

Mathe Übung 2 - Jasmin Piott

1. Varianten

$$1) (2, 4, 4, 4, 5) \quad n = 5 \quad \bar{x} = 19/5 = 3,8$$

$$s^2 = \frac{1}{5-1} \cdot [(2-19/5)^2 + (4-19/5)^2 + (4-19/5)^2 + (4-19/5)^2 + (5-19/5)^2]$$

$$\frac{1}{4} \cdot ((-1,8)^2 + (0,2)^2 + (0,2)^2 + (0,2)^2 + (1,2)^2)$$

$$\frac{1}{4} \cdot (3,24 + 0,04 + 0,04 + 0,04 + 1,44)$$

$$\frac{1}{4} \cdot (4,8) = \underline{\underline{1,2}} = s^2 \quad +$$

$$2) (3, 4, 5, 3, 2) \quad n = 5 \quad \bar{x} = 17/5 = 3,4$$

$$s^2 = \frac{1}{5-1} [(3-3,4)^2 + (4-3,4)^2 + (5-3,4)^2 + (3-3,4)^2 + (2-3,4)^2]$$

$$= \frac{1}{4} \cdot ((-0,4)^2 + (0,6)^2 + (1,6)^2 + (-0,4)^2 + (-1,4)^2)$$

$$= \frac{1}{4} \cdot (0,16 + 0,36 + 2,56 + 0,16 + 1,96)$$

$$= \frac{1}{4} \cdot 5,2 = \underline{\underline{1,3}} = s^2 \quad +$$

$$3) (120, 130, 110, 125, 140) \quad n = 5 \quad \bar{x} = 125$$

$$s^2 = \frac{1}{5-1} \cdot ((-5)^2 + (5)^2 + (-15)^2 + (0)^2 + (15)^2)$$

$$= \frac{1}{4} \cdot (25 + 25 + 225 + 225) = \frac{1}{4} \cdot (500)$$

$$\underline{\underline{s^2 = 125}} \quad +$$

$$4) (12,1, 11,9, 12,0, 12,2, 11,8) \quad n = 5 \quad \bar{x} = 12$$

$$s^2 = ((0,1)^2 + (-0,1)^2 + (0)^2 + (0,2)^2 + (-0,2)^2) \cdot \frac{1}{4}$$

$$= 0,25 \cdot 0,1 = \underline{\underline{0,025}} = s^2 \quad +$$

2. Standardabweichung

- 1) $s^2 = 1,2$ $s = \sqrt{1,2} = 1,09544 \approx 1,1$ +
- 2) $s^2 = 1,3$ $s = \sqrt{1,3} = 1,14017 \approx 1,14$ +
- 3) $s^2 = 125$ $s = \sqrt{125} = 11,18033 \approx 11,18$ +
- 4) $s^2 = 0,025$ $s = \sqrt{0,025} = 0,15811 \approx 0,16$ +

3. Spannweite

- 1) (2, 4, 4, 4, 5) Range = $5 - 2 = \underline{3}$ +
- 2) (3, 4, 5, 3, 2) Range = $5 - 2 = \underline{3}$ +
- 3) (120, 130, 110, 125, 140) Range = $140 - 110 = \underline{30}$ +
- 4) (12,1; 11,9; 12,0; 12,2; 11,8) Range = $12,2 - 11,8 = \underline{0,4}$ +

4. Skalenniveaus und Statistische Maße

- 1) (Ford, BMW, Ford, Tesla, BMW, VW)

$$x_{Mo_1} = \text{BMW} \quad x_{Mo_2} = \text{Ford} \quad +$$

- 2) (3 ✕, 4 ✕, 2 ✕, 3 ✕, 5 ✕, 4 ✕)

$$\text{Median: } 2, 3, \underline{3}, 4, 4, 5$$

$$x_{Me} = \underline{3,5} \quad +$$

$$\text{Modus: } x_{Mo_1} = 3 \quad x_{Mo_2} = 4 \quad +$$

- 3) (21, 22, 20, 21, 23, 21)

$$\bar{x} = (20 + 21 + 21 + 21 + 22 + 23) / 6 = 128 / 6 = 21,3 \quad +$$

$$x_{Me} = 21 \quad +$$

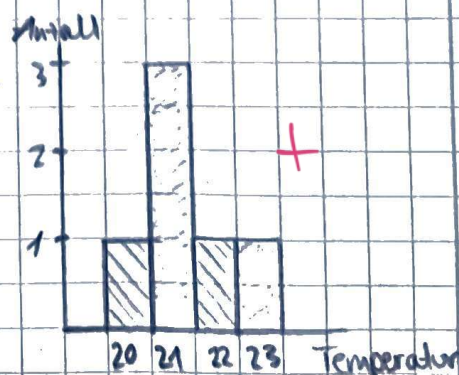
$$s^2 = \frac{1}{6-1} \cdot ((20-21,3)^2 + (21-21,3)^2 + (21-21,3)^2 + (21-21,3)^2 + (22-21,3)^2 + (23-21,3)^2)$$

$$= \frac{1}{5} \cdot (1,69 + 0,09 + 0,09 + 0,09 + 0,49 + 2,89)$$

$$= \frac{1}{5} \cdot (5,34) = \underline{1,068} = s^2 \quad +$$

$$s = \sqrt{1,068} = 1,03344 \quad +$$

$$\text{Range} = 23 - 20 = 3 \quad +$$



4) (4000, 4500, 4200, 4600, 4800, 4500)

$$\bar{x} = 4433, \bar{3} +$$

$$\text{Median} = 4000, 4200, \underline{4500, 4500}, 4600, 4800$$

$$x_{Me} = 4500 +$$

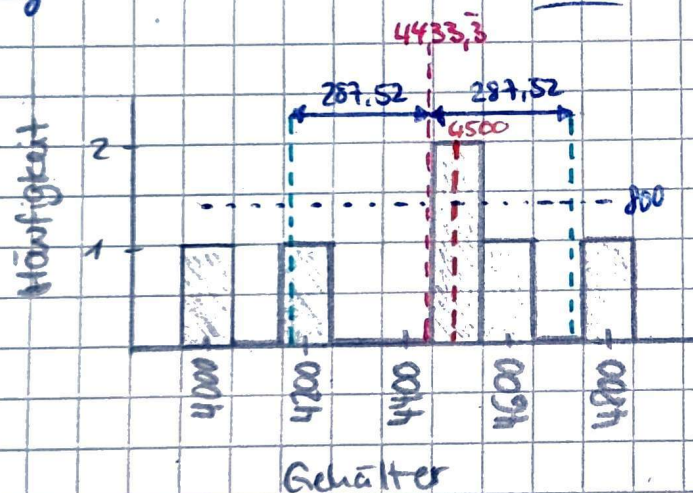
$$s^2 = \frac{1}{6-1} \cdot ((4000-4433,3)^2 + (4200-4433,3)^2 + (4600-4433,3)^2 + (4800-4433,3)^2 + 2(4500-4433,3)^2)$$

$$= \frac{1}{5} \cdot (187748,89 + 54428,89 + 27788,89 + 134468,89 + 8897,78)$$

$$s^2 = 0,2 \cdot (413333,34) = 82666,668 +$$

$$s = \sqrt{82666,668} = 287,518117 +$$

$$\text{Range} = 4800 - 4000 = \underline{800} +$$



$$--- \bar{x} \quad 4433, \bar{3}$$

--- Population Std.-Abw.

--- Median 4500

--- Spannweite 800

16

Mathe Übung 2 Jasmin Piott

March 28, 2024

Jasmin Piott

1 Aufgabe 1

1.1 (1)

```
[96]: def variance(data):  
    n = len(data)  
    mean = sum(data) / n  
    variance = sum((x - mean) ** 2 for x in data) / (n - 1)  
    return variance  
  
# Stichprobe  
stichprobe = [2, 4, 4, 4, 5]  
  
# Varianz berechnen  
varianz = variance(stichprobe)  
  
print("Stichprobe:", stichprobe)  
print("Varianz:", varianz)
```

Stichprobe: [2, 4, 4, 4, 5]

Varianz: 1.2

1.2 (2)

```
[97]: def variance(data):  
    n = len(data)  
    mean = sum(data) / n  
    variance = sum((x - mean) ** 2 for x in data) / (n - 1)  
    return variance  
  
# Stichprobe  
stichprobe = [3, 4, 5, 3, 2]  
  
# Varianz berechnen  
varianz = variance(stichprobe)
```

```
print("Stichprobe:", stichprobe)
print("Varianz:", varianz)
```

Stichprobe: [3, 4, 5, 3, 2]

Varianz: 1.3

1.3 (3)

```
[98]: def variance(data):
      n = len(data)
      mean = sum(data) / n
      variance = sum((x - mean) ** 2 for x in data) / (n - 1)
      return variance

      # Stichprobe
      stichprobe = [120, 130, 110, 125, 140]

      # Varianz berechnen
      varianz = variance(stichprobe)

      print("Stichprobe:", stichprobe)
      print("Varianz:", varianz)
```

Stichprobe: [120, 130, 110, 125, 140]

Varianz: 125.0

1.4 (4)

```
[99]: def variance(data):
      n = len(data)
      mean = sum(data) / n
      variance = sum((x - mean) ** 2 for x in data) / (n - 1)
      return variance

      # Stichprobe
      stichprobe = [12.1, 11.9, 12.0, 12.2, 11.8]

      # Varianz berechnen
      varianz = variance(stichprobe)

      print("Stichprobe:", stichprobe)
      print("Varianz:", varianz)
```

Stichprobe: [12.1, 11.9, 12.0, 12.2, 11.8]

Varianz: 0.024999999999999982

2 Aufgabe 2

2.1 (1)

```
[100]: import math

def variance(data):
    n = len(data)
    mean = sum(data) / n
    variance = sum((x - mean) ** 2 for x in data) / (n - 1)
    return variance

def standard_deviation(data):
    var = variance(data)
    std_dev = math.sqrt(var)
    return std_dev

# Stichprobe
stichprobe = [2, 4, 4, 4, 5]

# Standardabweichung berechnen
std_abweichung = standard_deviation(stichprobe)

# Runde auf 2 Nachkommastellen
std_abweichung_rounded = round(std_abweichung, 2)

print("Stichprobe:", stichprobe)
print("Standardabweichung:", std_abweichung)
print("Standardabweichung:", std_abweichung_rounded)
```

```
Stichprobe: [2, 4, 4, 4, 5]
Standardabweichung: 1.0954451150103321
Standardabweichung: 1.1
```

2.2 (2)

```
[101]: import math

def variance(data):
    n = len(data)
    mean = sum(data) / n
    variance = sum((x - mean) ** 2 for x in data) / (n - 1)
    return variance

def standard_deviation(data):
    var = variance(data)
    std_dev = math.sqrt(var)
    return std_dev
```

```

# Stichprobe
stichprobe = [3, 4, 5, 3, 2]

# Standardabweichung berechnen
std_abweichung = standard_deviation(stichprobe)

# Runde auf 2 Nachkommastellen
std_abweichung_rounded = round(std_abweichung, 2)

print("Stichprobe:", stichprobe)
print("Standardabweichung:", std_abweichung)
print("Standardabweichung:", std_abweichung_rounded)

```

Stichprobe: [3, 4, 5, 3, 2]
Standardabweichung: 1.140175425099138
Standardabweichung: 1.14

2.3 (3)

```

[102]: import math

def variance(data):
    n = len(data)
    mean = sum(data) / n
    variance = sum((x - mean) ** 2 for x in data) / (n - 1)
    return variance

def standard_deviation(data):
    var = variance(data)
    std_dev = math.sqrt(var)
    return std_dev

# Stichprobe
stichprobe = [120, 130, 110, 125, 140]

# Standardabweichung berechnen
std_abweichung = standard_deviation(stichprobe)

# Runde auf 2 Nachkommastellen
std_abweichung_rounded = round(std_abweichung, 2)

print("Stichprobe:", stichprobe)
print("Standardabweichung:", std_abweichung)
print("Standardabweichung:", std_abweichung_rounded)

```

Stichprobe: [120, 130, 110, 125, 140]

Standardabweichung: 11.180339887498949

Standardabweichung: 11.18

2.4 (4)

```
[103]: import math

def variance(data):
    n = len(data)
    mean = sum(data) / n
    variance = sum((x - mean) ** 2 for x in data) / (n - 1)
    return variance

def standard_deviation(data):
    var = variance(data)
    std_dev = math.sqrt(var)
    return std_dev

# Stichprobe
stichprobe = [12.1, 11.9, 12.0, 12.2, 11.8]

# Standardabweichung berechnen
std_abweichung = standard_deviation(stichprobe)

# Runde auf 2 Nachkommastellen
std_abweichung_rounded = round(std_abweichung, 2)

print("Stichprobe:", stichprobe)
print("Standardabweichung:", std_abweichung)
print("Standardabweichung:", std_abweichung_rounded)
```

Stichprobe: [12.1, 11.9, 12.0, 12.2, 11.8]

Standardabweichung: 0.1581138830084184

Standardabweichung: 0.16

3 Aufgabe 3

3.1 (1)

```
[104]: def spannweite(data):
        return max(data) - min(data)

# Stichprobe
stichprobe = [2, 4, 4, 4, 5]

# Spannweite berechnen
spannweite_stichprobe = spannweite(stichprobe)
```



```
print("Stichprobe:", stichprobe)
print("Spannweite:", spannweite_stichprobe)
```

Stichprobe: [2, 4, 4, 4, 5]
Spannweite: 3

3.2 (2)

```
[105]: def spannweite(data):
        return max(data) - min(data)

        # Stichprobe
        stichprobe = [3, 4, 5, 3, 2]

        # Spannweite berechnen
        spannweite_stichprobe = spannweite(stichprobe)

        print("Stichprobe:", stichprobe)
        print("Spannweite:", spannweite_stichprobe)
```

Stichprobe: [3, 4, 5, 3, 2]
Spannweite: 3

3.3 (3)

```
[106]: def spannweite(data):
        return max(data) - min(data)

        # Stichprobe
        stichprobe = [120, 130, 110, 125, 140]

        # Spannweite berechnen
        spannweite_stichprobe = spannweite(stichprobe)

        print("Stichprobe:", stichprobe)
        print("Spannweite:", spannweite_stichprobe)
```

Stichprobe: [120, 130, 110, 125, 140]
Spannweite: 30

3.4 (4)

```
[107]: def spannweite(data):
        return max(data) - min(data)

        # Stichprobe
```

```

stichprobe = [12.1, 11.9, 12.0, 12.2, 11.8]

# Spannweite berechnen
spannweite_stichprobe = spannweite(stichprobe)

# Runde auf 2 Nachkommastellen
spannweite_rounded = round(spannweite_stichprobe, 2)

print("Stichprobe:", stichprobe)
print("Spannweite:", spannweite_stichprobe)
print("Spannweite:", spannweite_rounded)

```

```

Stichprobe: [12.1, 11.9, 12.0, 12.2, 11.8]
Spannweite: 0.39999999999999986
Spannweite: 0.4

```

4 Aufgabe 4

4.1 (1)

```

[108]: #Diesen Code und Antwort bitte nicht zur Bewertung heranziehen, dieser ist nur
        ↳ der Vollständigkeit halber im Dokument verblieben.
        #Wegen falscher Lösung neuen Code angefordert (siehe darunterliegender Code).
        from statistics import mode

        # Datensatz von Automarken
        automarken = ['Ford', 'BMW', 'Ford', 'Tesla', 'BMW', 'VW']

        # Modus bestimmen
        modus = mode(automarken)

        print("Datensatz von Automarken:", automarken)
        print("Modus:", modus)

```

```

Datensatz von Automarken: ['Ford', 'BMW', 'Ford', 'Tesla', 'BMW', 'VW']
Modus: Ford

```

```

[109]: from collections import Counter

        def modes(data):
            counter = Counter(data)
            max_count = max(counter.values())
            return [val for val, count in counter.items() if count == max_count]

        # Datensatz von Automarken
        automarken = ['Ford', 'BMW', 'Ford', 'Tesla', 'BMW', 'VW']

```

```

# Moden bestimmen
moden = modes(automarken)

print("Datensatz von Automarken:", automarken)
print("Moden:", moden)

```

Datensatz von Automarken: ['Ford', 'BMW', 'Ford', 'Tesla', 'BMW', 'VW']
 Moden: ['Ford', 'BMW']

4.2 (2)

```

[110]: import matplotlib.pyplot as plt
from statistics import median
from collections import Counter

def modes(data):
    counter = Counter(data)
    max_count = max(counter.values())
    return [val for val, count in counter.items() if count == max_count]

# Datensatz von Film-Bewertungen
bewertungen = ['3 Sterne', '4 Sterne', '2 Sterne', '3 Sterne', '5 Sterne', '4_
↳Sterne']

# Wandle die Bewertungen in numerische Werte um
# Angenommen: 1 Stern = 1, 2 Sterne = 2, usw.
numerische_bewertungen = [int(bewertung.split()[0]) for bewertung in_
↳bewertungen]

# Median berechnen
median_bewertungen = median(numerische_bewertungen)

# Moden bestimmen
moden_bewertungen = modes(numerische_bewertungen)

# Balkendiagramm für die Verteilung
counter = Counter(numerische_bewertungen)
labels, values = zip(*counter.items())

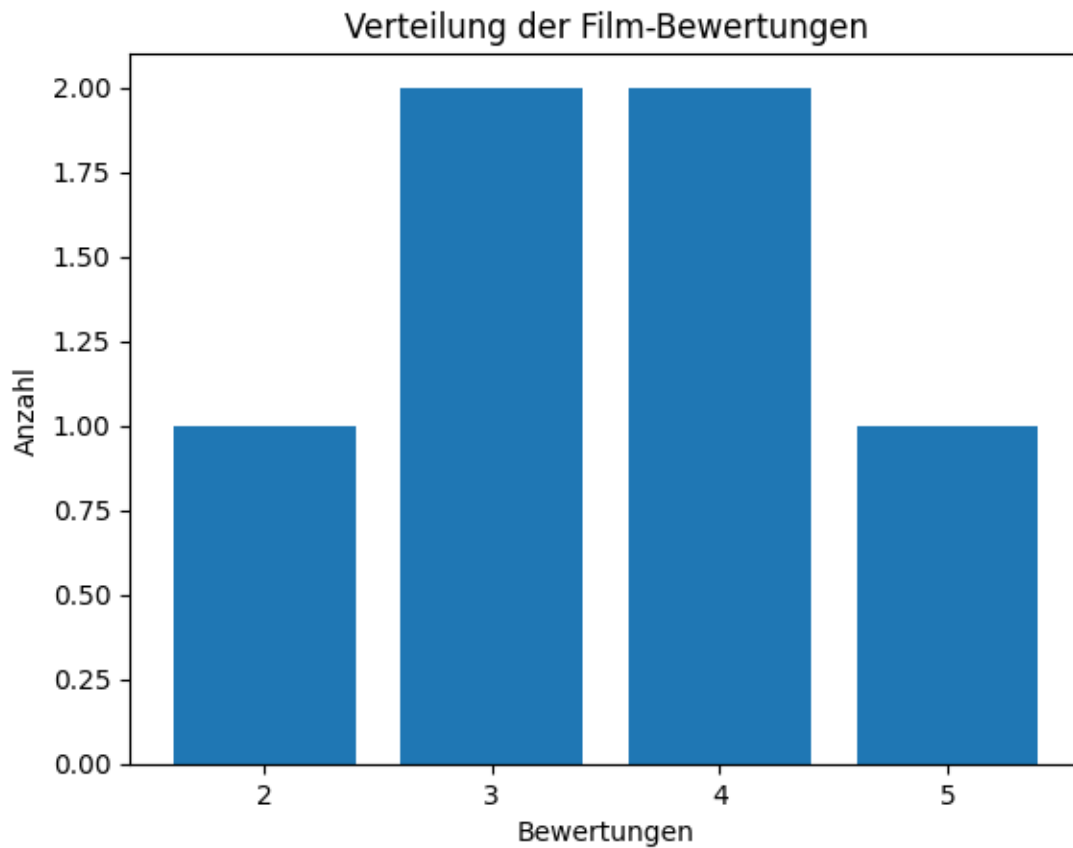
plt.bar(labels, values)
plt.xlabel('Bewertungen in Sternen')
plt.ylabel('Anzahl')
plt.title('Verteilung der Film-Bewertungen')
plt.xticks(labels) # Setze die x-Achsenbeschriftungen auf die Bewertungen
plt.show()

print("Datensatz von Film-Bewertungen:", bewertungen)

```



```
print("Numerische Bewertungen:", numerische_bewertungen)
print("Median:", median_bewertungen)
print("Modus:", moden_bewertungen)
```



Datensatz von Film-Bewertungen: ['3 Sterne', '4 Sterne', '2 Sterne', '3 Sterne', '5 Sterne', '4 Sterne']

Numerische Bewertungen: [3, 4, 2, 3, 5, 4]

Median: 3.5

Modus: [3, 4]

4.3 (3)

```
[111]: import matplotlib.pyplot as plt
import numpy as np
from statistics import mean, median, variance, stdev

# Datensatz von Temperaturen
temperaturen = [21, 22, 20, 21, 23, 21]

# Arithmetisches Mittel berechnen
```

```

mittelwert = mean(temperaturen)

# Median berechnen
median_temperatur = median(temperaturen)

# Varianz berechnen
varianz_temperatur = variance(temperaturen)

# Standardabweichung berechnen
std_abweichung_temperatur = stdev(temperaturen)

# Spannweite berechnen
spannweite_temperatur = max(temperaturen) - min(temperaturen)

# Histogramm für die Verteilung
bin_edges = np.arange(min(temperaturen) - 0.5, max(temperaturen) + 1.5, 1)
plt.hist(temperaturen, bins=bin_edges, edgecolor='black', alpha=0.7)
plt.xlabel('Temperaturen')
plt.ylabel('Anzahl')
plt.title('Verteilung der Temperaturen')
plt.xticks(np.arange(min(temperaturen), max(temperaturen) + 1))
plt.grid(axis='y', alpha=0.75)

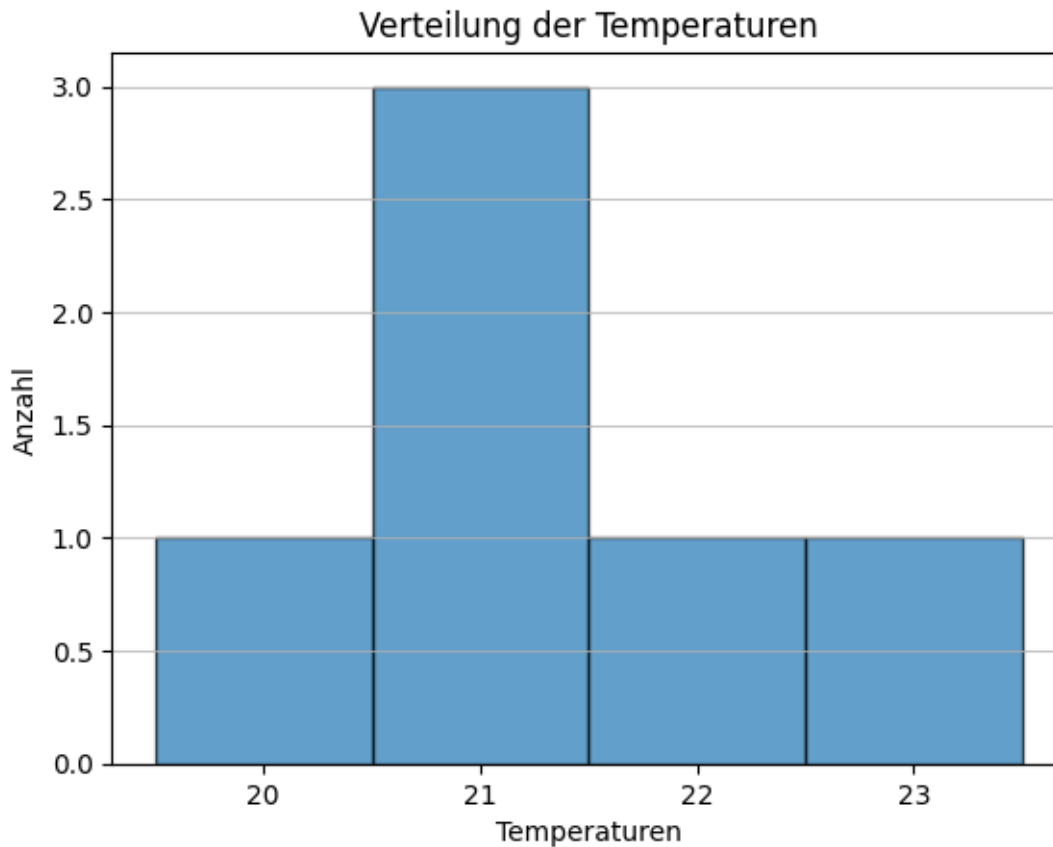
print("Datensatz von Temperaturen:", temperaturen)
print("Arithmetisches Mittel:", mittelwert)
print("Median:", median_temperatur)
print("Varianz:", varianz_temperatur)
print("Standardabweichung:", std_abweichung_temperatur)
print("Spannweite:", spannweite_temperatur)

```

```

Datensatz von Temperaturen: [21, 22, 20, 21, 23, 21]
Arithmetisches Mittel: 21.333333333333332
Median: 21.0
Varianz: 1.0666666666666667
Standardabweichung: 1.0327955589886444
Spannweite: 3

```



4.4 (4)

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

# Gehälter
gehaelter = [4000, 4500, 4200, 4600, 4800, 4500]

# Berechnungen
arithmetisches_mittel = np.mean(gehaelter)
median = np.median(gehaelter)
varianz = np.var(gehaelter, ddof=1) # Stichprobenvarianz
standardabweichung = np.std(gehaelter, ddof=1) # Standardabweichung mit ↵
↵ Stichprobengröße n-1
spannweite = np.ptp(gehaelter) # peak to peak, entspricht der Spannweite

# Histogramm
plt.hist(gehaelter, bins=5, edgecolor='black')
```

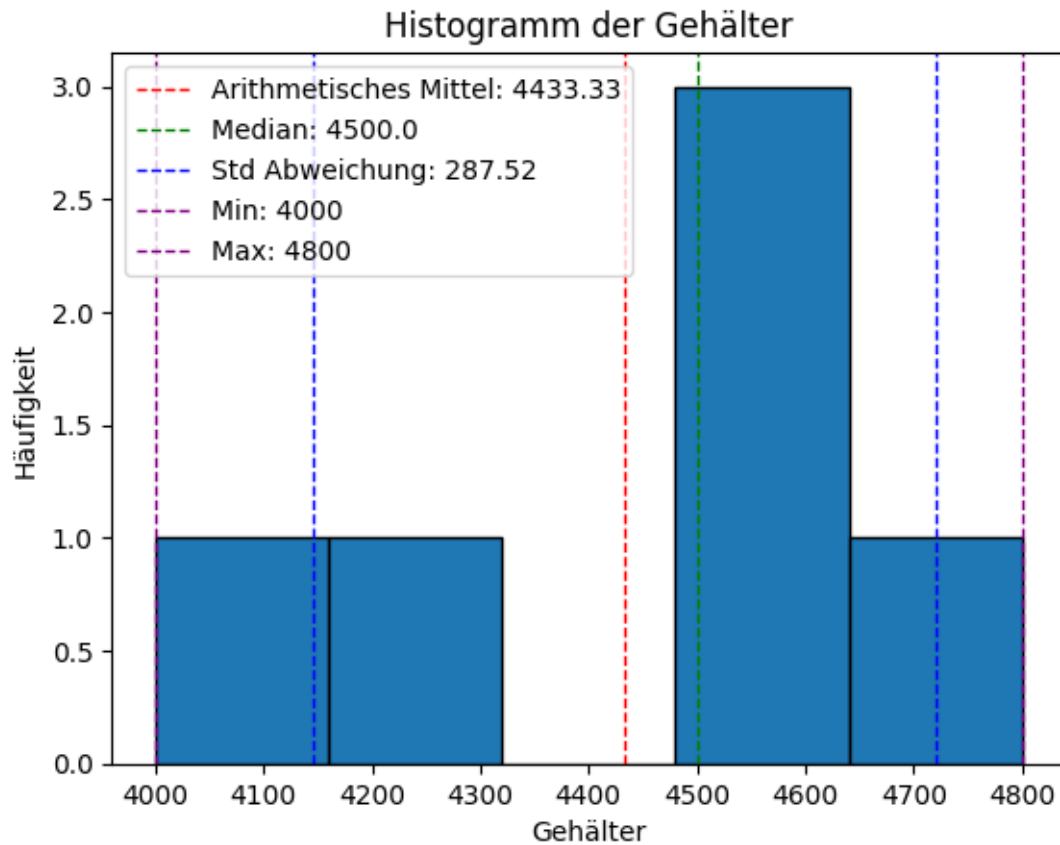


```

plt.axvline(arithmetisches_mittel, color='red', linestyle='dashed',
    ↳linewidth=1, label=f'Arithmetisches Mittel: {arithmetisches_mittel:.2f}')
plt.axvline(median, color='green', linestyle='dashed', linewidth=1,
    ↳label=f'Median: {median}')
plt.axvline(arithmetisches_mittel - standardabweichung, color='blue',
    ↳linestyle='dashed', linewidth=1, label=f'Std Abweichung: {standardabweichung:
    ↳.2f}')
plt.axvline(arithmetisches_mittel + standardabweichung, color='blue',
    ↳linestyle='dashed', linewidth=1)
plt.axvline(min(gehaelter), color='purple', linestyle='dashed', linewidth=1,
    ↳label=f'Min: {min(gehaelter)}')
plt.axvline(max(gehaelter), color='purple', linestyle='dashed', linewidth=1,
    ↳label=f'Max: {max(gehaelter)}')
plt.xlabel('Gehälter')
plt.ylabel('Häufigkeit')
plt.title('Histogramm der Gehälter')
plt.legend()
plt.show()

# Ausgabe der berechneten Werte
print("Datensatz von Gehältern:", gehaelter)
print(f'Arithmetisches Mittel: {arithmetisches_mittel:.2f}')
print(f'Median: {median}')
print(f'Varianz: {varianz:.2f}')
print(f'Standardabweichung: {standardabweichung:.2f}')
print(f'Spannweite: {spannweite}')

```



Datensatz von Gehältern: [4000, 4500, 4200, 4600, 4800, 4500]

Arithmetisches Mittel: 4433.33

Median: 4500.0

Varianz: 82666.67

Standardabweichung: 287.52

Spannweite: 800

[]: