

Einstieg in Matplotlib

Datenvisualisierung mit der Bibliothek Matplotlib

Was ist Matplotlib?

- erweiterbare Plotting-Bibliothek für Python

Matplotlib tries to make easy things easy and hard things possible

- angelehnt an die Programmiersprache MATLAB
- Ermöglicht es (interaktive) komplexe Visualisierungen zu erstellen

Beispiele

<https://matplotlib.org/gallery.html> (<https://matplotlib.org/gallery.html>).

Plot-Typen

- Matplotlib beherrscht alle (bekannten) Formen von Plots
- Wir beschränken uns heute auf einfache Plots:
 - Scatter-, Line-, Bar-, Pie-, Box-Plots

Daten anlegen/vorbereiten

- Damit wir etwas visualisieren können brauchen wir Daten
 - Für heute: Listen und .csv-Dateien

```
In [1]: y = [i for i in range(20,100,3)]  
        x = [i for i in range(len(y))]  
        print(x)
```

```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26]
```

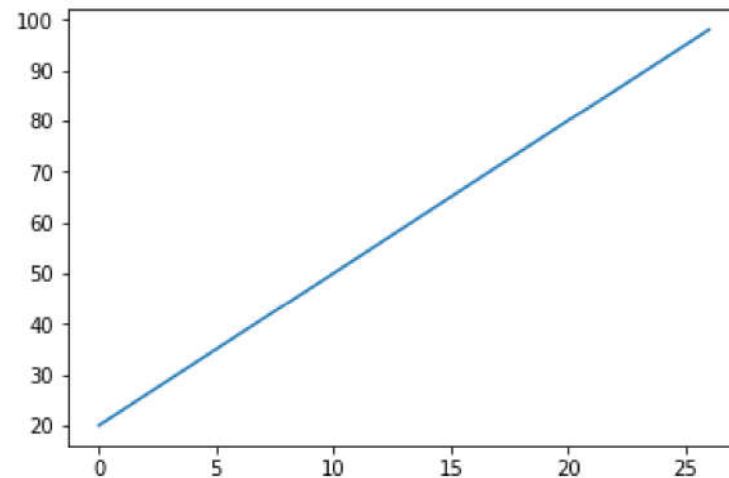
Scatter-Plot

```
In [2]: #%%matplotlib # gibt den Plot in einem neuen Fenster aus  
#y = [i for i in range(20,100,3)]  
#x = [i for i in range(len(y))]  
import matplotlib.pyplot as plot  
plot.scatter(x,y)  
plot.show()
```

<matplotlib.figure.Figure at 0x82e7110>

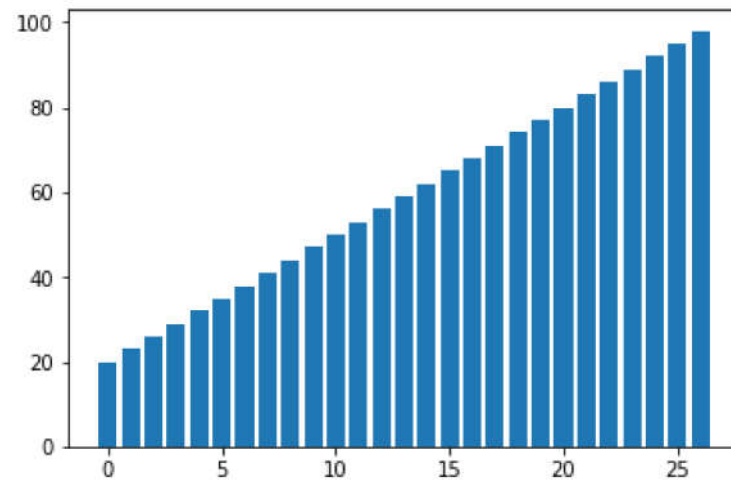
Klassischer Linien-Plot

```
In [3]: ##matplotlib # gibt den Plot in einem neuen Fenster aus  
        #y = [i for i in range(20,100,3)]  
        #x = [i for i in range(len(y))]  
        import matplotlib.pyplot as plot  
        plot.plot(x,y)  
        plot.show()
```



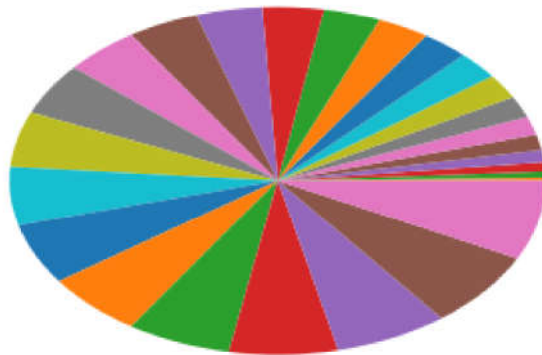
Balken-Plot

```
In [4]: ##matplotlib # gibt den Plot in einem neuen Fenster aus  
#y = [i for i in range(20,100,3)]  
#x = [i for i in range(len(y))]  
plot.bar(x,y)  
plot.show()
```



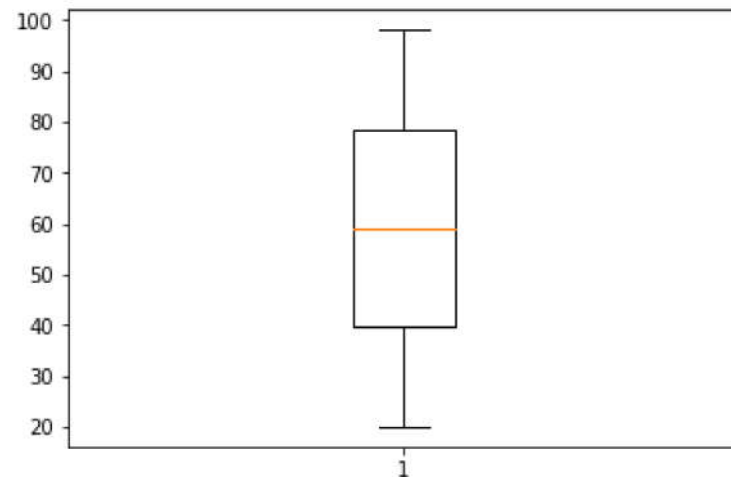
Tortendiagramm

```
In [5]: ##matplotlib # gibt den Plot in einem neuen Fenster aus  
        #y = [i for i in range(20,100,3)]  
        #x = [i for i in range(len(y))]  
        plot.pie(x)  
        plot.show()
```



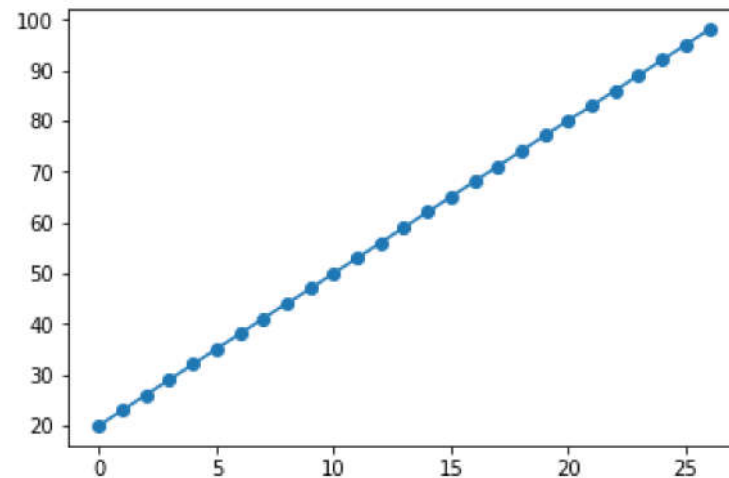
Box-Plot

```
In [6]: ##matplotlib # gibt den Plot in einem neuen Fenster aus  
#y = [i for i in range(20,100,3)]  
#x = [i for i in range(len(y))]  
plot.boxplot(y)  
plot.show()
```



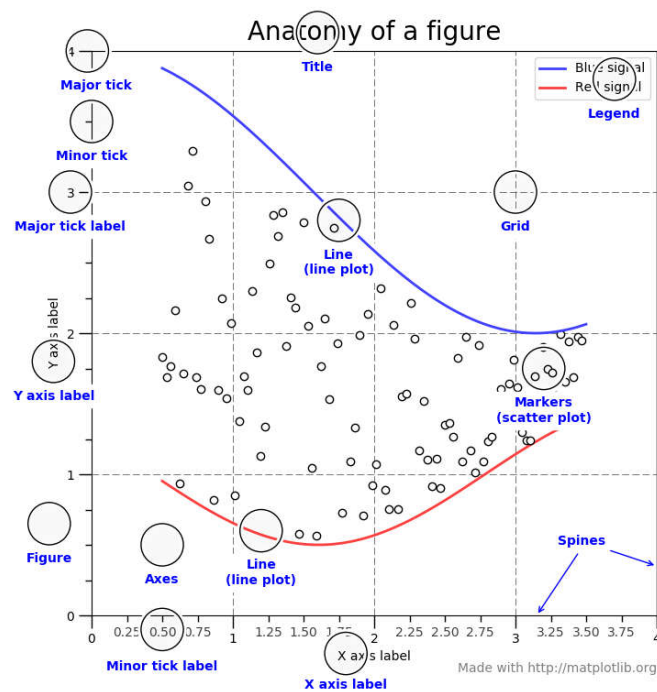
Mehrere Plots in einem Schaubild

```
In [7]: ##matplotlib # gibt den Plot in einem neuen Fenster aus  
#y = [i for i in range(20,100,3)]  
#x = [i for i in range(len(y))]  
plot.scatter(x,y)  
plot.plot(x,y)  
plot.show()
```



Komplexere Schaubilder

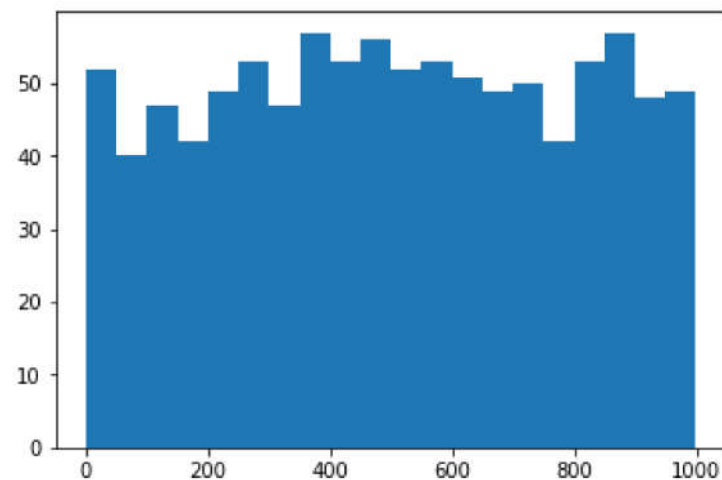
- Bisher haben wir einfache Schaubilder gezeigt
- Matplotlib erlaubt es, so ziemlich alles an einem Schaubild zu manipulieren



Beispiel: Zufallszahlen

- Modul random ermöglicht Erzeugung von Zufallszahlen

```
In [11]: import random
import matplotlib.pyplot as plot
thousand_samples = [random.randrange(1,1000) for i in range(1000)]
#print(thousand_samples)
plot.hist(thousand_samples,bins=20)
plot.show()
```



Beispiel: Kapitalverzinsung

```
In [9]: kapital = 50000
zinssatz = 1.05
jahre = [i for i in range(0,20)]
kapitalNachXJahren = [round(kapital*(zinssatz**jahr),2) for jahr in jahre]
print(kapitalNachXJahren)
```

```
[50000.0, 52500.0, 55125.0, 57881.25, 60775.31, 63814.08, 67004.78, 70355.02, 73872.7
7, 77566.41, 81444.73, 85516.97, 89792.82, 94282.46, 98996.58, 103946.41, 109143.73,
114600.92, 120330.96, 126347.51]
```

```

In [10]: ##matplotlib # gibt den Plot in einem neuen Fenster aus
import matplotlib.pyplot as plot

plot.plot(jahre,kapitalNachXJahren)
plot.scatter(jahre, kapitalNachXJahren)
plot.xticks(jahre)#Achsen-Tick-Beschriftung

plot.xlabel("Anzahl Jahre") #X-Achsen-Beschriftung
plot.ylabel("Kapital") #Y-Achsen-Beschriftung
plot.gca().annotate('Kapital nach 10 Jahren', xy=(10,kapitalNachXJahren[10]), xytext=(10
, kapitalNachXJahren[9]-10000),
                    arrowprops=dict(facecolor='red', shrink=0.25),
                    )
plot.title("Kapitalverzinsung von 50000€ bei einem Zinssatz von 5%")
plot.show()

```

