

# 20230320\_Data\_Bases\_Data\_Mining\_MV4

March 21, 2023

```
[1]: # Kunden Profilierung (Customer Profiling)
```

```
[2]: # Mittelwert
```

```
[4]: !pip install numpy # installation von dem Nummerischen Python (numpy)
```

Requirement already satisfied: numpy in /Users/h4/anaconda3/lib/python3.9/site-packages (1.23.2)

```
[8]: import numpy as np # importieren des Packets numpy (IMMER SCHREIBEN!)
```

```
[9]: speed = [99,86,87,88,111,86,103,87,94,78,77,85,86] # definieren wir einen  
↳ Datensatz: Geschwindigkeit der Autofahrer
```

```
[10]: mittelwert = np.mean(speed) # eine Variable "mittelwert" wurde definiert als  
↳ der numpy-mittelwert vom "speed"
```

```
[11]: print(mittelwert)
```

89.76923076923077

```
[12]: # Median
```

```
[13]: median = np.median(speed) # eine Variable "median" wurde definiert als der  
↳ numpy-median vom "speed"
```

```
[14]: print(median)
```

87.0

```
[15]: # Standard Abweichung
```

```
[16]: std = np.std(speed)
```

```
[17]: print(std) # eine Variable "Standard Abweichung" wurde definiert als der  
↳ numpy-std vom "speed"
```

9.258292301032677

```
[18]: # Varianz
```

```
[21]: varianz = np.var(speed) # eine Variable "varianz" wurde definiert als der  
      ↪ numpy-varianz vom "speed"
```

```
[22]: print(varianz)
```

85.71597633136093

```
[23]: # Datenverteilungen
```

```
[27]: uniform = np.random.uniform(0, 5, 250)  
      # wir eine Variable "uniform" definiert, welche 250 Zufallszahlen "random"  
      ↪ generiert zw. 0 und 5  
      # in einer Konstanten- bzw. Uniformverteilung
```

```
[28]: print(uniform)
```

```
[0.18619833 2.55713375 3.90502339 0.61688931 4.07302038 2.57883155  
4.85216981 4.62934039 2.73788753 2.19111022 3.77665255 3.80559836  
3.00207136 4.22394333 1.12639961 0.15895276 3.77045432 4.82007538  
4.4857832 1.78236464 2.28895605 4.95239682 1.41903019 4.48603896  
2.79389756 4.1055284 3.14954813 4.13959138 0.05458213 4.09868291  
1.87247833 3.84448556 3.77550836 3.49980035 3.24897842 3.85655931  
1.60044684 0.24239797 2.33708641 0.42440267 3.90322061 3.39810531  
1.78101386 2.41550138 3.74546831 2.48294502 2.8385474 0.72076301  
3.31611321 1.93078049 2.74243052 1.75856589 2.36212662 4.70760967  
3.50894454 4.19308551 3.41249601 2.23038135 1.13072441 1.13348392  
0.03798956 3.53377628 2.86848483 3.52820246 3.0111085 1.43458326  
0.91781206 4.28375441 3.5088016 4.32303549 2.89391851 1.67491558  
4.36513984 0.33765587 4.21949671 3.40006586 0.27590784 2.25545177  
3.81645382 2.03527953 2.85416322 4.70603944 4.84231668 3.55120335  
4.66686009 1.87125498 0.28234488 1.18261977 3.51099279 4.61126097  
4.34246837 1.37358764 1.22243483 4.89474013 0.49969974 1.0998422  
1.31024438 4.65630455 0.32088828 3.51298445 3.11781857 0.18359806  
2.55879791 3.73197847 2.7684555 3.75610576 4.90931899 0.04402693  
3.5945671 1.85159907 3.45507917 0.21123022 3.30617688 0.17028107  
4.54868411 2.46916326 4.23250725 0.53984821 4.43604276 1.55770339  
1.17153631 3.24575172 2.42071732 0.14175234 2.4543903 2.83227021  
3.0025745 4.01131652 3.24224532 1.17069585 1.99502634 0.96295534  
4.4272709 3.73516139 2.48068111 1.96589425 0.69668666 2.64037875  
2.451577 1.48301543 1.4742706 1.42810481 2.33415913 1.38344299  
0.09100329 1.17026433 0.26520083 4.05574317 1.04174656 0.14325851  
1.30760319 3.05092776 2.98170701 3.17478373 2.11439024 4.69771276  
1.46877789 2.84619234 2.81462469 4.32665558 3.39634564 3.70095135  
1.54087708 0.67353666 4.60564254 3.8556458 0.99464573 4.73138139  
0.83408528 3.50861395 0.60716146 2.78212436 1.33447024 2.9091149
```

```

1.77733983 2.75299912 0.93807203 4.29578951 4.04972834 2.42433765
2.91858987 2.17382577 2.16616452 2.03707272 1.83957879 2.33798949
0.7111836 4.74796094 0.6490208 2.68282 3.718694 1.37494008
3.73745285 4.91543305 0.24436832 1.1429624 0.38118029 1.96350015
0.7769232 4.66938871 2.85262675 2.55658816 2.3297026 0.74948254
1.93462225 3.15861629 2.42153426 2.54947599 2.95130331 3.66835359
1.71664914 1.12460751 4.79836276 3.20534878 0.01922036 0.90459994
4.55240402 4.21671165 4.95462559 0.37294902 0.70012275 0.8894247
3.38247463 2.42785699 3.05981307 2.93249109 0.31837463 4.82726123
1.85989639 1.85117046 4.32992472 0.7445471 3.85512008 1.70551948
1.97220587 4.99886188 4.67912915 1.66687504 3.62365758 1.71488266
3.39602144 3.61491623 4.56017774 4.92409706 3.20260302 3.00137327
0.03629324 2.23303447 0.60183473 0.37408658]

```

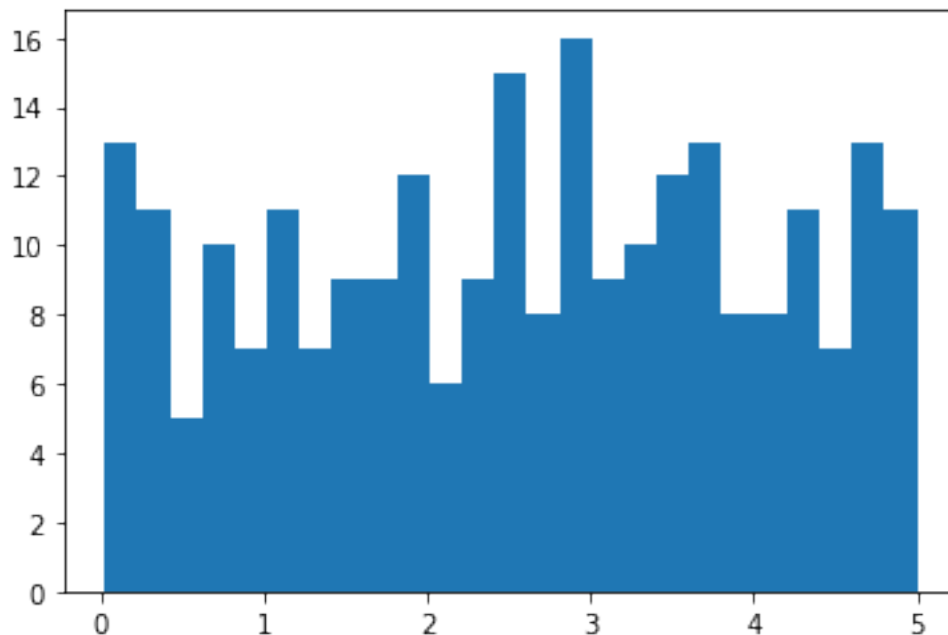
```

[36]: # für die graphische Darstellung benötigen wir "matplotlib.pyplot"

import matplotlib.pyplot as plt

plt.hist(uniform, 25) # darstellung eines Histograms mit 25 Balken von der
    ↪ Uniformverteilung
plt.show()

```



```

[37]: # beispielfrage Prüfung:
# 1. bitte geben Sie den Code für eine Uniformverteilung zw. 0 und 10 mit 500
    ↪ Datensätze

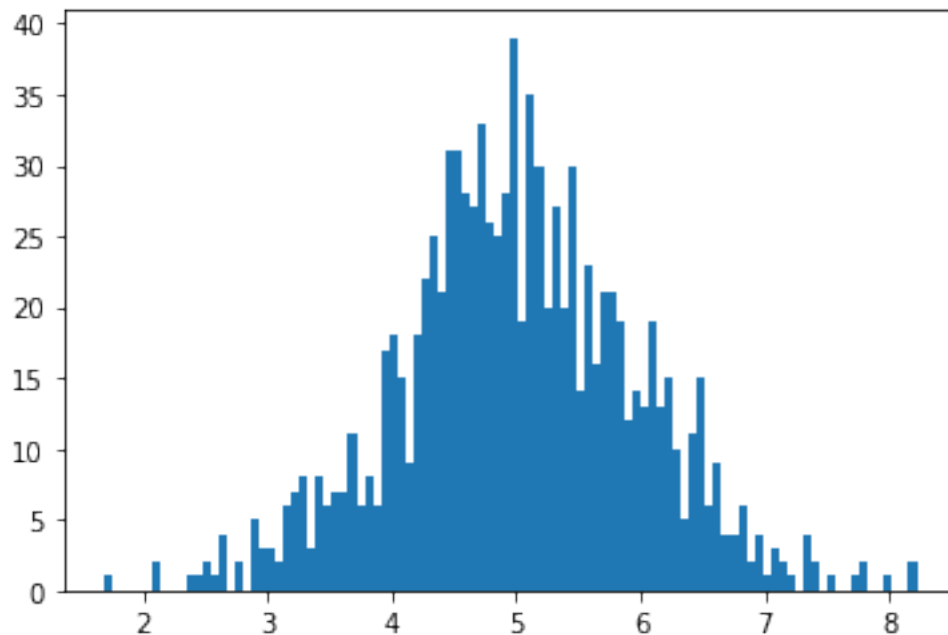
```

# 2. wenn ich Ihnen das Bild einer Uniformverteilung gebe (s.oben), bitte geben  
↪ Sie den Code welche Diese Verteilung erstellt.

```
[38]: # normalverteilung
```

```
[43]: normal = np.random.normal(5, 1, 1000) # normalverteilung mit Mittelwert 5, Std.  
↪ Abweichung 1 und 1000 Datensätze
```

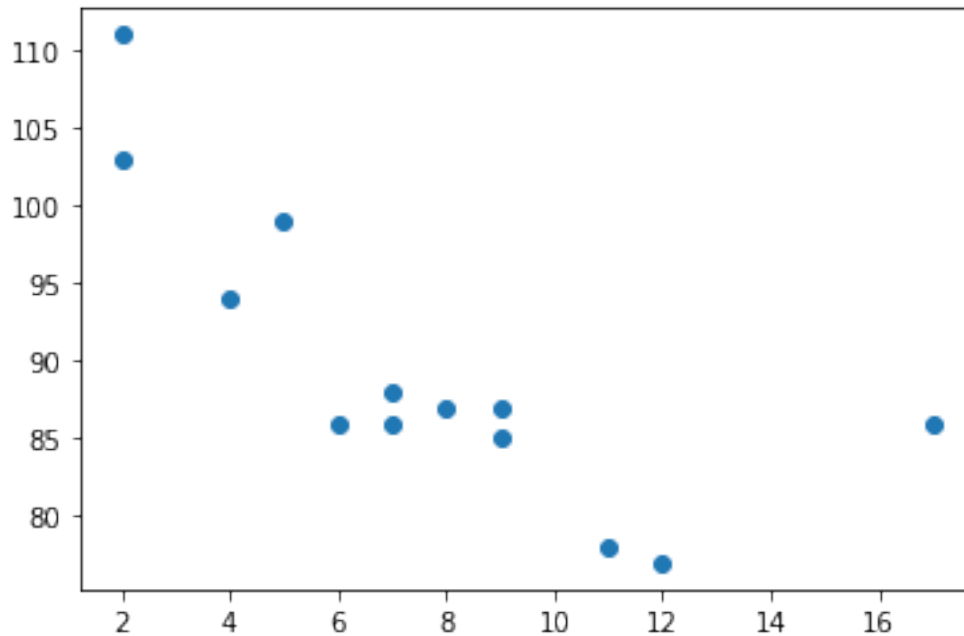
```
[44]: plt.hist(normal, 100) # histogram mit 100 Balken  
plt.show()
```



```
[45]: # 2 Variablen Graphische Darstellung - Scatter Plot
```

```
[46]: x = [5,7,8,7,2,17,2,9,4,11,12,9,6]  
y = [99,86,87,88,111,86,103,87,94,78,77,85,86]
```

```
[47]: plt.scatter(x,y)  
plt.show()
```



```
[48]: !pip install scipy
```

```
Requirement already satisfied: scipy in /Users/h4/anaconda3/lib/python3.9/site-
packages (1.9.1)
Requirement already satisfied: numpy<1.25.0,>=1.18.5 in
/Users/h4/anaconda3/lib/python3.9/site-packages (from scipy) (1.23.2)
```

```
[49]: from scipy import stats
```

```
[52]: slope, intercept, r, p, std_err = stats.linregress(x, y)
```

```
# slope = b1
# intercept = b0
# r, p und std_err können wir ignorieren
```

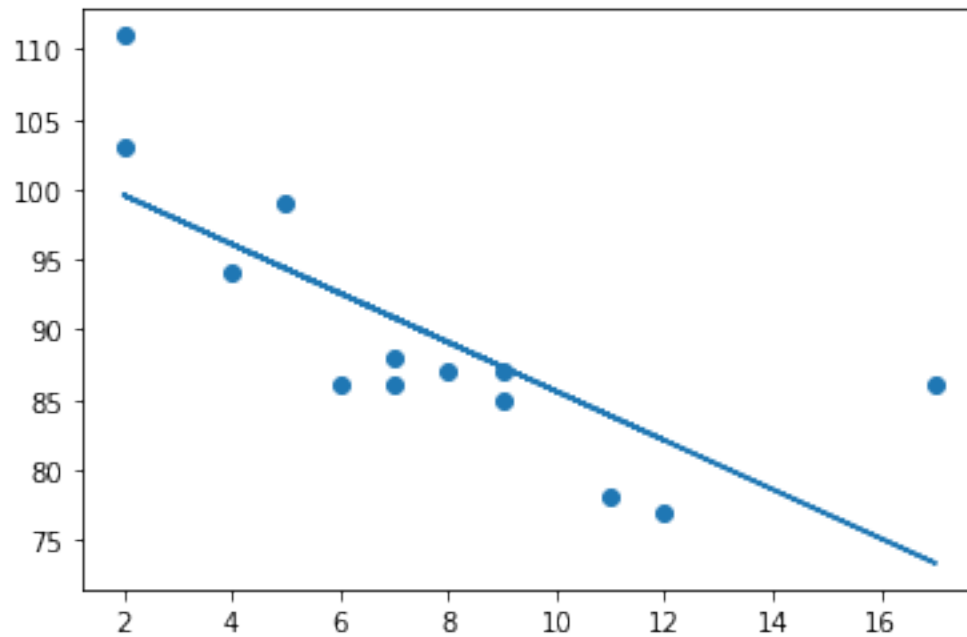
```
[53]: def myfunc(x):
      return slope * x + intercept # b1 * x + b0
```

```
# nicht prüfungsrelevant
```

```
[54]: mymodel = list(map(myfunc, x))
```

```
# nicht prüfungsrelevant
```

```
[57]: plt.scatter(x, y)
plt.plot(x, mymodel)
plt.show()
```



```
[58]: new_customer = myfunc(10)
```

```
[59]: new_customer
```

```
[59]: 85.59308314937454
```

```
[ ]:
```