

Lineare Regression

Richard Hunger

24. Oktober 2024

Inhalt

- Einführung für den Zweivariablen-Fall
- Durchführung in R
- Exkurs: Wahrscheinlichkeitsverteilungen
- Exkurs: Logik von Hypothesentests
- Prüfung der Voraussetzungen
- Bewertung der Modellgüte
- Vorhersage (Prädiktion)
- Weitere Themen
 - Standardisierte Koeffizienten
 - Fallzahlsschätzung
 - Fehlende Werte
- Quiz & Übungen

Grundprinzip

- **Regression:** “Zurückführen” der Ausprägung einer Variablen auf die Ausprägungen einer oder mehrerer andere Variablen
- (mindestens) 2 Variablen:
 - *abhängige* Variable (AV)
 - Regressand, endogene Variable, Kriterium, zu erklärende Variable
 - nur metrisches Skalenniveau
 - *aber nicht zwingend normalverteilt!*
 - *unabhängige* Variable (UV)
 - Regressor, exogene Variable, Prädiktor, erklärende Variable
 - metrisches Skalenniveau
 - kategoriales Skalenniveau
- **Ziel:** quantitative Beschreibung des Zusammenhangs von Variablen
- **Ergebnis:** Mathematisch Beschreibung des Zusammenhangs in Form einer Gleichung

einfache lineare Regression

- *einfach*: eine UV
- *linear*: Beziehung der Regressionskoeffizienten ist linear
 - die UV jedoch nicht zwingend!
- *Regression*: Zurückführen von AV auf UV
- Beispiele:
 - Klausurpunktzahl ← Lerndauer
 - Kraftstoffverbrauch ← Leistung

Regressionsgerade

Allgemeine Form

- Mathematisch Beschreibung des Zusammenhangs in Form einer Gleichung
- Allgemeine Form: $\hat{Y} = b_0 + b_1 \times X$
 - Y ... abhängige Variable
 - b_0 ... Konstante, *Intercept*
 - b_1 ... Regressionskoeffizient, Steigung, *Slope*
 - X ... unabhängige Variable
- Details zu den Symbolen:
 - $\hat{\bullet}$... mit "Hut", ein Wert ist geschätzt
 - b_{\bullet} ... konkreter, aus Stichprobe berechneter Wert
 - β_{\bullet} ... Wert in der Grundgesamtheit (wahrer Wert, idR unbekannt)
 - damit: $b_{\bullet} = \hat{\beta}_{\bullet}$
 - Variable (X) vs. einzerner Wert (x bzw. x_i)

Regressionsgerade

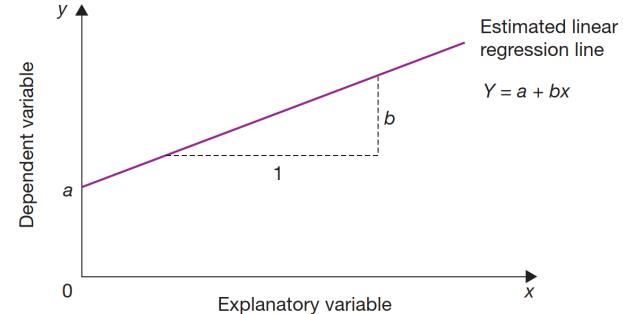
Bedeutung

- = Kern der Regressionsanalyse
- Schätzung der unbekannten Modellparameter: b_0 und b_1
- Regressionsgerade ermöglicht eine Vorhersage (Prädiktion)
 - für "neue" Beobachtungswerte
 - für bisher nicht beobachtete Zwischenwerte (Interpolation)
- Rückschluss auf die Bedeutung der UVs über Regressionskoeffizienten auf die AV (Inferenz)

Regressionsgerade

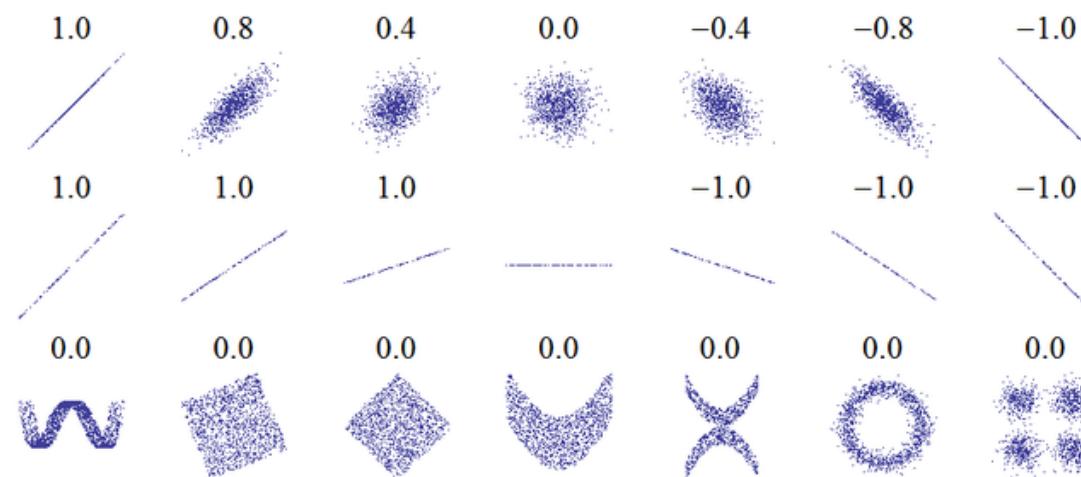
Interpretation

- Interpretation:
- b_0 = y-Wert an der Stelle $x = 0$
 - Schnittpunkt der RG mit y-Achse
 - nur interpretierbar, wenn $x = 0$ einen validen Wert darstellt
 - b_0 damit idR nicht interpretierbar
- b_1 = Veränderung des y-Wertes wenn sich x um genau 1 Einheit ändert
 - bzw. ändert sich X um 1 Einheit, ändert sich y um b_1 Einheiten
 - $b_1 > 0$... steigende Regressionsgerade
 - $b_1 < 0$... fallende Regressionsgerade
 - ↗ Bezug zur Korrelation



Bezug zur Korrelation

- 2 Variablen
- Korrelation = quantitative Beschreibung der Enge des Zusammenhangs zwischen zwei numerischen Variablen



Regressionsgerade

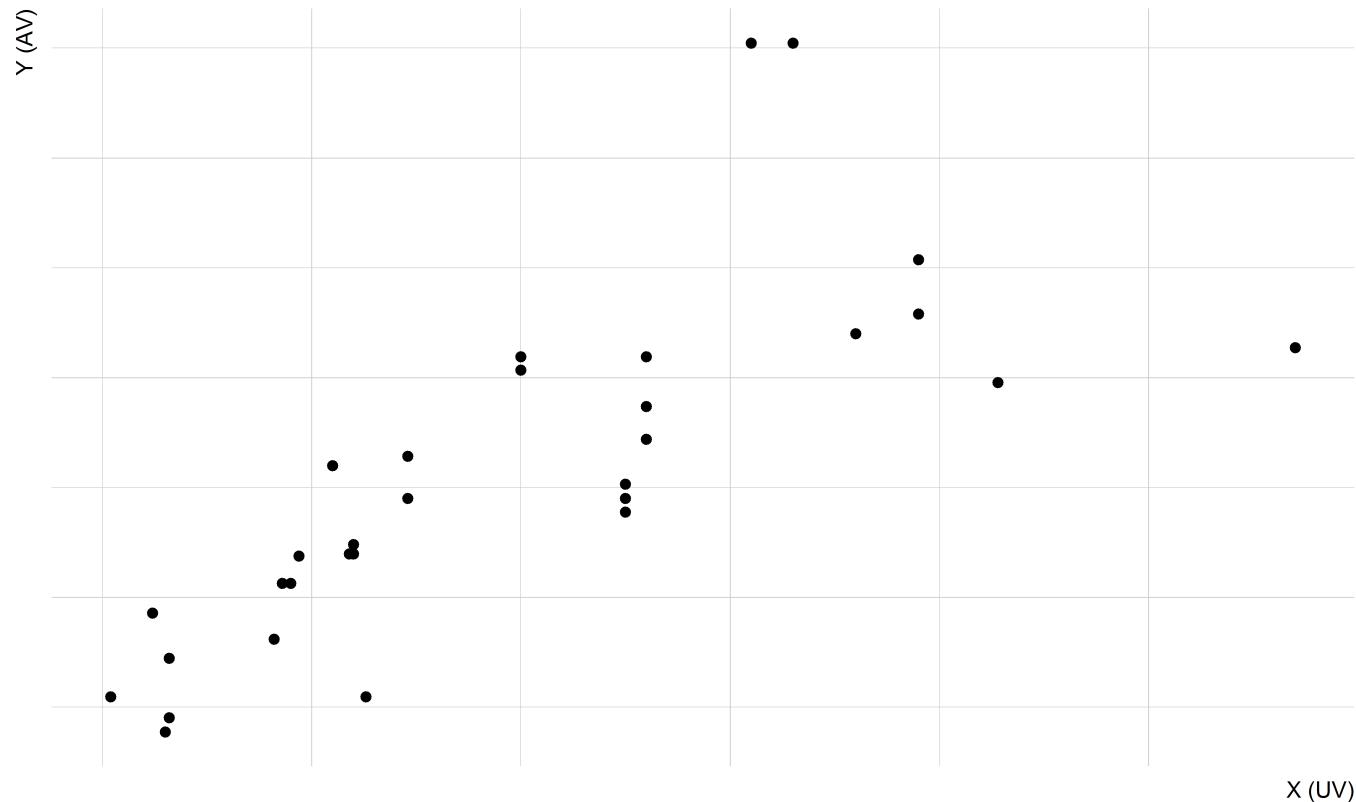
Bestimmung

- Ermittlung der Regressionsgerade (RG):
 - graphisch: Streudiagramm
 - Bestimmung einer Geraden, dass die Abweichungen minimal werden
 - Annäherung einer mathematischen Funktion an Datenpunkte mit dem Ziel möglichst minimaler Abweichung
 - mathematisch: *Methode der kleinsten Quadrate* (ordinary least squares, OLS)

Regressionsgerade

OLS

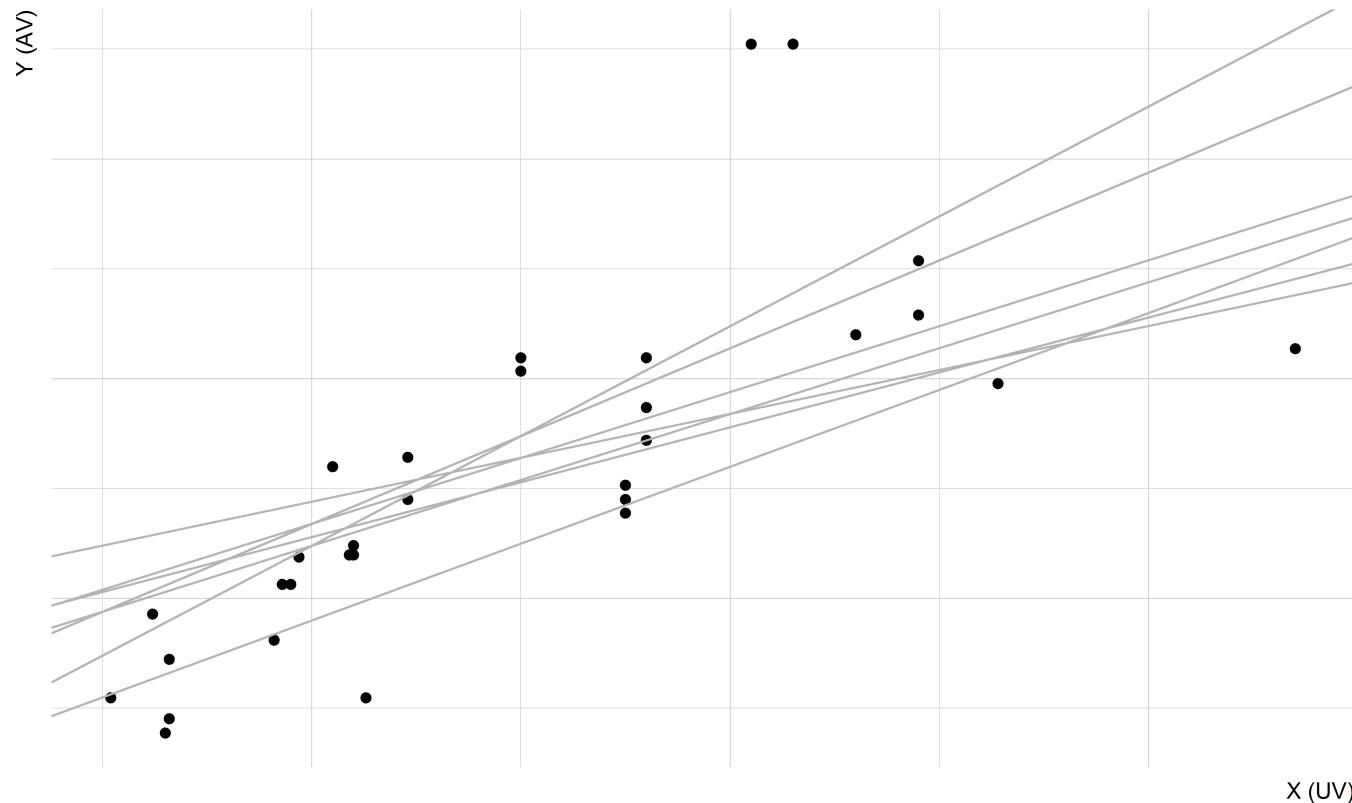
- Streudiagramm:



Regressionsgerade

OLS

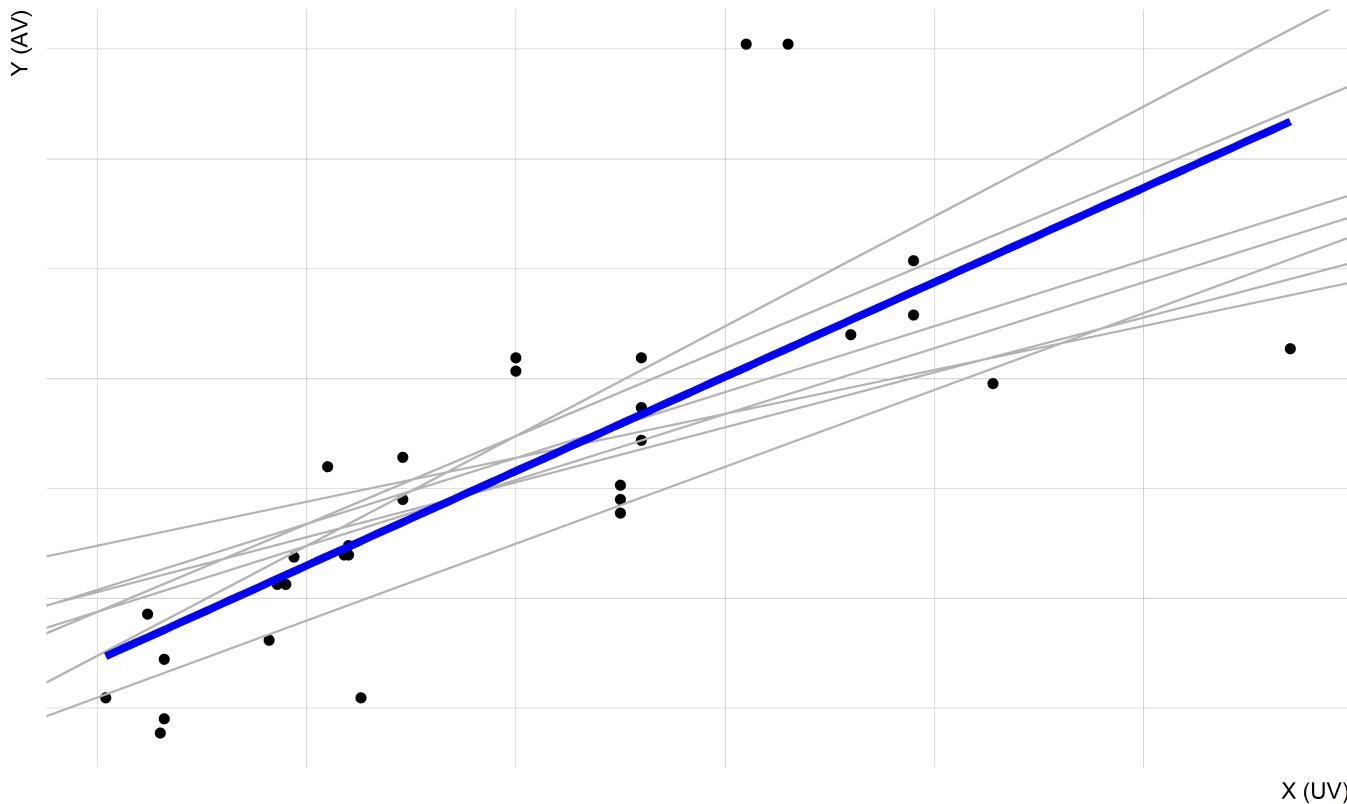
- mögliche Geraden:



Regressionsgerade

OLS

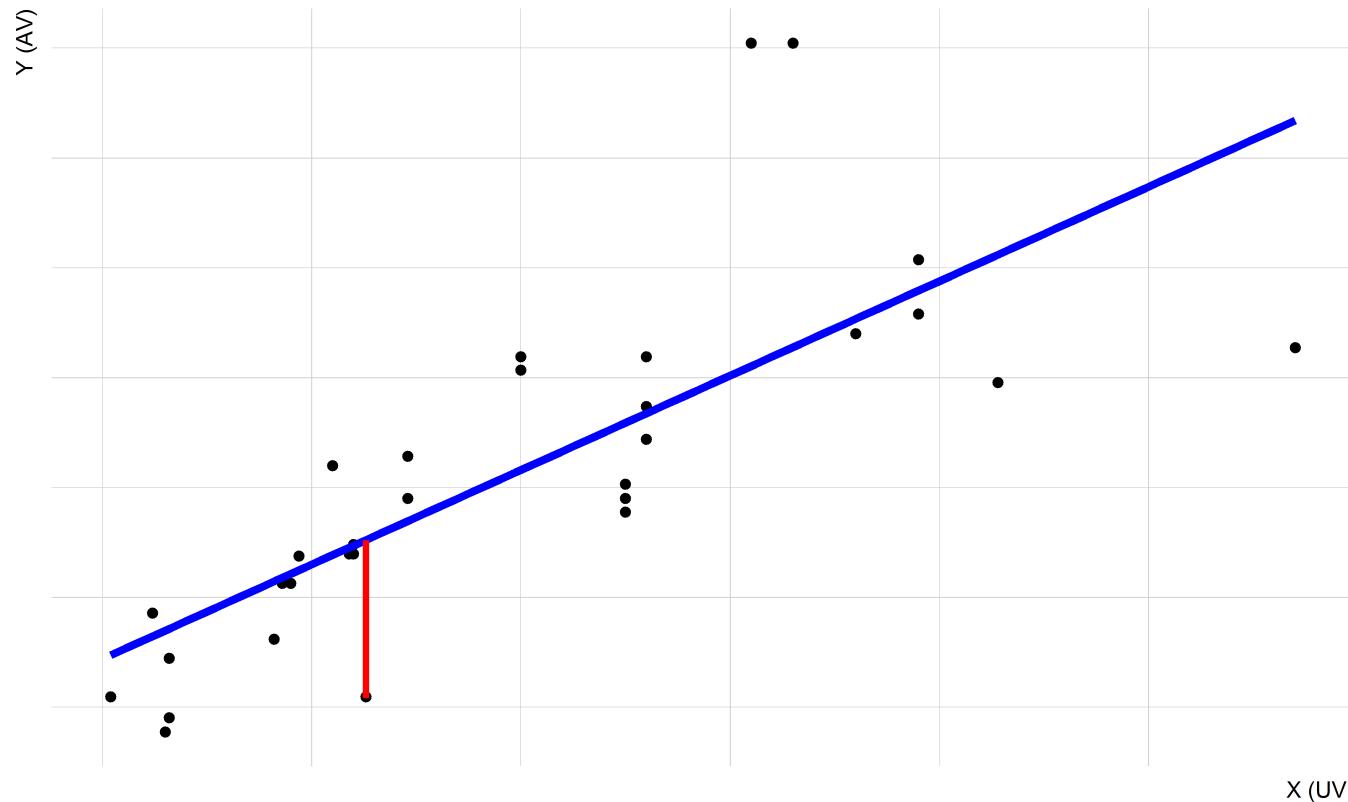
- DIE eine Gerade:



Regressionsgerade

OLS

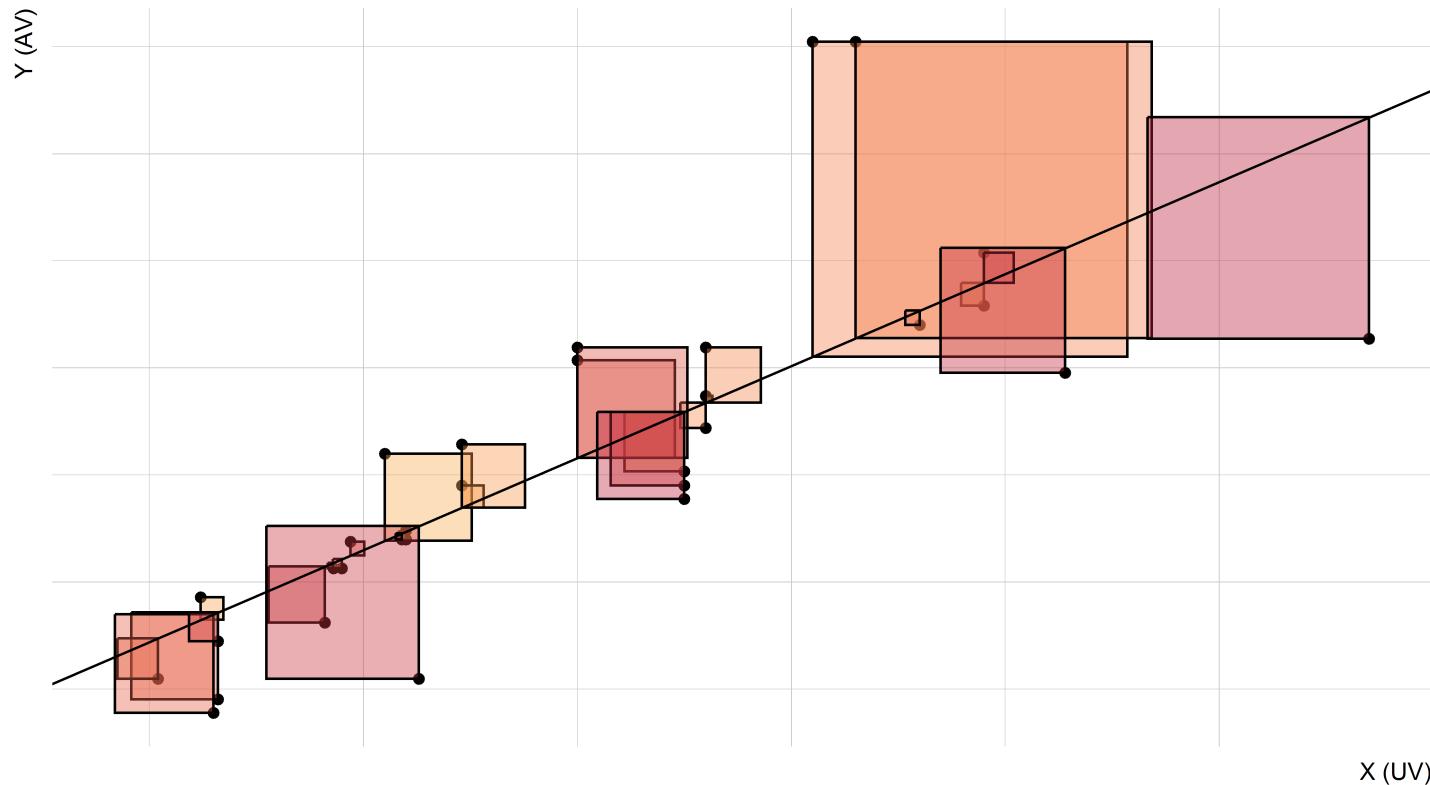
- Fehler/Abweichung/Residuum:



Regressionsgerade

OLS

- Minimierung der Abweichungsquadrate:



Regressionsgerade

OLS

- Berechnung:

- $b_1 = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})^2} = \frac{s_{xy}}{s_x^2}$

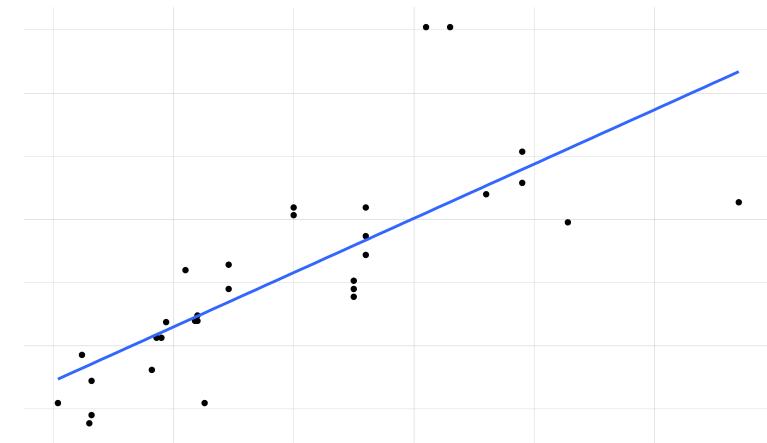
- $b_0 = \bar{y} - b_1 \times \bar{x}$

- → RG verläuft immer durch \bar{x} und \bar{y}

- Kovarianz:

- $\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y}) = s_{xy}$

- ...gemeinsame Varianz (Streuung) der Merkmale

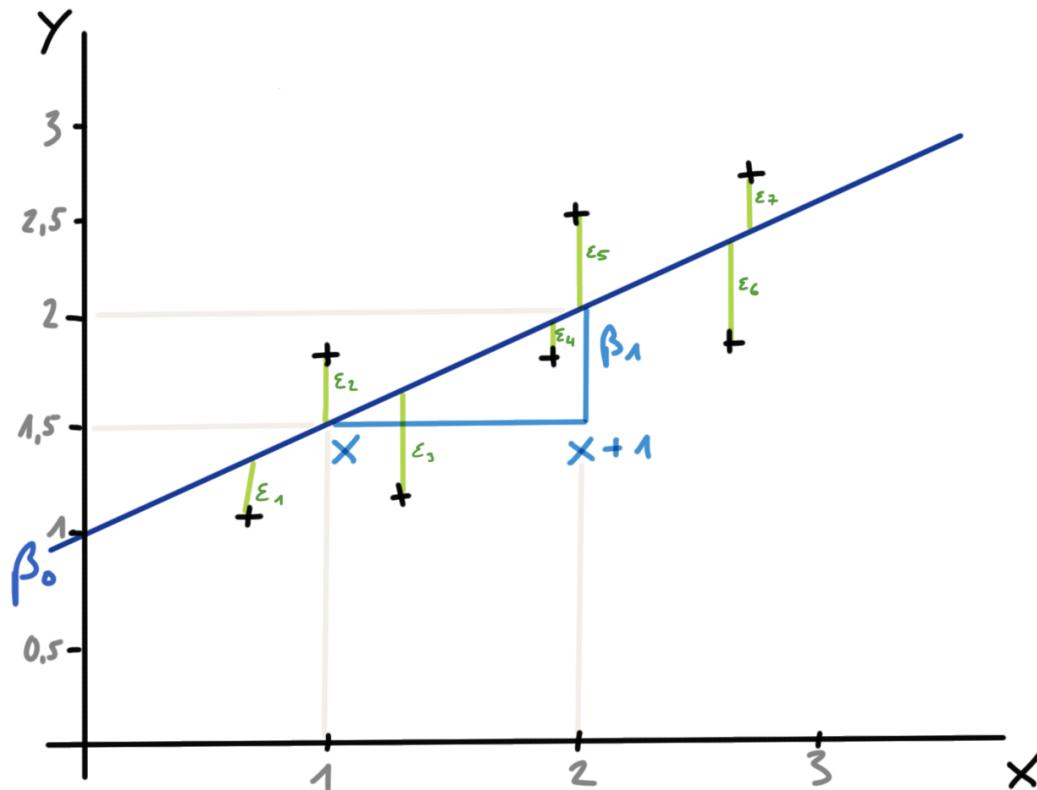


Einschränkungen

- "All models are wrong! But some are useful." - George Box
- Problem:
 - keine Beziehung von Variablen ist tatsächlich linear
 - niemals sind alle Einflussfaktoren auf die AV bekannt (oder messbar)
 - das wäre eine deterministisches Modell (á la physikalische Umrechnung)
- ⚠ Modellvorhersagen weichen von wahren Wert ab
- Regression ergibt ein *probabilistisches* Modell
 - $\hat{Y} = b_0 + b_1 X$... Modellvorhersage, Schätzwert ("mit Dach")
 - $Y = b_0 + b_1 X + \epsilon$... wahrer Wert ("ohne Dach"), aber mit Fehler
 - ϵ ... Fehlerterm, Abweichung, *Residuum*
 - $Y = \hat{Y} + \epsilon$
 - $\epsilon = Y - \hat{Y}$
- **Residuum:** beschreibt die Abweichung der Modellvorhersage vom beobachteten Wert

Residuum

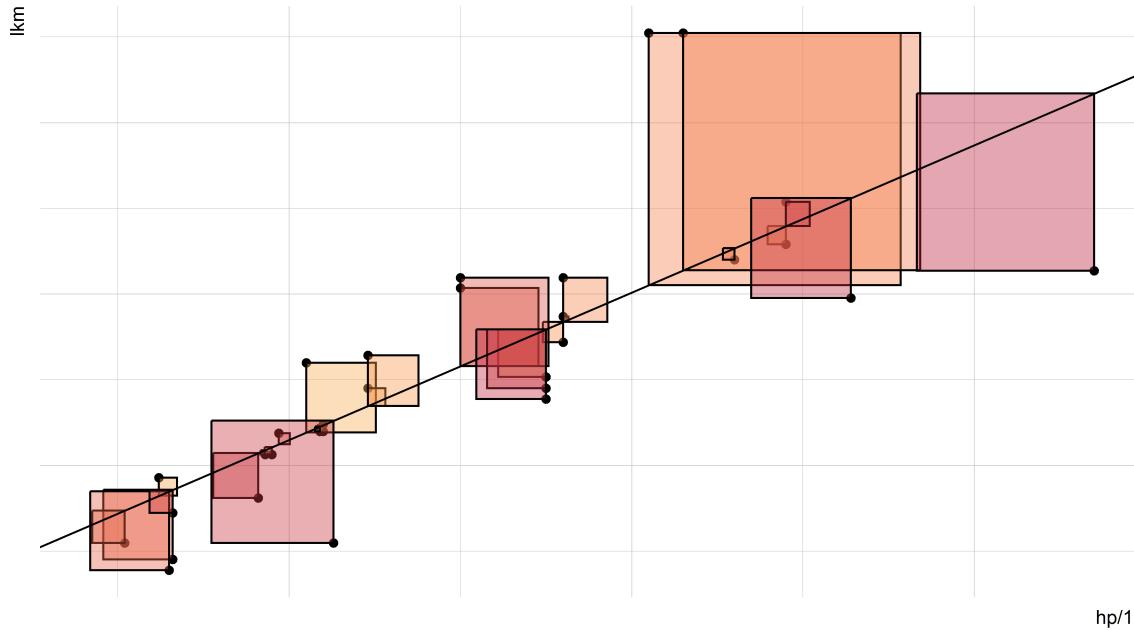
- Residuen und die RG:



Schätzung der Koeffizienten

- Was ist “minimale Abweichung”?
- Residuen aufsummieren?
 - ergibt immer 0
- Methode der kleinsten *Quadrate*!
 - Residuen quadrieren und dann summieren
 - Flächen sind immer positiv
 - je größere Residuen haben mehr Einfluss (zB. $1^2, 2^2, 3^2, \dots = 1, 4, 9, \dots$)
- Die RG minimiert die Summe der Abweichungsquadrate (SAQ):
 - = “Residual Sum of Squares, RSS”
 - = “Summe der Quadrate, $\text{SQ}_{\text{Rest}/\text{Residuen}}$ ”
 - = “Quadratsummen, QS”

Kleinste Quadrate Schätzung



- Die Koeffizienten der RG werden so geschätzt, dass sie die Summe der Abweichungsquadrate minimieren:
 - $\text{RSS} = \sum_{i=1}^n \epsilon_i = \sum_{i=1}^n y_i - \hat{y}_i = \min!$
 - $\epsilon \sim \mathcal{N}(0, \sigma_\epsilon)$

Arbeiten mit R



- Vorkenntnisse?
- “tidyverse”?

Beispieldatensatz

in R

- Eigenschaften von 32 Fahrzeugen (Autodatensatz)

- `?mtcars`

- `names(mtcars)`

- `summary(mtcars)`

- `df_cars <- mtcars`

Shortcuts

- `<-` ... Zuweisung: "Alt" + "-"
- `"ausführen"` ... "Strg" + "Enter" / "Alt" + "Enter"
- `%>%` ... Pipe: "Strg" + "Shift" + "m"

Funktionen

- `?xxx` ... Hilfe zu dem Befehl
- `names(xxx)` ... Namen eines Datensatzes / Objektes

Statistische Kennwerte

in R

- Autokauf. Welche Werte interessieren?
 - Kraftstoffverbrauch & Motorleistung
 - UV? AV?
 - AV: mpg, UV: hp
- statistische Kennwerte 1:

```
fivenum(df_cars$mpg)
fivenum(df_cars$hp)
```
- Funktionen
 - `$` ... Zugriff auf DF-Spalten / Unterobjekte
 - `fivenum()` ... 5-Zahlen Zusammenfassung
 - `get_summary_stats()` ... umfangreiche stat. Kennzahlen
 - `p_load()` ... laden & ggf. installieren von Paketen
- statistische Kennwerte 2:

```
pacman::p_load(tidyverse, rstatix)

df_cars %>%
  rstatix::get_summary_stats(mpg, hp) %>%
  flex()
```

variable	n	min	max	median	q1	q3	iqr	mad	mean	sd	se	ci
mpg	32	10.4	33.9	19.2	15.43	22.8	7.38	5.41	20.09	6.03	1.07	2.17
hp	32	52.0	335.0	123.0	96.50	180.0	83.50	77.10	146.69	68.56	12.12	24.72

Statistische Kennwerte

in R

- Berechnungen 1:

```
df_cars$1km <- 235.2145/df_cars$mpg
```

- Berechnungen 2:

```
df_cars <-  
  df_cars %>%  
  mutate(1km = 235.2145/mpg)
```

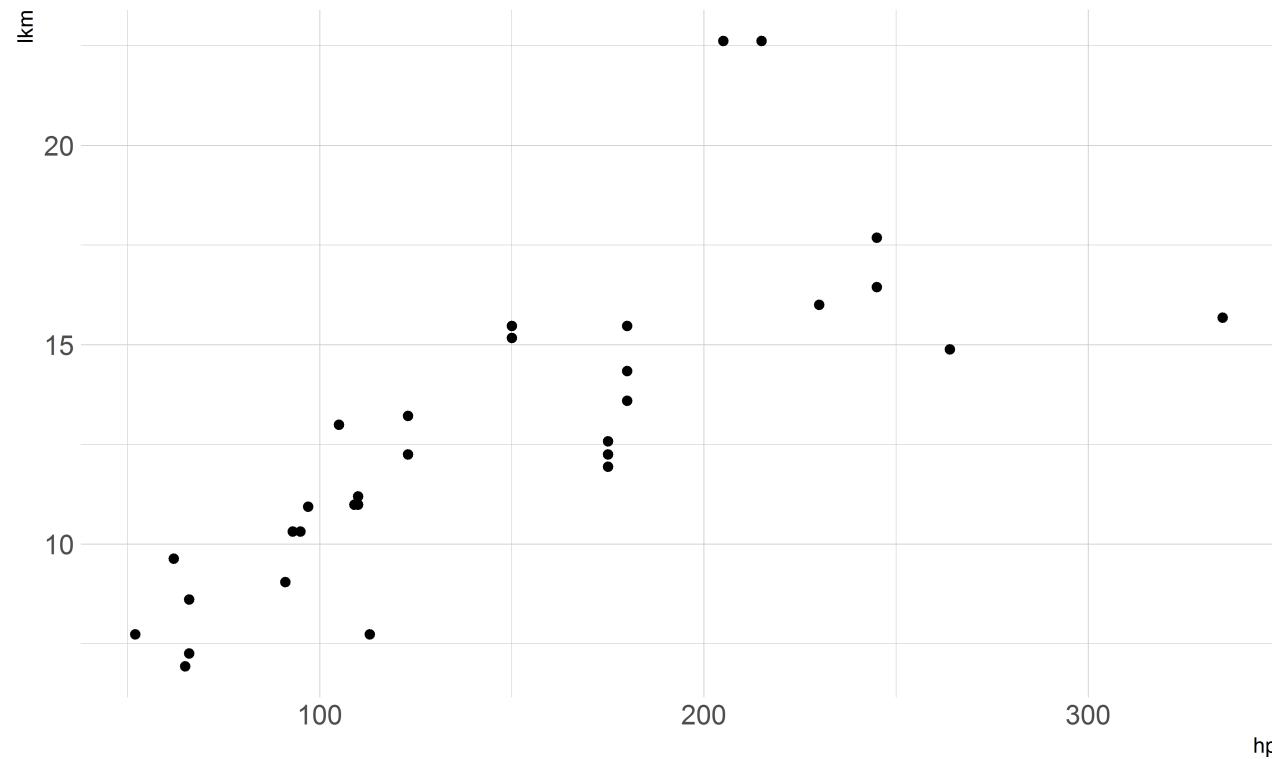
- Funktionen

- `mutate()` ... Erstellen neuer Variablen

Regression

in R

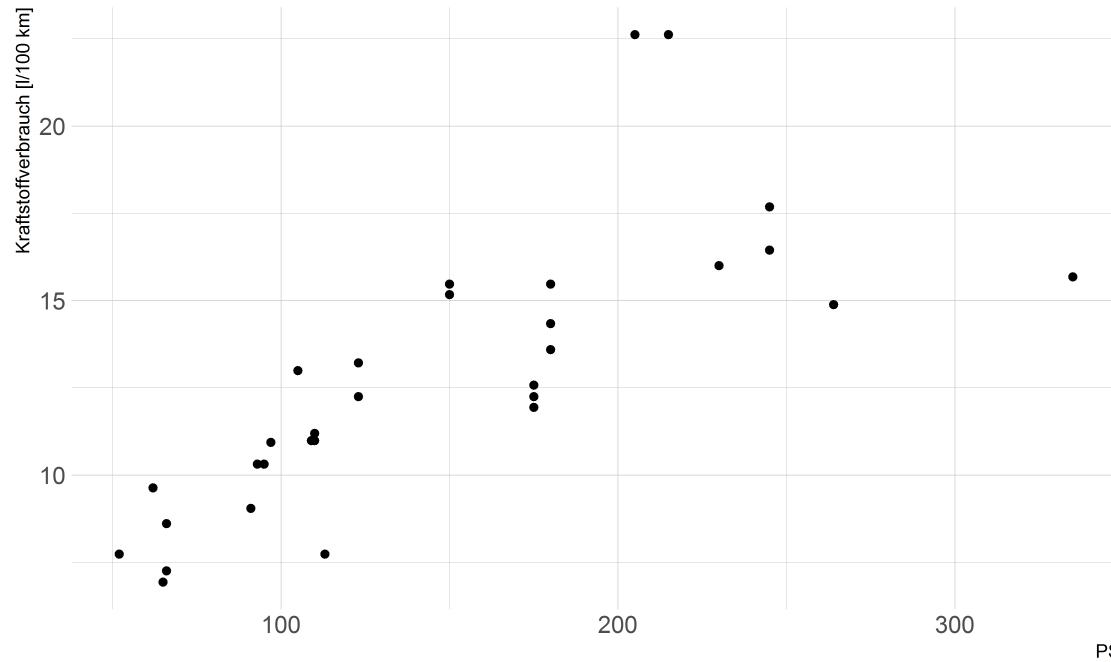
- Ziel: Vorhersage des Kraftstoffverbrauchs (l/100 km) durch Motorleistung (PS)
- Grundlage für folgende Beispiele und Berechnungen



Verteilungen visualisieren

- Streudiagramm:

```
df_cars %>%
  ggplot(aes(x = hp, y = lkm)) +
  geom_point() +
  labs(y = "Kraftstoffverbrauch [l/100 km]",
       x = "PS")
```

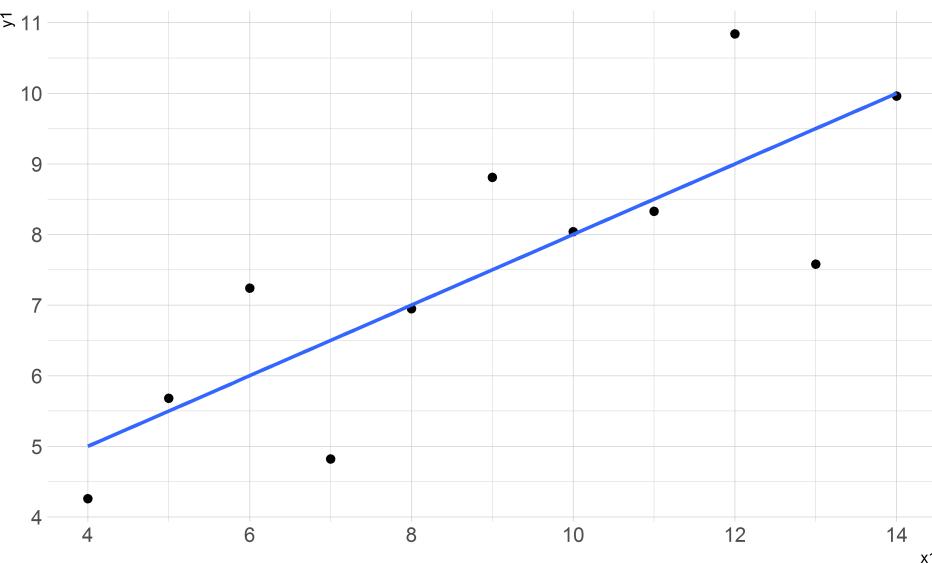


- ⚡ immer visualisieren

Visualisierung!

- Anscombe's Quartett
- statistische Kennwerte
- Streudiagramm

Eigenschaft	Wert
Mittelwert X	9.00
Mittelwert Y	7.50
Std. Abw. X	3.32
Std. Abw. Y	2.03
Korrelation	0.82
B ²	0.67
Regr. Gerade	$y = 3.00 + 0.50x$

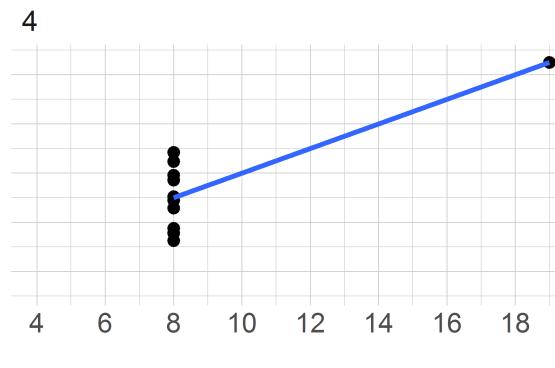
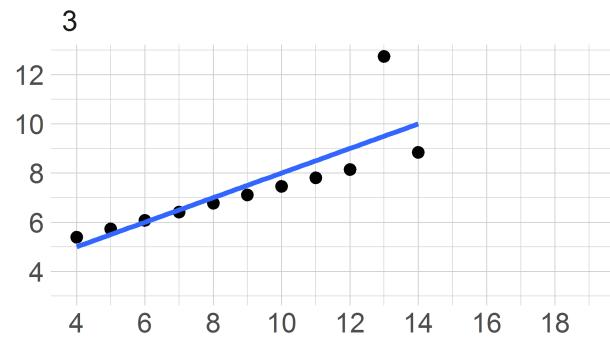
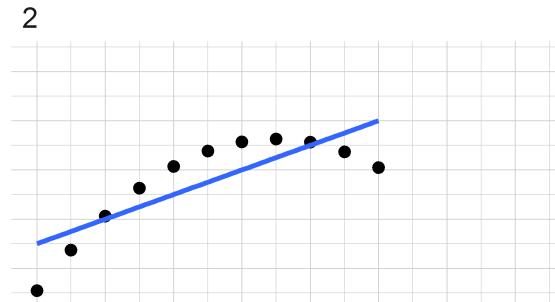
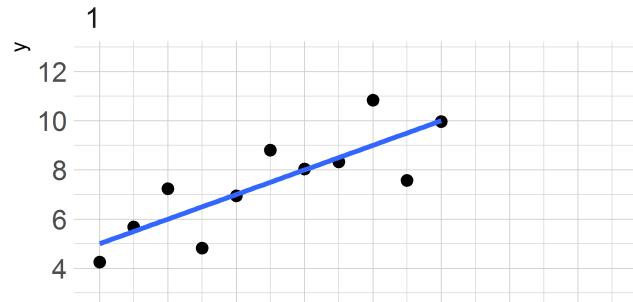


Visualisierung

Motivation

- Datensätze mit gleichen stat. Kennwerten:

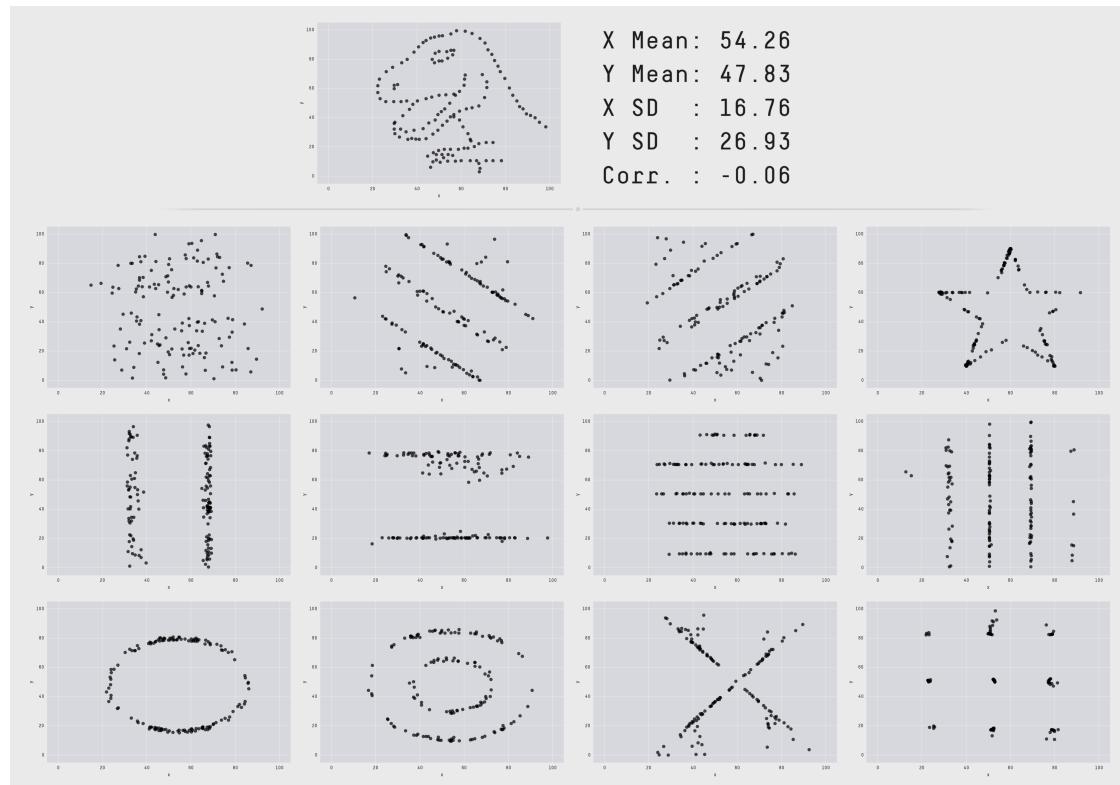
-



Visualisierung

"The Datasaurus Dozen"

- Albert Cairo



Quelle: <https://www.autodesk.com/research/publications/same-stats-different-graphs>

Regression in R

- Funktion:

```
?lm
```

- 2 wesentliche Argumente:

- *formula*: $y \sim x$
 - ohne Anführungszeichen
 - "y zurück führen auf x"
 - *data*

- Aufruf:

```
lm(  
  formula = lkm ~ hp,  
  data = df_cars)
```

- Funktionsargumente nach Position

```
lm(lkm ~ hp, df_cars)
```

Regression in R

- Ergebnis zwischenspeichern ...

```
lm_obj <- lm(lkm ~ hp, df_cars)
```

- ... und mehr sehen:

```
summary(lm_obj)
```

lm-Objekt inspizieren

- R-Objekt untersuchen 1:

```
names(lm_obj)
```

- R-Objekt untersuchen 2:

```
str(lm_obj)
```

- R-Objekt untersuchen 3: Cursor setzen und F2 drücken

```
lm_obj
```

Elemente `lm`-Objekt

- `coefficients` ... Regressionskoeffizienten
- `residuals` ... Fehlerterme (e)
- `rank` ... Anzahl geschätzter Parameter (b_0, b_1)
- `fitted.values` ... Vorhergesagte Werte (\hat{Y})
- `df.residual` ... Freiheitsgrade des Modells ($N - rank$)
- `terms` ... Formel

Regression in R

- Die Ausgaben von `lm_obj` und `summary(lm_obj)` liefern “*messy output*”
- schon besser:
 - Koeffizienten: `coef(lm_obj)`
 - 95%-Konfidenzintervalle: `confint(lm_obj)`
- am besten (“*tidy output*”):

```
broom:::tidy(lm_obj, conf.int = T) %>%
  flex(digits = 3)
```

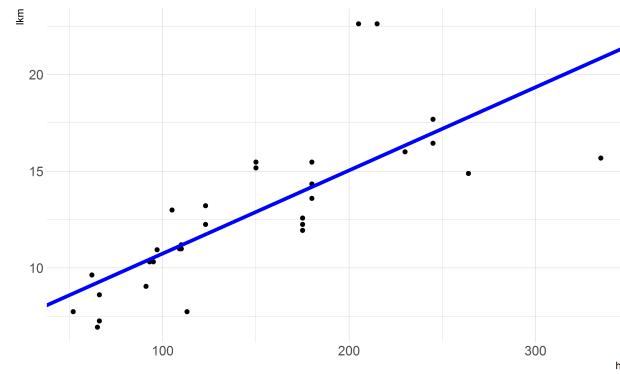
term	estimate	std.error	statistic	p.value	conf.low	conf.high
(Intercept)	6.449	1.074	6.006	p < .001	4.256	8.642
hp	0.043	0.007	6.464	p < .001	0.029	0.057

- geeignet für Weiterverarbeitung (“*pipeable*”)

Regressionskoeffizienten

- b_0 ... Ein Auto mit 0 PS benötigt für 100 km 6,45 Liter
- b_1 ... mit jedem zusätzlichem PS steigt der Kraftstoffverbrauch pro 100 km um 0,043 Liter

```
df_cars %>%
  ggplot(aes(hp, lkm)) +
  geom_point() +
  geom_abline(
    intercept = 6.45,
    slope = 0.043,
    color = 'blue',
    linewidth = 1.5)
```



Signifikanz der Koeffizienten

- $b_1 = 0.043$ ist ja fast 0! :(
- hat die Leistung (PS) dann überhaupt eine Bedeutung?
- **Signifikanzprüfung:**
 - ist der Regressionskoeffizient in der GG signifikant unterschiedlich von 0
 - bei einer Steigung von 0 verläuft die Gerade parallel zur x-Achse
 - Populationsparameter: β
 - $H_0 : \beta_1 = 0$
 - $H_A : \beta_1 \neq 0$
- Testverfahren: Analog zur klassischen t -Test Familie:
 - $t_{PG} = \frac{b_1}{SE_{b1}} \sim t_{1-\frac{\alpha}{2}; n-2}$
 - "Wald"-Test

Standardfehler der Koeffizienten

- SE = Standard Error ($\hat{\sigma}_{b1}$)
- Maß für die Genauigkeit/Präzision der Schätzung der Regressionskoeffizienten
- einfacher Standardfehler = Bereich in dem wahrer Wert des Koeffizienten mit 68%-Wk liegt (MW \pm 1SE)
- benötigt für Bestimmung der Konfidenzintervalle
- Formel:
$$\hat{\sigma}_{b1} = \frac{\sigma_\epsilon}{\sqrt{N} \times s_x}$$
- in R per Hand: `sigma(lm_obj) / sqrt(var(df_cars$hp) * 31)`
- Interpretation der Formel:
 - SE wächst mit der Residualvarianz des Modells
 - SE sinkt bei größerer Stichprobe
 - SE sinkt bei größerer Streuung des Prädiktors

Exkurs

Wahrscheinlichkeitsverteilungen

- theoretische, geschlossene Beschreibung von Zufallsgrößen
- 2 Gruppen:
 - diskrete Verteilungen
 - einzelne Werte
 - zB. Binomialverteilung ('Münzwurf')
 - zB. Poissonverteilung ('Warteschlange')
 - stetige Verteilungen:
 - kontinuierliche Merkmale
 - zB. Normalverteilung ('Körpergröße')
 - zB. t-Verteilung / "Student"-Verteilung

Exkurs

Wahrscheinlichkeitsverteilungen

- Wahrscheinlichkeitsdichtefunktion:
 - Wahrscheinlichkeit der Realisation einer bestimmten Ausprägung einer Zufallsvariable
 - "wie wahrscheinlich ist ein bestimmter Wert"
 - "Höhe des Funktionswertes" / y-Wert an der Stelle x
 - $P(X = x)$... Wahrscheinlichkeit (P), dass eine Zufallsvariable (X) den konkreten Wert (x) annimmt
- bei diskreten Verteilungen
 - abzählbare Menge an Ausprägungen
 - einzeln benennbar
 - ⚠ Summe der Einzelwahrscheinlichkeiten ("Balken") gleich 1
- bei stetigen Verteilungen
 - unendlich viele mögliche Ausprägungen
 - Fläche repräsentiert Wahrscheinlichkeit
 - ⚠ Fläche unter der Funktionskurve gleich 1

Exkurs

Wahrscheinlichkeitsverteilungen

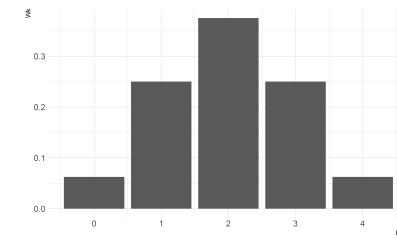
- zB. Münzwurf:
- zwei mögliche Versuchsausgänge, gleich bleibende Wahrscheinlichkeit
- folgt Binomialverteilung $Bi(n, p)$
- n ... Anzahl der Würfe
- p ... Erfolgswahrscheinlichkeit $P(X = k)$
 - $P(X = "Kopf") = p = \frac{1}{2}$
 - 1x Kopf(`x`), bei einem Wurf (`size`):
`dbinom(x = 1, size = 1, prob = 1/2)`
 - bei zwei Würfen (`size = 2`):
 - 0x Kopf: `dbinom(x = 0, size = 2, prob = 1/2)`
 - 1x Kopf: `dbinom(x = 1, size = 2, prob = 1/2)`
 - 2x Kopf: `dbinom(x = 2, size = 2, prob = 1/2)`

- Wahrscheinlichkeitswerte:

```
data.frame(Kopf = c(0, 1, 2, 3, 4)) %>%
  mutate(prob = dbinom(x = Kopf,
    size = 4, prob = 1/2)) %>%
  flex()
```

Kopf	prob
0	0.06
1	0.25
2	0.38
3	0.25
4	0.06

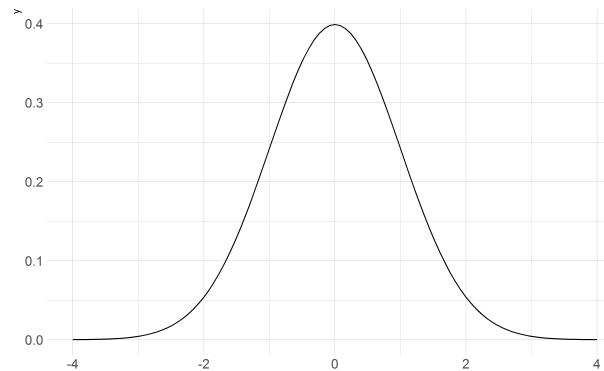
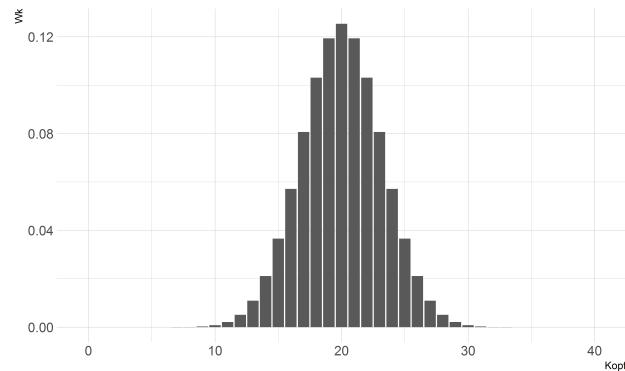
- Visualisierung: 4-facher Münzwurf = Dichtefunktion



Exkurs

Wahrscheinlichkeitsverteilungen

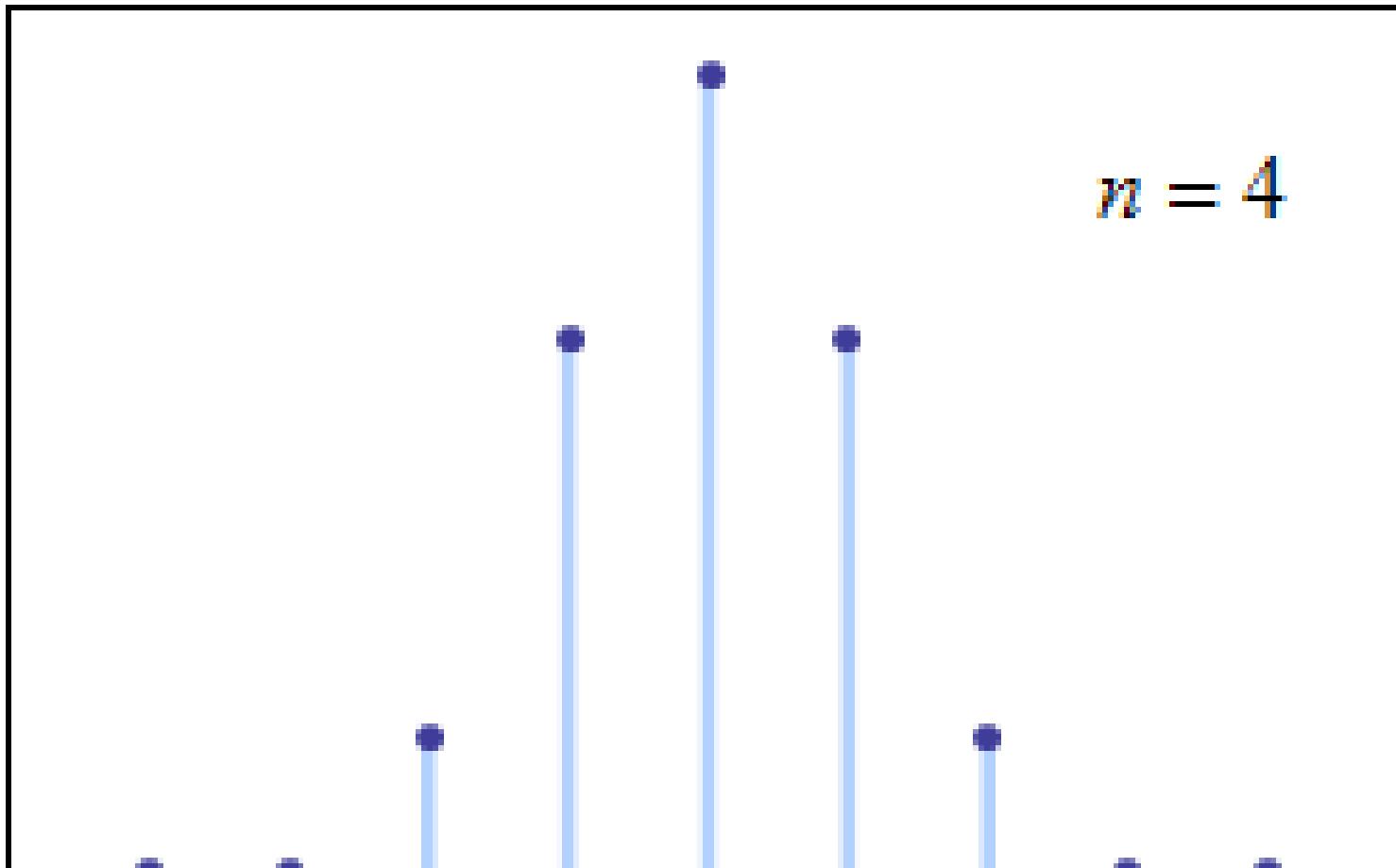
- bei zunehmendem n konvergiert Binomialverteilung zu Normalverteilung
- Visualisierung: 50-facher Münzwurf
- Visualisierung Normalverteilung



- ✓ bilden Grundlage für Hypothesenprüfung

Exkurs

Wahrscheinlichkeitsverteilungen



Exkurs

Wahrscheinlichkeitsverteilungen

- **Verteilungsfunktion:**
 - Wahrscheinlichkeit der Realisation einer Zufallsvariable bis zur Stelle x
 - $P(X < x)$
 - kumulierte Dichtefunktion (Wertebereich)
 - Wahrscheinlichkeit bis zu einem Wert x
 - diskrete Verteilungen: "summierte Balkenhöhen"
 - stetige Verteilungen: "summierte Flächen" (Integrale)
 - $P(X < x)$... Wahrscheinlichkeit (P), dass eine Zufallsvariable (X) *einen Wert kleiner als* den konkreten Wert (x) annimmt

Exkurs

Wahrscheinlichkeitsverteilungen

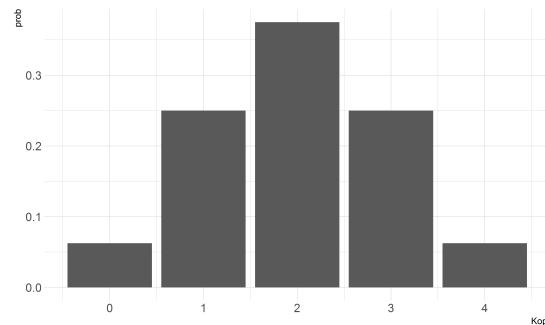
- Verteilungsfunktion beim Münzwurf
- zB. Münzwurf:
 - $P(X = \text{"Kopf"}) = p = \frac{1}{2}$
 - Binomialverteilung
 - 0-1x Kopf(`x`), bei einem Wurf (`size`): `qbinom(p = 1, size = 1, prob = 1/2)`
 - bei zwei Würfen (`size = 2`):
 - 0x Kopf: `pbinom(q = 0, size = 2, prob = 1/2)`
 - 0-1x Kopf: `pbinom(q = 1, size = 2, prob = 1/2)`
 - 0-2x Kopf: `pbinom(q = 2, size = 2, prob = 1/2)`
 - Vektorisierung:
 - `pbinom(q = c(0:2), size = 2, prob = 1/2)`

Exkurs

Wahrscheinlichkeitsverteilungen

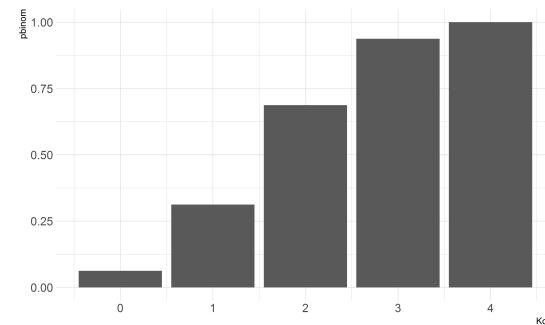
- Dichtefunktion/Wk. 4-facher Münzwurf

```
data.frame(Kopf = c(0,1,2,3,4)) %>%
  mutate(prob = dbinom(x = Kopf,
    size = 4, prob = 1/2)) %>%
  ggplot(aes(x = Kopf, y = prob)) +
  geom_col()
```



- Verteilungsfunktion

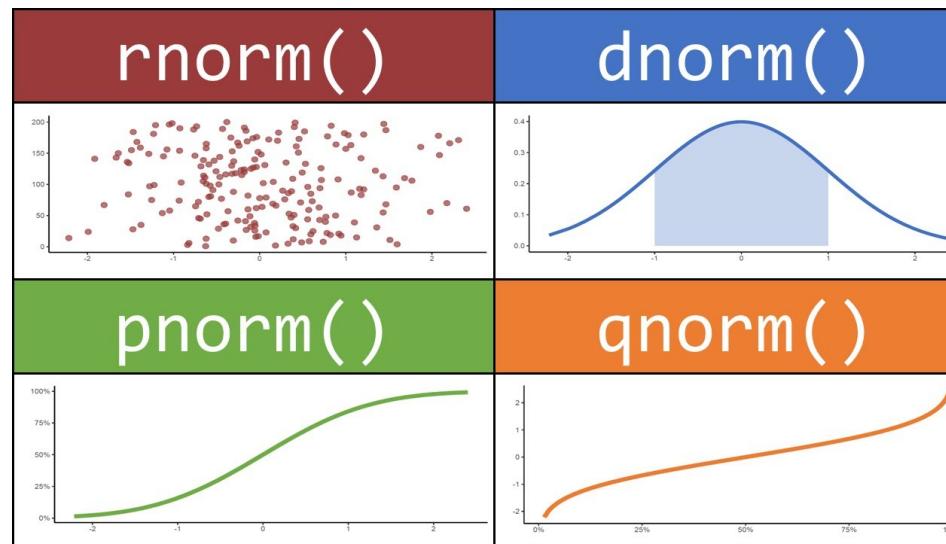
```
data.frame(Kopf = c(0,1,2,3,4)) %>%
  mutate(pbinom = pbinom(Kopf,
    size = 4, prob = 1/2)) %>%
  ggplot(aes(x = Kopf, y = pbinom)) +
  geom_col()
```



Exkurs

Wahrscheinlichkeitsverteilungen

- ?Distributions
- 4 Funktionsgruppen
 - `d***` ... Dichtefunktionen
 - `p***` ... Verteilungsfunktion
 - `q***` ... Quantilfunktionen (umgekehrte Verteilungsfunktion)
 - `r***` ... Zufallszahlen generieren
- Übersicht:



Exkurs

Wahrscheinlichkeitsverteilungen

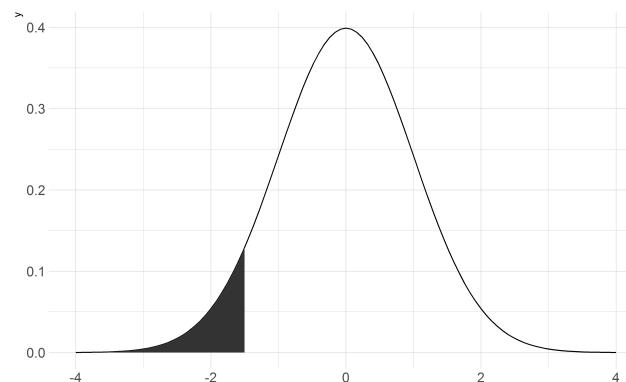
- Rechnen mit Wahrscheinlichkeitsverteilungen
- Wie wahrscheinlich ist 7x Kopf bei 10 Würfen?
 - `dbinom(7, 10, .5)`
- Wie wahrscheinlich sind höchstens 3x Kopf bei 10 Würfen?
 - Tipp: Verteilungsfunktion
 - `pbinom(3, 10, .5)`
 - `dbinom(0:3, 10, .5) %>% sum()`
- Wie wahrscheinlich sind *mehr* als 10x Kopf, bei 19 Würfen
 - Tipp: `?pbinom` bzw. "Gegenereignis"
 - `1 - pbinom(10, 19, .5)`
 - `pbinom(10, 19, .5, lower.tail = F)`
 - `dbinom(11:19, 19, .5) %>% sum()`

Exkurs

Wahrscheinlichkeitsverteilungen

- bei Stetigen Verteilungen: Berechnung über Flächen (\int_a^b)
- Flächeninhalte entsprechen Wahrscheinlichkeiten
- in der Praxis identisch zu diskreten Verteilungen
- Normalverteilung:

```
ggplot() +  
  xlim(-4, 4) +  
  stat_function(fun = dnorm) +  
  stat_function(fun = dnorm, geom='area', xlim = c(-4, -1.5))
```



- Funktionswert an der Stelle x: `dnorm(x = 0)`
- Wahrscheinlichkeit für Wert kleiner -1.5: `pnorm(q = -1.5)`

Exkurs

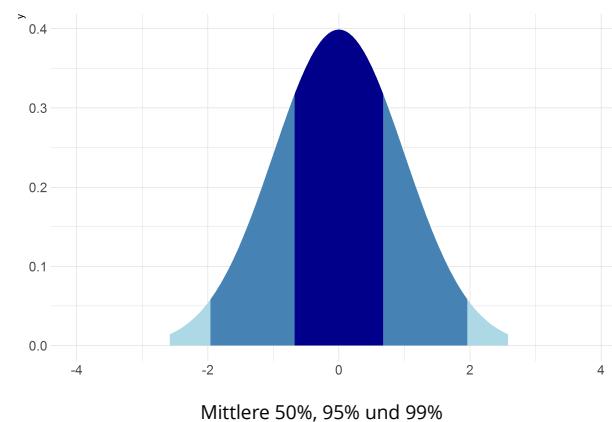
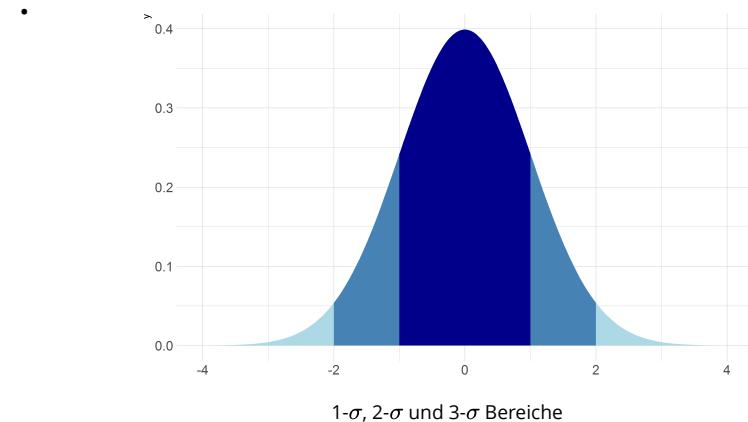
Standardnormalverteilug

- bisher nur *Standardnormalverteilug*
- “Gauß’sche Glockenkurve”,
- 2 Parameter: μ und σ
- Eigenschaften
 - symmetrisch um μ
 - eingipflig
 - Fläche = 1
 - Punktwahrscheinlichkeit = 0
 - Dichte stets > 0
- Formel: $\varphi(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$
- ergibt sich, wenn sich viele Einzeleinflüsse (additiv) überlagern

Exkurs

Standardnormalverteilung

- Flächen unter der Dichtekurve lassen sich Wahrscheinlichkeiten zuordnen
- Standardnormalverteilung: $\mu = 0, \sigma = 1$
- entspricht Quintilen



Intervallgrenzen für $X \sim N(\mu, \sigma^2)$	Intervallgrenzen für $Z \sim N(0, 1)$	Bezeichnung des Intervalls	Wahrscheinlichkeit p
$\mu - \sigma \leq X \leq \mu + \sigma$	$-1 \leq Z \leq 1$	1 σ -Bereich	0,6827
$\mu - 2\sigma \leq X \leq \mu + 2\sigma$	$-2 \leq Z \leq 2$	2 σ -Bereich	0,9545
$\mu - 3\sigma \leq X \leq \mu + 3\sigma$	$-3 \leq Z \leq 3$	3 σ -Bereich	0,9973
$\mu - 1,96\sigma \leq X \leq \mu + 1,96\sigma$	$-1,96 \leq Z \leq 1,96$	95 %-Referenzbereich	0,95
$\mu - 2,58\sigma \leq X \leq \mu + 2,58\sigma$	$-2,58 \leq Z \leq 2,58$	99 %-Referenzbereich	0,99

Intervalle und Wahrscheinlichkeiten der Normalverteilung

Exkurs

Übersicht Verteilungen

- Die häufigsten Verteilungen:

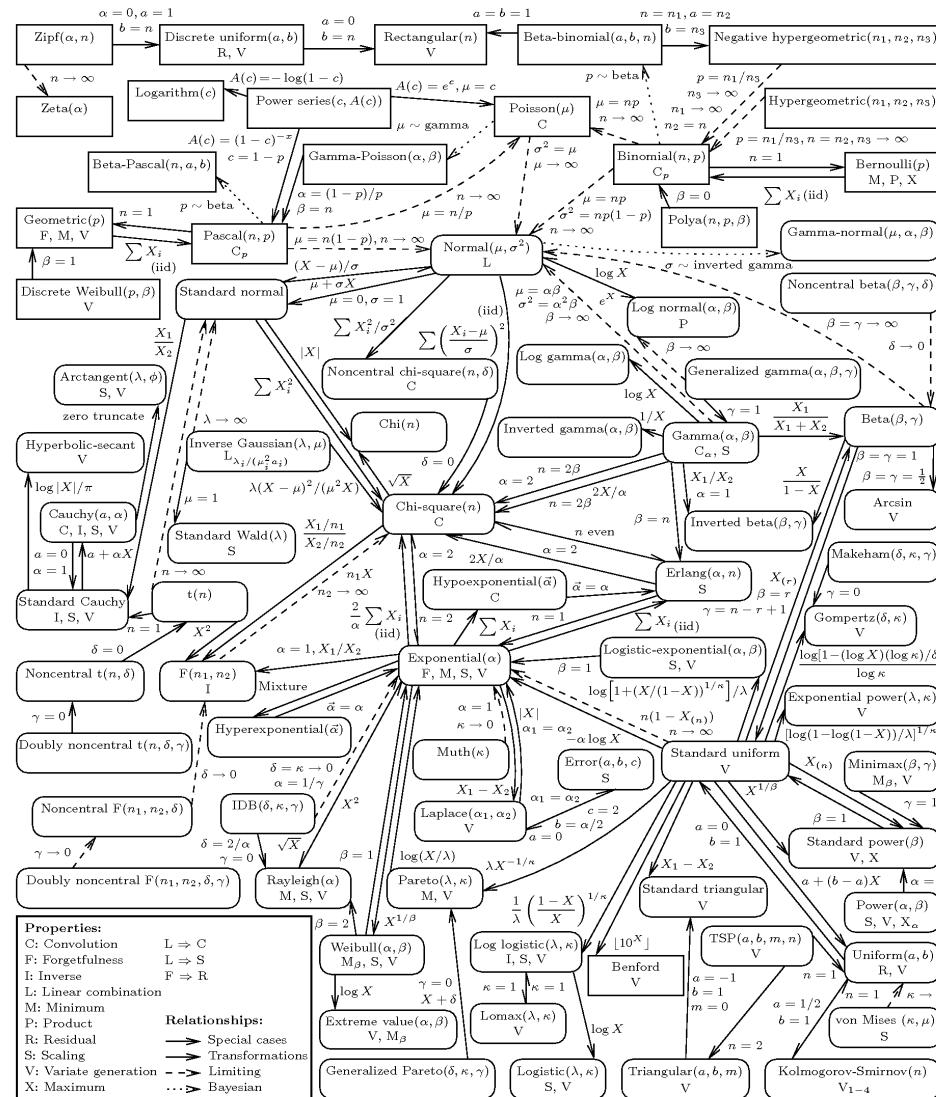
Intervallgrenzen für $X \sim N(\mu, \sigma^2)$	Intervallgrenzen für $Z \sim N(0, 1)$	Bezeichnung des Intervalls	Wahrscheinlich- keit p
$\mu - \sigma \leq X \leq \mu + \sigma$	$-1 \leq Z \leq 1$	1σ -Bereich	0,6827
$\mu - 2\sigma \leq X \leq \mu + 2\sigma$	$-2 \leq Z \leq 2$	2σ -Bereich	0,9545
$\mu - 3\sigma \leq X \leq \mu + 3\sigma$	$-3 \leq Z \leq 3$	3σ -Bereich	0,9973
$\mu - 1,96\sigma \leq X \leq \mu + 1,96\sigma$	$-1,96 \leq Z \leq 1,96$	95 %-Referenzbereich	0,95
$\mu - 2,58\sigma \leq X \leq \mu + 2,58\sigma$	$-2,58 \leq Z \leq 2,58$	99 %-Referenzbereich	0,99

Intervalle und Wahrscheinlichkeiten der Normalverteilung

Exkurs

Übersicht Verteilungen

- univariate Verteilungen:



Exkurs: Hypothesen

Grundlagen

- Hypothese: wissenschaftliche Vermutung über einen Unterschied / Zusammenhang
- Form: "ja/nein" bzw "entweder/oder"-Aussage (\rightarrow disjunkte Ereignisse)
 - Bsp: Impfen senkt Hospitalisationsrate (oder eben nicht)
 - Bsp: Lernzeitdauer hängt zusammen mit Zensurpunktzahl (oder eben nicht)
- 2 Arten von Hypothesen:
 - Nullhypothese (H_0):
 - keine Wirkung / Effekt / Unterschied / Zusammenhang
 - bisheriger Standard
 - Alternativhypothese (H_A/H_1):
 - eigentlich zu testende Vermutung

Hypothesen

Prinzipien

- K. Popper: Kriterium der Falsifizierbarkeit
 - 2 Arten von Wirklichkeitsaussagen:
 - Existenzsätze: "Es gibt Bier auf Hawaii!"
 - Allsätze: "Es gibt *kein* Bier auf Hawaii!"
 - \Rightarrow Hypothesen können nicht bewiesen, nur widerlegt werden!
 - (Arbeitshypothese \rightarrow bewährte Hypothese)
- Logik:
 - man hat eine Theorie / Vermutung (H_A)
 - man unterstellt das Gegenteil (H_0)
 - das kann/muss nicht bewiesen werden! (Bsp. Unschuldsvermutung)
 - nun versucht man zu zeigen, dass dieses Gegenteil (also H_0) wahr ist
 - gelingt dies nicht, so ist dies ein Beleg für die eigentliche Theorie (H_A)

Hypothesen

Münzwurf-Beispiel

- Hypothese: "Münze ist gezinkt!" (Kopf fällt zu oft)
- H_A : "Münze ist nicht fair" ($\pi_{Kopf} \neq 0.5$)
- H_0 : "Münze ist fair" ($\pi_{Kopf} = 0.5$)
- Experiment durchführen!
- Wie oft muss Kopf fallen, damit es *kritisch* wird?
- Wahrscheinlichkeiten kann ich für jedes p genau ausrechnen (Binomialverteilung)
- aber welches p wählen?
- das von H_0 !
- 7x Kopf hintereinander ist sehr unwahrscheinlich
- Restrisiko für Irrtum bleibt (\rightarrow Dichtefunktion NV $\neq 0$)

• Wie Wahrscheinlich ist:

Anzahl Würfe	Anzahl Kopf	P
1	1	0.5
2	2	0.25
3	3	0.125
4	4	0.0625
5	5	0.03125
6	6	0.015625
7	7	0.0078125

Hypothesen

- ⚠ Merke:



Hypothesen

α -Fehler (Irrtumswahrscheinlichkeit)

- α -Fehler: obwohl H_0 gilt verwirft der Test H_0
- formal: $P(\text{Test: } H_A | H_0 \text{ gilt})$
- α -Fehler ist nicht vermeidbar, aber kontrollierbar
- Konventionsgemäß: $\alpha = 0.05$ (α -Niveau, Signifikanzniveau, Irrtumswahrscheinlichkeit)
- andere Werte sind möglich, z.B. Medikamentenstudie
- $1 - \alpha$... statistische Sicherheit, Konfidenz
- tatsächliche Wahrscheinlichkeit für Fehlentscheidung = p -Wert

- Wahrheit, Testententscheidung & Resultat

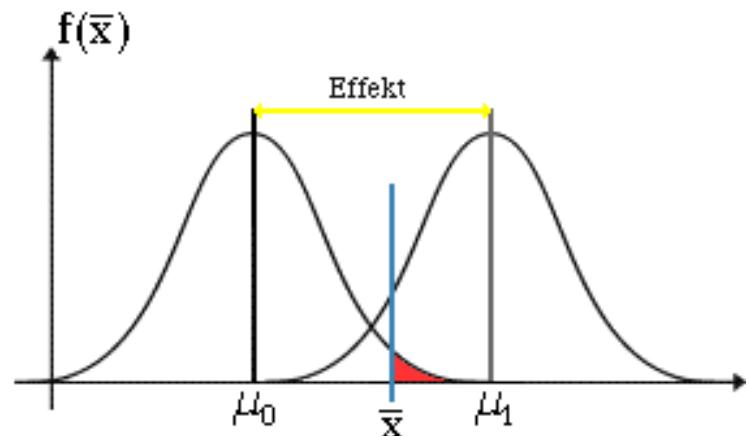
Testent-scheidung	Wirklichkeit	
	H_0 wahr	H_1 wahr
Für H_0	Richtige Entscheidung $1-\alpha$	Fehler 2. Art β
Für H_1	Fehler 1. Art α	Richtige Entscheidung $1-\beta$
Summe	1	1

Hypothesen

β -Fehler

- β -Fehler: eigentlich gilt H_A , aber der Test spricht für H_0
- formal: $P(\text{Test: } H_0 | H_A \text{ gilt})$
- Gegenwahrscheinlichkeit: $1 - \beta$... Power, Teststärke
- nicht im voraus bestimmbar, nicht kontrollierbar
- Konvention: $\beta = 0.20$
- α und β stehen in indirekter Beziehung:
 - $\downarrow \alpha \rightarrow \uparrow \beta$
 - $\uparrow \alpha \rightarrow \downarrow \beta$
- mit zunehmendem n sinkt β

- Fehlerarten - graphisch



Fehlerarten; Quelle: <http://www.mesosworld.ch>

Hypothesentestung

Simulation

- Beispiel:
 - 100 simulierte Stichproben
 - jeweils 2 Gruppen mit je 50 Beobachtungen
 - Werte sind immer aus gleicher GG gezogen ($NV, \mu = 0, \sigma = 1$)

- R-Code:

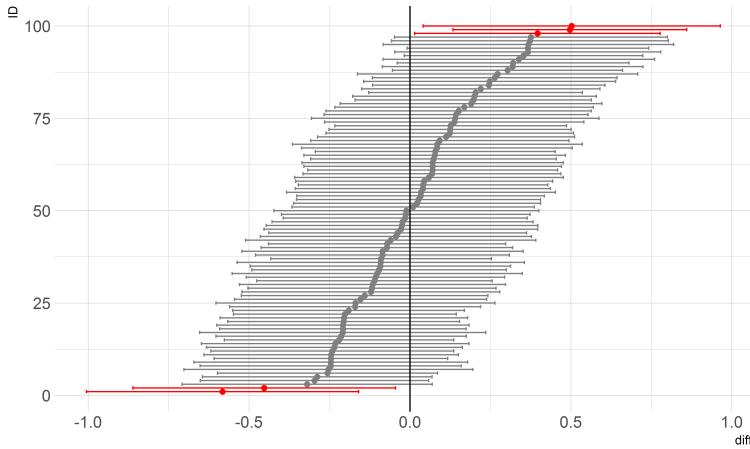
```
set.seed(99)

df_sim <-
  tibble(run = 1:100) %>%
  rowwise() %>%
  mutate(dat = tibble(
    group = c(rep('A', 50),
              rep('B', 50)),
    value = rnorm(100)) %>%
    list())
  ) %>%
  ungroup()
```

Testfehler

Simulation

- 100 t-Tests als Gruppenvergleich
- ca. 5% ergeben ein signifikantes Ergebnis
-



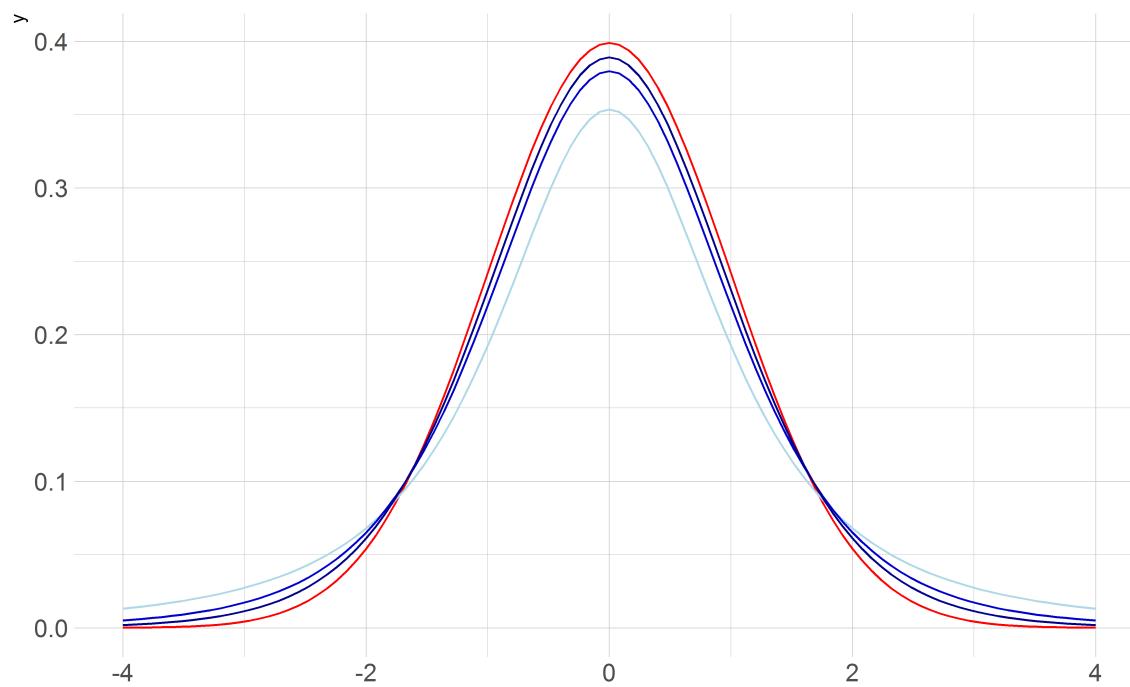
```
df_sim %>%
  rowwise() %>%
  mutate(rstatix::t_test(dat,
    value ~ group,
    detailed = T) %>%
    as.data.frame) %>%
  mutate(color =
    if_else(p<0.05,
      'red',
      'grey50')) %>%
  ungroup() %>%
  mutate(diff = estimate1-estimate2) %>%
  arrange(diff) %>%
  mutate(ID = row_number()) %>%
  ggplot(aes(y=ID,
    x = diff,
    xmin = conf.low,
    xmax=conf.high,
    color = color)) +
  geom_errorbar() +
  geom_point() +
  geom_vline(xintercept = 0) +
  scale_color_identity()
```

Prüfverteilung

- Warum t -Verteilung statt Normalverteilung?
- Extremwerte werden seltener in Stichproben beobachtet
- entsprechend wird die Streuung systematisch unterschätzt
- je größer die Stichprobe, umso kleiner der Effekt
- mit größerem n nähert sich t -Verteilung der Normalverteilung an
- Form der t -Verteilung wird mit einem Parameter beschrieben
- df ... Degrees of Freedom, Freiheitsgrade
- i.A.: $df = n - \text{Parameterzahl}$
- zB. Mittelwert: $df = n - 1$
- zB. Standardabweichung: $df = n - 2$
- zB. einfache Regression: $df = n - 2 (b_0, b_1)$

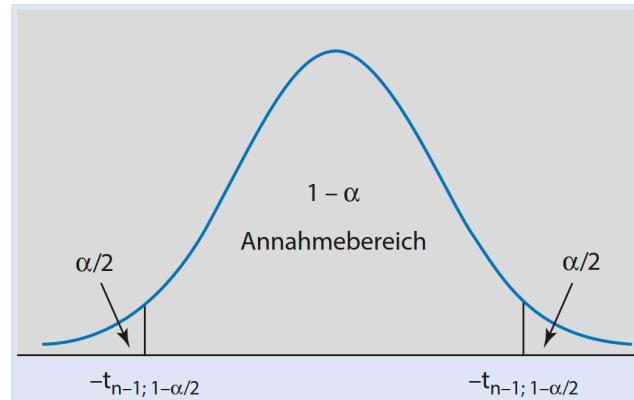
Prüfverteilung

```
ggplot() +  
  xlim(-4, 4) +  
  stat_function(fun = dnorm, color = 'red') +  
  stat_function(fun = dt, args = list(df = 2), color = 'lightblue') +  
  stat_function(fun = dt, args = list(df = 5), color = 'mediumblue') +  
  stat_function(fun = dt, args = list(df = 10), color = 'darkblue')
```



Bezug zur Regression

- Prüfverteilung:



- zweiseitige Fragestellung ("hauptsache nicht 0")
- Prüfgröße im mittleren Bereich: H_0 beibehalten, b -Koeffizient *nicht* signifikant von 0 verschieden
- Prüfgröße im äußeren Bereiche: H_0 ablehnen, b -Koeffizient *signifikant* von 0 verschieden
- Grenzen: kritische Werte
- ✓ wenn $p < \alpha$, dann H_0 ablehnen

zurück zu Regression in R

- p-Werte von Hand berechnen
 - `summary(lm_obj)`
 - $t = b/\text{SE}(b)$
 - `pt()`
 - `pt(q = 0.0430/0.00665, df = 30, lower.tail = F)*2`
- Der p-Wert zu b_0 ist klar kleiner als 0.05,
- damit hat die Leistung einen signifikanten Einfluss auf den Verbrauch
- **⚠️** Der betragsmäßige Wert von b_1 sagt nichts über die Bedeutung aus!
- der Wert des Regressionskoeffizienten b_1 hängt vom Wertebereich von X ab
 - z.B. Körpergröße in cm vs. in m, Koeffizient im 2. Fall nur 1/100 so groß
 - Wert des Reg.Koef. daher nicht isoliert interpretierbar

Regressionskoeffizienten

- Wie ändern Sich der Steigungskoeffizient, wenn Sie die PS in kW umrechnen? (1PS = 0,74 kW)
- Und was passiert mit dem p-Wert?
- Und was mit dem Achsenabschnitt?

```
• df_cars %>%
  mutate(kw = hp/0.74) %>%
  lm(lkm ~ kw, data = .) %>%
  broom::tidy()
```

- der p-Wert bleibt unverändert!

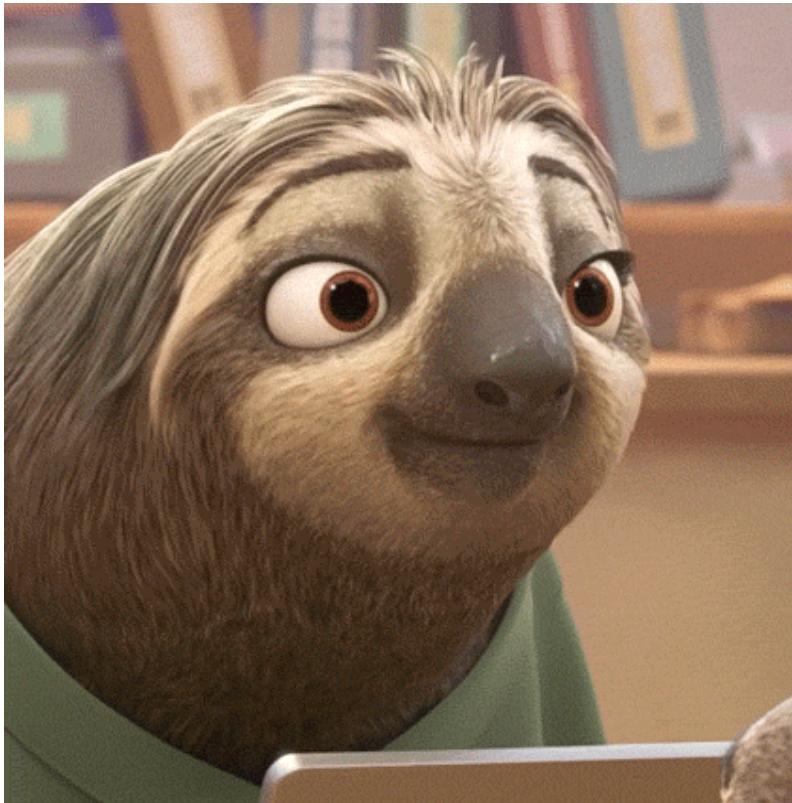
P-Werte

- Der p -Wert ist die *tatsächliche* Irrtumswahrscheinlichkeit einen Fehler 1. Art zu begehen.
- Mit welcher Wahrscheinlichkeit ist ein bestimmter Parameterwert zu beobachten, unter der Annahme von H_0
- Der α -Wert ist die Irrtumswahrscheinlichkeit, die maximal toleriert wird
- Konventionen:

p-Wert	Interpretation
$\geq .10$	nicht signifikant
$\geq .05$	tendenziell signifikant
$\geq .01$	signifikant
$\geq .001$	hoch signifikant
$< .001$	höchst signifikant

- $\Delta p \neq 0$ und $p \neq 1$
- $\checkmark p < 0.001$ und $p > .99$

Fertig?



- ... leider nicht.

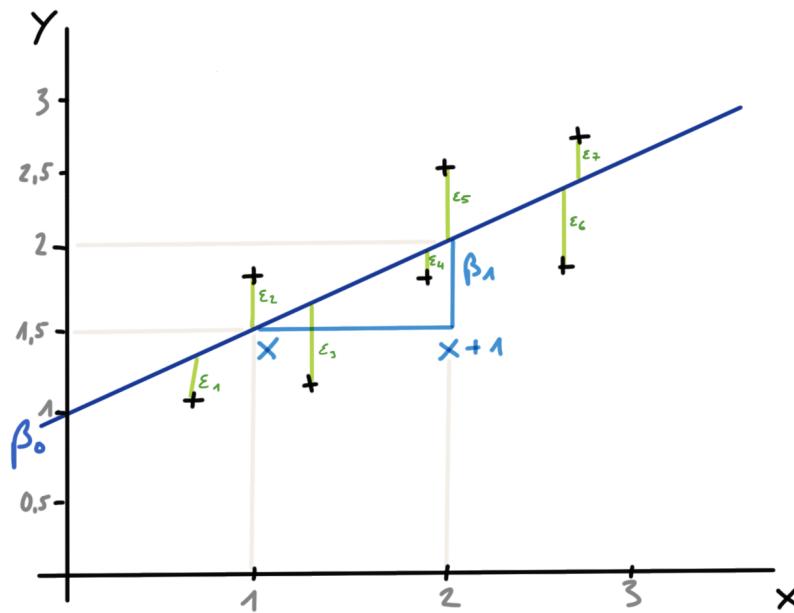
Modellannahmen

Voraussetzungen

- Das Modell basiert auf verschiedenen Annahmen / Voraussetzungen:
 1. lineare Beziehung der Variablen zueinander
 2. Normalverteilung der Residuen
 3. die Residuen sind unabhängig voneinander (keine Autokorrelation)
 4. die Varianzen der Residuen sind über den Wertebereich konstant (Homoskedastizität)
 5. Ausreißerdiagnostik 1: große Residuen
 6. Ausreißerdiagnostik 2: ungewöhnliche Prädiktorwerte
- ⇒ **Residuenanalyse**

Residuum

- **Residuum:** beschreibt die Abweichung zwischen beobachtetem Wert und Modellvorhersage
- $\epsilon_i = y_i - \hat{y}_i$
- Residuen und die RG:

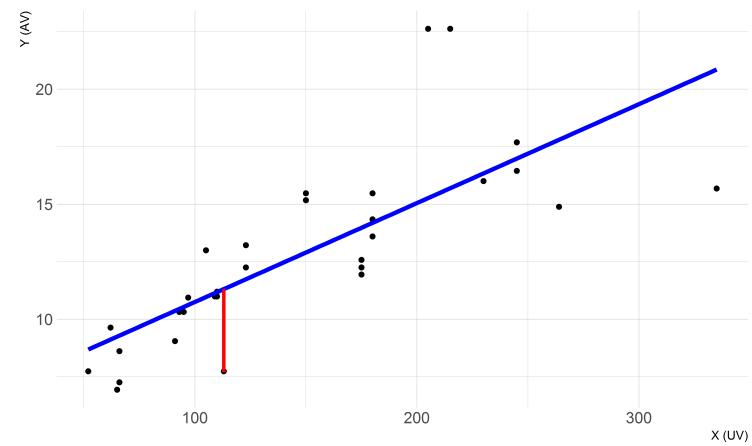


Modellvorhersage

- für Berechnung von $\epsilon_i = y_i - \hat{y}_i$ brauchen wir \hat{y}_i
- z.B. für das 28. Auto im Datensatz (Lotus Europa)
- Merkmalswerte:
 - `df_cars[28,]`
 - $hp = 113$, $lkm = 7.74$
- Bestimmung der Modellvorhersage:
 - Einsetzen des X-Wertes (hp) in Regressionsgleichung
 - $b_0 = 6.449$ und $b_1 = 0.04299$
 - $\hat{y}_{28} = 6.449 + 0.04299 \times 113$
 - $\hat{y}_{28} = 11.31$

Residuum

- Nun haben wir alles was wir zur Berechnung des Residuums brauchen:
 - $y_i = 7.74$
 - $\hat{y}_i = 11.31$
- Residuum:
 - $\epsilon_i = y_i - \hat{y}_i$
 - $\epsilon_{28} = 7.74 - 11.31 = -3.57$
- ⚡ Umständlich, oder?!



Residuen

- Geschätzte Werte: `fitted(lm_obj)`
- Residuen: `residuals(lm_obj)`
- Und für den Lotus Europa:
 - `fitted(lm_obj) [28]`
 - `residuals(lm_obj) [28]`

Weiterverarbeitung

- Und alle Werte (x, y, \hat{y}, ϵ) zusammen:

- `?bind_cols`

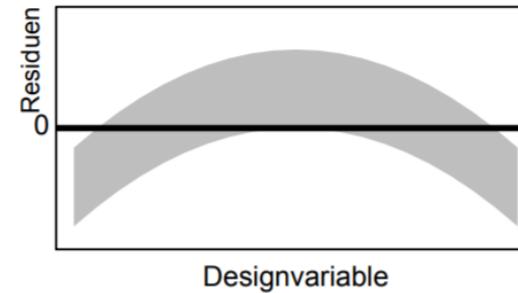
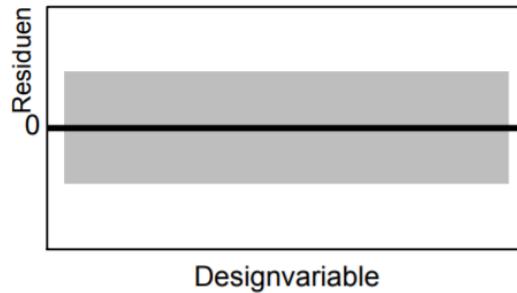
```
• df_res <-  
  bind_cols(df_cars,  
            fitted = fitted(lm_obj),  
            residuals = residuals (lm_obj))
```

- und wieder *tidy*: alle Werte mit einer Funktion: `broom::augment(lm_obj)`
- unser Lotus Europa: `broom::augment(lm_obj) [28,]`

Modellannahmen

1. Lineare Beziehung

- Grundannahme der *linearen* Regression ist die lineare Beziehung des Prädiktors zum Kriterium
- dafür **Residuenplots** erstellen
- einfaches Streudiagramm:
 - x-Achse: x_i
 - y-Achse: ϵ_i
- Plot:



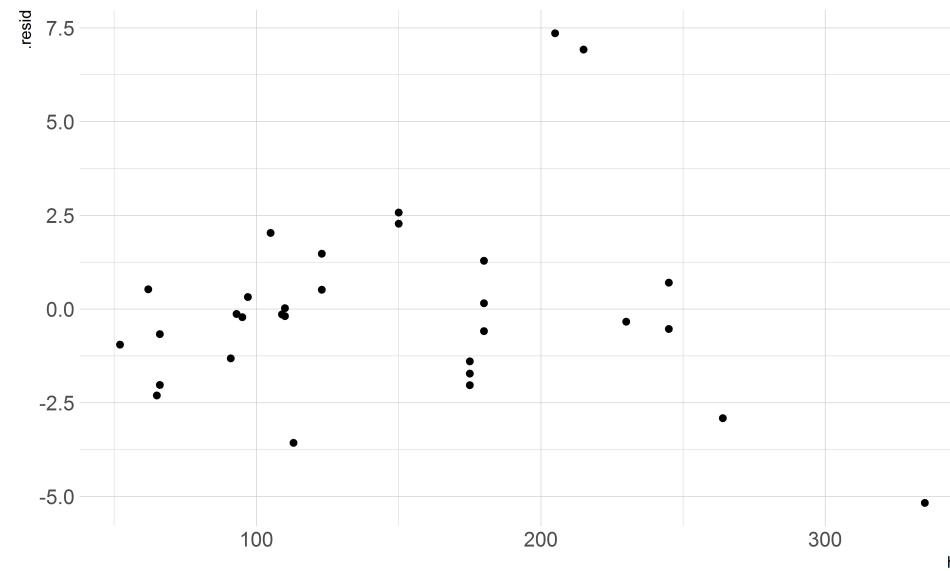
- die Residuen sollten über den gesamten x-Bereich gleichmäßig über- bzw. unterhalb der 0-Linie streuen (linke Abb)
- dabei sollte kein systematisches Muster erkennbar sein (z.B. rechte Abb)

Modellannahmen

1. Lineare Beziehung

- Erstellen Sie ein Residuenplots für unser Regressionsmodell!
- Streudiagramm: `ggplot(aes(x, y)) + geom_point()`

```
• broom::augment(lm_obj) %>%
  ggplot(aes(hp, .resid)) +
  geom_point()
```

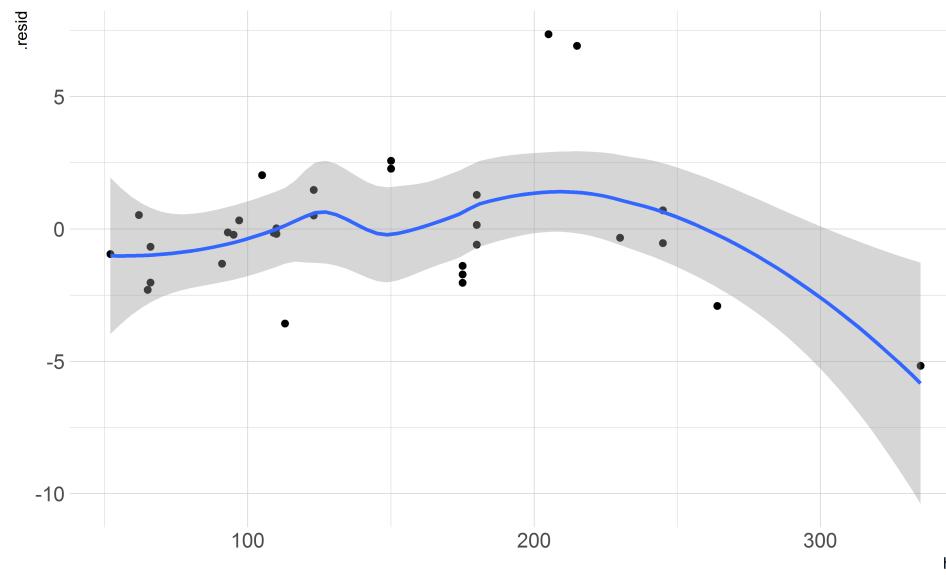


Modellannahmen

1. Lineare Beziehung

- für die Erkennung von Trends ist eine zusätzliche Anpassungslinie hilfreich

```
broom:::augment(lm_obj) %>%
  ggplot(aes(hp, .resid)) +
  geom_point() +
  geom_smooth(method = 'loess', formula = 'y ~ x')
```



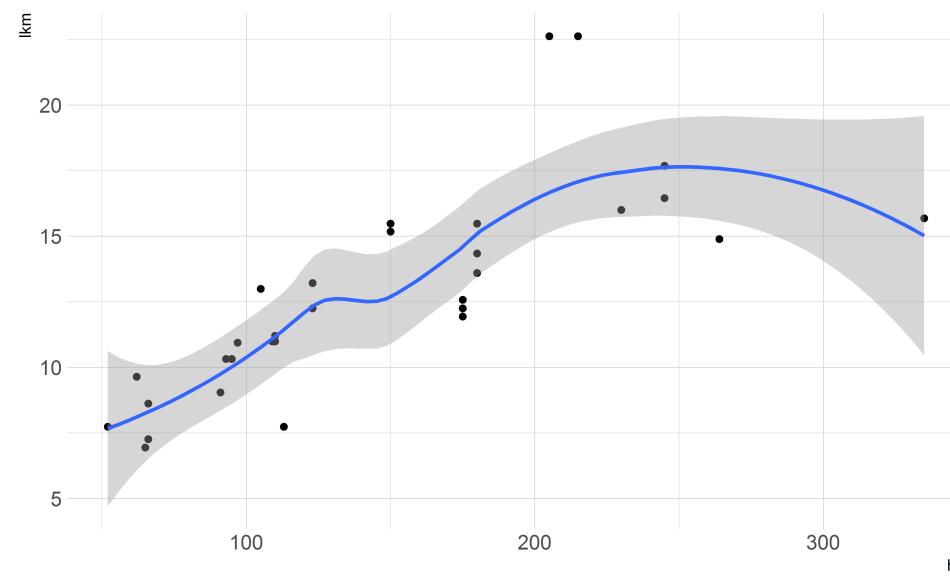
- diese sollte um die $y=0$ - Line verlaufen

Modellannahmen

1. Lineare Beziehung

- diese Beziehung kann (bei einem Prädiktor) auch direkt an den Rohdaten untersucht werden:

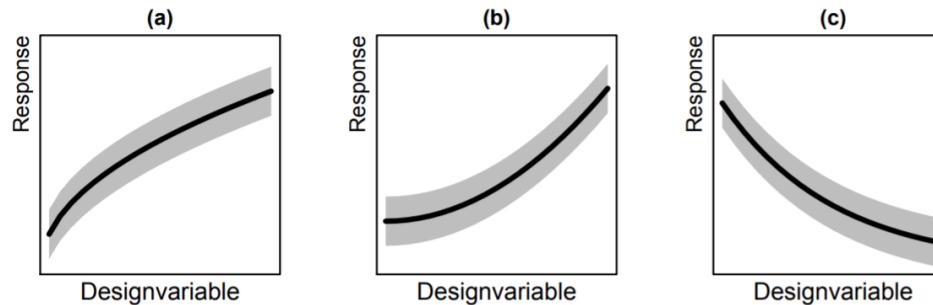
```
broom::augment(lm_obj) %>%
  ggplot(aes(hp, lkm)) +
  geom_point() +
  geom_smooth(method = 'loess', formula = 'y ~ x')
```



Modellannahmen

1. Lineare Beziehung

- weitere mögliche Beziehungen (hier wieder zwischen X und Y):

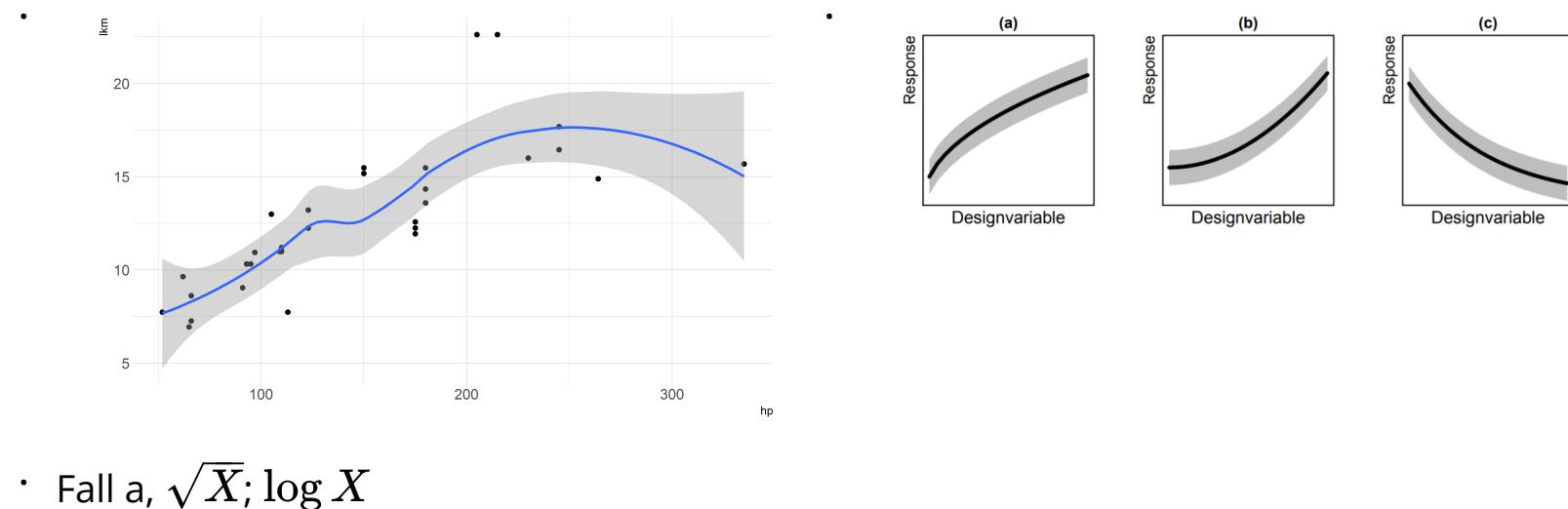


- Lösung: *linearisierende Transformationen* des Prädiktors X
- mögliche Transformationen:
 - a: \sqrt{X} ; $\log X$
 - b: X^2 ; e^X
 - c: $\frac{1}{X}$; e^{-X}
- oder Hinzunahme weiterer Prädiktoren

Modellannahmen

1. Lineare Beziehung

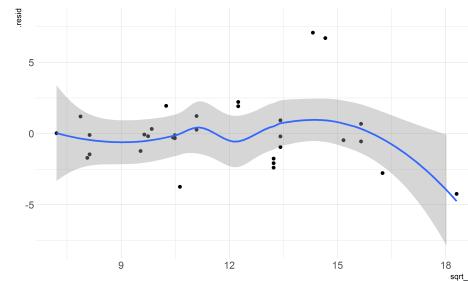
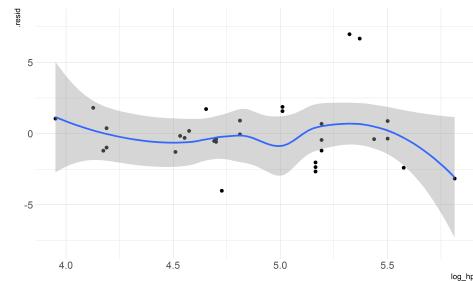
- Welchen der Fälle beschreibt den Zusammenhang zwischen PS und Verbrauch am ehesten?



Modellannahmen

1. Lineare Beziehung

- Wiederholen Sie die Regression mit einem transformierten Prädiktor!
- Probieren Sie dabei sowohl 1. logarithmieren und 2. radizieren.
- Inspizieren Sie die Residuenplots. Achten Sie auf die unterschiedlichen y-Achsen.
- Wie verändert sich die x-Achse? Welche Transformation wirkt also stärker?
- Welche Transformation scheint geeigneter? Warum?
- `## `geom_smooth()` using method = 'loess' and formula = y ~ log_hp`

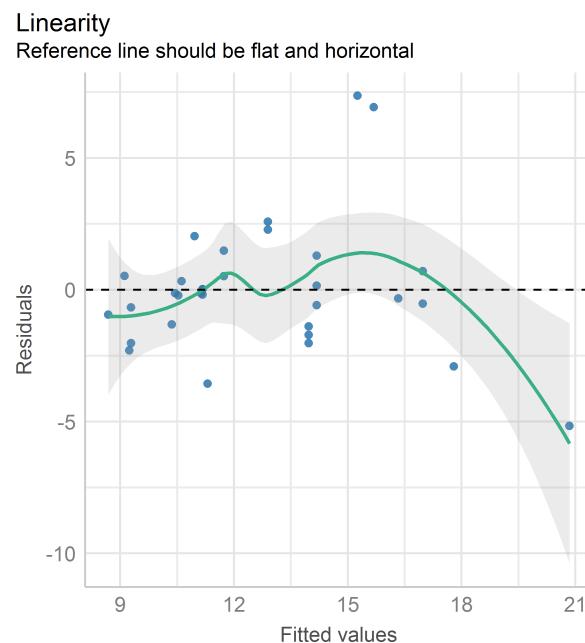


Modellannahmen

1. Lineare Beziehung

- Die Visualisierung mit einer Funktion:

```
performance::check_model(lm_obj, check = 'linearity')
```



Modellannahmen

1. Lineare Beziehung

- inferenzstatistischer Test: **Rainbow-Test**
- prinzipielles Vorgehen
 - Regressionsmodell mit den mittleren 50% der Daten berechnen (Standard)
 - auch bei nicht-linearen Beziehung idR im mittleren Bereich grobe Linearität
 - Regressionsmodell mit allen Daten berechnen
 - Vergleich der Güte der Modelle
 - Bei vergleichbarer Modellgüte kann Linearität unterstellt werden
- H_0 : lineare Beziehung besteht

```
lmtest::raintest(lm_obj, fraction = .5) %>%
  broom::tidy() %>%
  flex()
```

```
## Multiple parameters; naming those columns df1, df2
```

df1	df2	statistic	p.value	method
16	14	1.3	p = 0.315	Rainbow test

Modellannahmen

2. Normalverteilung der Residuen

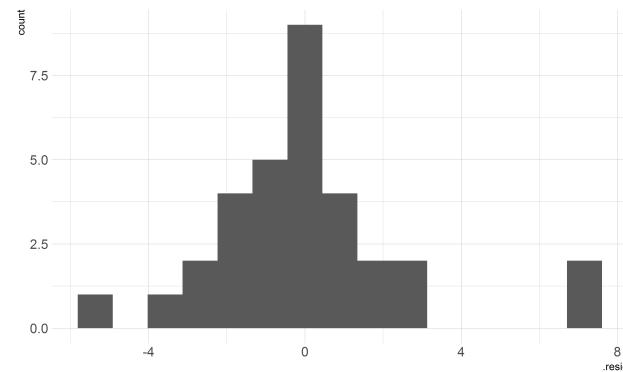
- Der Mittelwert der Residuen beträgt 0 (*immer*)
 - dh: im Durchschnitt “trifft” der vorhergesagte Wert den beobachteten Wert
- die Fehler sollen dabei aber gleichmäßig um die 0 streuen
 - dh. die Residuen sollen normalverteilt sein
- mathematisch: $\epsilon \sim \mathcal{N}(0, \sigma_\epsilon)$

Modellannahmen

2. Normalverteilung der Residuen

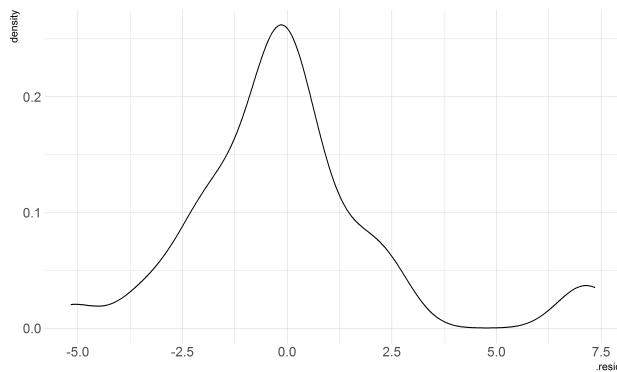
- graphische Inspektion der Residuen 1
- Visualisierung der Verteilungen
- Histogramm:

```
lm_obj %>%
  broom::augment() %>%
  ggplot(aes(x = .resid)) +
  geom_histogram(bins = 15)
```



- Dichteplot:

```
lm_obj %>%
  broom::augment() %>%
  ggplot(aes(x = .resid)) +
  geom_density()
```



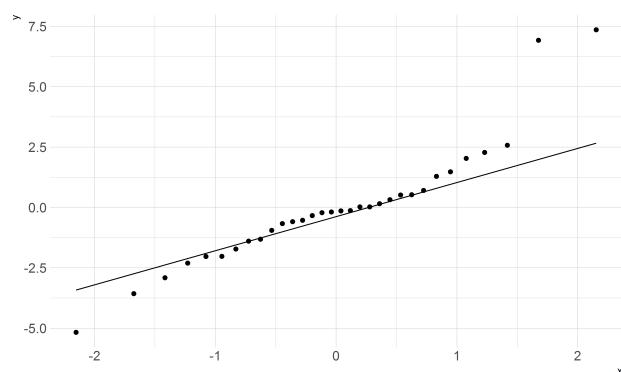
Modellannahmen

2. Normalverteilung der Residuen

- graphische Inspektion der Residuen 2
- mit QQ-Plot:

```
lm_obj %>%
  broom::augment() %>%
  ggplot(aes(sample = .resid)) +
  geom_qq(show.legend = T) +
  geom_qq_line()
```

- QQ-Plot:
 - x-Achse: theoretische Quantile
 - y-Achse: beobachtet Quantile
 - bei NV liegen die Punkte auf einer Geraden

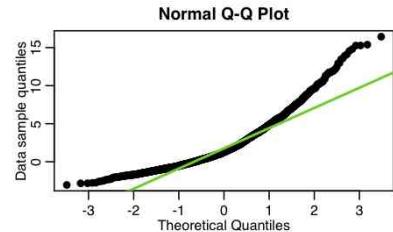
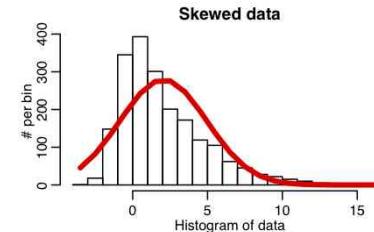
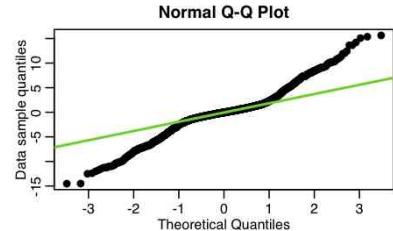
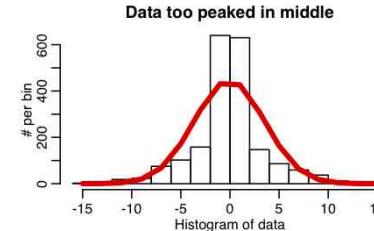
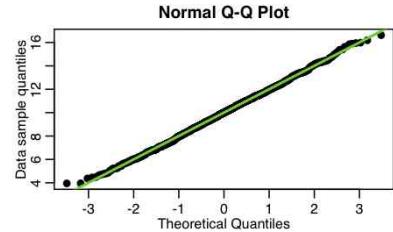
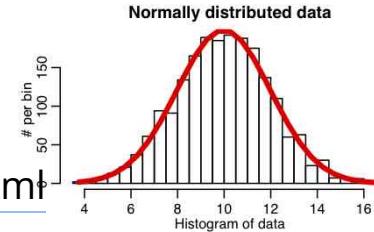


Modellannahmen

2. Normalverteilung der Residuen

- Interpretation braucht Übung!
- <https://xiongge.shinyapps.io/QQplots/>
- Detrended-QQ-Plots & PP-Plots:
<https://cran.r-project.org/web/packages/qqplotr/index.html>
- QQ-Plot > Dichteplot > Histogramm

Beispiele:



Modellannahmen

2. Normalverteilung der Residuen

- auch inferenzstatistische Prüfung möglich
- Statistischer Test: *Shapiro-Wilk-Test*

```
lm_obj %>%
  broom::augment() %>%
  rstatix::shapiro_test(.resid) %>%
  flex()
```

variable	statistic	p
.resid	0.89	0

- $H_0 : X \sim \mathcal{N}$... X ist normalverteilt
- $H_1 : X \not\sim \mathcal{N}$... X ist nicht normalverteilt
 - ✓ $p > 0,05$... Normalverteilungsannahme nicht widerlegt
- Kurzform in R: `performance::check_normality(lm_obj)`
- Test wird bei großen Stichproben zu oft signifikant

Modellannahmen

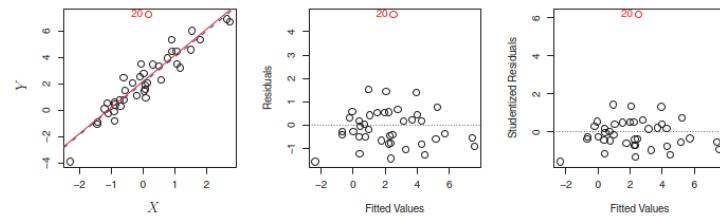
2. Normalverteilung der Residuen

- verletzte NV-Annahme der Residuen
- 1 Ignorieren
 - ab ca. $N = 100$
 - gewisse Robustheit Linearen Regression
- 2 Gegenmaßnahmen
 - Elimination von Ausreißern
 - Transformation des Kriteriums
 - (Verwendung der Bootstrap-Technik)
 - (Verwendung eines generalisierten linearen Modells)

Modellannahmen

2. Normalverteilung der Residuen

- Elimination von Ausreißern
 - Ausreißer:
 - vorhergesagter Y-Wert (\hat{Y}) weicht erheblich von beobachtetem Wert (Y) ab
 - sehr großes Residuum
 - Konsequenz:
 - deutlich verringerte Modellgüte
 - zu große Standardfehler
- Beispiele:



Modellannahmen

2. NV Resi - Ausreißerdiagnostik

- ab wann ist ein Residuum "zu groß"?
 - **standardisiertes Residuum:** $\epsilon_{i,std} = \epsilon_i / \hat{SE}_{\epsilon_i}$
 - Division des Residuums durch seinen Standardfehler
 - Vgl. z-Standardisierung, Allg. Wald-Statistiken
 - "internally studentized residual"
- Motivation:
 - Residuen sind unmittelbar von den Werten abhängig
 - durch die Division entsteht ein Dimensionsloses Maß
 - Analogie zu allgemeiner Teststatistik: Parameter / SE(Parameter)
- kritisch: $|Werte| > 2-3$
- enthalten in `broom::augment() : .std.resid`
- oder: `rstandard(lm_obj)`
- weitere Ausreißerdiagnostik & Residuenarten folgen...

Modellannahmen

2. NV Resi - Ausreißerdiagnostik

- Führen Sie die Regression unter Ausschluss der kritischen Beobachtungen erneut durch. Wie sind nun die Residuen verteilt? Welche Ergebnisse zeigt nun der Shapiro-Wilk-Test?
- Neuberechnung

```
lm_obj_wo_outliers <-
  df_cars %>%
  rownames_to_column('Automodell') %>%
  filter(!Automodell %in%
    c('Maserati Bora',
      'Lincoln Continental',
      'Cadillac Fleetwood')) %>%
  lm(lkm ~ hp, data = .)
```

- Verteilung der Residuen

```
lm_obj_wo_outliers %>%
  broom::augment() %>%
  ggplot(aes(.std.resid)) +
  geom_density()
```

- Test

```
lm_obj_wo_outliers %>%
  broom::augment() %>%
  rstatix::shapiro_test(.std.resid)
```

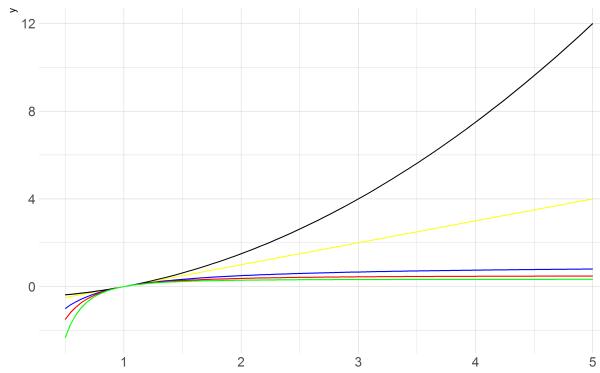
- Plot

```
lm_obj_wo_outliers %>%
  broom::augment() %>%
  ggplot(aes(hp, .std.resid)) +
  geom_point()
```

Modellannahmen

2. NV Resi - Transformation

- Transformation des Kriteriums: **Box-Cox-Transformation**
- Verteilung der Residuen normalisieren
- Regressand Y wird potenziert mit einem Wert (lambda)
 - $Y' = \frac{Y^\lambda - 1}{\lambda}$, für $\lambda \neq 0$
 - $\ln(Y)$, für $\lambda = 0$
- Wirkung:



Modellannahmen

2. NV Resi - Transformation

- Box-Cox-Transformation
- Berechnung:
 - Maximum-Likelihood-Schätzung
 - `MASS::boxcox(lm_obj)`
 - Punktschätzer und 95%-KI
 - `bc <- MASS::boxcox(lm_obj, plotit = F)`
 - x = Lambda-Wert, y = log-Likelihood
 - Wert des maximalen Likelihoods bestimmen
 - `lambda <- bc$x[which.max(bc$y)]`
 - wenn 0 im KI -> Logarithmieren
 - wenn 1 im KI -> keine Transformation
 - Y-transformieren
 - `y_trans = (y^lambda - 1) / lambda`

Modellannahmen

2. NV Resi - Transformation

- **Box-Cox-Transformation**
- Einschränkungen:
 - nur für ausschließlich positive y-Werte definiert (auch keine 0!)
 - ggf. minimal geshiftete Version des Kriteriums verwenden
 - Erweiterung um zweiten Parameter, dann auch Anwendung auf negative Werte möglich
 - Interpretation ändert sich entsprechend
 - Rücktransformation für Interpretation hilfreich
 - $Y = (Y' \times \lambda + 1)^{(1/\lambda)}$, für $\lambda \neq 0$
 - $Y = \exp(Y')$, für $\lambda = 0$

Modellannahmen

2. NV Resi - Transformation

- Unsere Residuen sind ja nicht normalverteilt:

```
- lm_obj %>% broom::augment() %>% shapiro_test(.std.resid)
```

- Führen Sie die Box-Cox-Transformation durch und wiederholen Sie die lineare Regression. Wie fällt nun der Test auf NV der Residuen aus?

```
#boxcox <- MASS::boxcox(lm_obj)
df_cars %>%
  mutate(lkm_log = log(lkm)) %>%
  lm(lkm_log ~ hp, data = .) %>%
  broom::augment() %>%
  rstatix::shapiro_test(.std.resid) %>%
  flex()
```

variable	statistic	p
.std.resid	0.97	0.56

Modellannahmen

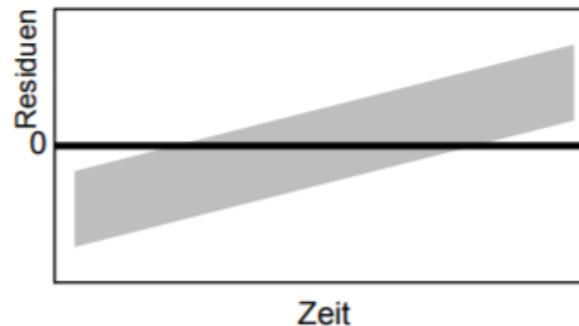
3. keine Autokorrelation

- die Residuen aufeinanderfolgender Beobachtungen ($\epsilon_1, \epsilon_2, \dots$) müssen unkorreliert sein
- besonders relevant bei Zeitreihendaten
- mathematisch: $\epsilon_{i+1} \perp \epsilon_i$
- ... das Residuum der Beobachtung i enthält keine Information über das Residuum der darauf folgenden Beobachtung $i+1$
- Unabhängigkeit der Residuen
- Konsequenz bei verletzter Annahme:
 - Standardfehler werden unterschätzt
 - Konfidenzintervalle zu schmal
 - zu kleine p-Werte

Modellannahmen

3. keine Autokorrelation

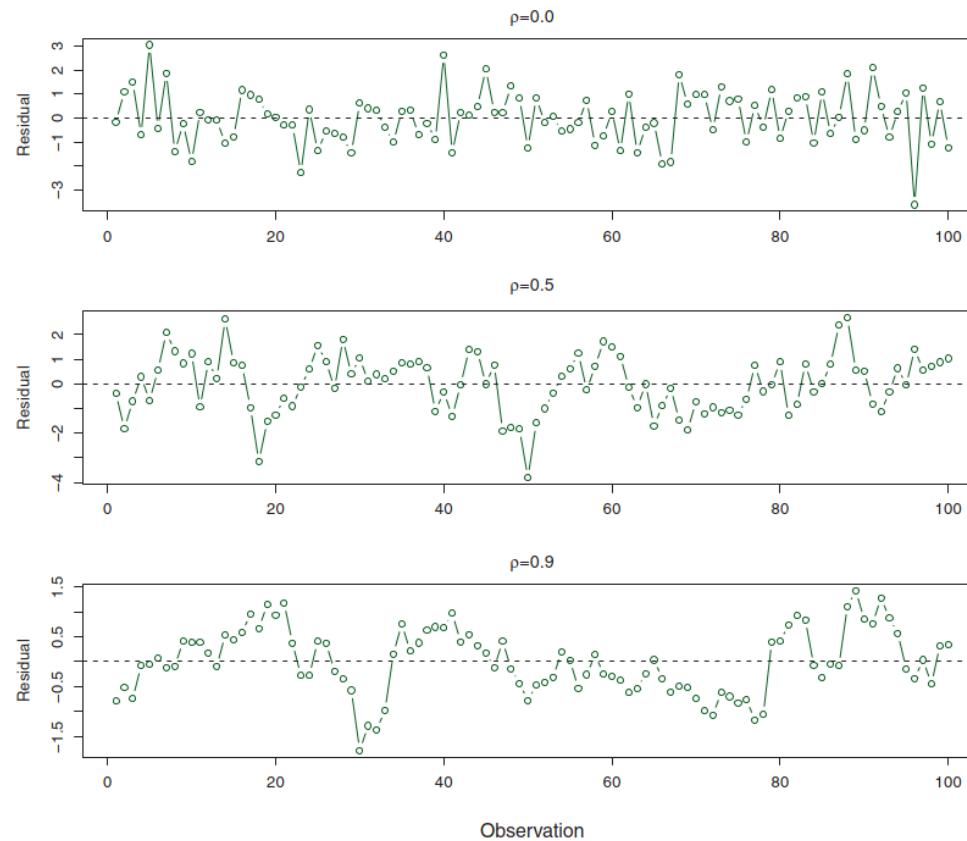
- insbesondere bei Zeitreihendaten problematisch
- aber auch bei Clusterdaten, z.B.
 - mehrere Kinder einer Familie
 - Personen eines Stadtviertels
 - mehrere Patienten eines Chirurgen
- Diagnostik:
 - Streudiagramm: Residuen (Y) als Funktion der Zeit (X)
 - keine deutlichen Muster erkennbar



Modellannahmen

3. keine Autokorrelation

- Muster sind ggf. schwer identifizierbar:



Modellannahmen

3. keine Autokorrelation

- Berechnung der Autokorrelation:

- `residuals(lm_obj) %>% acf(plot = F)`

- Plotten der Autokorrelation

- `residuals(lm_obj) %>% acf()`
 - “blaue Linien” ... Signifikanzschwelle

- ⚡ keine Sortierung der Werte!

- `residuals(lm_obj) %>% sort %>% acf()`

Modellannahmen

3. keine Autokorrelation

- inferenzstatistisch: **Durbin-Watson-Test**
 - prüft Autokorrelation mit lag 1 (Autokorrelation 1. Ordnung)
 - $H_0 : \rho \leq 0$; $H_A : \rho > 0$
- mit `lmtest`

```
lmtest::dwtest(lm_obj) %>%  
  broom::tidy() %>%  
  flex()
```

statistic	p.value	method	alternative
0.97	p < .001	Durbin-Watson test	true autocorrelation is greater than 0

- mit `car` (bootstrapped p-Wert!)

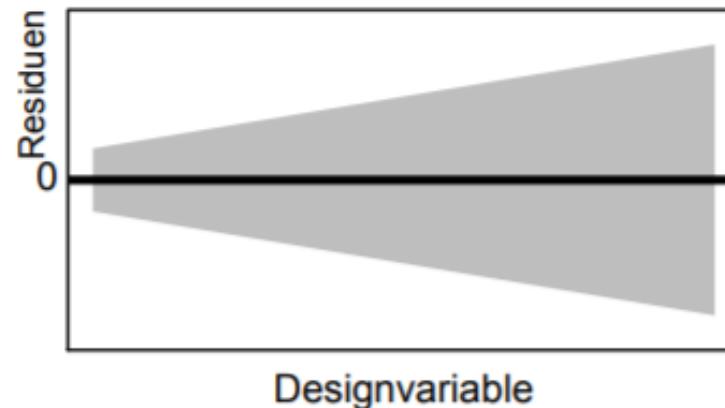
```
car::durbinWatsonTest(lm_obj, alternative="positive") %>%  
  broom::tidy() %>%  
  flex()
```

statistic	p.value	autocorrelation	method	alternative
0.97	p = 0.001	0.52	Durbin-Watson Test	positive

Modellannahmen

4. Homoskedastizität

- = gleiche Varianz der Residuen über den Bereich der vorhergesagten Werte (Varianzhomogenität)
- Das Modell ist über die Breite der Regressor-Variable gleich gut/schlecht
- Schätzfehler unabhängig vom Wert des Regressors
- Begriff der Forderung: *Homoskedastizität*
- Verletzung: *Heteroskedastizität*
 - im Residuenplot: z.b. Trichterform erkennbar



Modellannahmen

4. Homoskedastizität

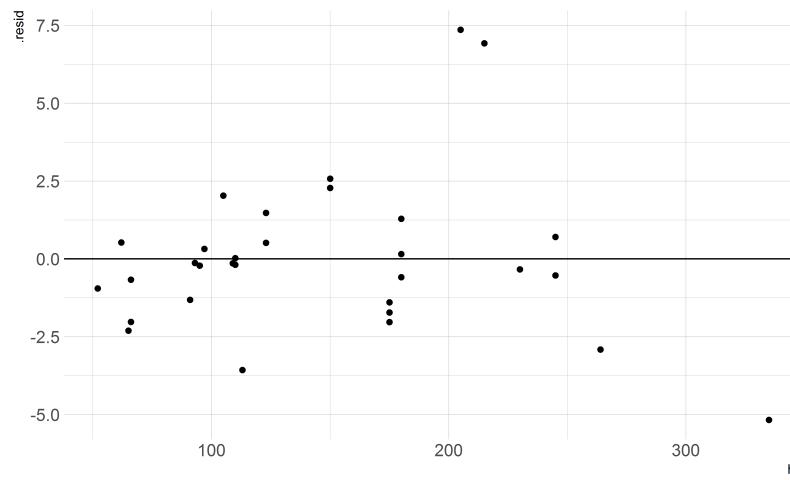
- Konsequenzen Heteroskedastizität:
 - zu große Standardfehler zu den Koeffizienten
 - Reg.koeffizienten iA. korrekt, aber uU. inkonsistente Schätzung
- daraus kann resultieren:
 - fehlerhafte Signifikanztests
 - falsche Konfidenzintervalle
- Muster:
 - milde Heteroskedastizität -> keine gravierenden Auswirkungen
 - kleine Fehlervarianz bei extremen Regressorwerten -> überschätzte Standardfehler, zu breite KIs
 - Zunahme der Fehlervarianz mit Regressorwerten -> unterschätzte Standardfehler, zu schmale KIs
- mögliche Ursachen:
 - mit Kriteriumswert wachsender Messfehler
 - Modellspezifikationsfehler (zB. nicht modellierte Interaktion)

Modellannahmen

4. Homoskedastizität

- Visuelle Diagnostik: *Residuenplot*
 - x = Regressorwert
 - y = Residuum
- Plot:

```
broom:::augment(lm_obj) %>%
  ggplot(aes(hp, .resid)) +
  geom_point() +
  geom_abline(intercept = 0, slope = 0)
```



Modellannahmen

4. Homoskedastizität

- Visuelle Diagnostik: *Spread & Level-Plot*
 - x = logarithmierte Modellprognosen
 - y = logarithmierte Beträge der Residuen
- Plot-Daten:

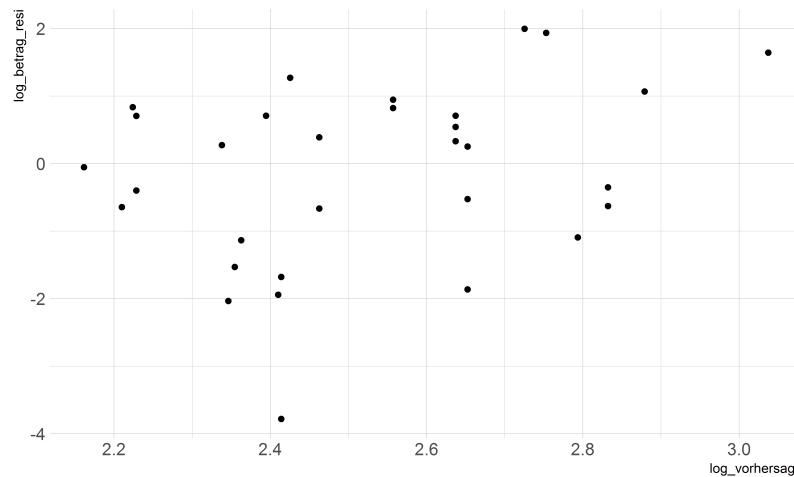
```
spread_level_data <-  
  broom::augment(lm_obj) %>%  
  mutate(log_vorhersage = log(.fitted)) %>%  
  mutate(log_betrag_resi = log(abs(.resid)))
```

Modellannahmen

4. Homoskedastizität

- *Spread & Level-Plot*

```
spread_level_data %>%
  ggplot(aes(x = log_vorhersage, y = log_betrag_resi)) +
  geom_point()
```



- Details:
 - Ausschluss von Fällen mit einem Prognosewert ≤ 0 (kein Logarithmus definiert)
 - nimmt Residualvarianz mit dem prognostizierten Wert zu (vgl. "Trichterform"), dann steigt die Fehlervarianz mit zunehmendem Vorhersagewert

Modellannahmen

4. Homoskedastizität

- *Spread & Level-Plot*
- Lässt sich nun ein signifikanter Zusammenhang feststellen?
- Regression der logarithmierte Beträge der Residuen (y) auf die logarithmierten Modellprognosen (x)
- Überprüfung des Regressionskoeffizienten
- Berechnung:

```
lm(log_betrag_resi ~ log_vorhersage, data = spread_level_data) %>%  
  broom::tidy() %>%  
  flex()
```

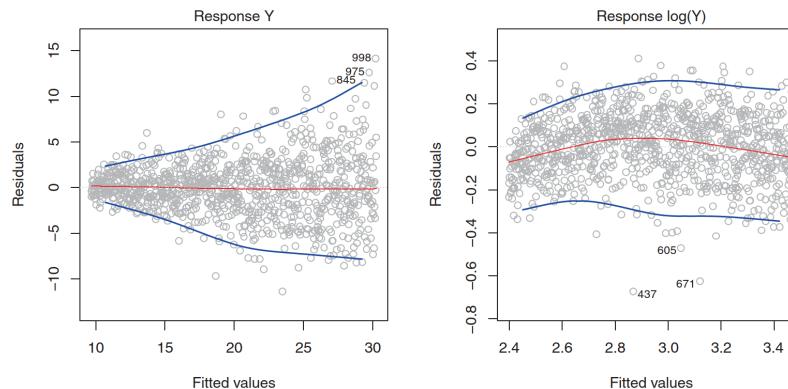
term	estimate	std.error	statistic	p.value
(Intercept)	-5.31	2.87	-1.85	p = 0.074
log_vorhersage	2.01	1.13	1.78	p = 0.086

- ✓ Steigungskoeffizient ist 2.06, aber nicht signifikant -> keine Heteroskedastizität

Modellannahmen

4. Homoskedastizität

- und wenn der Steigungskoeffizient signifikant ist?
- also wenn *Heteroskedastizität* besteht
- Lösung: *Fehlervarianz-stabilisierende Transformation* des Kriteriums
 - Allgemein: \sqrt{Y} ; $\log Y$; $\frac{1}{Y}$
 - spezifisch: $Y_{korr} = Y^{1-b_1}$, mit b_1 = RK der Steigung aus Spread & Level-Plot
- Auswirkung der Transformation Beispiel:



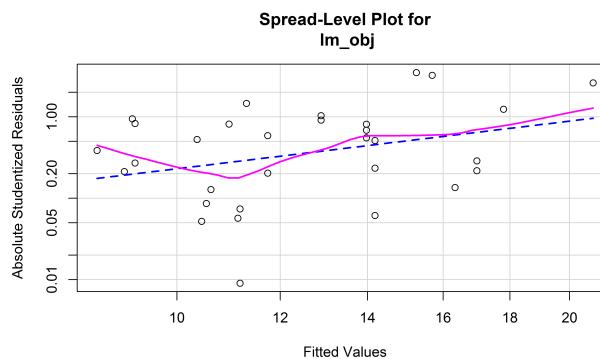
- ⚠ Transformationen können sich auch auf Linearität und/oder Residuenverteilung auswirken

Modellannahmen

4. Homoskedastizität

- *Spread & Level-Plot*, einfach
- in Kurzform:

```
sldat <- car::spreadLevelPlot(lm_obj)
```



- Exponent

```
sldat$PowerTransformation
```

- dieser wird bereits so berichtet, dass er direkt als Exponent verwendet werden kann
- $Y_{korr} = Y^{1-b_1}$; laut unserer Berechnung: $1-2.06 = -1.06$

Modellannahmen

4. Homoskedastizität

- Da die Homogenitätsannahme nur knapp eingehalten wird ($p = 0.086$), führen Sie die Regression noch einmal mit dem transformierten Kriterium durch. Inspizieren Sie die diagnostischen Plots.
- Vorgehen:
 - Steigungskoeffizient im obigen Spread & Level-Plot: 2.013
 - Korrektur: $Y_{korr} = Y^{1-b_1}$
 - Regression neu berechnen

```
# transformation & neuberechnung der Regression
lm_obj2 <-
  df_cars %>%
  mutate(lkm_korr = lkm^(1-2.013)) %>%
  lm(lkm_korr ~ hp, data = .)
```

Modellannahmen

4. Homoskedastizität

- auf diesem Ergebnis die Spread & Level-Plot-Werte berechnen
- Spread & Level-Plot: $x = \log(\text{.fitted})$, $y = \log(\text{abs}(\text{.resid}))$

```
# Berechnung der Werte für S-L-Plot
slp_data <-
  lm_obj2 %>%
  broom::augment() %>%
  mutate(log_vorhersage = log(.fitted)) %>%
  mutate(log_betrag_resi = log(abs(.resid)))
```

- mit diesen Werten eine neue Regression durchführen

```
# Regression mit S-L-Plot-Daten
slp_data %>%
  lm(log_betrag_resi ~ log_vorhersage, data = .) %>%
  broom::tidy() %>%
  flex()
```

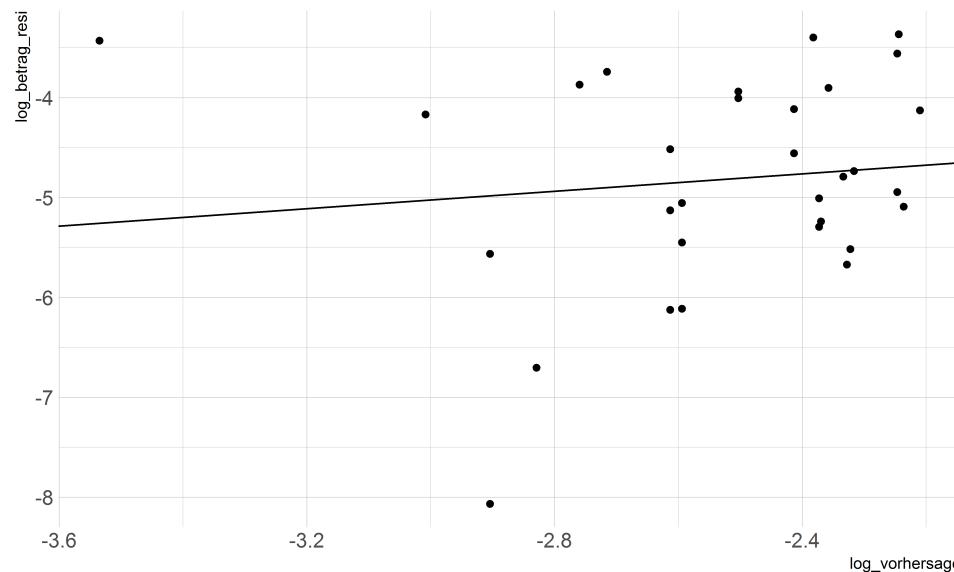
term	estimate	std.error	statistic	p.value
(Intercept)	-3.72	1.69	-2.19	p = 0.036
log_vorhersage	0.44	0.67	0.65	p = 0.518

Modellannahmen

4. Homoskedastizität

- diese Werte plotten

```
# visualisierung
slp_data %>%
  ggplot(aes(x = log_vorhersage, y = log_betrag_resi)) +
  geom_point() +
  geom_abline(intercept = -3.7168806, slope = 0.4359636)
```



Modellannahmen

4. Homoskedastizität

- bei "milder Verletzung" der Homoskedastizität, keine gravierenden Auswirkungen
- aber was ist mild?
- **Maximalquotientenkriterium**
- Prinzip: der Fehler (Residuum) ist unabhängig vom x-Wert überall ungefähr gleich groß
- Verhältnis bilden, zwischen maximaler und minimaler Fehlerstreuung
- dafür braucht es die Zusammenfassung von Bereichen des x-Wertes (Kategorisierung)
- je Bereich wird die Standardabweichung der Residuen berechnet
- Quotient: höchste / niedrigste SD berechnen
- Quotient < 1,5 ... unproblematisch
- Quotient 1.5-3 ... grenzwertig
- Quotient >3 ... inakzeptabel

```
lm_obj %>%  
  broom::augment() %>%  
  mutate(hp_gruppiert =  
         Hmisc::cut2(hp, g = 4,  
                      levels.mean=T)) %>%  
  group_by(hp_gruppiert) %>%  
  mutate(sd_e_ingruppe = sd(.resid)) %>%  
  group_by(hp_gruppiert) %>%  
  summarize(  
    MW_Gruppe = mean(hp),  
    SD_e_Gruppe = mean(sd_e_ingruppe)  
  ) %>%  
  flex()
```

hp_gruppiert	MW_Gruppe	SD_e_Gruppe
73.75	73.75	0.97
111.11	111.11	1.56
170.62	170.63	1.81
248.43	248.43	4.71

Modellannahmen

4. Homoskedastizität

- teststatistisches Verfahren: *Breusch-Pagan-Heteroskedastizitätstest*
- Vorgehen:
 - einfachen Residuen (e_i) quadrieren -> (e_i^2)
 - Regressionsanalyse durchführen (Y = quadrierte Residuen; X = Regressor aus dem ursprünglichen Modell)
 - unkorrigiertes R² dieses Modells mit Stichprobengröße N multiplizieren
 - das ist die Breusch-Pagan-Prüfgröße
 - diese ist Chi-Quadrat-verteilt mit df = 1

Modellannahmen

4. Homoskedastizität

- in R:

```
lm_obj %>%
  broom::augment() %>%
  mutate(resid_squared = .resid^2) %>%
  lm(resid_squared ~ hp, data = .) %>%
  broom::glance() %>%
  flex()
```

r.squared	adj.r.squared	sigma	statistic	p.value	df	logLik	AIC	BIC	deviance	df.residual	nobs
0.170.14	11.99	6.02	p = 0.020	1	-123.87	253.75	258.15	4'315.72	30	32	

- Ergebnis: $R^2 = 0.167$
- $0.167 * 32 = 5.345$
- `pchisq(5.345, df = 1)`, $p = 0.02079$

Modellannahmen

4. Homoskedastizität

-  in Kurzform: `lmtest::bptest(lm_obj)`
- Nullhypothese = Homoskedastizität
- ⚡ bei großen Stichproben wird der Test u.U. auch bereits bei minimalen Verletzungen der Homoskedastizitäts-Annahme signifikant.
- ✓ Auch graphische Methoden nutzen!

Modellannahmen

4. Homoskedastizität

- Wenn alles nichts hilft:
 - (Weighted-Least-Squares Regression)
 - (Bootstrap-Verfahren)
 - Schätzverfahren verwenden, die bezüglich der Verletzung "immun" sind:
 - sog. heteroskedastizitätskonsistenten Schätzer (HCC)
 - bzw. auch robuste Schätzer
- Modelle mit robusten Standardfehlern können *immer* berechnet werden

Modellannahmen

4. Homoskedastizität

- klassisch:

```
lm_obj %>%
  broom::tidy(conf.int = T) %>%
  flex(digits = 3)
```

term	estimate	std.error	statistic	p.value	conf.low	conf.hi
(Intercept)	6.449	1.074	6.006	p < .001	4.256	8.642
hp	0.043	0.007	6.464	p < .001	0.029	0.057

- HCC-Schätzung

```
lm_obj %>%
  lmtest::coeftest(
  vcov = sandwich::vcovHC(
    lm_obj, type = "HC3")) %>%
  broom::tidy(
    conf.int = T, digits = 3) %>%
  flex(digits = 3)
```

term	estimate	std.error	statistic	p.value	conf.low	conf.high
(Intercept)	6.449	1.330	4.847	p < .001	3.732	9.166
hp	0.043	0.011	3.919	p < .001	0.021	0.065

- Hier der Fall: Zunahme der Fehlervarianz mit Regressorwerten -> unterschätzte SEs, zu schmale KIs

Modellannahmen

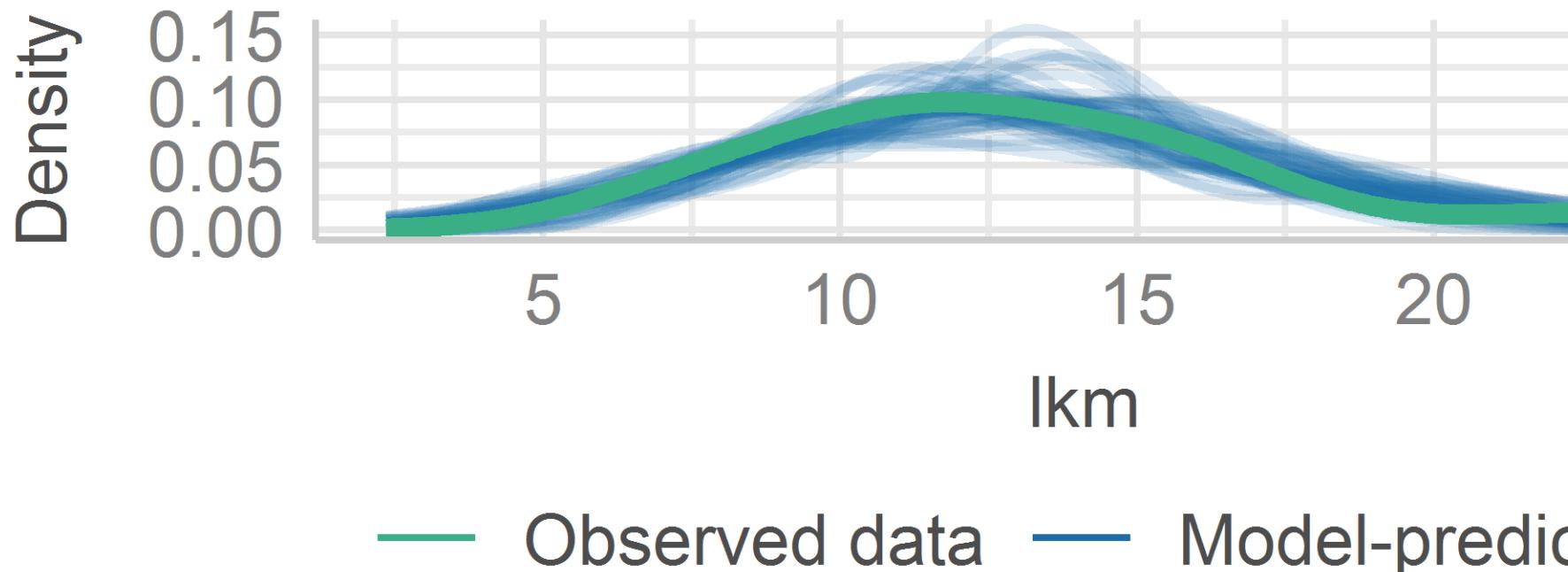
All-in-One

- visuelle Modellinspektion zusammen:

```
· performance::check_model(lm_obj)
```

Posterior Predictive Check

Model-predicted lines should resemble observed data



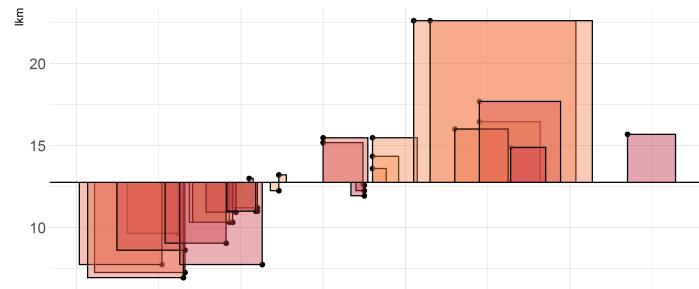
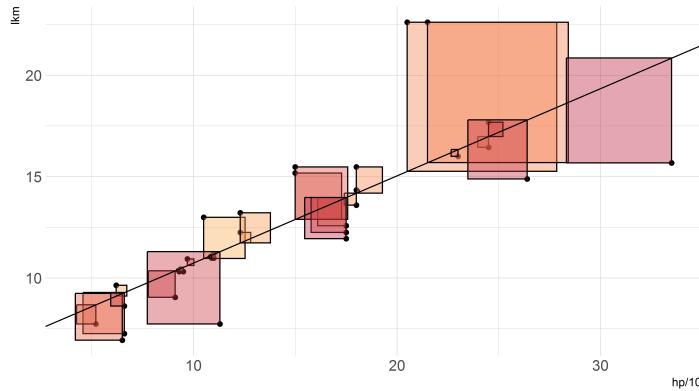
Homogeneity of Variance

Chill...



Kleinste Quadrate Schätzung

- Recap: Bestimmung der RK der RG mittels OLS-Methode:
- Abweichungsquadrate nach Regression:
 - Abweichungsquadrate (vom Mittelwert)



- $SQ_{Residuen} = 193.4$
- $SQ_{Total} = 462.7$
- vgl. GG-Varianz: $s_{GG}^2 = \frac{1}{n-1} \sum(x_i - \bar{x})$
- `var(df_cars$lkm) * 31`

Varianzzerlegung

- Varianzzerlegung: $SQ_{Total} = SQ_{Residuum} + SQ_{Regression}$

- vollständige Formel:

$$\underbrace{\sum_{i=1}^n (Y_i - \bar{Y})^2}_{\text{Variation von Y}} = \underbrace{\sum_{i=1}^n (Y_i - \hat{Y}_i)^2}_{\text{Variation der Residuen}} + \underbrace{\sum_{i=1}^n (\hat{Y}_i - \bar{Y})^2}_{\text{Variation der Regresswerte}}$$

- Ein Teil der Streuung der Y-Werte werden also durch das Regressionsmodell erklärt
- Wie groß ist dieser Anteil?

Modellgüte

Bestimmtheitsmaß R^2/B

- Determinationskoeffizient:

$$R^2 = \frac{SQ_{Regression}}{SQ_{Total}} = 1 - \frac{SQ_{Residuum}}{SQ_{Total}}$$

- ... gibt den Anteil der durch das Modell aufgeklärten Varianz an.
- liegt zwischen 0 und 1
 - 0 keine Verbesserung gegenüber der Mittelwertschätzung
 - 1 perfekte Vorhersage durch das Modell
- interpretierbar als %-Wert
- Maßzahlen für die Modellgüte:

- $SQ_{Total} = 462.7$
- $SQ_{Residuen} = 193.4$
- $SQ_{Regression} = 462.7 - 193.4 = 269.3$
- $R^2 = 269.3/462.7 = 0.58$

Modellgüte

Bestimmtheitsmaß R^2/B

- in R:

```
lm_obj %>%
  broom::glance() %>%
  flex()
```

r.squared	adj.r.squared	sigma	statistic	p.value	df	logLik	AIC	BIC	deviance	df.residual	nobs
0.58	0.57	2.54	41.79	p < .001	1	-74.19	154.37	158.77	193.35	30	32

- keine festen Interpretationsmaßstäbe
- je nach Domäne: Psychologie, Soziologie vs. Physik, Technik
- bei der einfachen Regression gilt: $r_{xy}^2 = R^2$

Modellgüte

ANOVA der Regression

- Die Quadratsummen können noch mehr...
- Prüfung der Signifikanz des Gesamtmodells mittels ANOVA
- in R:

```
lm_obj %>%
  anova() %>%
  broom::tidy() %>%
  flex()
```

term	df	sumsq	meansq	statistic	p.value
hp	1	269.31	269.31	41.79	p < .001
Residuals	30	193.35	6.45		

- ANOVA prüft zerlegte Varianzen hinsichtlich Signifikanz (*Quadratsummenzerlegung*)
- Beziehung der Prüfstatistiken (bei einfacher Regression):
 - $\sqrt{F - \text{Wert}} = |t|$
 - $\sqrt{41.78} = 6.46$

Modellgüte

ANOVA der Regression

- F-Wert direkt (*statistic*):

```
lm_obj %>%
  broom::glance() %>%
  flex()
```

r.squared	adj.r.squared	sigma	statistic	p.value	df	logLik	AIC	BIC	deviance	df.residual	nobs
0.58	0.57	2.54	41.79	p < .001	1	-74.19	154.37	158.77	193.35	30	32

Modellgüte & -genauigkeit

Übersicht

- diverse Maßzahlen und Kriterien
 - Bestimmtheitsmaß
 - ANOVA
 - *Residual Sum of Squares* (RSS)
 - *Mean Squared Error* (MSE)
 - *Root Mean Squared Error* (RMSE)
 - *Residual Standard Error* (RSE)

Modellgüte & -genauigkeit

RSS

- **Residual Sum of Squares (RSS)**
- = Gesamtsumme der Abweichungsquadrate $SQ_{Residuum}$
 - Abweichung: $e = y - \hat{y}$
 - Quadrieren: $e^2 = (y - \hat{y})^2$
 - Summe: $\text{RSS} = \sum e^2 = \sum ((y - \hat{y})^2)$
- Berechnung:

```
rss <- sum(residuals(lm_obj) ^ 2)
```

- Probleme
 - Abhängig von Fallzahl
 - Abhängig von Maßeinheit (vgl m vs. mm)
 - quadrierte Einheit

Modellgüte & -genauigkeit

MSE

- Mean Squared Error (MSE)
- = durchschnittliche Größe der Abweichungsquadrate
 - Abweichung (Error): $e = y - \hat{y}$
 - Quadriert (Squared): $e^2 = (y - \hat{y})^2$
 - Summe (Squared): $\text{RSS} = \sum e^2 = \sum ((y - \hat{y})^2)$
 - Mittelwert (Mean): $\text{MSE} = \text{RSS}/n$
- Berechnung:

```
mse <- mean(residuals(lm_obj)^2)
mse <- rss/nobs(lm_obj)
```

- Probleme
 - ✓ ~~Abhängig von Fallzahl~~
 - Abhängig von Maßeinheit (vgl m vs. mm)
 - quadrierte Einheit

Modellgüte & -genauigkeit

RMSE

- Root Mean Squared Error (RMSE)
- = Wurzel der durchschnittlichen Größe der Abweichungsquadrate
 - Abweichung (Error): $e = y - \hat{y}$
 - Quadriert (Squared): $e^2 = (y - \hat{y})^2$
 - Summe (Squared): $\text{RSS} = \sum e^2 = \sum ((y - \hat{y})^2)$
 - Mittelwert (Mean): $\text{MSE} = \text{RSS}/n$
 - Wurzel (Root): $\text{RMSE} = \sqrt{\text{MSE}}$
- Berechnung:

```
rmse <- sqrt(mean(residuals(lm_obj)^2))
rmse <- sqrt(mse)
```

- Probleme
 - ✓ ~~Abhängig von Fallzahl~~
 - Abhängig von Maßeinheit (vgl m vs. mm)
 - ✓ ~~quadratierte Einheit~~

Modellgüte & -genauigkeit

RSE

- Residual Standard Error
- = ✓ Standardabweichung der Residuen
 - $RSE = \sqrt{\frac{RSS}{n-2}}$
 - Vgl: $\epsilon \sim \mathcal{N}(0, \sigma)$
- ✎ Berechnung:

```
rse <- sqrt( sum(residuals(lm_obj)^2) / lm_obj$df.residual )
sigma(lm_obj)
broom::glance(lm_obj)$sigma
```

- interpretierbar als die durchschnittliche Abweichung der Beobachtungen von der Schätzung (Regressionsgrade)
- Probleme
 - ✓ ~~Abhängig von Fallzahl~~
 - Abhängig von Maßeinheit (vgl m vs. mm)
 - ✓ ~~quadrierte Einheit~~

Modellgüte & -genauigkeit

relativer RSE

- um Abhängigkeit von Maßeinheit zu beheben
- Ins Verhältnis zum Mittelwert setzen

```
• sigma(lm_obj) / mean(df_cars$1km)
```

```
## [1] 0.1990364
```

- prozentualer Fehler bei der Vorhersage ca. 20%
- Probleme
 - ✓ ~~Abhängig von Fallzahl~~
 - ✓ ~~Abhängig von Maßeinheit (vgl m vs. mm)~~
 - ✓ ~~quadrierte Einheit~~

Modellgüte

Zusammenfassung in R

-  `broom::glance(lm_obj)`
- viele wichtige Gütemaße sind in der `broom::glance()`-Funktion vereint:
 - `r.squared` ... Bestimmtheitsmaß / Determinationskoeffizient (R^2)
 - `sigma` ... Std. Abweichung der Residuen ($\epsilon \sim \mathcal{N}(0, \hat{\sigma}_\epsilon)$)
 - `statistic` ... F-Wert der ANOVA
 - `p.value` ... p-Wert der ANOVA
 - `deviance` ... Summe der Abweichungsquadrate ($SQ_{Residuum}$)

Chill...



Kategoriale Prädiktoren

- Prädiktoren können auch kategoriale Variablen sein
- Berücksichtigung im Modell mittels sog. *Dummy-Kodierung*
 - für jede Ausprägung der kategorialen Variable wird eine neue Variable mit "0/1"-Kodierung erstellt
- Führen Sie eine lineare Regression mit der Zylinderzahl durch (*cyl*).
 - `table(df_cars$cyl)`
 - `df_cars %>% group_by(cyl) %>% get_summary_stats(lkm)`
 - `lm(lkm ~ cyl, data = df_cars) %>% summary()`
 - Was ist das Problem?
 - Die Variable muss entweder eine Zeichenfolge sein oder als kategoriale Variable deklariert werden!
 - `factor()`

Kategoriale Prädiktoren

- Berechnung einer neuen Variable mittels Kombination von `mutate()` und `factor()`.

```
df_cars %>%
  mutate(cyl_factor = factor(cyl)) %>%
  lm(lkm ~ cyl_factor, data = .) %>%
  tidy() %>%
  flex()
```

term	estimate	std.error	statistic	p.value
(Intercept)	9.05	0.70	13.01	p < .001
cyl_factor6	2.92	1.12	2.62	p = 0.014
cyl_factor8	7.01	0.93	7.54	p < .001

Kategoriale Prädiktoren

- Wenn die kategoriale Variable k Ausprägungen hat werden $k-1$ Dummy-Variablen erstellt
- Die entfallende Ausprägung wird als *Referenzkategorie* bezeichnet.
- in R inspizieren:

```
df_cars %>%
  mutate(cyl_factor = factor(cyl)) %>%
  lm(lkm ~ cyl_factor, data = .) %>%
  model.matrix()
```

Kategoriale Prädiktoren

- Wert der Referenzkategorie wird durch den Intercept repräsentiert
- Interpretation:
 - $\hat{y} = b_0 + b_1 \times X_1 + b_2 \times X_2$
 - $\hat{y} = 9.05 + 2.92 \times X_1 + 7.01 \times X_2$
 - bei `cyl = 4` sind $X_1 = 0$ und $X_2 = 0$
 - für ein Fahrzeug mit `cyl = 4`, gilt somit:
 - $\hat{y} = 9.05 + 2.92 \times 0 + 7.01 \times 0 = 9.05$
 - für ein Fahrzeug mit `cyl = 8`, gilt somit:
 - $\hat{y} = 9.05 + 2.92 \times 0 + 7.01 \times 1 = 16.06$

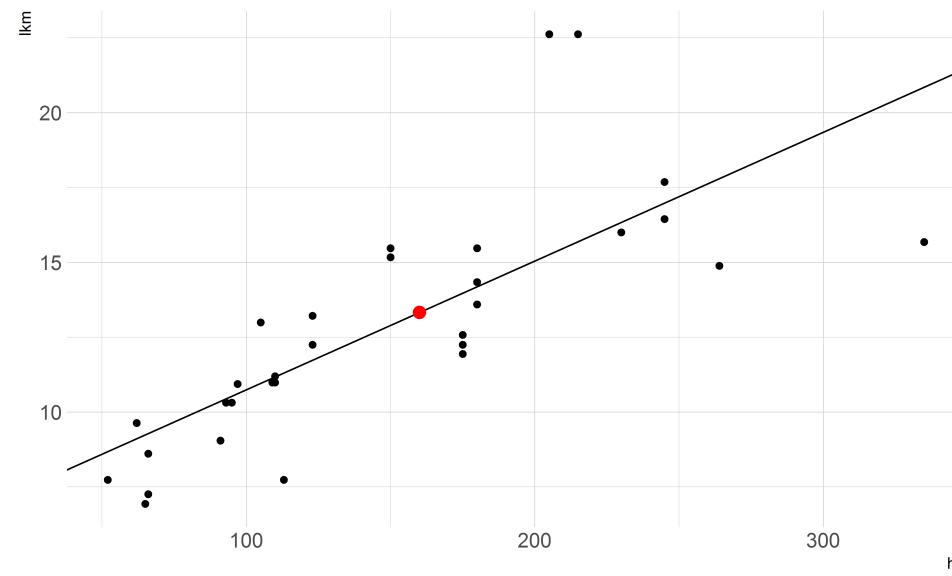
Prädiktion

- bisher: `fitted()` und `augment()`
- Das Modell zur Vorhersage neuer/unbekannter Werte nutzen
- prinzipielles Vorgehen:
 - Einsetzen der konkreten beobachteten Werte der Beobachtungseinheit in Regressiongleichung
 - `broom::tidy(lm_obj)`
 - $\hat{y}_1 = b_0 + b_1 \times x_1$
 - $\hat{y} = 6.45 + 0.043 \times \text{hp}$
 - zB Auto mit 160 PS: (unbeobachteter Wert)
 - $\hat{y} = 6.45 + 0.043 \times 160 = 13.33$

Prädiktion

- $\hat{y} = 6.45 + 0.043 \times 160 = 13.33$
- Visualisierung:

```
df_cars %>%
  ggplot(aes(hp, lkm)) +
  geom_point() +
  geom_abline(intercept = 6.45, slope = 0.043) +
  geom_point(aes(x = 160, y=13.33), color = 'red', size = 3)
```



Prädiktion

- umständlich
- in R:

- `predict()`

```
predict(  
  object = lm_obj,  
  newdata = data.frame(hp = c(160, 170))  
)
```

- Ergebnis: *Punktschätzer*

Prädiktion

- die `predict`-Ausgabe ist ein profaner Vektor
- das Gegenteil von `tidy`
- also eleganter:

```
tibble(hp = c(111, 72, 155, 245.5)) %>%
  mutate(vorhersage =
    predict(
      object = lm_obj,
      newdata = .)
  ) %>%
  flex()
```

hp	vorhersage
111.0	11.22
72.0	9.54
155.0	13.11
245.5	17.00

- ⚡ ABER: Schätzungen sind immer mit Unsicherheit behaftet

Prädiktion

- Quantifizierung der Unsicherheit durch Konfidenzintervallen (KI):
- KI geben jeweils Bereiche an, in denen der wahre Wert des Schätzers mit einer bestimmten Wahrscheinlichkeit liegt
- dabei existieren 2 Arten von Unsicherheit hinsichtlich der Vorhersage:
 - **Konfidenz:** bezogen auf den *durchschnittlichen Verbrauch* aller Autos mit 160 PS
 - dh. in welchem Bereich der durchschnittliche (*wahre*) Verbrauch aller Autos mit 160 PS (Mittelwert)
 - **Prädiktion:** bezogen auf den Verbrauch in dem ein einzelnes Auto mit 160 PS hat
 - dh. in welchem Bereich liegt der Verbrauch *eines einzelnen* Autos mit 160 PS
- entsprechende Argumente in `predict()`-Funktion:
- Konfidenz:
- Prädiktion

```
predict(  
  object = lm_obj,  
  newdata = data.frame(hp = 160),  
  interval = 'confidence'  
)
```

```
predict(  
  object = lm_obj,  
  newdata = data.frame(hp = 160),  
  interval = 'prediction'  
)
```

Prädiktion

- Vergleich der beiden Intervalle:
- Konfidenz:

```
predict(  
  object = lm_obj,  
  newdata = data.frame(hp = 160),  
  interval = 'confidence'  
)
```

- Prädiktion

```
predict(  
  object = lm_obj,  
  newdata = data.frame(hp = 160),  
  interval = 'prediction'  
)
```

- **Konfidenzintervalle** geben den Bereich an, in dem der *wahre* Wert in der Grundgesamtheit liegt
- **Prädiktionsintervalle** geben den Bereich an, in dem eine *einzelne* Beobachtungseinheit liegt
- beide Angaben sind immer mit gewisser Unsicherheit behaftet (idR 5%, Alpha)
- Prädiktionsintervalle sind *immer* breiter als Konfidenzintervalle
- zusätzliche Berücksichtigung der Variation der Beobachtungseinheiten

Prädiktion

- Vorsicht bei der Verwendung ungewöhnlicher Werte
 - zuverlässige Schätzung nur im *Stützbereich* der Regression
 - Stützbereich: $[x_{min}, x_{max}]$
 - darüber hinaus: *Extrapolation*

Standardisierte Regressionskoef.

- werden Kriterium und Regressor(en) vor der Regression standardisiert, erhält man die sog. *Standardisierten Regressionskoeffizienten*
- Standardisierung: **z-Transformation**
 - $$z = \frac{x - \bar{x}}{s_x}$$
 - Kombination aus *Verschiebung* und *Stauchung/Streckung*
- Ergebnis:
 - Mittelwert ist nun 0
 - Standardabweichung ist 1

Standardisierte Regressionskoef.

- Zur Erinnerung: $z = \frac{x - \bar{x}}{s_x}$
- z-Transformieren Sie die lkm-Variable!
 - Hinweis: `mean` und `sd`

```
mean(df_cars$lkm)
sd(df_cars$lkm)
df_cars %>% mutate(lkm_scaled = (lkm-12.75503)/3.863242)
```
- in R: `scale(x)` `df_cars %>% mutate(lkm_scaled = scale(lkm))`
- lassen Sie sich die statistischen Kennwerte der transformierten Variable anzeigen.
 - `df_cars %>% mutate(lkm_scaled = scale(lkm)) %>%
get_summary_stats(lkm, lkm_scaled)`

Standardisierte Regressionskoef.

- Standardisieren Sie Kriterium und Regressor und führen Sie eine Regression mit den standardisierten Variablen durch.

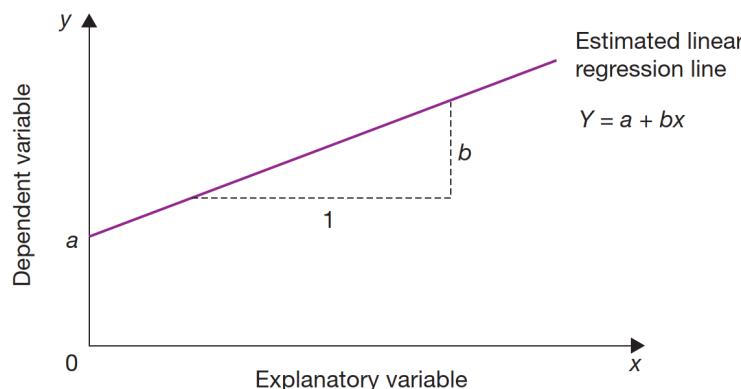
```
df_cars %>%
  mutate(lkm_scaled = scale(lkm)) %>%
  mutate(hp_scaled = scale(hp)) %>%
  lm(lkm_scaled ~ hp_scaled, data = .) %>%
  broom::tidy() %>% flex()
```

term	estimate	std.error	statistic	p.value
(Intercept)	0.00	0.12	0.00	p > .999
hp_scaled	0.76	0.12	6.46	p < .001

- Vergleichen Sie die Ausgaben dieser Regression mit dem Originalmodell.
 - b_0 wird 0
 - p-Wert des Regressors bleibt unverändert
- Vergleichen Sie den Std. Regressionskoef. mit der bivariaten Korrelation. Was fällt Ihnen auf?
 - `cor(df_carslkm, df_carshp)`
 - bei bivariater Regression ist der Wert identisch mit der Pearson-Korrelation.

Standardisierte Regressionskoef.

- Veränderung der Interpretation der Koeffizienten:
 - Veränderung in x entspricht nun der Veränderung um genau eine SD
 - die entsprechende Veränderung in y ist nun auch in SD angegeben



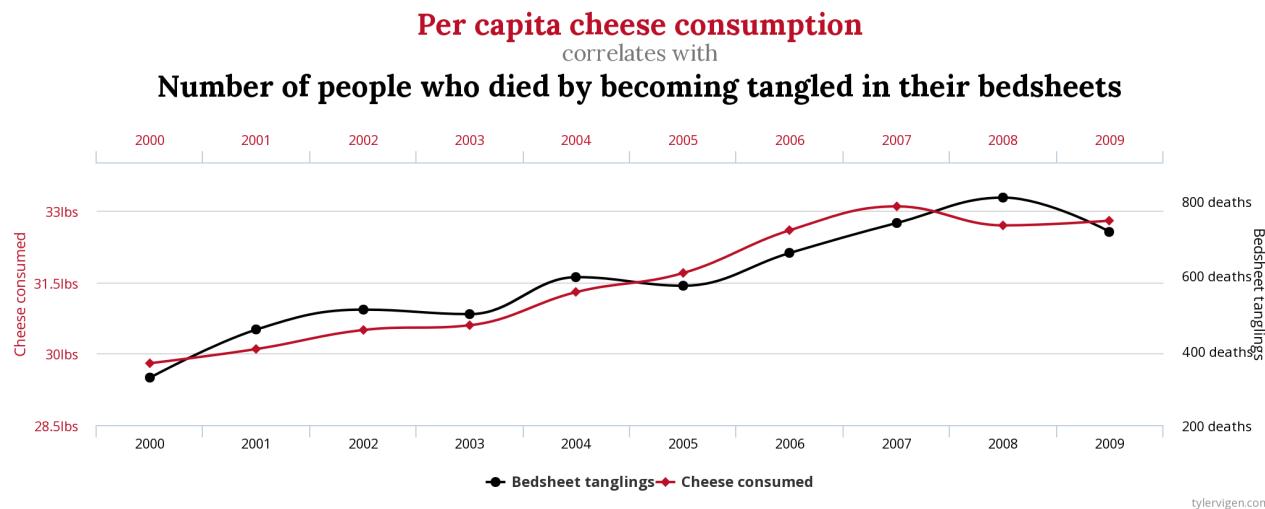
- Direkt aus gefittetem Modell: `lm.beta::lm.beta(lm_obj)`
- Relevanz:
 - Bei Variablen mit Maßeinheiten nicht zu empfehlen
 - bei "einheitsfreier" Messung (zB. Kreativität, Soz. Status) ggf. hilfreich
- **⚠ Abhängig von Streuung in Stichprobe!**
 - weniger Varianz -> kleinere Koeffizienten
 - insbesondere nur bei Populationsrepräsentativen Stichproben

Irrtümer

- Der Regressor muss eine bestimmte Verteilung haben!
 - ✓ Es wird keine Annahme über die Verteilung des Regressors benötigt.
- Das Kriterium muss einer bestimmten Verteilung folgen (idR Normalverteilung)!
 - ✓ Es wird keine Annahme über die Randverteilung des Kriteriums benötigt. Die Normalverteilungsvoraussetzung besteht für die Residuen des Modells.
- Regressor und Kriterium müssen gemeinsam bivariat normalverteilt sein!
 - ✓ Es ist keine bivariate Normalverteilung von Regressor und Kriterium erforderlich.

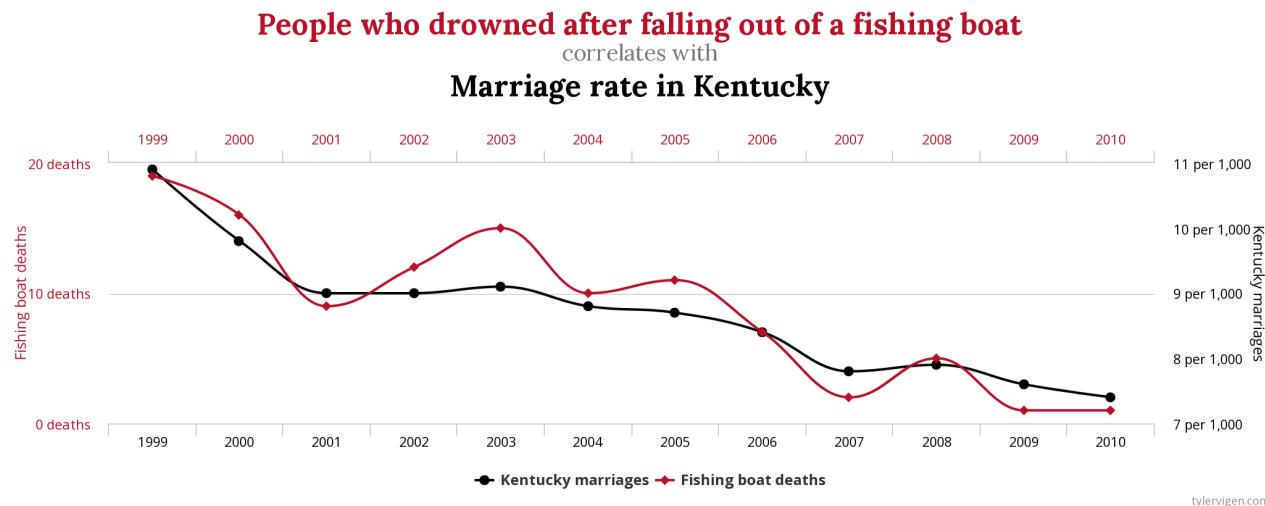
Kausalität

- Korrelation \neq Kausalität
- Prädiktion \neq Kausalität
- Beispiel:



Kausalität

- Beispiel:



Quiz 1

- Welches multiple Bestimmtheitsmaß R^2 kann man bei der Durchführung einer einfachen linearen Regression zur Vorhersage einer Kriteriumsvariable Y durch einen Prädiktor X erwarten, wenn die Korrelation zwischen dem Prädiktor und dem Kriterium 0.7 beträgt?
- 1.40
- 0.70
- 0.49
- 0.35
- ✓ 0.49

Quiz 2

Gegeben sei ein Datensatz mit zwei Variablen. Variable A soll mithilfe einer einfachen linearen Regression zur Vorhersage von Variable B verwendet werden. Der Mittelwert von Variable A beträgt 25. Der Mittelwert von Variable B beträgt 40. Die Korrelation der beiden Variablen beträgt 0. Welcher Wert, der im Rahmen einer einfachen linearen Regression ermittelten Regressionskonstante, ist zu erwarten?

- 40
- 0
- 25
- 1
- ✓ 40

Quiz 3

Welche der folgenden Aussagen bezüglich der Zentrierung des Prädiktors im Rahmen einer einfachen linearen Regression sind wahr?

- Der Mittelwert des Prädiktors beträgt nach der Zentrierung 1, die Standardabweichung bleibt unverändert.
- Zentrierung bedeutet, dass von jedem Wert des Prädiktors die Standardabweichung der Prädiktorvariable abgezogen wird.
- Eine unmittelbare Vorhersage von Werten des Kriteriums aus Werten des Prädiktors ist nach der Zentrierung nicht mehr möglich.
- Die Regressionskonstante entspricht nach der Zentrierung dem erwarteten Wert der Kriteriumsvariable bei durchschnittlicher Ausprägung des Prädiktors.
- ✓ c) und d)

Quiz 4

- Eine Voraussetzung der Durchführung einer einfachen linearen Regression ist die Normalverteilung der Modellfehler, deren Realisierung die Residuen sind, die die Abweichungen der Schätzwerte von den Messwerten auf der Regressionsgerade beschreiben. Mit welchem/welchen Test(s) kann diese Voraussetzung geprüft werden?
- Breusch-Pagan-Test
- Durbin-Watson-Test
- Mauchly-Test
- Shapio-Wilk-Test
- ✓ Shapio-Wilk-Test

Quiz 5

- Was versteht man unter einer Ausreißerkorrektur (im Rahmen einer einfachen linearen Regression)?
- Ausschluss von zufällig ausgewählten 10 % der verfügbaren Datenpunkte
- Ausschluss von Extremwerten aus der Regression
- Regression basierend auf Extremwerten
- Regression basierend auf standardisierten Werten
- ✓ Ausschluss von Extremwerten aus der Regression

Quiz 6

- Welche der folgenden Aussagen bezüglich einer einfachen linearen Regression sind wahr?
- Die Prädiktoren einer einfachen linearen Regression müssen metrisch sein.
- Es ist möglich eine einfache lineare Regression mit kategorialem Prädiktor durchzuführen.
- Die Kriteriumsvariable einer einfachen linearen Regression kann ein beliebiges Skalenniveau aufweisen.
- Die Kriteriumsvariable einer einfachen linearen Regression muss metrisch sein.
- ✓ b) und d)

Quiz 7

- Wie viele Dummyvariablen werden zur Durchführung einer einfachen linearen Regression bei einem kategorialen Prädiktor mit 5 Stufen benötigt?
- 25
- 6
- 4
- 5
- ✓ 4

Quiz 8

- Es wird eine einfache lineare Regression zur Vorhersage der Kriteriumsvariable „Zufriedenheit am Arbeitsplatz“ durch den Prädiktor „Berufsgruppe“ durchgeführt. Es handelt sich um einen kategorialen Prädiktor mit 3 Stufen. Bei den untersuchten Berufsgruppen handelt es sich um Zahnärzte (Gruppe 1), Psychotherapeuten (Gruppe 2) und Apotheker (Gruppe 3). Es wird eine Dummykodierung angewandt. Im Rahmen der Regression werden die Regressionskonstante b_0 sowie die Regressionskoeffizienten b_1 und b_2 der Dummyvariablen ermittelt. Wie ist der Wert des Regressionskoeffizienten b_2 interpretierbar?
- Der Regressionskoeffizient b_2 entspricht der Mittelwertsdifferenz zwischen der Gruppe 3 der Apotheker und der Referenzgruppe 1 der Zahnärzte.
- Der Regressionskoeffizient b_2 entspricht der Abweichung des Mittelwertes der Gruppe 2 der Psychotherapeuten vom Mittelwert der Gruppe 3 der Apotheker.
- Der Regressionskoeffizient b_2 entspricht der Mittelwertsdifferenz zwischen der Gruppe 2 der Psychotherapeuten und der Referenzgruppe 1 der Zahnärzte.
- Der Regressionskoeffizient b_2 entspricht der Abweichung von Gruppe 2 der Psychotherapeuten vom Mittelwert aller Gruppenmittelwerte.
- ✓ a)

Praktisch Arbeiten mit R

- R-Markdown?
- R-Markdown!

Übung

- Laden Sie folgenden Datensatz:

```
df_ads <- rio:::import("https://raw.githubusercontent.com/Statistican/Datasets/main/Werbung.csv")
```

- Beschreiben Sie die Daten kurz.
- Ermitteln Sie die statistischen Kennzahlen der Variablen.
- Visualisieren Sie die Daten.
 - Insbesondere die bivariaten Beziehungen zwischen den verschiedenen möglichen Prädiktoren und dem Kriterium.
 - Welche möglichen Probleme können Sie aus den bivariaten Streudiagrammen für die Regression erkennen?
- Führen Sie drei getrennte lineare Regressionen mit jeweils einem Prädiktor durch.
- Prüfen Sie die Voraussetzungen der Regressionsmodelle:
- Visualisieren Sie die Regressionsgeraden.
- Wie ist die Qualität der Regressionsmodelle zu bewerten?

Übung

Lösungen

- Beschreibung & Kennzahlen:

```
# V1-Variable ausschließen  
df_ads <- df_ads %>% select(-V1)  
  
# deskr. Zusammenfassung  
df_ads %>% get_summary_stats() %>% mutate(VK = sd/mean) %>% flex()
```

variable	n	min	max	median	q1	q3	iqr	mad	mean	sd	se	ci	VK
TV	200	0.7	296.4	149.75	74.38	218.83	144.45	108.82	147.04	85.85	6.07	11.97	0.58
Radio	200	0.0	49.6	22.90	9.98	36.53	26.55	19.79	23.26	14.85	1.05	2.07	0.64
Zeitung	200	0.3	114.0	25.75	12.75	45.10	32.35	23.13	30.55	21.78	1.54	3.04	0.71
Verkauf	200	1.6	27.0	12.90	10.38	17.40	7.03	4.82	14.02	5.22	0.37	0.73	0.37

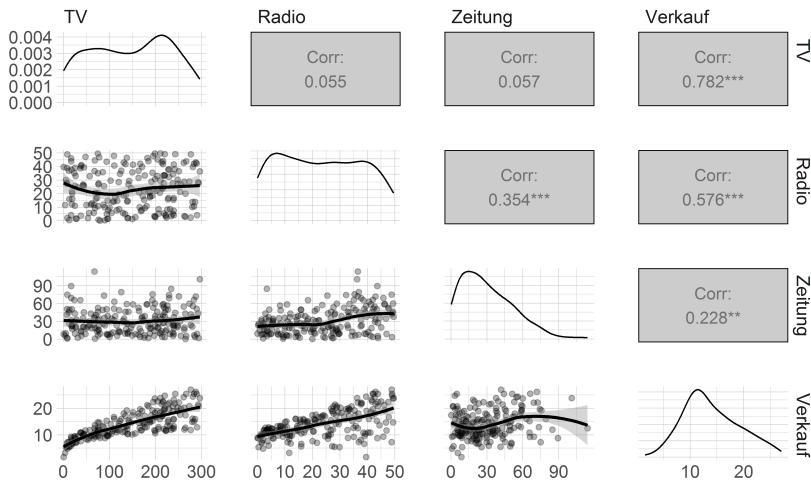
- Werbeausgaben: TV > Zeitung > Radio
- absolute Streuung: TV > Zeitung > Radio
- relative Streuung: Zeitung > Radio > TV

Übung

Lösungen

- Visualisierung & Beschreibung:

```
# vis. Zusammenfassung
GGally::ggpairs(df_ads, progress = F, lower = list(continuous = wrap('smooth', method = 'loess', alpha = .3)))
```



- Verteilung: TV & Radio ~ gleichverteilt; Zeitung: rechtsschief; Verkauf: leicht rechtsschief und steilgipflig
- keine krit. Multikollinearität zw. Prädiktoren
- enge Zusammenhang Outcome-Prädiktor: TV > Radio > Zeitung
- zunehmende Streuung bei Verkauf bei höheren Prädiktorwerten (insbes. bei TV & Radio)
- mögl. Probleme:
 - Heteroskedastizität
 - nicht-lineare Beziehung

Übung

Lösungen

- 3 Regressionen:

	formula	r.squared	adj.r.squared	sigma	statistic	p.value	df	logLik	AIC	BIC	deviance	df.residual	nobs
Verkauf ~ TV	Verkauf ~ TV	0.61	0.61	3.26	312.14	p < .001	1	-519.05	1'044.09	1'053.99	2'102.53	198	200
Verkauf ~ Radio	Verkauf ~ Radio	0.33	0.33	4.27	98.42	p < .001	1	-573.34	1'152.67	1'162.57	3'618.48	198	200
Verkauf ~ Zeitung	Verkauf ~ Zeitung	0.05	0.05	5.09	10.89	p = 0.001	1	-608.34	1'222.67	1'232.57	5'134.80	198	200

- Modellgüte: TV > Radio > Zeitung