

Statistik für Data Scientists

Vorlesung 1: Einführung & Daten

Prof. Dr. Siegfried Handschuh

Universität St. Gallen

Warum Statistik?

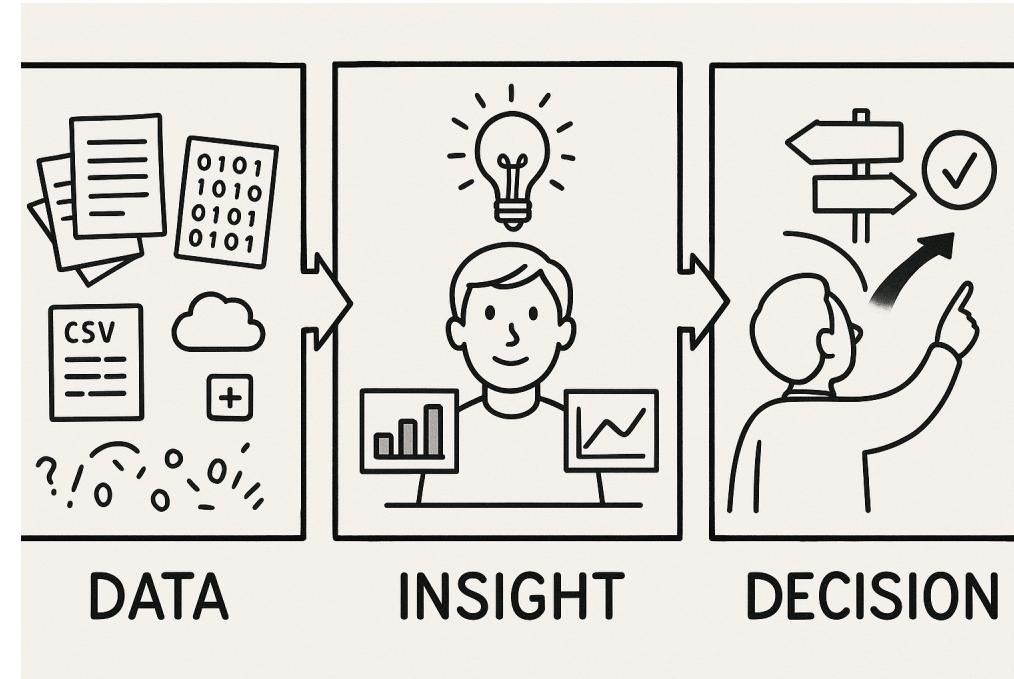
👉 Statistik = Werkzeugkasten für Data Science

Debugging von ML-Modellen

Evaluierung von A/B-Tests

Analyse von Logfiles

Datenqualität prüfen vor Modellierung



Fahrplan heute

1. Datentypen & Messniveaus
2. Bias in Daten
3. Missing Values
4. Erste EDA (Teaser)
5. Zusammenfassung & Ausblick

Datentypen



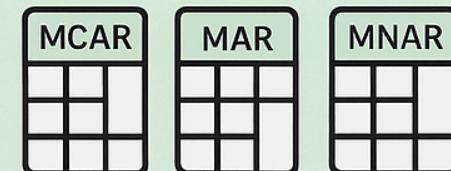
Statistik hängt von
Datentypen ab

Bias



Systematische
Verzerrung erkennen

Missing Values



MCAR, MAR, MNAR
unterscheiden

EDA

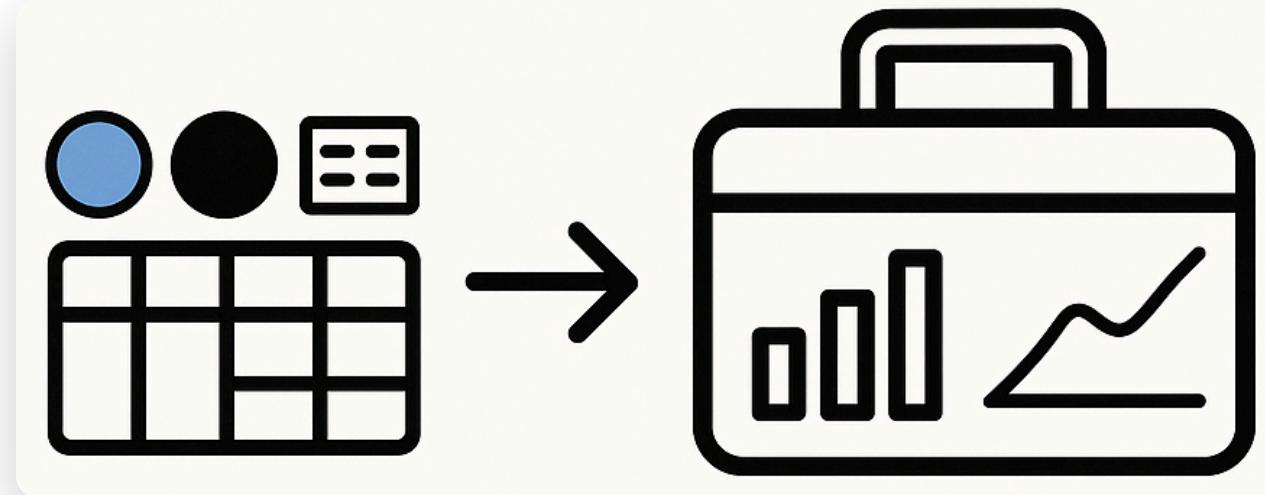


Daten anschauen
& einfache Plots

Block A: Datentypen & Messniveaus

Grundidee

- Statistik hängt vom **Datentyp** ab
- Unterschiedliche Methoden für verschiedene Skalen



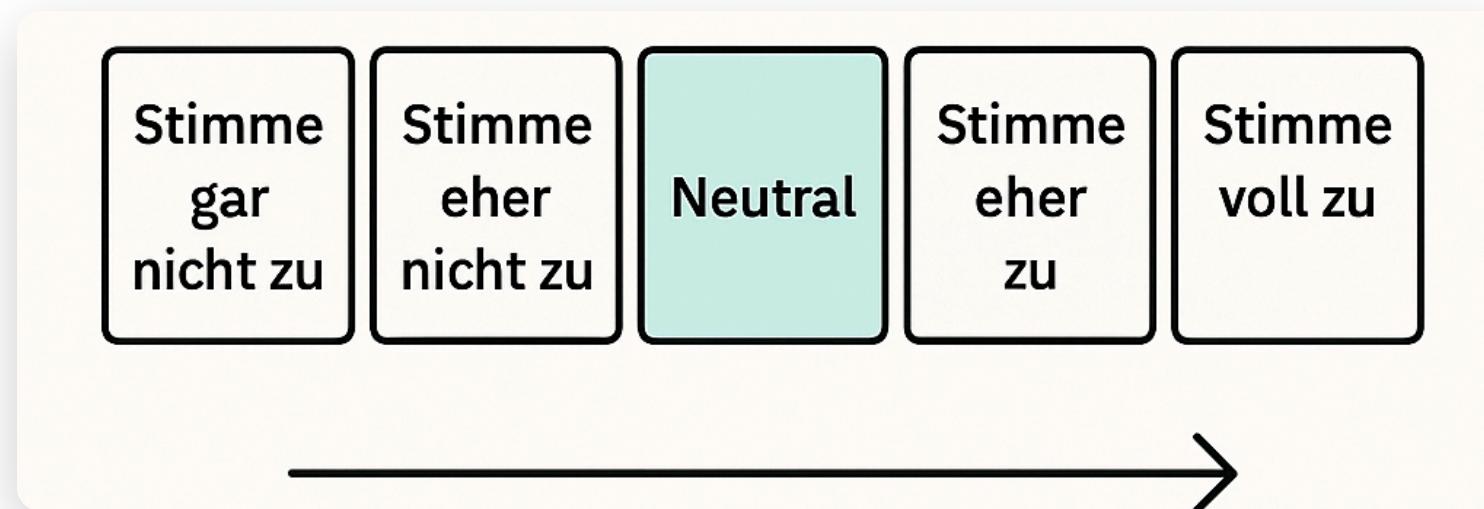
Nominalskala

- Kategorien ohne Ordnung
- Beispiele: HTTP Status Codes, Geschlecht, Kundennummern



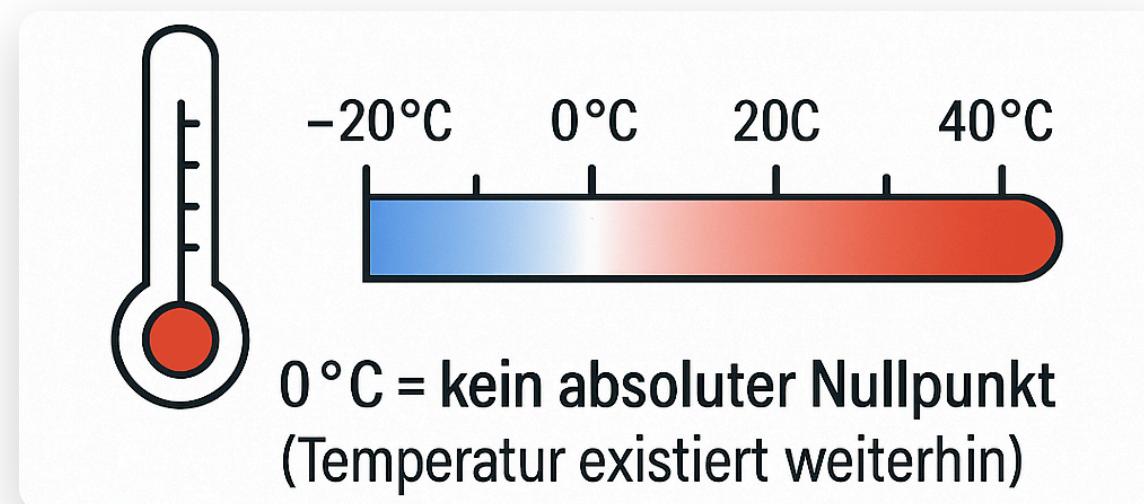
Ordinalskala

- Kategorien mit Rangfolge
- Beispiele: Schulnoten, Star Ratings (⭐1–⭐5)



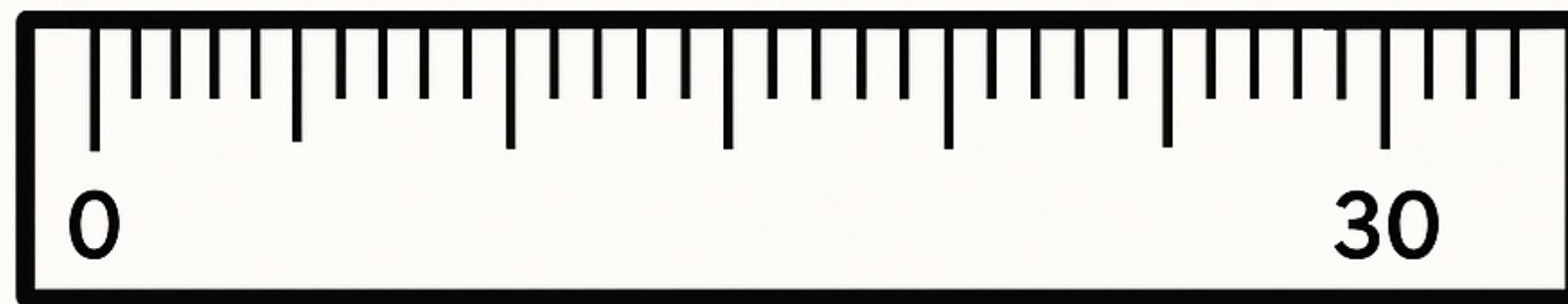
Intervallskala

- Abstände sinnvoll, kein natürlicher Nullpunkt
- Beispiele: Temperatur in °C, Timestamps



Ratioskala

- Abstände + Nullpunkt vorhanden
- Beispiele: Alter in Jahren, Dateigrösse in Bytes



Quiz: Welche Skala?

- PLZ = ?
- Alter in Jahren = ?
- RAM Grösse = ?
- Fehler Severity (Low/Med/High) = ?

Python-Snippet – Datentypen prüfen

```
import seaborn as sns  
titanic = sns.load_dataset("titanic")  
print(titanic.info())
```

👉 Pandas unterscheidet zwischen
`int64`, `float64`, `category`,
`object`, `bool`

```
<class 'pandas.DataFrame'>  
RangeIndex: 891 entries  
Data columns (15 columns):  
survived      891 non-null int64  
pclass        891 non-null int64  
sex           891 non-null object  
age            714 non-null float64  
sibsp          891 non-null int64  
fare           891 non-null float64  
embarked       889 non-null object  
class          891 non-null category  
adult_male     891 non-null bool  
deck           203 non-null category
```

Block B: Bias in Daten

Definition

Bias = systematische Verzerrung in Daten
oder Analyse

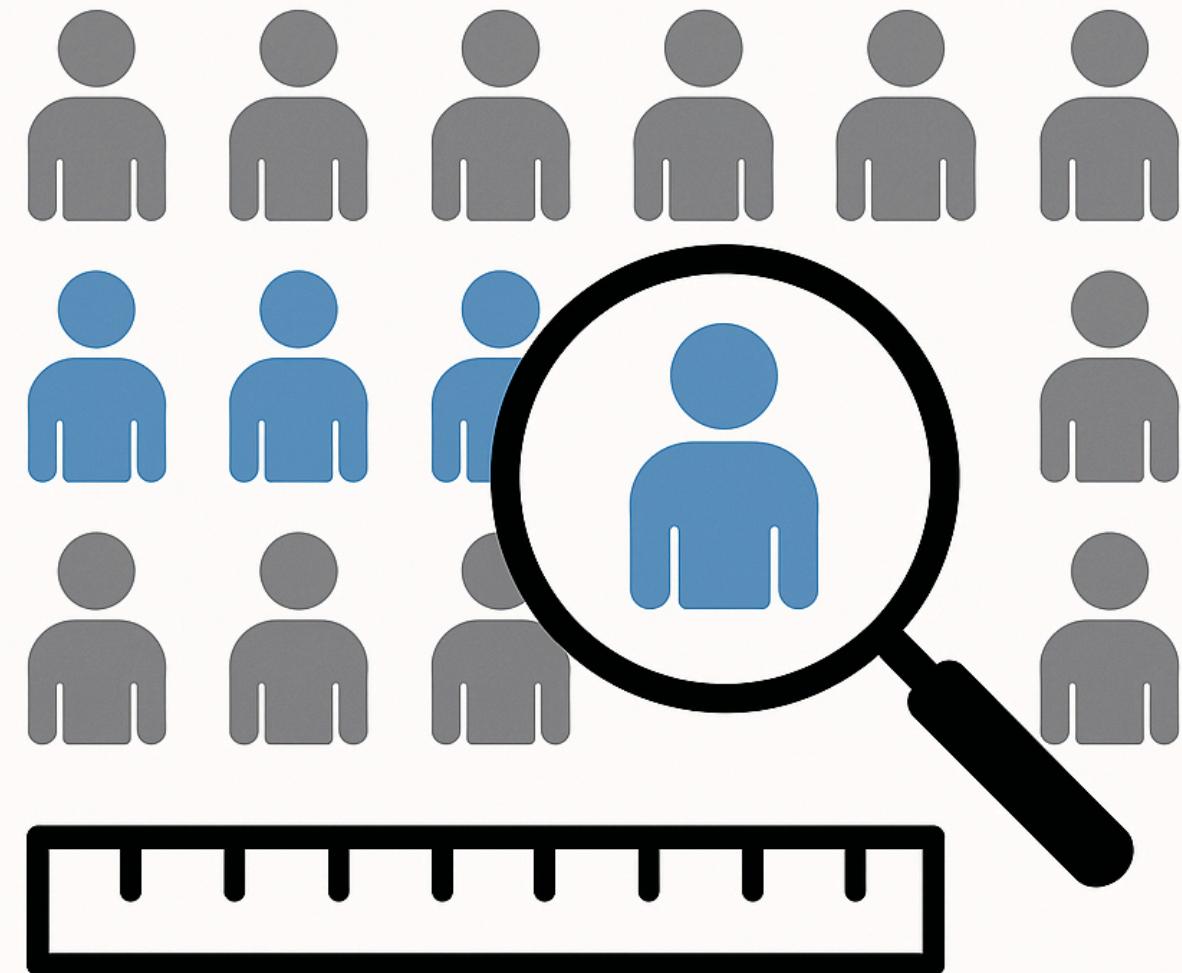


BIAS

Sampling Bias

Stichprobe nicht repräsentativ

Beispiel: Nur aktive Nutzer in einer App berücksichtigt (blau)



Survivorship Bias

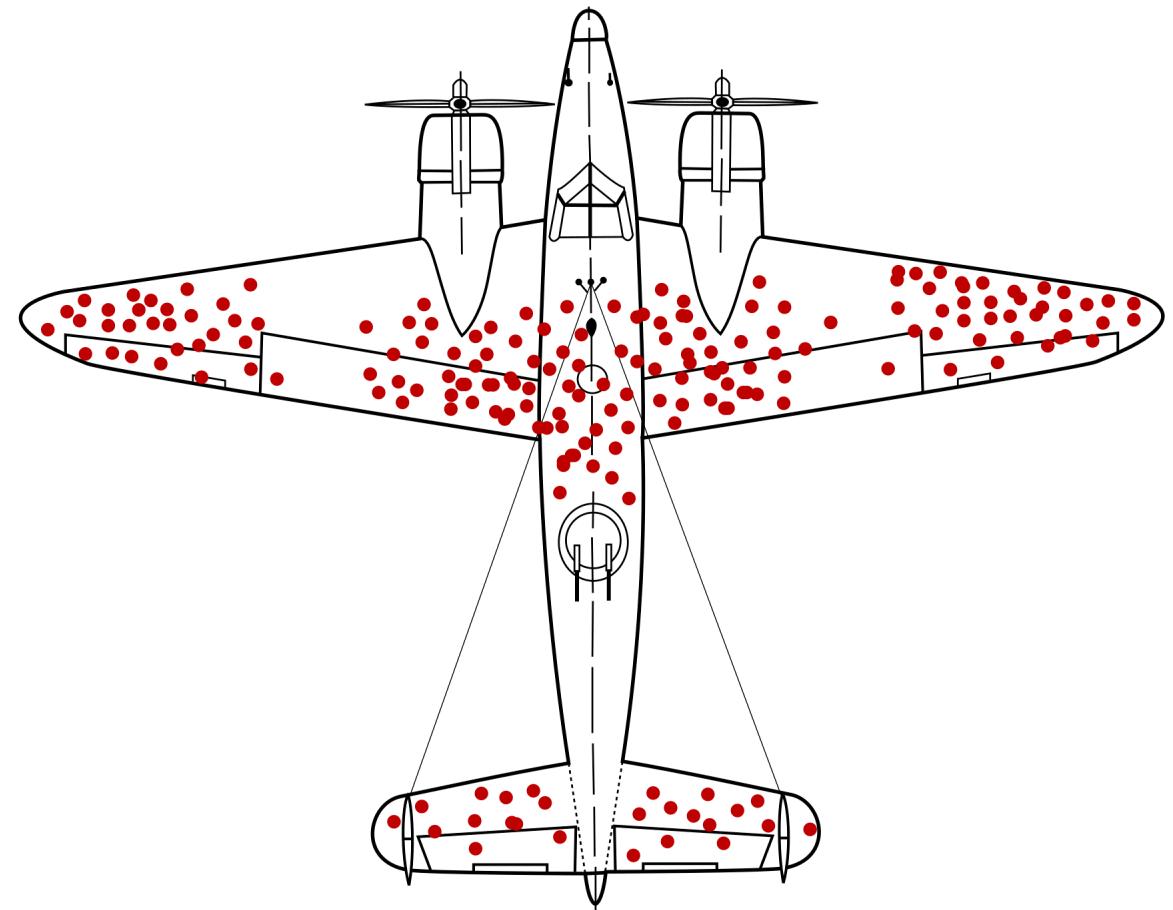
Nur „Überlebende“ betrachtet

Beispiel

Weltkrieg USA:

Falsch: Flügel und Rumpf verstärken!

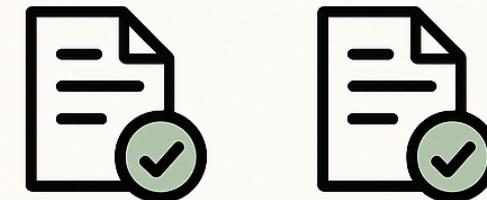
Richtig: Verstärkt die Motoren!



Confirmation Bias

Nur bestätigende Evidenz gesucht

Beispiel: Nur Benchmarks
berichten, die Hypothese stützen

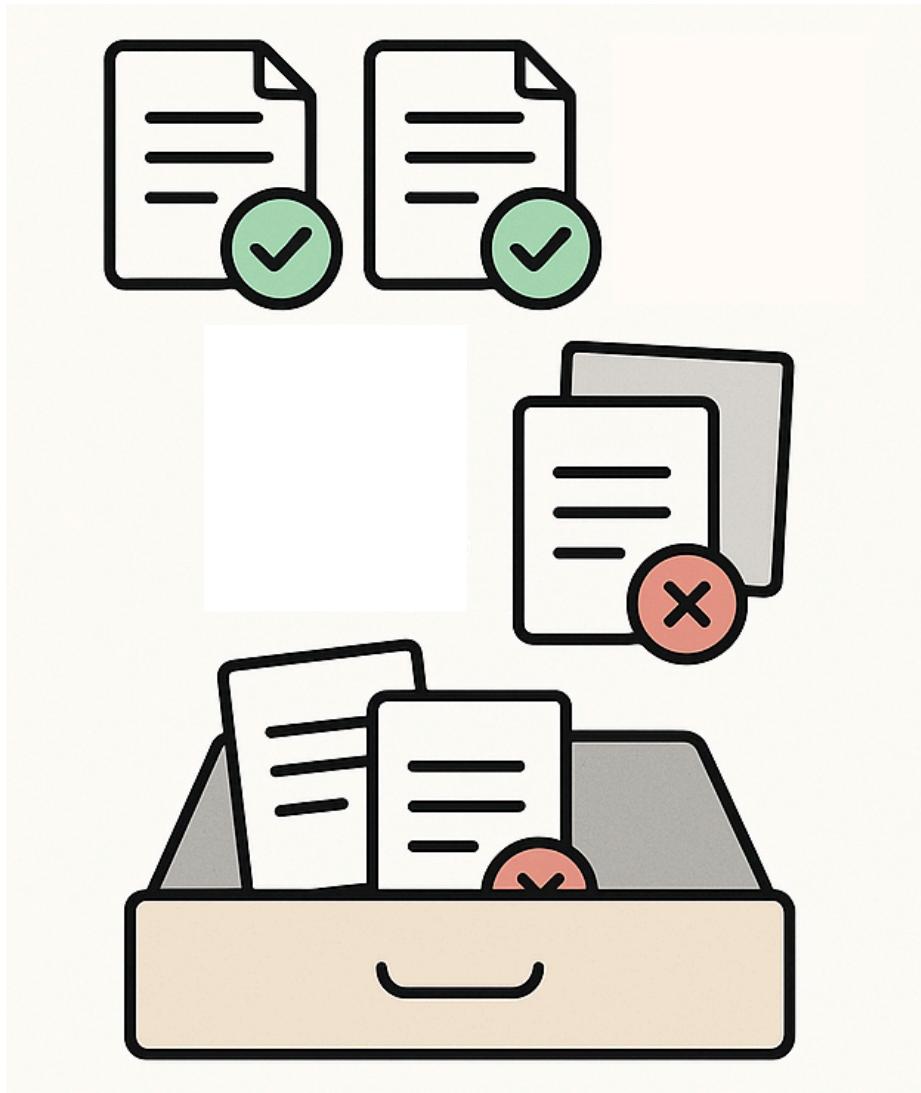


Confirmation Bias
Nur bestätigende Evidenz
gesucht

Publication Bias

Nur „positive“ Ergebnisse veröffentlicht

Besonders in wissenschaftlichen Studien
verbreitet



Measurement Bias

Messungen fehlerhaft oder verzerrt

Beispiel: Sensorfehler, falsche Logdaten



Diskussionsfrage



Wo könnte Bias in Informatikdaten auftreten?

- Benchmarking von Algorithmen
- Logging von Systemfehlern
- Trainingsdaten für ML

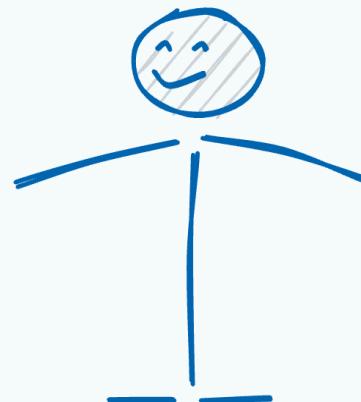
Block C: Missing Values

Missing Value



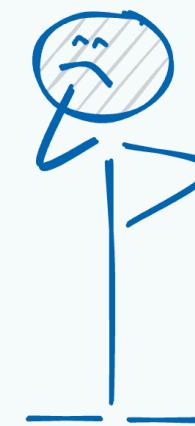
Missing values, let's
impute them quickly

	0	1	2	3	4
0	X				
1		X		X	
2	X				
3			X	X	
4	X			X	X



I must understand WHY
do I have missing values
before imputing them

	0	1	2	3	4
0	X				
1		X		X	
2		X			
3			X	X	
4	X			X	X



Arten von Missingness

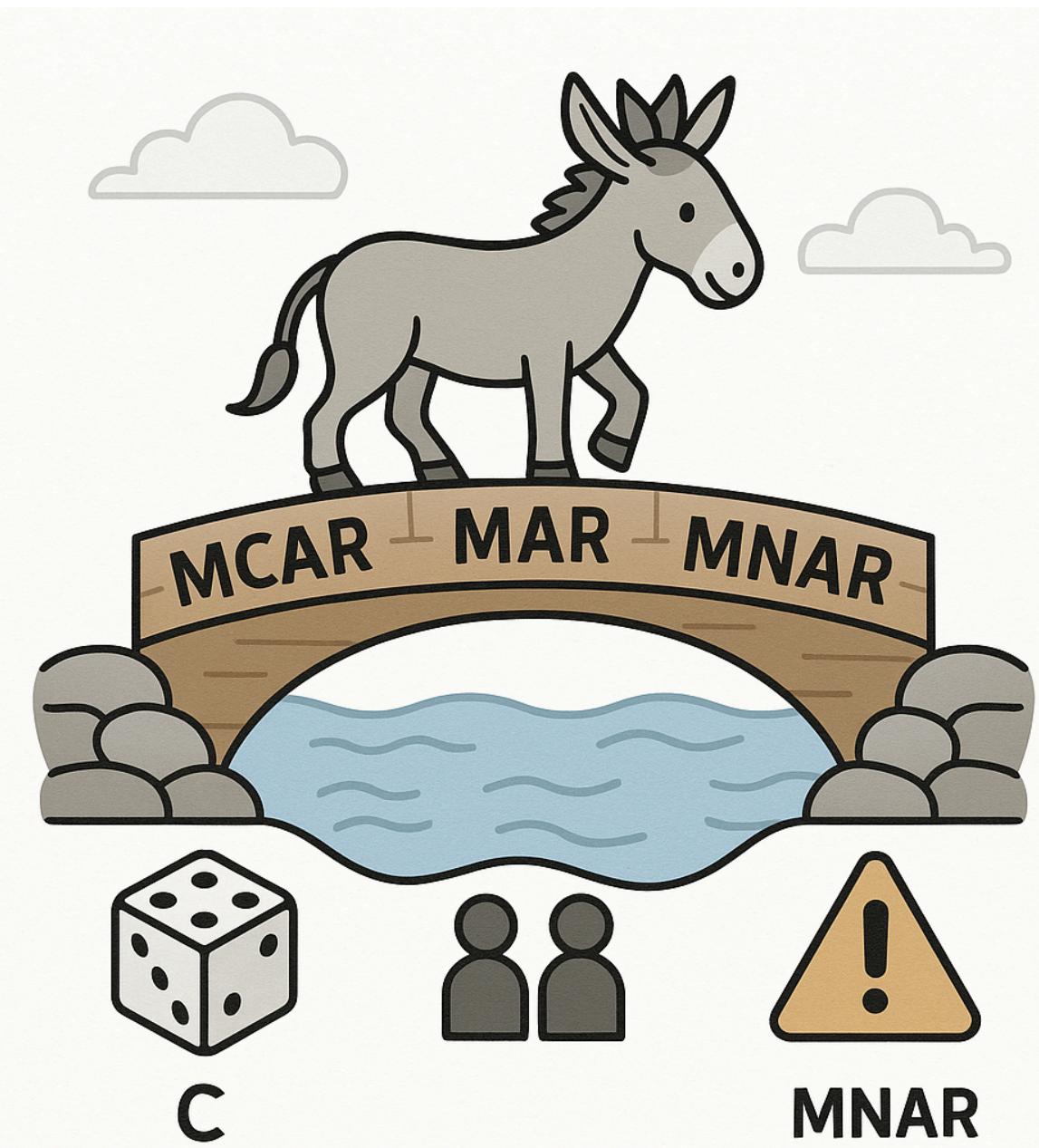
- MCAR: Missing Completely At Random
- MAR: Missing At Random: abhängig von beobachteten Variablen
- MNAR: Missing Not At Random: systematisch fehlend

Eselnbrücke

MCAR: Completely = Chaos,
reiner Zufall

MAR: At Random" = Abhängig
von Anderen (beobachteten)
Variablen

MNAR: Not Random = Nicht
zufällig, systematisches Problem



MCAR – Completely at Random

- Fehlende Werte rein zufällig
- Beispiel: Log-Server verliert zufällig Einträge
- Konsequenz: Keine Verzerrung → Einfachste Situation, die meisten Methoden funktionieren

MCAR
Completely At Random

X	X		X	X	X	X	X
X	X	X			X	X	X
X		X	X	X		X	X
X	X				X		X
X		X	X	X		X	X
X	X		X		X	X	X

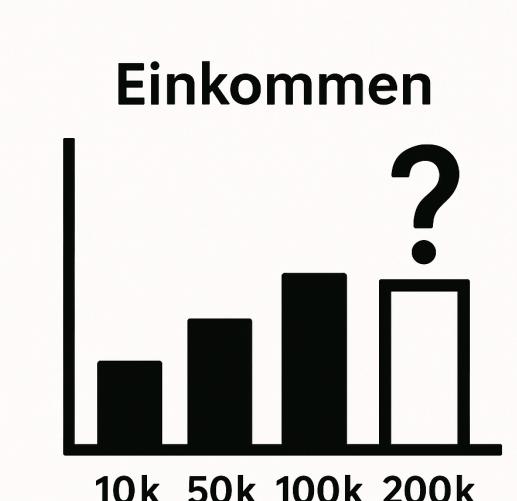
MAR – At Random

- Fehlende Werte abhängig von beobachteten Variablen
- Beispiel: Alter fehlt häufiger bei Frauen im Titanic-Datensatz (Es galt als unhöflich, Frauen nach ihrem Alter zu fragen)
- Konsequenz: Modellierbar → Fortgeschrittene Imputationsmethoden nötig

Geschlecht	Alter
Mann	32
Mann	45
Frau	?
Mann	28

MNAR – Not at Random

- Fehlende Werte abhängig vom wahren Wert selbst
- Beispiel: Einkommen fehlt häufiger bei sehr Reichen
- Konsequenz: Verzerrung, schwer zu korrigieren → Schwierigste Situation, spezielle Modellierung erforderlich



Lösungsansätze

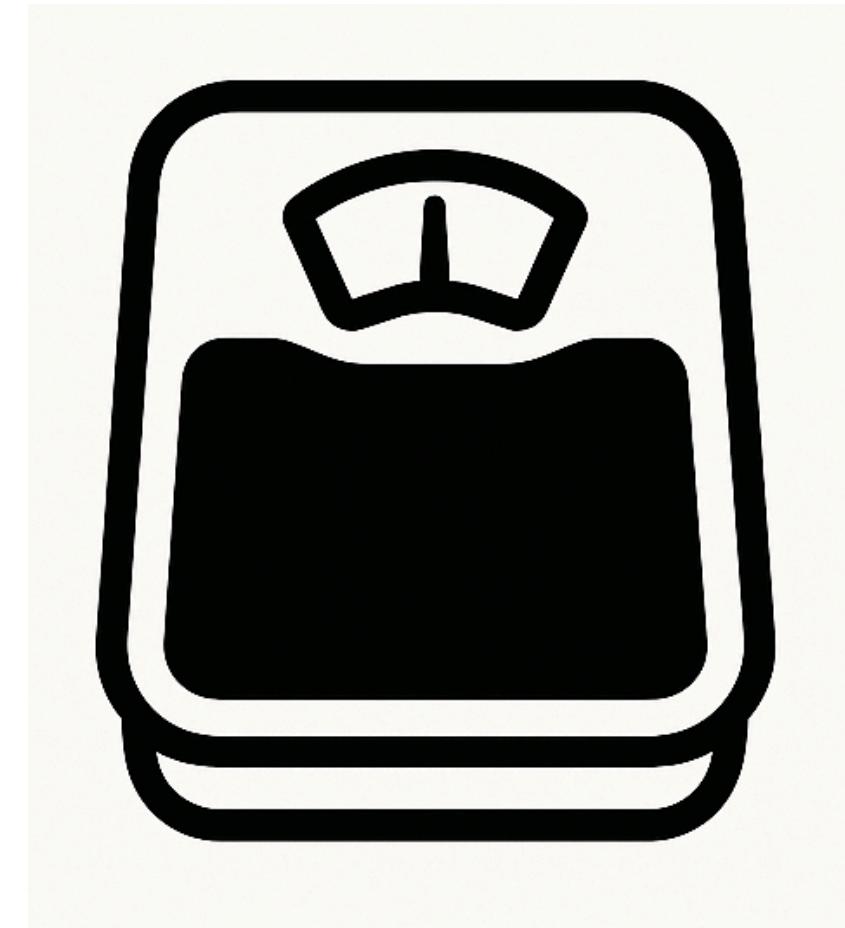
%	MCAR	MAR	MNAR
< 5%	Löschen OK	Gruppenweise Imputation	Domain Knowledge → Expertenwissen (bspw. Arzt)
5- 20%	Multiple Imputation	Multivariate Imputation by Chained Equations (MICE) / K-Nearest Neighbors Imputation (KNN)	Selection Models bspw. Heckman-Model
> 20%	Multiple Imputation	MICE + Sensitivity	Explizite Modellierung → Modelliere WARUM?

Interaktive Frage

👉 In einer Fitness-App fehlen viele Gewichtseinträge.

Ist das MCAR, MAR oder MNAR?

Warum?



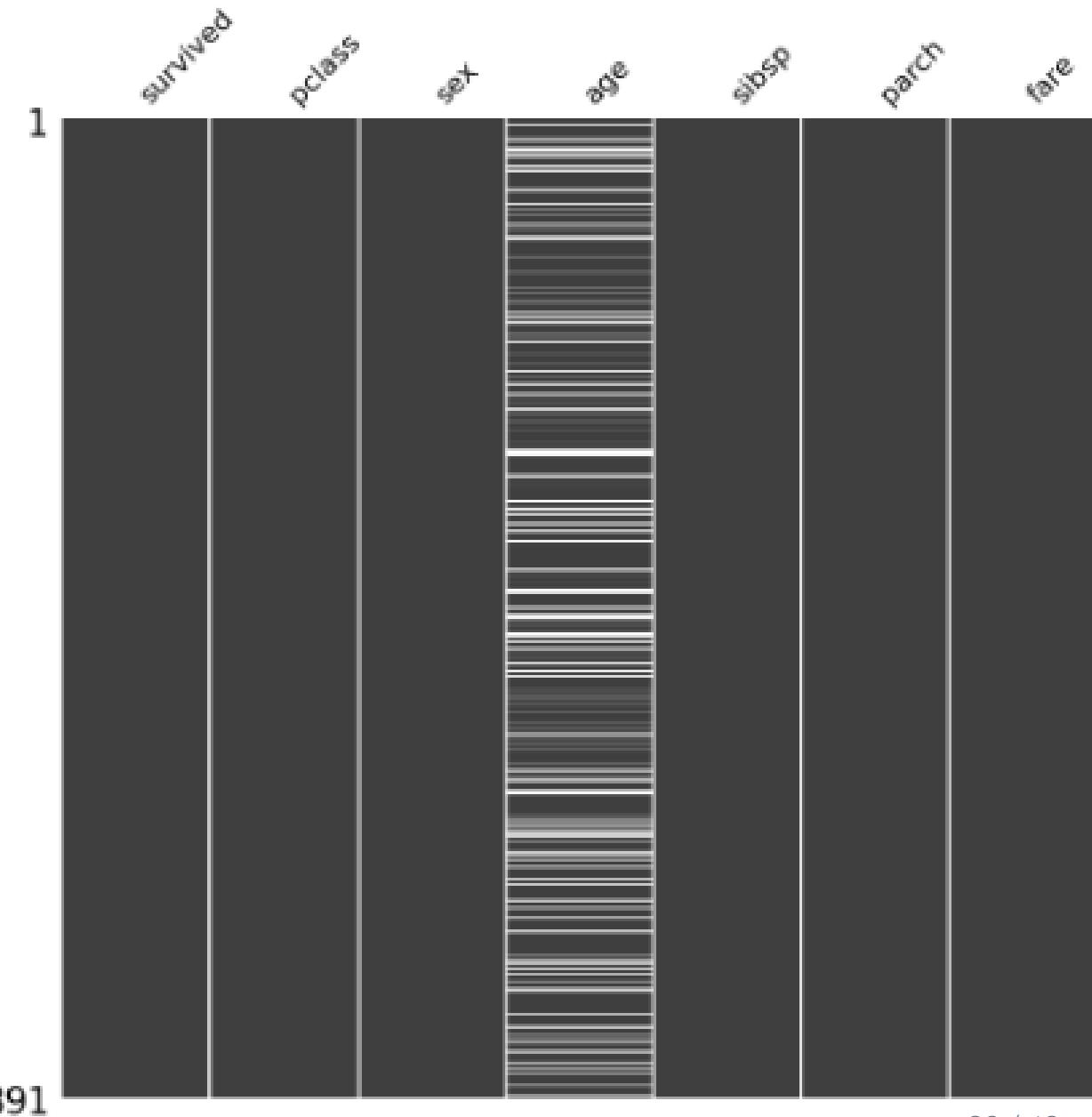
Python-Snippet – Missing Values zählen

```
titanic.isna().sum().head()
```

```
survived      0
pclass        0
sex           0
age          177
sibsp         0
dtype: int64
```

Missing Data Matrix

```
import missingno as msno  
msno.matrix(titanic)
```



Strategien

- Zeilen löschen (nur wenn wenige fehlen)
- Median/Mean-Imputation
- Gruppenweise Imputation (z.B. Median je Geschlecht)
- Modellbasierte Imputation (Regression, ML)

Median / Mean Imputation

Aspekt	Beschreibung
Methode	Ersetze fehlende Werte durch Median oder Mittelwert aller vorhandenen Werte
Vorteile	Einfach und schnell; keine komplexe Modellierung nötig; erhält zentrale Tendenz der Daten
Nachteile	Unterschätzt die Varianz (Daten wirken „zu gleichmäßig“); kann Zusammenhänge zwischen Variablen verzerren; ignoriert Unsicherheit der Imputation
Geeignet für	Kleine Anteile fehlender Werte (< 5%); unkritische Variablen; MCAR-Szenarien
Beispiel	<pre>df['age'].fillna(df['age'].median())</pre>



Beispiel: Median-Imputation

Szenario: E-Commerce-Analyse mit 10.000 Kunden

Problem: 300 Einkommensangaben fehlen (3%)

Lösung: Ersetze fehlende Werte mit Median (48.000€)

Kennzahl	Wert	Bedeutung
Fehlende Daten	3% (300 von 10.000)	Unter 5%-Schwelle ✓
Median Einkommen	48.000€	Robuster als Mittelwert
Missing-Typ	MCAR	Zufällig fehlend ✓
Variable	Nebenvariable	Nicht Hauptziel der Analyse ✓



Achtung: 300 identische Werte reduzieren die Varianz künstlich!

Wann Median-Imputation verwenden?

	
< 5% Daten fehlen	> 5% Daten fehlen
Nebenvariable	Hauptvariable der Analyse
MCAR (zufällig)	MNAR (systematisch)
Schnelle Lösung nötig	Präzise Varianz wichtig

Gruppenweise Imputation

Idee: Fehlende Werte innerhalb von Untergruppen ersetzt (z.B. Median je Geschlecht)

Aspekt	Beschreibung
Methode	Berechne Median/Mean für jede Gruppe separat und imputiere gruppenspezifisch
Vorteile	Berücksichtigt Unterschiede zwischen Gruppen; Weniger Verzerrung als globale Imputation; Einfach zu implementieren und interpretieren
Nachteile	Varianz-Unterschätzung innerhalb der Gruppen; Benötigt vollständige Gruppenvariable; Mindestgröße pro Gruppe erforderlich
Geeignet für	<ul style="list-style-type: none">• MAR-Szenarien• Kategorien wie Geschlecht, Altersgruppe, Region• 5-15% fehlende Werte



Praktisches Beispiel

Gruppe	N	Median Einkommen	Fehlende Werte	Imputierter Wert
Männer	5.200	52.000€	180	→ 52.000€
Frauen	4.800	44.000€	120	→ 44.000€
Gesamt	10.000	48.000€	300 (3%)	Gruppenspezifisch

Vergleich: Globale Imputation würde alle 300 Werte mit 48.000€ ersetzen

Vorteil hier: Geschlechtsspezifische Unterschiede bleiben erhalten

Modellbasierte Imputation

Idee: Fehlende Werte werden vorhergesagt (Regression, Random Forest, ML)

Aspekt	Beschreibung
Methode	Trainiere Modell auf vollständigen Daten → Sage fehlende Werte vorher
Modelle	<ul style="list-style-type: none">• Lineare/Logistische Regression; Random Forest / Gradient Boosting; Neuronale Netze; MICE (iterativ)
Vorteile	Nutzt komplexe Beziehungen zwischen Variablen; Erhält Varianz besser als einfache Methoden Kann nicht-lineare Muster erfassen
Nachteile	Rechenaufwendig und komplex; Gefahr von Overfitting; Benötigt ausreichend Trainingsdaten; Modellqualität entscheidend



Praktisches Beispiel: Einkommen vorhersagen

Prädiktor-Variablen	Ziel	Modell	Ergebnis
<ul style="list-style-type: none">• Alter: 35 Jahre• Bildung: Master• Berufserfahrung: 10 Jahre• Branche: IT	Einkommen (fehlt)	Random Forest trainiert auf 8.000 Fällen	Vorhergesagt: 68.500€ (mit CI: 62-75k€)

Python: `IterativeImputer(estimator=RandomForestRegressor())`

Vergleichstabelle Imputationsmethoden

Methode	Komplexität	Varianz	Beziehungen	Wann nutzen?
Mean/Median	★	✗ Unterschätzt	✗ Ignoriert	MCAR, < 5% fehlend
Gruppenweise	★★	⚠️ Teilweise	✓ Innerhalb Gruppen	MAR mit klaren Gruppen
KNN	★★★	✓ Erhält	✓ Lokal	Lokale Cluster
MICE	★★★★	✓ Erhält	✓ Global	MAR, viele Variablen
Deep Learning	★★★★★	✓ Erhält	✓ Komplex	Grosse Datensätze

Block D: Erste EDA (Teaser)

EDA – Grundidee

„Getting to know your data“

- Muster erkennen
- Fehler finden
- Hypothesen generieren

.head()

```
print(titanic.head())
```

```
survived  pclass      sex    age   sibsp   parch      fare embarked class \
0         0        3  male  22.0       1       0    7.2500      S  Third
1         1        1 female  38.0       1       0   71.2833      C  First
2         1        3 female  26.0       0       0    7.9250      S  Third

who  adult_male  deck  embark_town  alive  alone
0   man        True    NaN  Southampton    no  False
1 woman       False     C  Cherbourg    yes  False
2 woman       False    NaN  Southampton    yes  True
```

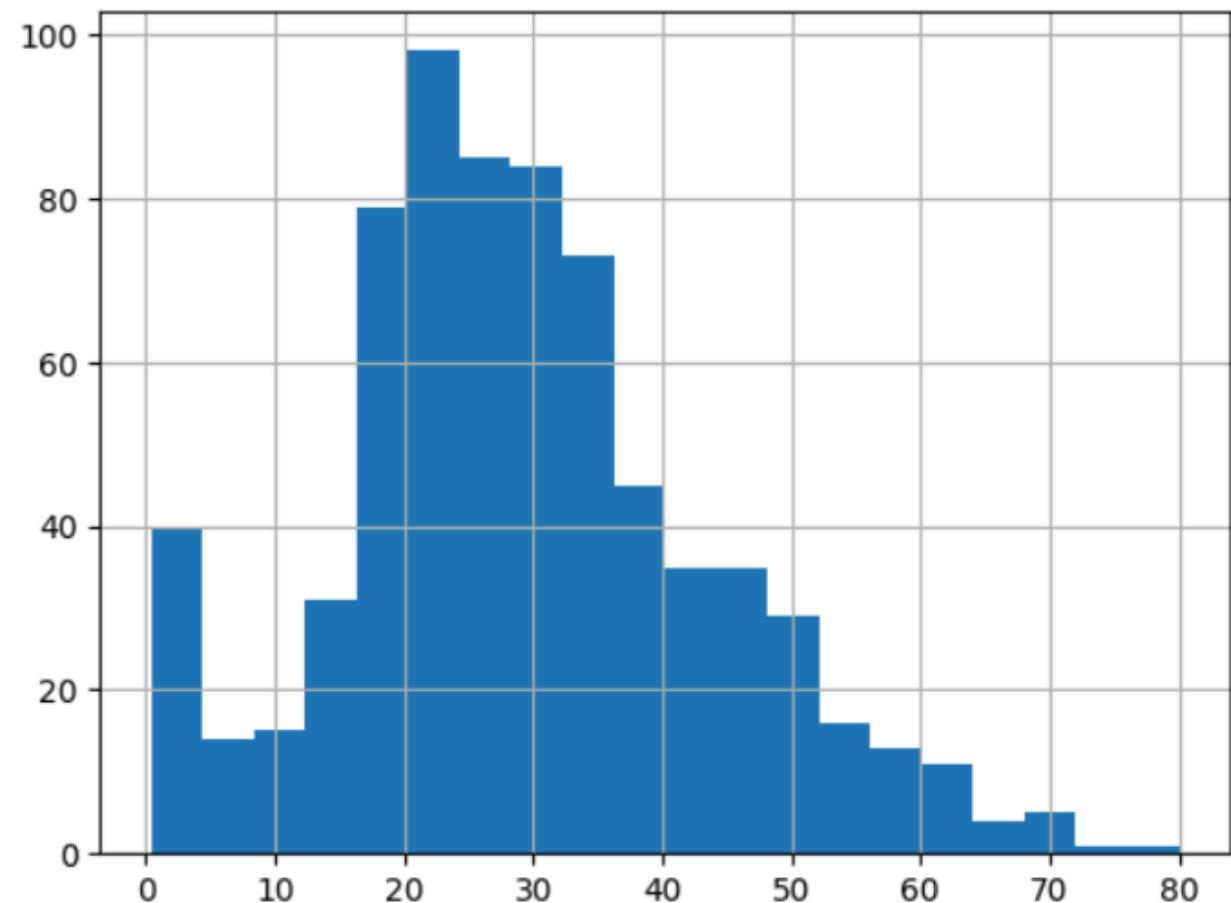
.describe()

```
print(titanic.describe())
```

	survived	pclass	age	sibsp	parch	fare
count	891.000000	891.000000	714.000000	891.000000	891.000000	891.000000
mean	0.383838	2.308642	29.699118	0.523008	0.381594	32.204208
std	0.486592	0.836071	14.526497	1.102743	0.806057	49.693429
min	0.000000	1.000000	0.420000	0.000000	0.000000	0.000000
25%	0.000000	2.000000	20.125000	0.000000	0.000000	7.910400
50%	0.000000	3.000000	28.000000	0.000000	0.000000	14.454200
75%	1.000000	3.000000	38.000000	1.000000	0.000000	31.000000
max	1.000000	3.000000	80.000000	8.000000	6.000000	512.329200

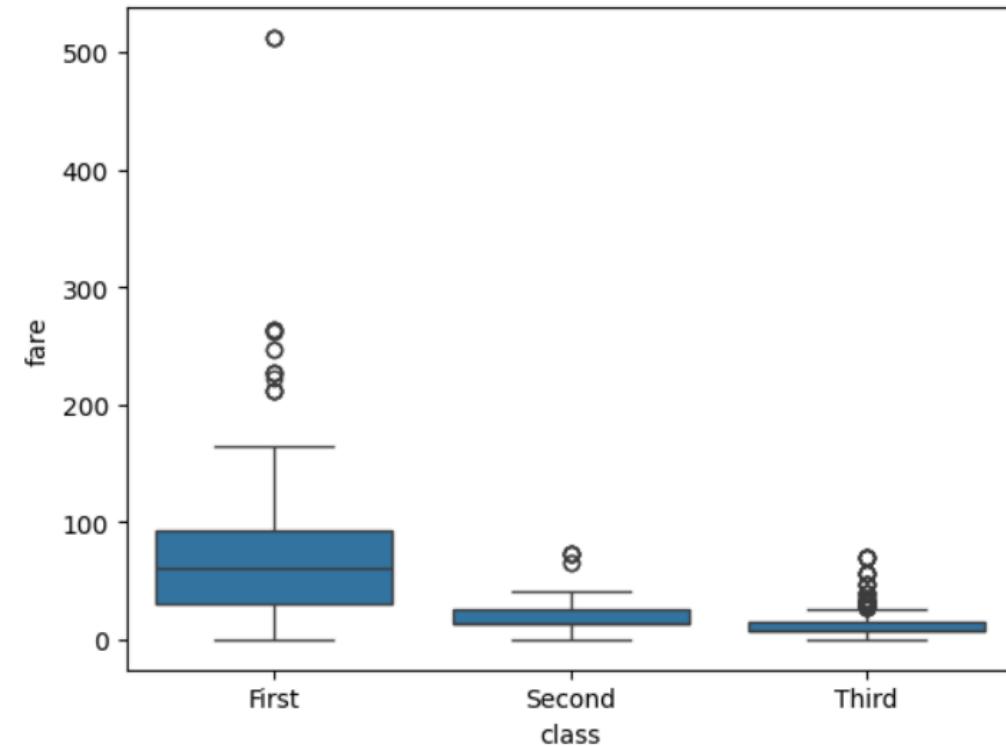
Histogramm (Teaser)

```
import matplotlib.pyplot as plt
titanic["age"].hist(bins=20)
plt.show()
```



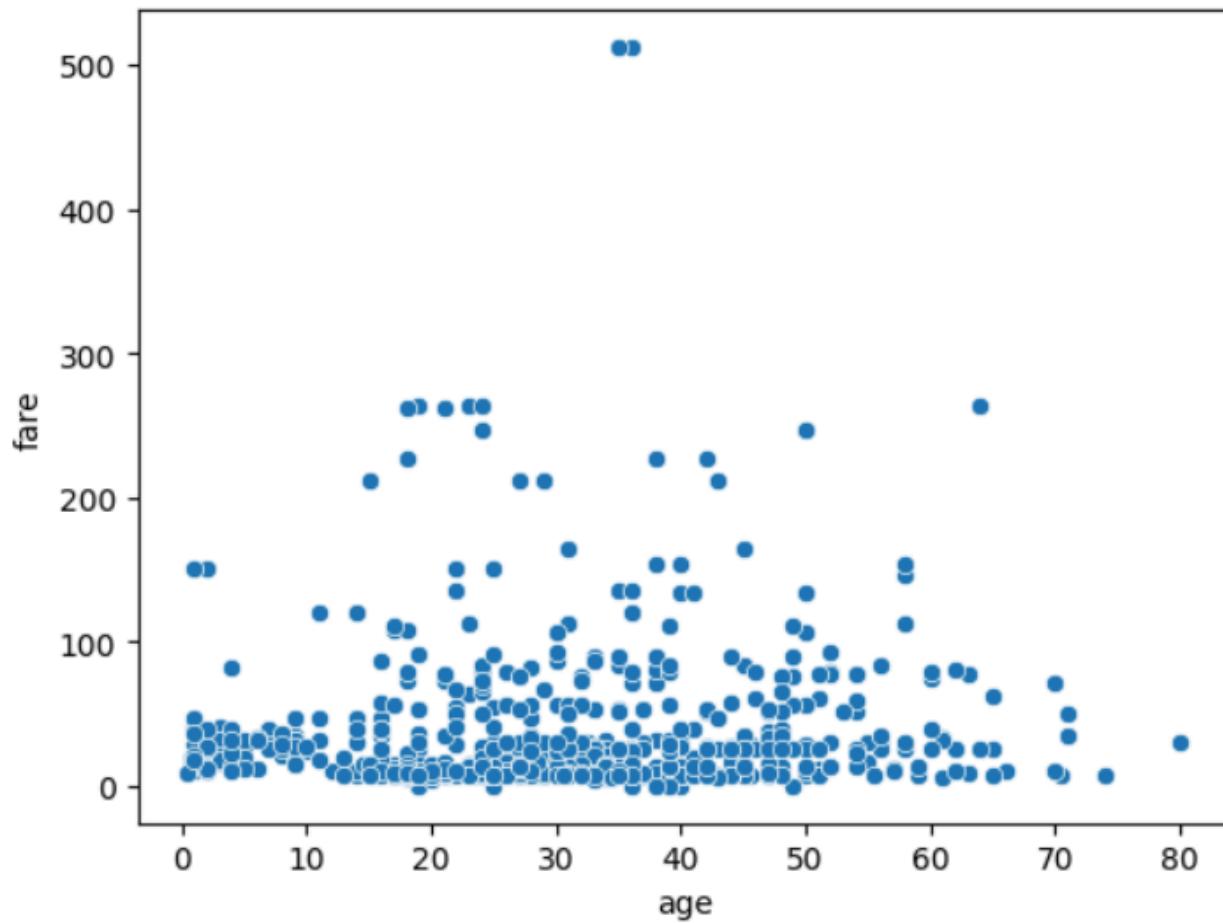
Boxplot (Teaser)

```
import seaborn as sns  
sns.boxplot(x="class", y="fare", data=titanic)
```



Scatterplot (Teaser)

```
{ sns.scatterplot(x="age", y="fare", data=titanic)
```



Block E: Zusammenfassung & Ausblick

Key Takeaways

Statistik hängt von Datentypen ab

Bias = systematische Verzerrung
erkennen

Missing Values unterscheiden
(MCAR, MAR, MNAR)

Erste EDA = Daten anschauen &
einfache Plots

Key Takeaways

Datentypen



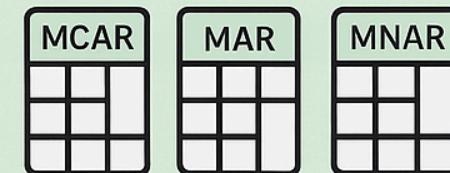
Statistik hängt von
Datentypen ab

Bias



Systematische
Verzerrung erkennen

Missing Values



MCAR, MAR, MNAR
unterscheiden

EDA



Daten anschauen
& einfache Plots

Ausblick

- Nächste Woche: Deskriptive Statistik & Visualisierung
- Lage- und Streuungsmasse
- Histogramme, Boxplots, Scatterplots im Detail



Vorbereitung:

- Practical Statistics for Data Scientists (*PSDS*) Kapitel 1
- Statistics for Data Scientists (*SDS*) Kapitel 1