Topics in Social Data Science

Week 1

Introduction and Python

Agenda

Who is teaching?

About this course

Social Data Science

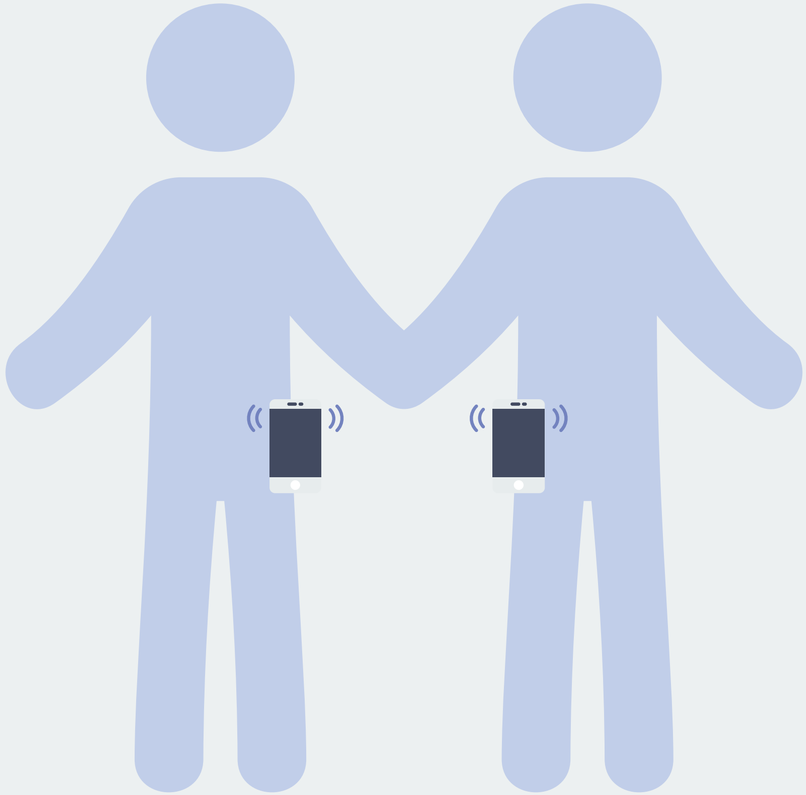
Python

Centre for Social Data Science (SODAS)

Centre for Social Data Science (SODAS)

http://sodas.ku.dk/

- Interdisciplinary research center working on all things related to social data.
- Knowledge hub for Data Science in the faculty of Social Sciences.
- Bridge between technical sciences research at DIKU and DTU, and social sciences at KU.



Teachers

David Dreyer Lassen (DDL)

Professor at Economics | Economics

@daviddlassen **econ.ku.dk/ddl/**

Andreas Bjerre-Nielsen (ABN)

Post Doc. at SODAS | Economics

abjer.github.io/

Snorre Ralund (SR)

PhD student at SODAS | Sociology

@snorreralund

Ulf Aslak Jensen (UA)

PhD student at SODAS | Physics, Math, CS

@ulfaslak **ulfaslak.com**

What is this course about?

Answering questions $\overbrace{\text{with data}}^{\text{in the Social Sciences}}$

What is this course about?

1. **Introduction and Python [UA]**
2. Scraping and Structuring [SR/ABN]
3. Data Modeling [ABN]
4. Networks 1 [UA]
5. Networks 2 [UA]
6. Spatial data [ABN]
7. Text [SR]
8. Machine Learning 1 [UA]
9. Machine Learning 2 [SR]
10. Natural language processing [SR]
11. Machine Learning 3 [ABN]
12. Wrap-up/project work
13. Project work

Answering questions with data

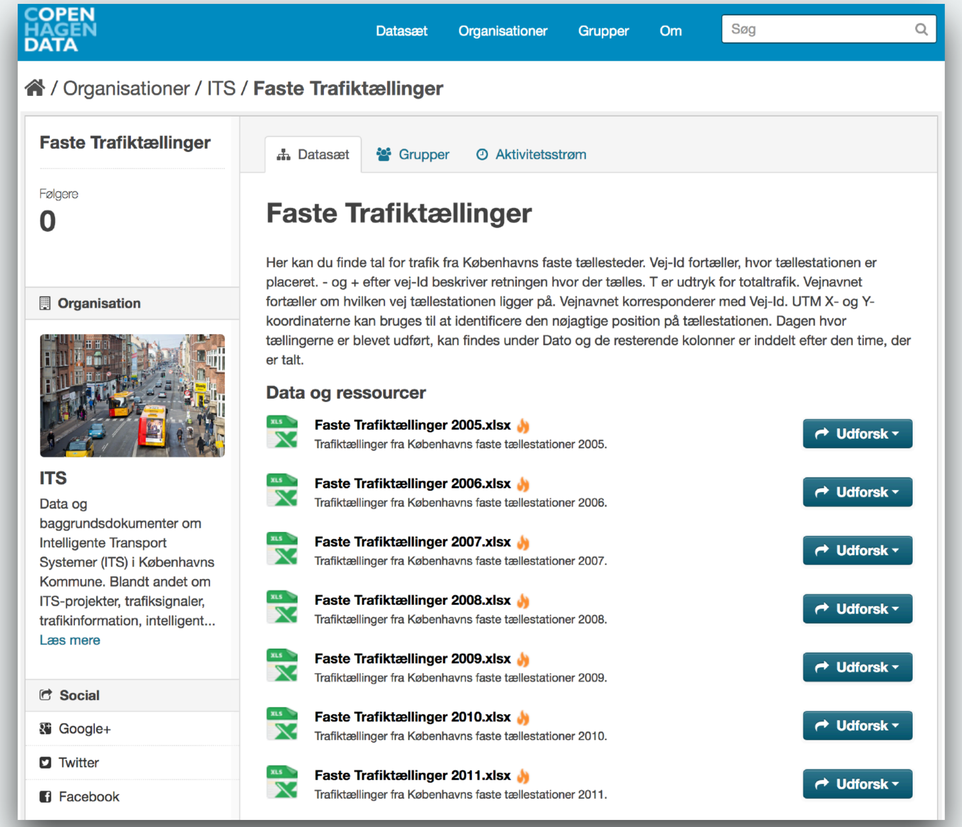
```python
def randomize_by_edge_swaps(G, num_iterations):
    """Randomizes the graph by swapping edges in such a way that
    preserves the degree distribution of the original graph.

    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 familiar with Python for Data Science**

What is this course about?

1. Introduction and Python [UA]
2. **Scraping and Structuring [SR/ABN]**
3. Data Modeling [ABN]
4. Networks 1 [UA]
5. Networks 2 [UA]
6. Spatial data [ABN]
7. Text [SR]
8. Machine Learning 1 [UA]
9. Machine Learning 2 [SR]
10. Natural language processing [SR]
11. Machine Learning 3 [ABN]
12. Wrap-up/project work
13. Project work

Answering questions with data

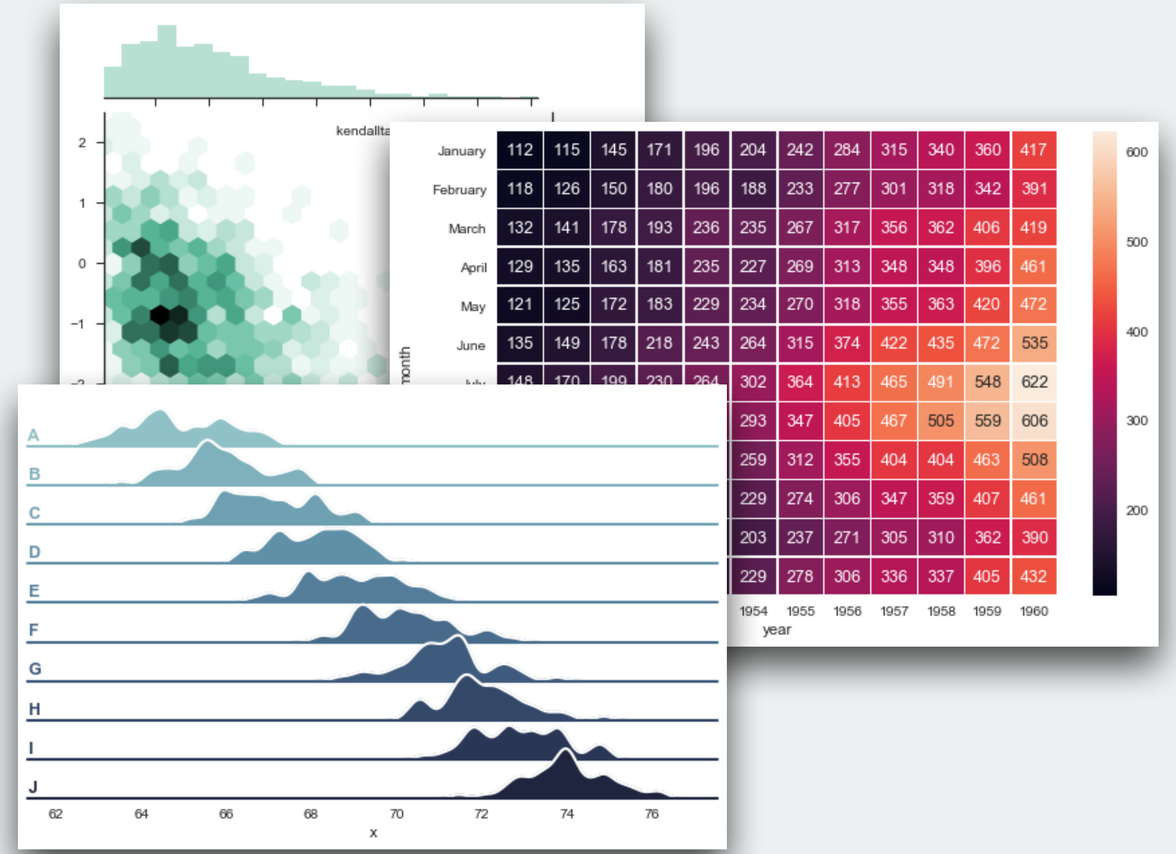


Aim: **Learn to use `pandas` for scraping and structuring data**

What is this course about?

1. Introduction and Python [UA]
2. Scraping and Structuring [SR/ABN]
3. **Data Modeling [ABN]**
4. Networks 1 [UA]
5. Networks 2 [UA]
6. Spatial data [ABN]
7. Text [SR]
8. Machine Learning 1 [UA]
9. Machine Learning 2 [SR]
10. Natural language processing [SR]
11. Machine Learning 3 [ABN]
12. Wrap-up/project work
13. Project work

Answering questions with data

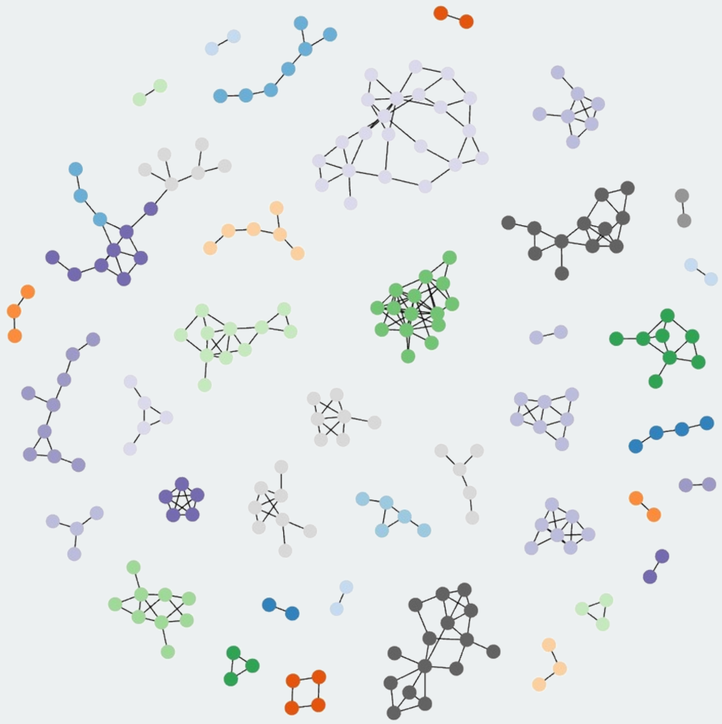


Aim: **Get comfortable exploring a new dataset**

What is this course about?

1. Introduction and Python [UA]
2. Scraping and Structuring [SR/ABN]
3. Data Modeling [ABN]
4. **Networks 1 [UA]**
5. Networks 2 [UA]
6. Spatial data [ABN]
7. Text [SR]
8. Machine Learning 1 [UA]
9. Machine Learning 2 [SR]
10. Natural language processing [SR]
11. Machine Learning 3 [ABN]
12. Wrap-up/project work
13. Project work

Answering questions with data

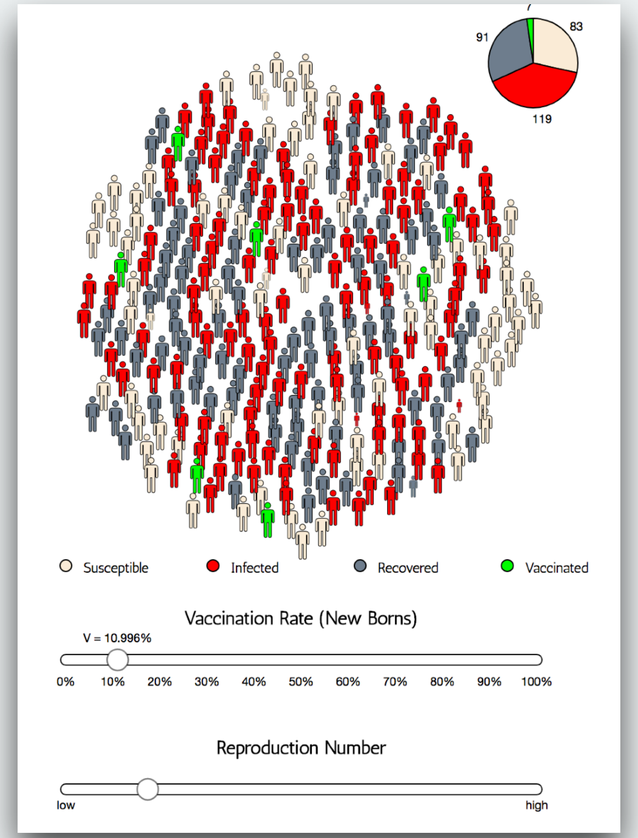


Aim: **Understand the basic properties of Networks and when they are useful**

What is this course about?

1. Introduction and Python [UA]
2. Scraping and Structuring [SR/ABN]
3. Data Modeling [ABN]
4. Networks 1 [UA]
5. **Networks 2 [UA]**
6. Spatial data [ABN]
7. Text [SR]
8. Machine Learning 1 [UA]
9. Machine Learning 2 [SR]
10. Natural language processing [SR]
11. Machine Learning 3 [ABN]
12. Wrap-up/project work
13. Project work

Answering questions with data

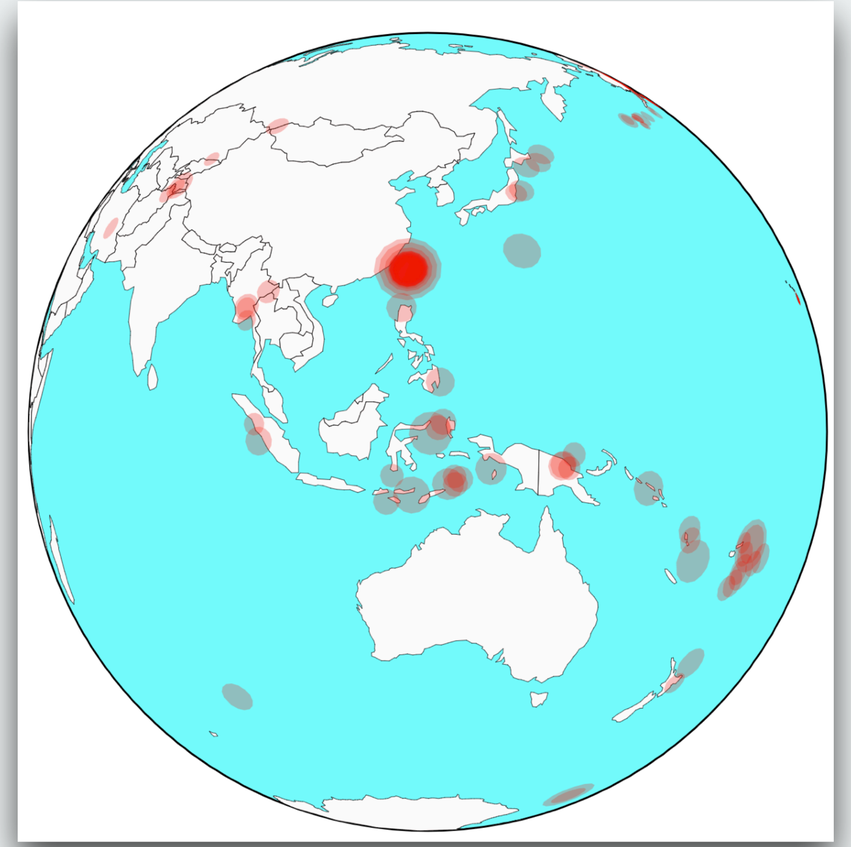


Aim: **Learn methods for modeling processes using networks**

What is this course about?

1. Introduction and Python [UA]
2. Scraping and Structuring [SR/ABN]
3. Data Modeling [ABN]
4. Networks 1 [UA]
5. Networks 2 [UA]
6. **Spatial data [ABN]**
7. Text [SR]
8. Machine Learning 1 [UA]
9. Machine Learning 2 [SR]
10. Natural language processing [SR]
11. Machine Learning 3 [ABN]
12. Wrap-up/project work
13. Project work

Answering questions with data

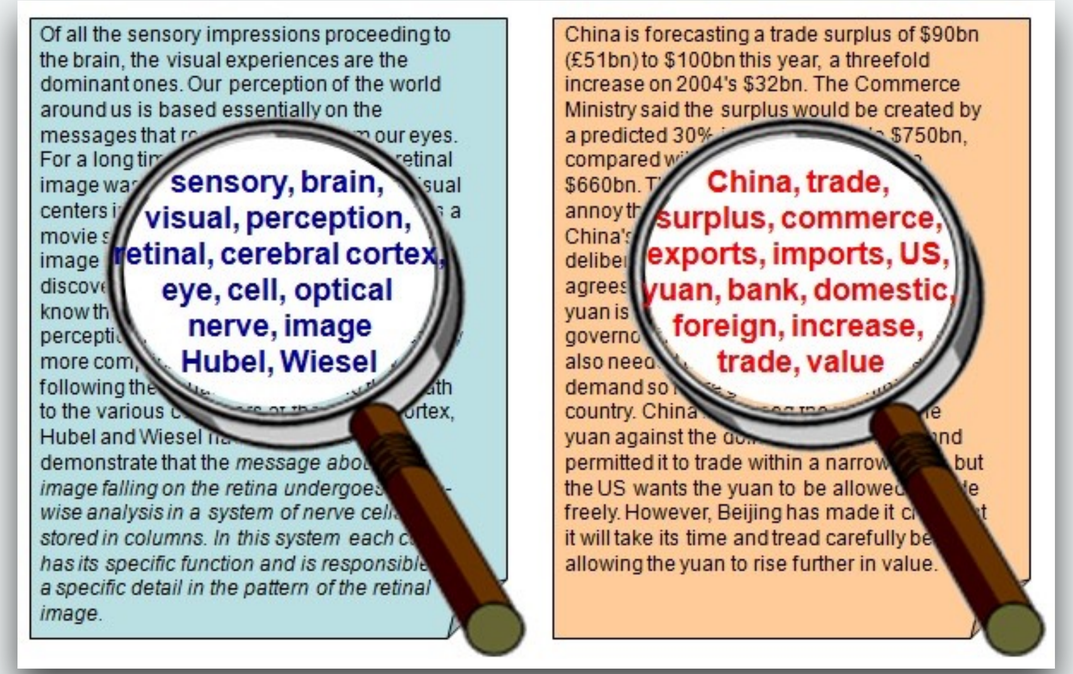


Aim: **Learn how to work with geospatial data**

What is this course about?

1. Introduction and Python [UA]
2. Scraping and Structuring [SR/ABN]
3. Data Modeling [ABN]
4. Networks 1 [UA]
5. Networks 2 [UA]
6. Spatial data [ABN]
7. **Text [SR]**
8. Machine Learning 1 [UA]
9. Machine Learning 2 [SR]
10. Natural language processing [SR]
11. Machine Learning 3 [ABN]
12. Wrap-up/project work
13. Project work

Answering questions with data

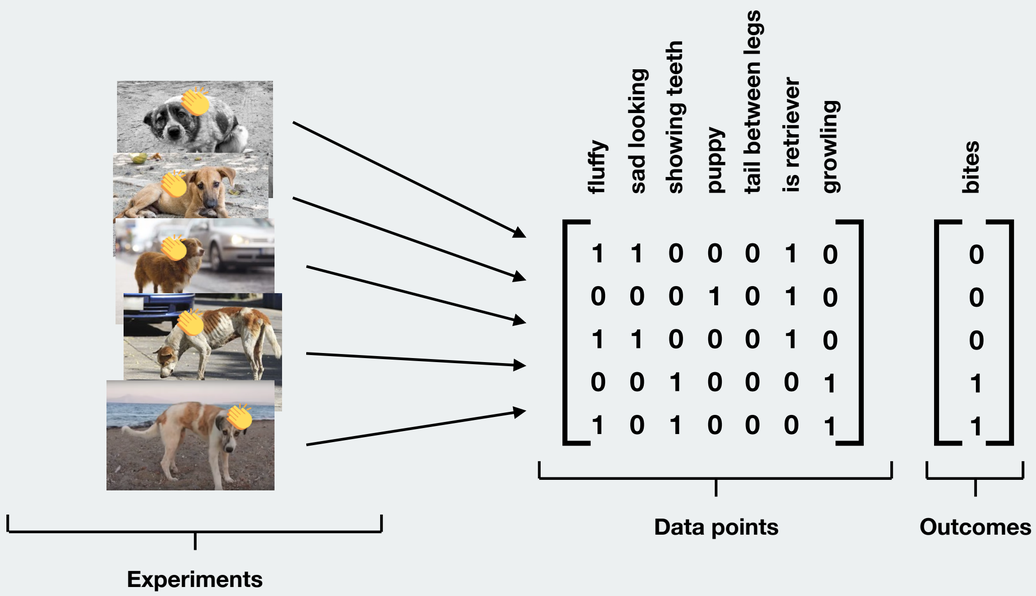


Aim: **Understand how text can be treated as data**

What is this course about?

1. Introduction and Python [UA]
2. Scraping and Structuring [SR/ABN]
3. Data Modeling [ABN]
4. Networks 1 [UA]
5. Networks 2 [UA]
6. Spatial data [ABN]
7. Text [SR]
8. **Machine Learning 1 [UA]**
9. Machine Learning 2 [SR]
10. Natural language processing [SR]
11. Machine Learning 3 [ABN]
12. Wrap-up/project work
13. Project work

Answering questions with data

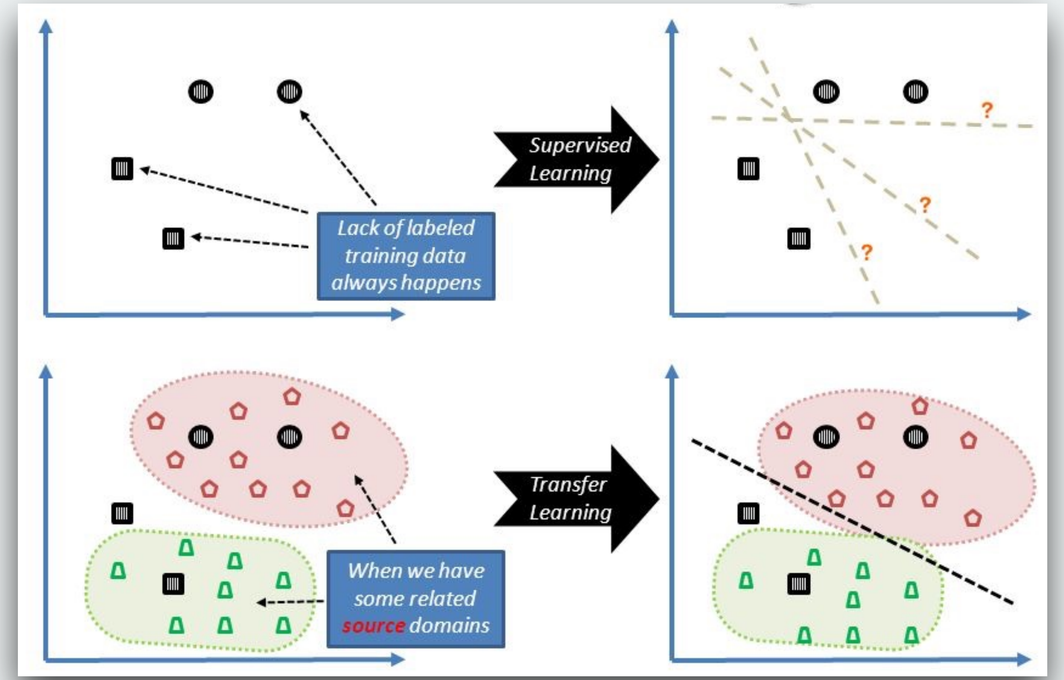


Aim: **Understand the basic paradigm, learn about model fitting, validation, etc.**

What is this course about?

1. Introduction and Python [UA]
2. Scraping and Structuring [SR/ABN]
3. Data Modeling [ABN]
4. Networks 1 [UA]
5. Networks 2 [UA]
6. Spatial data [ABN]
7. Text [SR]
8. Machine Learning 1 [UA]
9. **Machine Learning 2 [SR]**
10. Natural language processing [SR]
11. Machine Learning 3 [ABN]
12. Wrap-up/project work
13. Project work

Answering questions with data



Aim: **Be able to build robust models using methods from unsupervised- and transfer learning.**

What is this course about?

1. Introduction and Python [UA]
2. Scraping and Structuring [SR/ABN]
3. Data Modeling [ABN]
4. Networks 1 [UA]
5. Networks 2 [UA]
6. Spatial data [ABN]
7. Text [SR]
8. Machine Learning 1 [UA]
9. Machine Learning 2 [SR]
10. **Natural language processing [SR]**
11. Machine Learning 3 [ABN]
12. Wrap-up/project work
13. Project work

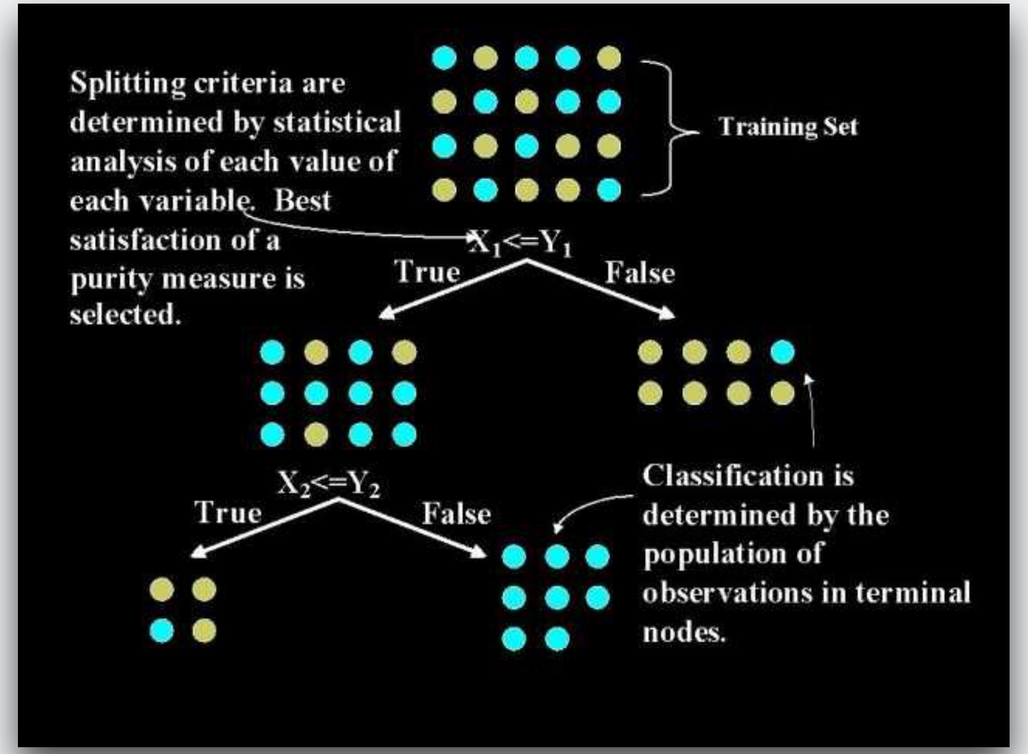
Answering questions with data

Aim: **Get introduced to the (huge) field of NLP**

What is this course about?

1. Introduction and Python [UA]
2. Scraping and Structuring [SR/ABN]
3. Data Modeling [ABN]
4. Networks 1 [UA]
5. Networks 2 [UA]
6. Spatial data [ABN]
7. Text [SR]
8. Machine Learning 1 [UA]
9. Machine Learning 2 [SR]
10. Natural language processing [SR]
11. **Machine Learning 3 [ABN]**
12. Wrap-up/project work
13. Project work

Answering questions with data

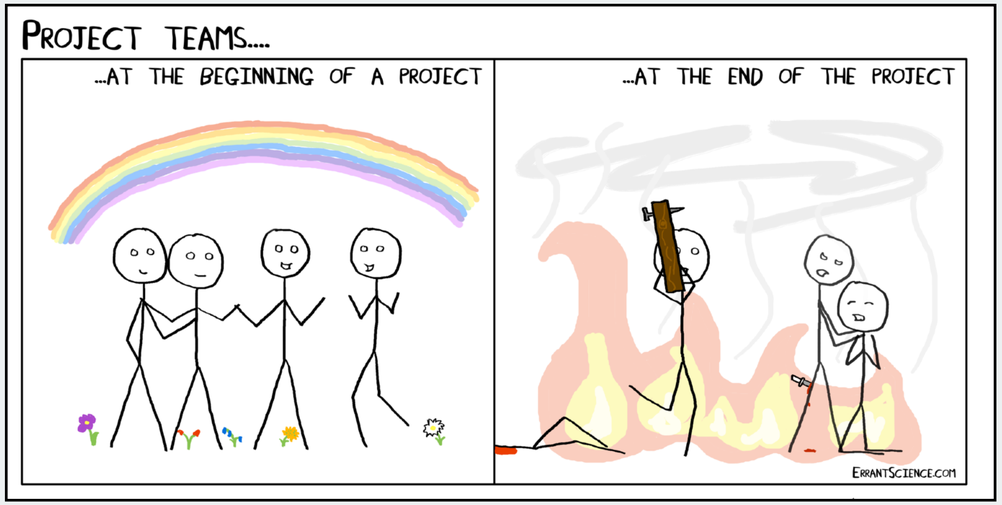


Aim: **Learn about machine learning and causality.**

What is this course about?

1. Introduction and Python [UA]
2. Scraping and Structuring [SR/ABN]
3. Data Modeling [ABN]
4. Networks 1 [UA]
5. Networks 2 [UA]
6. Spatial data [ABN]
7. Text [SR]
8. Machine Learning 1 [UA]
9. Machine Learning 2 [SR]
10. Natural language processing [SR]
11. Machine Learning 3 [ABN]
12. **Wrap-up/project work**
13. **Project work**

Answering questions with data



Aim: **Synthesize the things you have learned and make an awesome project**

What will you learn?

Knowledge and competences

- Get to know the landscape of problems and tools in Data Science
- Know where to start when you want to analyze something that requires lots of complex data
- Rethink how to do things in your own field. Get some business ideas maybe?

Concrete skills

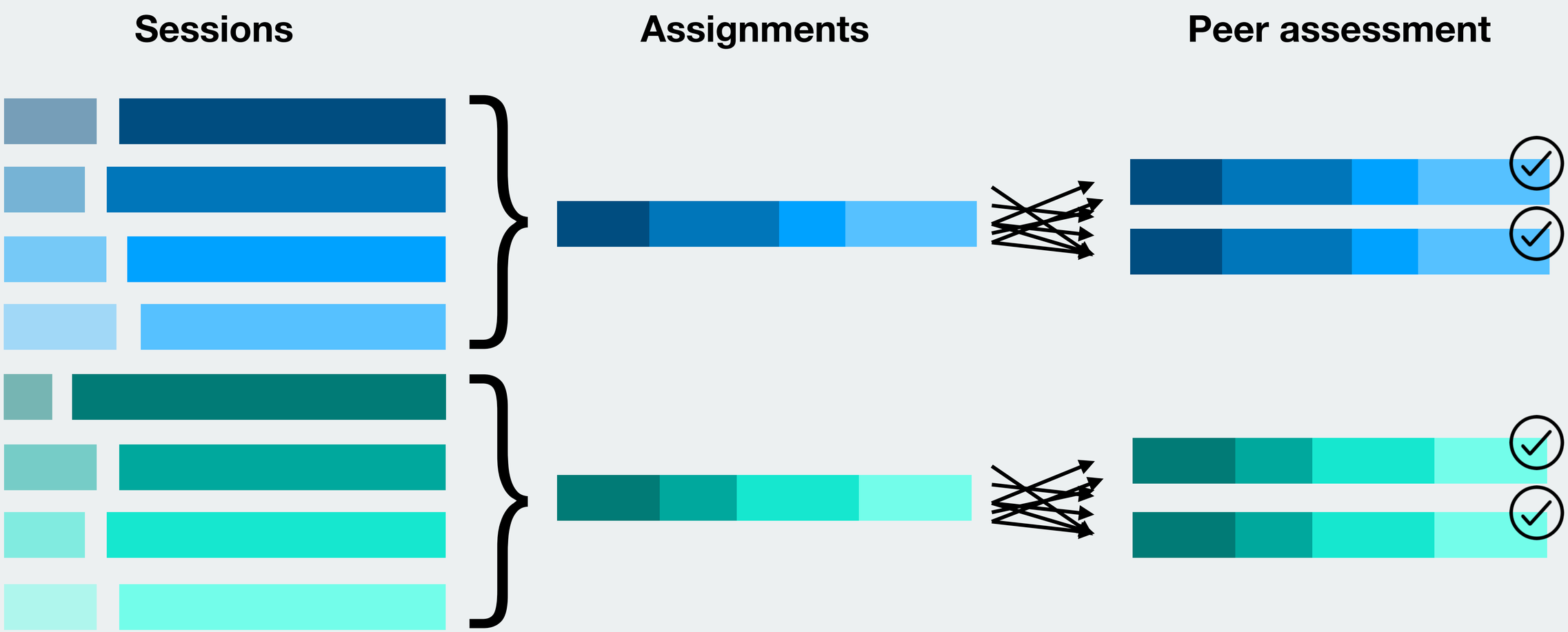
- Coding in Python
- Handle large amounts of data
- Build models for predictions based on data
- Visualize complex data

How assignments work

Sessions

Lecture | **Solve exercises**

How assignments work



The final project

Apr 26

We validate your project idea

May 28

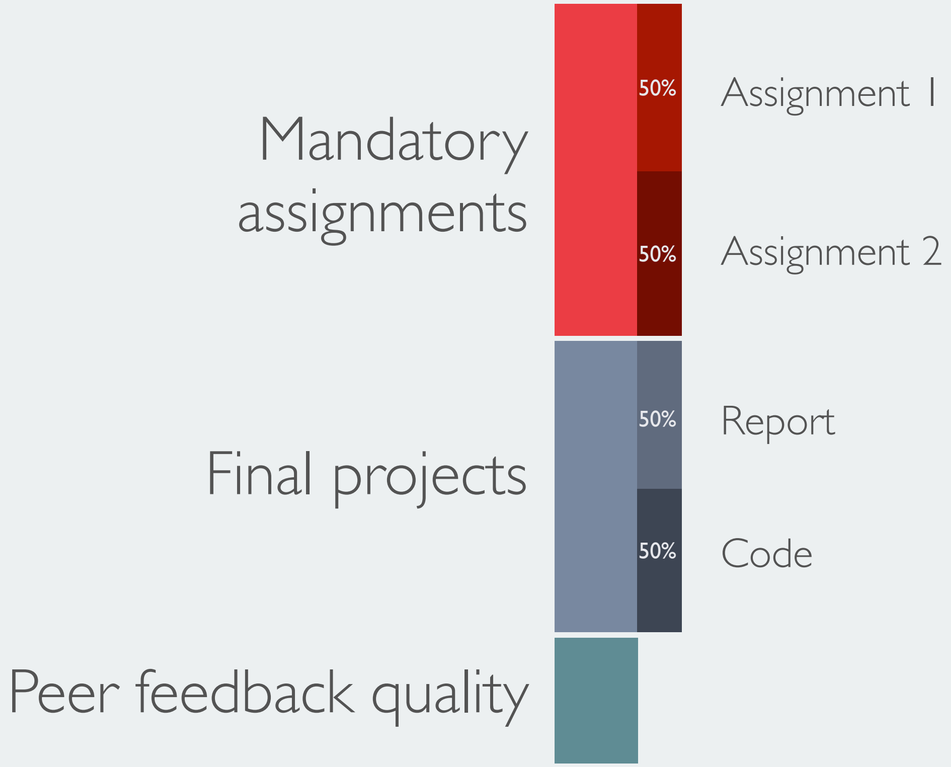
You deliver a report (and code on GitHub)

Deliverable

Sometime in June

We give you a grade

How to do well in this course



Best strategy:

Show up **prepared** and do as many of the **exercises** as possible

1. Complete the *preparation* for each session

2. Be inquisitive. Ask lots of questions in class, raise issues on GitHub and up your Googling-game

How to do well in this course

Algorithm that solves 95% of all exercises:

1. **Rephrase question in general wording**
2. Google search
3. Pick first link that leads to StackOverflow
4. Read most upvoted answer

Ex 1.1.13: Take a list `a = list("justreadtheinstructions")` and

1. count the number of times each element occurs using `Counter`,
2. report the two most common elements

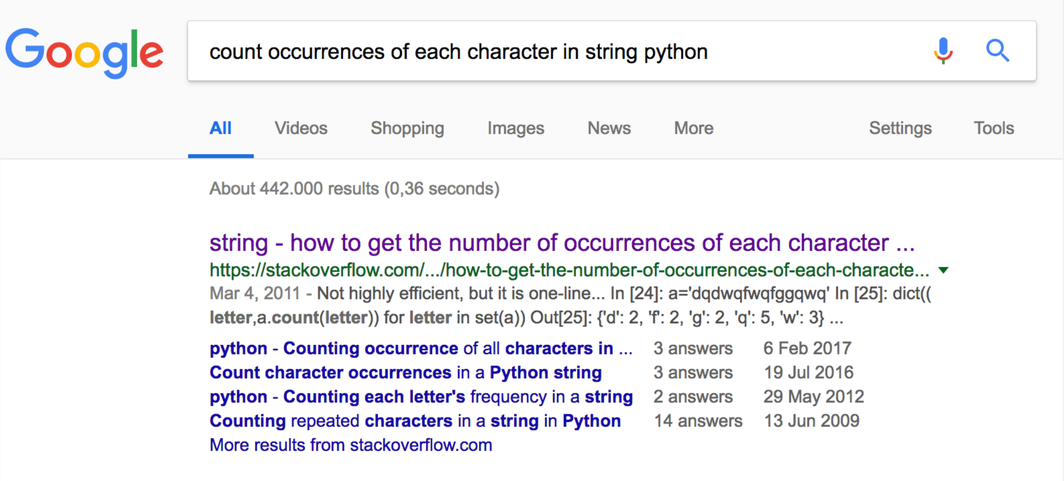
Hint: you can import `Counter` from `collections`

→ "count occurrences of each character in string python"

How to do well in this course

Algorithm that solves 95% of all exercises:

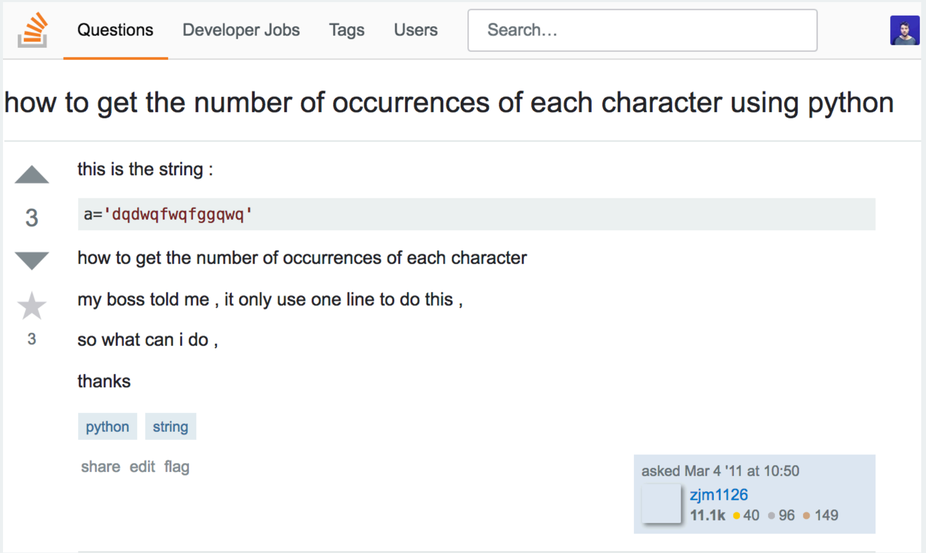
1. Rephrase question in general wording
2. **Google search**
3. Pick first link that leads to StackOverflow
4. Read most upvoted answer



How to do well in this course

Algorithm that solves 95% of all exercises:

1. Rephrase question in general wording
2. Google search
3. **Pick first link that leads to StackOverflow**
4. Read most upvoted answer



How to do well in this course

Algorithm that solves 95% of all exercises:

1. Rephrase question in general wording
2. Google search
3. Pick first link that leads to StackOverflow
4. **Read most upvoted answer**

15

In 2.7 and 3.1 there's a tool called Counter:

```
>>> import collections
>>> results = collections.Counter("dqdwqfwqfggqwq")
>>> results
Counter({'q': 5, 'w': 3, 'g': 2, 'd': 2, 'f': 2})
```

Docs. As pointed out in the comments it is not compatible with 2.6 or lower, but it's backported.

share edit flag

answered Mar 4 '11 at 10:57

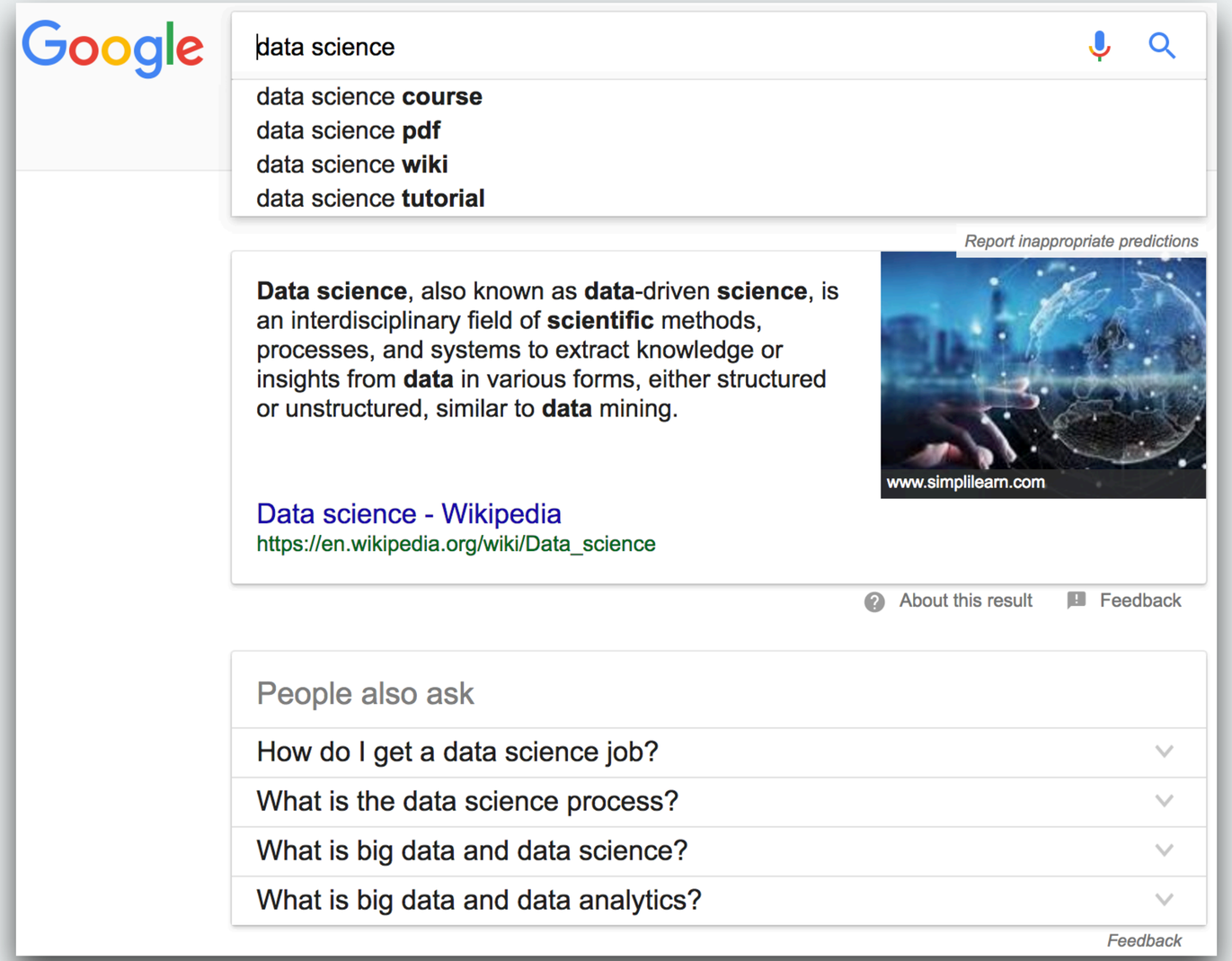
Bjorn

3,651 13 28

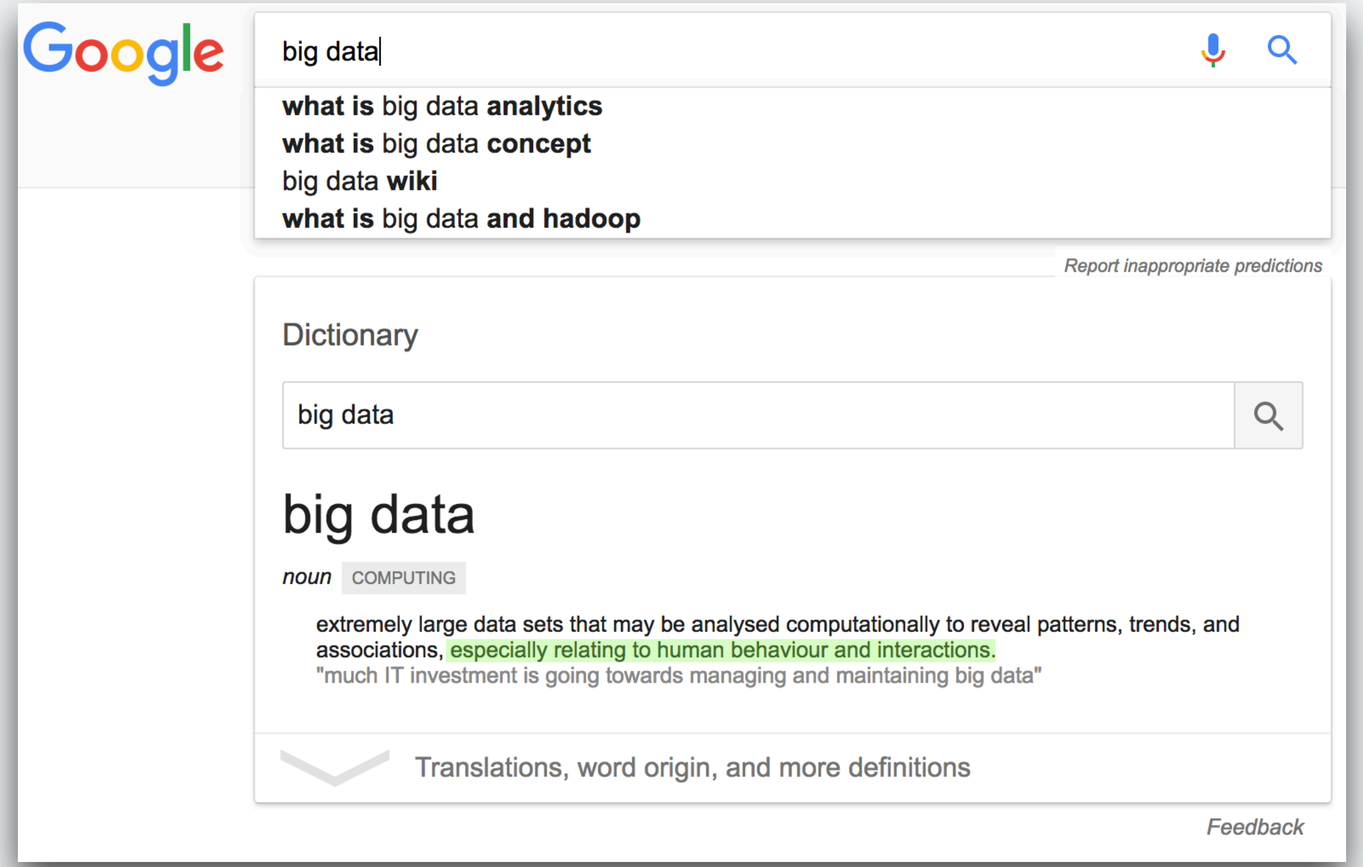
Everything else is on GitHub

What is Social Data Science?

What is Social **Data Science**?



Internet Live Stats


Google
big data
what is big data analytics
what is big data concept
big data wiki
what is big data and hadoop
Report inappropriate predictions
Dictionary
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
Feedback

Social Data Science ≈ Data Science ≈ Big Data

What do we use it *for*? **Understanding people**

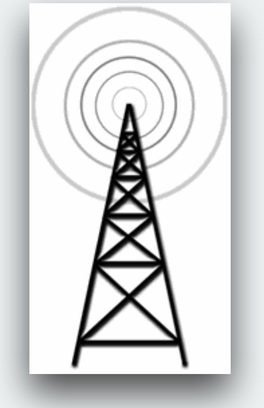
What do we use it *for*?

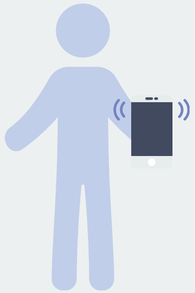
Ads

theTradeDesk

Omnicom WPP

Research





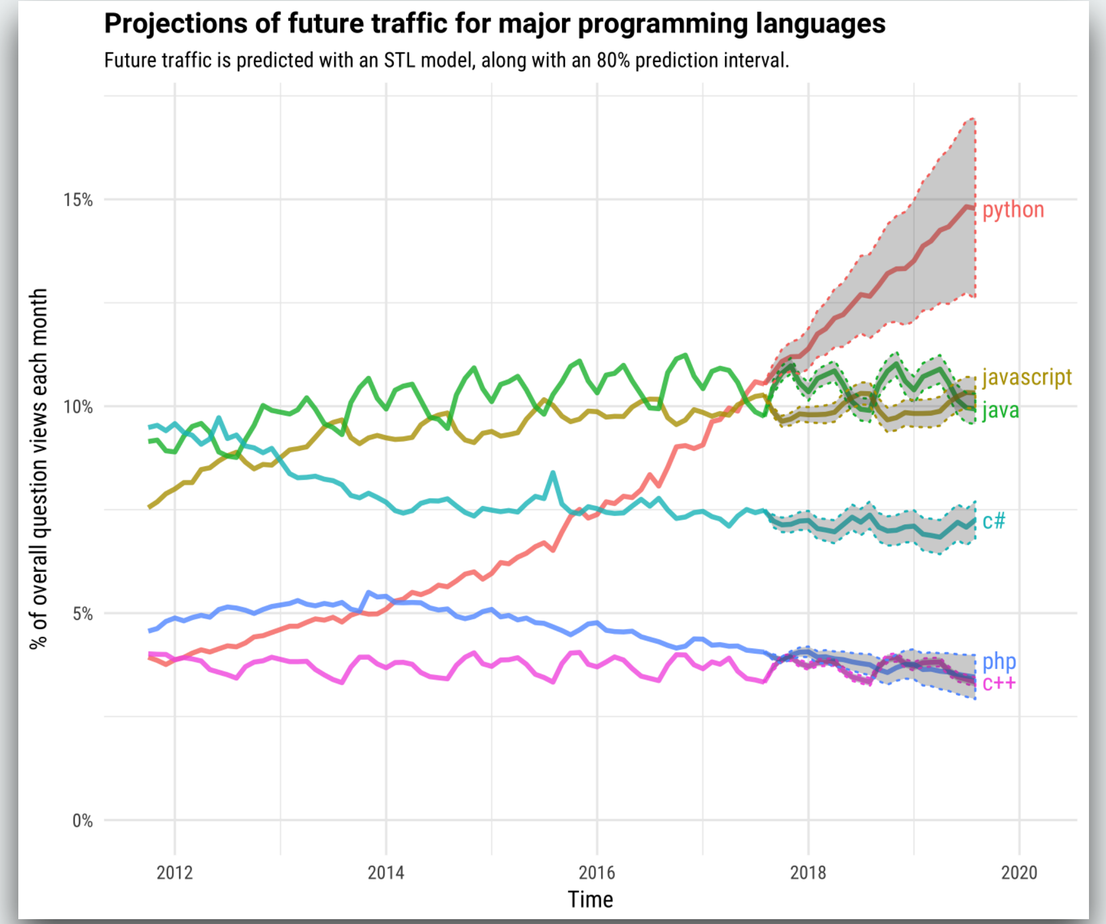
User customization

YouTube amazon.com

NETFLIX TED talk

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.

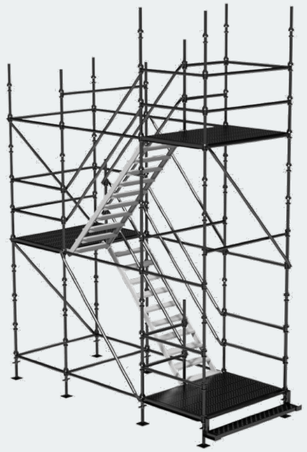
Why use Python?

Why use Python?

Why use Python?

→

→



Why use Python?

→ STATA® →

Why use Python?

→

→

Why use Python?

Reason 1: It's low-level enough that we can build anything we want

Reason 2: It's high-level enough that we can do it in our lifetimes

How to use Python?

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")
```

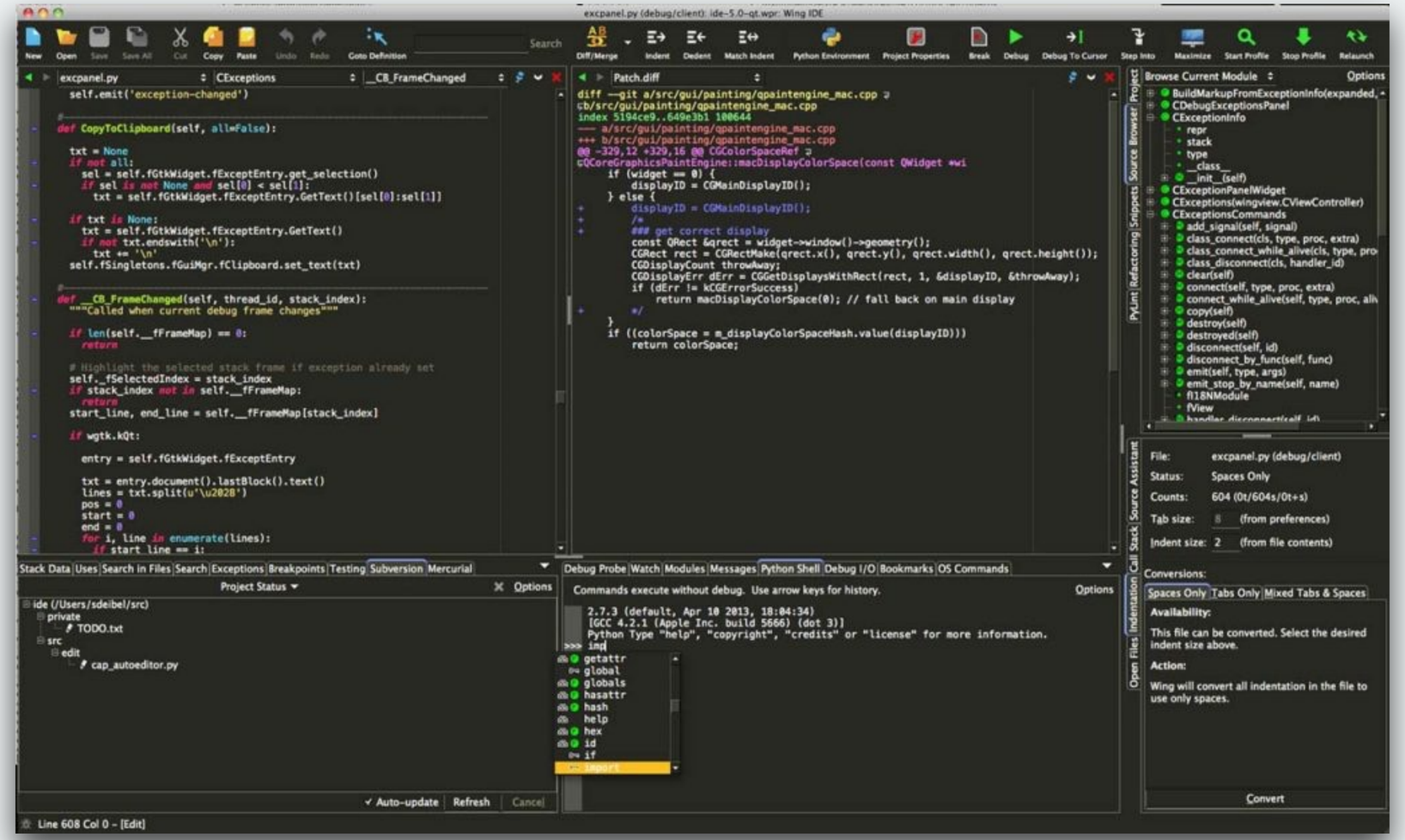
In ulfaslak on master, File is untracked, Line 26, Column 25 Tab Size: 4 Python

→

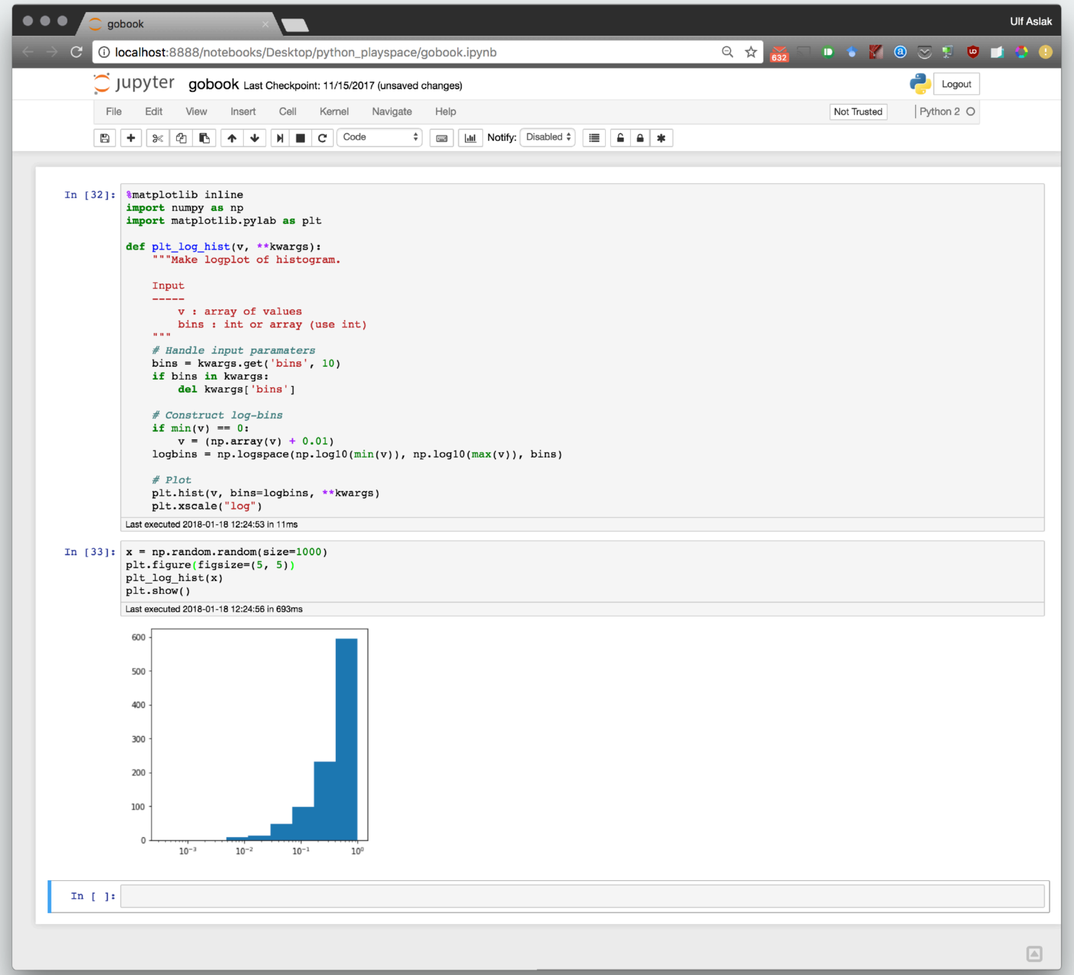
Desktop — ulfaslak@UAM — ~/Desktop — -zsh — 68×24

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

Using Python in an IDE



Using Python with Jupyter notebooks



Python for Data Science

- **NumPy** — Arrays and linear algebra
- **SciPy** — Computational tools
- **Matplotlib** — Basic plotting library
- **pandas** — Operating on numerical tables
- **seaborn** — High-level plotting library
- **scikit-learn** — Machine learning tools

Let’s get started!

https://github.com/abjer/tsds