Matplotlib



BIN-OWE2

Studiewijzer

Les	Onderwerp	
	Algemeen	**Python**
1	Debuggen en testen	Exceptions
2		Matplotlib
3		Datastructuren
4		Regular Expressions
5		Object Oriëntatie
6		Recursie
7		GUI programmeren

Doelstellingen

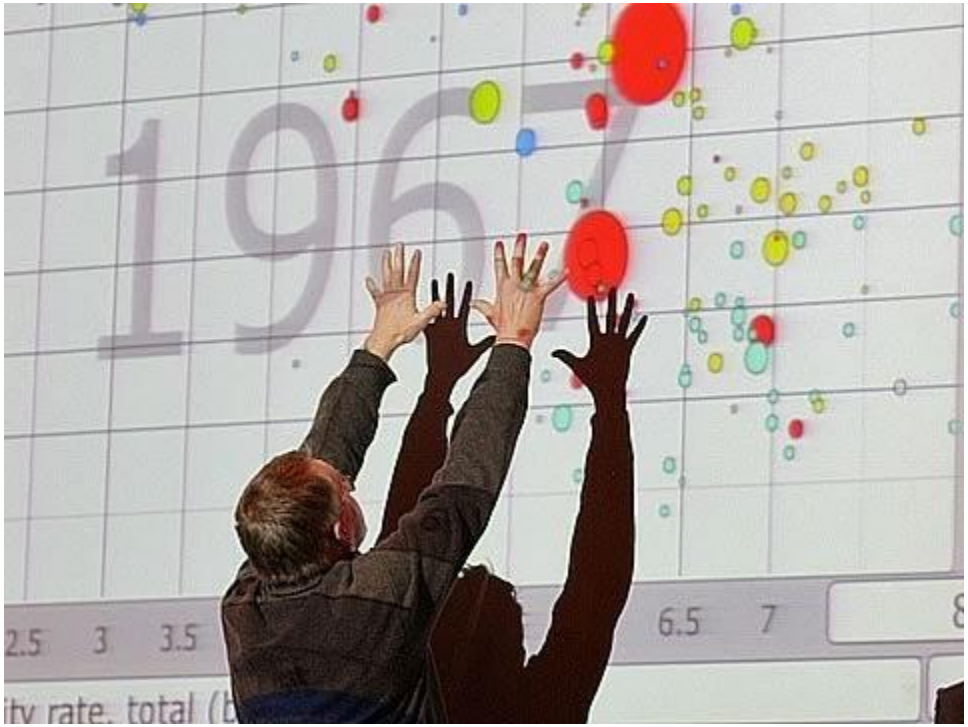
- Aan het eind van deze les kan je
  - Een simpele grafiek maken met behulp van Matplotlib
  - Een aantal grafiek eigenschappen veranderen

- Aan het eind van deze week begrijp je
  - Hoe verschillende grafieken van Matplotlib hun input verwachten
  - Hoe je Matplotlib kunt installeren

Inhoud

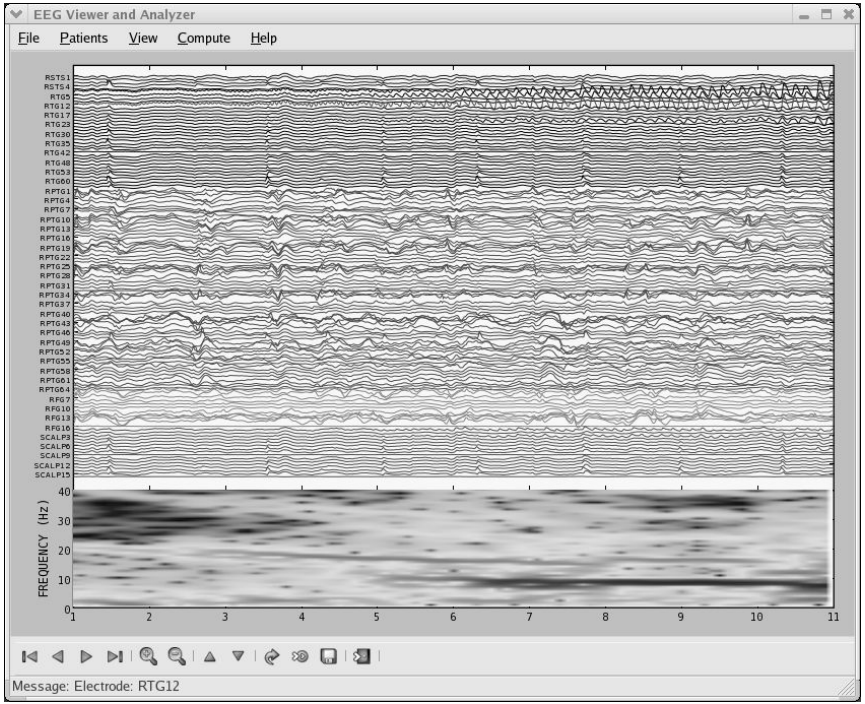
- Introductie Matplotlib
- Installatie Matplotlib
- Verschillende grafiek types
  - Scatter
  - Lijn
  - Bar
  - Taart
- Subplots en grafiek eigenschappen

Statistiek is leuk!



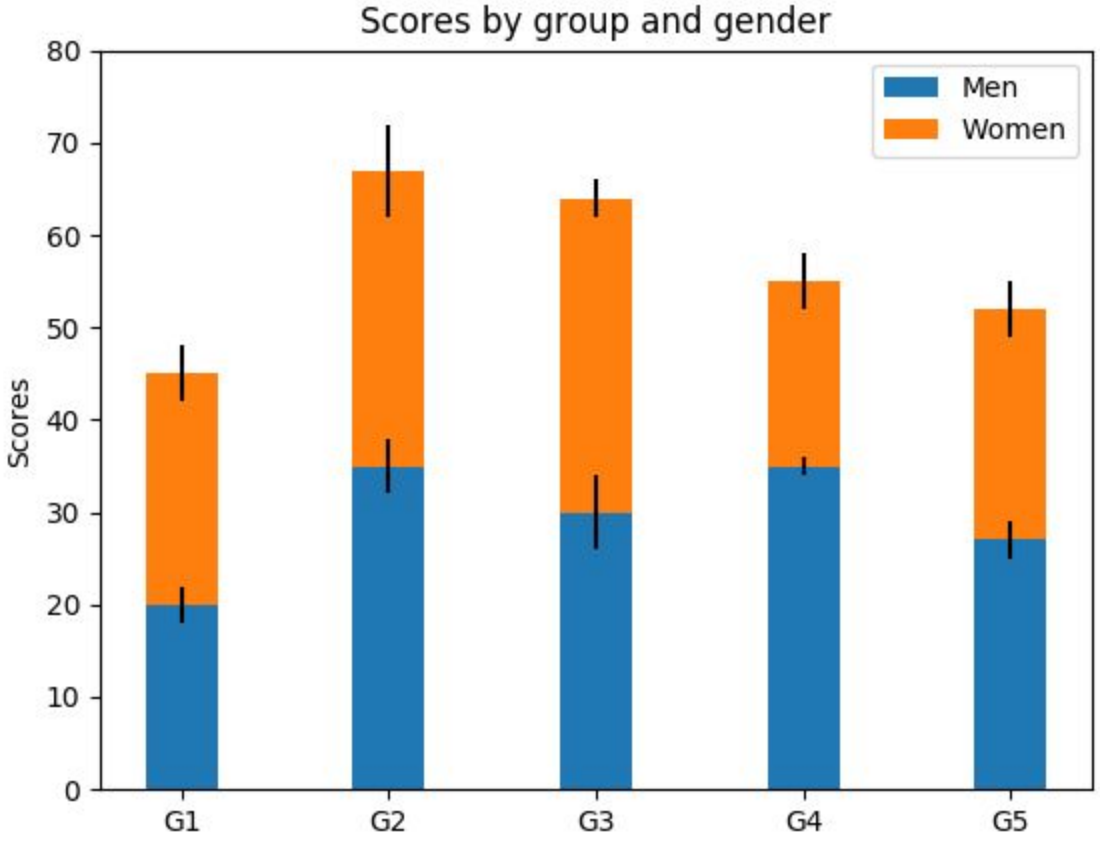
Matplotlib's nederige begin

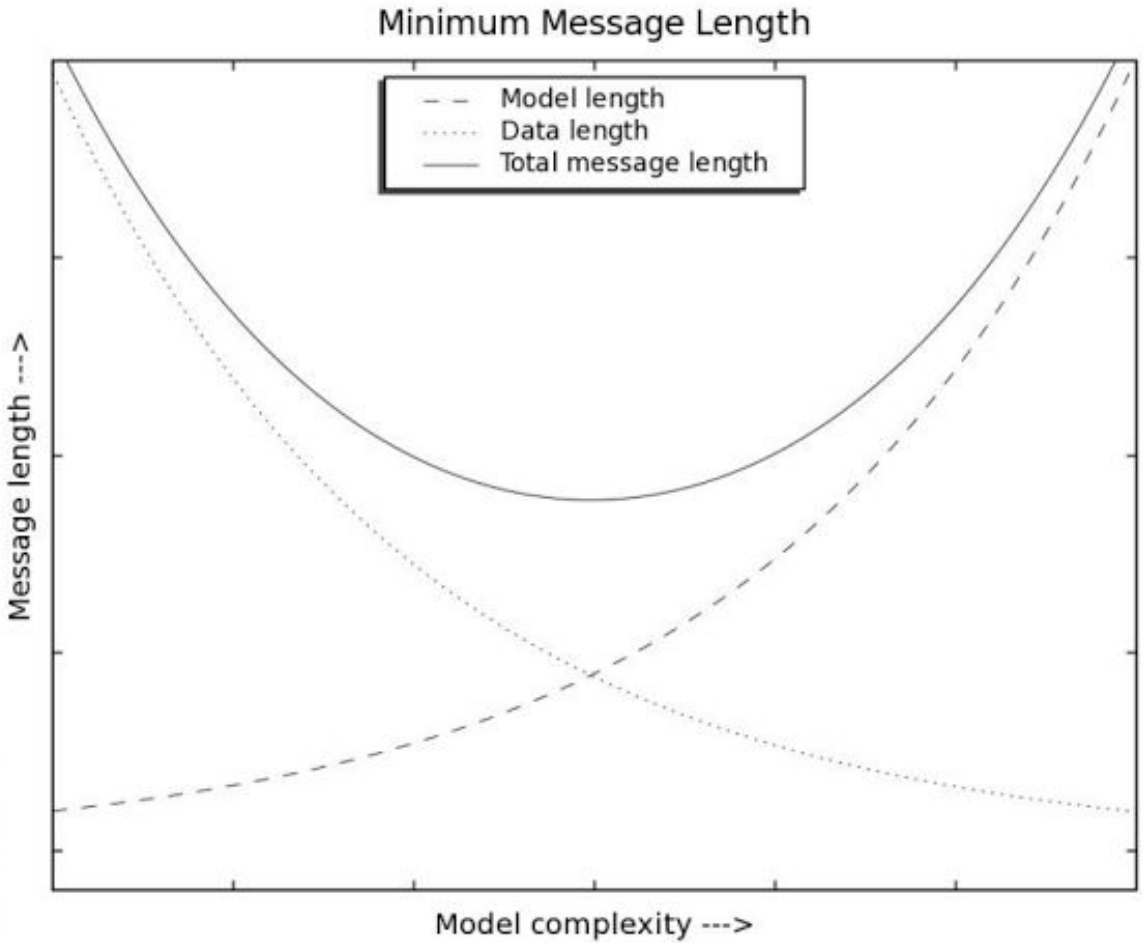
Begon vanuit de behoefte om EEG data weer te geven zonder dure software -> het dongel probleem

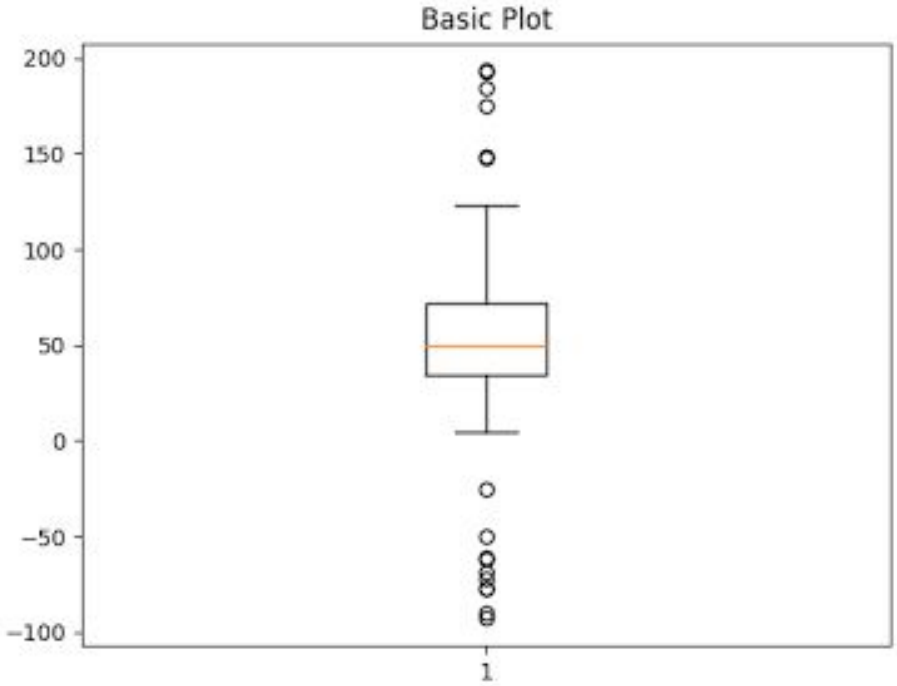


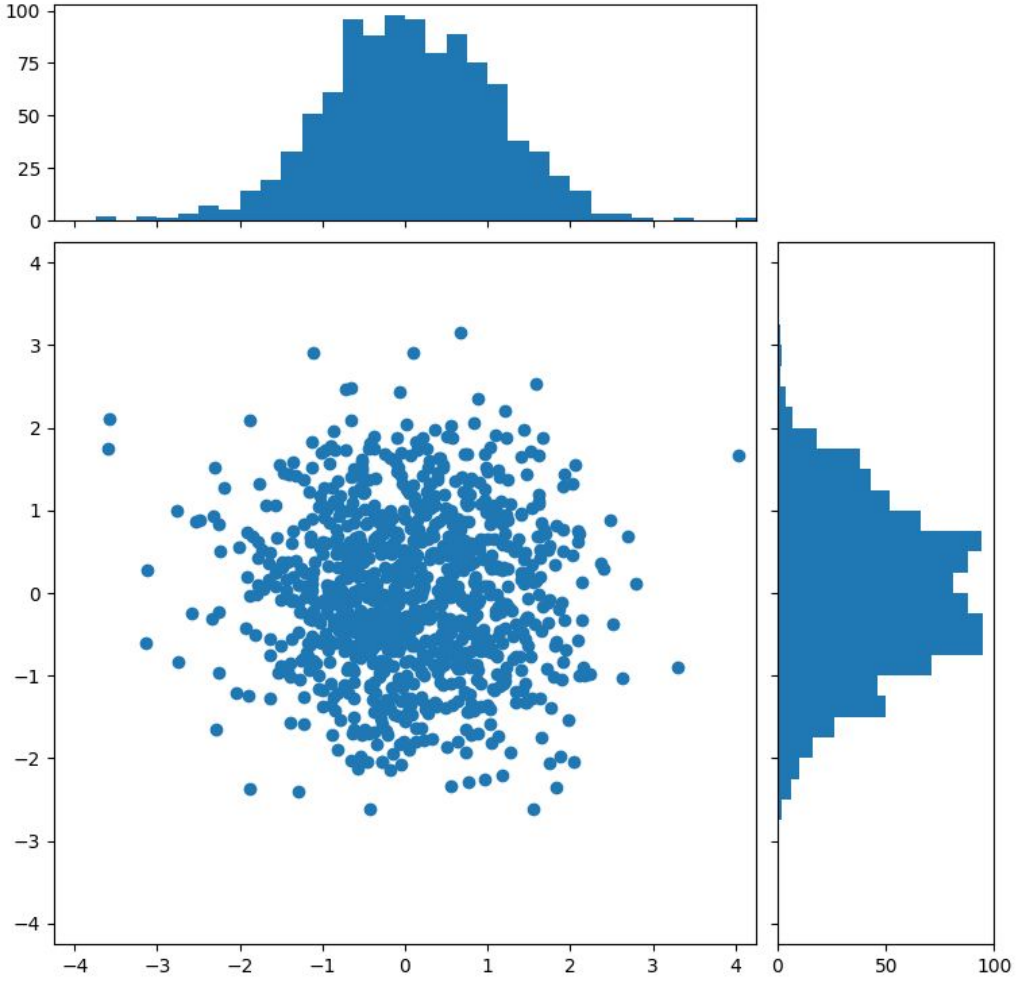
HAN

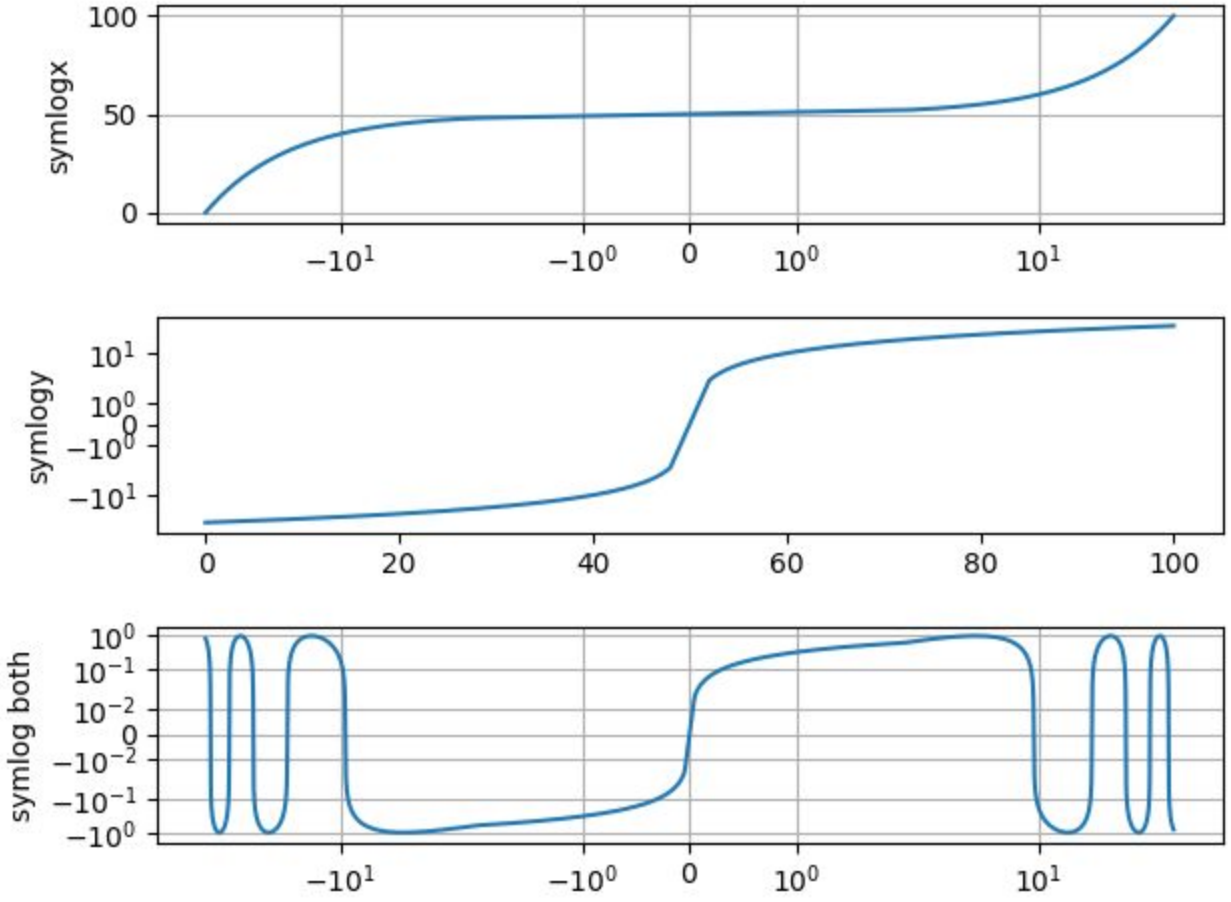
Bron: http://www.aosabook.org/en/matplotlib.html

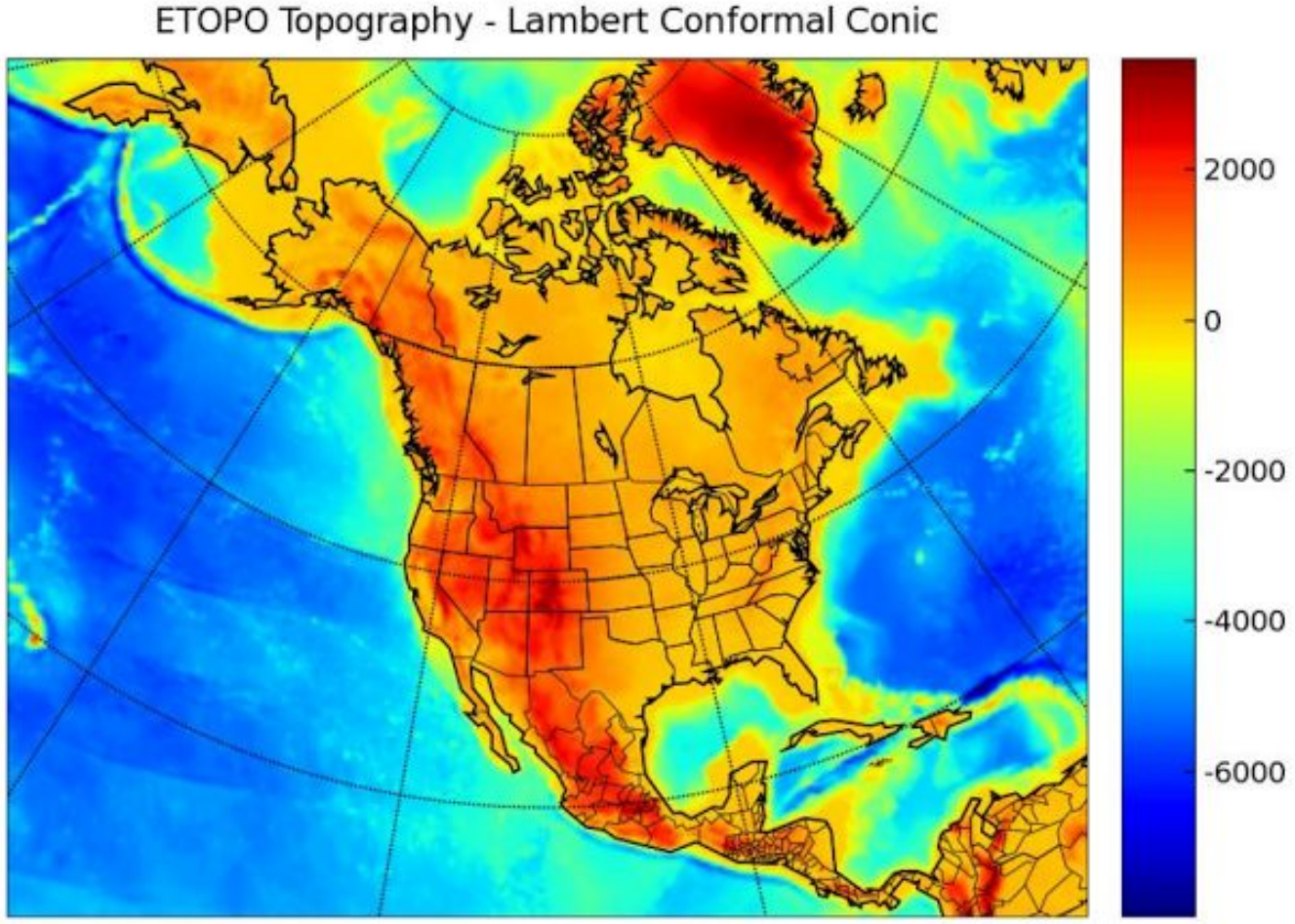

Scores by group and gender
Men
Women
Scores
80
70
60
50
40
30
20
10
0
G1
G2
G3
G4
G5


Minimum Message Length
Model length
Data length
Total message length
Message length --->
Model complexity --->

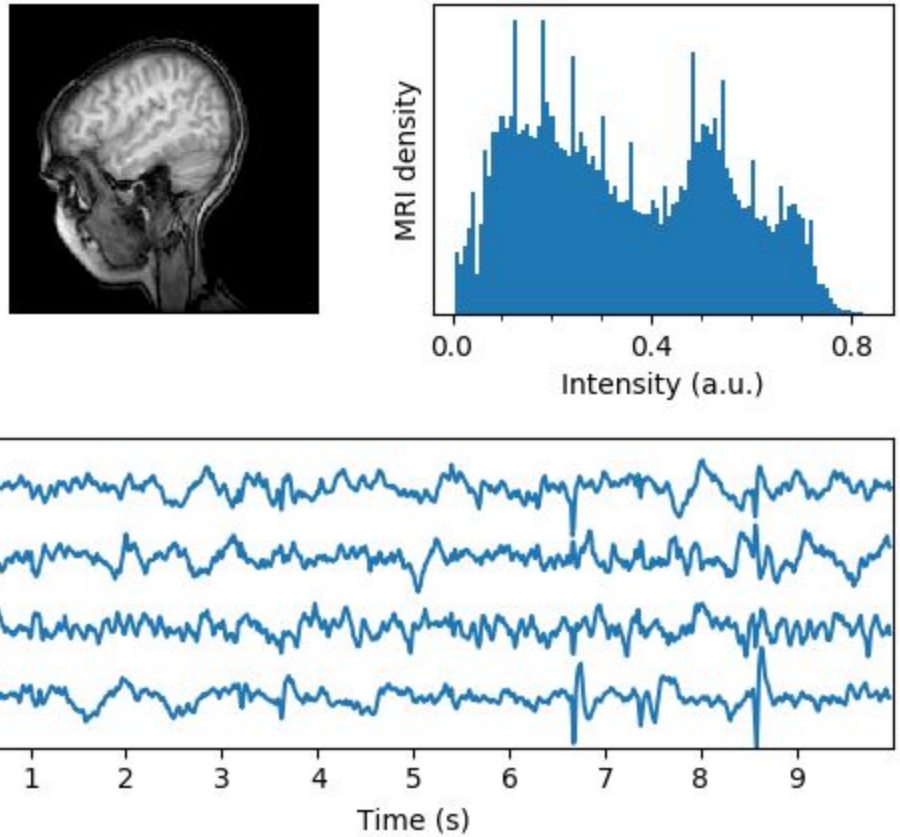

Basic Plot
200
150
100
50
0
−50
−100
1


100
75
50
25
0
4
3
2
1
0
−1
−2
−3
−4
−4
−3
−2
−1
0
1
2
3
4
0
50
100

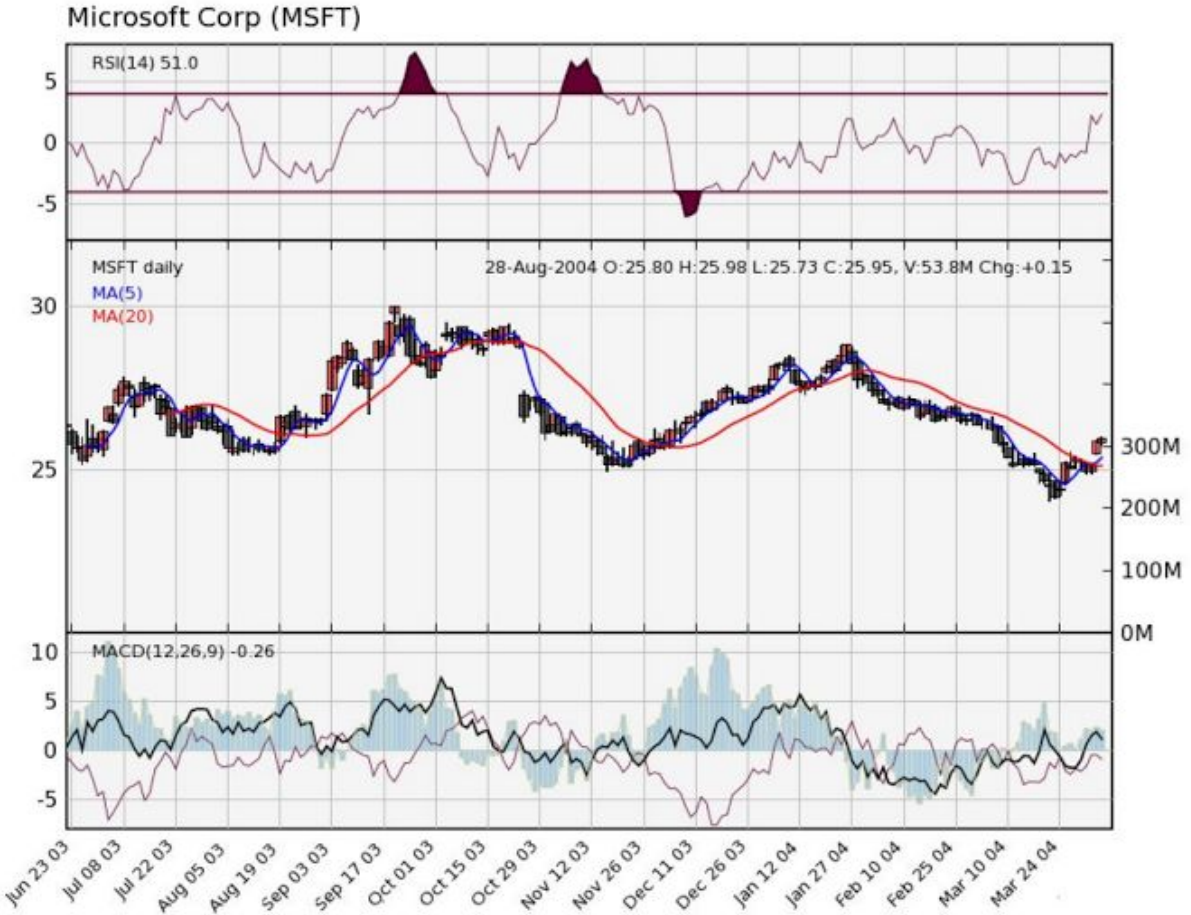

symlogx
100
50
0
-10^1
-10^0
0
10^0
10^1
symlogy
10^1
10^0
0
-10^0
-10^1
0
20
40
60
80
100
symlog both
10^0
10^{-1}
10^{-2}
0
-10^{-2}
-10^{-1}
-10^0
-10^1
-10^0
0
10^0
10^1


ETOPO Topography - Lambert Conformal Conic
2000
0
-2000
-4000
-6000

HAN



Bron: http://matplotlib.org/examples/pylab_examples/mri_with_eeg.html


Microsoft Corp (MSFT)
RSI(14) 51.0
5
0
-5
MSFT daily
28-Aug-2004 O:25.80 H:25.98 L:25.73 C:25.95, V:53.8M Chg:+0.15
MA(5)
MA(20)
30
25
300M
200M
100M
0M
MACD(12,26,9) -0.26
10
5
0
-5
Jun 23 03
Jul 08 03
Jul 22 03
Aug 05 03
Aug 19 03
Sep 03 03
Sep 17 03
Oct 01 03
Oct 15 03
Oct 29 03
Nov 12 03
Nov 26 03
Dec 11 03
Dec 26 03
Jan 12 04
Jan 27 04
Feb 10 04
Feb 25 04
Mar 10 04
Mar 24 04

HAN

Inhoud

- Introductie Matplotlib
- Installatie Matplotlib
- Verschillende grafiek types
  - Scatter
  - Lijn
  - Bar
  - Taart
- Subplots en grafiek eigenschappen

Installatie

- Ga naar http://matplotlib.org/users/installing.html
- Er zijn een aantal opties:
  - Downloaden en uitvoeren
  - Pip
  - Etc.
- **Op deze pagina staan ook dependencies genoemd.**
  - Afhankelijk van je installatie tactiek moet je ook de dependencies installeren

Opdracht 1 (5 - 20 min)

Controleer of je matplotlib hebt

(import matplotlib)

Zo nee: installeer matplotlib

Inhoud

- Introductie Matplotlib
- Installatie Matplotlib
- Verschillende grafiek types
  - Scatter
  - Lijn
  - Bar
  - Taart
- Subplots en grafiek eigenschappen

Verschillende grafieken

- **Grofweg zijn er voor nu een paar belangrijke grafieken te onderscheiden:**
  - Scatterplot
  - Lijndiagram
  - Staafdiagram
  - Taartdiagram

- **Natuurlijk zijn er veel meer, zoals al te zien in de introductie**
  - Boxplots
  - Intensiteit
  - Plaatjes
  - Etc.

Begin

- Vaak nodig:
  - `import matplotlib.pyplot as plt`
  - `from random import randint`
  - `range()`

- Data
  - X data
  - Y data
- Deze matchen op basis van index
  - Dus moeten ze even lang zijn!

Data parsing voor een grafiek

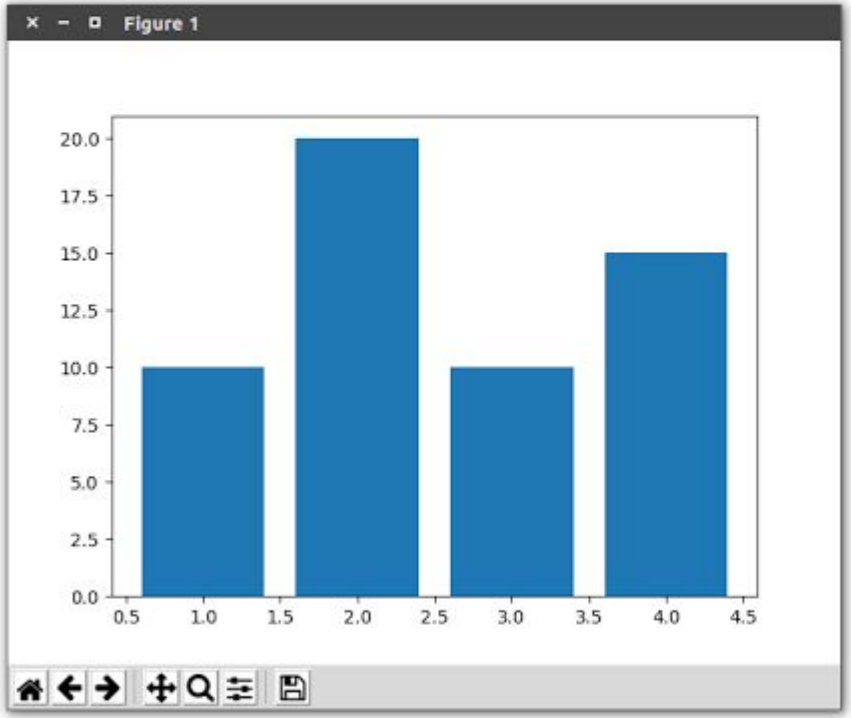
- Vorig blok gehad over:
  - Lijsten
  - 2D lijsten
  - Bestanden

- Dit blok ook daarbij:
  - Dictionaries
  - Sets

- Eigenlijk kan je iedere datastructuur gebruiken als basis

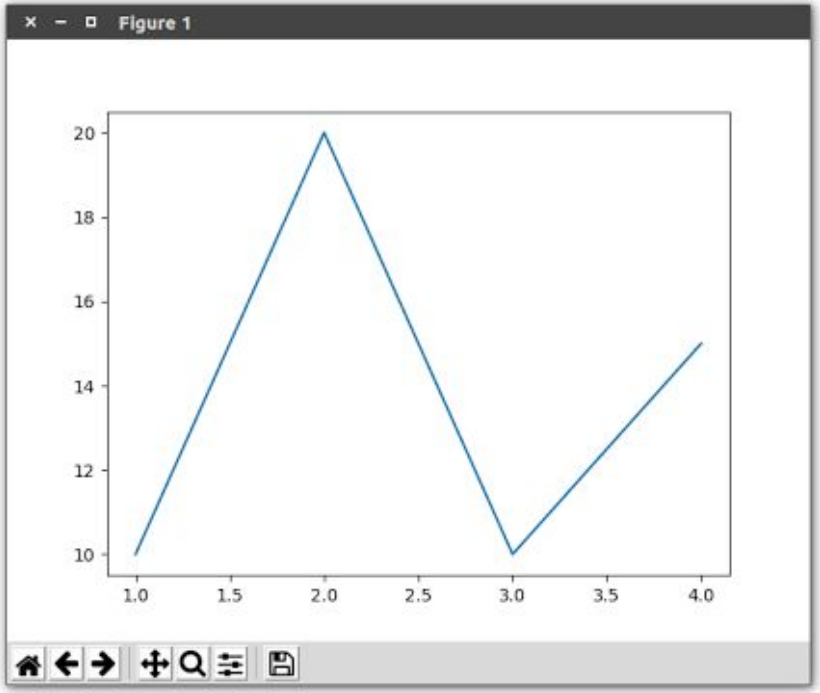
Staafdiagram

```
# Staafdiagram
barx = [1,2,3,4]
bary = [10,20,10,15]
plt.bar(barx,bary)
plt.show()
```



Lijndiagram

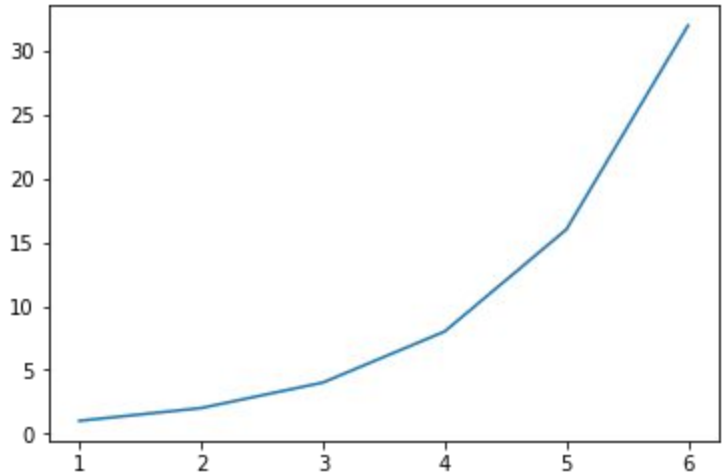
```
# Lijndiagram
linex = [1,2,3,4]
liney = [10,20,10,15]
plt.plot(linex,liney)
plt.show()
```



Smooth line? *lets* meer programmeerwerk

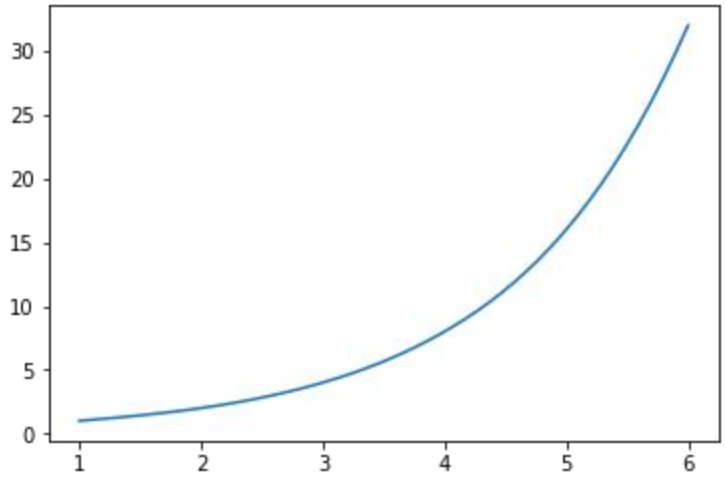
```
import matplotlib.pyplot as plt
from scipy.interpolate import spline
import numpy as np

x = np.array([1,2,3,4,5,6])
y = np.array([2**0,2**1,2**2,2**3,2**4,2**5]) #machten van 2, want redenen

# getal bepaald hoeveel datapunten tussen x.min en x.max te genereren
x_smooth = np.linspace(x.min(),x.max(),50)
#smoothen de curve van x en y mbv de nieuwe datapunten
y_smooth = spline(x,y,x_smooth)

plt.plot(x_smooth,y_smooth)
plt.show()
```

np.linspace(x.min(),x.max(),6)



Bij weinig genereren valt er weinig te smoothen

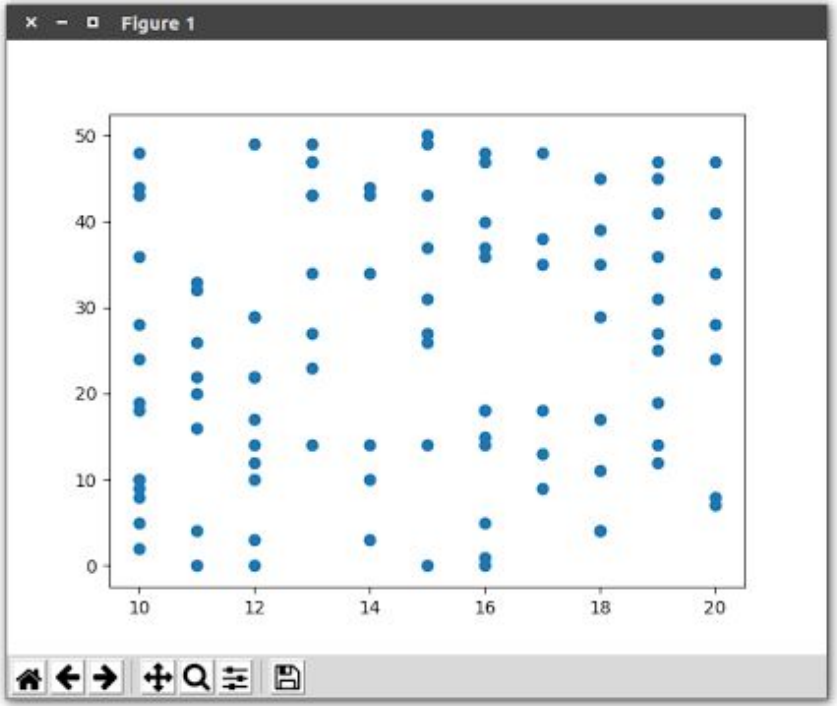
np.linspace(x.min(),x.max(),50)



Hoe meer datapunten op x, hoe meer je moet genereren

Scatterplot

```
# Scatterplot
scatx = []
scaty = []
for i in range(0,100):
    scatx.append(randint(10,20))
    scaty.append(randint(0,50))

plt.scatter(scatx,scaty)
plt.show()
```



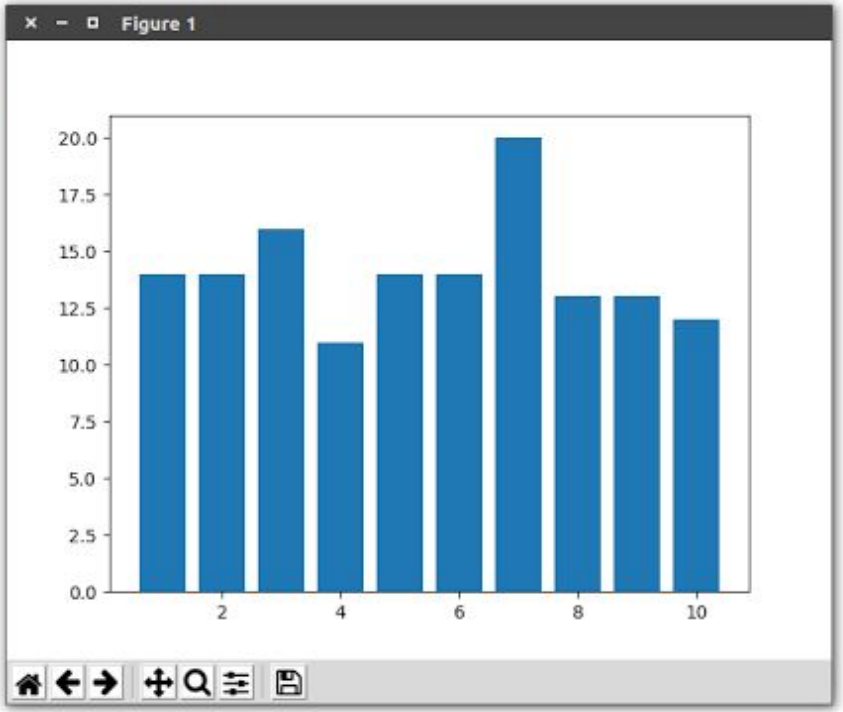
Opdracht 2 (5-10 min)

- Gebruik de volgende data:
  - x_data = [1,2,3,4,5,6,7,8,9,10]
  - y_data = [] # random getallen tussen 10 en 20

- **Maak een staafdiagram die gebruik maakt van deze data**

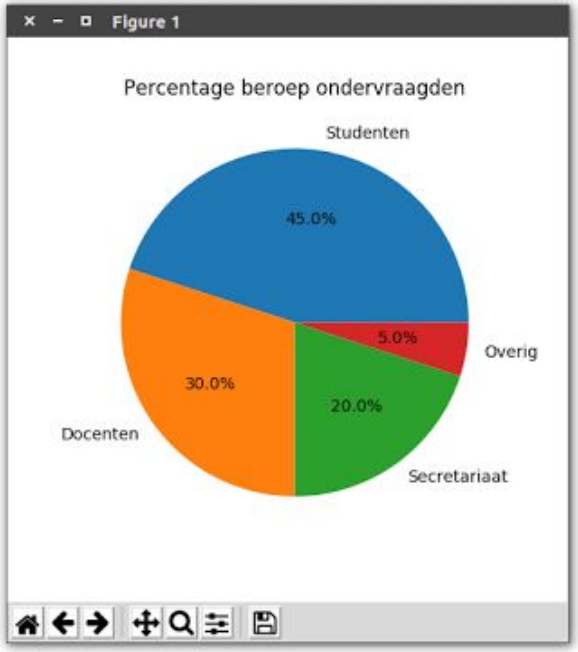
Mogelijke uitwerking

```
# Opdracht 2
x_data = [1,2,3,4,5,6,7,8,9,10]
# random getallen tussen 10 en 20
y_data = []
for i in range(len(x_data)):
    y_data.append(randint(10,20))
plt.bar(x_data,y_data)
plt.show()
```



Taartdiagram

```
# Taart diagram
labels = ['Studenten', 'Docenten', 'Secretariaat','Overig']
percentages = [45, 30, 20, 5]
plt.pie(percentages, labels=labels, autopct='%1.1f%%')
plt.title("Percentage beroep ondervraagden")
plt.show()
```



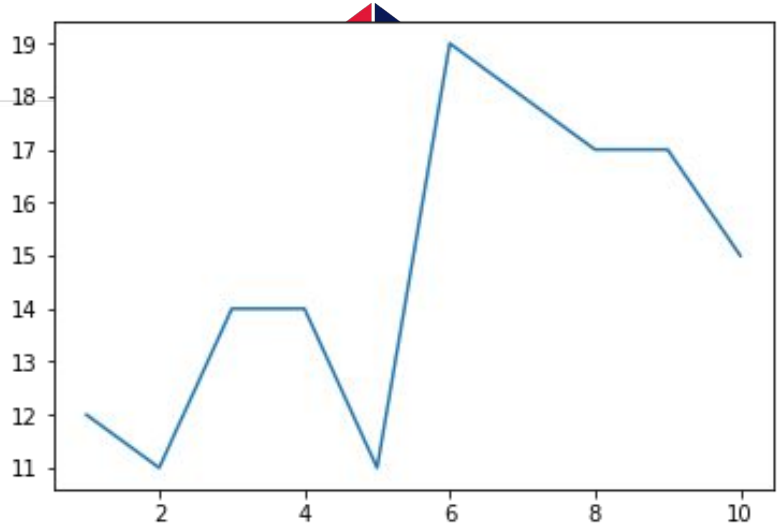
Input

- Niet alleen lists!
  - np.array()
    - Array like object
    - Lijkt veel op Python lists
    - Grote hoeveelheid berekeningen die je kan loslaten via numpy
  - Plaatjes
    - Vragen iets meer bewerking

Opdracht 3 (10 min)

- Maak een lijndiagram met de volgende data:
  - x_data = [1,2,3,4,5,6,7,8,9,10]
  - y_data = [] # random getallen tussen 10 en 20
- Converteer deze data naar numpy arrays
  - x_np = np.array(x_data)
- Plot de data in een lijndiagram
- Observeer dat er geen verschil is tussen np.array en list

Mogelijke uitwerking

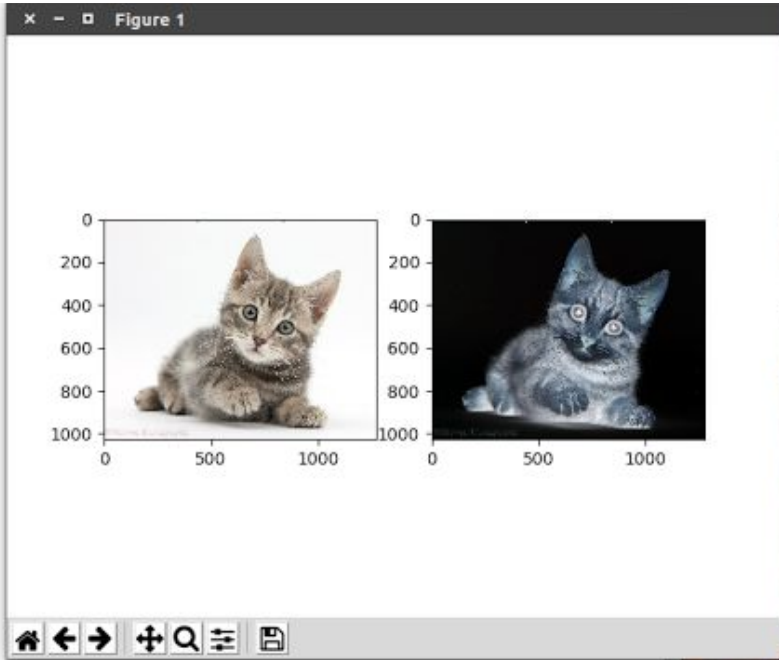



```
import matplotlib.pyplot as plt
import numpy as np
from random import randint

x_data = [1,2,3,4,5,6,7,8,9,10]
y_data = [randint(10,20) for i in range(0,10)]

x_np = np.array(x_data)
y_np = np.array(y_data)

plt.plot(x_np,y_np)
plt.show()
```

HAN

Plaatjes vergelijken

```
# Plaatjes
from scipy import ndimage
from scipy import misc
kitten = ndimage.imread("kitten.jpg")
kitten_new = ndimage.imread("kitten_new.jpg")
fig, axs = plt.subplots(nrows=1, ncols=2)
axs[0].imshow(kitten)
axs[1].imshow(kitten_new)
plt.show()
```



Inhoud

- Introductie Matplotlib
- Installatie Matplotlib
- Verschillende grafiek types
  - Scatter
  - Lijn
  - Bar
  - Taart
- Subplots en grafiek eigenschappen

Subplots

- Er zijn meerdere manieren waarop je een plaatje met meerdere datasets kunt maken
  - Meerdere lijnen in een grafiek
  - Meerdere grafieken in een plaatje
  - Misschien nog meer?

Subplots: meerdere grafieken

- Weinig extra code voor nodig

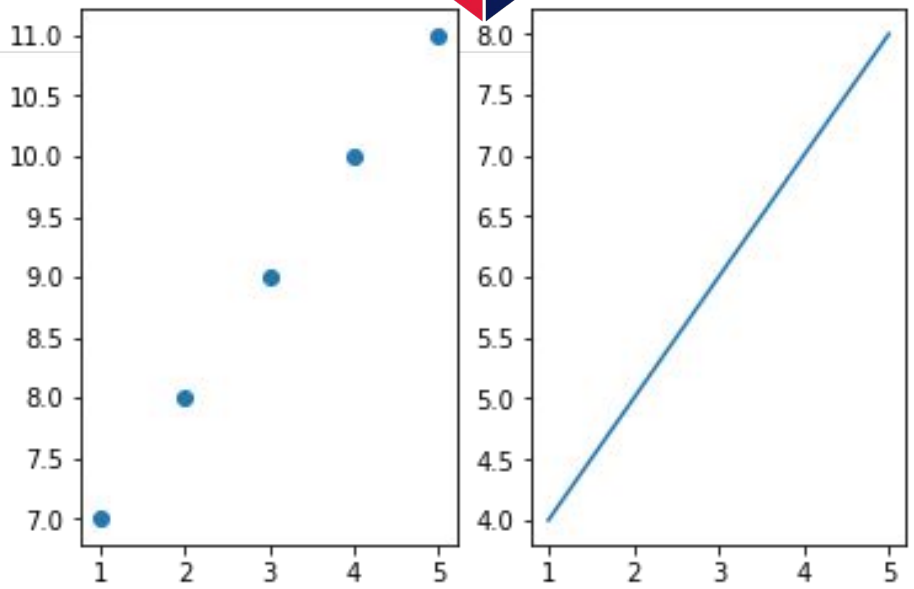
```
fig, axs = plt.subplots(nrows=2, ncols=1)

axs[0]= plt.plot(x_test,y_test)
axs[1]= plt.plot(x_placebo,y_placebo)

fig.show() # kan ook met plt.show()
```

Begrippen

nrows	Bepaald aantal rijen in fig
ncols	Bepaald aantal kolommen in fig
fig	Soort raamwerk waarin zich alle figuren gaan bevinden
axs	Individuele raampjes in fig, waarin je dus een plot kwijt kan

Voorbeeld

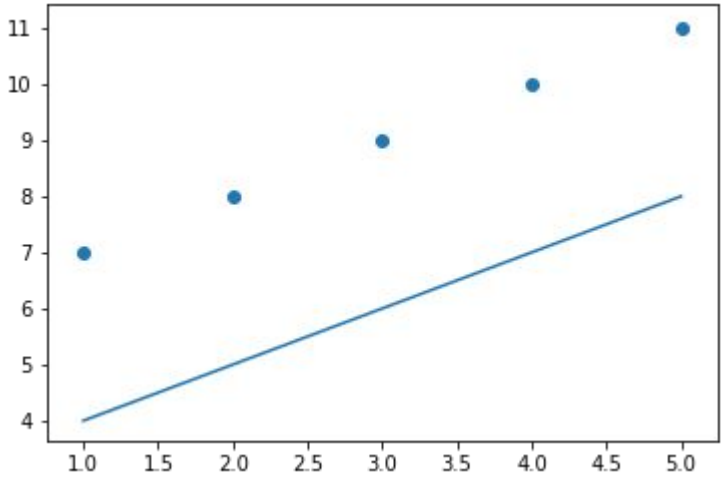
```
import matplotlib.pyplot as plt

x_data = [1,2,3,4,5]
y_data = [4,5,6,7,8]
x_test = [1,2,3,4,5]
y_test = [7,8,9,10,11]

fig, axs = plt.subplots(ncols=2,nrows=1)

axs[1].plot(x_data,y_data)
axs[0].scatter(x_test,y_test)

plt.show()
```



Subplots: meerdere lijnen

- Ook hier is weer weinig extra code nodig

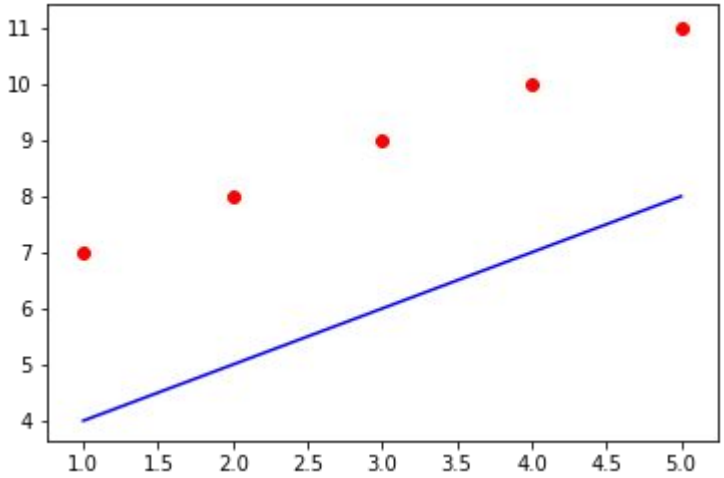
```
import matplotlib.pyplot as plt

x_data = [1,2,3,4,5]
y_data = [4,5,6,7,8]
x_test = [1,2,3,4,5]
y_test = [7,8,9,10,11]

plt.plot(x_data,y_data)
plt.scatter(x_test,y_test)

plt.show()
```



Subplots: meerdere lijnen

- Dan zou het ook wel leuk zijn als het een ander kleurtje was

```
import matplotlib.pyplot as plt


x_data = [1,2,3,4,5]
y_data = [4,5,6,7,8]
x_test = [1,2,3,4,5]
y_test = [7,8,9,10,11]


plt.plot(x_data,y_data, color='blue')
plt.scatter(x_test,y_test, color='red')


plt.show()
```



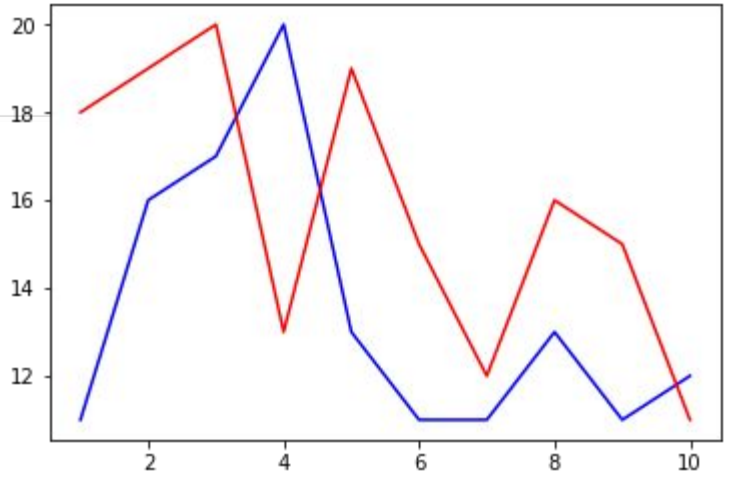
Opdracht 4 (10-15 min)

- Neem de data die je gebruikt hebt voor het maken van de lijndiagram
  - x_data = [1,2,3,4,5,6,7,8,9,10]
  - y_data = [] # random getallen tussen 10 en 20

- Maak twee versies van deze data en verander ze een klein beetje ten opzichte van elkaar

- Gebruik deze data om twee lijnen in een grafiek te maken, geef ze allebei een andere kleur

Mogelijke uitwerking



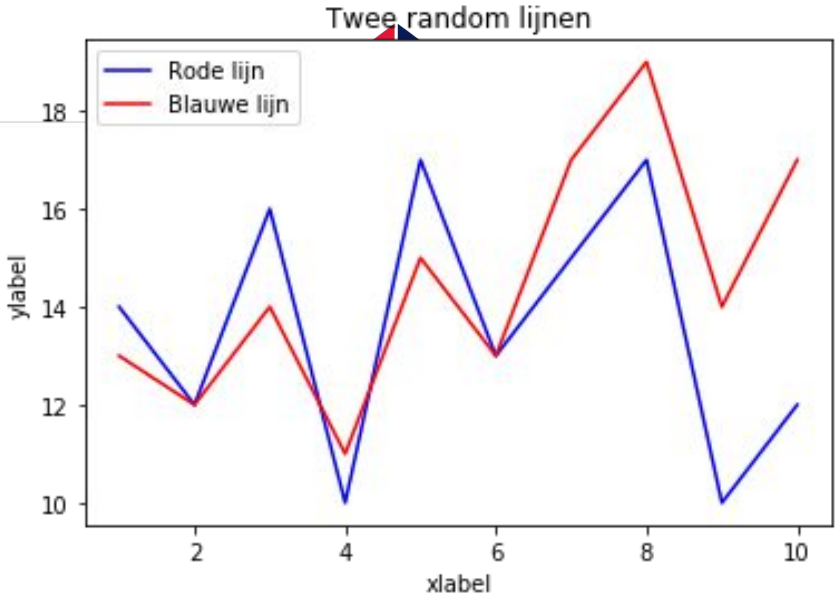

```
import matplotlib.pyplot as plt
from random import randint

x_data = [1,2,3,4,5,6,7,8,9,10]
y1_data = [randint(10,20) for i in range(0,10)]
y2_data = [randint(10,20) for i in range(0,10)]

plt.plot(x_data,y1_data, color='blue')
plt.plot(x_data,y2_data, color='red')

plt.show()
```

HAN

Grafiek eigenschappen

Titel	`plt.title()`
Legenda	`plt.legend()`
X-as omschrijving	`plt.xlabel()`
Y-as omschrijving	`plt.ylabel()`

Voorbeeld



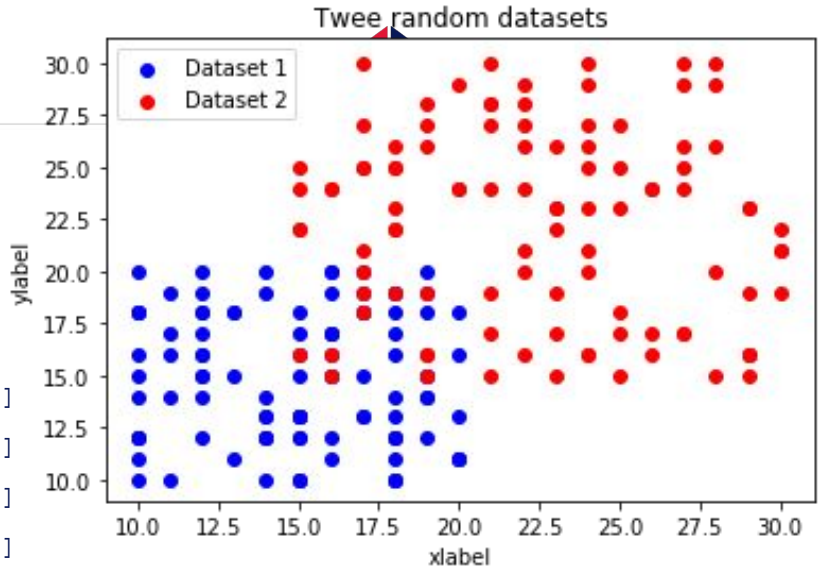

```
import matplotlib.pyplot as plt
from random import randint

x_data = [1,2,3,4,5,6,7,8,9,10]
y1_data = [randint(10,20) for i in range(0,10)]
y2_data = [randint(10,20) for i in range(0,10)]

plt.plot(x_data,y1_data, color='blue', label="Rode lijn")
plt.plot(x_data,y2_data, color='red', label="Blauwe lijn")

plt.title("Twee random lijnen")
plt.legend()
plt.xlabel("xlabel")
plt.ylabel("ylabel")
plt.show()
```

HAN

Opdracht 5 (10-15 min)

- Maak een scatterplot met twee datasets (rood en blauw)
- Gebruik hiervoor de data die je al eerder hebt gebruikt
- Maak het plaatje compleet door het volgende toe te voegen:
  - Een titel
  - Omschrijvingen van de x en y as
  - Legenda

Mogelijke uitwerking




```
import matplotlib.pyplot as plt
from random import randint

x1_data = [randint(10,20) for i in range(0,100)]
y1_data = [randint(10,20) for i in range(0,100)]
x2_data = [randint(15,30) for i in range(0,100)]
y2_data = [randint(15,30) for i in range(0,100)]

plt.scatter(x1_data,y1_data, color='blue', label="Dataset 1")
plt.scatter(x2_data,y2_data, color='red', label="Dataset 2")

plt.title("Twee random datasets")
plt.legend()
plt.xlabel("xlabel")
plt.ylabel("ylabel")
plt.show()
```

HAN

Samenvatting

- Matplotlib is een heel erg handig tool om grafieken te maken
- Het heeft een leercurve, maar je kan er snel behoorlijk veel mee
- Met relatief weinig commando's kan je een grafiek maken waarmee je snel inzicht krijgt in je data

Opdracht

Afvinkopdracht 2

Verantwoording

- In deze uitgave is géén auteursrechtelijk beschermd werk opgenomen
- Alle teksten © Martijn van der Bruggen/Esther Kok/HAN tenzij expliciet externe bronnen zijn aangegeven
- Screenshots op basis van eigen werk auteur en/of vernoemde sites en/of fair use
- Eventuele images zijn opgenomen met vermelding van bron