

# Matplotlib

Matplotlib er et data visualiserings værktøj for python. Det er et af de mest brugte værktøjer og er kompatible med de fleste andre data biblioteker så som polars og pandas. Dog er det ikke et standardbibliotek så man skal huske at installere det ved at køre

```
pip install matplotlib
```

Matplotlib har mange forskellige ting man kan bruge. Den understøtter også direkte de fleste typer af diagrammer og plots man vil lave. Man kan også tilpasse de forskellige dele af grafen til bedre at formulere sine data. De nysgerrige kan gå ind på deres link og læse mere om hvad den kan [her](#)

## Simpel plotting

Vil man bare plotte sine data med en linje mellem hvert punkt bruger man bare `pyplots plot` funktionen. Den tager som input x data og y data og sætter dem på et kanvas med linjer mellem hvert punkt

```
import matplotlib.pyplot as plt

data_1 : list[int] = [0,1,2,3,4]
data_2 : list[int] = [0,1,2,3,4]
plt.plot(data_1,data_2)
plt.show()
```

Man kan også navngive sine data ved at bruge labels f.eks. `plt.plot(data_1,data_2, label="Good data")` og derefter kalde `plt.legend()`

Man kan også give dem navne i den række følge de er blevet plottet ved at give en liste af labels med i funktionskaldet. Legend funktionen genererer og sætter rækkefølgen efter hvad der blev dannet først. Dog kan man ændre rækkefølgen selv. Et fuldt eksempel der gør brug af alle tre kunne se således ud

```
import matplotlib.pyplot as plt

x : list[int] = [0,1,2,3,4]
y_1 : list[int] = [0,1,2,3,4]
y_2 : list[int] = [1,2,3,4,5]
y_3 : list[int] = [2,3,4,5,6]
plot_1, = plt.plot(x,y_1)
plot_2, = plt.plot(x,y_2)
plot_3, = plt.plot(x,y_3)
labels = ["Normal", "Normal +2", "Normal +1"]
plt.legend(handles=[plot_1,plot_3,plot_2],labels=labels)
```

```
plt.show()
```

Læg mærke til bruget af komma efter plot\_x

Hvis man ønsker at navngive akser, kan man bruge pyplots x/ylabel som tager navnet i form af en string.

## Animationer

Matplotlib understøtter også at man løbende kan opdatere plottet, f.eks. i form af animationer. Til det kan man bruge **animation.FuncAnimation**. Denne funktion skal have tre inputs. En pyplot figure den kan tegne i, en funktion den skal kalde der fortæller hvordan den skal opdaterer plottet, og en samling af x værdier. Derudover kan man også definere en starts funktion, der kan opsætte figuren. En implementation der tegner en sinus kurve løbende kan implementeres således

```
import matplotlib.pyplot as plt
import numpy as np
from matplotlib.animation import FuncAnimation

fig, ax = plt.subplots()
x_data, y_data = [], []
ln, = ax.plot(x_data, y_data)

ax.set_xlim(0, np.pi)
ax.set_ylim(-1, 1)

def update(frame):
    x_data.append(frame)
    y_data.append(np.cos(frame*np.pi))
    ln.set_data(x_data, y_data)
    return ln,

animation = FuncAnimation(fig, update, frames=np.linspace(0, np.pi, 128))
plt.show()
```