DataCraft

2 Lagemaße

DataCraft

Mittelwert = Durchschnitt = arithmetisches Mittel

- Der Mittelwert ist ein Lagemaß und wird berechnet aus der Summe der Werte, geteilt durch die Anzahl
- darf nur für metrische Merkmale berechnet werden

- $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i = \frac{x_1 + x_2 + \ ... + x_n}{n}$

- Der Mittelwert wird anhand konkreter Daten (Stichprobe) berechnet und ist ein Schätzer für den **Erwartungswert** einer theoretischen Verteilung

Python:

```
import statistics
statistics.mean(x)

import pandas as pd
df["Spalte"].mean()
```

Beispiel: Mittlere Körpergröße

Person	Körpergröße (cm)
Sandra	171
Vitali	188
Emre	175

$$\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i$$

$\frac{1}{n} = \frac{1}{3}$

$\bar{x} = \frac{1}{3} \cdot 534\,\text{cm} = 178\,\text{cm}$

171 cm + 188 cm + 175 cm = 534cm

Der Mittelwert ist nur eine Kennzahl über das "Zentrum". Sie verrät nichts darüber, wie die Daten verteilt sind

x
-3
-2
-1
0
1
2
3

y
-100
-100
100
100

z
-300
10
20
30
40
200

Alle drei Datensätze haben den Mittelwert 0

ACHTUNG:

Der Mittelwert ist nur sehr eingeschränkt interpretierbar, weil der Wert…

…keine Informationen zur **Streuung** der Daten beinhaltet.

…keine Informationen zum **Stichprobenumfang** beinhaltet.

…sehr anfällig ist für **Ausreißer**, die den Mittelwert verzerren können.

Mittelwert miteinander verrechnen

Um die Mittelwerte zweier Stichproben / Gruppen miteinander zu verrechnen, muss mit der jeweiligen Anzahl gewichtet werden

$$\overline{x + y} = \frac{n_x \bar{x} + n_y \bar{y}}{n_x + n_y}$$

Beispiel

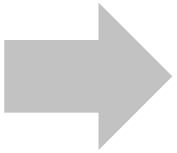
Gruppe 1 besteht aus 3 Personen, deren durchschnittlicher Warenkorb 20€ beträgt. Gruppe 2 besteht aus 7 Personen, deren durchschnittlicher Warenkorb 10€ beträgt.

Der gemeinsame Mittelwert ist dann ___€

DataCraft

Der Mittelwert ist empfindlich gegenüber Ausreißern

x
1
2
3
4
5
1000

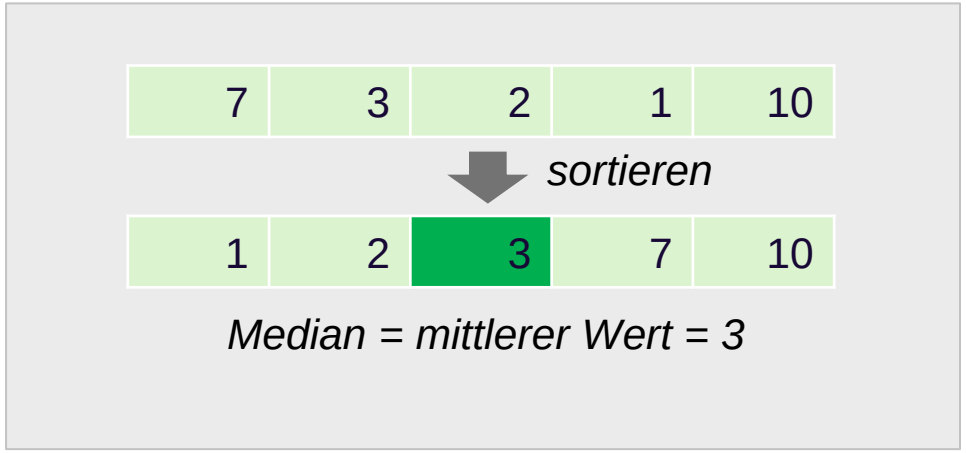


$$\bar{x} = \frac{1}{6}\sum_{i=1}^{6} x_i = \frac{1+2+3+4+5+1002}{6} = 169{,}5$$

DataCraft

Median = Zentralwert

- Der Median ist ein Lagemaß, welches genau "in der Mitte" steht, wenn man die Werte der Größe nach sortiert
- Der Median darf für **ordinale Merkmale** berechnet werden (natürlich auch für metrische Merkmale)



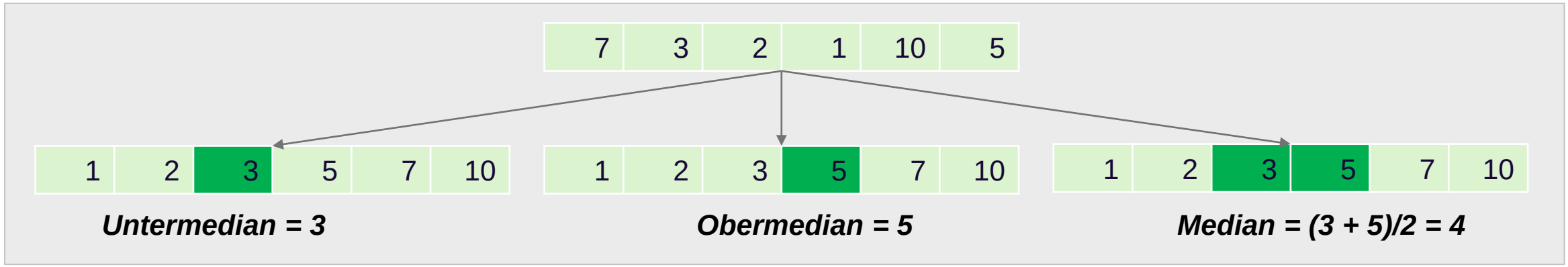
Python:

```
import statistics
statistics.median(x)

import pandas as pd
df["Spalte"].median()
```

Median

- Bei einer geraden Anzahl von Werten gibt es drei Möglichkeiten
  - Untermedian: der Wert links von der Mitte
  - Obermedian: der Wert rechts von der Mitte
  - Mittelwert der beiden Werte (sofern das erlaubt ist)



Python:

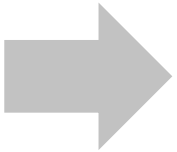
```
import statistics
statistics.median_low(x)
statistics.median_high(x)
```

```
import pandas as pd
df["Spalte"].quantile(interpolation="lower")
df["Spalte"].quantile(interpolation="upper")
```

DataCraft

Der Median ist robust gegenüber Ausreißern

x
1
2
3
4
5
1000



Mittelwert = 169,5

Median = 3,5

Quantile = Verallgemeinerung des Konzepts "Median":

Der Median ist der Wert, für den 50% der Daten kleiner und 50% der Daten größer sind

- 0,5-Quantil = Median
- 0,25-Quantil = 1. Quartil = 25% der Daten sind kleiner und 75% größer
- 0,75-Quantil = 3. Quartil = 75% der Daten sind kleiner und 25% größer
- p-Quantil (für p zwischen 0 und 1) = p% der Daten sind kleiner

Terzile: p = 1/3 bzw. 2/3
Quintile: p = 0,2 bzw. 0,4 bzw. 0,6 bzw. 0,8
Dezile: p ist ein Vielfaches von 0,1
Perzentile: p ist ein Vielfaches von 0,01 (also %)

Quantile in Python

DataCraft

Python:

```
import statistics
# Quartile (4 Unterteilungen)
statistics.quantiles(x, n=4)
# Dezile (10 Unterteilungen)
statistics.quantiles(x, n=10)
```

```
import pandas as pd
# Quartile
df["Spalte"].quantile([0.25, 0.5, 0.75])
# Dezile
df["Spalte"].quantile([0.1, 0.2, 0.3, 0.4,
                       0.5, 0.6, 0.7, 0.8, 0.9])
```

Pandas bietet außerdem die Funktion **describe()**, welche standardmäßig die Quartile ausgibt. Über den Parameter *percentiles* können beliebige Quantile ausgegeben werden

Ausreißerbereinigung

DataCraft

Eine einfache Möglichkeit, Ausreißer auszusortieren, funktioniert über Quantile

Behalte nur die Werte, die größer als das 0,01-Quantil und kleiner als das 0,99-Quantil sind.

Damit werden 2% der Daten aussortiert, die kleinsten und die größten 1%.

Eventuell müssen die Grenzen angepasst werden!

Python:

```
df[(df["Spalte"] > df["Spalte"].quantile(0.01)) & \
          [(df["Spalte"] < df["Spalte"].quantile(0.99))]
```

Modus

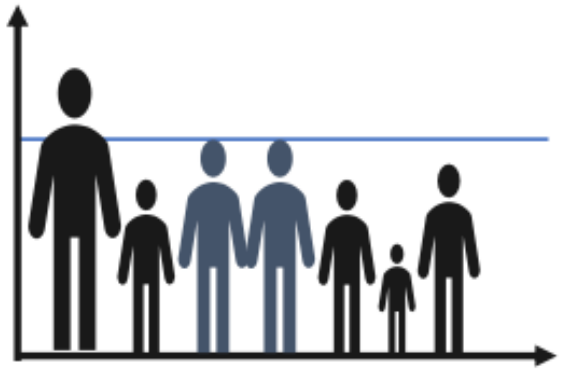
Für nominale Daten können weder Mittelwert noch Quantile berechnet werden

Modus = Modalwert = häufigster Wert einer Stichprobe

Der Modus kann zwar immer berechnet werden, ist aber selten aussagekräftig. Besser ist es, die Häufigkeiten aller Ausprägungen aufzulisten bzw. grafisch darzustellen.

Lagemaße

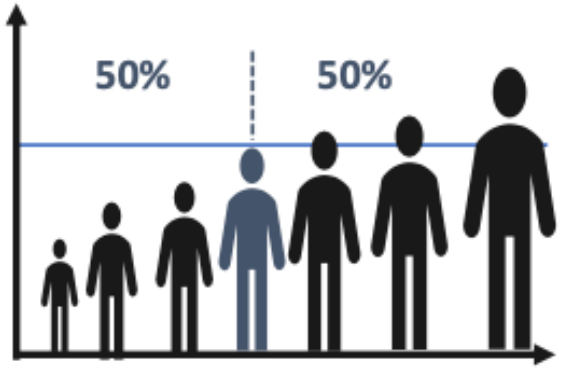
Modus



Häufigster Wert

Mindest-Voraussetzung: Nominalskala

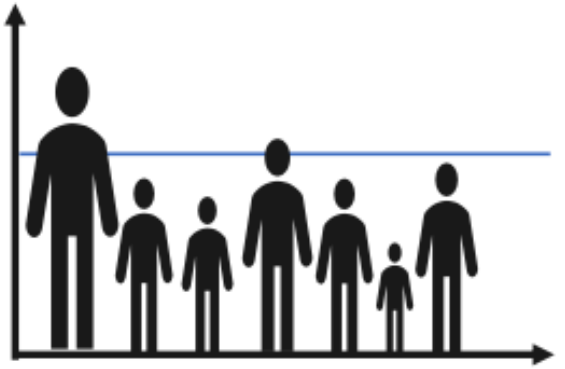
Median



Zentralwert. Über und unter dem Wert liegen jeweils 50% der Fälle

Mindest-Voraussetzung: Ordinalskala

Arithmetisches Mittel



Summe aller Werte dividiert durch die Anzahl aller Werte

Mindest-Voraussetzung: Intervall- oder Verhältnisskala

3
Streuungsmaße

Varianz – der mittlere Quadratische Abstand vom Mittelwert

DataCraft

- Stichprobenvarianz:

$$s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2$$

$$= \frac{1}{n-1}[(x_1 - \bar{x})^2 + (x_2 - \bar{x})^2 + \cdots + (x_n - \bar{x})^2]$$

- Quadrieren, damit positive und negative Abweichungen vom Mittelwert sich nicht aufheben.

Python:

```
import statistics

statistics.variance(x)

import pandas as pd

df["Spalte"].var()
```

Standardabweichung (& Varianz): Wie geht das statistisch?

Statistisches Zeichen für Varianz

Alles aufsummieren von $i = 1$ (erster Wert der Spalte) bis $i = n$ (letzter Wert der Spalte)

$$s^2 = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x})^2$$

Teile die Summe durch $n-1$, also durch die Gesamtzahl aller Merkmalsträger-1

Für jedes i den Abstand des Wertes x_i zum Mittelwert $\bar{x}$ berechnen und quadrieren

$$s = \sqrt{s^2}$$

Statistisches Zeichen für Standardabweichung

Standardabweichung

DataCraft

- Die (empirische) **Standardabweichung** gibt an, wie stark die Daten um den Mittelwert streuen
- Die Standardabweichung ist nie negativ. 0 bedeutet keine Streuung, d.h. konstanter Wert
- Die Varianz ist das Quadrat aus der Standardabweichung bzw. die Standardabweichung die Wurzel aus der Varianz
- Die Standardabweichung darf nur für metrische Daten berechnet werden
- die Standardabweichung reagiert empfindlich auf Ausreißer

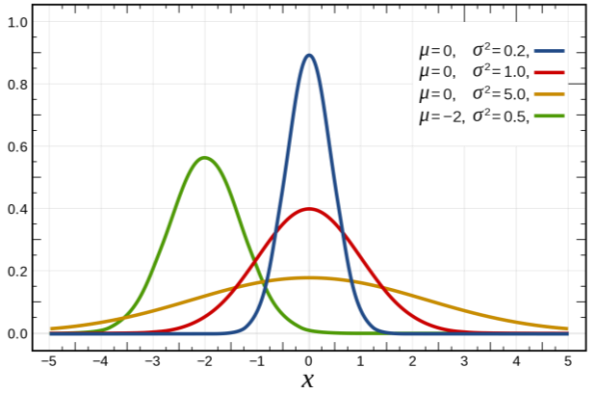
Python:

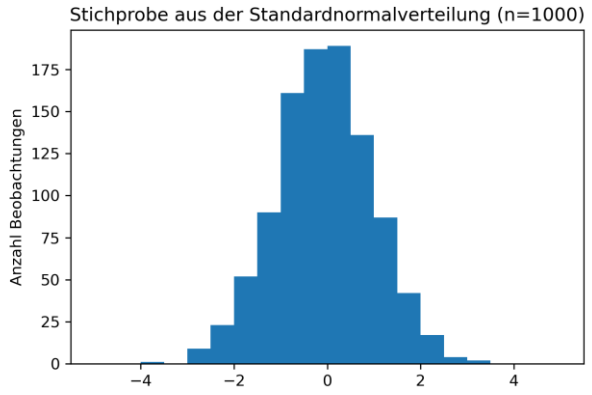
```
import statistics
statistics.stdev(x)

import pandas as pd
df["Spalte"].std()
```

Unterschied zwischen theoretischer Betrachtung und Stichprobe/Daten

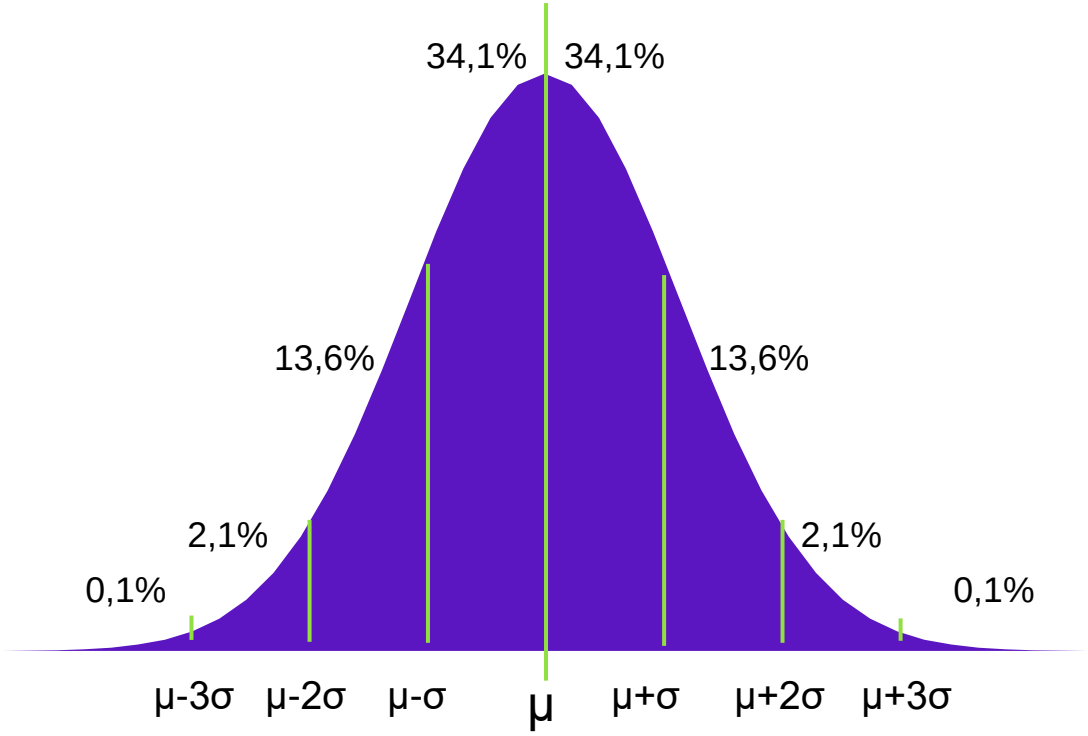
- μ (griechisch my, sprich mü) ist der Erwartungswert (theoretischer Mittelwert)
- $\bar{x}$ ist der empirische Mittelwert. $\bar{x}$ ist eine Schätzung von μ.
- σ (griechisch sigma) ist die (theoretische) Standardabweichung, σ² die Varianz
- s ist die empirische Standardabweichung, s² die empirische oder Stichprobenvarianz. s ist eine Schätzung von σ.



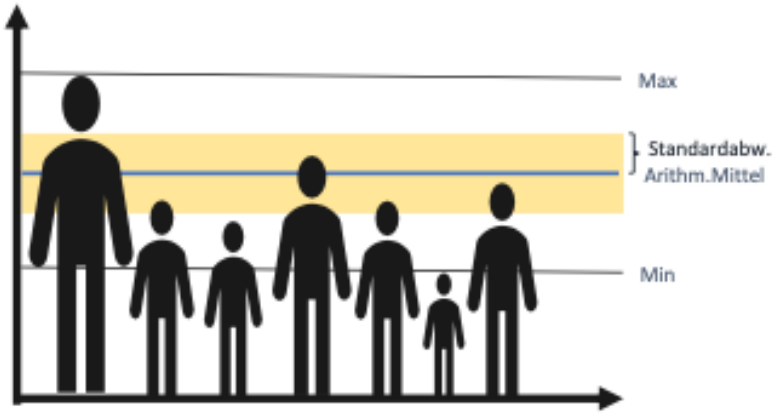


Bei der Normalverteilung liegen

- 68% der Daten liegen innerhalb eines Abstands einer Standardabweichung vom Mittelwert μ
- 95% der Daten liegen in einem 2*σ-Abstand vom Mittelpunkt
- 99% der Daten liegen in einem 3*σ-Abstand vom Mittelpunkt



Standardabweichung & Varianz

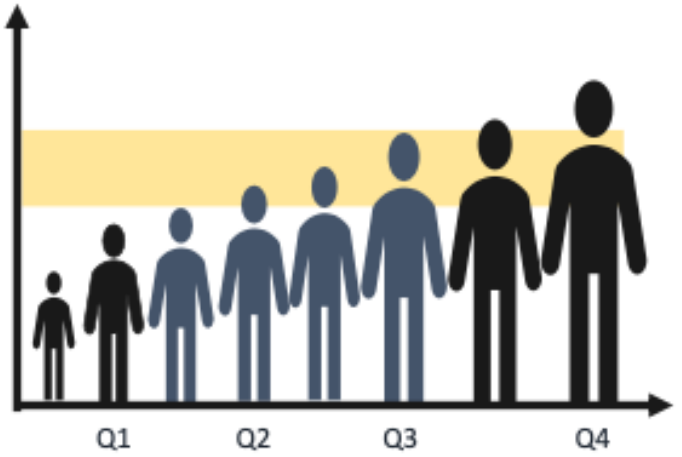


- **Varianz**: Durchschnittliche quadratische Entfernung aller Werte von Mittelwert
- **Standardabweichung**: Wurzel der Varianz

Spannweite

Differenz zwischen größtem und kleinsten Wert im Datensatz

Interquartilsabstand



Breite des Intervalls, in dem die mittleren 50 % der Stichprobenelemente liegen