

Matplotlib



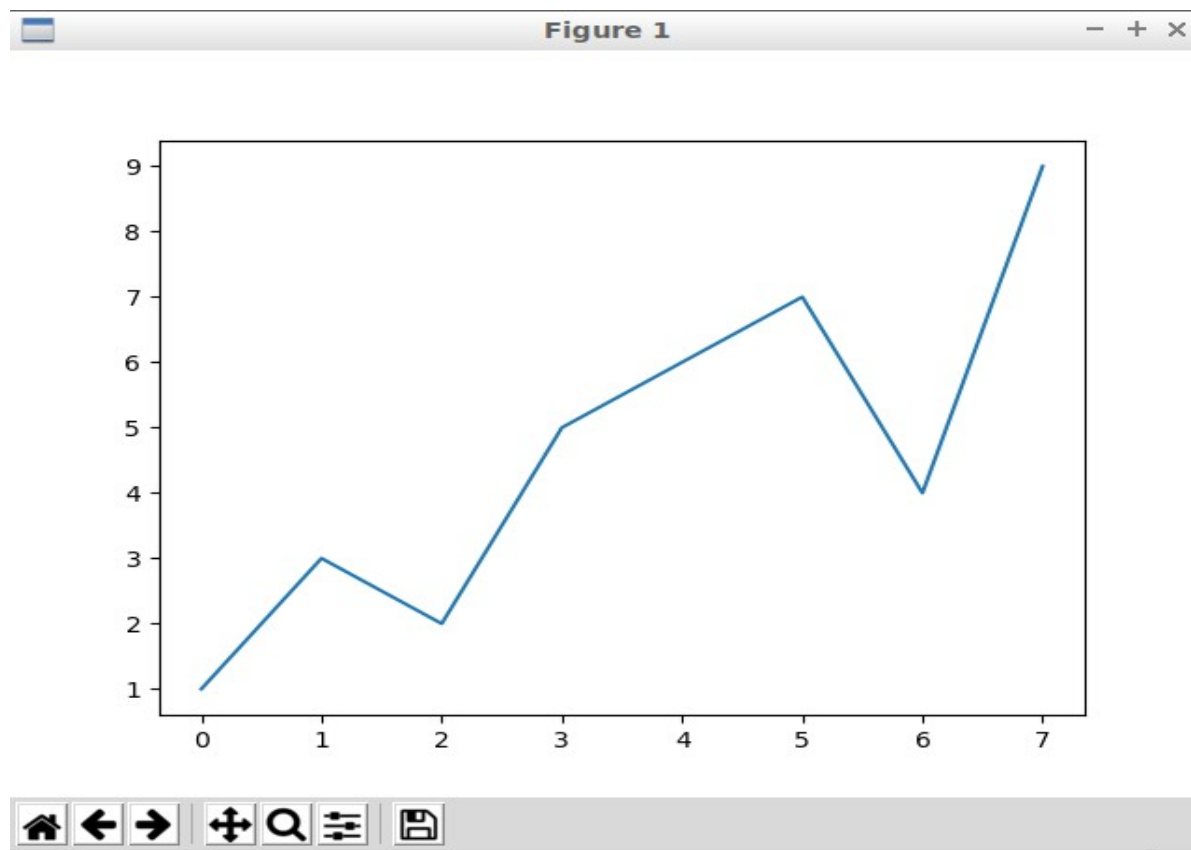
Matplotlib is een module voor Python waarmee verschillende typen grafieken gemaakt kunnen worden.

In de volgende dia's gaan we stap voor stap de mogelijkheden van matplotlib verkennen.

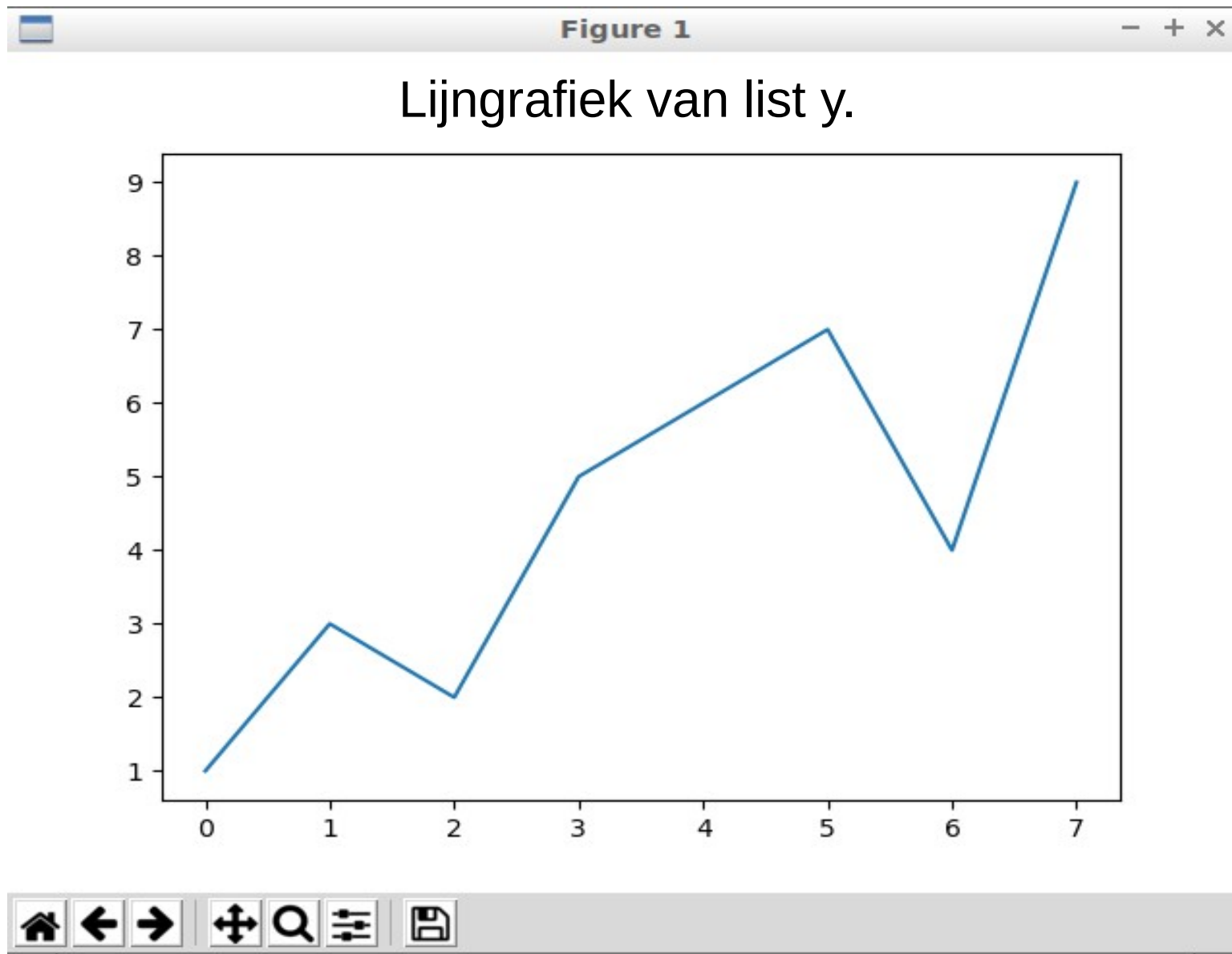
Dat doen we aan de hand van korte scripts,

Matplotlib

```
import time
import matplotlib.pyplot as plt # importeer matplotlib met als alias plt.
y = [1,3,2,5,6,7,4,9] # een list met willekeurige getallen
plt.plot (y) # De waarden in list y worden geplot waarbij
              # de x-as wordt bepaald door het aantal elementen in y.
plt.show()   # De functie show zorgt ervoor dat de grafiek getoond wordt.
```



Matplotlib



X schaal wordt bepaald door het aantal elementen in y.

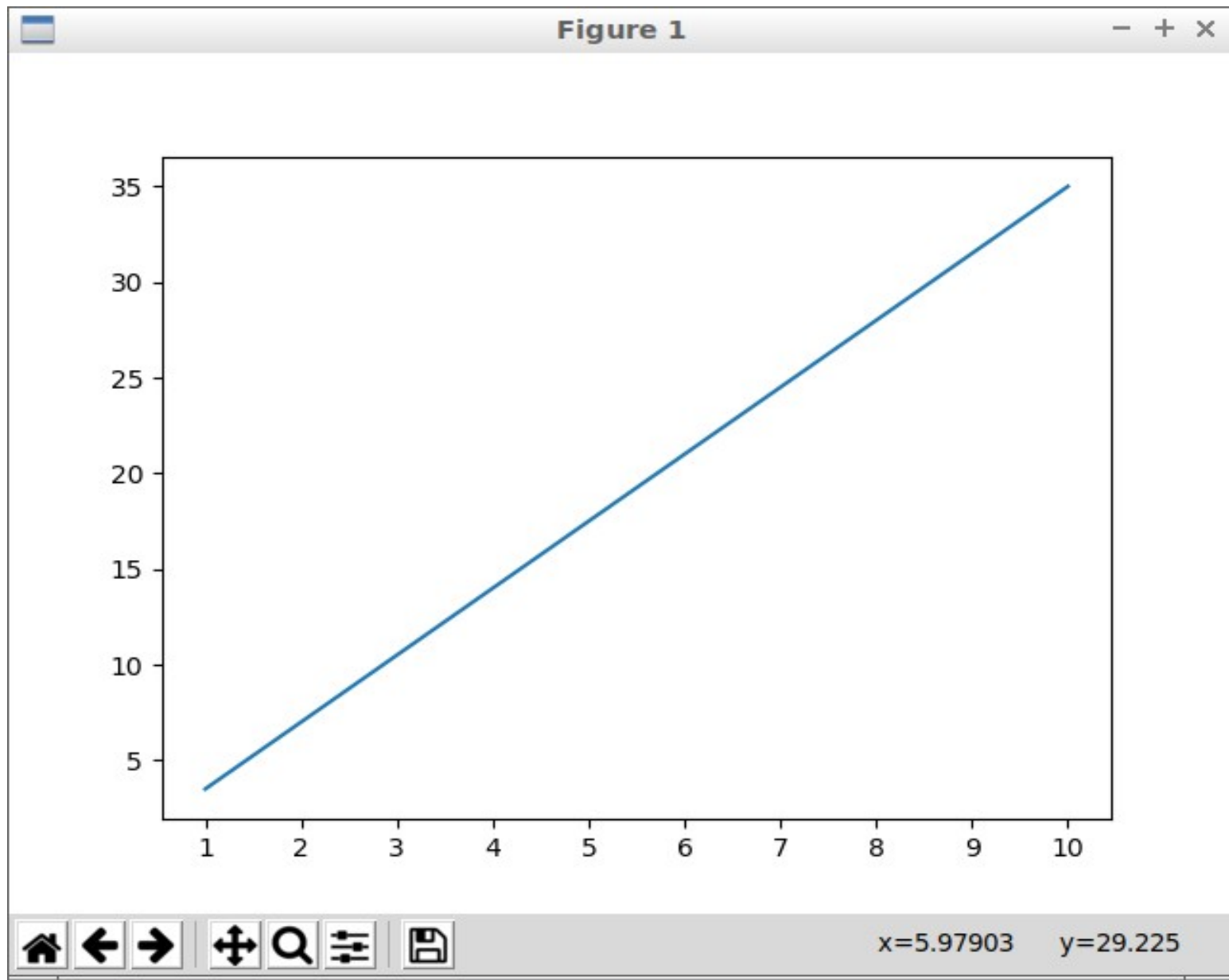
Matplotlib

```
# matplotlib_02.py vermenigvuldiging van een array
import matplotlib.pyplot as plt
import numpy as np
x = np.array(range(1,11)) # genereer een np array met getallen 1 tot 10
y = x*3.5                 # vermenigvuldig het array met 3.5
plt.plot(x,y)            # plot arrays x en y
plt.xticks(x)            # x-as gelijk aan anta elementen in array x
plt.show()
```

Met `np.array` genereren we een np list van 1 tot en met 10
np list y is het product van np list x en 3.5

De plot levert een grafiek op met een linear verband tussen x en y.
Zie de volgende dia.

Matplotlib

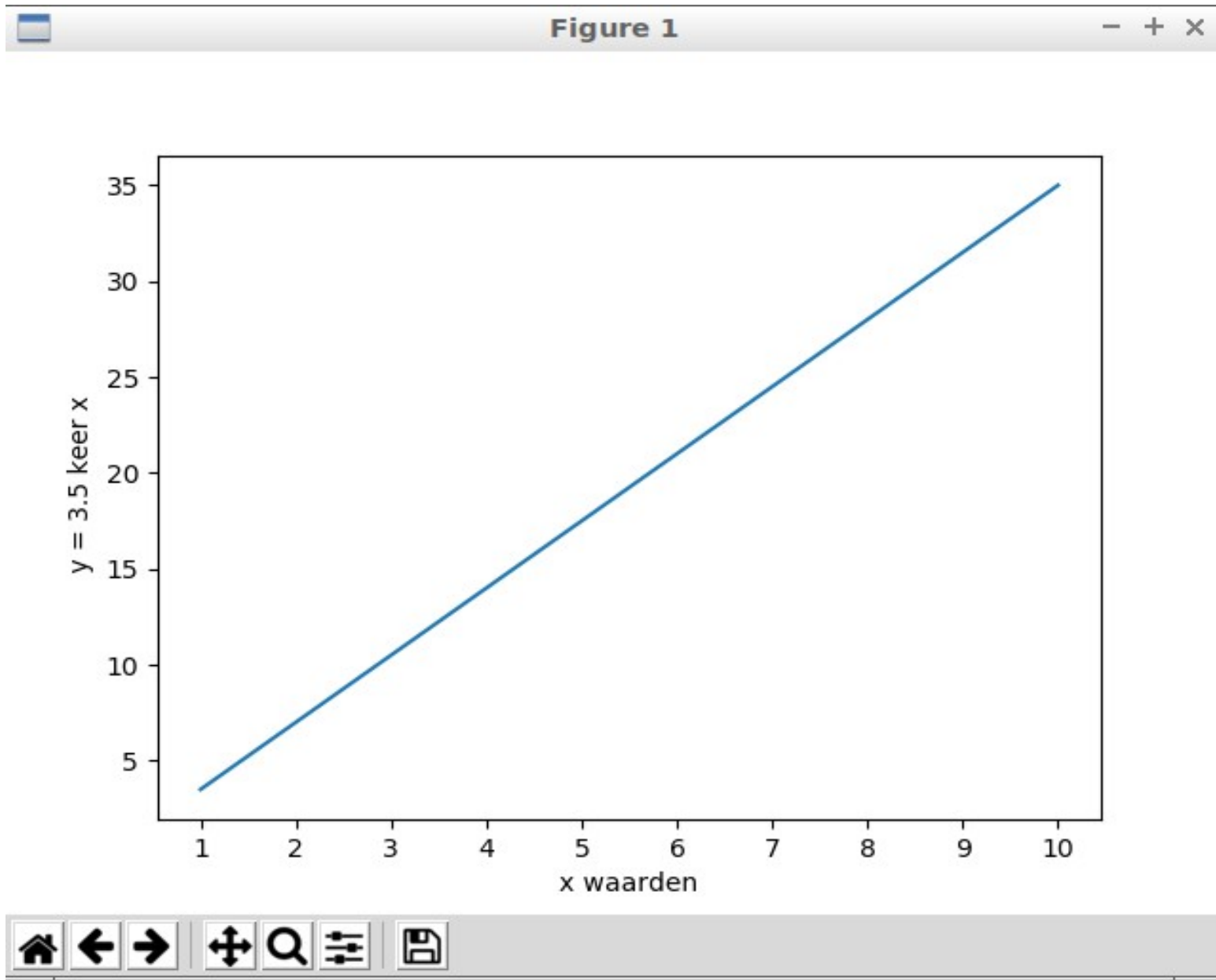


Matplotlib

```
#matplotlib_03.py labels plaatsen bij x en y as
import matplotlib.pyplot as plt
import numpy as np
x = np.array(range(1,11))
y = x*3.5
plt.plot(x,y)
plt.xticks(x) # geef met xticks aan hoed schaal er uit moet zien
plt.xlabel("x waarden")
plt.ylabel(" y = 3.5 keer x")
plt.show() # functie zorgt er voor dat de grafiek getoond wordt
```

Het script is bijna gelijk aan het vorige script.
In dit script zijn labels toegevoegd aan de x en y schaal
zodat je weet wat de parameters van het grafiek zijn.

Matplotlib

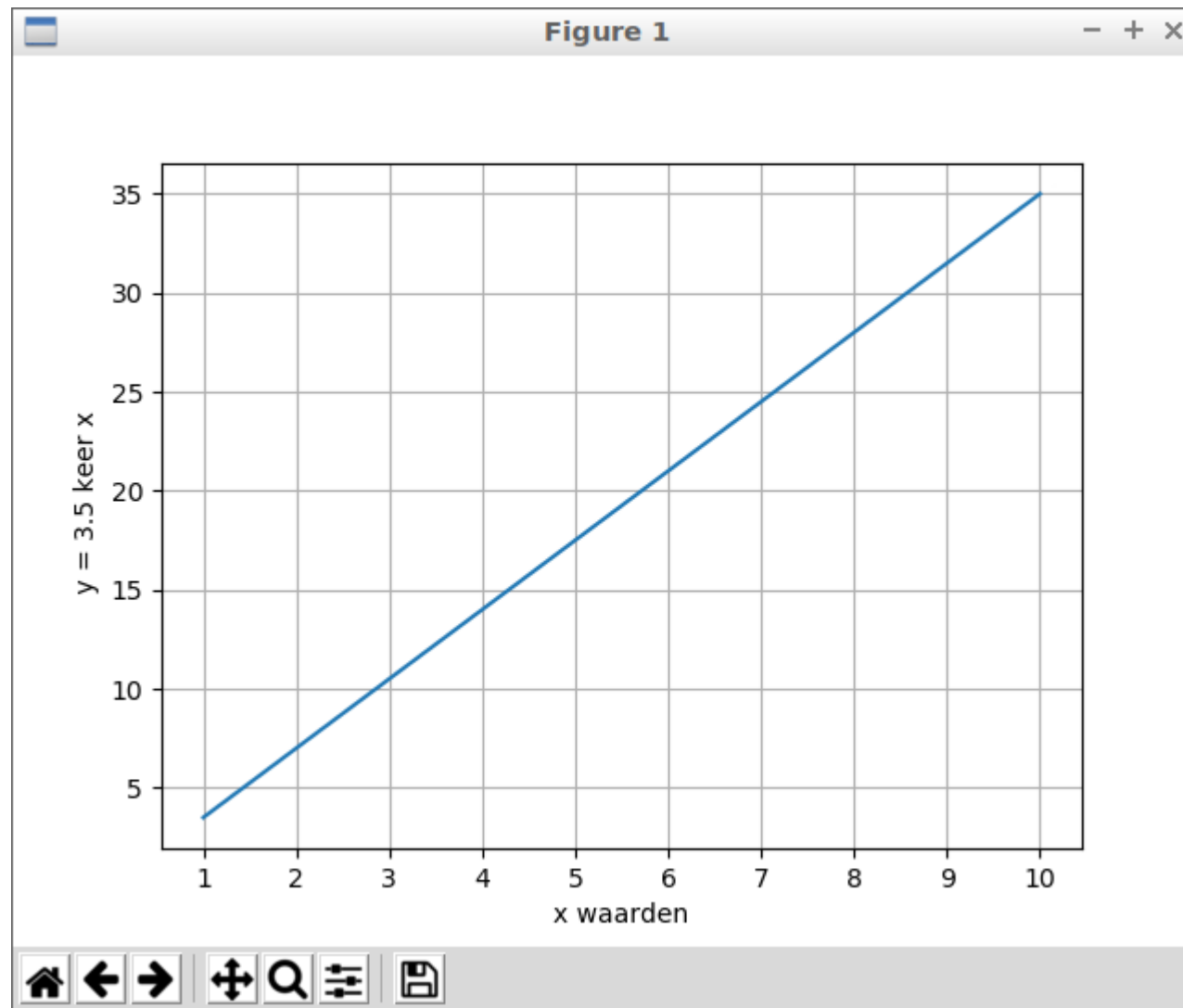


Matplotlib

```
#matplotlib_04.py een raster aan de grafiek toevoegen
import matplotlib.pyplot as plt
import numpy as np
x = np.array(range(1,11))
y = x*3.5
plt.plot(x,y)
plt.xticks(x)
plt.xlabel("x waarden")
plt.ylabel("y = 3.5 keer x")
plt.grid("True")
plt.show()
```

Met de `plt.grid` functie voegen we een raster toe aan de grafiek.

Matplotlib



Met de `plt.grid` functie voegen we een raster toe aan de grafiek.

Matplotlib

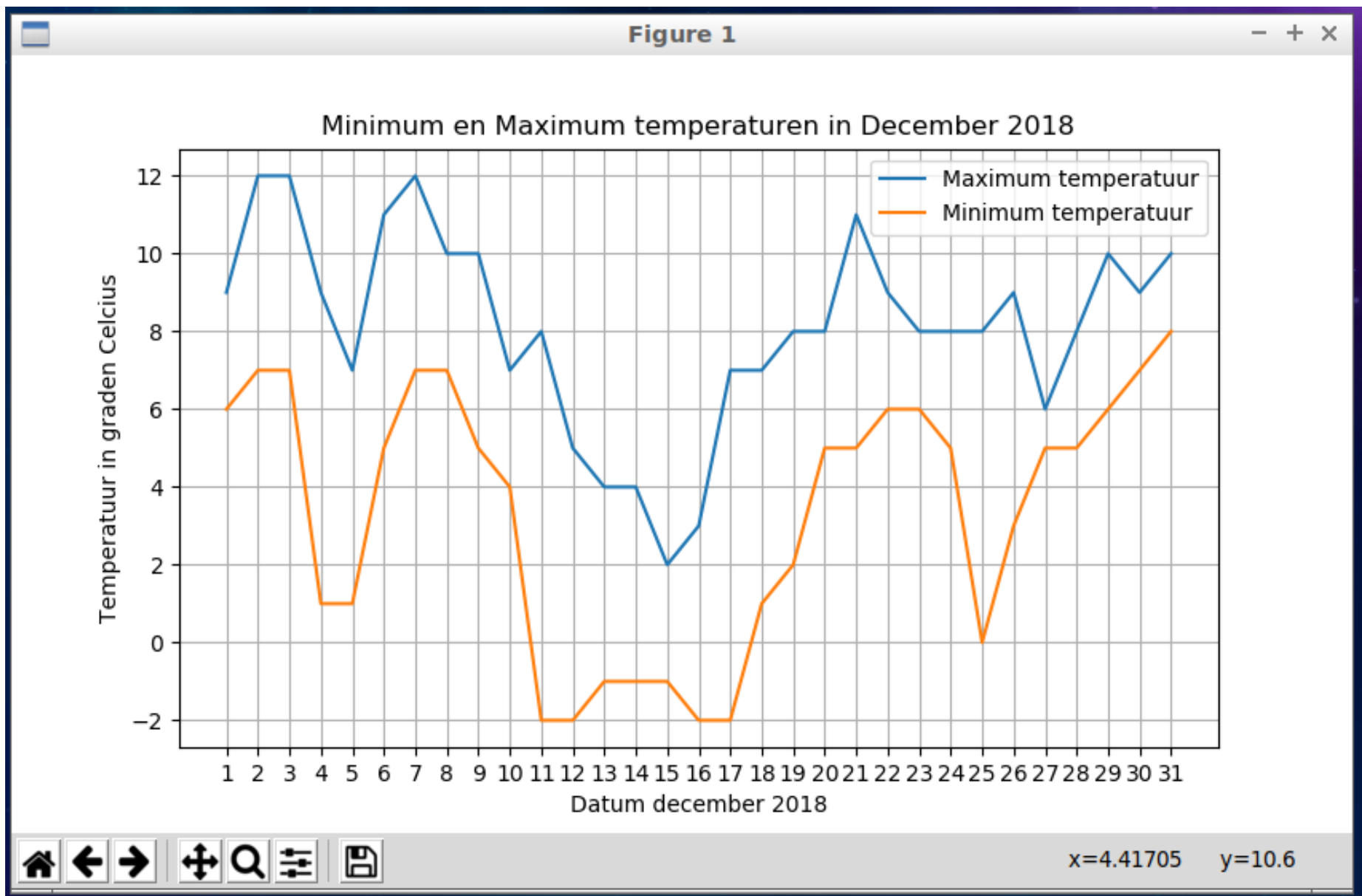
Temperatuur grafiek Leeuwarden december 2018, bron KNMI.

```
#temp_plot-dec_2018.py
# Maximum en minimum etmaaltemperaturen in december 2018 in Leeuwarden bron KNMI

import matplotlib.pyplot as plt
datum = list(range(1,32))
temp_max = [9,12,12,9,7,11,12,10,10,7,8,5,4,4,2,3,7,7,8,8,11,9,8,8,8,9,6,8,10,9,
temp_min = [6,7,7,1,1,5,7,7,5,4,-2,-2,-1,-1,-1,-2,-2,1,2,5,5,6,6,5,0,3,5,5,6,7,8
plt.plot(datum, temp_max, label = "Maximum temperatuur")
plt.plot(datum, temp_min, label = "Minimum temperatuur")
plt.xlabel("Datum december 2018")
plt.xticks(datum)
plt.grid(True)
plt.ylabel("Temperatuur in graden Celcius")
plt.title("Minimum en Maximum temperaturen in December 2018")
plt.legend()
plt.show()
```

De data van de dagen in december (datum) genereren we met de list(range) functie.

Matplotlib



Matplotlib

In bovenstaande grafiek komen we verschillende attributen tegen die de betekenis van de curves moeten verduidelijken.

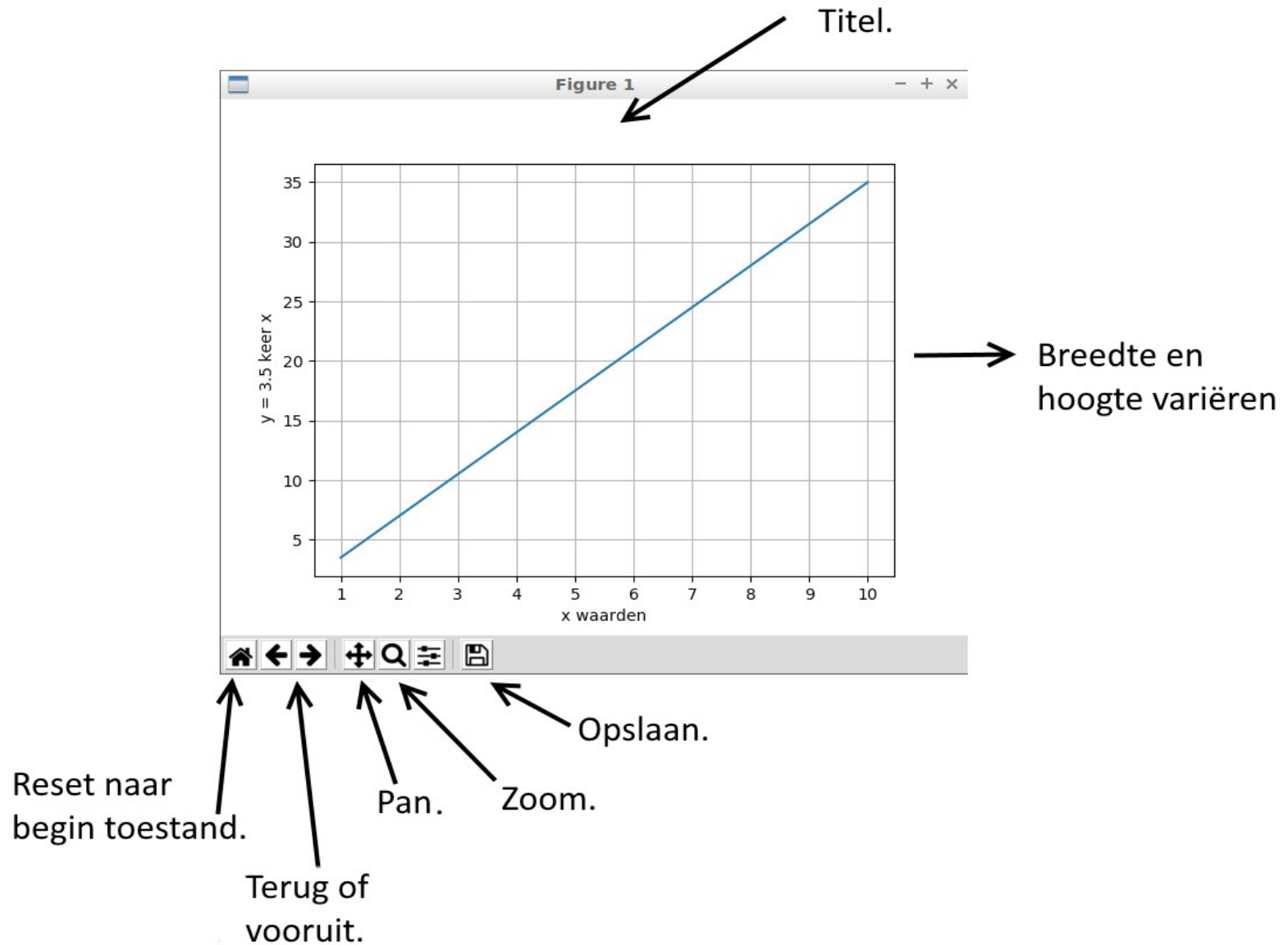
Bij de plot functie zien we de toevoeging *label* waarmee duiden we aan welke data de curve vertegenwoordigd.

Bij de x en y schaal geven we aan om welke grootheid het gaat.

Verder kunnen we de plot nog een titel meegeven.

De x en y label tekst zien we terug in de legenda.

Matplotlib



Matplotlib

Grafiek van de leeftijd opbouw van een voetbalteam.

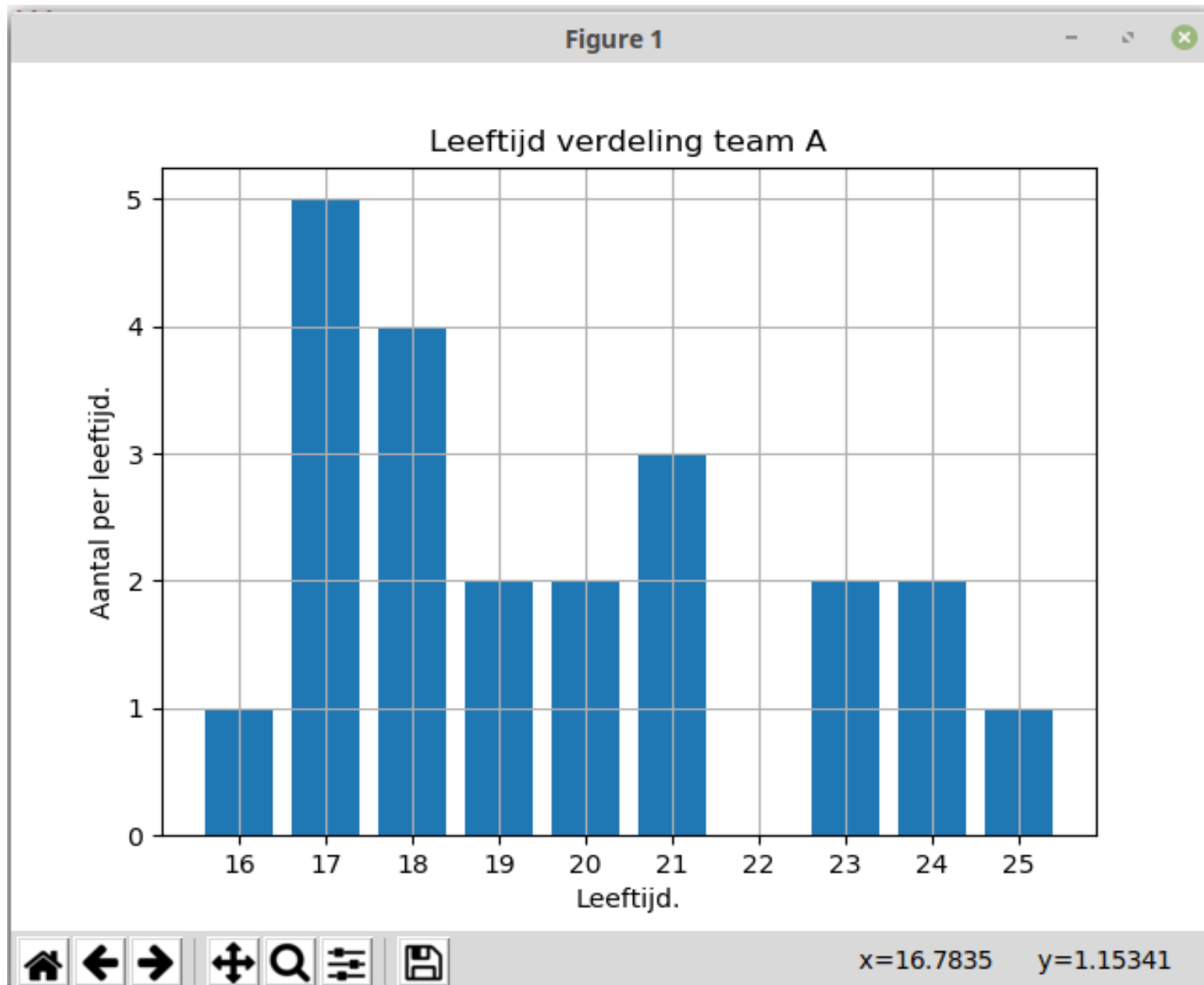
```
#data_viz_lft.py
'''
Veronderstel je hebt een spreadsheet met namen en leeftijden
van bijvoorbeeld een voetbalteam. Leeftijden lopen van 16 tot 25 jaar.
Je kunt de kolom leeftijden exporteren naar een python lijst.
Veronderstel verder dat je een staafdiagram van de leeftijdsopbouw wilt maken.
Dan dien je te weten hoe vaak een bepaalde leeftijd voorkomt in het bestand.

'''
a = 16 # start leeftijd van de teller voor de while loop
lpg = [] # het aantal deelnemers per leeftijds groep
leeftijden = [16, 17, 18, 19, 20, 21, 22, 23, 24, 25]
deelnemer_lft = [ 17, 17, 21, 19, 21, 17, 23, 17, 25, 24, 24, 19, 16, 18, 18, 17, 20, 23, 21, 18, 18, 20]
while a <= 25: # while loop doorloopt reeks van 16 tot en met 25
    aantal = deelnemer_lft.count(a) # telt aantal deelnemers per leeftijd
    lpg.append(aantal) # aantal toegevoegd aan list lpg
    a = a + 1 # teller met 1 verhoogd
print(lpg)

leeftijd =[ 16, 17, 18, 19, 20, 21, 22, 23, 24, 25] # x-as van de grafiek
import matplotlib.pyplot as plt
plt.ylabel("Aantal per leeftijd.")
plt.xlabel("Leeftijd." )
plt.grid (True)

plt.title("Leeftijd verdeling team A")
plt.xticks (leeftijd)
plt.bar( leeftijd, lpg)
plt.show()
```

Matplotlib



Matplotlib

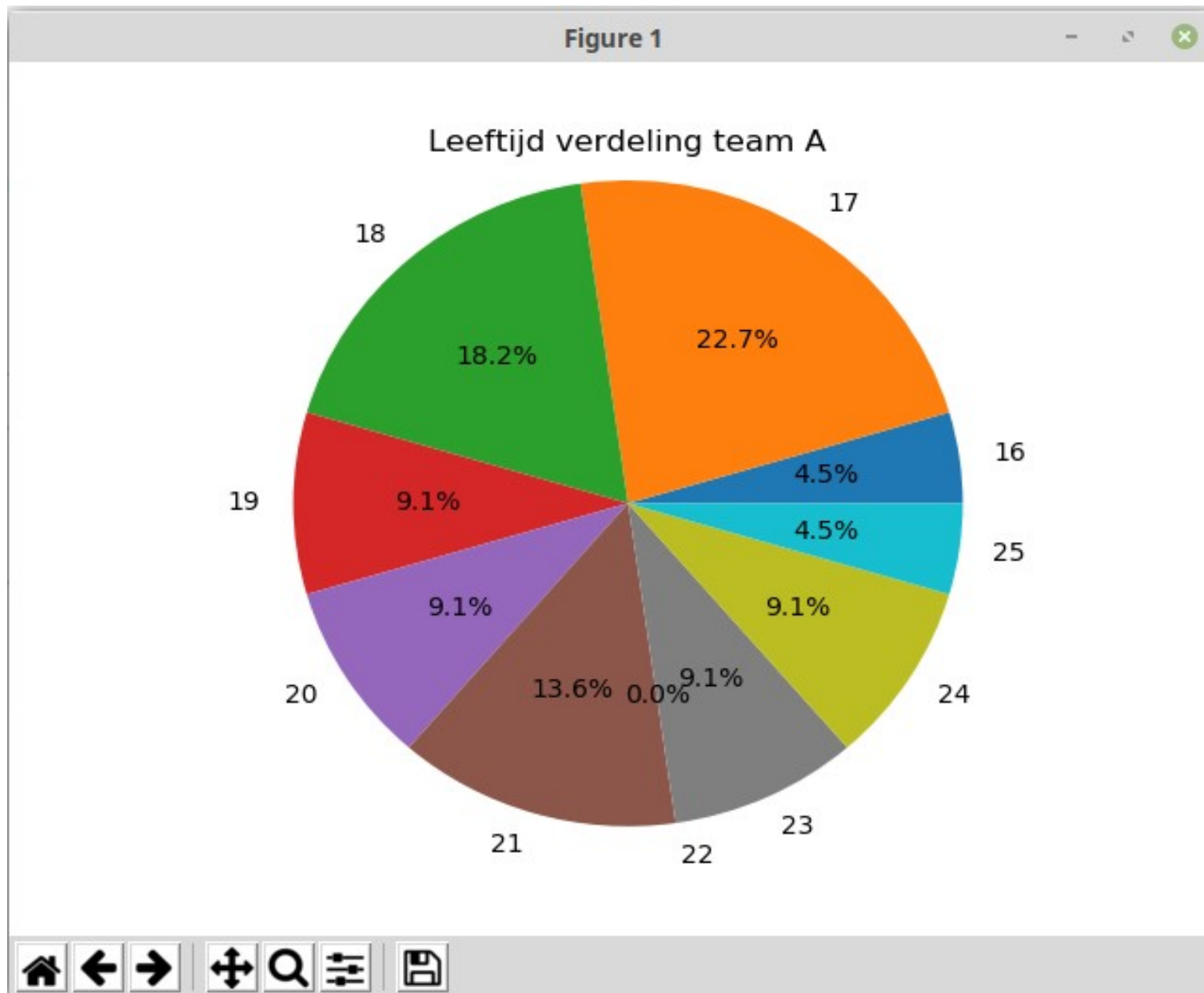
Plot van de leeftijd opbouw van het team in piechart vorm.

```
#data_viz_lft_pie.py
...
Veronderstel je hebt een spreadsheet met namen en leeftijden
van bijvoorbeeld een voetbalteam. Leeftijden lopen van 16 tot 25 jaar.
Je kunt de kolom leeftijden exporteren naar een python lijst.
Veronderstel verder dat je een piechart van de leeftijdsopbouw wilt maken.
Dan dien je te weten hoe vaak een bepaalde leeftijd voorkomt in het bestand.
...
a = 16 # start leeftijd van de teller voor de while loop
lpg = [] # het aantal deelnemers per leeftijds groep
leeftijden = [16, 17, 18, 19, 20, 21, 22, 23, 24, 25]
deelnemer_lft = [ 17, 17, 21, 19, 21, 17, 23, 17, 25, 24, 24, 19, 16, 18, 18, 17, 20, 23, 21, 18, 18, 20]
while a <= 25: # while loop doorloopt reeks van 16 tot en met 25
    aantal = deelnemer_lft.count(a) # telt aantal deelnemers per leeftijd
    lpg.append(aantal) # aantal toegevoegd aan list lpg
    a = a + 1 # teller met 1 verhoogd
print(lpg)

bins = [ 16, 17, 18, 19, 20, 21, 22, 23, 24, 25] # We geven hieraan hoe de plot er uit komt te zien

import matplotlib.pyplot as plt
plt.axis ("equal")
plt.pie(lpg, labels = bins, radius = 1.2, autopct = "%0.1f%%") # namen, grootte en decimalen van de percentages
plt.title("Leeftijd verdeling team A") # titel van de grafiek, n dit geval een taartdiagram
plt.show()
```

Matplotlib



Matplotlib

Regenval en maximum temperatuur in maart 2018 (1)

```
#Grafiek regenval en temperatuur in maart 2018
#voor dit programma moet het csv bestand regenval_maart2.csv
#in de zelfde map staan als het script regenval_maart_2018.py
import matplotlib.pyplot as plt
import numpy as np
datum = []
geregend = []
temp = []
import csv
with open("regenval_maart2.csv") as csvfile:
    regen = csv.reader(csvfile, delimiter = ",")
    for column in regen:
        #datum.append(column [0])
        geregend.append(column [1])
        temp.append (column [2])

geregend = (geregend[1:31])
geregend = list (map(int,geregend))
print(geregend)
print()
```

Matplotlib

Regenval en maximum temperatuur in maart 2018 (2)

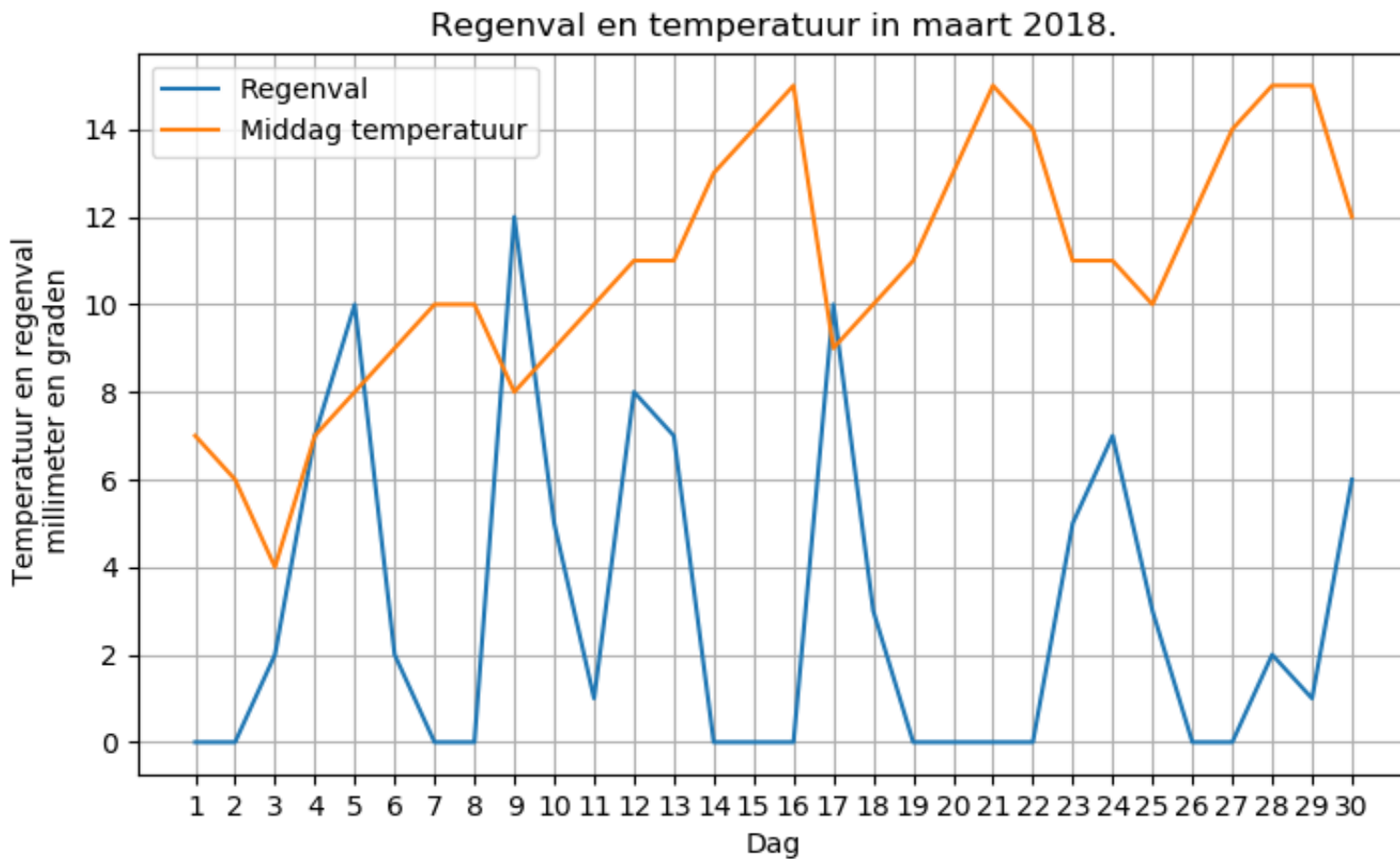
```
temp = (temp[1:31]) # de tekst Temperatuur wordt verwijderd
temp = list (map(int,temp)) # conversie van string naar integer data
print (temp)
datum = list (range(1,31)) # list datum wordt gegenereerd
print (datum)
```

```
plt.plot(datum, geregend, label = "Regenval")
plt.plot(datum, temp, label = "Middag temperatuur")
plt.title("Regenval en temperatuur in maart 2018.")
plt.xticks(datum)
plt.xlabel ("Dag")
plt.ylabel (" Temperatuur en regenval\n millimeter en graden")
plt.legend()
plt.grid(True)
plt.show()
```

Matplotlib

Regenval en maximum temperatuur in maart 2018

Figure 1



Matplotlib

VRAGEN?

Demo Temperatuur meting.

Matplotlib
