Statistik1

Geradenmodelle 2

Sebastian Sauer

Einstieg

Lernziele

- Sie können Regressionsmodelle für Forschungsfragen mit binärer, nominaler und metrischer UV erläutern und in R anwenden.
- Sie können Interaktionseffekte in Regressionsmodellen erläutern und in R anwenden.
- Sie können den Anwendungszweck von Zentrieren und *z*-Transformationen zur besseren Interpretation von Regressionsmodellen erläutern und in R anwenden.

Benötigte R-Pakete & Daten

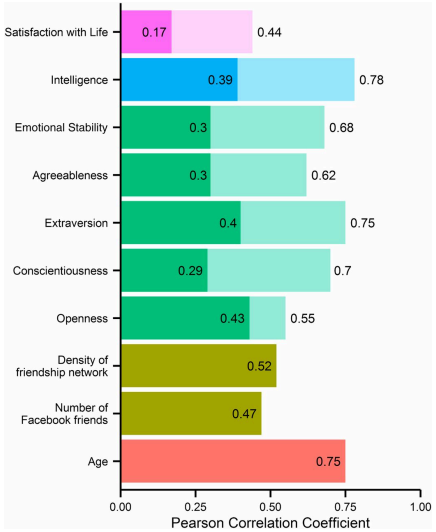
- Pakete: `tidyverse` (Wickham et al., 2019), `easystats` (Lüdecke et al., 2022), `yardstick` (Kuhn et al., 2024), optional `ggpubr` (Kassambara, 2023)
- Daten: die bekannten Mariokart-Daten *und* neu die **Wetterdaten** des Deutschen Wetterdienstes (DWD) (2025a, 2025b)
- Temperatur in Grad Celsius, Niederschlag (`precip`) in mm pro Quadratmeter

Forschungsbezug: Gläserne Kunden

Wie gut kann man Ihre Persönlichkeit vorhersagen?

Kosinski et al. (2013) zeigten, wie gut Persönlichkeitszüge anhand von Facebook-Likes vorhergesagt werden können — sexuelle Orientierung, Ethnie, Alter, Geschlecht, uvm.

Das eingesetzte Modell: ein *lineares Modell*, wie in diesem Kapitel behandelt.



Vorhersagegüte für numerische Merkmale, gemessen als Korrelation zwischen Vorhersage und tatsächlichem Wert (Kosinski et al., 2013)

Wetter in Deutschland

Metrische UV: Temperatur im Zeitverlauf

Forschungsfrage: Um wie viel ist die Temperatur in Deutschland pro Jahr gestiegen (letzte ca. 100 Jahre)?

```
1  lm_wetter1 <- lm(temp ~ year, data = wetter)
```

Laut Modell wurde es pro Jahr im Schnitt um 0.01 °C wärmer — pro Jahrzehnt 0.1 °C, pro Jahrhundert 1 °C.

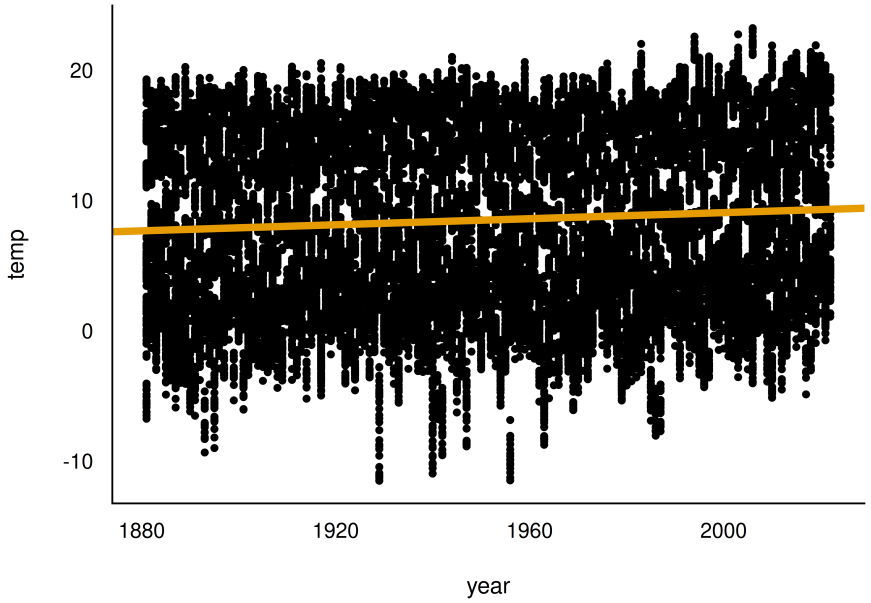
Punkt- und Bereichsschätzung

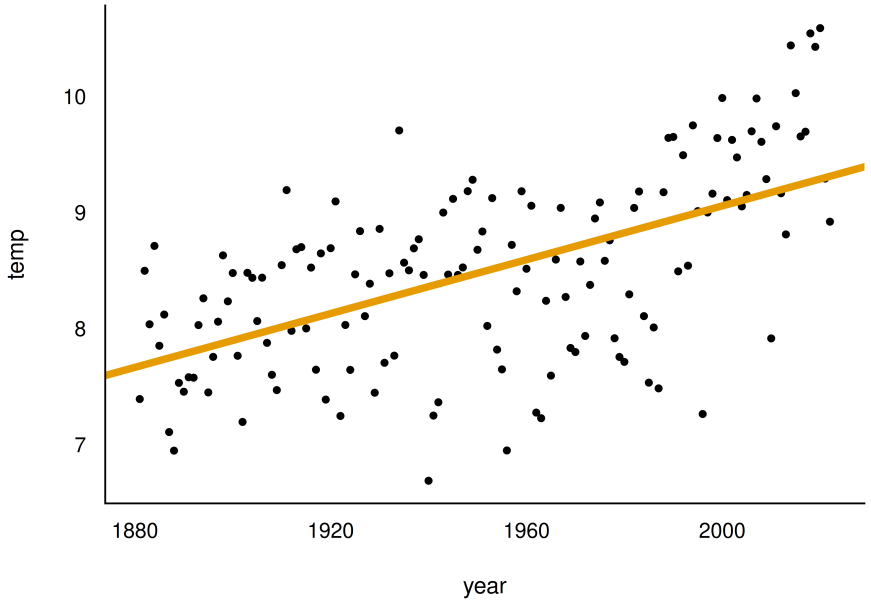
1. **Punktschätzung**: der “Best-Guess” für einen Koeffizienten in der Population (Spalte `Coefficient`)
2. **Bereichsschätzung** (Konfidenzintervall, CI): ein Bereich plausibler Werte, z. B. mit 95 % Sicherheit

> **Definition: Konfidenzintervall**
>
> Ein Konfidenzintervall gibt einen Schätzbereich plausibler Werte für einen Populationswert an, auf Basis der Schätzung, die uns die Stichprobe liefert.

Je schmaler das CI, desto genauer wird der Effekt geschätzt.

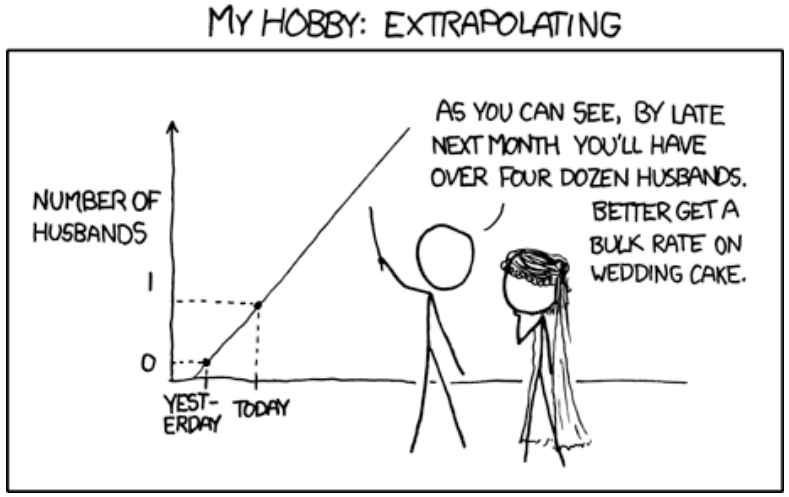
Temperaturverlauf visualisiert





UV zentrieren: Warum?

Der Achsenabschnitt (β_0) ist der Y-Wert an der Stelle $x = 0$. In den Wetterdaten wäre `year = 0` Christi Geburt — sinnlos zu extrapolieren!



Sinnvoller: einen Referenzwert festlegen, z. B. 1950 — oder gleich den Mittelwert der UV abziehen (*Zentrierung*).

Zentrierte Variable im Modell

```
wetter <- wetter |> mutate(year_c = year - mean(year))  # "c" wie centered
lm_wetter1_zentriert <- lm(temp ~ year_c, data = wetter)
```

- Die Steigung bleibt durch das Zentrieren **unverändert** — nur der Achsenabschnitt ändert sich.
- Das mittlere Jahr der Messreihe ist 1951: Im Jahr 1951 lag die mittlere Temperatur bei ca. 8.5 °C.
- Regressionsgleichung: `temp_pred = 8.49 + 0.01*year_c`

Das Modell erklärt aber nur wenig Varianz — die Jahreszeit spielt z. B. eine viel größere Rolle als die Jahreszahl.

Definition: Binäre Variable

Definition: Binäre Variable

Eine *binäre* UV, auch *Indikatorvariable* oder *Dummyvariable* genannt, hat nur zwei Ausprägungen: 0 und 1.

Beispiele: *weiblich* (0/1), oder *after_1950* (1 für Jahre nach 1950, sonst 0).

Forschungsfrage: War es in der zweiten Hälfte des 20. Jahrhunderts im Schnitt wärmer als davor?

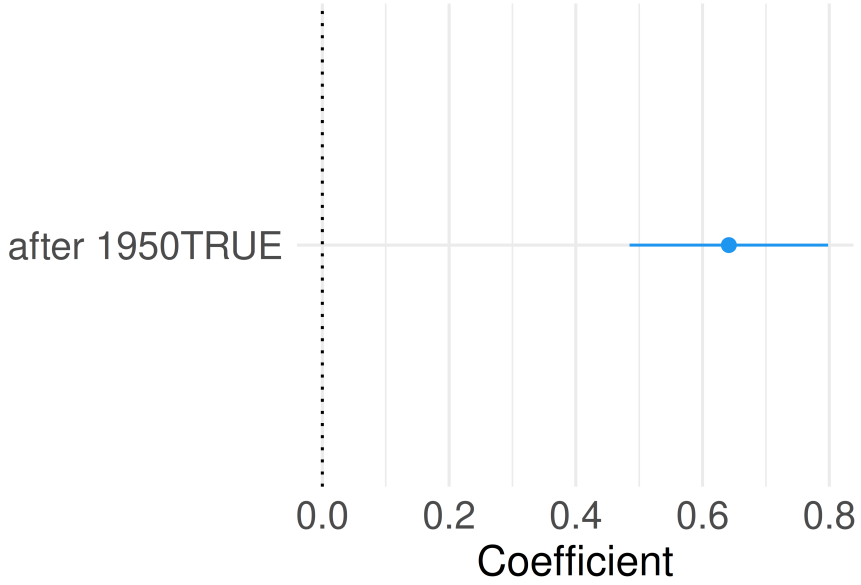
Binäre UV in R erzeugen

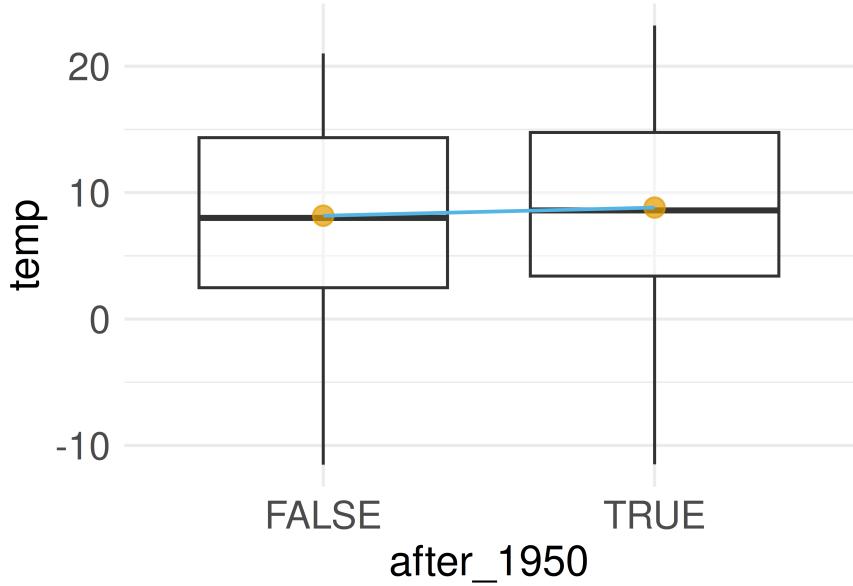
```r
year <- c(1940, 1950, 1960)
after_1950 <- year > 1950  # prüfe, ob das Jahr größer als 1950 ist
after_1950
```

```r
wetter <- wetter |> mutate(after_1950 = year > 1950)
lm_wetter_bin_uv <- lm(temp ~ after_1950, data = wetter)
```

TRUE/FALSE werden von R automatisch als 1/0 verstanden — eine logische Variable ist bereits binär.

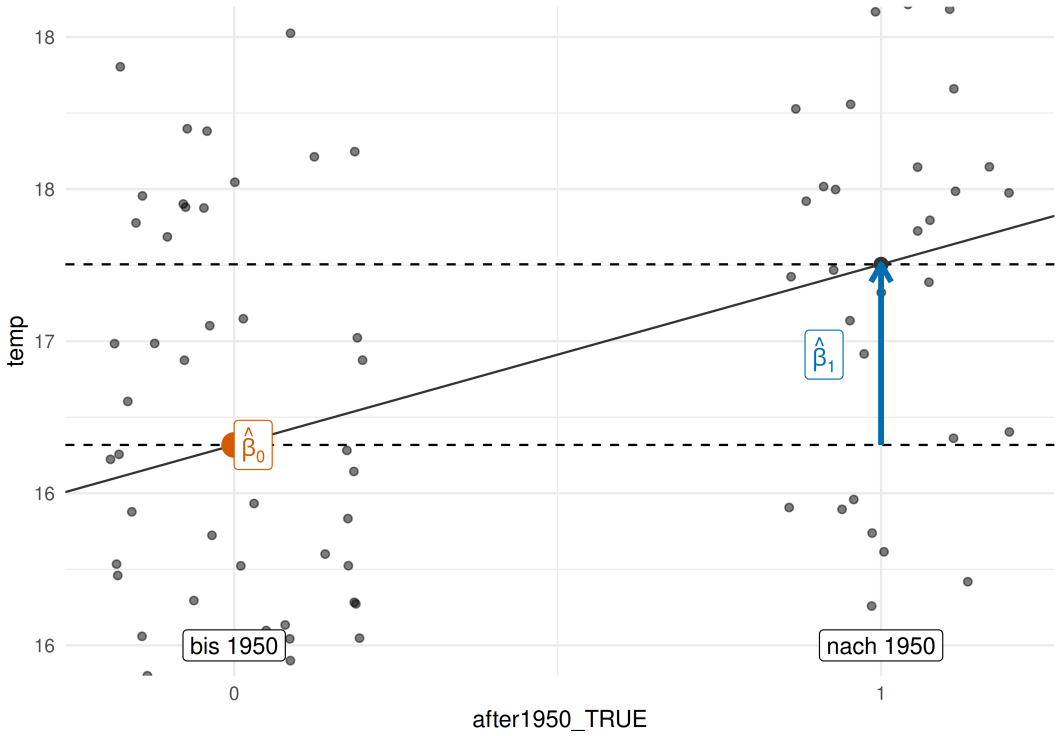
Modell mit binärer UV





Es wurde ein gutes halbes Grad wärmer nach 1950 — die Modellgüte ist aber gering.

Interpretation: binäre UV



1. Mittelwert der 1. Gruppe (bis 1950): Achsenabschnitt (β_0)
2. Mittelwert der 2. Gruppe (nach 1950): Achsenabschnitt (β_0) + Steigung (β_1)

Modellwerte konkret

Für die Modellwerte $\hat{y}$ gilt:

- Bis 1950: $\hat{y} = \beta_0 = 17.7$
- Nach 1950: $\hat{y} = \beta_0 + \beta_1 = 17.7 + 0.6 = 18.3$

β_1 lässt sich bei nominalen/binären UV als *Schalter* verstehen, bei metrischen UV als *Dimmer*.

Nominale UV: mehrstufig

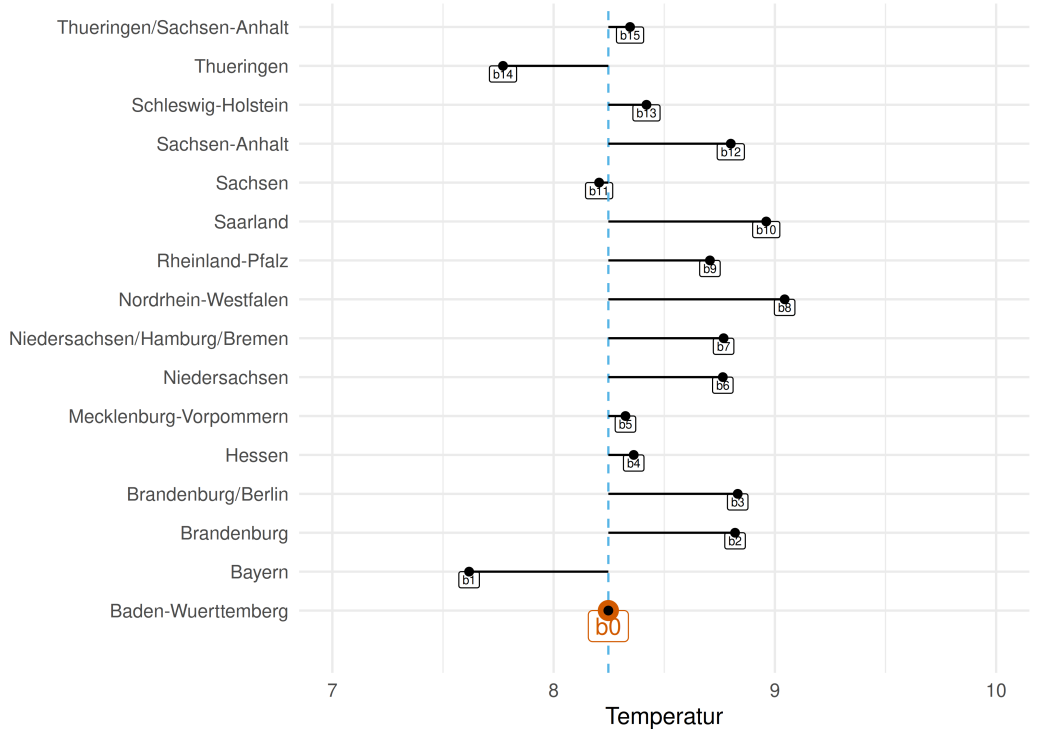
Forschungsfrage: Gibt es substanzielle Temperaturunterschiede zwischen den Bundesländern?

```
1  lm_wetter_region <- lm(temp ~ region, data = wetter)
```

Eine nominale UV mit k Stufen wird von `lm` in $k - 1$ binäre Variablen umgewandelt. Bei `character`-Variablen wählt R den alphabetisch ersten Wert als Referenzgruppe.

1. Mittelwert 1. Gruppe: Achsenabschnitt (β_0)
2. Mittelwert 2. Gruppe: β_0 + Steigung der 1. Gerade (β_1)
3. Mittelwert 3. Gruppe: β_0 + Steigung der 2. Gerade (β_2), usw.

Interpretation: nominale UV



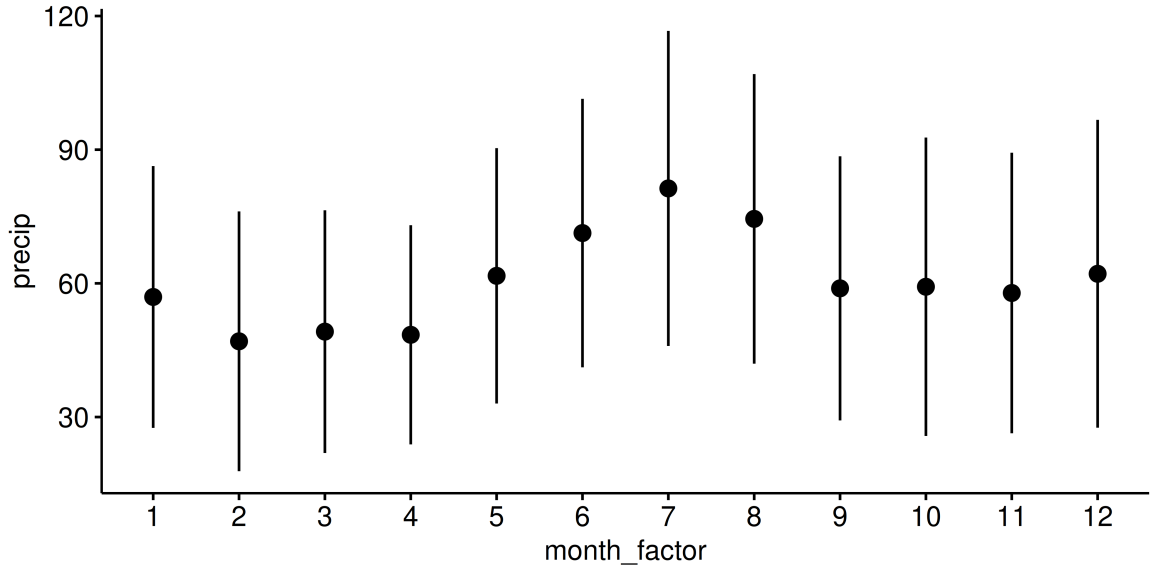
Da Baden-Württemberg alphabetisch das erste Bundesland ist, wird es als Referenzgruppe gewählt — ihr Mittelwert entspricht dem Achsenabschnitt.

Faktorvariablen

```
wetter <- wetter |> mutate(month_factor = factor(month))
lm_wetter_month_factor <- lm(precip ~ month_factor, data = wetter)
```

- `factor()` wandelt eine numerische Variable in eine nominalskalierte Variable um.
- Faktorstufen werden bei `factor`-Variablen mitgespeichert (bei `character` nicht) — auch nach dem Filtern.
- Referenzgruppe ändern: `relevel(month_factor, ref = "7")`

Niederschlag nach Monat



Referenzgruppe Januar; die 11 übrigen Monate werden im Modell als Unterschied zu Januar ausgegeben.

Definition: Multiple Regression

Definition: Multiple Regression

Eine multiple Regression beinhaltet mehr als eine X-Variable. Die Modellformel spezifiziert man so:

$$y \sim x_1 + x_2 + \ldots + x_n$$

Hat man mehrere UV, trennt man sie mit einem Plus-Zeichen — dieses hat *keine* arithmetische Funktion, sondern bedeutet “und noch folgende UV”.

Multiple Regression: Beispiel

```
lm_year_month <- lm(precip ~ year_c + month_factor,
                    data = wetter_month_1_7)
```

Interpretation der Koeffizienten:

1. Achsenabschnitt (β_0): Niederschlag im Referenzjahr 1951, Referenzmonat Januar — 57 mm
2. β_1 (`year_c`): +0.03 mm Niederschlag pro Jahr (im Referenzmonat)
3. β_2 (`month_factor7`): im Juli knapp 25 mm mehr als im Referenzmonat Januar

Regressionsgleichung: `precip_pred = 56.94 + 0.03*year_c + 24.37*month_factor_7`

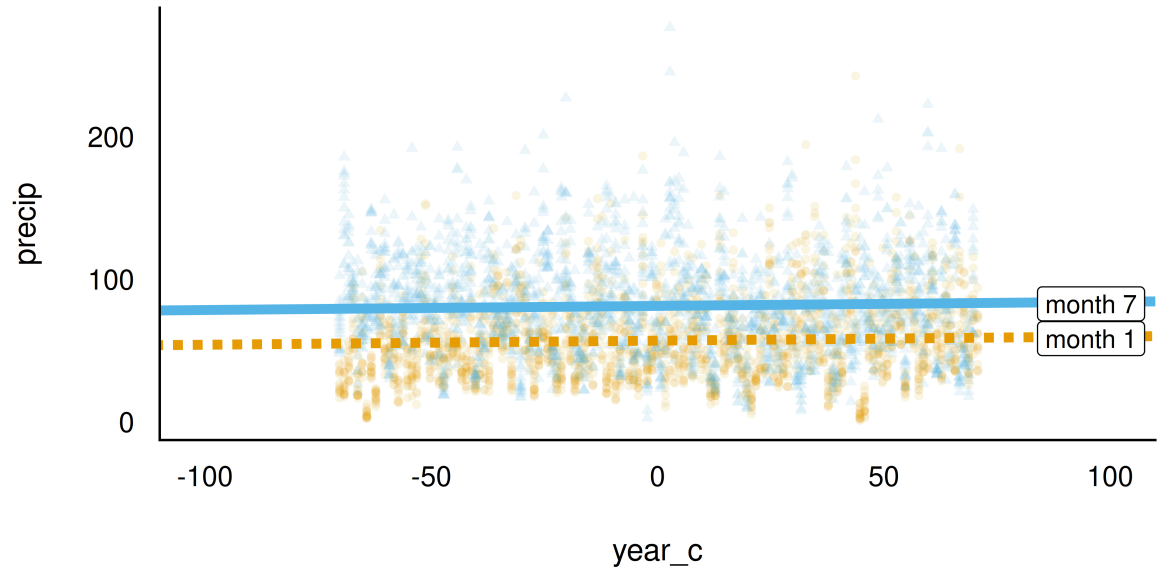
Vorhersage mit `predict`

Statt die Regressionsgleichung von Hand auszurechnen, lässt man R rechnen:

```
neue_daten <- tibble(year_c = 2020 - 1951,
                     month_factor = factor("7"))
predict(lm_year_month, newdata = neue_daten)
```

Alle Koeffizienten beziehen sich auf die AV *unter der Annahme, dass alle übrigen UV den Wert Null (bzw. Referenzwert) haben*.

Parallele Regressionsgeraden



Ein additives Modell (`y ~ x1 + x2`) zwingt die Regressionsgeraden der Gruppen zur Parallelität.

Definition: Interaktionseffekt

Definition: Interaktionseffekt

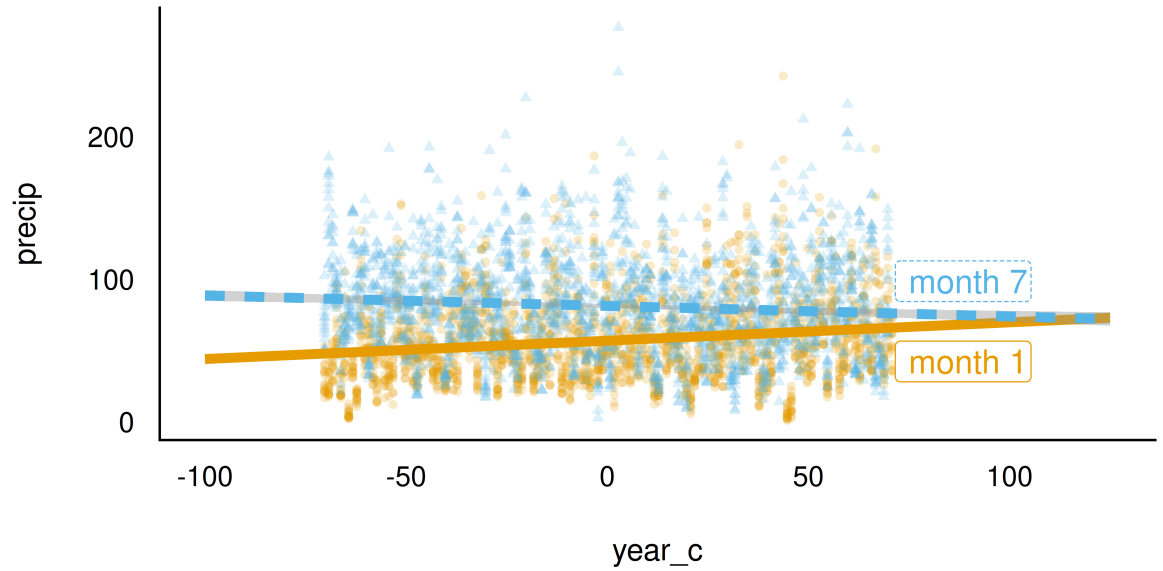
Einen Interaktionseffekt von x1 und x2 kennzeichnet man in R mit dem Doppelpunkt-Operator:

```
y ~ x1 + x2 + x1:x2
```

In Worten: y wird modelliert als eine Funktion von x1 und x2 *und* dem Interaktionseffekt von x1 mit x2.

```
1 lm_year_month_interaktion <- lm(
2   precip ~ year_c + month_factor + year_c:month_factor,
3   data = wetter_month_1_7)
```

Nicht-parallele Regressionsgeraden



Sind die Regressionsgeraden mehrerer Gruppen nicht parallel, liegt ein Interaktionseffekt vor.

Interaktion interpretieren

- Der Interaktionsterm (`year_c × month_factor[7]`) zeigt, wie *unterschiedlich* sich die Niederschlagsmenge zwischen den Monaten über die Jahre verändert.
- Pro Jahr ist die Zunahme im Juli um 0.20 mm geringer als im Referenzmonat Januar → für Juli insgesamt: `0.13 - (-0.20) = 0.07`
- Gleichung: `precip_pred = 56.91 + 0.13*year_c + 24.37*month_factor_7 - 0.20*year_c:month_factor_7`

Wichtig

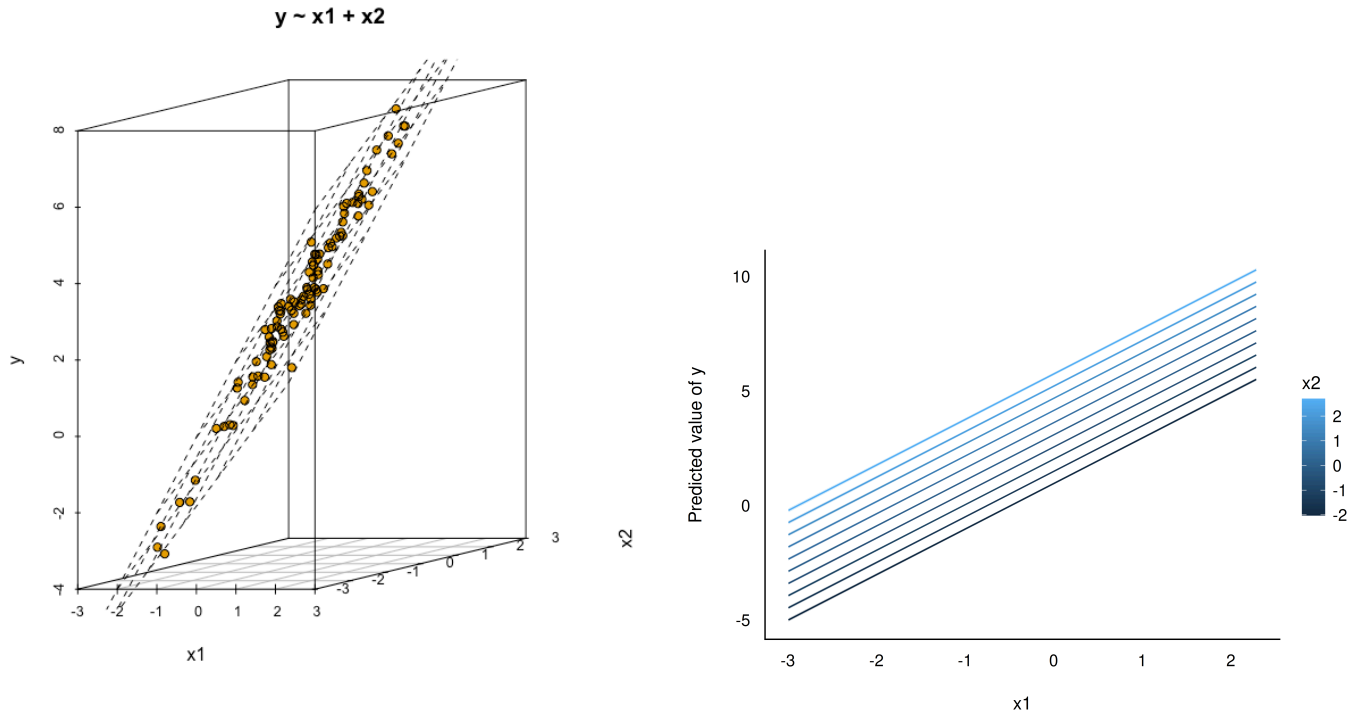
Der Achsenabschnitt gibt den Wert der AV an unter der Annahme, dass alle UV den Wert Null aufweisen.

Das R^2 steigt durch den Interaktionsterm hier nur geringfügig — im Zweifel gilt: so einfach wie möglich modellieren.

Modelle mit vielen UV

Zwei metrische UV: die Regressionsebene

Ein Modell mit zwei metrischen UV lässt sich im 3D-Raum als *Regressionsebene* visualisieren.



Viele UV ins Modell?

Grundsätzlich können viele UV in ein Modell aufgenommen werden — Richtlinien (Gelman et al., 2021):

1. Variablen aufnehmen, die vermutlich Ursachen der Zielvariable sind.
2. Bei UV mit starken Effekten: auch deren Interaktionseffekte erwägen.
3. Präzise geschätzte UV (schmales CI) eher im Modell belassen.

Geht es um Theorieprüfung statt Modellgüte: nur die von der Theorie geforderten UV aufnehmen.

Fallbeispiel zur Prognose

Prognose des Verkaufspreises

Auftrag: den Verkaufspreis von Mariokart-Spielen möglichst exakt vorhersagen.

Vorgehen: Datensatz zufällig in **Train-Sample** (70 %, zum Modellieren) und **Test-Sample** (30 %, zur Güteprüfung) aufteilen.

Modell “all-in”: alle Variablen rein?

```
lm_allin <- lm(total_pr ~ ., data = mariokart_train)
r2(lm_allin)  # fast perfekte Modellgüte!
```

Der Punkt (`.`) steht für “alle Variablen außer `total_pr`”.

Vorsicht

Zu gut, um wahr zu sein — und es *ist* zu gut, um wahr zu sein: Das Modell hat einfach den Auktionstitel `Title` auswendig gelernt.

Idiografische Informationen wie Namen/Titel helfen nicht, allgemeine Muster zu erkennen.

Vorhersagefehler bei neuen Titeln

```
predict(lm_allin, newdata = mariokart_test)
# Fehler!
```

Kommt im Test-Sample eine Ausprägung von `title` vor, die im Train-Sample nicht existierte, scheitert `predict`.

Nominalskalierte UV mit vielen Ausprägungen sind problematisch und führen oft zu **Overfitting**.

Modell “all-in” ohne Titelspalte

```
mariokart_train2 <- mariokart_train |> select(-c(title, id))
lm_allin_no_title_no_id <- lm(total_pr ~ ., data = mariokart_train2)
r2(lm_allin_no_title_no_id)  # ordentliches R²
```

Vorhersagegüte im Test-Sample (mit dem Paket `yardstick`):

```
yardstick::mae(data = mariokart_test, truth = total_pr,
               estimate = lm_allin_predictions)
yardstick::rmse(data = mariokart_test, truth = total_pr,
               estimate = lm_allin_predictions)
```

Mittlerer Vorhersagefehler (RMSE): ca. 13 Euro. R^2 im Test-Sample: nur 17 %.

Overfitting

Die Modellgüte im Test-Sample ist (leider) oft *geringer* als im Train-Sample — die Trainings-Güte ist mitunter übermäßig optimistisch. Dieses Phänomen heißt **Overfitting** (Gelman et al., 2021).

Deshalb: vor dem Ausliefern von Vorhersagen die Modellgüte immer an einem *neuen* Datensatz (Test-Sample) prüfen.

`rsq()` (aus `yardstick`) berechnet die Modellgüte für beliebige Vorhersagen — im Gegensatz zu `r2()`, das nur im Train-Sample funktioniert.

Praxisbezug

Kundenabwanderung vorhersagen

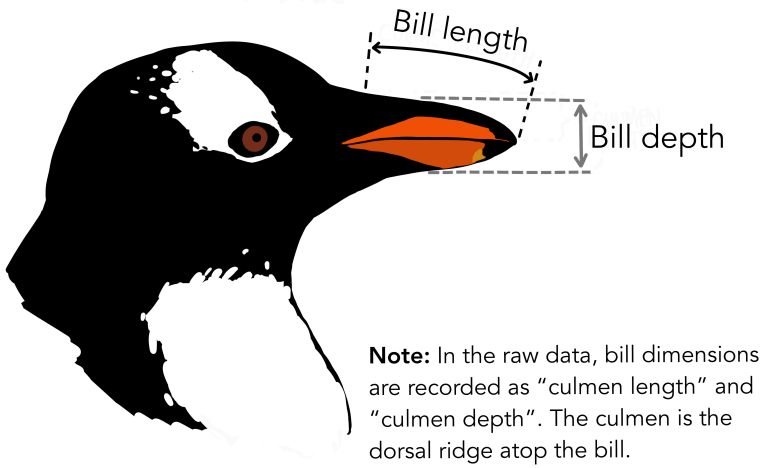
Ein Anwendungsbeispiel moderner Datenanalyse: die Vorhersage, welche Kunden “abwanderungsgefährdet” sind (*customer churn*).

Lalwani et al. (2022) nutzten u. a. lineare Regression zur Vorhersage abgewanderter Kunden und berichten eine Genauigkeit von über 80 % im besten Modell.

Wie man mit Statistik lügt

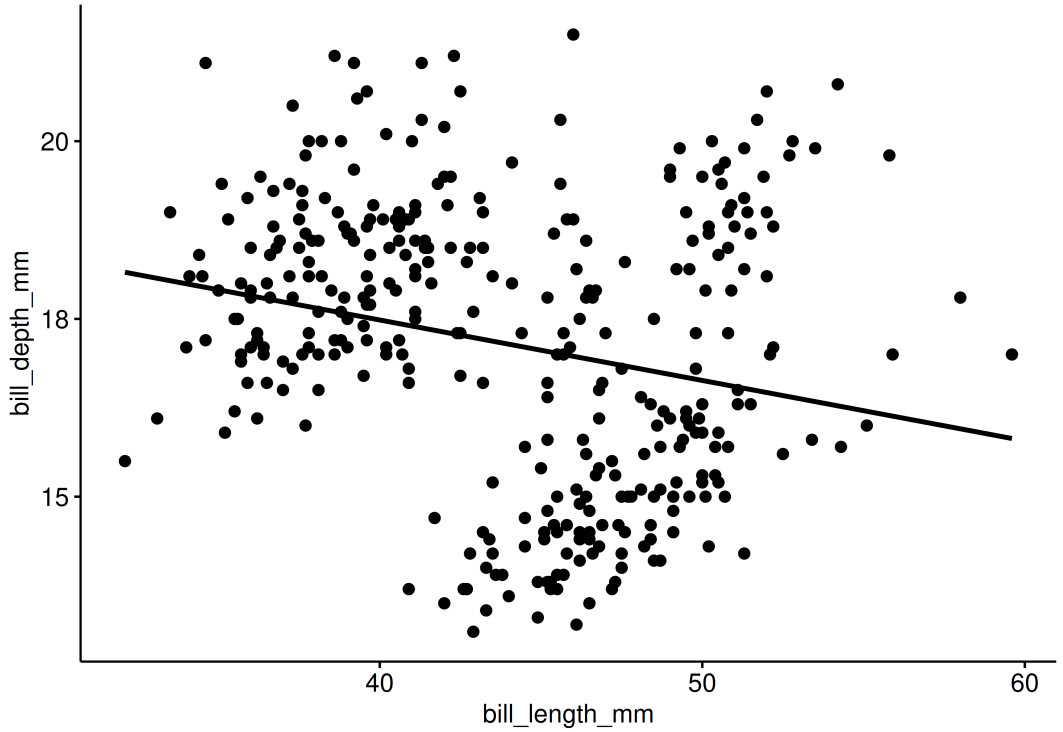
Pinguine drehen auf

Ein Forscherteam untersucht Schnabellänge und Schnabeltiefe bei $n = 344$ Pinguinen der Palmer Station, Antarktis.



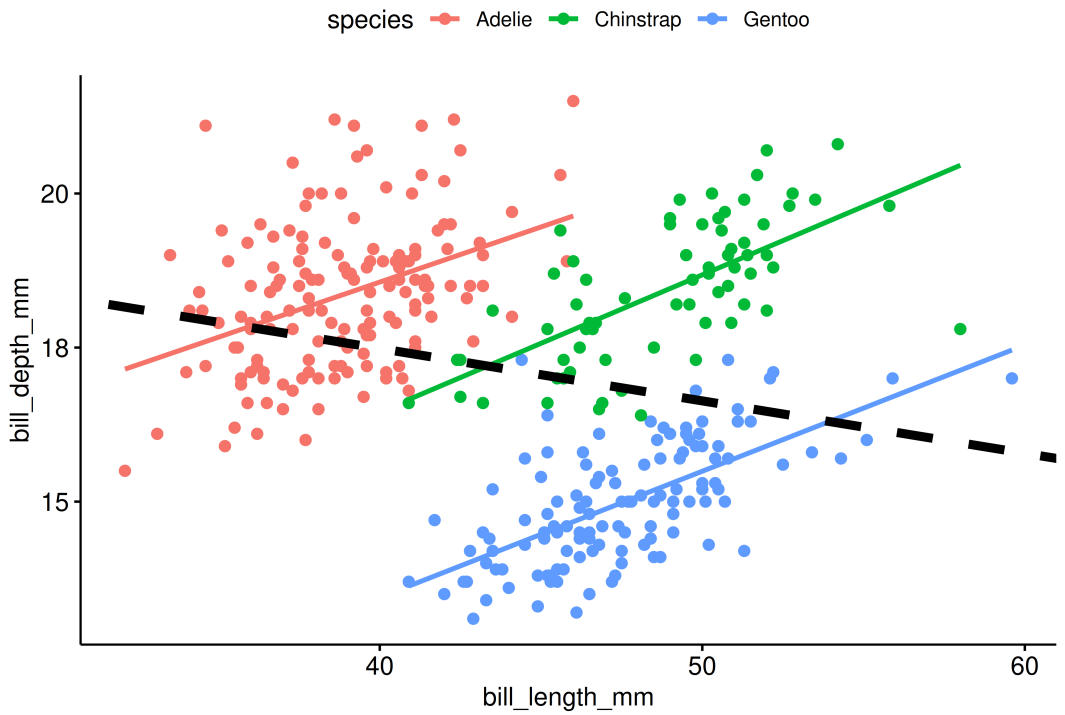
Schnabellänge und Schnabeltiefe (Horst, 2024)

Analyse 1: Gesamtdaten



Klarer Fall: ein *negativer* Zusammenhang. Nobelpreis-verdächtig — oder?

Analyse 2: Aufteilung in Arten

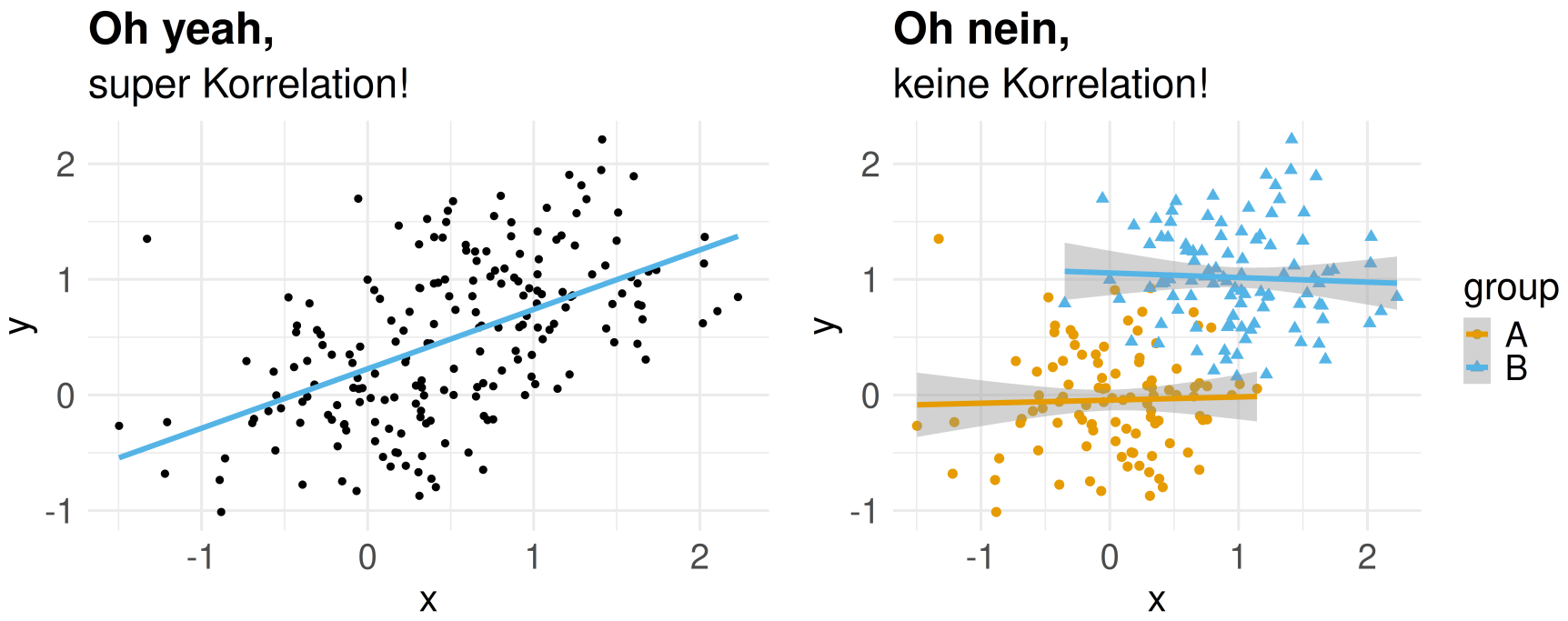


Gleiche Daten, gegenteiliges Ergebnis! Ohne Hintergrundwissen ist nicht entscheidbar, welche Analyse “richtig” ist.

Kausale Interpretation: Vorsicht

Wichtig

Interpretiere Modellkoeffizienten nur kausal, wenn du ein Kausalmodell hast.



Kippt das Vorzeichen eines Effekts durchs Hinzufügen einer Variable, spricht man von **Simpsons Paradox** (Gelman et al., 2021).

Wie man mit Statistik lügt

Deskriptiv statt kausal

Kennt man die kausale Struktur nicht, sollte man Modellkoeffizienten nur *deskriptiv* interpretieren — z. B. “Wo es viele Störche gibt, gibt es auch viele Babys” (Matthews, 2000).

Unser Gehirn ist auf kausales Denken geprägt: Deskriptive Befunde werden daher immer wieder unzulässig kausal interpretiert.

Was war noch mal das Erfolgsgeheimnis?

Dranbleiben zahlt sich aus

Wenn Sie an der Statistik dranbleiben, wird sich der Erfolg einstellen.

Was war noch mal das Erfolgsgeheimnis?

Zusammenfassung

- **Metrische UV**: Achsenabschnitt oft sinnlos ohne *Zentrierung* der UV
- **Binäre/nominale UV**: β_0 = Mittelwert der Referenzgruppe, β_1, β_2, . . . = Differenz zu weiteren Gruppen
- **Multiple Regression**: mehrere UV, additiv verbunden mit `+`
- **Interaktion** (`x1:x2`): Effekt einer UV hängt vom Wert einer anderen UV ab — Regressionsgeraden nicht mehr parallel
- **Train-/Test-Sample**: Modellgüte immer an neuen Daten prüfen — Vorsicht vor **Overfitting**
- **Kausale Interpretation** von Koeffizienten nur mit einem Kausalmodell — sonst droht **Simpsons Paradox**

Literaturhinweise

- Gelman et al. (2021) — “Regression und andere Geschichten”, anspruchsvoll, aber sehr empfehlenswert
- Sauer (2019) — einsteigerfreundliche Alternative
- McElreath (2020) — sehr empfehlenswerte Lektüre

Referenzen

Deutscher Wetterdienst. (2025a). *Regional averages DE, monthly air temperature mean*. https://opendata.dwd.de/climate_environment/CDC/regional_averages_DE/monthly/air_temperature_mean/.

Deutscher Wetterdienst. (2025b). *Regional averages DE, monthly precipitation mean*. https://opendata.dwd.de/climate_environment/CDC/regional_averages_DE/monthly/precipitation/.

Gelman, A., Hill, J., & Vehtari, A. (2021). *Regression and Other Stories*. Cambridge University Press.

Horst, A. (2024). *Statistics Artwork* [Artwork]. https://allisonhorst.com/

imgflip. (2024). *Imageflip Meme* [Artwork]. https://imgflip.com

Kassambara, A. (2023). *ggpubr: 'ggplot2' Based Publication Ready Plots*. https://CRAN.R-project.org/package=ggpubr

Kosinski, M., Stillwell, D., & Graepel, T. (2013). Private Traits and Attributes Are Predictable from Digital Records of Human Behavior. *Proceedings of the National Academy of Sciences*, *110*(15), 5802–5805. https://doi.org/10.1073/pnas.1218772110

Kuhn, M., Vaughan, D., & Hvitfeldt, E. (2024). *yardstick: Tidy Characterizations of Model Performance*. https://CRAN.R-project.org/package=yardstick

Lalwani, P., Mishra, M. K., Chadha, J. S., & Sethi, P. (2022). Customer Churn Prediction System: A Machine Learning Approach. *Computing*, *104*(2), 271–294. https://doi.org/10.1007/s00607-021-00908-y

Lüdecke, D., Ben-Shachar, M. S., Patil, I., Wiernik, B. M., Bacher, E., Thériault, R., & Makowski, D. (2022). easystats: Framework for Easy Statistical Modeling, Visualization, and Reporting. *CRAN*. https://doi.org/10.32614/CRAN.package.easystats

Matthews, R. (2000). Storks Deliver Babies (P= 0.008). *Teaching Statistics*, *22*(2), 36–38. https://doi.org/10.1111/1467-9639.00013

McElreath, R. (2020). *Statistical Rethinking: A Bayesian Course with Examples in R and Stan* (2. Aufl.). Taylor and Francis, CRC Press.

Sauer, S. (2019). *Moderne Datenanalyse mit R: Daten einlesen, aufbereiten, visualisieren und modellieren*. Springer. https://www.springer.com/de/book/9783658215866

Wickham, H., Averick, M., Bryan, J., Chang, W., McGowan, L. D., François, R., Grolemund, G., Hayes, A., Henry, L., Hester, J., Kuhn, M., Pedersen, T. L., Miller, E., Bache, S. M., Müller, K., Ooms, J., Robinson, D., Seidel, D. P., Spinu, V., … Yutani, H. (2019). Welcome to the tidyverse. *Journal of Open Source Software*, *4*(43), 1686. https://doi.org/10.21105/joss.01686