

# Introduction to Data Science

## FS 2023

Prof. Dr. rer. nat. habil. Ralf-Peter Mundani  
DAViS

## Something about me... (keine latente Information 😊)

- Ralf-Peter Mundani
  - Studium Informatik (TUM)
  - Promotion Informatik / Computational Engineering (U Stuttgart)
  - Habilitation HPC (TUM)
  - Gastprofessor KAUST, KSA (2011)
  - Adjunct Teaching Professor (PD) TUM (seit 2014)
  - Dozent FHGR (seit 2019)
- Interessen
  - Numerische Simulation
  - Hoch- und Höchstleistungsrechnen
  - Data Science

## Kursinhalte / Umsetzung

- Inhalte
  - Vorhersagemodellen
  - Klassifikationsverfahren
  - überwachtes Lernen / Kreuzvalidierung
  - lineare Algebra / Singulärwertzerlegung / PCA
  - Tensorfaktorisierung
  - unüberwachtes Lernen / Beispiele Tensorfaktorisierung
- Umsetzung
  - Vorlesung + integrierte Übung
  - Nachbereitung (freiwillig)
  - Lernertragskontrolle (Prüfungswoche, schriftlich, 60 Minuten)



Quelle: der-querschnitt.de

## Kursinhalte / Umsetzung

- Voraussetzungen / Erwartungen
  - Umgang mit Windows / Linux / MacOS / you\_name\_it
  - gute Kenntnisse in Python
  - ~~exzellente~~ Kenntnisse in Mathematik 😊
  - Mitarbeit
  - Interesse & Spass am Ausprobieren

***"Let's get ready to rumble."***

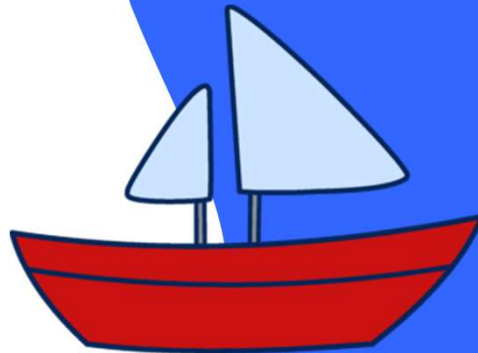


© Michael Buffer

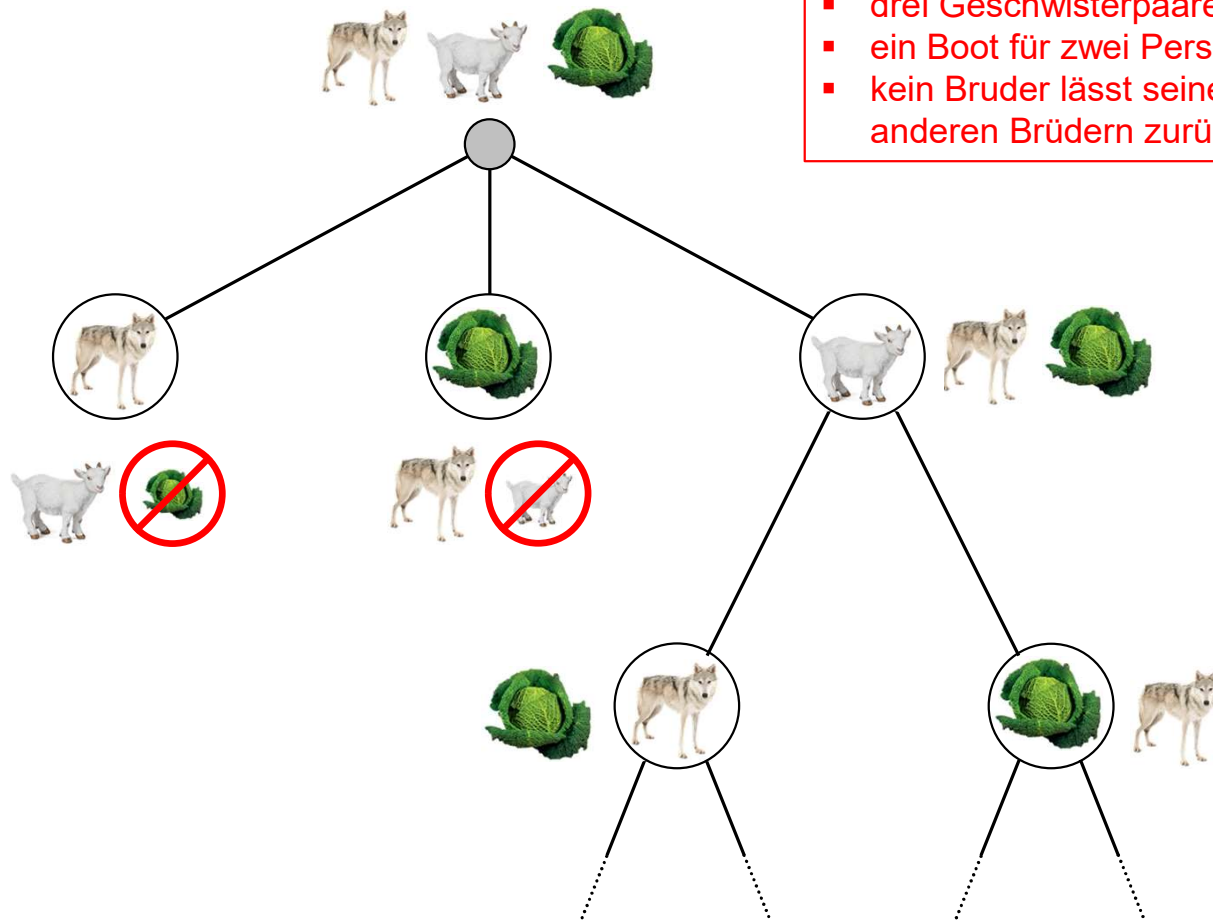
Motivation...?

**Motivation...!**

## Flussfahrt mit Huhn Ziege



## Flussfahrt mit Huhn Ziege



Alternativ (schwieriger):

- drei Geschwisterpaare (Bruder/Schwester)
- ein Boot für zwei Personen
- kein Bruder lässt seine Schwester allein mit anderen Brüdern zurück

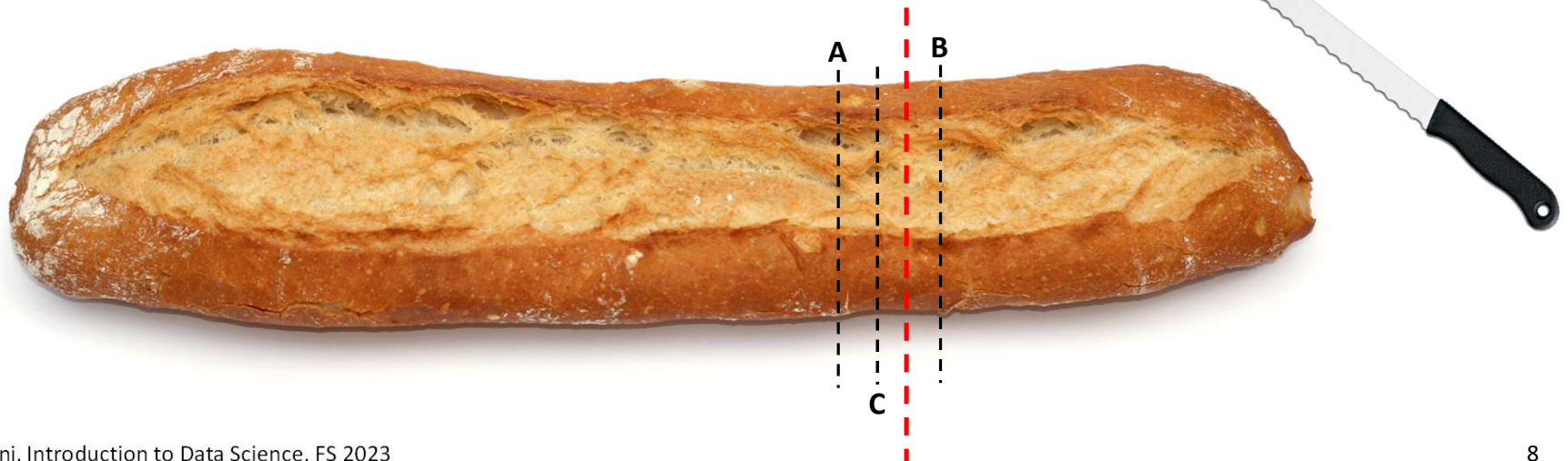
## Gefangenendilemma (nicht Spieltheorie 😊): *gerechte Brotteilung*



A

B

C










Ab sofort nicht im Handel erhältlich...

# Data Science: mehr als nur ein Hype...?



data science

X




Alle Bilder Bücher News Videos Mehr Suchfilter

Ungefähr 5'140'000'000 Ergebnisse (0,49 Sekunden)

5'140'000'000 Ergebnisse



**Gesponsert**


 fhgr.ch  
[https://www.fhgr.ch/data\\_science/studieren](https://www.fhgr.ch/data_science/studieren)

**Data Science - Bachelorstudium Data Science**

Die FH Grabbünden bildet Sie zu Fach- und Führungskräften im Bereich **Data Science** aus. Besuchen Sie unsere Infoanlässe online oder vor Ort und melden Sie sich zum Studium an.

**Data Science** ist ein interdisziplinäres Wissenschaftsfeld, welches wissenschaftlich fundierte Methoden, Prozesse, Algorithmen und Systeme zur Extraktion von Erkenntnissen, Mustern und Schlüssen sowohl aus strukturierten als auch unstrukturierten Daten ermöglicht.



 Wikipedia  
[https://de.wikipedia.org/wiki/Data\\_Science](https://de.wikipedia.org/wiki/Data_Science)

**Data Science - Wikipedia**

Informationen zu hervorgehobenen Snippets • Feedback geben

Ähnliche Fragen :

Wie viel verdient man als Data Scientist? 😊

Was macht man als Data Scientist?

Ist Data Science die Zukunft?

Ist Data Science gefragt?

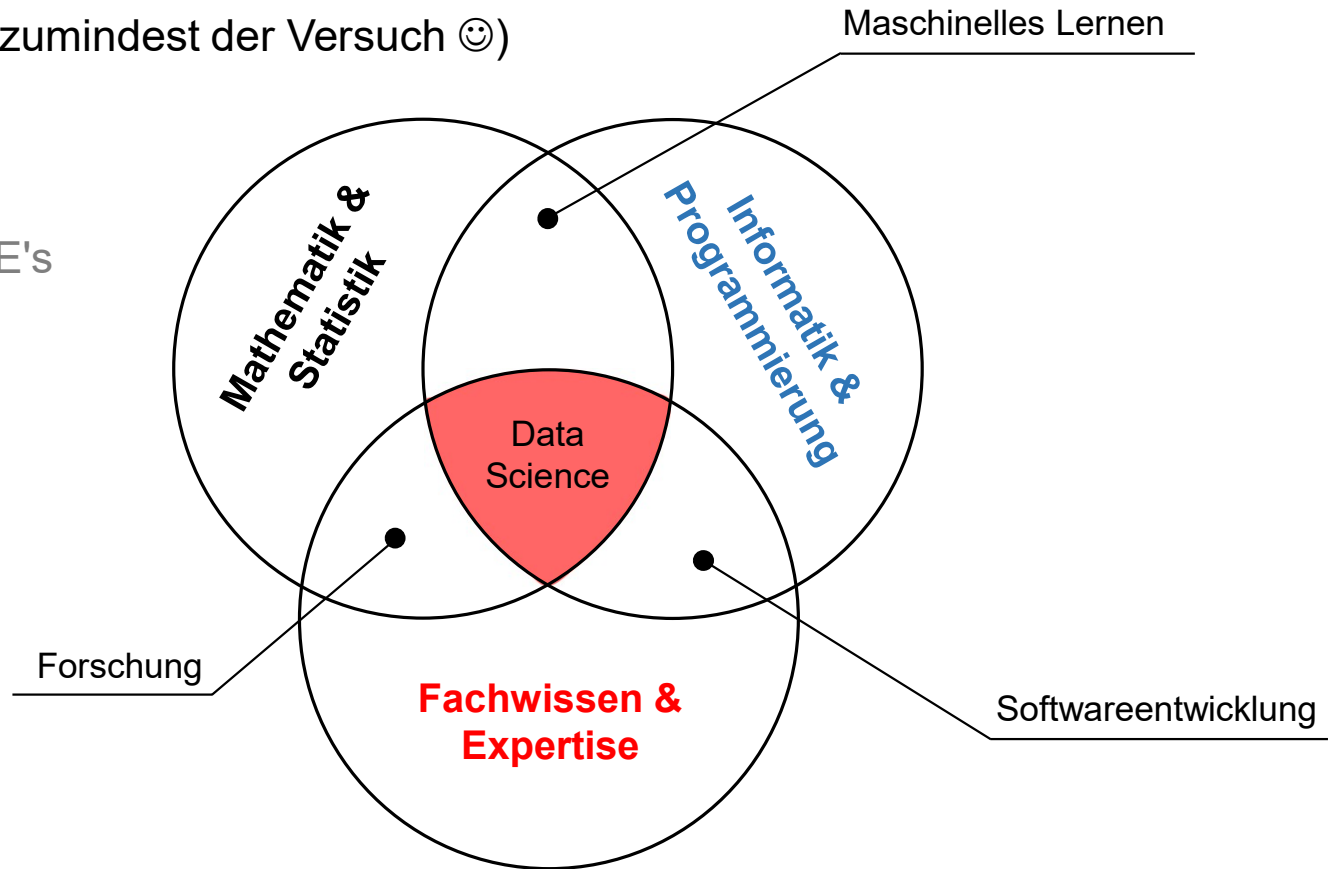
**Data Science**

Data Science bezeichnet generell die Extraktion von Wissen aus Daten, um daraus zu lernen. Data Science ist ein interdisziplinäres Wissenschaftsfeld, welches wissenschaftlich fundierte Methoden, Prozesse, ... [Wikipedia](#)

## Data Science: mehr als nur ein Hype...!

- Definition: Data Science (zumindest der Versuch 😊)

- Statistik
- Lineare Algebra
- Analysis, ODE's / PDE's
- Algorithmen
- Datenstrukturen
- Programmierung
- Fachwissen
- Expertise
- ...



## Data Science: mehr als nur ein Hype...!

- Ausgangslage & Begrifflichkeiten
  - Daten / latente Informationen
  - Analysen / Auswertungen
  - Modelle / Vorhersagen / Klassifikationen
  - Lernen / Lernerfolge
    - überwachtes Lernen (engl. *supervised learning*)
    - unüberwachtes Lernen (engl. *unsupervised learning*)
  - Algorithmen / Programme
  - Exploration / Visualisierung
- **Ziel:** (neue) Erkenntnisse gewinnen / besseres Verständnis / bessere Prognosen

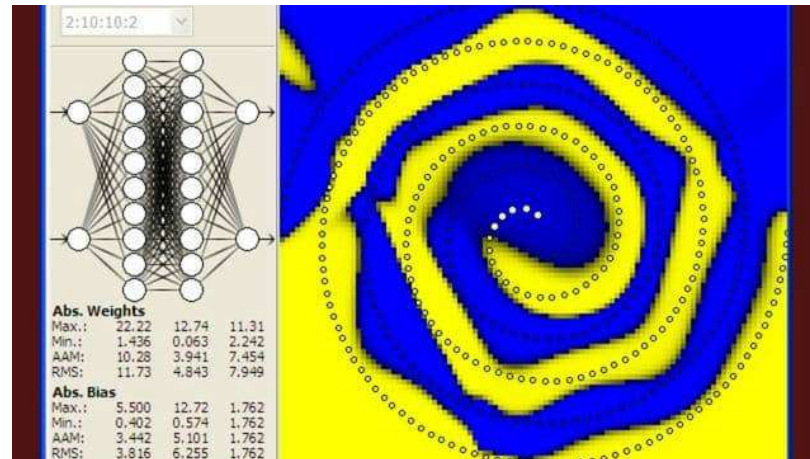
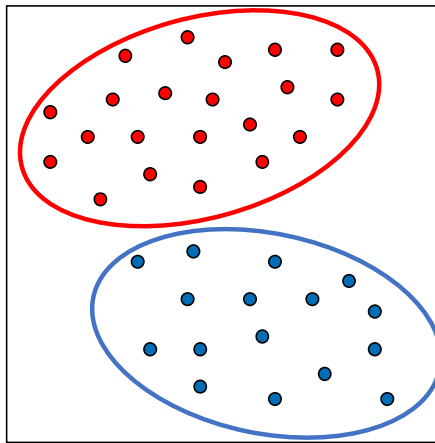
## Bevor wir starten...



- Python: benötigte Pakete
  - **numpy**: Paket für wissenschaftliches Rechnen (insb. numerische Algorithmen)
  - **matplotlib**: Paket für Visualisierung von Daten
  - **scikit-learn**: Paket mit verschiedenen Data Science-Werkzeugen
- Programmierumgebung
  - PyCharm
  - Integrated Development and Learning Environment (IDLE Shell)
  - Jupyter Notebook
  - *...you name it...*

## Klassifikations- / Vorhersagemodelle

- viele Aufgaben lassen sich (mathematisch) als Klassifikations-/Vorhersageproblem ausdrücken



source: vimeocdm.com

- Fragen über Fragen
  - welche Algorithmen, welche Modelle
  - 'löst meine Algorithmus das Problem richtig' oder 'löse ich das richtige Problem'

## Klassifikations- / Vorhersagemodelle

- typische Vertreter
  - lineare Regression (bekannt aus der Statistik)
  - *k*-nächste-Nachbarn (*k-nearest neighbours* oder *k-NN*)
  - *k*-Mitten (*k-means*)
- Vorgehen
  - mathematisches Grundlagen
  - algorithmisches Design
  - prototypische Umsetzung
  - Spass beim Ausprobieren 😊

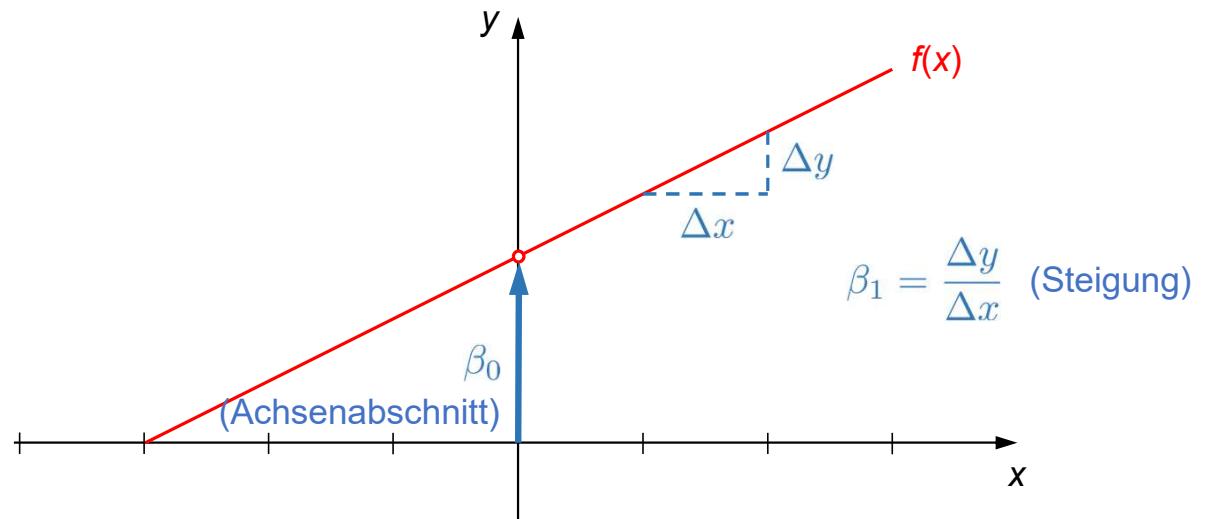
## Phantastische Regressionswesen und wo sie zu finden sind...

- lineare Regression
  - eine der am meisten genutzten statistischen Methoden
  - drückt die mathematische Beziehung zwischen zwei Variablen aus
  - **Annahme**: lineare Zusammenhang zwischen **Antwort**- und **Prädiktorvariable**
  - Linearität vereinfacht möglichen tatsächlichen Zusammenhang, ABER sie ist ein guter Startpunkt für weitere Untersuchungen
- oder frei nach Hamlet: *linear oder nicht-linear, das ist hier die Frage...?*
  - 1) je mehr Waren verkauft werden, desto höher der Umsatz
  - 2) jede infizierte Person, steckt zwei weitere Personen an



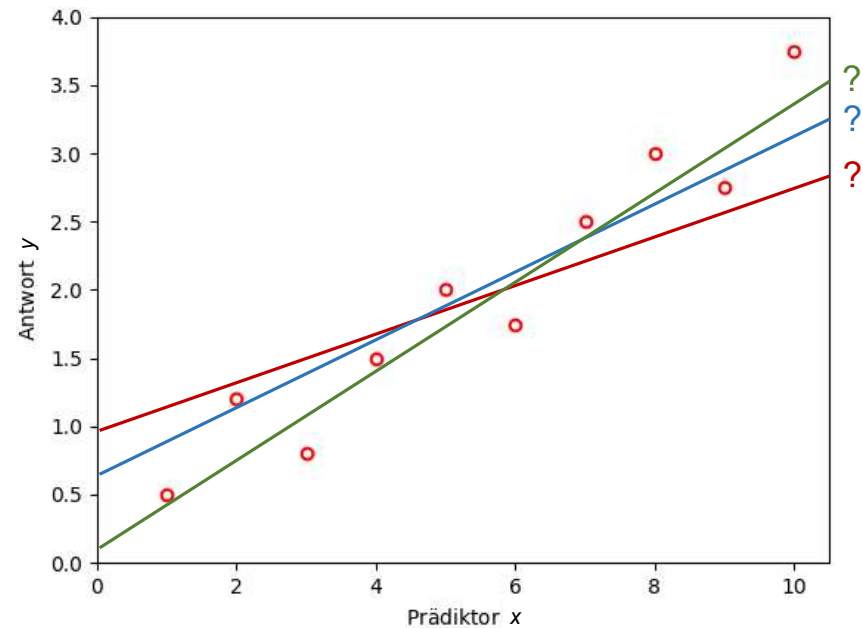
# Phantastische Regressionswesen und wo sie zu finden sind...

- warum lineare Regression...?
  - um Beziehungen besser zu verstehen / besser beschreiben zu können
  - etwa: gibt es einen Zusammenhang zwischen Anzahl der Freunde in sozialen Medien und der Zeit, die eine Person täglich auf derartigen Plattformen verbringt
- Ausgangspunkt: Geradengleichung
  - $y = f(x) = \beta_0 + \beta_1 * x$
- Preisfrage: Bedeutung...?
  - $\beta_1 = 0, 1, \infty$



## Mal wieder ein bisschen Statistik 😊

- lineare Regression
  - Ausgangslage:  $n$  Punkte mit Koordinaten  $(x_i, y_i)$
  - gesucht: Modell für Zusammenhang zwischen Prädiktor- ( $x$ ) und Antwortvariable ( $y$ )
  - wichtig: Modell soll Tendenz / Abweichung berücksichtigen
- Aufgabe: finde optimale Gerade
$$y = f(x) = \beta_0 + \beta_1 * x,$$
d.h. finde **beste** Werte  $\beta_0, \beta_1$  für gegebene Punkte  $(x_i, y_i)$
- Frage: was ist optimale Gerade...?



## Mal wieder ein bisschen Statistik 😊

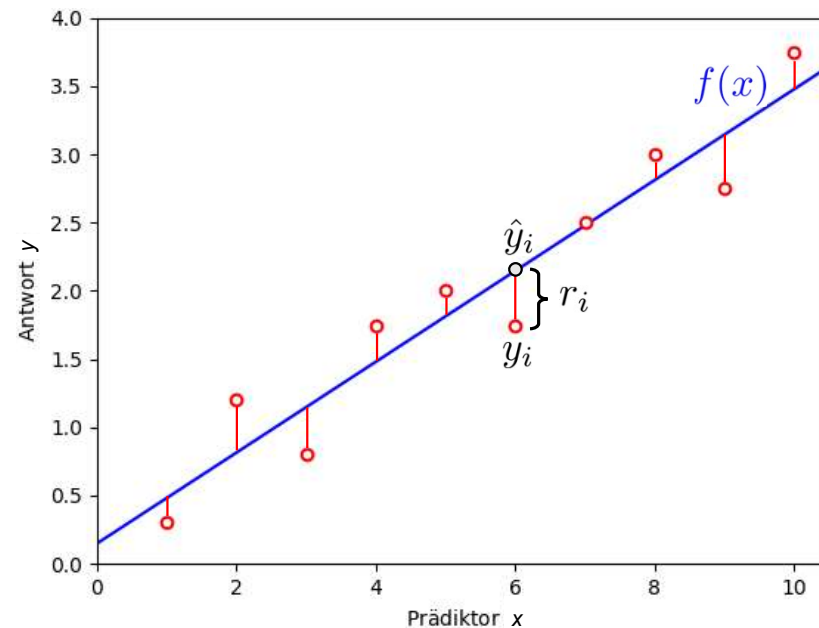
- lineare Regression
  - Definition von Optimalität (*least squares method*)
    - Gerade  $f(x)$  liegt optimal zu allen Punkten  $(x_i, y_i) \leftrightarrow$  vertikale Abstände (genauer: Quadrat der Abstände) zwischen den Punkten und der Regressionsgeraden minimal

- Bestimmung der Abstände (Residuum)

$$r_i = y_i - \hat{y}_i \stackrel{f(x)}{=} y_i - \beta_0 - \beta_1 * x_i$$

- Summe der Fehlerquadrate (RSS)

$$\begin{aligned} RSS(\beta_0, \beta_1) &= \sum_i r_i^2 \\ &= \sum_i (y_i - \beta_0 - \beta_1 * x_i)^2 \end{aligned}$$



## Phantastische Regressionswesen und wo sie zu finden sind...



- kurzes Pythonmezzo
  - Daten in Python einlesen und plotten
  - Nutzung der Bibliotheken `numpy`, `matplotlib`

- Textdatei (CSV) laden

```
import numpy as np  
data = np.loadtxt("meine_textdatei.txt", delimiter=",")
```

- Ausgabe einzelner Zeilen und Spalten mit „:“

```
print(data[:,0])  
print(data[:,1])
```

## Phantastische Regressionswesen und wo sie zu finden sind...



- Daten plotten

```
import matplotlib.pyplot as plt
plt.xlabel('Text')
plt.ylabel('Text')
plt.title('Text')
plt.axis([xmin, xmax, ymin, ymax])
plt.grid(True)
plt.plot(data_x, data_y, style)1
plt.show()
```

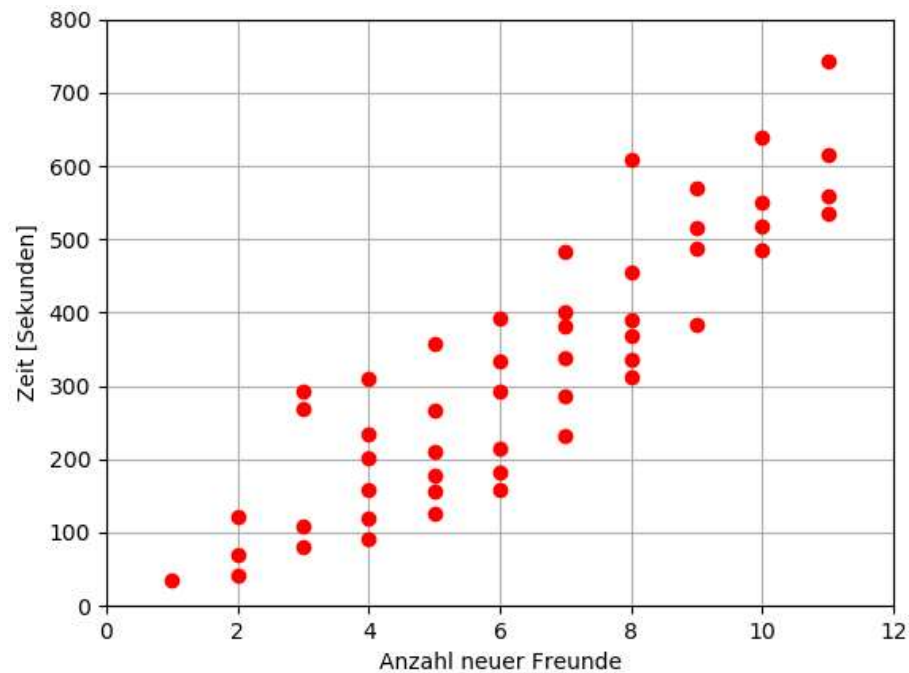
# x: data[:,0], y: data[:,1]

---

<sup>1</sup> siehe auch [https://matplotlib.org/api/markers\\_api.html](https://matplotlib.org/api/markers_api.html)

## Phantastische Regressionswesen und wo sie zu finden sind...

- einlesen und plotten (als rote Punkte mittels 'ro') der Datei 'smp\_data.txt'



## Phantastische Regressionswesen und wo sie zu finden sind...

- ...zurück zur Regressionsgeraden

- Gegeben:  $RSS(\beta_0, \beta_1) = \sum_i r_i^2$
- Gesucht:  $\beta_0$  und  $\beta_1$  sodass  $\min_{\beta_0, \beta_1} \sum_i r_i^2$



© Sidney Harris

- aus der Analysis: Minimum einer Funktion durch Ableitung

- $RSS(\beta_0, \beta_1) = \sum_i (y_i - \beta_0 - \beta_1 * x_i)^2$
- Ableitung nach  $\beta_0$

$$\sum_i 2(y_i - \beta_0 - \beta_1 * x_i)(-1) = -\sum_i y_i + \sum_i \beta_0 + \sum_i \beta_1 * x_i = 0 \quad (1)$$

- Ableitung nach  $\beta_1$

$$\sum_i 2(y_i - \beta_0 - \beta_1 * x_i)(-x_i) = -\sum_i x_i * y_i + \sum_i \beta_0 * x_i + \sum_i \beta_1 * (x_i)^2 = 0 \quad (2)$$

## Phantastische Regressionswesen und wo sie zu finden sind...

- damit erhalten wir Formeln zur Berechnung von  $\beta_0$  und  $\beta_1$ 
  - aus (1) folgt

$$n * \beta_0 + \beta_1 \sum_i x_i = \sum_i y_i$$

$$\beta_0 = \frac{1}{n} \sum_i y_i - \frac{1}{n} * \beta_1 \sum_i x_i \quad (3) \quad (\beta_0 = \bar{y} - \beta_1 * \bar{x})$$

- aus (2) folgt

$$\beta_0 \sum_i x_i + \beta_1 \sum_i (x_i)^2 = \sum_i x_i * y_i \quad (4)$$

- aus (3) in (4) folgt

$$\beta_1 = \frac{n \sum_i x_i * y_i - \sum_i x_i \sum_i y_i}{n \sum_i (x_i)^2 - (\sum_i x_i)^2}$$



## Phantastische Regressionswesen und wo sie zu finden sind...

- Berechnung der Regressionsgeraden  $y = f(x) = \beta_0 + \beta_1 * x$

$$\beta_0 = \frac{1}{n} \sum_i y_i - \frac{1}{n} * \beta_1 \sum_i x_i$$

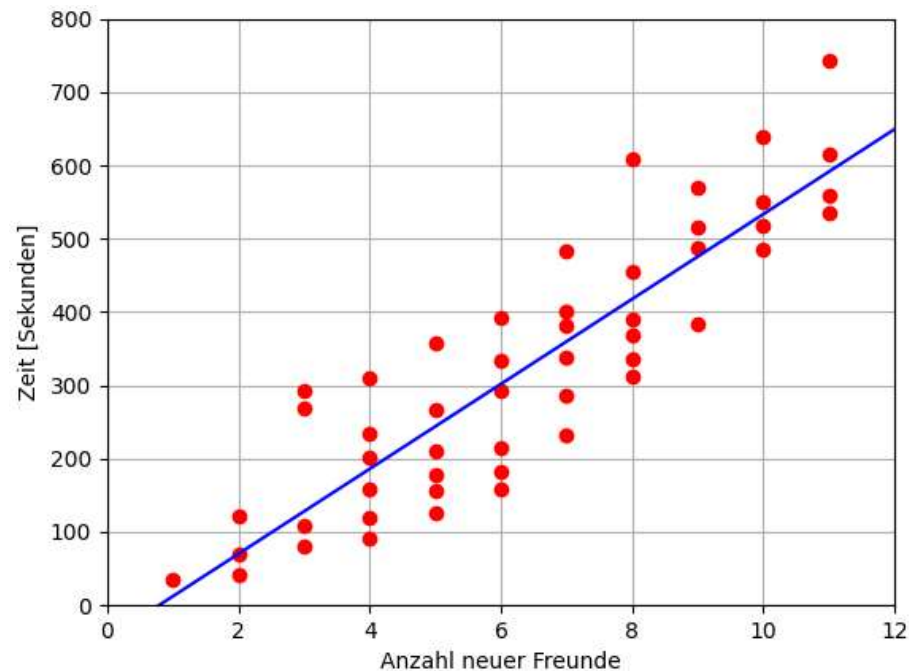
$$\beta_1 = \frac{n \sum_i x_i * y_i - \sum_i x_i \sum_i y_i}{n \sum_i (x_i)^2 - (\sum_i x_i)^2}$$

- aus unseren Daten 'smp\_data.txt' folgt

- $\beta_0 = -45.50$

- $\beta_1 = 57.88$

- Frage: geht's auch einfacher...?



## Paket: scikit-learn (Machine Learning in Python)



- Modell aufsetzen und trainieren

```
from sklearn.linear_model import LinearRegression as lr
model = lr()
model.fit(xdata, ydata)
```

- Problem: *xdata* ist Spaltenvektor → Daten umformen (`numpy.reshape`)

```
xdata = data[:,0].reshape((-1, 1))
ydata = data[:,1]
```

- Bedeutung von  $(-1, 1)$ : unbekannte Anzahl an Zeilen bei genau einer Spalte → Spaltenvektor

## Paket: scikit-learn (Machine Learning in Python)



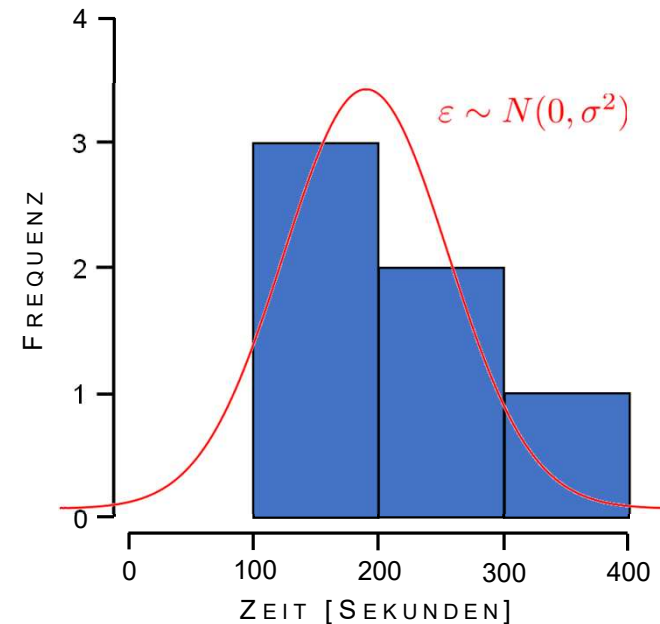
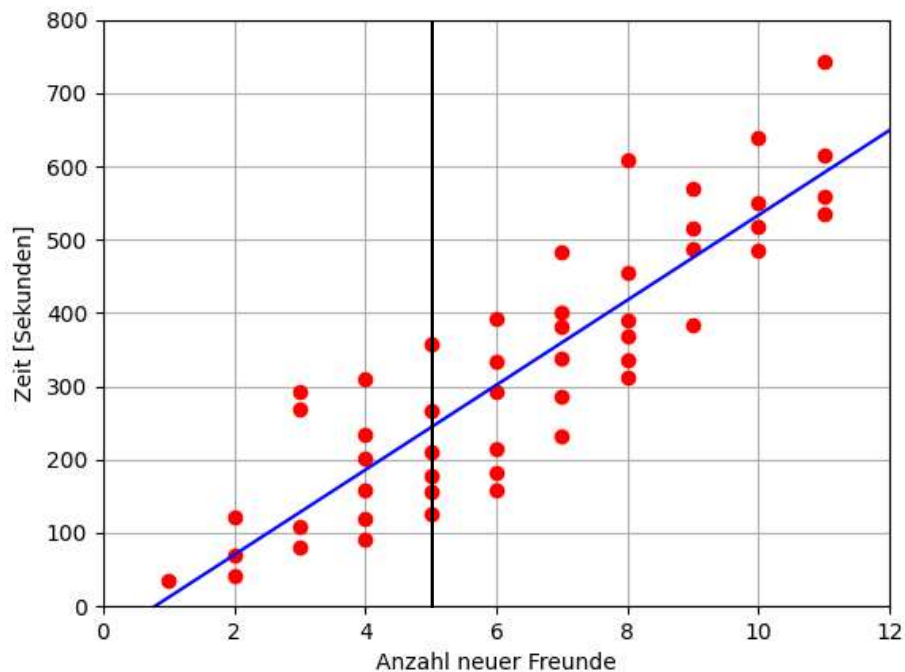
- das ganze Programm...

```
import ...  
data = np.loadtxt('smp_data.dat', delimiter=',')  
xdata = data[:,0].reshape((-1,1))  
ydata = data[:,1]  
model = lr()  
model.fit(xdata, ydata)  
y_pred = model.predict(xdata)  
plt.plot(xdata, y_pred, 'b-')  
plt.plot(xdata, ydata, 'ro')  
plt.show()
```

# Vorhersage bestimmen...

## Ein kleine Geschichte über die Fehleranalyse...

- mittels Regressionsgerade lässt sich für jeden x-Wert der zugehörige y-Wert abschätzen
- ABER: wie viel Vertrauen haben wir in diese Ergebnisse



## Ein kleine Geschichte über die Fehleranalyse...

- modifizierte Gleichung

$$y = f(x) = \beta_0 + \beta_1 * x + \varepsilon$$

mit  $\varepsilon$  als Fehler (oder Residuum oder Rauschen)

- Frage: kann es eine bessere Gerade geben mit kleineren Fehlern...? ✗
- Frage: können wir den Fehler berechnen...? ✓
- Frage: lässt sich damit eine Aussage über die Streuung treffen...? ✓

- Berechnung des Fehlers

- zur Erinnerung:  $r_i = y_i - \hat{y}_i$
- damit ergibt sich **mittlere quadratische Abweichung**:  $MSE = \frac{\sum_i r_i^2}{n}$

## Ein kleine Geschichte über die Fehleranalyse...

- MSE ist ein gutes Mass für die **Varianz** (vgl. Varianz aus der Statistik), d.h. wie stark die vorhergesagten Werte ( $\hat{y}_i$ ) von den tatsächlichen / beobachteten Werten ( $y_i$ ) variieren



```
from sklearn.metrics import mean_squared_error as mse
```

```
mse(y_true, y_pred, squared=True)
```

optional: **False** liefert RMSE

- Bestimmtheitsmass  $R^2$ 
  - drückt Anteil der Variabilität in den Messwerten aus

- Berechnung:  $r2\_score = 1 - \frac{\sum_i r_i^2}{\sum_i (y_i - \bar{y})^2}$

```
from sklearn.metrics import r2_score
```

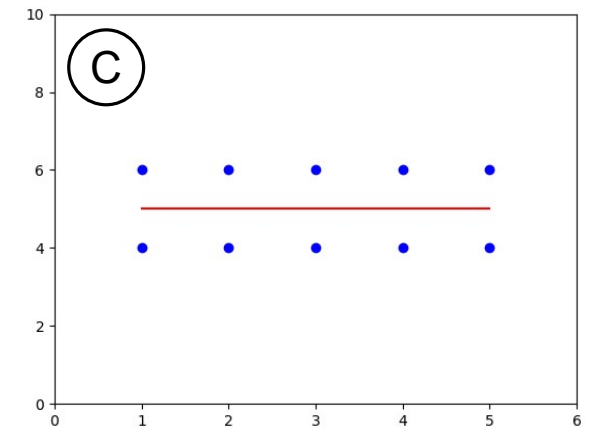
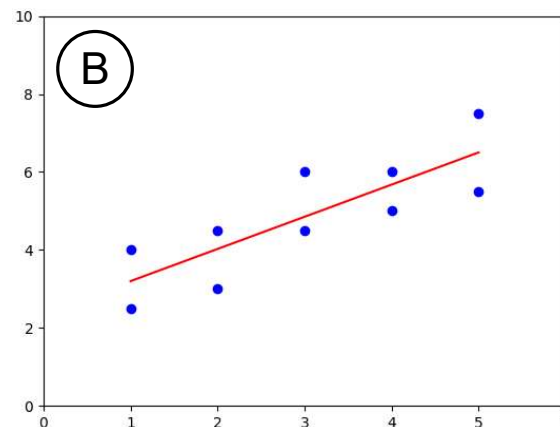
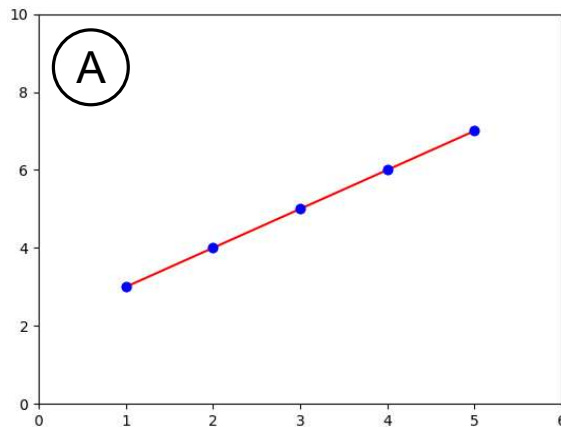
```
r2_score(y_true, y_pred)
```

Vorsicht: je kleiner der MSE, desto kleiner der Fehler, aber je kleiner  $R^2$ , desto höher die Variabilität (→ desto schlechter die Güte der Regressionsgeraden)

## Ein kleine Geschichte über die Fehleranalyse...



- kleines Fehlerquiz
  - für drei Datensätze A, B und C wurde eine lineare Regression durchgeführt

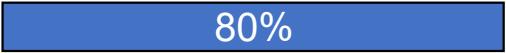
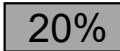


- folgende Werte wurden für Bestimmtheitsmass und mittlere quadratische Abweichung ermittelt

$R^2$ : 0.0, 0.7, 1.0    sowie    MSE: 0.0, 0.6, 1.0

- Frage: welche Werte gehören zu welchem Datensatz...?

## Ein kleine Geschichte über die Fehleranalyse...

- Kreuzvalidierung (engl. *cross validation*)
  - Datensatz wird in zwei Teile zerlegt (z.B.)
    - Trainingsdaten (  80% )
    - Testdaten (  20% )
  - Modell wird mit Trainingsdaten trainiert
  - MSE der Testdaten berechnen und mit MSE der Trainingsdaten vergleichen
  - sind die Werte nahezu identisch
    - Modell ist brauchbar
    - andernfalls Gefahr des Under- / Overfitting (Unter- / Überanpassung)



## Ein kleine Geschichte über die Fehleranalyse...



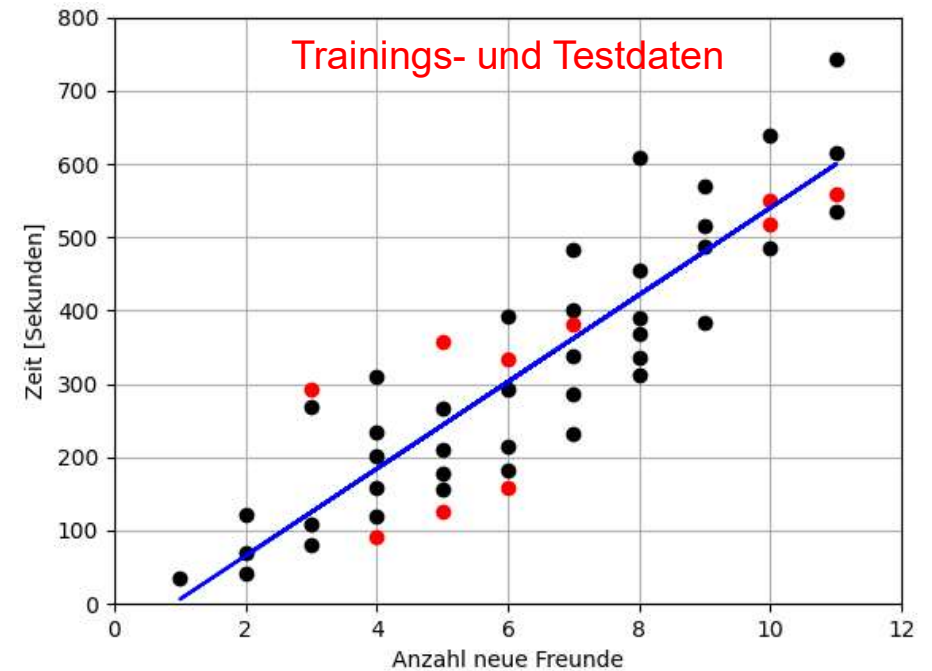
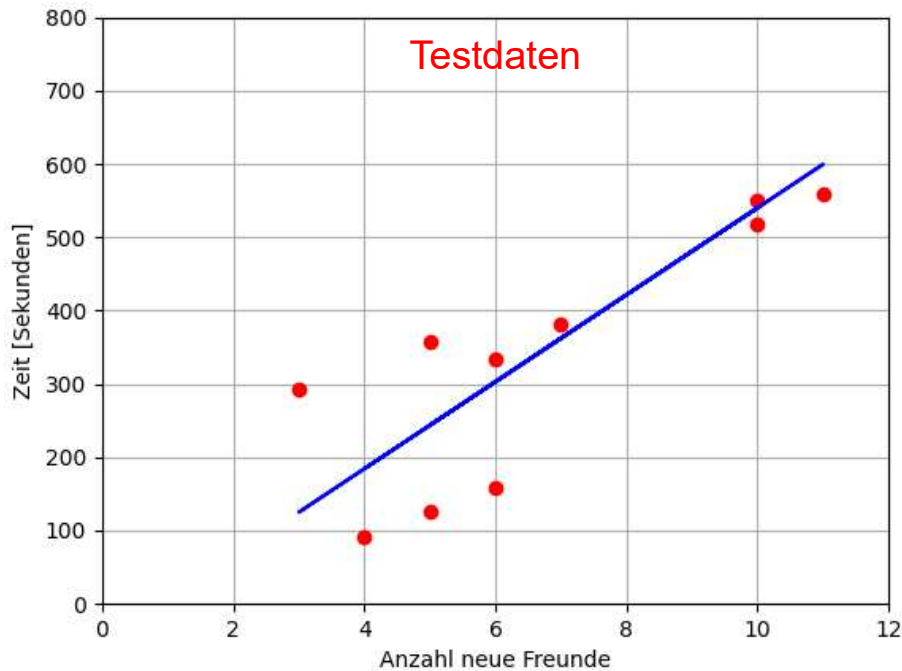
- Kreuzvalidierung im praktischen Einsatz

```
from sklearn.model_selection import train_test_split  
x_train, x_test, y_train, y_test = train_test_split(  
    xdata, ydata, test_size=val_ts, random_state=val_rs)
```

- Werte
  - $val\_ts \in [0, 1] \rightarrow$  Grösse (in [%]) der Testdaten ( $\Rightarrow train\_size = 1 - val\_ts$ )
  - $val\_rs$  [int]  $\rightarrow$  bestimmt zufällige Durchmischung der Daten vor dem Split
- Aufgabe
  - Aufteilung der Daten ( $val\_ts = 0.2$ ) und Bestimmung des Fehlers ( $MSE, R^2$ )

## Ein kleine Geschichte über die Fehleranalyse...

- Kreuzvalidierung: Ergebnisse



$$MSE = 8807.15, \quad R^2 = 0.67$$

## Praktische Übung



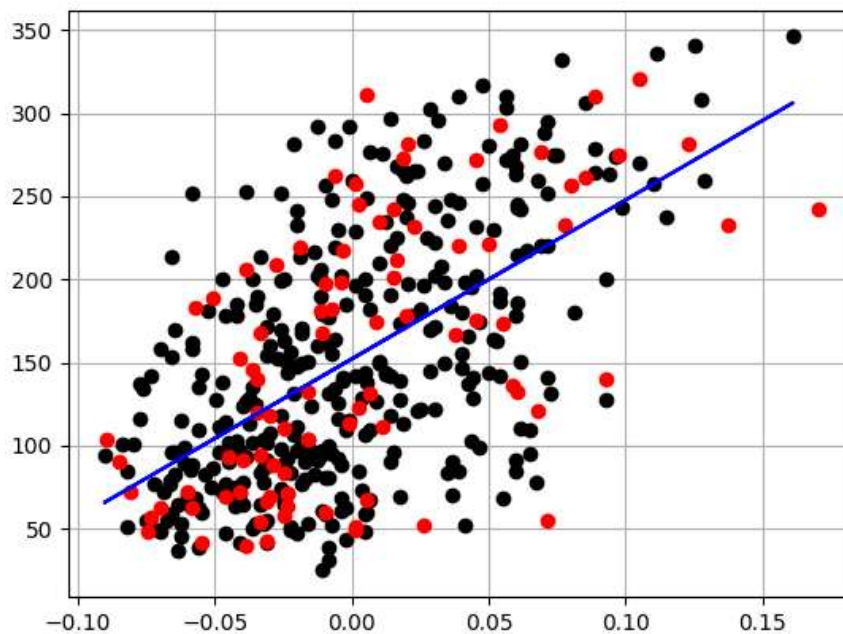
- Datensatz: Diabetes (aus dem Paket `sklearn`)

```
from sklearn import datasets  
  
diabetes_x, diabetes_y = datasets.load_diabetes(return_X_y=True)  
  
diabetes_x = diabetes_x[:, np.newaxis, 2]
```

- Aufgabe
  - Datensatz in Trainings- und Testdaten aufteilen (80% : 20%)
  - Modell trainieren
  - Modell testen und Fehler ( $MSE$ ,  $R^2$ ) bestimmen
  - Ergebnisse plotten

## Praktische Übung

- Datensatz: Diabetes (aus dem Paket `sklearn`)



$R^2_{\text{train}} = 0.338$

$R^2_{\text{test}} = 0.362$

Fragen...?