

---

## Statistik Rechnerüb. SWB 2 Sommersemester 2022, Ag 4

Nachname: [Scholz](#)

Vorname: [Noah](#)

Matrikelnummer: [767535](#)

Semester: [2](#)

Email-Adresse: [noscit00@hs-esslingen.de](mailto:noscit00@hs-esslingen.de)

---

Abgabe-Schlusstermin: Donnerstag, 23.06.2022

# statistik\_labor

June 22, 2022

```
[209]: import pandas as pd
import numpy as np
from scipy.stats import kde, norm
import matplotlib.pyplot as plt
import statistics as st
```

```
[230]: # 1. Daten laden

df1 = pd.read_csv('../sr_aufg_4_35_MR1.txt', sep=' ', decimal=',', header=None)
df2 = pd.read_csv('../sr_aufg_4_35_MR2.txt', sep=' ', decimal=',', header=None)
df3 = pd.read_csv('../sr_aufg_4_35_MR3.txt', sep=' ', decimal=',', header=None)

pd.set_option('display.max_columns', None)

# Get ammount of groups
def groups(df):
    return int(np.sqrt(len(df)))

#b)
#MR1
mean1 = df1.mean()
var1 = df1.var()
std1 = df1.std()
median1 = df1.median()
spn1 = df1.max() - df1.min()

#MR2
mean2 = df2.mean()
var2 = df2.var()
std2 = df2.std()
median2 = df2.median()
spn2 = df2.max() - df2.min()

#MR3
mean3 = df3.mean()
var3 = df3.var()
std3 = df3.std()
```

```

median3 = df3.median()
spn3 = df3.max() - df3.min()

table_data = [['Mittelwert', mean1.values[0], mean2.values[0], mean3.
↳values[0]], ['Varianz', var1.values[0], var2.values[0], var3.
↳values[0]], ['Standardabweichung', std1.values[0], std2.values[0], std3.
↳values[0]], ['Median', median1.values[0], median2.values[0], median3.
↳values[0]], ['Spannweite', spn1.values[0], spn2.values[0], spn3.values[0]]]
table = pd.DataFrame(table_data, columns=['', 'MR1', 'MR2', 'MR3'])
table = table.round(4)

table

```

```

[230]:

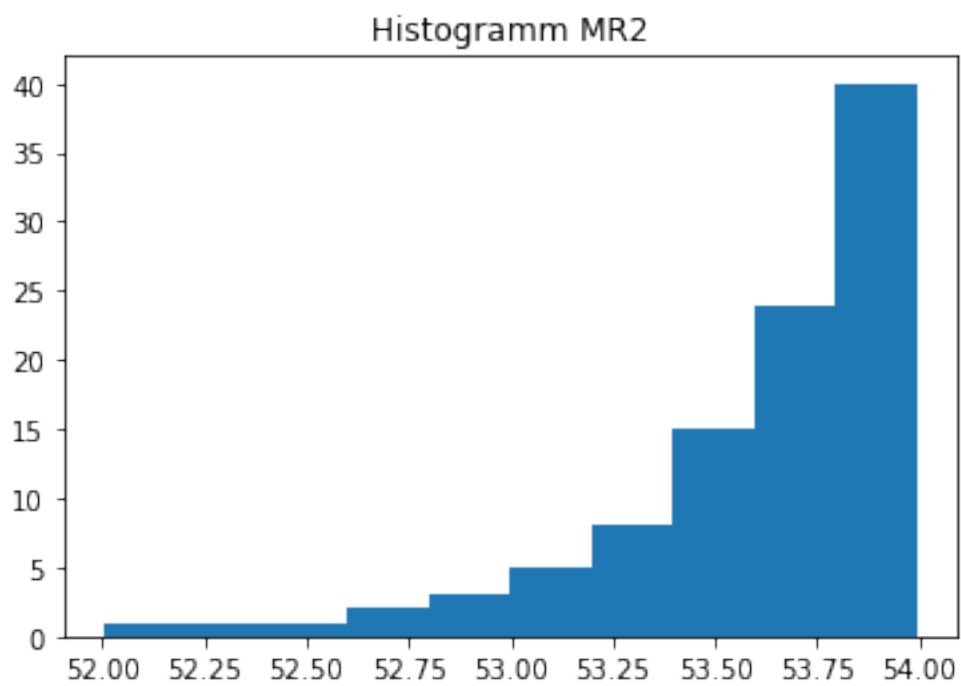
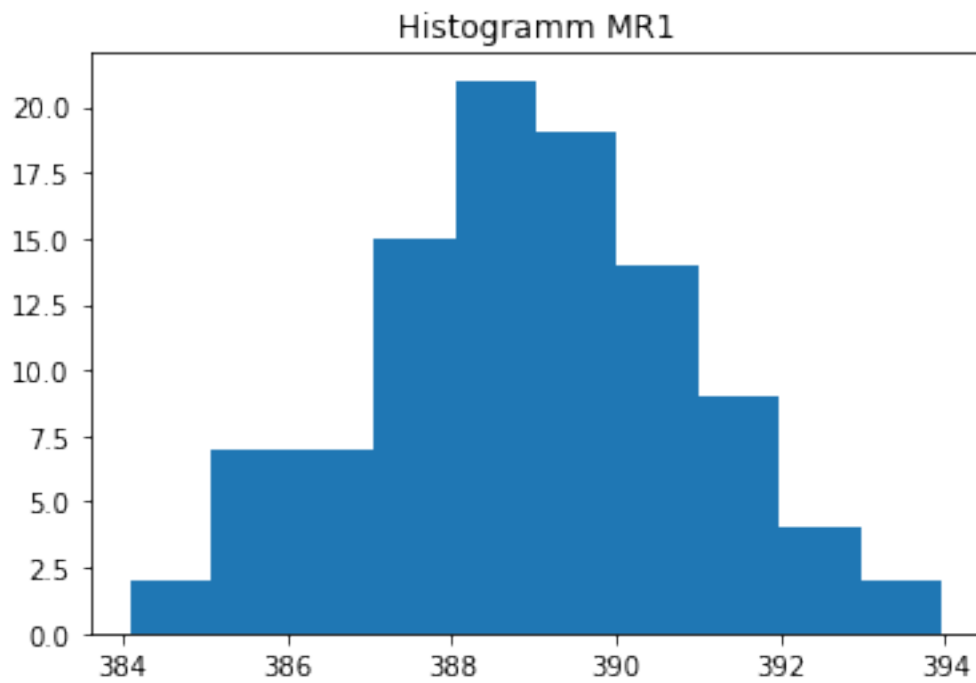
```

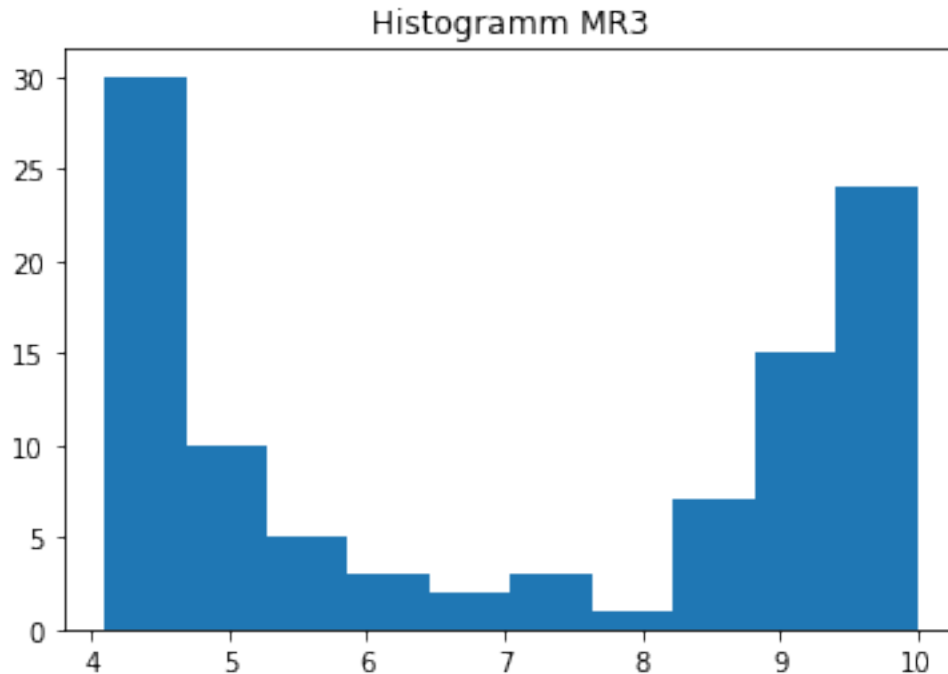
		MR1	MR2	MR3
0	Mittelwert	388.9164	53.6083	7.0063
1	Varianz	3.8441	0.1426	5.4403
2	Standardabweichung	1.9606	0.3777	2.3324
3	Median	388.8469	53.7132	6.9713
4	Spannweite	9.8708	1.9879	5.9050

```

[232]: # c)
# plot histogramm
plt.hist(df1, bins=groups(df1), label='MR1')
plt.title('Histogramm MR1')
plt.show()
plt.hist(df2, bins=groups(df2), label='MR2')
plt.title('Histogramm MR2')
plt.show()
plt.title('Histogramm MR3')
plt.hist(df3, bins=groups(df3), label='MR3')
plt.show()

```



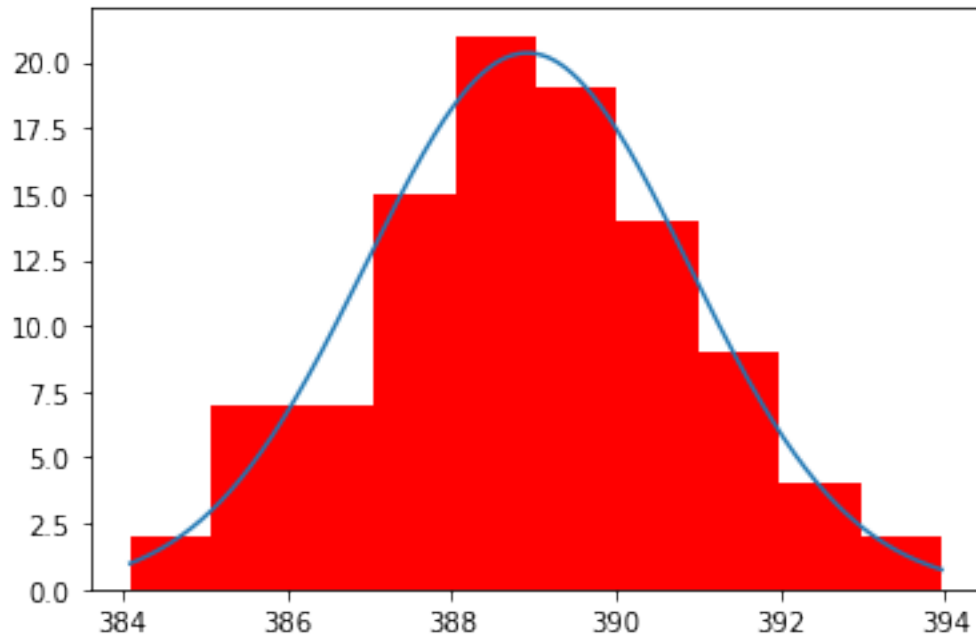


- c) Für jedes Histogramm habe ich eine unterschiedliche Anzahl gewählt. Die Anzahl habe ich so gewählt, dass im Histogramm keine Lücken entstehen. Zum erstellen des Histogramms habe ich die Python funktion `hist()` der Bibliothek `matplotlib.pyplot` benutzt.
- e) Die Messreihe 1 ist Normalverteilt, da das Histogramm stark einer Gauß-Glocke oder auch Glockenkurve ähnelt.

```
[257]: # e)
axis = np.arange(df1[0].min(),df1[0].max(),0.01)
nv = norm.pdf(axis, mean1, std1)

plt.hist(df1, bins=groups(df1), label='MR1', color='red')
plt.plot(axis, nv*100)

plt.show()
```



```
[260]: sigma = std1[0]
mu = mean1[0]
print(sigma, mu)
```

```
1.9606376097181832 388.91642499999995
```

f.1 Die im Histogramm dargestellten Anzahlen unterscheiden sich um einen Faktor von 100 von den Funktionswerten der Normalverteilungsdicht

f.2 Die Skalierung der Y-Achse musste nicht geändert werden, statt dessen wurden die Funktionswerte der Dichtefunktion mit einem Faktor von 100 multipliziert. So passte die Skalierung wieder