Business Analytics

04 | Diagnostische Analyse - Part 1

Prof. Dr. Felix Zeidler | FH Bielefeld | SoSe 2023

Inhaltsverzeichnis

(1) Programmierkonzept: For-Schleife

Programmierkonzept

Importe und Einstellungen

Programmierkonzept

Was ist eine For-Schleife?

Was ist eine For-Schleife?

Eine For-Schleife ist eine Methode in der Programmierung, um wiederholende Aufgaben zu vereinfachen, indem sie eine bestimmte Aktion mehrmals nacheinander ausführt. Sie geht Schritt für Schritt durch eine Sammlung von Elementen und wendet auf jedes Element den gleichen Vorgang an, wodurch Zeit und Aufwand gespart werden.

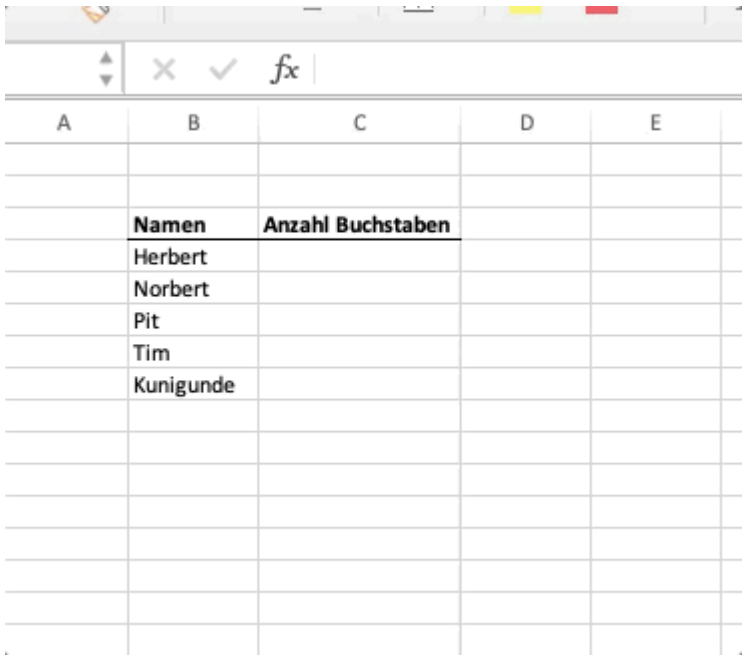
Gründe für die Verwendung einer For-Schleife:

- **Automatisierung:** For-Schleifen ermöglichen es, wiederkehrende Aufgaben oder Berechnungen effizient durchzuführen, ohne redundanten Code zu schreiben.
- **Lesbarkeit:** For-Schleifen verbessern die Lesbarkeit und Verständlichkeit des Codes, indem sie klare Strukturen für wiederholende Prozesse bieten.
- **Flexibilität:** For-Schleifen können mit verschiedenen iterierbaren Objekten arbeiten, was sie für zahlreiche Anwendungsfälle und Datenstrukturen nützlich macht.

Programmierkonzept

Analogie: Excel

Excel: For-Schleife



	A	B	C	D	E
		Namen	**Anzahl Buchstaben**		
		Herbert			
		Norbert			
		Pit			
		Tim			
		Kunigunde			

Kommentare:

- Funktion `LÄNGE` in jedem Wort
- **runterkopieren** der Formel wendet diese auf jedes Wort in den nächsten Zeilen an

> For-Scheifen in Python sind konzeptionell das gleiche wie **runterkopieren** in Excel

Programmierkonzept

Syntax in Python

Syntax in Python

Die Syntax einer For-Schleife in Python ist wie folgt:

```
1 for element in collection:
2     # do something with element
```

- `element`: Der Name der Variablen, die für jedes Element der Sammlung verwendet wird. Kann frei gewählt werden.
- `in`: Das Schlüsselwort, das die Schleife mit der Sammlung verbindet.
- `collection`: Die Sammlung, die durchlaufen werden soll. Kann jedes Objekt sein, das iterierbar ist, z. B. eine Liste

Programmierkonzept

Beispiel: For-Schleife

Bilden wir das Excel-Beispiel aus der vorherigen Folie in Python um.

```
1 namen = ["Herbert", "Norbert", "Pit", "Tim", "Kunigunde"]
2 for name in namen:
3     anz_buchstaben = len(name)
4     print(anz_buchstaben)
```

- `name`: Name der Variablen, die für jedes Element der Sammlung verwendet wird. Kann frei gewählt werden.
- `anz_buchstaben`: Name der Variablen, die für die Anzahl der Buchstaben verwendet wird. Kann frei gewählt werden.
- `len(name)`: Die Funktion `len()` gibt die Länge eines Objekts zurück. In diesem Fall die Länge des Strings `name`.
- `print(anz_buchstaben)`: Die Funktion `print()` gibt den Inhalt der Variablen `anz_buchstaben` aus.

Programmierkonzept

Beispiel: For-Schleife (cont'd)

Wichtig: im Beispiel werden die Ergebnisse nur ausgegeben, jedoch nicht gespeichert.

- die Variable `anz_buchstaben` wird für jeden Durchlauf der Schleife neu definiert, d.h. der Inhalt der Variable wird überschrieben
- die Variable `anz_buchstaben` hat nach dem letzten Durchlauf der Schleife den Wert der letzten Iteration

Beispiel: Ausgabe der Variable `anz_buchstaben` nach dem letzten Durchlauf der Schleife

```
1 namen = ["Herbert", "Norbert", "Pit", "Tim", "Kunigunde"]
2 for name in namen:
3     anz_buchstaben = len(name)
4
5 anz_buchstaben
```

9

Programmierkonzept

Beispiel: For-Schleife mit Speicherung

Wenn wir die Ergebnisse der Berechnung speichern möchten, können wir eine leere Liste definieren und die Ergebnisse der Berechnung mit der Funktion `append()` hinzufügen.

Beispiel:

```
1 namen = ["Herbert", "Norbert", "Pit", "Tim", "Kunigunde"]
2 länge_namen = []
3 for name in namen:
4     anz_buchstaben = len(name)
5     länge_namen.append(anz_buchstaben)
6
7 länge_namen
```

```
[7, 7, 3, 3, 9]
```

- `länge_namen`: Name der Liste, in der die Ergebnisse gespeichert werden sollen. Kann frei gewählt werden.
- `länge_namen.append(anz_buchstaben)`: Die Funktion `append()` fügt ein Element an das Ende der Liste an. In diesem Fall die Anzahl der Buchstaben `anz_buchstaben`.

Programmierkonzept

Aufgabe: Mehrwertsteuer

Aufgabe: For-Schleife | Code

Wir haben eine Liste von Preisen (netto) und möchten für jeden Preis, den Preis inkl. Mehrwertsteuer (brutto) berechnen. Die Ergebnisse sollen in einer Liste gespeichert werden.

```
1 # Annahmen
2 preise = [10, 20, 30, 40, 50]
3 mehrwertsteuer_satz = 0.19
4
5 # For-Schleife zur Berechnung der Mehrwertsteuer
```

Programmierkonzept

Drei nützliche Funktionen für For-Schleifen

`enumerate()`

Die Funktion `enumerate()` gibt ein Objekt zurück, das die Elemente einer Sammlung enthält und die Nummer des Elements enthält.

```
1 namen = ["Herbert", "Norbert", "Pit", "Tim", "Kunigunde"]
2 for i, name in enumerate(namen):
3     print(i, name)
```

- `i`: Nummer des Elements (beginnend bei 0)
- `name`: Element der Sammlung

Programmierkonzept

Drei nützliche Funktionen für For-Schleifen

`zip()`

Die Funktion `zip()` gibt ein Objekt zurück, das die Elemente mehrerer Sammlungen enthält.

```
1 vornamen = ["Herbert", "Norbert", "Pit", "Tim", "Kunigunde"]
2 nachnamen = ["Müller", "Schmidt", "Meier", "Schulze", "Schmidt"]
3
4 for vorname, nachname in zip(vornamen, nachnamen):
5     print(vorname, nachname)
```

- `vorname`: Element der Sammlung `vornamen`
- `nachname`: Element der Sammlung `nachnamen`

Programmierkonzept

Drei nützliche Funktionen für For-Schleifen

`range()`

Die Funktion `range()` gibt ein Objekt zurück, das eine Sequenz von Zahlen enthält.

```
1 for i in range(5):
2     print(i)
```

- `i`: Element der Sequenz von Zahlen, die von `range()` zurückgegeben wird
- `range` kann mit drei Argumenten aufgerufen werden: `range(start, stop, step)`
  - `start`: Startwert der Sequenz (Standardwert: 0)
  - `stop`: Stopwert der Sequenz (Standardwert: 1)
  - `step`: Schrittweite der Sequenz (Standardwert: 1)
  - Beispiel: `range(1, 10, 2)` gibt die Sequenz `1, 3, 5, 7, 9` zurück

Programmierkonzept

Aufgabe: rollierende Durchschnitte

Aufgabenstellung:

In dieser Übungsaufgabe erhaltet ihr eine Liste mit täglichen Aktienkursen eines Unternehmens für einen Monat. Eure Aufgabe besteht darin, die rollierenden Durchschnittskurse für einen Zeitraum von 5 Tagen zu berechnen und auszugeben.

Anforderungen:

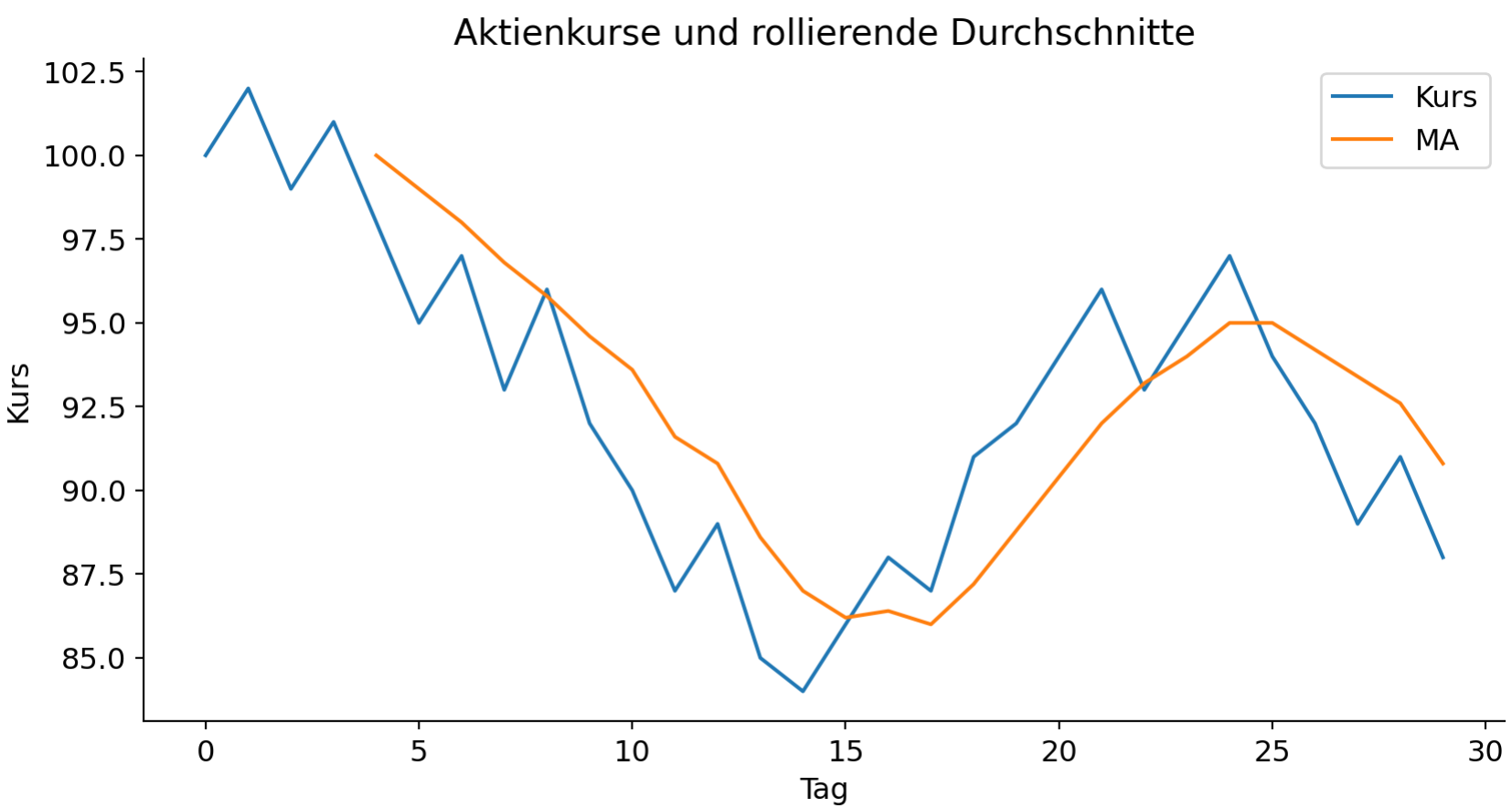
- Verwende eine For-Schleife, um durch die Liste der Aktienkurse zu iterieren.
- Berechne den 5-Tage-Durchschnitt für jeden Kurs in der Liste, beginnend mit dem fünften Tag.
- Speichere die berechneten Durchschnittskurse in einer neuen Liste.
- Speicher Aktienkurse und Durchschnittskurse in einem DataFrame
- Plotte die Aktienkurse und die Durchschnittskurse in einem Diagramm.

```
1 aktienkurse = [100, 102, 99, 101, 98, 95, 97, 93, 96, 92,
2                90, 87, 89, 85, 84, 86, 88, 87, 91, 92, 94,
3                96, 93, 95, 97, 94, 92, 89, 91, 88]
```

Programmierkonzept

Lösung: rollierende Durchschnitte

Plot | Code



Lösung: eigene Funktion

Rollierender Durchschnitt sollte - bei obigen Beispiel - im Idealfall mit eigener Funktion berechnet werden.

Beispiel:

▶ Code

Programmierkonzept

Lösung: Pandas

Rollierender Durchschnitt kann in Pandas einfach berechnet werden.

Beispiel:

▶ Code

(2) Modelle: Lineare Regression

Lernziele und Ziele des Kapitels

Ziele dieses Kapitels

1. (Wiederholung von) Grundlagen “linearer Regression”
2. Verwendung linearer Regression zur Beurteilung von Zusammenhängen zwischen Variablen
3. Beurteilung der “Güte” von linearen Regressionsmodellen
4. Erlernen der Durchführung linearer Regression in Python

Konzept: Lineare Regression

Lineare Regression ist eine der flexibelsten und am häufigsten verwendeten statistischen Methoden in Forschung und betrieblicher Praxis. Sie wird verwendet, um die Beziehung zwischen einer **abhängigen** und einer oder mehreren **unabhängigen** Variablen zu analysieren.

Lineare Regression wird verwendet für

1. **Inferenz**, d.h. zum Testen einer zuvor entwickelten Hypothese über die Beziehung zwischen interessierenden Variablen
2. **Prognose**, d.h. zur Schätzung des Wertes einer abhängigen Variable anhand der Werte unabhängiger Variablen

Der primäre Anwendungsfall für die lineare Regressionsanalyse ist die Analyse von **kausalen Beziehungen**.

Diese Beziehung kann ausgedrückt werden als

$$y = f(x)$$

Einfache Lineare Regression

Einfache (lineare) Regression:

Wenn wir ausdrücken möchten, dass wir an eine Beziehung zwischen *Umsatz* und *Preis* glauben, können wir dies wie folgt angeben:

$$\text{Umsatz} = f(\text{Preis})$$

Mit Hilfe der linearen Regression kann diese **Beziehung quantifiziert** werden, d.h. wir können bestimmen, wie stark sich der *Umsatz* ändert, wenn wir den *Preis* ändern.

Stochastisches Modell:
Es ist sehr unwahrscheinlich, dass die Beziehung zwischen den oben genannten Variablen vollständig deterministisch ist (wie in der obigen Formel angenommen). Daher müssen wir dem Modell Unsicherheit hinzufügen. Das resultierende **stochastische Modell** wird häufig in der Regressionsanalyse verwendet und ist wie folgt beschrieben:

$$\hat{y} = f(x) + \epsilon$$

Hier ist ϵ eine **Zufallsvariable** (genannt Fehlerterm / Residuum), die nicht beobachtet werden kann und angenommen wird, dass sie einer Standardnormalverteilung folgt (d.h. $\epsilon \sim N(0, 1)$).

Das stochastische Modell wird benötigt, um die **Regressionsmodelle mit Hilfe von statistischen Tests zu bewerten**.

Multiple lineare Regression

In vielen (wenn nicht den meisten) Forschungs- oder Geschäftsfragen können wir keine monokausale Beziehung annehmen.

Stattdessen wird y wahrscheinlich von zahlreichen Faktoren beeinflusst.

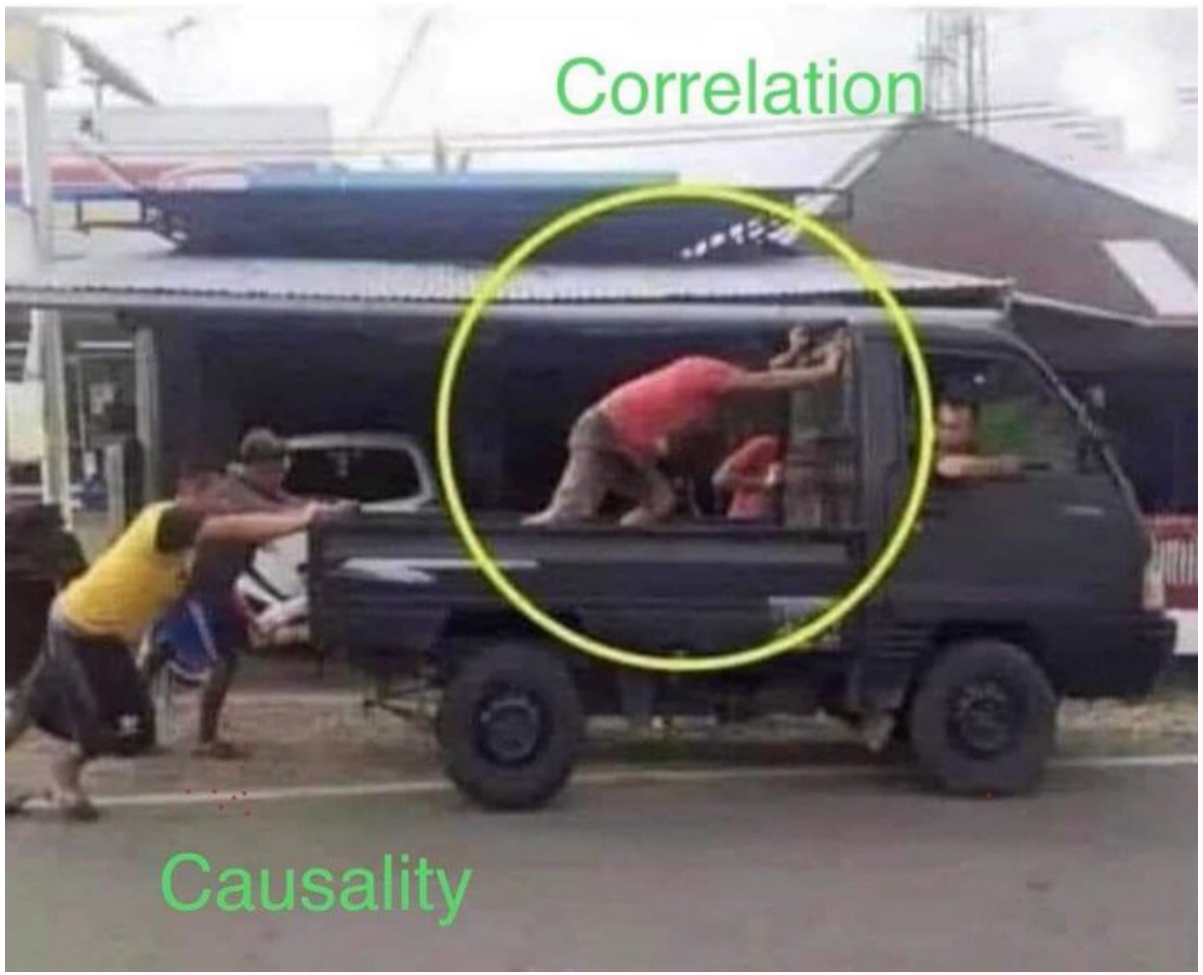
In unserem obigen Beispiel können *Umsätze* auch von den Werbeausgaben abhängen, aber auch von anderen Faktoren wie dem Zustand der Wirtschaft, dem Preis, dem Verhalten der Wettbewerber usw.

Für eine solche Beziehung verwenden wir eine **multiple Regressionsanalyse**, die wie folgt ausgedrückt werden kann:

$$y = f(X)$$

$$\text{wobei } X = \begin{bmatrix} 1 & x_{12} & \cdots & x_{1p} \\ 1 & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n2} & \cdots & x_{np} \end{bmatrix}$$

Kausalität vs. Korrelation



https://pbs.twimg.com/media/EzSYVwnVIAUG6v9?format=jpg&name=small

Kausalität vs. Korrelation (cont'd)

Wichtig:

- Während wir versuchen, eine **kausale** Beziehung zwischen y und X zu modellieren, können wir in der Praxis nicht feststellen, ob die Beziehung tatsächlich kausal ist.
- Stattdessen **approximieren** wir Kausalität, indem wir die **Korrelation** bewerten.

Was kann mit linearer Regression beantwortet werden

Typische Hypothesen, die unter anderem mit linearer Regression angesprochen werden können:

#	Hypothese	Abhängige Variable	Unabhängige Variable
1	Ist der Umsatz pro Verkäufer abhängig von der Anzahl der Kundenbesuche?	Umsatz pro Verkäufer (pro Zeitraum)	Anzahl Kundenbesuche pro Verkäufer (pro Zeitraum)
2	Ändert sich der Umsatz, wenn die Werbeausgaben verdoppelt werden?	Umsatz pro Zeitraum	Werbeausgaben pro Zeitraum
3	Wie wirkt sich eine Preiserhöhung von x% auf den Umsatz aus, wenn gleichzeitig die Werbeausgaben um 10% erhöht werden?	Umsatz pro Zeitraum	Werbeausgaben, Preis, …
4	…	…	…

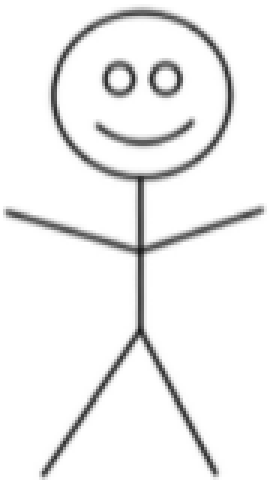
Was ist ein Modell?

Was ist ein Modell?

- Ein Modell ist eine vereinfachte Darstellung der Realität.
- Modelle sind sehr nützlich, aber es ist immer ein schmaler Grat zwischen Vereinfachung und Komplexität.
  - Wenn wir die Realität so genau wie möglich modellieren wollen, kann unser Modell zu komplex werden.
  - Wenn unser Modell zu einfach ist, beschreibt es die Realität möglicherweise nicht gut genug für unsere Zwecke. Es gibt kein gutes oder schlechtes Modell.
- Es ist hilfreicher, ein Modell als geeignet oder nicht geeignet für unser aktuelles Problem zu betrachten.

> Lineare **Regressionsmodelle** sind recht einfach, aber dennoch für viele Forschungs- und praktische Probleme sehr geeignet

Was ist ein Modell? (cont'd)



Datensatz für Beispiele

Datensatz:

Im Folgenden verwenden wir den Werbedatensatz aus "Introduction to Statistical Learning[1] für die folgenden Beispiele. Der Datensatz enthält

- Verkäufe in Tausend Einheiten
- Werbebudgets in Tausenden von Dollar für TV, Radio und Zeitungen
- Link: *https://www.statlearning.com/s/Advertising.csv*

1. Quelle: https://www.statlearning.com/resources-first-edition

Datensatz für Beispiele

```python
import pandas as pd

link_to_csv = "https://www.statlearning.com/s/Advertising.csv"
df = pd.read_csv( link_to_csv
                 ,usecols=["TV", "radio", "newspaper","sales"])
df.head()
```

	TV	radio	newspaper	sales
0	230.1	37.8	69.2	22.1
1	44.5	39.3	45.1	10.4
2	17.2	45.9	69.3	9.3
3	151.5	41.3	58.5	18.5
4	180.8	10.8	58.4	12.9

- `usecols` ist eine Liste der Spalten, die wir aus der CSV-Datei auslesen möchten.

Einfaches Modell: Verkäufe vs. TV-Werbeausgaben

Nehmen wir an, wir glauben an eine einfache lineare Beziehung zwischen **Verkäufen** und den **TV**-Werbeausgaben. Wir können dies als

$$\text{sales} = f(\text{TV})$$

beschreiben.

Dies impliziert, dass wir an eine kausale Beziehung zwischen beiden Variablen glauben. Konkret glauben wir, dass die *TV*-Werbeausgaben die *Verkäufe* antreiben oder beeinflussen.

Regressionsfunktion:

Ein einfaches lineares Regressionsmodell der oben genannten Formulierung könnte sein:

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x$$

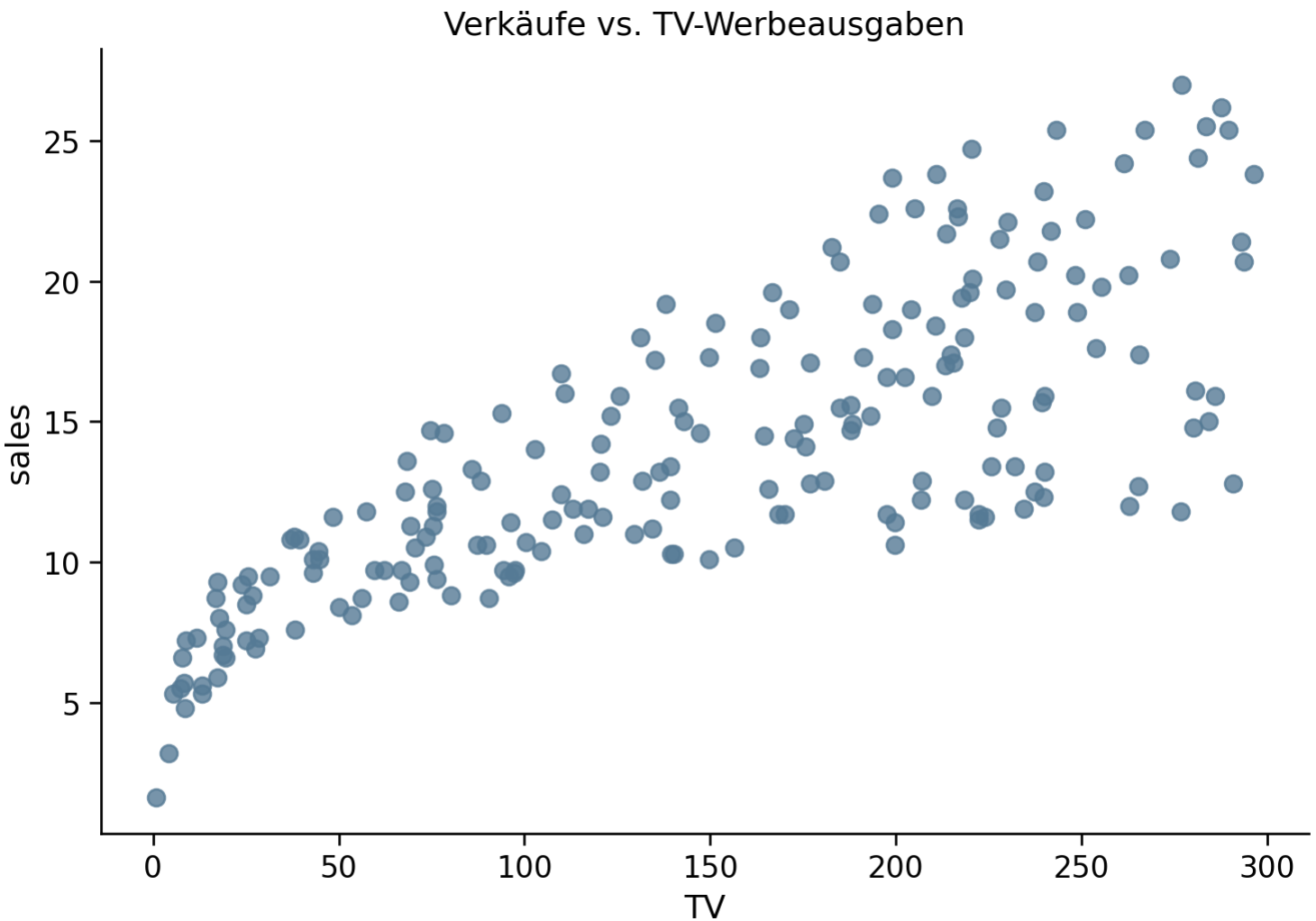
wobei $\hat{y}$ die Vorhersage der abhängigen Variable y bei gegebenem X bezeichnet. $\hat{\beta}_0$ und $\hat{\beta}_1$ sind die Koeffizientenschätzungen.

Für unser Beispiel sieht die Regressionsfunktion wie folgt aus:

$$\text{sales} = \hat{\beta}_0 + \hat{\beta}_1 \text{TV}$$

Da die mathematische Formulierung eine Linie impliziert, stellt β_0 den Schnittpunkt der Linie mit der y-Achse dar und β_1 repräsentiert die Steigung der Linie.

Einfaches Modell: Verkäufe vs. TV-Werbeausgaben



Einfaches Modell: Verkäufe vs. TV-Werbeausgaben

Nehmen wir an, dass β_0 gleich 20 und β_1 gleich 5 ist. Dies würde bedeuten, dass unser Regressionsmodell wie folgt aussieht:

$$\text{Sales} = 20 + 5 \cdot \text{TV}$$

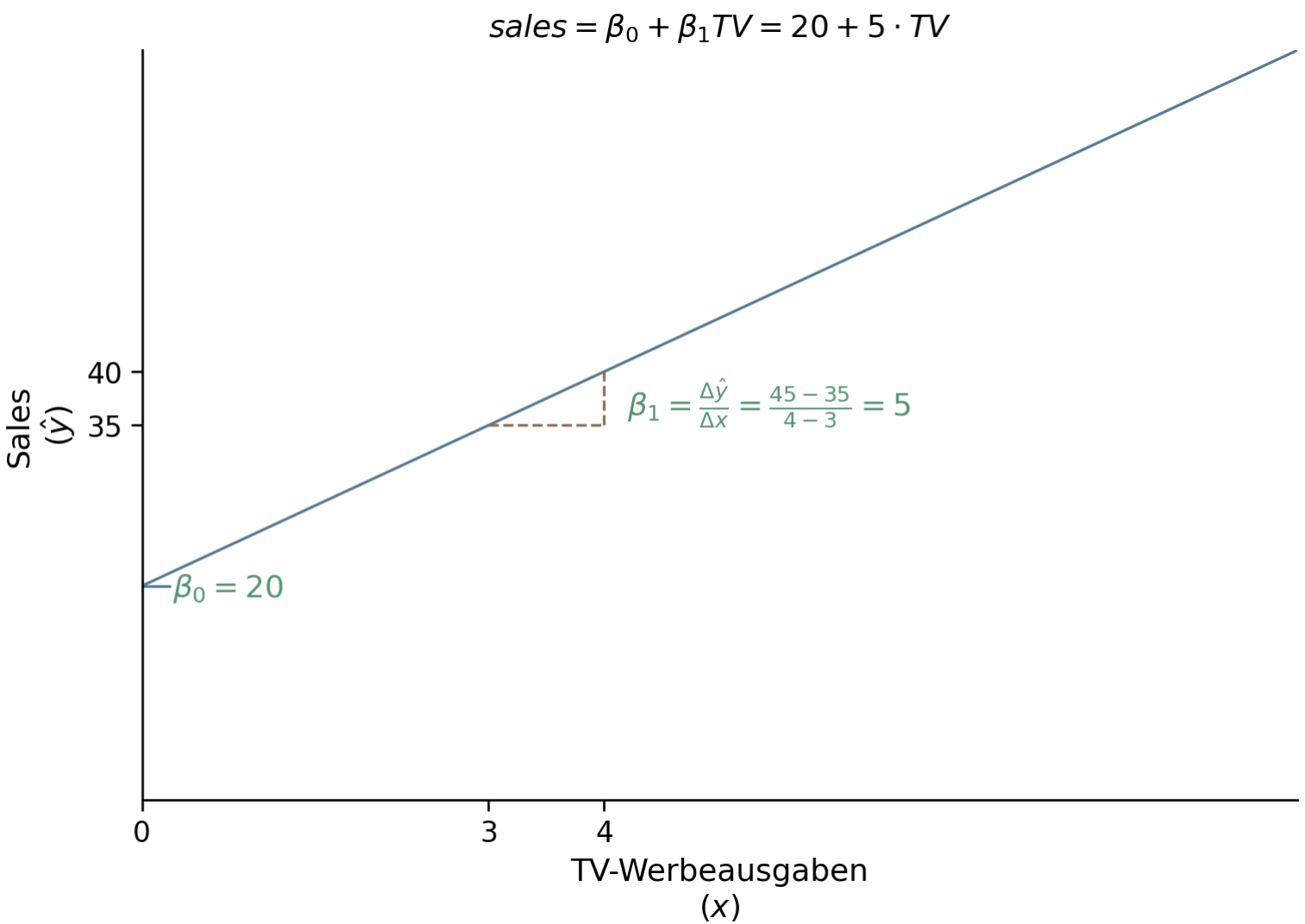
Dies würde bedeuten, dass wir x (d.h. die Höhe der Ausgaben für TV-Werbung) einsetzen und $\hat{y}$ (d.h. unsere auf x basierenden Umsatzprognosen) berechnen könnten.

Wenn wir zum Beispiel annehmen, dass wir 100 $ für TV-Werbung ausgegeben haben, würden wir Sales von

$$\text{Sales} = 20 + 5 \cdot 100 = 520$$

Wenn wir 100.000 USD für TV-Werbung ausgeben, würden wir nach unserem Modell 520.000 an Sales erzielen.

Einfaches Modell: Verkäufe vs. TV-Werbeausgaben



Einfaches Modell: analytische Lösung

Wir müssen Koeffizienten finden, die eine Linie angeben, die die wahre Beziehung so gut wie möglich beschreibt.

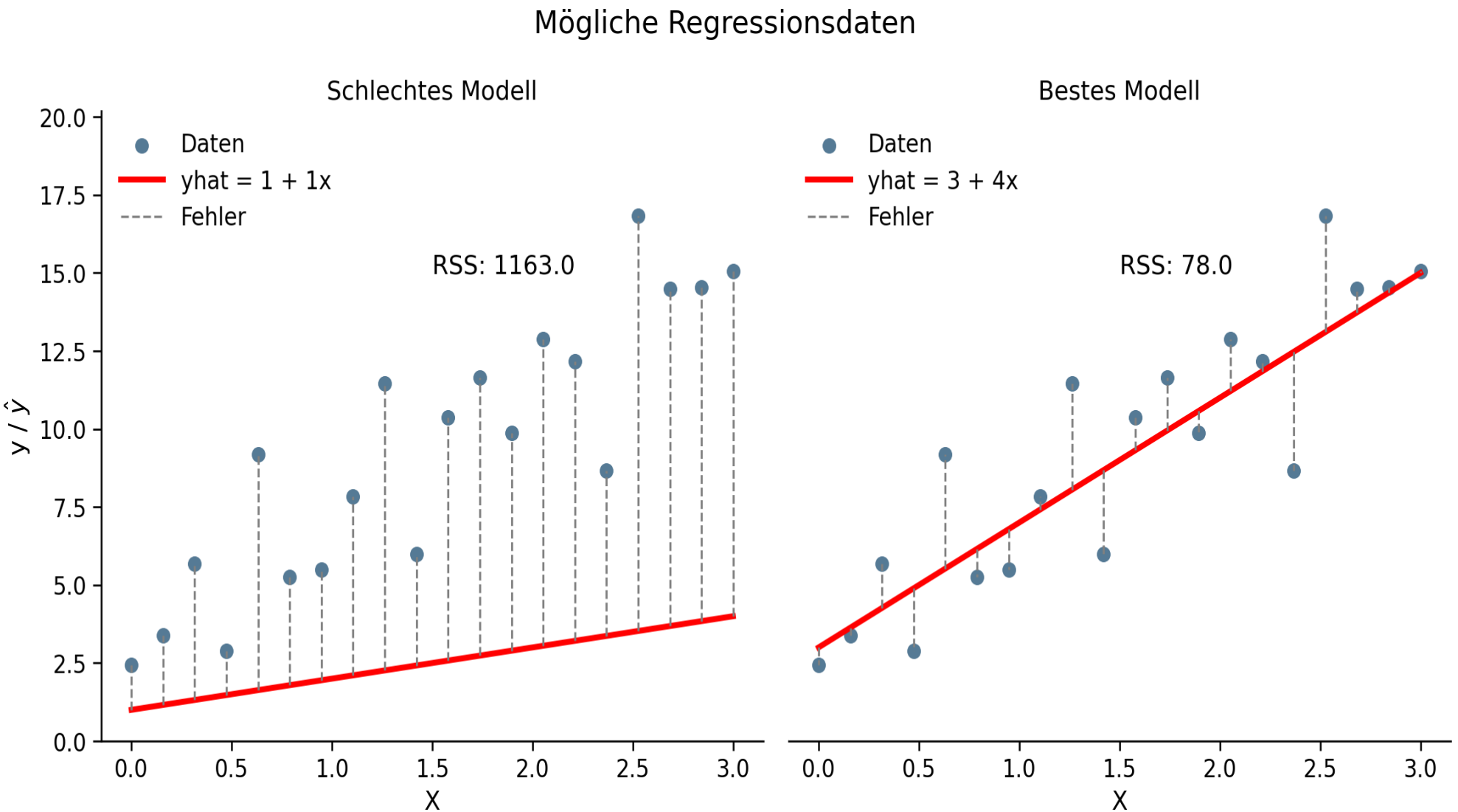
Dies wird durch die Minimierung des Kleinste-Quadrate-Kriteriums erreicht, d.h. wir wollen die Residualsumme der Quadrate (RSS; manchmal SSR) minimieren

$$\text{RSS} = \sum_{i=1}^{n} e_i^2$$

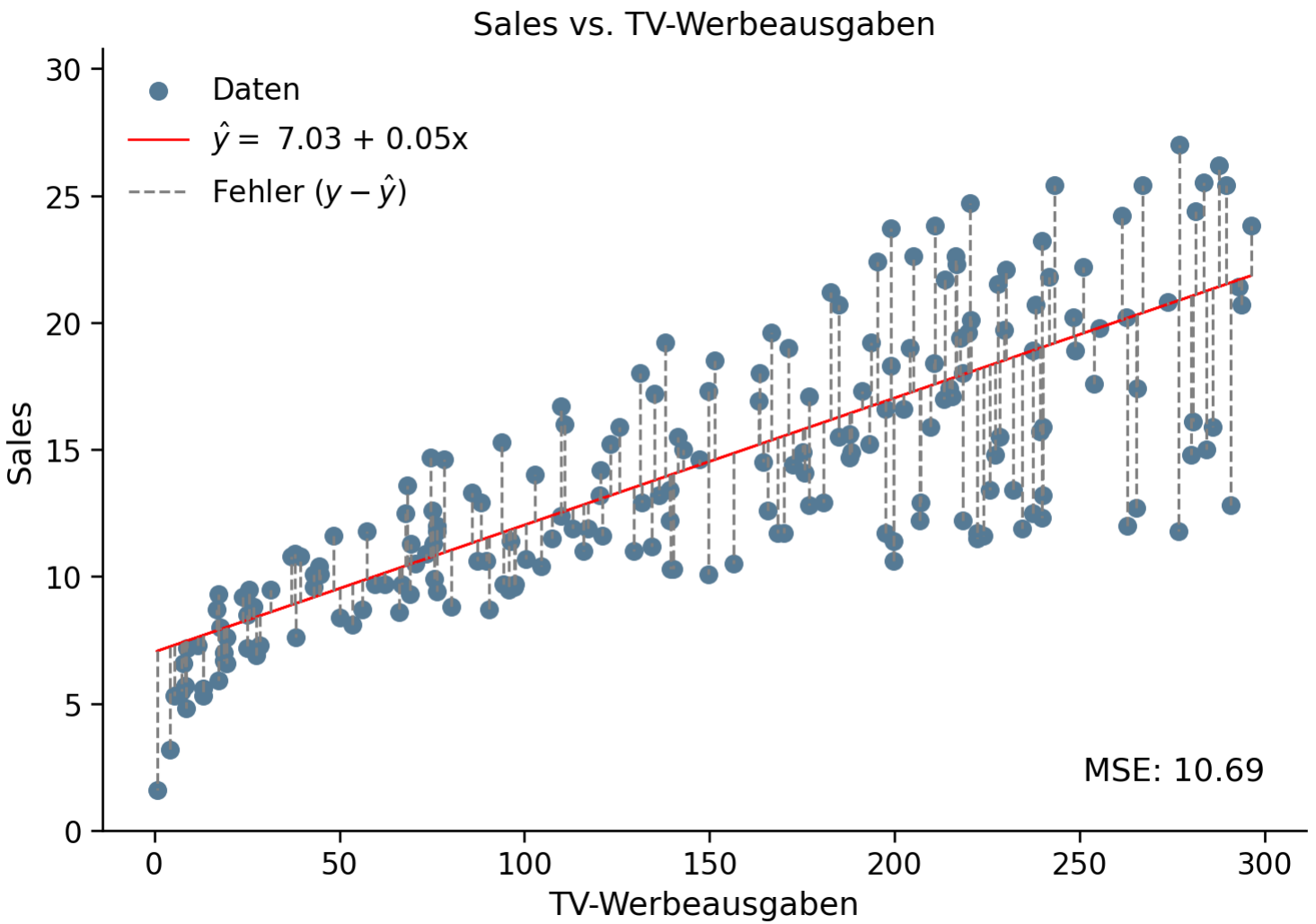
wobei e_i definiert ist als

$$e_i = y_i - \hat{y}_i$$

Einfaches Modell: analytische Lösung (cont'd)



Einfaches Modell: analytische Lösung (cont'd)



Einfaches Modell: analytische Lösung (cont'd)

Analytisch können wir dann β_0 und β_1 so ableiten, dass die RSS minimiert wird:

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}$$

Wir schreiben $\hat{\beta}_i$, um anzuzeigen, dass es sich um einen Schätzer handelt. Wir lassen es weg, wenn es aus dem Kontext klar ist.

Multiple Regression

Für die meisten Fragen benötigen wir mehr als eine unabhängige Variabel. Wenn dies der Fall ist, hat das Regressionsmodell die folgende Form:

$$\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_p x_p$$

Beispiel:
Wir könnten die Beziehung zwischen *Absatz* und Werbeausgaben unter Verwendung aller drei Medienarten, *Fernsehen*, *Zeitung* und *Radio*, beschreiben. Dies würde dann wie folgt beschrieben werden:

$$\text{Sales} = \beta_0 + \beta_1 \text{TV} + \beta_2 \text{newspaper} + \beta_3 \text{radio}$$

Multiple regression: analytical solution

Um für β_j zu lösen, gehen wir im Grunde genommen denselben Weg wie im einfachen Regressionsfall. Das Lösen des Gleichungssystems ist jedoch etwas komplexer. Es beinhaltet das Lösen eines Systems linearer Gleichungen der folgenden Form:

$$y = X\beta + \epsilon$$

wobei:

$$y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}, X = \begin{pmatrix} 1 & x_{11} & x_{12} & \dots & x_{1p} \\ 1 & x_{21} & x_{22} & \dots & x_{2p} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{n1} & x_{n2} & \dots & x_{np} \end{pmatrix}, \beta = \begin{pmatrix} \beta_0 \\ \vdots \\ \beta_p \end{pmatrix}, \epsilon = \begin{pmatrix} \epsilon_0 \\ \epsilon_1 \\ \vdots \\ \epsilon_n \end{pmatrix},$$

Ohne auf die Details der Mathematik einzugehen, kann diese Gleichung wie folgt gelöst werden:

$$\hat{\beta} = (X^T X)^{-1} X^T y = X^{-1} \cdot y$$

Beispiel: Sales vs. TV-Werbeausgaben

Ansatz 1: manuelle Implementierung der analytischen Lösung

```
1  import numpy as np
2  x = df["TV"]
3  xbar = np.mean(x)
4  y = df["sales"]
5  ybar = np.mean(y)
6
7  b1 = sum((x-xbar)*(y-ybar)) / sum((x - xbar)**2)
8  b0 = ybar - b1*xbar
9
10 print("b0:", b0, "und b1: ", b1)
```

```
b0: 7.032593549127705 und b1:  0.04753664043301969
```

Beispiel: Sales vs. TV-Werbeausgaben

Ansatz 2: Modul `statsmodels` nutzen

```python
import statsmodels.formula.api as smf

# Modell definieren und Daten übergeben
model = smf.ols("sales ~ TV", data=df)

# Modellschätzung (engl: "fitting")
model = model.fit()

paras = model.params # Modellparameter

print(paras)
```

```
Intercept    7.032594
TV           0.047537
dtype: float64
```

Beispiel: Sales vs. TV-Werbeausgaben, Newspaper & Radio

Modul `statsmodels` kann auch für multiple Regressionen genutzt werden:

```python
import statsmodels.formula.api as smf

# Modell definieren und Daten übergeben
model = smf.ols("sales ~ TV + newspaper + radio", data=df)

# Modellschätzung (engl: "fitting")
model = model.fit()

paras = model.params # Modellparameter

print(paras)
```

```
Intercept    2.938889
TV           0.045765
newspaper   -0.001037
radio        0.188530
dtype: float64
```

Was können wir mit einem geschätzten Modell machen?

1. Prognose:

Das endgültige Modell kann verwendet werden, um die Frage zu beantworten

- "Wenn mir X gegeben ist, wie wird dann y sein?"
- **Beispiel:** Ich beabsichtige, 920.000 USD für TV-Werbung auszugeben. Welchen Umsatz erwarten wir dann?

2. Diagnose und Schlussfolgerung:

Das endgültige Modell kann verwendet werden, um Fragen wie folgende zu beantworten:

- "Wie können wir die (kausale) Beziehung zwischen y und X quantifizieren?"
- **Beispiel:** Wenn ich die Ausgaben für TV-Werbung um 100.000 USD erhöhe, um wie viel wird mein Umsatz voraussichtlich steigen?

Quellen