

Hausaufgabe 4 a)

```
In [1]: import math
data = [80,81,84,74,93,80,108,62,81,51,71,57,80,
        75,77,60,86,50,89,54,90,73,60,83,54,85,58,
        79,57,88,68,85,75,65,76,58,91,50,54,86,53,
        78,52,83,60,60,92,43,89,60,84,69,50,77,57,
        80,61,82,48,79,54,80,73,81,62,81,74,59,81,
        66,87,53,80,82,58,81,49,92,50,88,77,65,76,
        87,87,74,81,71,50,62,56,82,78,48,49,71,73,
        79,87,93]
n=len(data)
n
```

Out[1]: 100

```
In [2]: ordnungstatistik = sorted(data)
print(ordnungstatistik)

[43, 48, 48, 49, 49, 50, 50, 50, 50, 50, 51, 52, 53, 53, 54, 54, 54, 54, 56,
57, 57, 57, 58, 58, 58, 59, 60, 60, 60, 60, 60, 61, 62, 62, 62, 65, 65, 66, 6
8, 69, 71, 71, 71, 73, 73, 73, 74, 74, 74, 75, 75, 76, 76, 77, 77, 77, 78, 7
8, 79, 79, 79, 80, 80, 80, 80, 80, 80, 81, 81, 81, 81, 81, 81, 81, 82, 82, 8
2, 83, 83, 84, 84, 85, 85, 86, 86, 87, 87, 87, 87, 88, 88, 89, 89, 90, 91, 9
2, 92, 93, 93, 108]
```

```
In [3]: durchschnitt = sum(ordnungstatistik)/len(ordnungstatistik)
print("durchschnitt",durchschnitt)

durchschnitt 71.62
```

```
In [4]: # Achtung: python listen sind 0-indiziert!
achte_quantil = 1/2* (ordnungstatistik[int(n*0.08)-1]+ordnungstatistik[int(n
*0.08)])
zweiundneunzigstes_quantil = 1/2* (ordnungstatistik[int(n*0.92)-1]+ordnungst
atistik[int(n*0.92)])
median = 1/2* (ordnungstatistik[int(n*0.5)-1]+ordnungstatistik[int(n*0.5)])
print("8-quantil",achte_quantil,ordnungstatistik[int(n*0.08)-1],ordnungstati
stik[int(n*0.08)])
print("92-quantil",zweiundneunzigstes_quantil,ordnungstatistik[int(n*0.92)-
1],ordnungstatistik[int(n*0.92)])
print("median",median,ordnungstatistik[int(n*0.5)-1],ordnungstatistik[int(n*
0.5)])

8-quantil 50.0 50 50
92-quantil 89.0 89 89
median 75.0 75 75
```

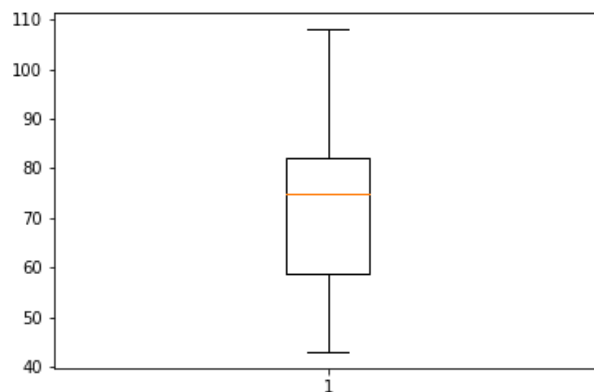
```
In [5]: std_empirisch = ((1/(len(ordnungstatistik)-1)) *sum([(x-durchschnitt)**2 for
x in ordnungstatistik]))**0.5
print("std_empirisch", std_empirisch)
std_normal = ((1/(len(ordnungstatistik))) *sum([(x-durchschnitt)**2 for x in
ordnungstatistik]))**0.5
print("std_normal",std_normal)

std_empirisch 14.152688290872012
std_normal 14.081747050703614
```

Teil b

```
In [6]: import matplotlib.pyplot as plt
import numpy as np
```

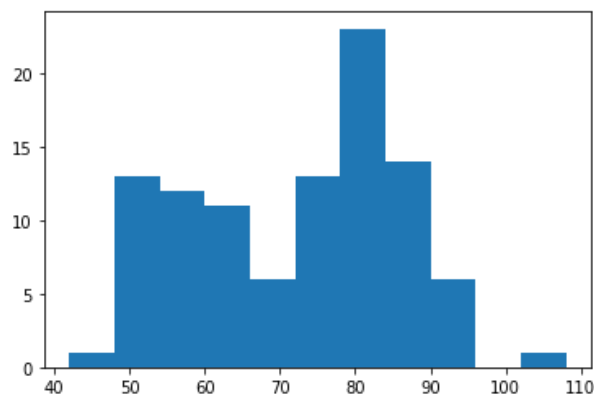
```
In [7]: l_quartil = 1/2* (ordnungstatistik[int(n*0.25)-1]+ordnungstatistik[int(n*0.25)])
u_quartil = 1/2* (ordnungstatistik[int(n*0.75)-1]+ordnungstatistik[int(n*0.75)])
plt.boxplot(ordnungstatistik)
plt.show()
#sanity check für die Quartile
print("unteres quartil",l_quartil, "oberes quartil", u_quartil)
```



unteres quartil 58.5 oberes quartil 82.0

Teilaufgabe c

```
In [8]: #manuelle berechnung der bins
bins = np.ones((max(ordnungstatistik)-42)//6+1)*6
bins[0] = 42
bins = np.cumsum(bins)
plt.hist(ordnungstatistik,bins=bins, label = map(str,bins))
plt.show()
```



Aufgabe 5

für die minimierung ableiten w.r.t. a:

$$\nabla_a \sum_{i=1}^n (x_i - a)^2 = \sum_{i=1}^n \nabla_a (x_i - a)^2 = \sum_{i=1}^n \nabla_a (x_i^2 - 2ax_i + a^2) = \sum_{i=1}^n (2a - 2x_i) = n * 2a - \sum_{i=1}^n 2x_i \rightarrow$$

auflösen:

$$n * 2a - \sum_{i=1}^n 2x_i = 0 \iff n * a - \sum_{i=1}^n x_i = 0 \iff a - \frac{1}{n} \sum_{i=1}^n x_i = 0 \text{ Der Letzte schritt geht, weil } n > 0$$

für alle möglichen messreihen gilt. da $\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$ und die Funktion konvex ist, muss im minimum $a = \bar{x}$ gelten. (bzw

$$f''(a) = n * 2a \frac{\partial}{\partial a} = 2n > 0 (\forall n > 0))$$

In []: