DATA VISUALISATION

Florian Langelaar

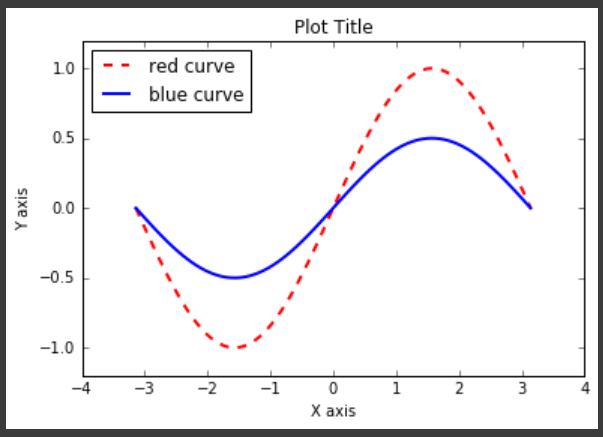
ASSIGNMENTS

Assignment 1

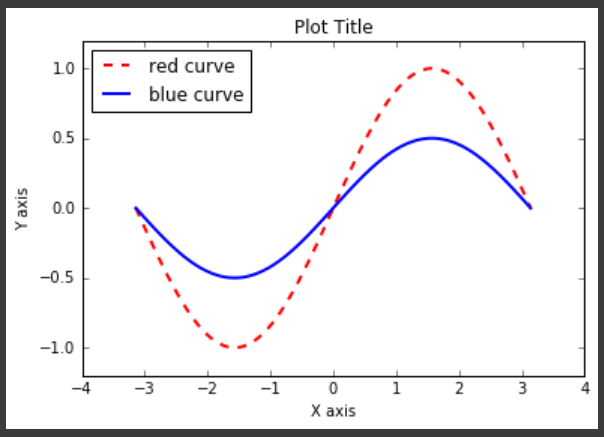
https://github.com/Floor9494/ADS_DV/blob/master/DV-assignment-01.ipynb

Realiseer plot 1 met subplots (plot 2)

Plot 1



Plot 2



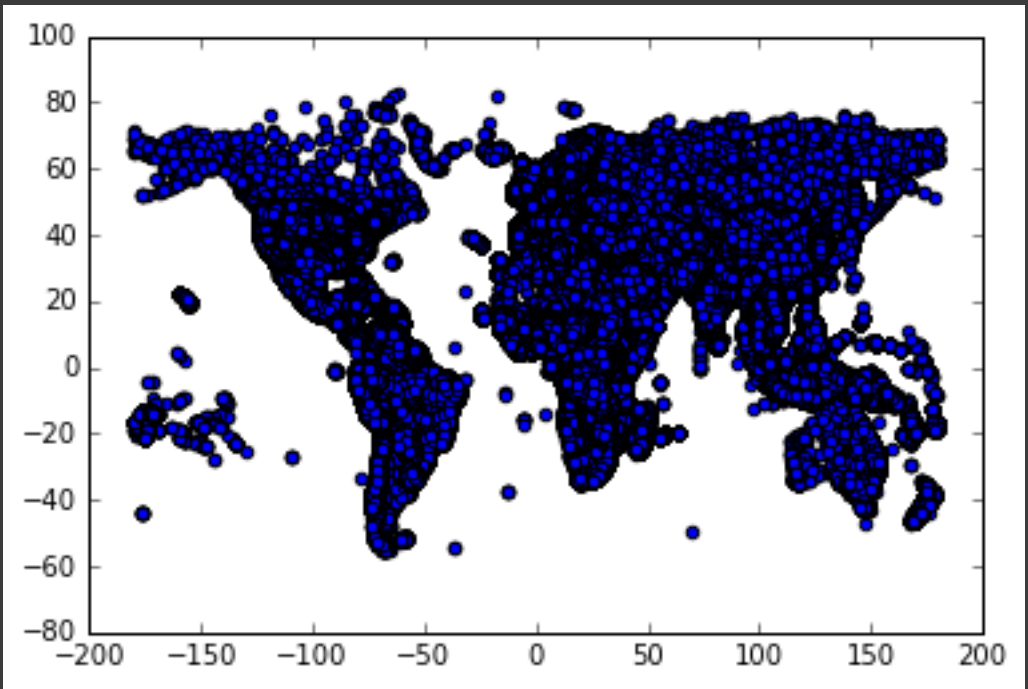

```
import numpy as np
import matplotlib as mpl
from matplotlib import pyplot as plt

%matplotlib inline

X = np.linspace(-np.pi, np.pi, 100)
Y = np.sin(X)

fig, ax = plt.subplots()
ax.plot(X, Y, 'r--', linewidth=2);
ax.plot(X, Y/2, 'b-', linewidth=2);

plt.xlabel('X axis')
plt.ylabel('Y axis')
plt.title('Plot Title')
plt.xlim(-4, 4)
plt.ylim(-1.2, 1.2)
plt.legend(['red curve', 'blue curve'], loc='best');
```

Assignment 2

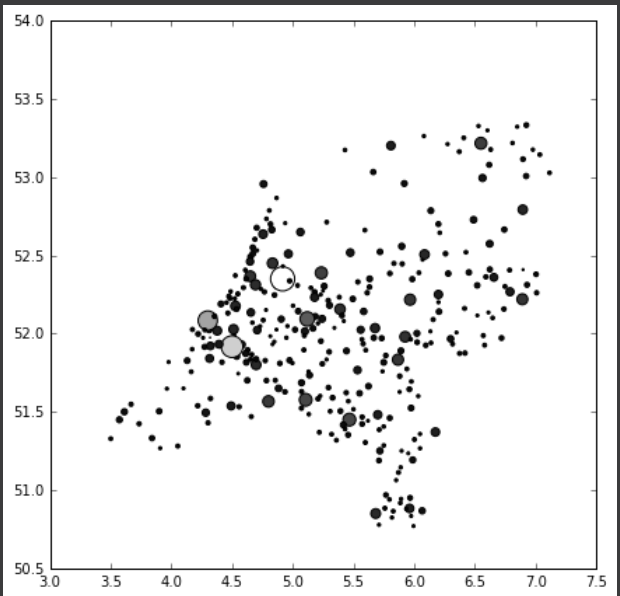
https://github.com/Floor9494/ADS_DV/blob/master/DV-assignment-02.ipynb

Maak een scatter-plot van de steden in de dataset.

```
import numpy as np
from matplotlib import pyplot as plt
%matplotlib inline

plt.figure();
x = cities['Longitude'];
y = cities['Latitude'];

fig, ax = plt.subplots()
ax.scatter(x, y)
```

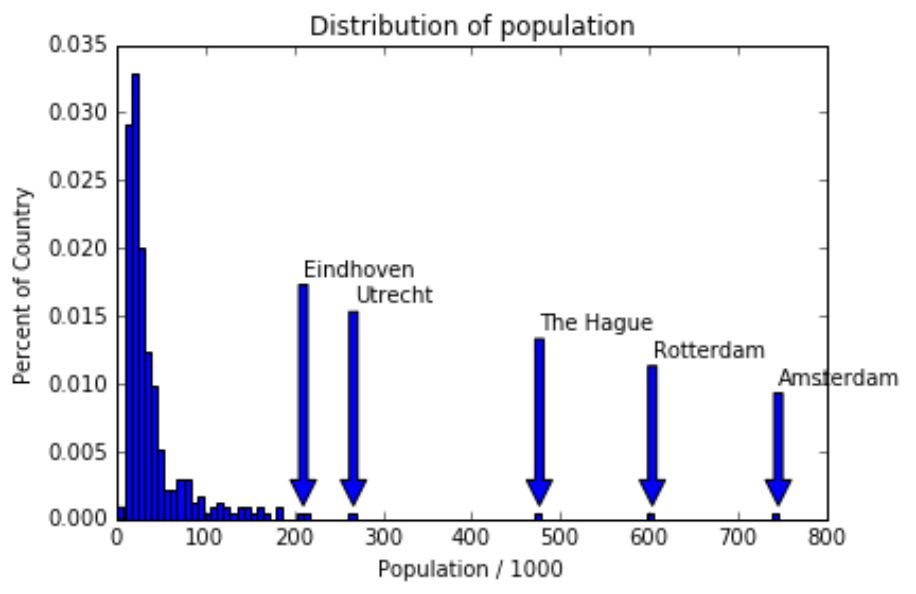


Plot alle steden in Nederland (niet de Antillen) met verschillende grote voor de hoeveelheid inwoners.



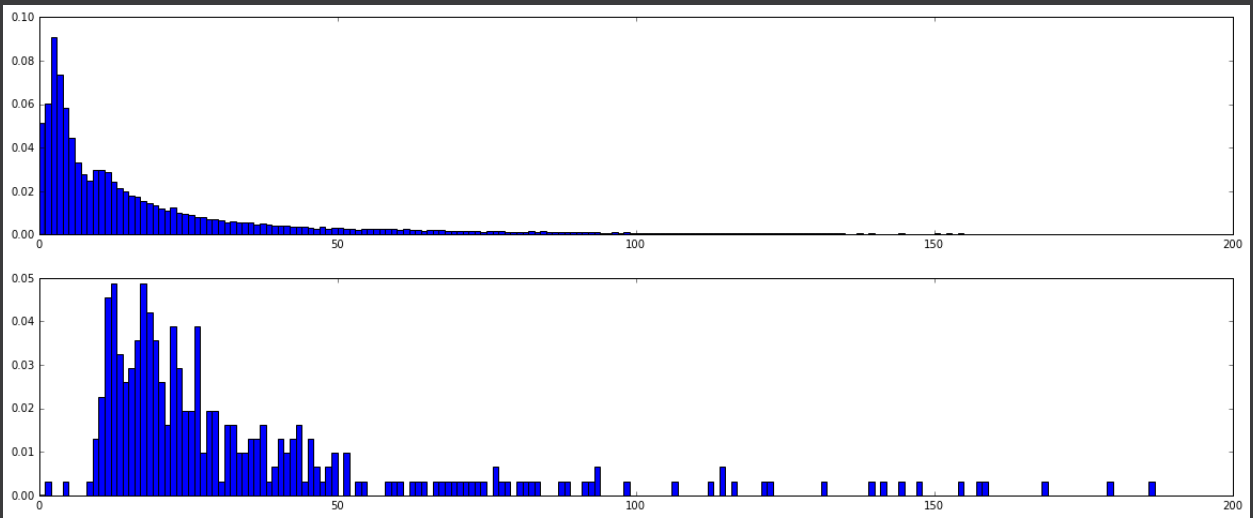

```
import math

dutch_cities = cities[ cities['Country'] =='nl' ]
plt.figure(figsize=[7,7]);

x = dutch_cities['Longitude']
y = dutch_cities['Latitude']
t = dutch_cities['Population'].tolist()
q = []
for p in t:
    if not math.isnan(p):
        p = p / 2500
        q.append(p)

plt.scatter(x, y, s = q, c = t)
plt.xlim(3, 7.5);
plt.ylim(50.5, 54);

plt.gray()
```

Maak een histogram met de grootste steden en geef Amsterdam en Eindhoven aan.




```
plt.figure();
plt.hist(np.asarray(dutch_cities.dropna().Population/1000), 100, normed=1);
plt.xlabel('Population / 1000');
plt.ylabel('Percent of Country');
plt.title('Distribution of population');

ax = plt.subplot();
#ax.arrow(400, 0.03, 1, -0.5, fc="k", ec="k", head_width=0.05, head_length=0.1 );

overlap = 0.01

bigCities = dutch_cities.sort_values(by='Population', ascending=False).head(5)
for ind, x in bigCities.iterrows():
    pop = x['Population']/1000

    plt.annotate(x['AccentCity'],
             xy=(pop, 0.001),
             xytext=(pop, overlap),
             arrowprops=dict(facecolor='blue')
            )
    overlap += 0.002
```

Vergelijk de distributie van Nederland met de rest van de wereld.



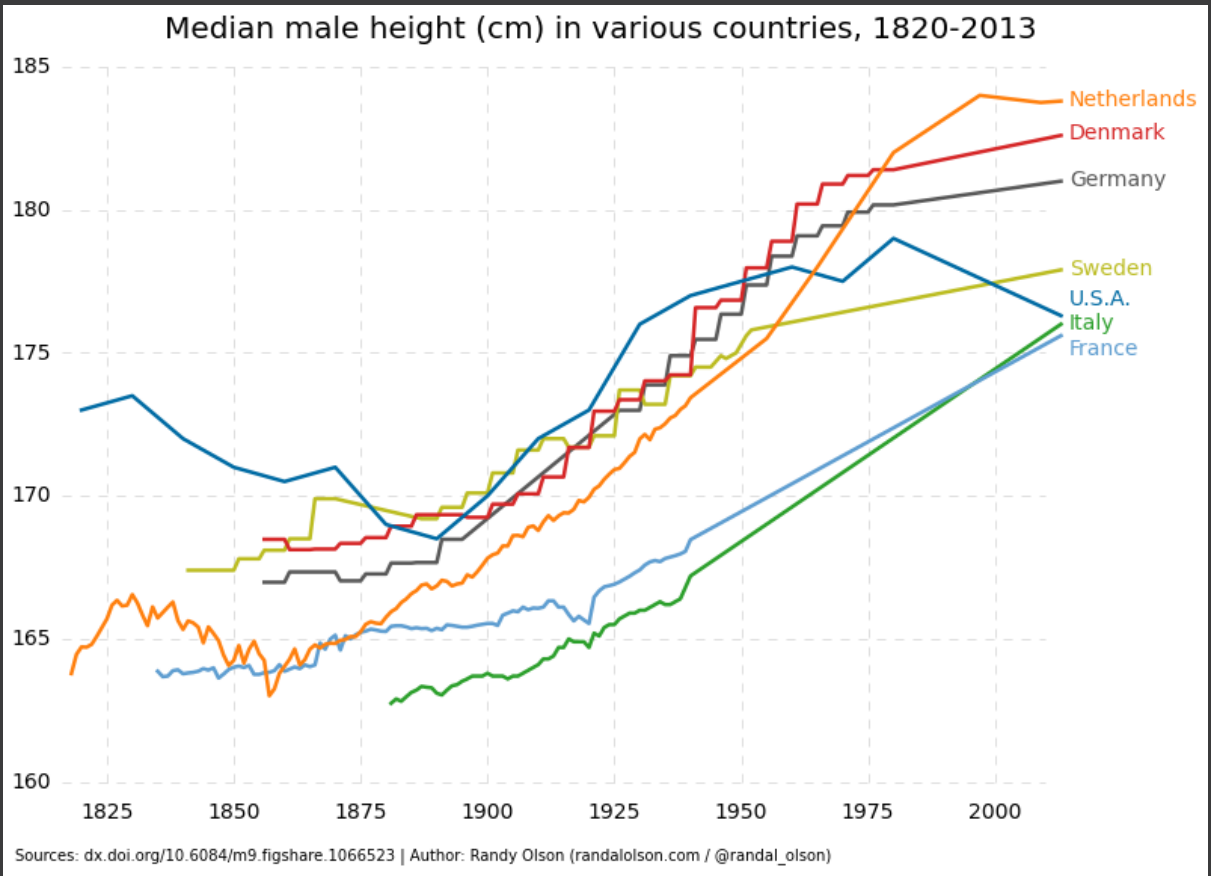

```
plt.figure(figsize=[20, 8]);
plt.subplot(2,1,1);
plt.hist(np.asarray(cities.dropna().Population/1000), bins=np.arange(0, 200, 1), normed=1);
plt.subplot(2,1,2);
plt.hist(np.asarray(dutch_cities.dropna().Population/1000), bins=np.arange(0, 200, 1), normed=1);
## add the subplot of the world cities below this Dutch one
```

Je ziet dat in het buitenland meer grote steden zijn en dat in Nederland de kleine steden een groter aandeel hebben in de verdeling.

Assignment 3

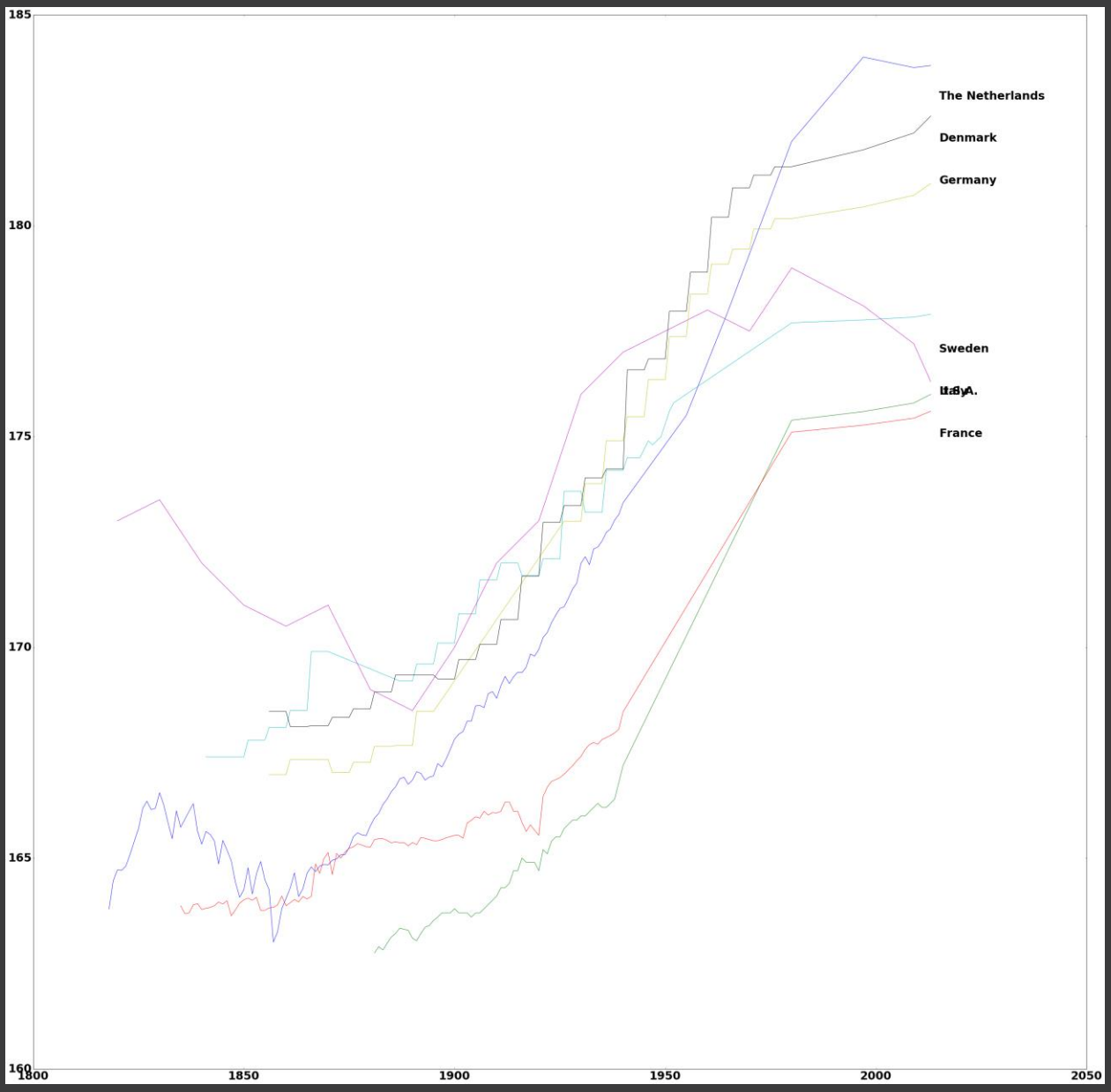
https://github.com/Floor9494/ADS_DV/blob/master/DV-assignment-03.ipynb

Maak de volgende grafiek na:




```
import pandas as pd
import numpy as np
import matplotlib
import matplotlib.pyplot as plt
%matplotlib inline

plt.figure(figsize=[50, 50])
font = {'weight': 'bold', 'size': 30}
matplotlib.rc('font', **font)

heights = pd.read_csv('http://files.figshare.com/1545826/world_heights.csv')
heights_cleaned = heights.interpolate()

df = pd.DataFrame(heights_cleaned)
hc = heights_cleaned.sort_values(by='Year', ascending=False).head(1)
for ind, x in hc.iterrows():
    for idx, i in enumerate(x):
        if(i < 200):
            h = int(float(i))
            n = hc.columns[idx]
            x = df['Year'].tolist()
            y = df[n].tolist()
            plt.plot(x, y)
            plt.text(2015, h, n)
plt.show()
```

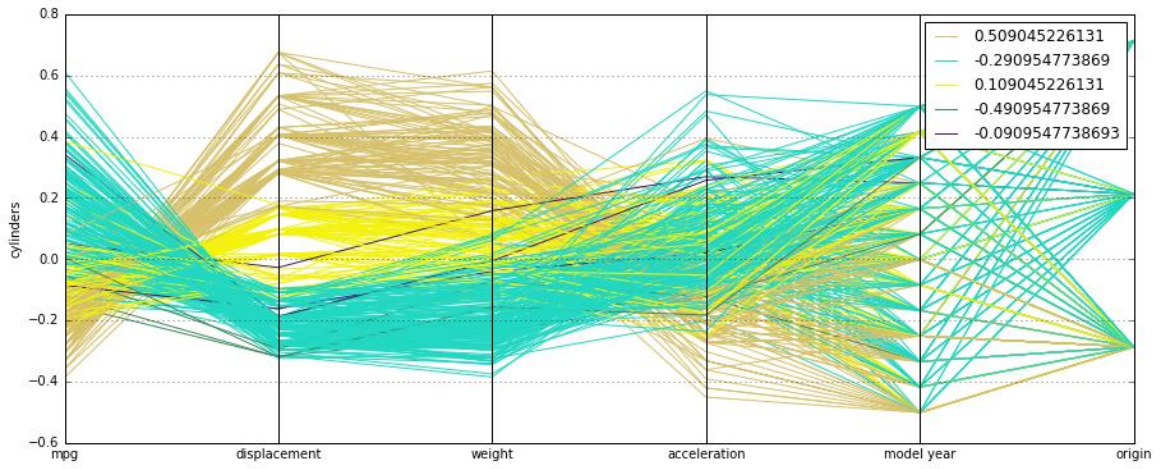
Nagemaakte gafiek:



Assignment 4

https://github.com/Floor9494/ADS_DV/blob/master/DV-assignment-04.ipynb

Creëer een parallel coördinaten plot.

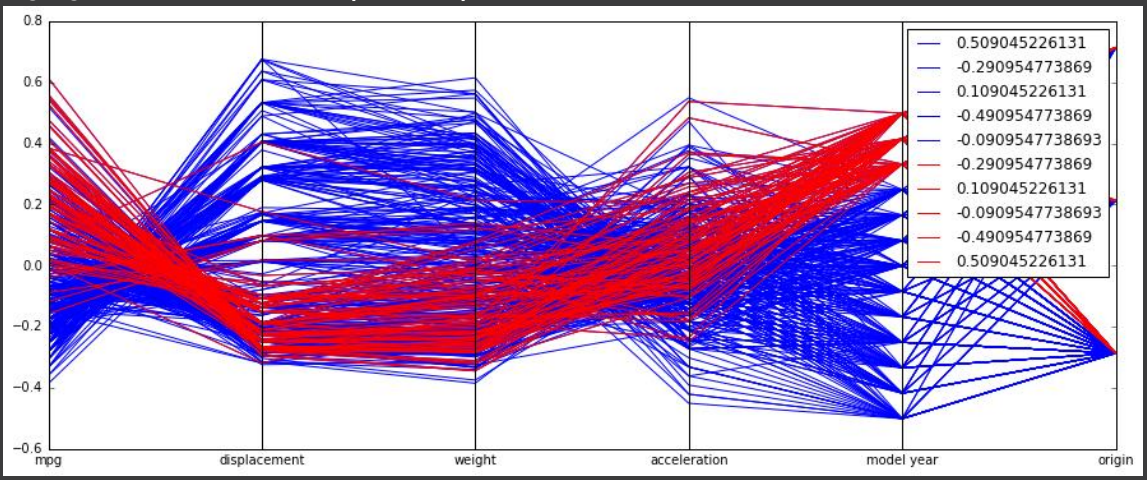



```
# The data file is quite nasty with several different delimeters that read_csv cannot handle very well
names=['mpg','cylinders','displacement','horsepower','weight','acceleration','model year','origin','car na
me','j','k','l','m','n']
cars = pd.read_csv('https://archive.ics.uci.edu/ml/machine-learning-databases/auto-mpg/auto-mpg.data', del
imiter=r"\s+", names=names, header=None, engine='python')
# Create a subset of dataset with all useful features
cars = cars.iloc[:,[0,1,2,4,5,6,7]]

# Create a normalized dataset
cars_norm = (cars - cars.mean()) / (cars.max() - cars.min())
```

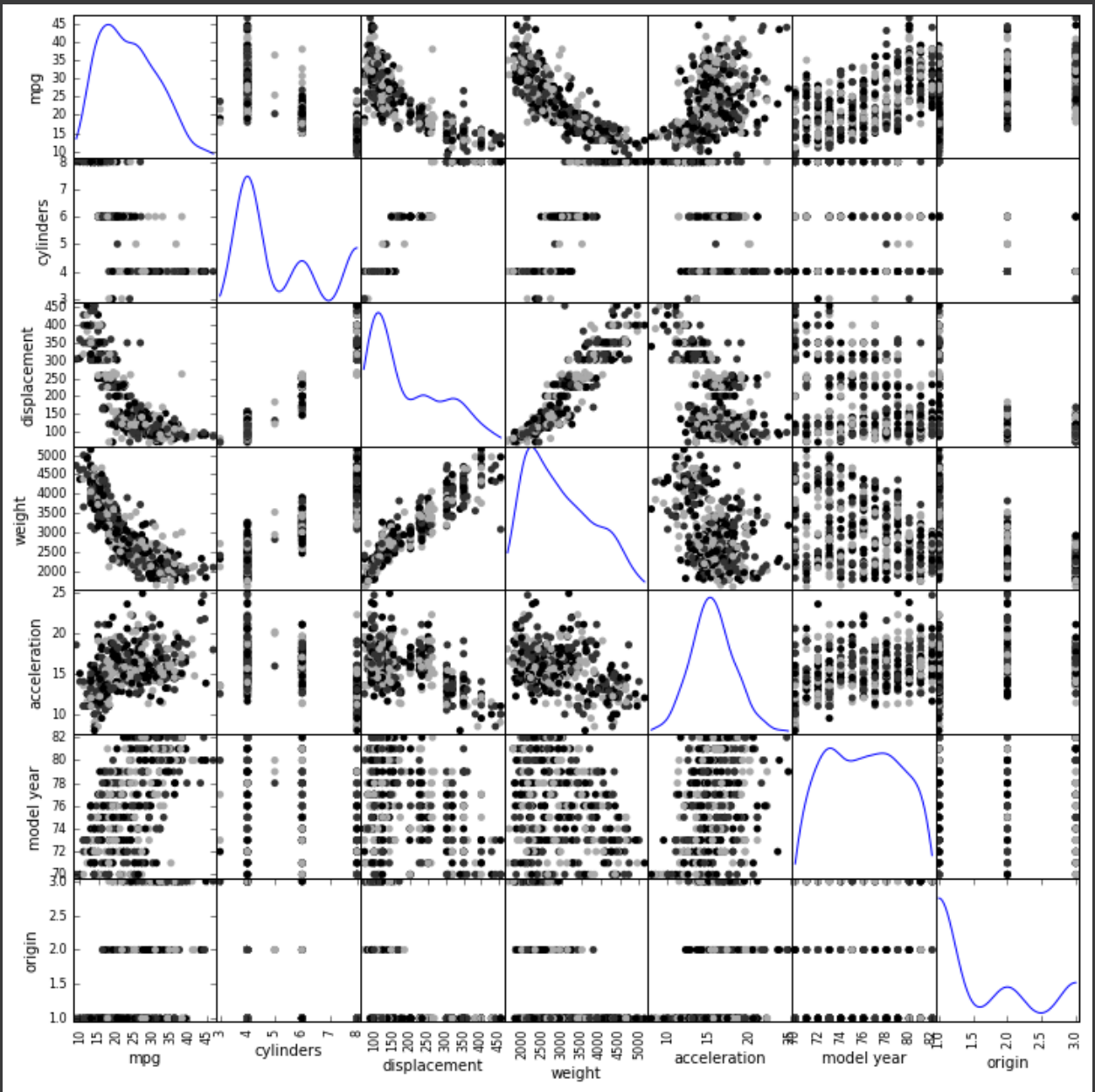
```
# Create the parallel coordinates plot here
fig = plt.figure(figsize=[15,6])
ax = parallel_coordinates(cars_norm,'cylinders')
ax.set_ylabel('cylinders');
```

(Het verband tussen gewicht en acceleratie is negatief. Hoog gewicht > Lage acceleratie. Te zien midden in de grafiek)

Highlight de auto's die ouder zijn dan 80 jaar.



Creëer nu van deze dataset een scatter matrix.

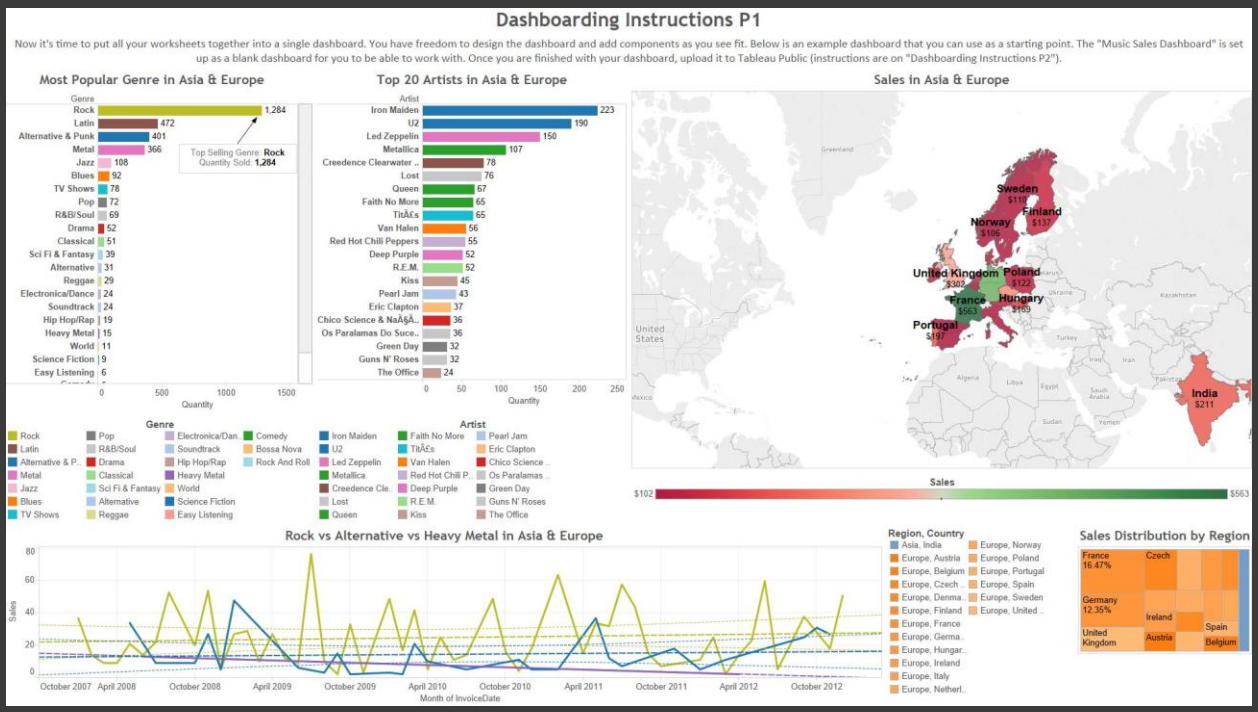



```
## Create the scatter matrix here
scatter_matrix(cars, alpha=1, figsize=(12, 12), diagonal='kde', c=['#aaaaaa','#333333','#000000'], s=100,
linewidth=0);
```

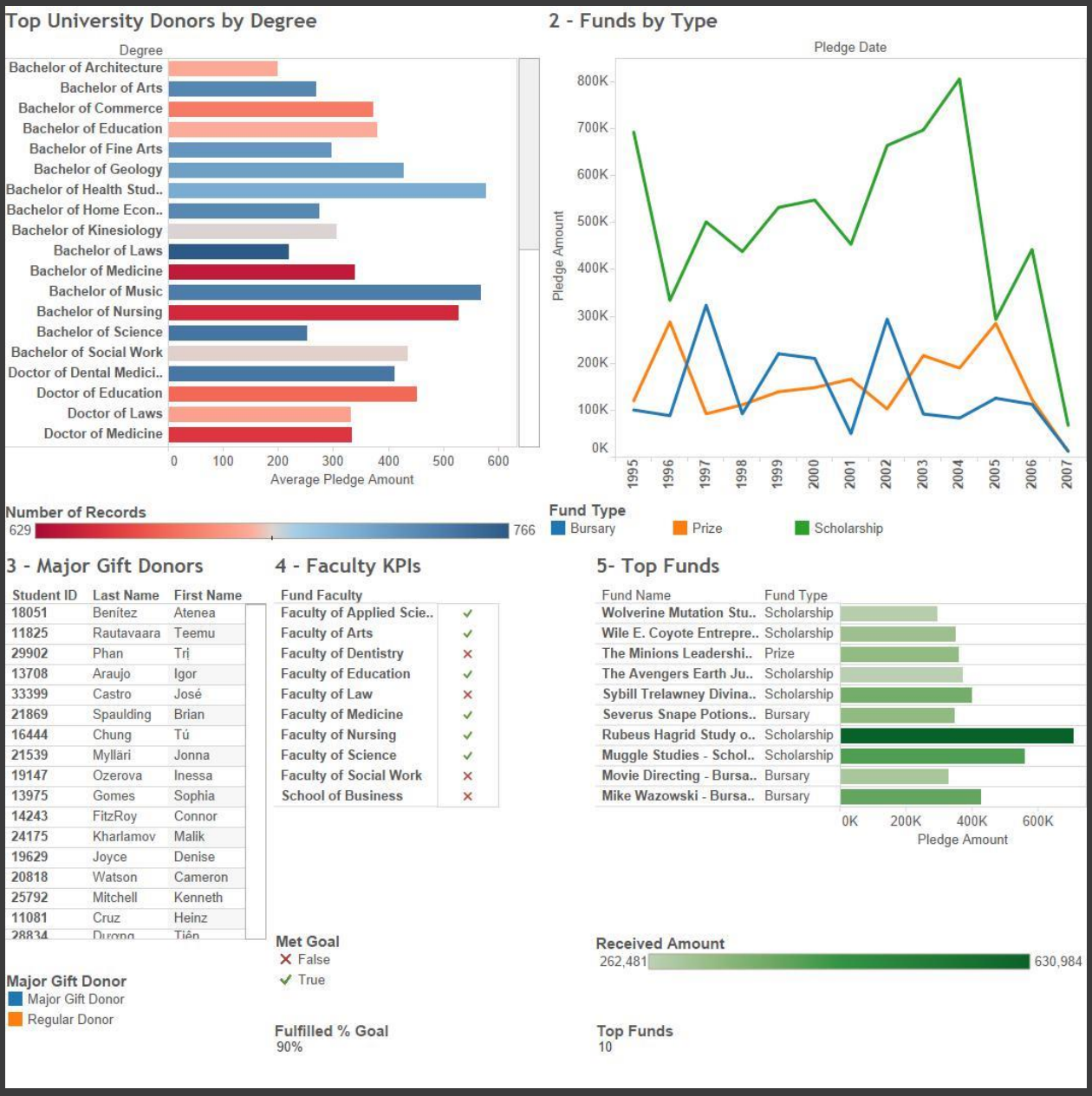
Assignment 5

https://github.com/Floor9494/ADS_DV/blob/master/DV-assignment-05.ipynb

Dashboard 1

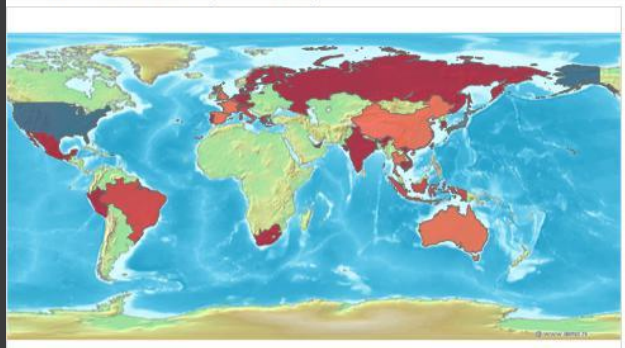


Dashboard 2


Top University Donors by Degree
Degree
Bachelor of Architecture
Bachelor of Arts
Bachelor of Commerce
Bachelor of Education
Bachelor of Fine Arts
Bachelor of Geology
Bachelor of Health Stud..
Bachelor of Home Econ..
Bachelor of Kinesiology
Bachelor of Laws
Bachelor of Medicine
Bachelor of Music
Bachelor of Nursing
Bachelor of Science
Bachelor of Social Work
Doctor of Dental Medici..
Doctor of Education
Doctor of Laws
Doctor of Medicine
0 100 200 300 400 500 600
Average Pledge Amount
Number of Records
629 766
2 - Funds by Type
Pledge Date
Pledge Amount
800K 700K 600K 500K 400K 300K 200K 100K 0K
1995 1996 1997 1998 1999 2000 2001 2002 2003 2004 2005 2006 2007
Fund Type
Bursary
Prize
Scholarship
3 - Major Gift Donors
Student ID Last Name First Name
18051 Benitez Atenea
11825 Rautavaara Teemu
29902 Phan Tri
13708 Araujo Igor
33399 Castro José
21869 Spaulding Brian
16444 Chung Tú
21539 Mylläri Jonna
19147 Ozerova Inessa
13975 Gomes Sophia
14243 FitzRoy Connor
24175 Kharlamov Malik
19629 Joyce Denise
20818 Watson Cameron
25792 Mitchell Kenneth
11081 Cruz Heinz
28834 Dương Tiến
Major Gift Donor
Major Gift Donor
Regular Donor
4 - Faculty KPIs
Fund Faculty
Faculty of Applied Scie..
Faculty of Arts
Faculty of Dentistry
Faculty of Education
Faculty of Law
Faculty of Medicine
Faculty of Nursing
Faculty of Science
Faculty of Social Work
School of Business
Met Goal
False
True
Fulfilled % Goal
90%
5- Top Funds
Fund Name Fund Type
Wolverine Mutation Stu.. Scholarship
Wile E. Coyote Entrepre.. Scholarship
The Minions Leadershi.. Prize
The Avengers Earth Ju.. Scholarship
Sybill Trelawney Divina.. Scholarship
Severus Snape Potions.. Bursary
Rubeus Hagrid Study o.. Scholarship
Muggle Studies - Schol.. Scholarship
Movie Directing - Bursa.. Bursary
Mike Wazowski - Bursa.. Bursary
0K 200K 400K 600K
Pledge Amount
Received Amount
262,481 630,984
Top Funds
10

Dashboard 3

1 - Restaurants By Country



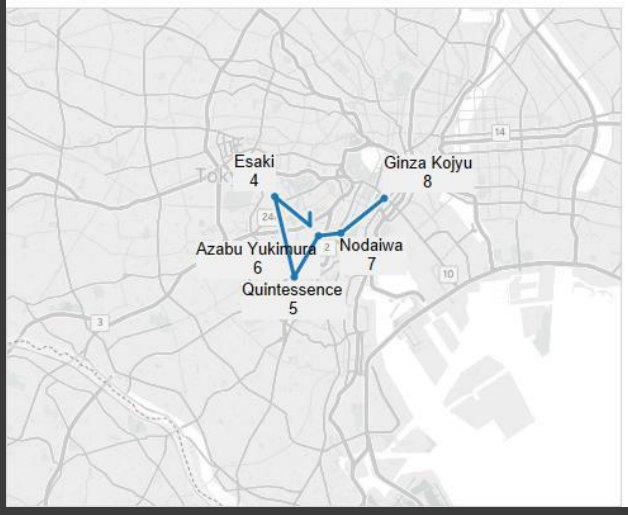
Number .. 1 9

2 - Michelin Restaurants - Europe

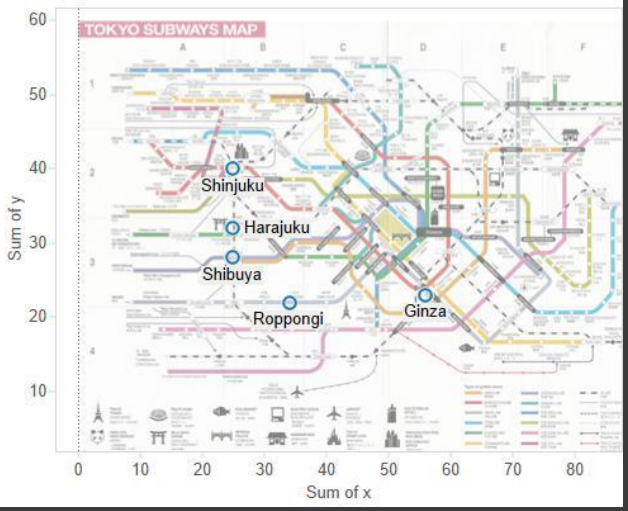
Count of.. 1 9

Region Europe Show history

3 - Japan Itinerary



4 - Areas Near Shibuya

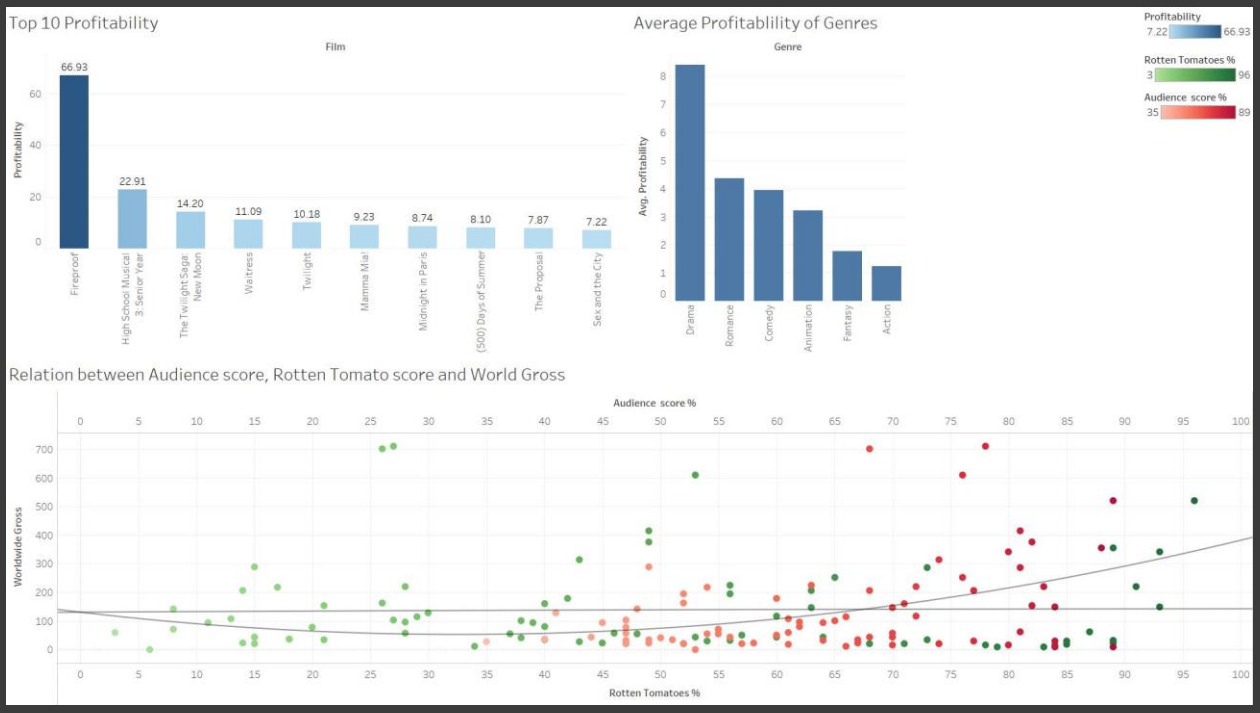


Assignment 6

https://github.com/Floor9494/ADS_DV/blob/master/DV-assignment-06.ipynb

Dashboard 1

- Wat zijn de top 10 meest winstgevende films.
- Meest winstgevende genre.
- Relatie tussen Rotten Tomato en populariteit.

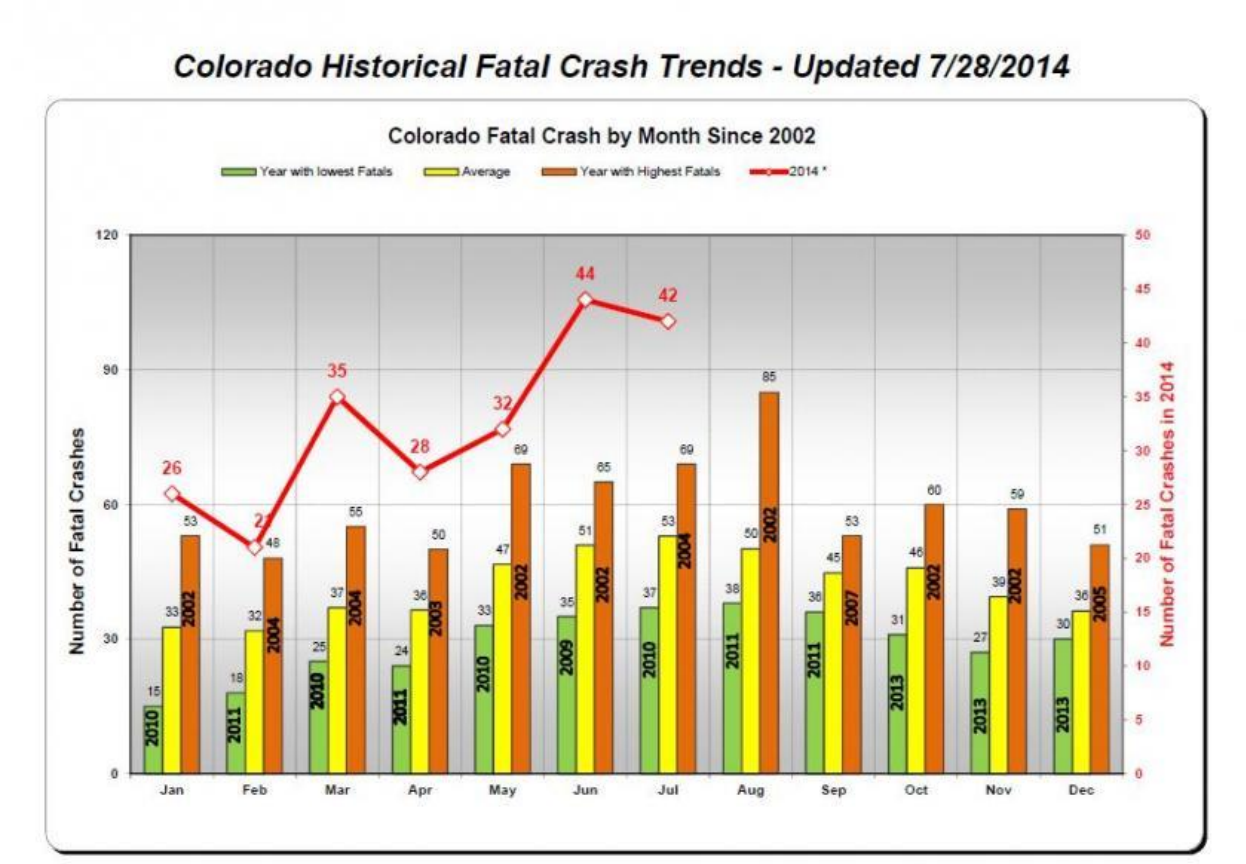


De rechte lijn geeft de relatie tussen Rotten Tomato en World Gross aan. De andere lijn de relatie tussen Audience Score.

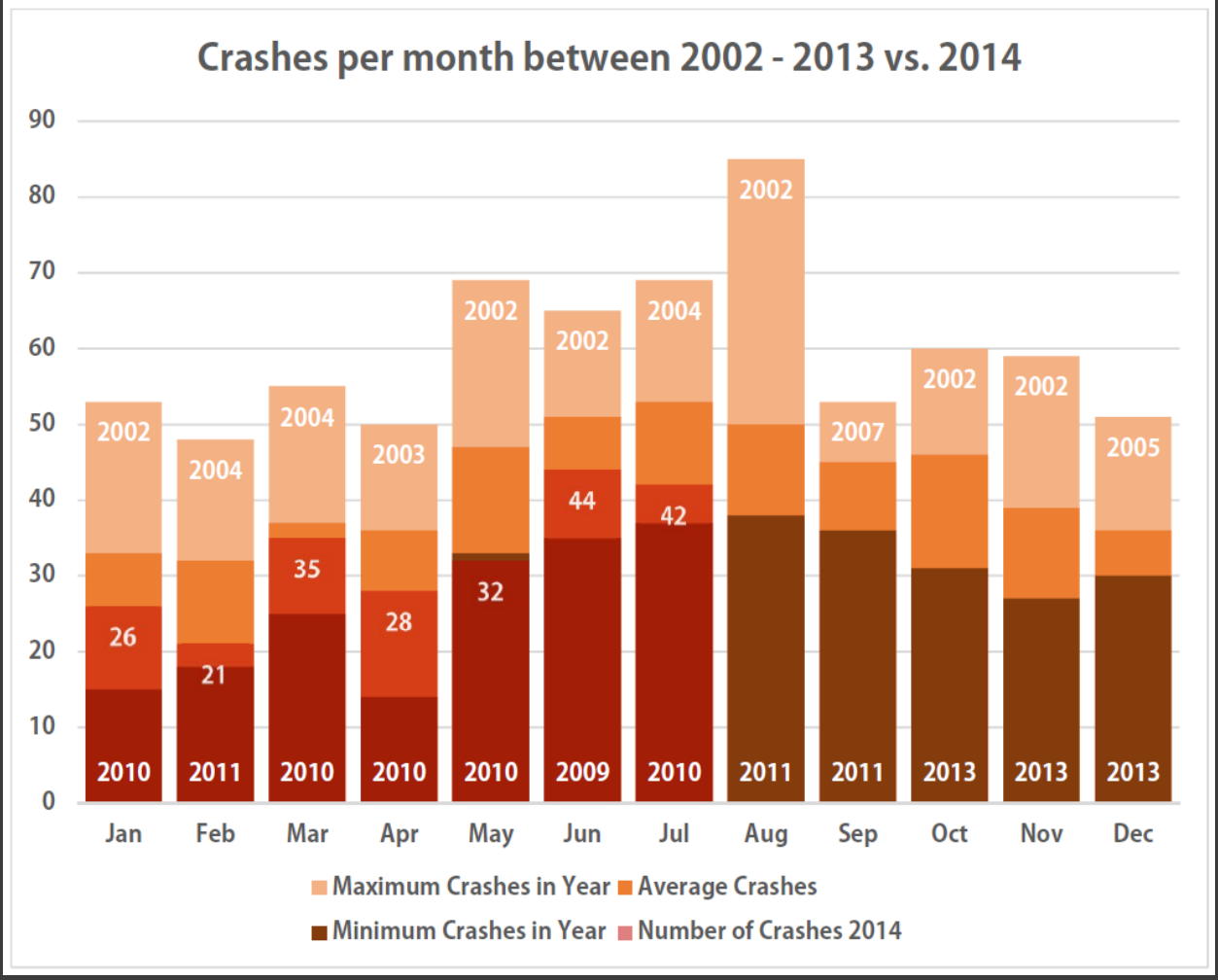
Assignment 7

https://github.com/Floor9494/ADS_DV/blob/master/DV-assignment-07.ipynb

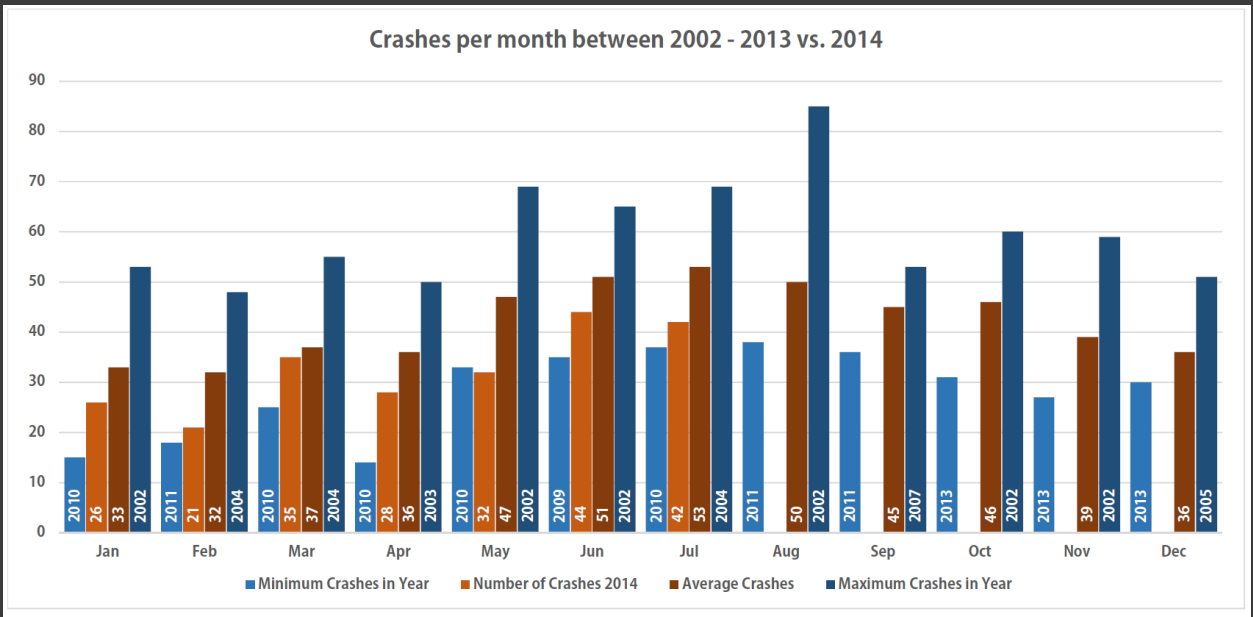
Verbeter een 'slechte' datavisualisatie.



Eerst heb ik de assen gesynchroniseerd. De 2014 lijn geeft hierdoor een heel scheef beeld. Mijn eerste idee was om de crashes te groeperen per maand en de grotere waardes naar achteren plaatsen, dat de waardes als het ware op elkaar lagen. Op zich was ik er best tevreden over, maar leek dit, na feedback, te veel op een stacked grafiek. In zo'n grafiek geeft elke kleur een gedeelte van het totaal aan, niet een continue waarde vanaf 0.



Hierna heb ik nog een versie gemaakt door nog een keer te kijken naar de originele grafiek, de beste onderdelen hieruit te halen en daarop verder te borduren. Dit was de overzichtelijkheid van het staafdiagram. Door de 2014 data toe te voegen aan het diagram is de data al iets duidelijker. De feedback erna was al een stuk positiever, want ze snapte daadwerkelijk wat de data zei.



POSTER

OPZOEK NAAR DATA

WORLD-FOOD-FACTS

Deze dataset leek op eerste inzicht vrij interessant, maar moest heel erg schoongemaakt worden. Ook zaten er weinig continue datapoints in, dus was het verbinden vinden vrij lastig zonder alleen maar te kijken naar hoe vaak records voorkomen.

TRAFFIC

Wederom weer een interessante dataset, maar deze was veel te groot. Tableau liep al constant vast bij het inladen van de data, dus nieuwe set.

MOVIE METADATA

De afgelopen maanden ben ik eigenlijk steeds meer films gaan kijken, omdat ik erachter kwam dat ik veel 'klassiekers' nog niet gezien had. Om deze klassiekers te vinden heb ik veel de top 200 van IMDB gebruikt. Dus toen ik deze data tegenkwam was ik wel geïnteresseerd.

Het vinden van relaties tussen data was helaas wel wat minder. Na alles tegen elkaar af te zetten, bleven er maar een paar data punten over. Het niet vinden van de correlaties kan ook liggen aan de limitaties van (mij met) tableau. Bijvoorbeeld meerdere assen samenvoegen (2 kan, maar meer niet).

Ik wilde sowieso testen wat het verband tussen budget en gross was, maar ook wat de invloed van IMDB-score was. Verder vond ik ook wel een aardig verband tussen critic- en uservotes. Ook vroeg ik mij laatst af of films steeds langer werden, dit heb ik allemaal proberen te verwerken in 1 poster.

Version 1.0

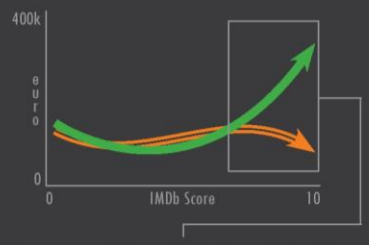
https://github.com/Floor9494/ADS_DV/blob/master/poster/Poster.pdf

Feedback

- Titel niet goed leesbaar.
- Geen zichtbare correlatie tussen data.
- Erg leeg.

Version 2.0

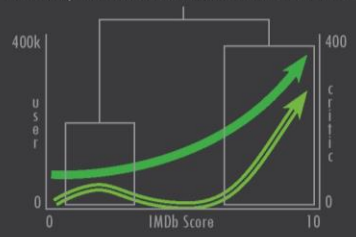
https://github.com/Floor9494/ADS_DV/blob/master/poster/Poster%20v2.0.pdf



Budget
Gross

Here we can see that, generally, a higher rated movie has a higher budget, but this does not mean it has a higher gross. Quite the opposite actually.

Next to the fact that higher rated movies have more reviews (everyone wants to see high rated movies), there is an increase at movies that have an IMDb rating around 3. Also noteworthy is the lack of user reviews at 'mediocre' films.



Critic Reviews
User Reviews

THE IMDb EFFECT

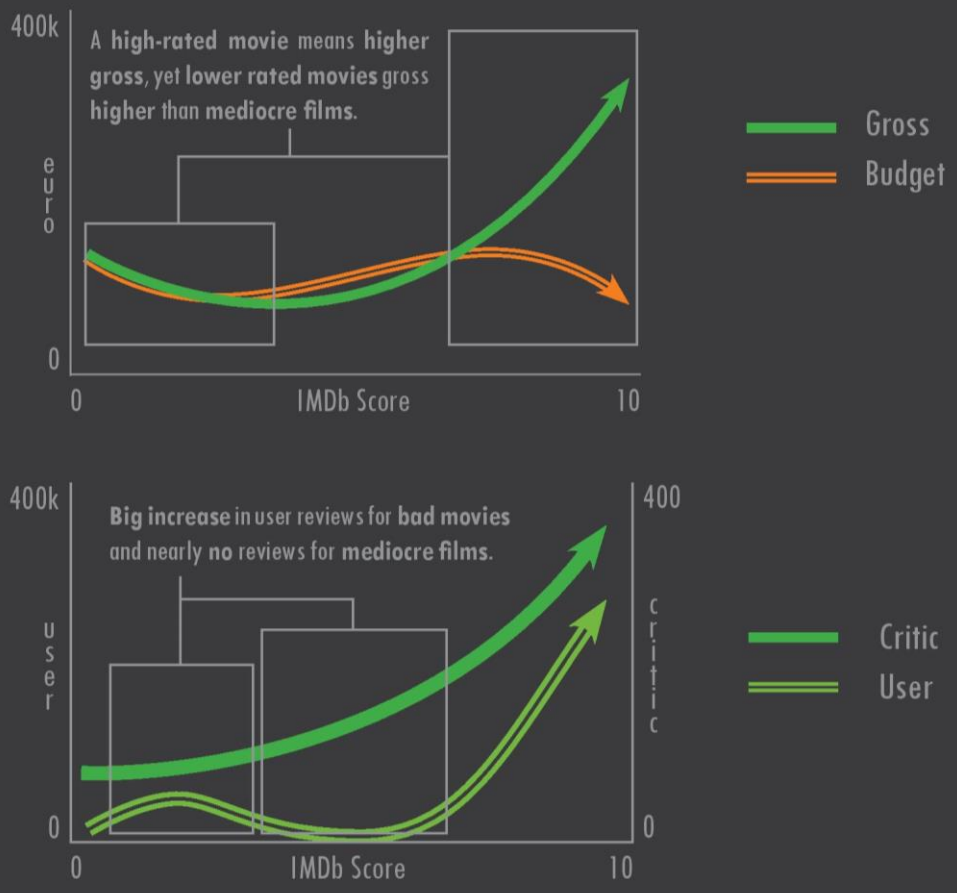
info from 5000 movies
over 100 years

Feedback

- Minder tekst (want, DATA-visualisatie).
- Grotere grafieken.
- Ondertitel duidelijker maken.

Version 2.1

https://github.com/Floor9494/ADS_DV/blob/master/poster/Poster%20v2.1.pdf

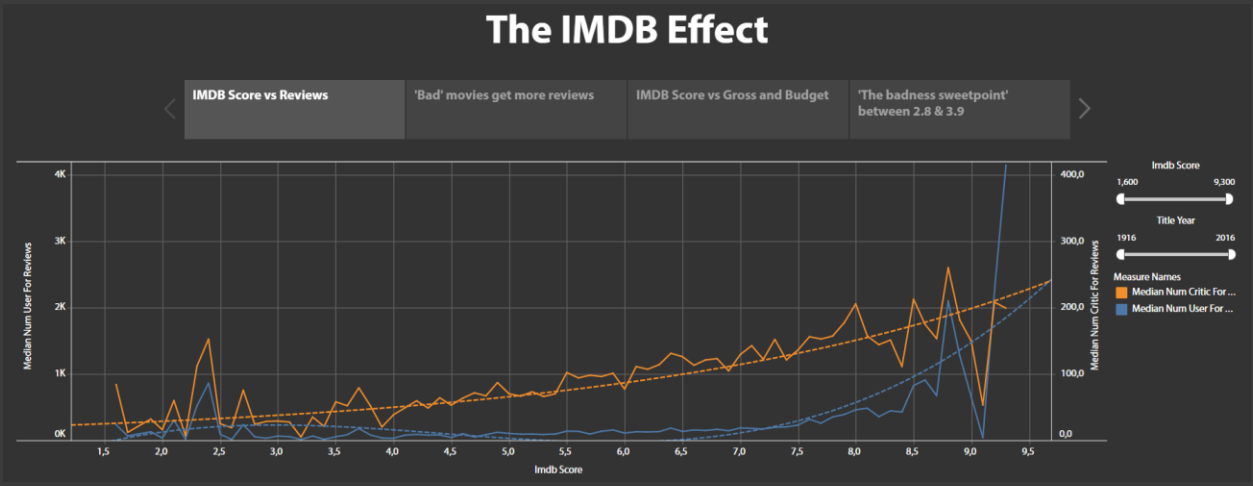


THE IMDb EFFECT

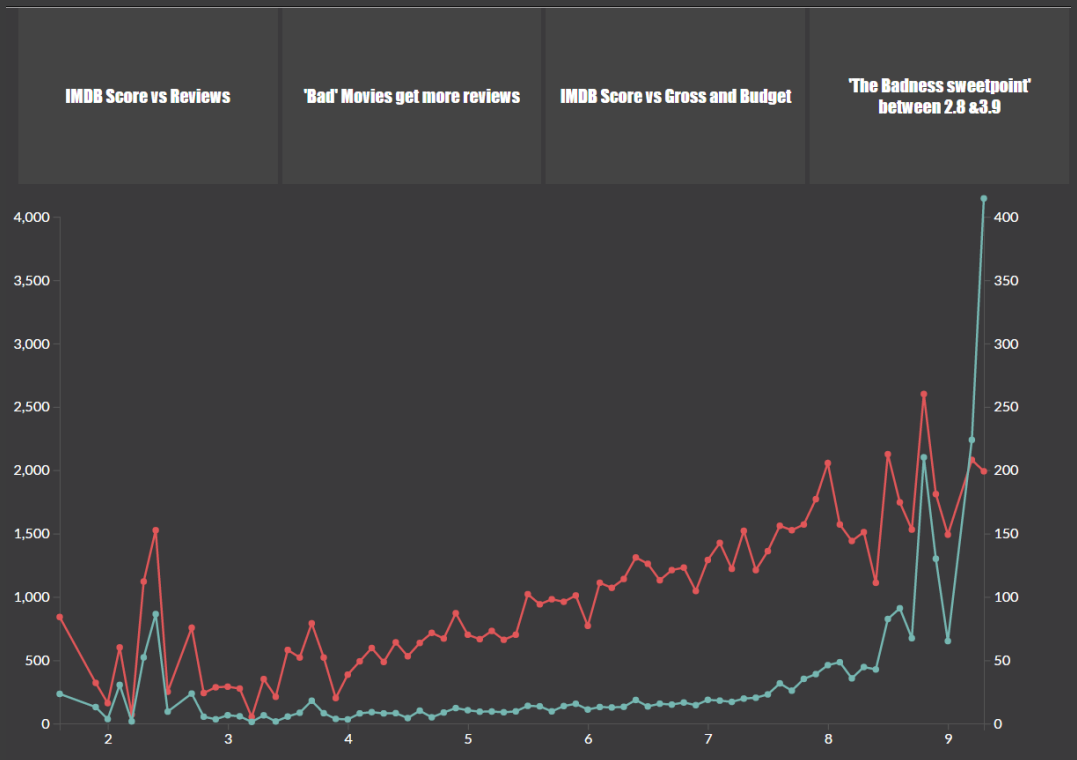
info from 5000 movies
over 100 years

DATA VISUALISATION

https://public.tableau.com/profile/publish/DVPosterData/Story1/



http://athena.fhict.nl/users/i306476/ADS_DV/



Bij het onderzoeken naar de data van mijn poster liep ik tegen een aantal limitaties van tableau aan. Ergens wist ik dat dit makkelijker zou moeten kunnen, dus leek het mij bij de interactieve data-visualisatie interessant dit te vergelijken met bijvoorbeeld D3js.

Tableau

Voor

- Heel snel data tegen elkaar afzetten.
- Heel goed in dataverwerking (mean, min, max, etc.)
- Goed en snel met filters.
- Makkelijk aan te leren.

Tegen

- Als je meer data met elkaar wil vergelijken moet je vaak meerdere grafieken maken. Alles in 1 grafiek wordt heel lastig gemaakt.
- CSV-export met veel omwegen half mogelijk.
- Kleuren en fonts onhandig aan te passen.
- Weinig support online.

D3js

Voor

- Mogelijkheden zijn praktisch eindeloos.
- Veel documentatie en hulp voor te vinden.

Tegen

- Steile leercurve.
- Eigenlijk alleen maar een visualisatie tool, niet echt voor dataverwerking.