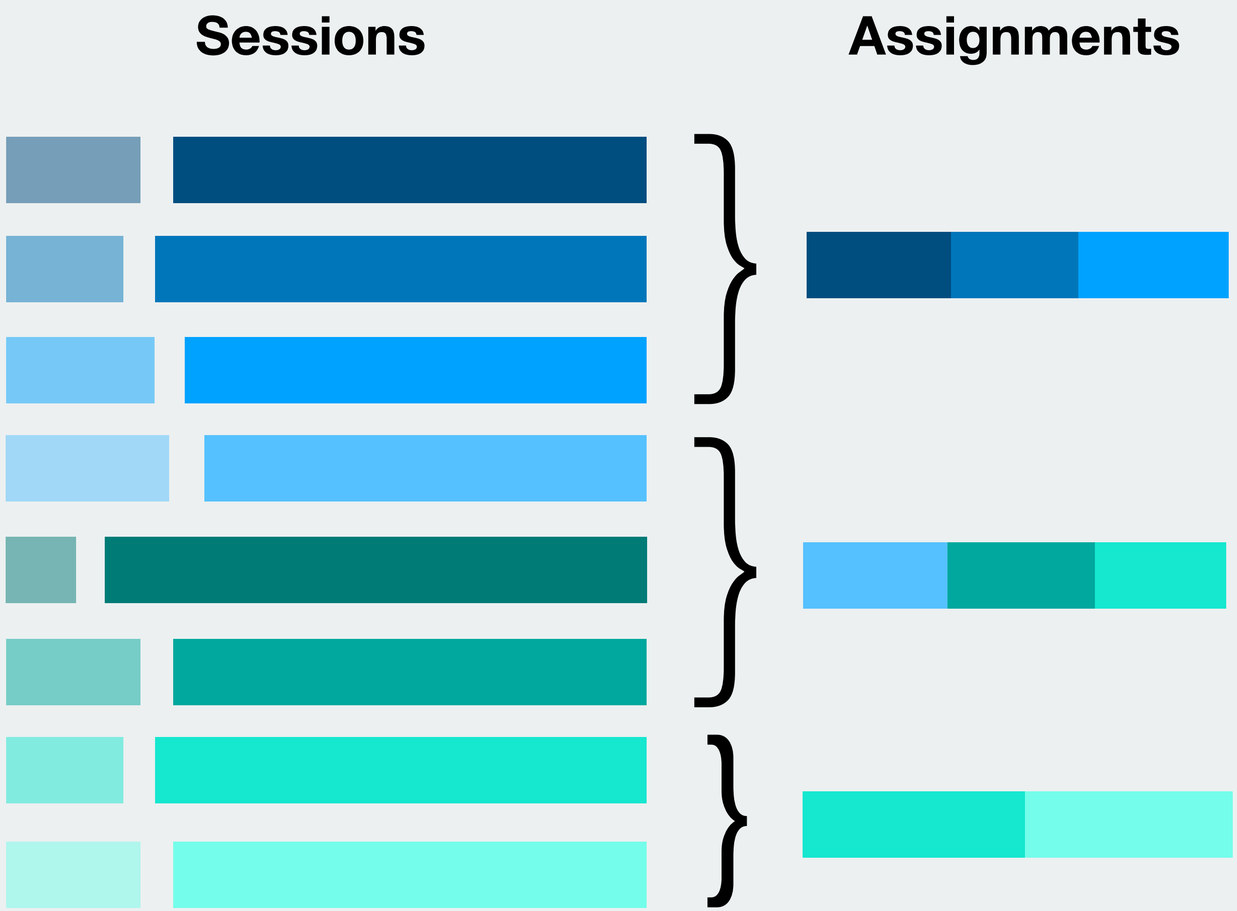
Sessions

Talk **Solve exercises**

How assignments work



How assignments work



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

A note on learning in general

To ways of learning:

1. *Social learning*: observe what others do and mimic

2. *Learning by doing*: make mistakes, ask others and fix mistakes

A note on learning in general

To ways of learning:

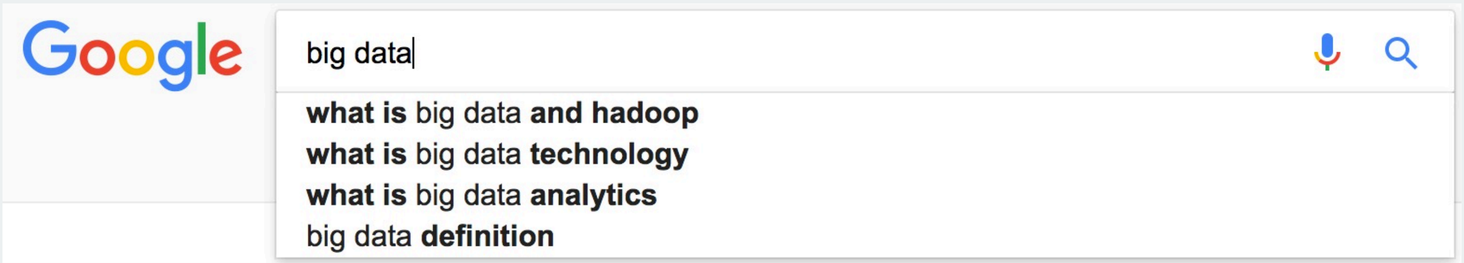
PAINLESS, SLOW 1. *Social learning*: observe what others do and mimic

PAINFUL, FAST 2. *Learning by doing*: make mistakes, ask others and fix mistakes

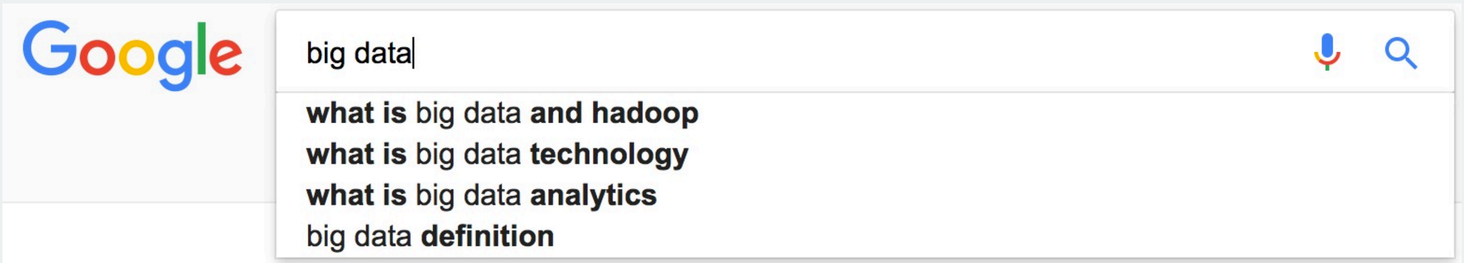
Everything else is on the wiki on Canvas

What is Big Data?

What is Big Data?

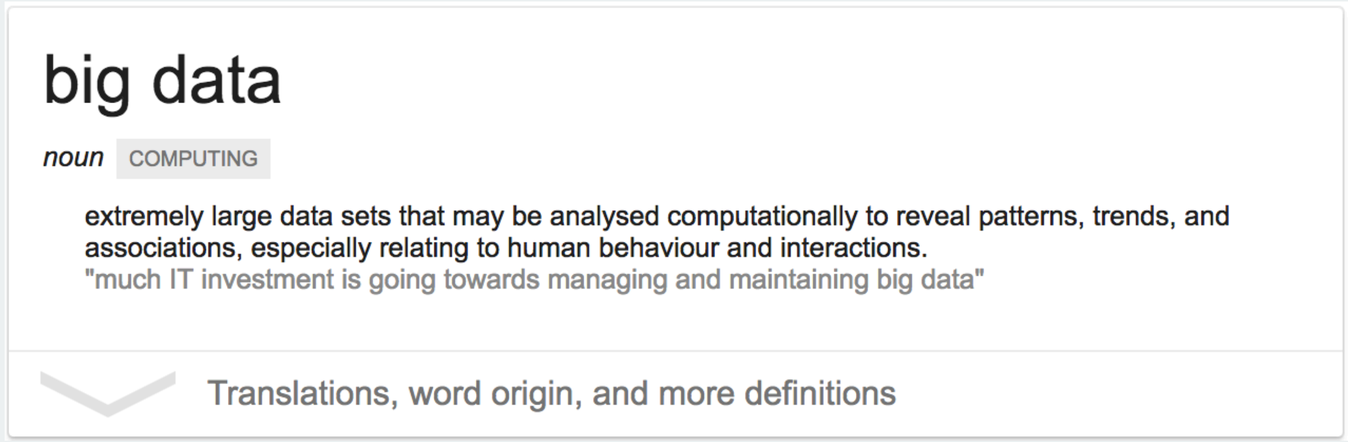


What is Big Data?

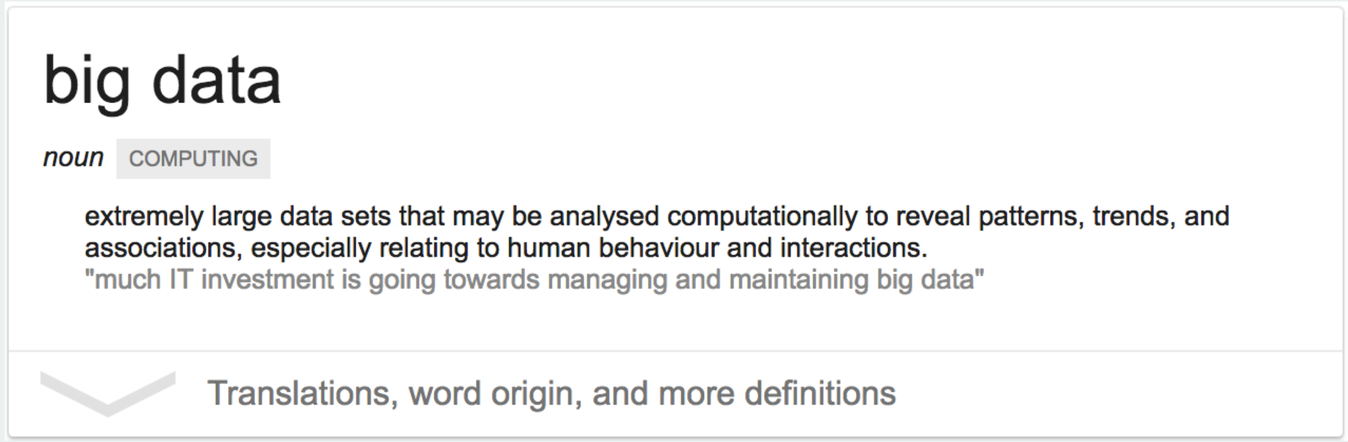


http://www.internetlivestats.com/

What is Big Data?



What is Big Data?



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

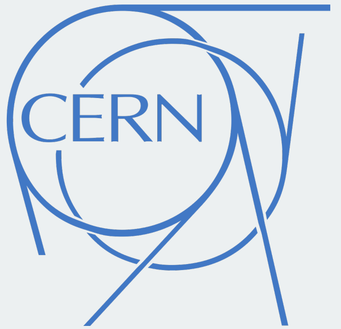
Omnicom WPP

User customization

YouTube amazon.com

NETFLIX

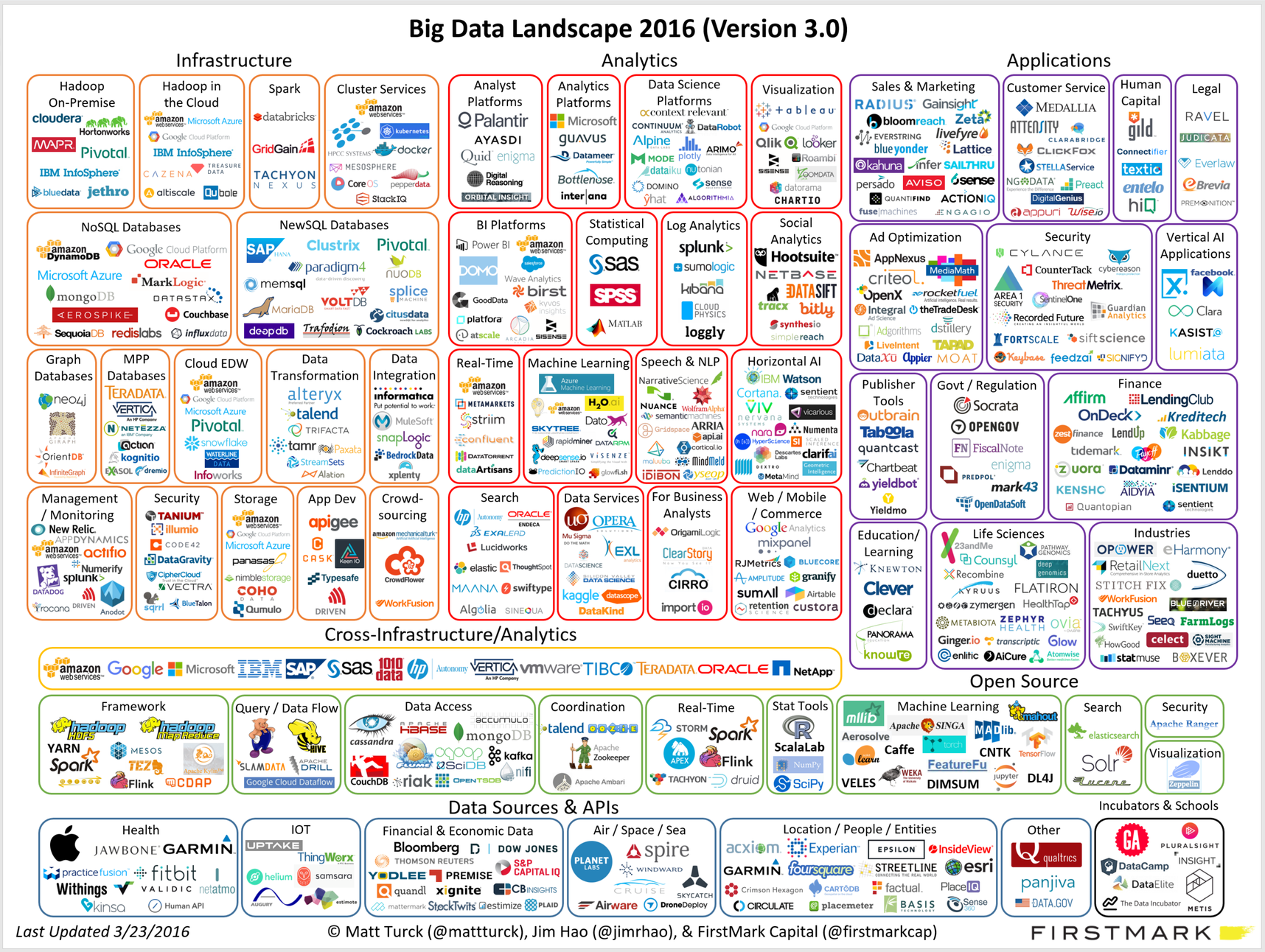
Research



Ted Talk: "Is Big Data Killing Creativity?"

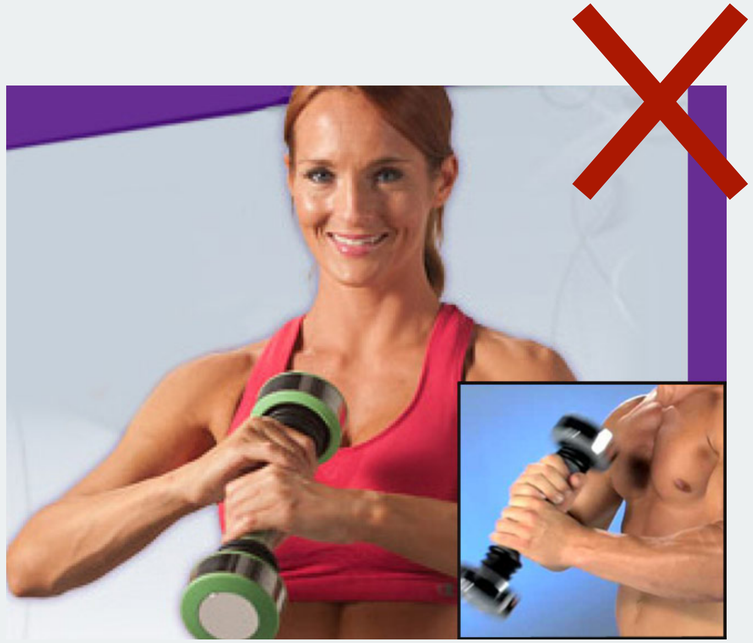
How does one work with Big Data?

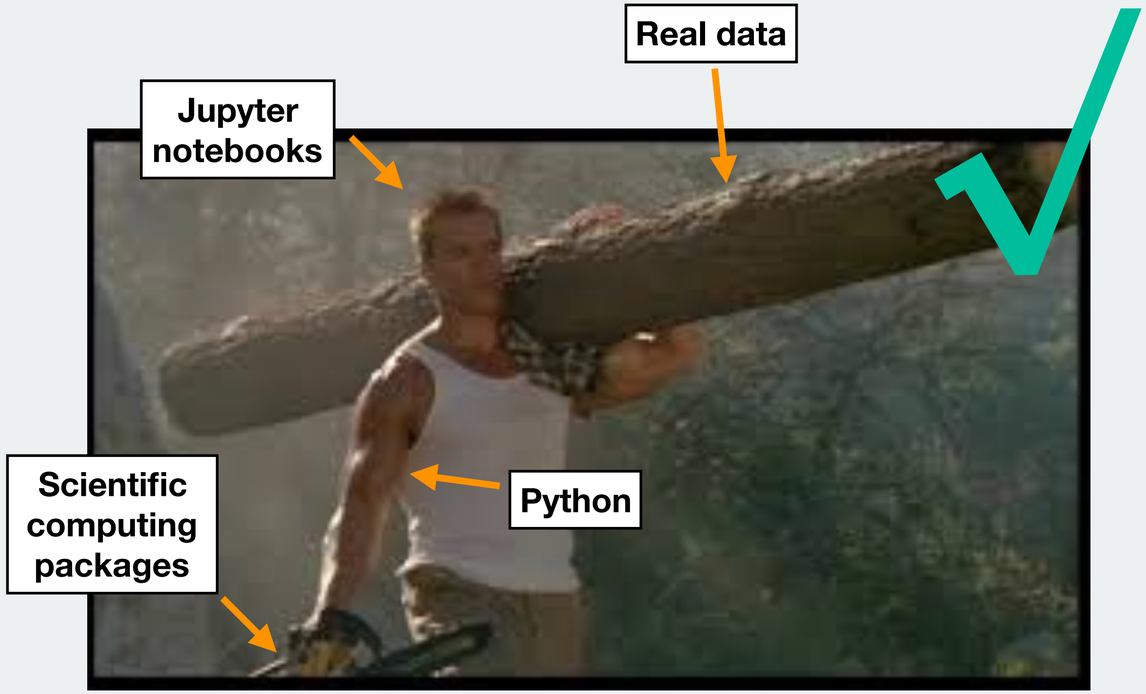
How does


Big Data Landscape 2016 (Version 3.0)
Infrastructure
Hadoop On-Premise
Hadoop in the Cloud
Spark
Cluster Services
NoSQL Databases
NewSQL Databases
Graph Databases
MPP Databases
Cloud EDW
Data Transformation
Data Integration
Management / Monitoring
Security
Storage
App Dev
Crowd-sourcing
Analytics
Analyst Platforms
Analytics Platforms
Data Science Platforms
Visualization
BI Platforms
Statistical Computing
Log Analytics
Social Analytics
Real-Time
Machine Learning
Speech & NLP
Horizontal AI
Search
Data Services
For Business Analysts
Web / Mobile / Commerce
Applications
Sales & Marketing
Customer Service
Human Capital
Legal
Ad Optimization
Security
Vertical AI Applications
Publisher Tools
Govt / Regulation
Finance
Education / Learning
Life Sciences
Industries
Cross-Infrastructure/Analytics
Open Source
Framework
Query / Data Flow
Data Access
Coordination
Real-Time
Stat Tools
Machine Learning
Search
Security
Visualization
Data Sources & APIs
Health
IOT
Financial & Economic Data
Air / Space / Sea
Location / People / Entities
Other
Incubators & Schools
Last Updated 3/23/2016
© Matt Turck (@mattturck), Jim Hao (@jimrhao), & FirstMark Capital (@firstmarkcap)
FIRSTMARK

How does one work with Big Data?

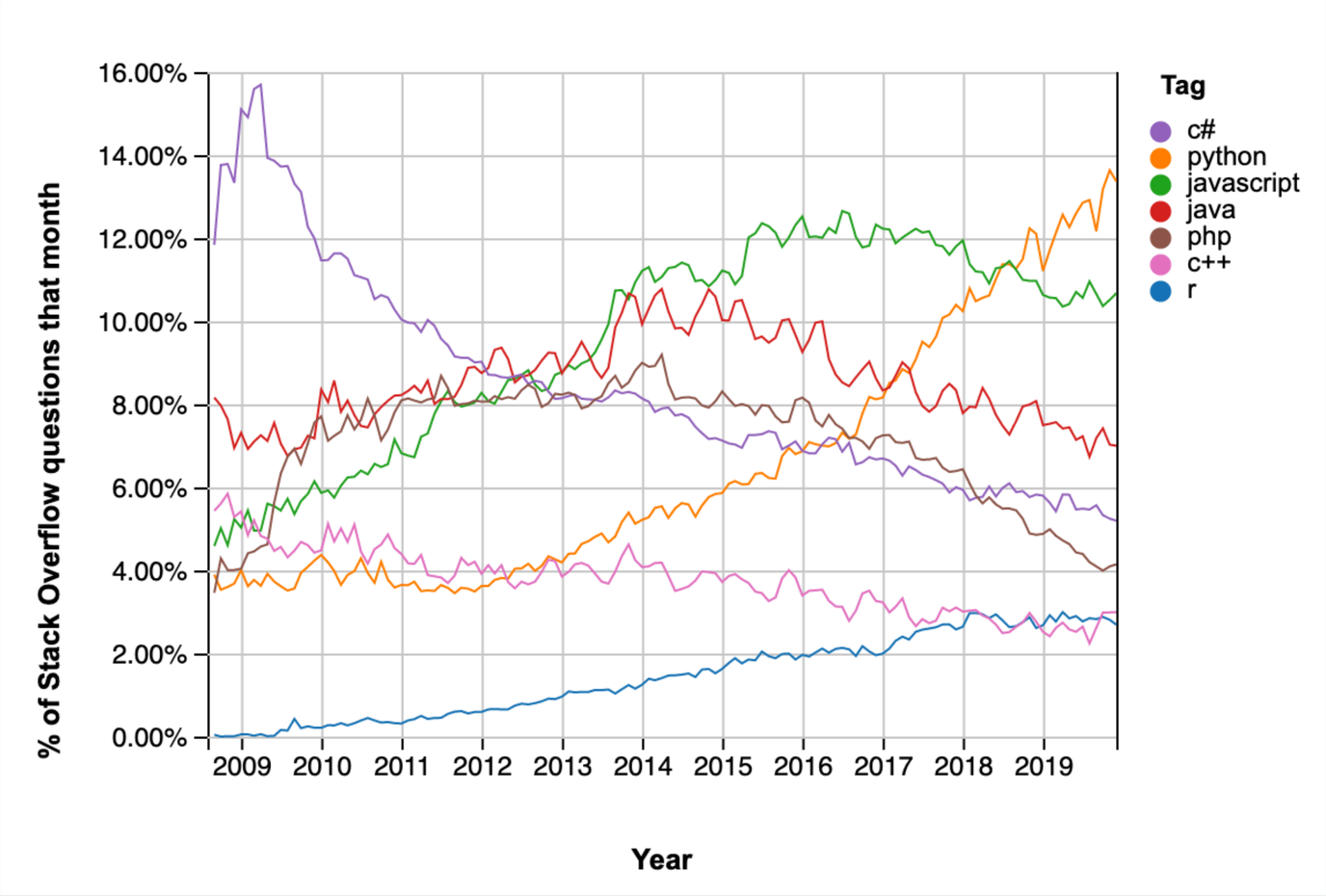
In this course: Everything from scratch





What is Python?

What is Python?



https://insights.stackoverflow.com

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

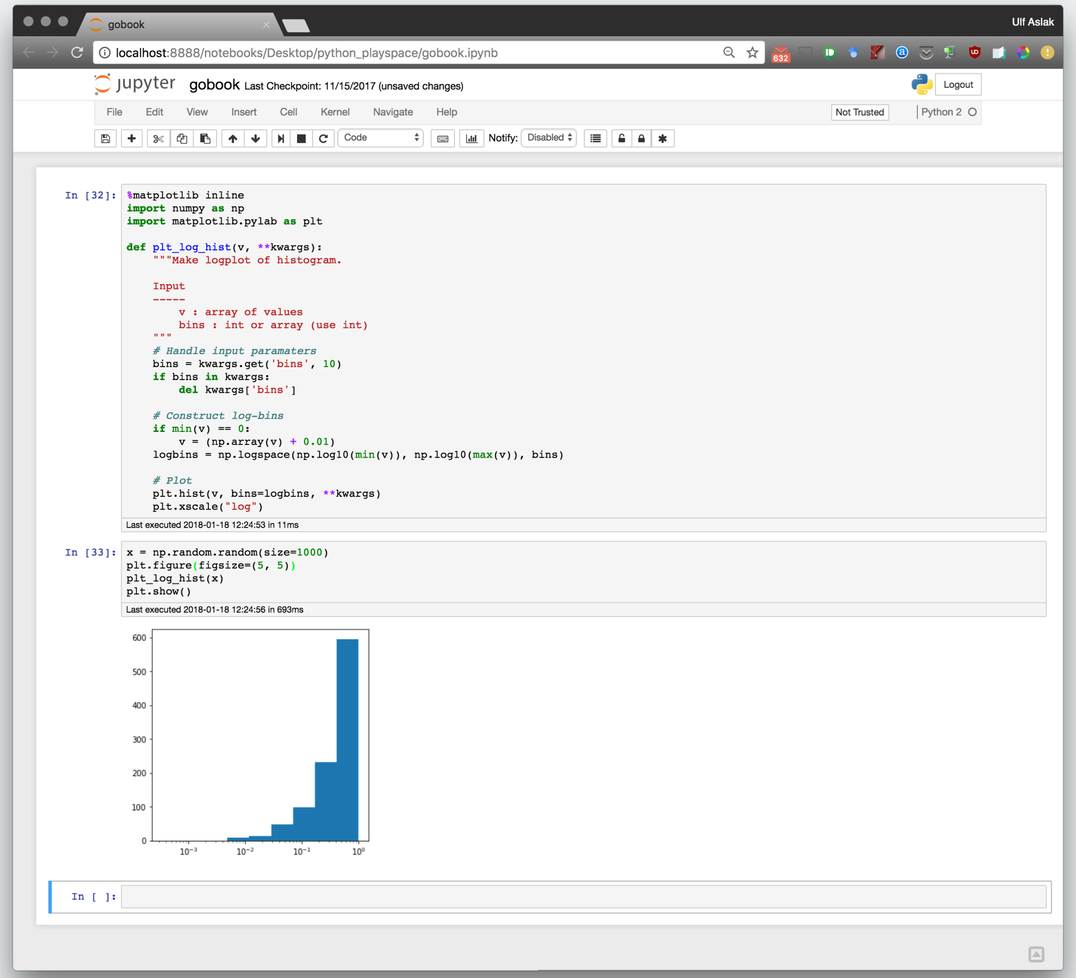
```python
import numpy as np
import matplotlib.pylab as plt

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

    Input
    -----
        v : array of values
        bins : int or array (use int)
    """
    # Handle input paramaters
    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")

if __name__ == '__main__':
    x = np.random.random(size=1000)
    plt.figure(figsize=(10, 10))
    plt_log_hist(x)
    plt.savefig("a_pretty_plot.pdf")
```

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

Desktop — ulfaslak@UAM — ~/Desktop — -zsh — 68×24

```
➜ Desktop git:(master) ✗ python my_script.py
```

Using Python with Jupyter notebooks



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/practical_data_science

4. In your terminal/console, navigate into the exercises folder inside of the newly created **practical_data_science** directory, and run the following command: jupyter notebook exercises_week1.ipynb.