Contents

Chapter 1

Fortgeschrittene Themen der linearen Regression

In diesem Kapitel werden die folgenden R Pakete verwendet:

```
library(here)
library(tidyverse)
library(data.table)
library(latex2exp)
library(icaeDesign)
library(ggpubr)
library(lmtest)
library(sandwich)
library(MASS)
```

1.1 Annahmen und Eigenschaften des einfachen OLS Modells

In diesem Abschnitt werden wir zunächst unser neu gewonnenes Wissen über Matrixnotation aus dem letzten Kapitel verwenden um die uns bereits bekannten Annahmen des OLS Modells in Matrixschreibweise auszudrücken. Das wird sich als enorm hilfreich erweisen da alle modernen Texte und fortgeschrittenen Lehrbücher die Matrixschreibweise verwenden und alle relevanten Beweise und Herleitung sich dieser Notation bedienen.

Danach werden wir uns mit den wichtigen Eigenschaften *Erwartungstreue*, *Effizienz* und *Konsistenz* von Schätzern beschäftigen. Alles drei sind erstrebenswerte Eigenschaften, über die der OLS Schätzer auch verfügt wenn die Annahmen für das OLS Modell erfüllt sind. Allerdings kann er diese Eigenschaften verlieren wenn einzelne Annahmen verletzt sind. Um die Konsequenzen verletzter Annahmen zu illustrieren verwenden wir häufig die Methode der *Monte Carlo Simulation*, die wir am Ende dieses Abschnitts einführen werden.

1.1.1 Annahmen im Matrixschreibweise

An dieser Stelle werden wir die uns aus diesem Abschnitt bekannten Annahmen für die OLS Schätzung in Matrixschreibweise ausdrücken und leicht zusammenfassen, bzw. ihre Reihenfolge an die in der Literatur typische Reihenfolge anpassen.

Zu diesem Zweck betrachten wir das folgende Modell:

$$\boldsymbol{y} = \boldsymbol{x_1}\beta_1 + ... + \boldsymbol{x_k}\beta_k + \boldsymbol{\epsilon}$$

in dem $\boldsymbol{y}$ der $1 \times n$ Vektor mit den n Beobachtungen der abhängigen Variable ist. Für jede der k unabhängige Variable haben wir die Beobachtungen in einem $1 \times n$ Vektor $\boldsymbol{x_i}(i \in k)$ gesammelt.

Diese k Vektoren werden häufig in der $n \times k$ Matrix $\boldsymbol{X}$ zusammengefasst, sodass die folgende kompakte Schreibweise verwendet werden kann:

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$$

$\boldsymbol{\beta}$ ist der Vektor der (unbeobachtbaren) Modellparameter $\beta_0, ..., \beta_k$, die wir schätzen wollen, und $\boldsymbol{\epsilon}$ ist der Vektor der (ebenfalls unbeobachtbaren) Fehlerterme.

Der OLS Schätzer $\hat{\boldsymbol{\beta}}$ für $\boldsymbol{\beta}$ ist durch folgende Gleichung definiert (für die Herleitung siehe hier):

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{X'X})^{-1}(\boldsymbol{X'Y})$$

Unter bestimmten Annahmen hat dieser Schätzer die attraktiven Eigenschaften *Erwartungstreue* und *Effizienz* unabhängig der Stichprobengröße und in großen Stichproben zudem die Eigenschaft der *Konsistenz*. Die relevanten Annahmen sind dabei die folgenden:

A1: Der Zusammenhang zwischen abhängiger und unabhängigen Variablen ist linear

Diese Annahme ergibt sich unmittelbar aus der Formulierung: $\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$. Wenn der Zusammenhang zwischen abhängiger und unabhängigen Variablen nicht linear ist können wir das klassische OLS Modell in der Regel nicht verwenden. Häufig können wir aber die Daten so transformieren, dass wir deren Verhältnis als linearen Zusammenhang darstellen können. So ist z.B. der folgende Zusammenhang nicht linear:

$$\boldsymbol{y} = \boldsymbol{x_1}^{\beta_1} + e^{\epsilon} \tag{1.1}$$

wir können aber einfach die Variablen logarithmieren und erhalten somit die folgende lineare Gleichung, die wir dann mit OLS schätzen können:

$$\ln(\boldsymbol{y}) = \ln(\boldsymbol{x_1})\beta_1 + \boldsymbol{\epsilon}$$

Es ist dabei zu beachten, dass das Regressionsmodell kein Problem mit nichtlinearen Transformationen für die abhängigen Variablen wie $\ln(\boldsymbol{x_i})$ hat, sondern dass nur der funktionale Zusammenhang linear sein muss. Daher sprechen wir häufig davon, dass mit OLS zu schätzende Zusammenhänge *linear in den Parametern* sein müssen -

nicht notwendigerweise linear per se. Wir werden uns mit den relevanten Transformationen später in diesem Kapitel beschäftigen.

Dennoch gibt es natürlich auch viele Zusammenhänge, die nicht linear sind und auch nicht in eine lineare Funktion transformiert werden können. Für diese Zusammenhänge müssen wir andere Schätzverfahren benutzen.

A2: Exogenität der unabhängigen Variablen

Die Annahme kombiniert die beiden Annahmen, die wir vorher unter dem Titel "Unabhängigkeit der Fehler mit den erklärenden Variablen" und "Erwartungswert der Fehler gleich Null" kennen gelernt haben. In der fortgeschrittenen Literatur ist die Referenz zur Exogenität der unabhängigen Variablen gebräuchlicher. Formal können wir schreiben:

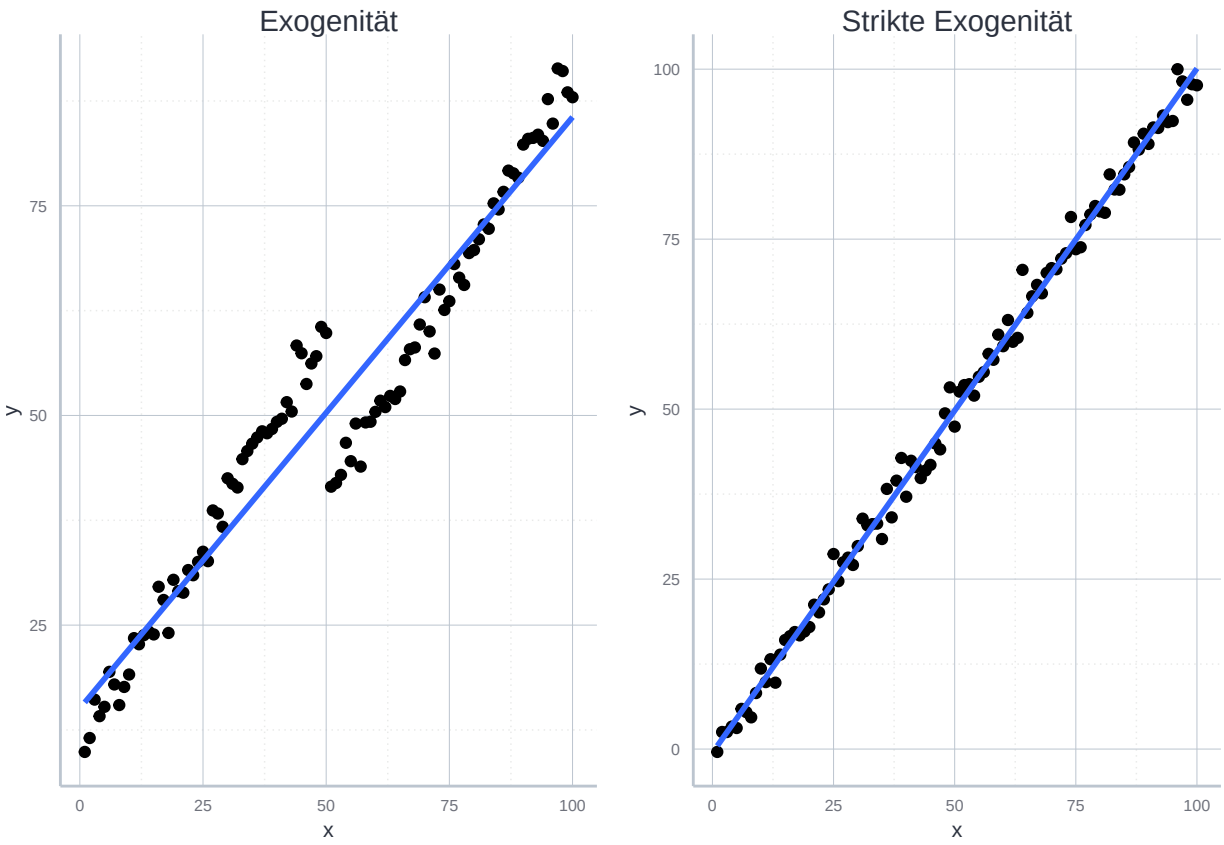
$$\mathbb{E}\left[\boldsymbol{\epsilon}|\boldsymbol{x}\right] = 0$$

Daher kommt der Begriff "Exogenität": Die unabhängigen Variablen enthalten keine Informationen über die Fehlerterme. Mit Informationen über $\boldsymbol{x}$ können wir die Fehler des Modells also nicht vorhersagen - denn $\boldsymbol{x}$ ist *exogen*. Man kann übrigens formal zeigen, dass $\mathbb{E}\left[\boldsymbol{\epsilon}|\boldsymbol{x}\right] = 0$ auch impliziert dass $\mathbb{E}\left[\boldsymbol{\epsilon}\right] = 0$.[1] Der bedingte Erwartungswert von Null impliziert also den unbedingten Erwartungswert von Null - aber nicht umgekehrt.

Manchmal wird daher auch eine noch strengere Annahme verwendet: *strikte Exogenität*. Darunter verstehen wir die Annahme, dass $\mathbb{E}\left[\epsilon_i|\boldsymbol{X}\right] = 0$ bzw. für alle ϵ_i: $\mathbb{E}\left[\boldsymbol{\epsilon}|\boldsymbol{X}\right] = 0$. Hier nehmen wir sogar an, dass jeder einzelne Fehlerterm auch mit den unabhängigen Variablen für andere Beobachtungen nicht korreliert. Das impliziert, dass $\text{Cov}(\epsilon_i, \boldsymbol{x}) = 0 \forall i$.

> **Bedingter vs. unbedingter Erwartungswert der Fehler** Auf den ersten Blick klingt es komisch, dass der bedingte Erwartungswert der Fehler von Null, $\mathbb{E}\left[\boldsymbol{\epsilon}|\boldsymbol{x}\right] = 0$, den unbedingten Erwartungswert von Null, $\mathbb{E}\left[\boldsymbol{\epsilon}\right] = 0$, impliziert, aber nicht andersherum. Folgendes Beispiel illustriert dieses Problem:

[1] Wen der Beweis interessiert wird in Greene (2018) fündig.



In der linken Abbildung haben wir einen bedingten Erwartungswert von Null: für jede beliebige Beobachtung in $\boldsymbol{x}$ ist der Erwartungswert der Fehler Null. Daraus ergibt sich, dass der Erwartungswert für alle Fehler zusammen auch Null ist. In der rechten Abbildung ist der bedingte Erwartungswert nicht Null: für die untere Hälte der Beobachtungen in $\boldsymbol{x}$ ist der Erwartungswert 1, für die obere Hälfte der Beobachtungen ist der Erwartungswert -1. Für die gesamten Daten ergibt sich dabei auch ein Erwartungswert von 0, aber eben *nicht* für jede einzelne Beobachtung. Häufig tritt diese Problem bei quadratischen Zusammenhängen auf.

Wichtig ist festzuhalten, dass dies eine Annahme über nicht zu beobachtende Größen darstellt: die tatsächlichen Fehlerterme $\boldsymbol{\epsilon}$ können wir in der Praxis nicht beobachten. Wir sprechen daher auch von einer Annahme über die *Population*. Alles was wir aus der Population direkt beobachten können ist eine Stichprobe. Und innerhalb der Stichprobe können wir als Annäherung der Fehlerterme $\boldsymbol{\epsilon}$ die Residuen $\boldsymbol{e}$ berechnen. Die 'echten' Fehlerterme können wir aber nicht beobachten.

A3: Keine perfekte Multikollinearität

Die Annahme, dass die unabhängigen Variablen nicht linear voneinander abhängig sind ist notwendig damit der OLS Schätzer $\hat{\boldsymbol{\beta}} = (\boldsymbol{X'X})^{-1}(\boldsymbol{X'Y})$ überhaupt berechnet werden kann. Dann wären zwei oder mehrere unabhängigen Variablen linear abhängig könnten wir von $\boldsymbol{X}$ keine Inverse $\boldsymbol{X}^{-1}$ bilden und der OLS Schätzer von $\boldsymbol{\beta}$ wäre nicht *identifizierbar*. Häufig wir diese Annahme auch in 'Matrizensprache' formuliert. Dann sprechen wir von der Annahme, dass die Matrix $\boldsymbol{X}$ *vollen Rang* hat. Damit ist aber das gleiche gemeint. Die Annahme impliziert zudem, dass wir $n \geq k$ und dass es eine gewisse Variation in den unabhängigen Variablen gibt. All das ist in der Praxis aber immer erfüllt - nur mit dem Problem der nicht perfekten Kollinearität - also der Situation wo die abhängigen Variablen stark miteinander korrelieren - müssen wir uns häufig herumschlagen. Doch dazu später mehr.

A4: Konstante Varianz und keine Autokorrelation der Fehlerterme

Vorher hatten wir diese beiden Annahmen als separate Annahmen formuliert. In der Literatur werden sie jedoch oft zusammengefasst, weil sich beide Annahmen um die Struktur der *Varianz-Kovarianz-Matrix* einer Schätzung drehen. Für eine Schätzung mit n Beobachtungen handelt es sich dabei um eine $n \times n$-Matrix, auf deren Hauptdiagonalen die Varianzen der Fehlerterme und in den sonstigen Elementen die Kovarianzen der einzelnen Fehlerpaare gesammelt sind. Für den Fall von zwei abhängigen Variablen hätten wir also folgende Varianz-Kovarianz Matrix:

$$\begin{pmatrix} Var(\epsilon_1|\boldsymbol{X}) & Cov(\epsilon_1, \epsilon_2|\boldsymbol{X}) \\ Cov(\epsilon_2, \epsilon_1|\boldsymbol{X}) & Var(\epsilon_2|\boldsymbol{X}) \end{pmatrix}$$

Die Annahme der konstanten Varianz - oder "Homoskedastizität" - bezieht sich also auf die Hauptdiagonale der Varianz-Kovarianz-Matrix und sagt:

$$Var(\epsilon_i|\boldsymbol{X}) = \sigma^2 \quad \forall i$$

Die Annahme nichtautokorrelierter Fehler bezieht sich dann auf die Elemente außerhalb der Hauptdiagonalen der Varianz-Kovarianz-Matrix und sagt:

$$Cov(\epsilon_i, \epsilon_j|\boldsymbol{X}) = 0 \quad \forall i \neq j$$

Bei den Fehlertermen ϵ_i handelt es sich ja im Zufallsvariablen. Aufgrund der Definition der Varianz und A2, gemäß derer gilt, dass $\mathbb{E}(\epsilon|\boldsymbol{X}) = 0$, bekommen wir für die Varianz der Fehler:

$$\begin{aligned} Var(\epsilon_i|\boldsymbol{X}) &= \mathbb{E}\left[(\epsilon_i - \mathbb{E}(\epsilon_i|\boldsymbol{X}))^2 |\boldsymbol{X}\right] \\ &= \mathbb{E}\left[\epsilon_i^2 - 2\epsilon_i \mathbb{E}(\epsilon_i|\boldsymbol{X}) + \mathbb{E}(\epsilon_i|\boldsymbol{X})^2 |\boldsymbol{X}\right] \\ &= \mathbb{E}\left[\epsilon_i^2|\boldsymbol{X}\right] = \mathbb{E}\left[\epsilon_i \epsilon_i|\boldsymbol{X}\right] \end{aligned}$$

Die zweite Zeile ergibt sich dabei aus der *zweiten binomischen Formel*. Für die Kovarianz gilt entsprechend:

$$\begin{aligned} Cov(\epsilon_i, \epsilon_j|\boldsymbol{X}) &= \mathbb{E}\left[(\epsilon_i - \mathbb{E}(\epsilon_i|\boldsymbol{X}))(\epsilon_j - \mathbb{E}(\epsilon_j|\boldsymbol{X}))|\boldsymbol{X}\right] \\ &= \mathbb{E}\left[(\epsilon_i\epsilon_j - \epsilon_i\mathbb{E}(\epsilon_j|\boldsymbol{X}) - \epsilon_j\mathbb{E}(\epsilon_i|\boldsymbol{X}) + \mathbb{E}(\epsilon_j|\boldsymbol{X})\mathbb{E}(\epsilon_i|\boldsymbol{X}))|\boldsymbol{X}\right] \\ &= \mathbb{E}\left[\epsilon_i\epsilon_j|\boldsymbol{X}\right] \end{aligned}$$

Hier haben wir in der zweiten Zeile die *dritte binomische Formel* verwendet.

Daher kann die Annahme von Homoskedastizität und keiner Autokorrelation auch folgendermaßen ausgedrückt werden:

$$\mathbb{E}(\boldsymbol{\epsilon}\boldsymbol{\epsilon}'|\boldsymbol{X}) = \begin{pmatrix} \mathbb{E}(\epsilon_1\epsilon_1|\boldsymbol{X}) & \mathbb{E}(\epsilon_1\epsilon_2|\boldsymbol{X}) & ... & \mathbb{E}(\epsilon_1\epsilon_n|\boldsymbol{X}) \\ \mathbb{E}(\epsilon_2\epsilon_1|\boldsymbol{X}) & \mathbb{E}(\epsilon_2\epsilon_2|\boldsymbol{X}) & ... & \mathbb{E}(\epsilon_2\epsilon_n|\boldsymbol{X}) \\ & & \vdots & \\ \mathbb{E}(\epsilon_n\epsilon_1|\boldsymbol{X}) & \mathbb{E}(\epsilon_n\epsilon_2|\boldsymbol{X}) & ... & \mathbb{E}(\epsilon_n\epsilon_n|\boldsymbol{X}) \end{pmatrix} = \begin{pmatrix} \sigma^2 & 0 & ... & 0 \\ 0 & \sigma^2 & ... & 0 \\ & & \vdots & \\ 0 & 0 & ... & \sigma^2 \end{pmatrix}$$

oder zusammengefasst:

$$\mathbb{E}(\boldsymbol{\epsilon}\boldsymbol{\epsilon}'|\boldsymbol{X}) = \sigma^2\boldsymbol{I}$$

A5: Normalverteilung der Fehlerterme:

Die letzte typischerweise gemachte Annahme ist die Normalverteilung der Fehlerterme, bedingt wie immer auf die unabhängigen Variablen:

$$\boldsymbol{\epsilon}|\boldsymbol{X} \propto \mathcal{N}(\boldsymbol{0}, \sigma^2\boldsymbol{I})$$

Diese Annahme vereinfacht zahlreiche Herleitungen ist in der Praxis allerdings weniger relevant, da sie leicht abzuschwächen ist.

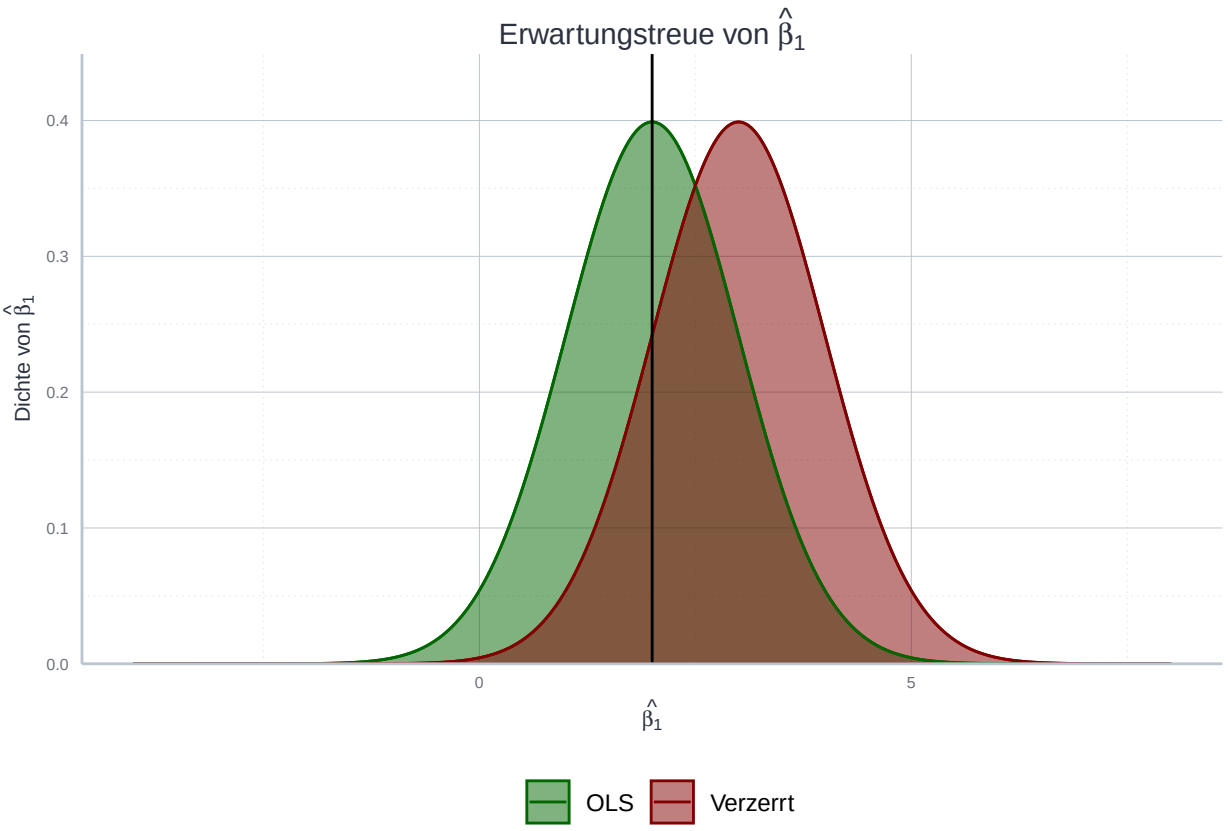
1.1.2 Erwartungstreue, Effizienz und Konsistenz

Unter den oben beschriebenen Annahmen weist $\hat{\boldsymbol{\beta}}$ drei wichtige Eigenschaften auf: (1) er ist *erwartungstreu* und (2) er ist *effizient*, auch in kleinen Stichproben. In großen Stichproben ist er zudem (3) *konsistent*. Alle Eigenschaften beziehen sich auf die *Verteilung* von $\hat{\boldsymbol{\beta}}$ (wie im einführenden Kapitel beschrieben handelt es sich bei $\hat{\boldsymbol{\beta}}$ ja um eine Zufallsvariable).

Ohne die Konzepte schon eingeführt zu haben wollen dennoch bereits an dieser Stelle festhalten, dass für die Erwartungstreue nur A1 und A2 relevant ist. Annahmen A4 und A5 sind nur für Inferenz und Standardfehler sowie die Effizienz von Bedeutung. A3 ist wie oben beschrieben notwendig, damit der OLS Schätzer überhaupt identifizierbar ist.

Unter **Erwartungstreue** verstehen wir die Eigenschaft, dass der Schätzer im Mittel den 'wahren Wert' β trifft, also $\mathbb{E}(\hat{\boldsymbol{\beta}}) = \beta$. Der Schätzvorgang ist also nicht systematisch verzerrt. Das bedeutet natürlich nicht, dass wir für eine *einzelne* Schätzung gilt $\hat{\boldsymbol{\beta}} = \beta$, aber dass β der wahrscheinlichste Wert für $\hat{\boldsymbol{\beta}}$ ist. Oder technisch: das Mittel unendlich vieler Schätzungen mit $\hat{\boldsymbol{\beta}}$ ist gleich β.

Diese Eigenschaft des OLS-Schätzers wird in folgender Abbildung illustriert:

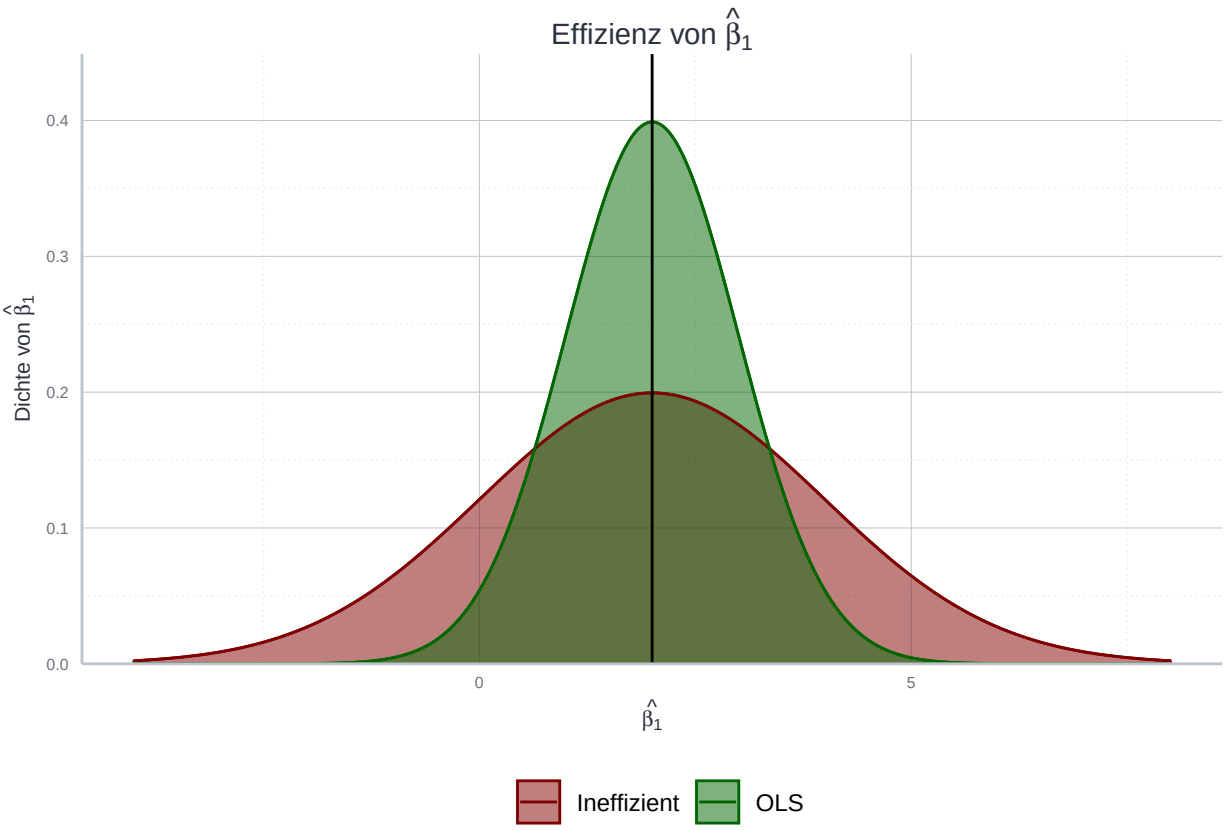


Wir können beweisen, dass $\hat{\boldsymbol{\beta}}$ unter Annahmen A1, A2 und A3 erwartungstreu ist. Dies gilt unabhängig der Stichprobengröße und unabhängig davon ob Annahmen A4 und A5 erfüllt sind.[2]

Unter **Effizienz** verstehen wir die Eigenschaft, dass es keinen alternativen Schätzer für β gibt, der eine geringere Varianz aufweist. Effizienz ist dabei ein *relatives Maß*: ein Schätzer ist effizienter als ein anderer, wenn seine Varianz geringer ist.

Die Eigenschaft der Effizienz wird in folgender Abbildung illustriert:

[2] Die Beweise werden später ergänzt und sind zu diesem Zeitpunkt nicht zentral.



Da wir hier die zugrundeliegenden Daten selbst herstellen wissen wir, dass für den wahren Wert gilt $\beta_1 = 2.0$. Um die Effizienz des OLS-Schätzers beweisen zu können reichen Annahmen A1-A3 nicht aus: hierfür benötigen wir auch die Annahme A4! Unter Annahmen A1-A4 gilt die Effizienz des OLS-Schätzers aber auch unabhängig von der Stichprobengröße.

Dass die Eigenschaften der Erwartungstreue und Effizienz beim OLS-Schätzer unabhängig von der Stichprobengröße gelten ist eine tolle Sache. Solche stichprobenunabhängigen Beweise funktionieren in realen Settings, in denen bestimmte Annahmen leicht verletzt sind und die zu schätzenden Funktionen komplexer werden häufig nicht. Daher versucht man Eigenschaften von Schätzern wenigstens für große Stichproben zu beweisen. Diese Beweise sind wegen bestimmten Gesetzen wie dem Gesetz der großen Zahl oder dem Zentralen Grenzwertsatz oft deutlich einfacher. Wie sprechen dann von *asymptotischen Eigenschaften*, da sie für den Schätzer zutreffen wenn die Stichprobengröße gegen Unendlichkeit wächst.

Allerdings bleibt dann unklar welche Eigenschaften der Schätzer in kleinen Stichproben tatsächlich hat. Auch ab welcher Größe eine Stichprobe als "groß" gilt kann nicht ohne Weiteres beantwortet werden. Um die Schätzereigenschaften für kleine Stichproben zu untersuchen bleibt dann nur die Methode der *Monte Carlo Simulation*, die weiter unten eingeführt wird.

Vorher wollen wir jedoch die wichtigste Eigenschaft von Schätzern für große Stichproben anhand des OLS-Schätzers einführen: die **Konsistenz**. Ein konsistenter Schätzers trifft im Mittel den wahren Wert und seine Varianz geht mit wachsender Stichprobengröße gegen Null. Wir können also sagen, dass unsere Schätzungen bei wachsender Stichprobengröße immer genauer wird.

Formal drücken wir dies unter Verwendung von Grenzwerten aus:

$$\lim_{N \to \infty} \mathbb{P}(|\hat{\beta} - \beta| > \epsilon) = 0 \leftrightarrow \text{plim}(\hat{\beta}) = \beta$$

wobei ϵ hier eine beliebig kleine Zahl ist.

In der klassischen statistischen Analyse betrachten wir Erwartungstreue als eine notwendige Eigenschaft: wir möchten in der Regel keine Schätzer verwenden, deren geschätzte Werte systematisch von dem wahren Wert abweichen. Es sei an dieser Stelle jedoch bereits erwähnt, dass es sinnvolle Ausnahmen von dieser Regel geben kann, nämlich dann wenn wir große Zugewinne an Effizienz für kleine Abstriche in der Erwartungstreue 'erkaufen' können.

In der Literatur wird diese Fragestellung unter dem Stichwort *bias-variance trade-off* diskutiert. Weitergehende Informationen finden Sie in der weiterführenden Literatur. An dieser Stelle wollen wir uns zunächst auf die erwartungstreuen Schätzer konzentrieren, da dies tatsächlich auch die am weitesten verbreiteten Schätzmethoden sind.

1.1.3 Abweichungen von den OLS Annahmen

Wenn alle Annahmen des OLS-Schätzers erfüllt sind können wir also ohne Bedenken die Parameter unseres statistischen Modells mit der klassischen OLS Methode schätzen. Aber was ist wenn eine Annahme nicht erfüllt ist?

Im folgenden wollen wir uns diesem Problem annähern indem wir die folgenden Fragen für die verschiedenen Annahmen anhand der folgenden beiden Leitfragen diskutieren: (1) Unter welchen praktisch relevanten Situationen kann die Annahme verletzt sein? (2) Wie können wir testen ob die Annahme verletzt ist? (3) Was sind die Konsequenzen wenn die Annahme verletzt ist? (4) Was können wir tun um trotz verletzter Annahme konsistente und möglichst effiziente Schätzer zu bekommen.

Diese Fragen sind in der der Praxis nicht einfach zu beantworten. Ein Grund dafür ist, dass wir die 'wahren Werte' der zu schätzenden Parameter in der Regel nicht beobachten können. Da wir zudem den 'wahren' datenerzeugenden Prozess nicht kennen, können wir nie mit Sicherheit sagen, ob eine bestimmte Annahme verletzt ist oder nicht.

Dennoch gibt es zwei Möglichkeiten die relevanten Informationen zu den Schätzern zu bekommen: zum einen können wir häufig mathematisch beweisen, dass eine Schätzer erwartungstreu oder effizient ist. Ein Beispiel dafür ist der Beweis der Erwartungstreue des OLS-Schätzers hier oder der Beweis der Effizienz des OLS-Schätzers hier. Dies ist aber nicht immer möglich und manchmal auch recht aufwendig und wenig intuitiv.

Die zweite Möglichkeit ist die Analyse von Schätzern mit Hilfe von künstlichen Datensätzen und so genannten Monte-Carlo Simulationen. Hier definieren wir unseren datenerzeugenden Prozess selbst und erstellen dann einen künstlichen Datensatz. Diese Vorgehensweise ist zwar weniger 'sicher' als ein mathematischer Beweis aber häufig intuitiver und in vielen Fällen tatsächlich auch die einzige Möglichkeit, inbesondere wenn wir Schätzereigenschaften für kleine Stichproben analysieren wollen. Daher wird diese Methode im folgenden kurz beschrieben und später für die Illustration der Folgen von verletzten Annahmen verwendet.

1.1.4 Monte Carlo Simulationen in R

Der Ablauf einer Monte Carlo Simulation ist immer der folgende:

1. Definiere das zu untersuchende Merkmal des datenerzeugenden Prozesses
2. Formalisiere den datenerzeugenden Prozess als Funktion
3. Erstelle viele künstliche Stichproben für das zu untersuchende Merkmal; erstelle dabei eine Kontrollgruppe in der das zu untersuchende Merkmal nicht vorhanden ist und eine Testgruppe mit dem Merkmal und wende den zu untersuchenden Schätzer auf die künstlichen Stichproben an
4. Analysiere die Verteilung des Schätzers für die Kontrollgruppe und die Testgruppe

5. Interpretiere die Ergebnisse

Wir erstellen also selbst einen datenerzeugenden Prozess und untersuchen dann das Verhalten des interessierenden Schätzers im Kontext dieses datenerzeugenden Prozesses. Wenn wir z.B. untersuchen möchten welchen Effekt Heteroskedastie auf den OLS Schätzer hat dann erstellen wir künstliche Datensätze über einen datenerzeugenden Prozess in den wir Heteroskedastie eingebaut haben und und über einen Prozess für den wir wissen, dass er durch Homoskedastie gekennzeichnet ist. Dann schätzen wir ein Modell jeweils für die beiden Prozesse und vergleichen die Eigenschaften des OLS-Schätzers. Somit können wir Rückschlüsse auf die Implikationen von Heteroskedastie schließen.

Im folgenden wollen wir die Methode der Monte-Carlo Simulation über genau dieses Beispiel einführen.

1. Schritt: Definition des zu untersuchenden Merkmals

Wie gerade beschrieben möchten wir untersuchen welchen Effekt Heteroskedastie auf die Eigenschaften des OLS Schätzers hat. Das zu untersuchende Merkmal des datenerzeugenden Prozesses ist also *Heteroskedastie*.

2. Schritt: Formalisierung des datenerzeugenden Prozesses

Wir formalisieren jetzt einen datenerzeugenden Prozess, der alle Annahmen des OLS Schätzers erfüllt außer ggf. der Annahme der Homoskedastie. Der Einfachheit halber wollen wir einen Prozess mit einer erklärenden Variablen erstellen, also einen Prozess, der durch folgende Gleichung beschrieben werden kann:

$$Y = \beta_0 + \beta_1 x_1 + \epsilon$$

wobei wir annehmen, dass $\epsilon \propto \mathcal{N}(\mu, \sigma)$ und σ im Falle der Kontrollgruppe konstant (Fall der Homoskedastie) und im Falle der Testgruppe variabel ist (Fall der Heteroskedastie).

Wir definieren also folgende Funktion, die für gegebene Werte für β_0 und β_1 und ein gegebenes $\boldsymbol{X}$ eine Stichprobe erstellt indem $\boldsymbol{y}$ gemäß des Modells $y = \beta_0 + \beta_1 x + \epsilon$ künstlich hergestellt wird.

```
dgp <- function(x1, beta0, beta1, hetero=FALSE){
  y <- rep(NA, length(x1))
  sd_hetero <- 0.25 * x1
  sd_homo <- mean(sd_hetero)
  if (hetero){
    errors <- rnorm(n = length(x1), mean = 0,
                    sd = sd_hetero)
  } else {
    errors <- rnorm(n = length(x1), mean = 0,
                    sd = sd_homo
                    )
  }
  for (i in 1:length(x1)){
    y[i] <- beta0 + beta1*x1[i] + errors[i]
  }
  final_data <- tibble(y=y, x1=x1, errors=errors)
  return(final_data)
}
```

3. Schritt: Künstlichen Datensatz erstellen und Schätzer darauf anwenden

Wir simulieren nun das Ziehen einer Stichprobe aus dem künstlich erstellten DGP indem wir jeweils 1000 Beobachtungen kreieren. Da das Ziehen einer Stichprobe immer ein Zufallsprozess ist erstellen wir 1000 Stichproben und wenden darauf dann jeweils unseren OLS-Schätzer an. Die geschätzten Koeffizienten und Standardfehler speichern wir in einer Liste, da wir sie später dann analysieren wollen.

Dazu definieren wir die folgende Funktion:

```
mcs <- function(n_stichproben,
                x1, wahres_b0, wahres_b1, schaetzgleichung,
                heterosk=FALSE){
  schaetzung_b1 <- rep(NA, n_stichproben)
  stdfehler_b1 <- rep(NA, n_stichproben)
  for (i in 1:n_stichproben){
    # Stichprobe ziehen:
    stichprobe <- dgp(x1 = x1, beta0 = wahres_b0,
                      beta1 = wahres_b1,
                      hetero = heterosk)
    # Parameter schätzen:
    schaetzung <- summary(
      lm(formula = schaetzgleichung,
         data = stichprobe)
      )
    # Relevante Werte speichern:
    schaetzung_b1[i] <- schaetzung$coefficients[2]
    stdfehler_b1[i] <- schaetzung$coefficients[4]
  }
  # In einer Tabelle zusammenfassen:
  Fall_Bezeichnung <- ifelse(heterosk, "Heteroskedastie", "Homoskedastie")
  ergebnisse <- tibble(
    b1_coef=schaetzung_b1,
    b1_stdf=stdfehler_b1,
    Fall=rep(Fall_Bezeichnung,
             n_stichproben)
  )
return(ergebnisse)
}
```

Damit können wir die Simulation sehr einfach für die beiden relevanten Fälle ausführen.

Wir definieren nun die Parameter und die wahren Werte. Hierbei ist es wichtig, die Funktion `set.seed` zu verwenden. Das ist wichtig um unsere Monte-Carlo Simulation reproduzierbar zu machen, denn mit `set.seed` setzen wir die Anfangsbedingungen für den Zufallszahlen-Generator von R. Das bedeutet, dass wir für den gleichen Seed immer die gleichen Zufallszahlen produzieren und somit unsere Simulationsergebnisse immer vollständig reproduzierbar bleiben.

```
set.seed("1234")
n_stichproben <- 250
```

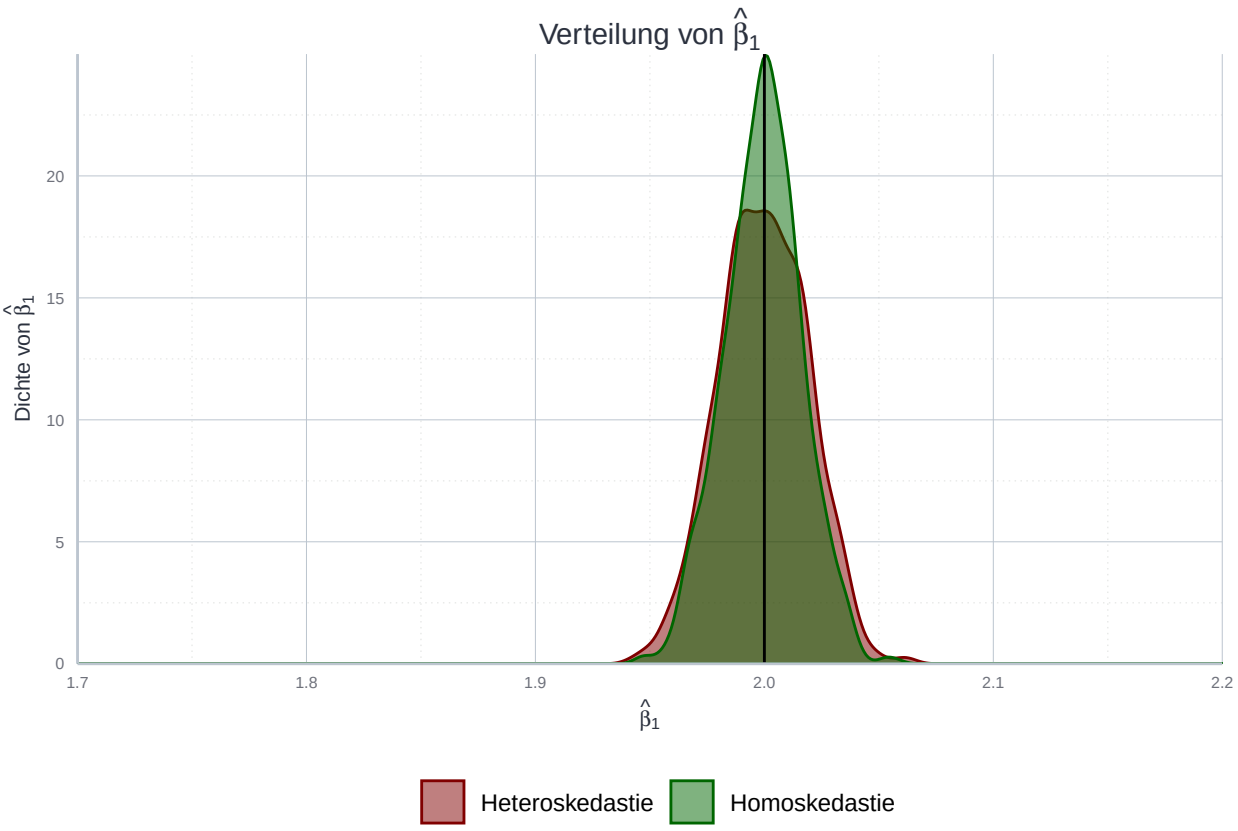
```
n_beobachtungen <- 1000
x_data <- runif(n = n_beobachtungen, min = 1, max = 10)
wahres_b0 <- 1
wahres_b1 <- 2
schaetzgleichung <- as.formula("y~x1")
```

```
set.seed("1234")
homosc_results <- mcs(1000, x_data,
                      wahres_b0, wahres_b1,
                      schaetzgleichung, heterosk = F)
hetero_results <- mcs(1000, x_data,
                      wahres_b0, wahres_b1,
                      schaetzgleichung, heterosk = T)
full_results <- rbind(homosc_results, hetero_results)
```

4. Schritt: vergleichende Analyse der Schätzereigenschaften

Als erstes wollen wir die Ergebnisse grafisch analysieren. Zu diesem Zweck visualisieren wir Verteilung der geschätzten Werte für β_1 und zeichnen zudem den wahren Wert ein:

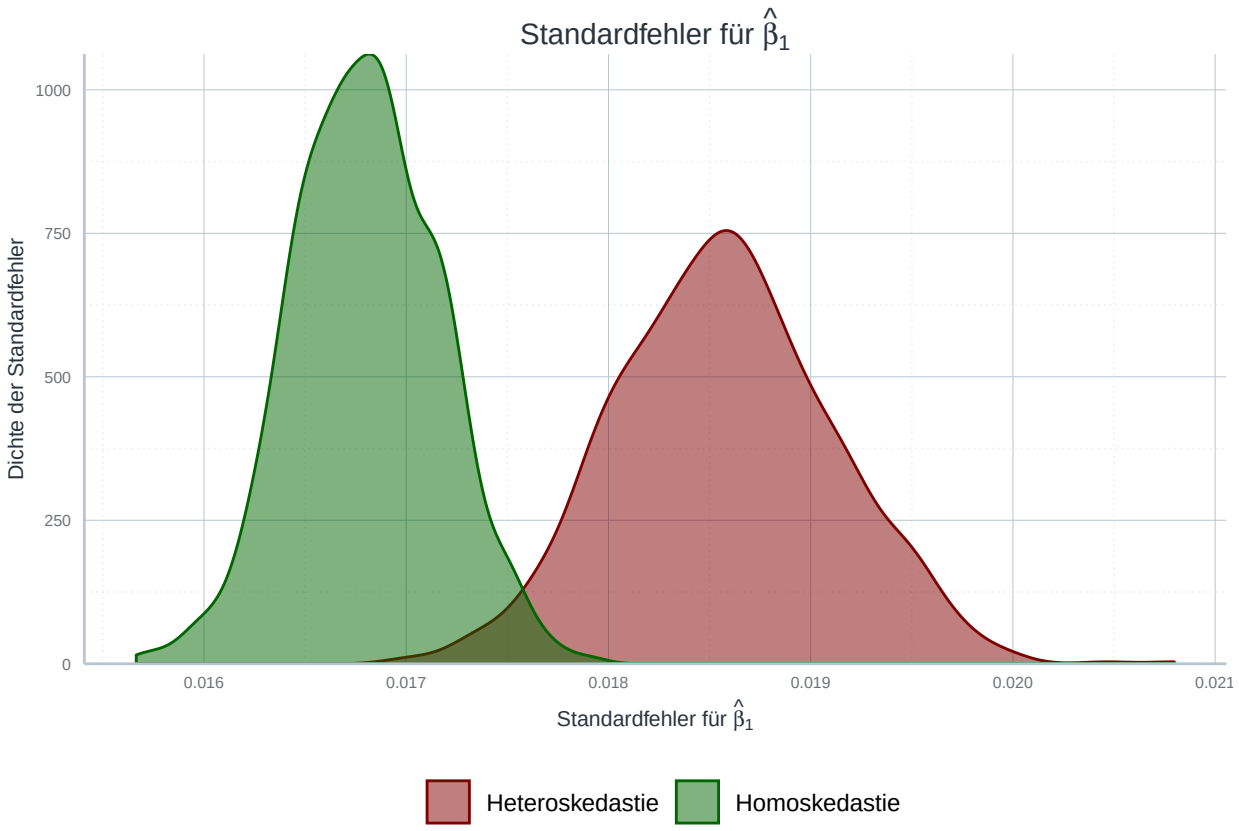
```
beta_1_plot <- ggplot(data = full_results,
                      mapping = aes(x=b1_coef, color=Fall, fill=Fall)) +
  geom_density(alpha=0.5) +
  scale_y_continuous(expand = expand_scale(c(0, 0), c(0, 0.05))) +
  scale_x_continuous(limits = c(1.7, 2.2), expand = c(0,0)) +
  geom_vline(xintercept = wahres_b1) +
  ylab(TeX("Dichte von $\\hat{\\beta}_1$")) +
  xlab(TeX("$\\hat{\\beta}_1$")) +
  ggtitle(TeX("Verteilung von $\\hat{\\beta}_1$")) +
  scale_color_manual(values = c("Homoskedastie"="#006600",
                                "Heteroskedastie"="#800000"),
  aesthetics = c("color", "fill")) +
  theme_icae()

beta_1_plot
```



Wie wir sehen ändert die Verletzung der Homoskedastie-Annahme nichts an der Erwartungstreue des Schätzers: im Mittel trifft der Schätzer den wahren Wert β_1! Allerdings nimmt die Genauigkeit ab, da die Streuung um den wahren Wert herum im heteroskedastischen Fall zunimmt!

Wir wollen im Folgenden noch untersuchen wie sich Heteroskedastie auf die Standardfehler der Regression auswirkt (der Code zum Erstellen der Plots ist äquivalent zu oben):

```
beta_1_stdf_plot
```



Wir wir sehen weichen die Standardfehler im heteroskedastischen Fall deutlich von denen im homoskedastischen Fall ab! Welche Standardfehler sind nun die richtigen?

Ohne auf die mathematische Herleitung genauer einzugehen (siehe Kapitel 4 in Greene (2018)) wollen wir dennoch festhalten, dass die geschätzten Standardfehler unter Heteroskedastie *falsch* sind. Wir können ohne eine Korrektur also keine Aussagen über die Schätzunsicherheit und Signifikanz der Ergebnisse treffen.

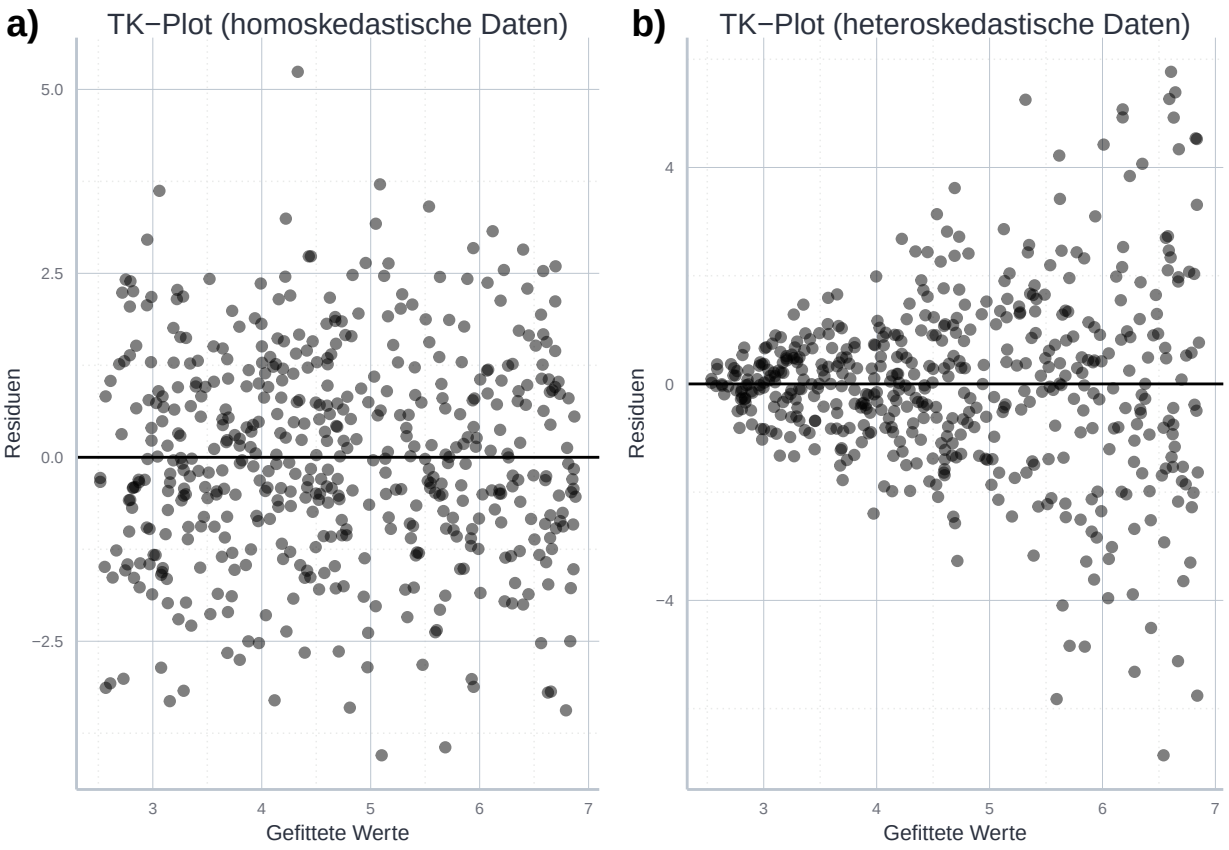
Das alles bedeutet zwar, dass der OLS Schätzer auch im Falle von Heteroskedastie noch erwartungstreu ist, allerdings die Genauigkeit des Schätzers sinkt und die Standardfehler falsch berechnet werden. Da der Fokus hier auf der Beschreibung der Monte-Carlo Simulationsmethode lag werden wir uns mit den möglichen Lösungen erst weiter unten befassen.

1.2 Heteroskedastie

Wie oben beschrieben bedeutet Heteroskedastie, dass die Varianz der Fehlerterme nicht konstant ist.

1.2.1 Liegt Heteroskedastie vor?

Heteroskedastie kann grafisch oder über statistische Tests identifiziert werden. Um Heteroskedastie grafisch zu identifizieren verwenden wir den aus dem vierten Kapitel bekannten Tukey-Anscombe-Plot, in dem wir auf der x-Achse die gefitteten Werte $\hat{Y}$ und auf der y-Achse die Residuen e abbilden:



Im Optimalfall ist die Varianz der Fehler konstant. Das ist in Abbildung (a) der Fall: die Residuen streuen recht zufällt um die Mittelwert 0 herum. In diesem Fall besteht kein Grund zur Annahme, dass Heteroskedastizität vorliegt. Anders in Abbildung (b): hier wird die Varianz nach rechts klar größer. Das lässt große Zweifel an der Annahme der Homoskedastizität aufkommen.

In der Praxis ist es sinnvoll zusätzlich zur grafischen Inspektion noch statistische Tests zu verwenden. Hier gibt es ein breites Angebot an Tests. Viele davon sind in dem Paket lmtest (Zeileis and Hothorn, 2002) gesammelt. Wir gehen auf die mathematische Herleitung der Tests hier nicht ein. Genauere Informationen finden Sie in den unten angegebenen weiterführenden Quellen.

Häufig verwendet wird z.B. der **Breusch-Pagan Test**, den wir mit der Funktion `bptest()` durchführen können. Diese Funktion nimmt als einziges zwingende Argument das Regressionsobjekt. Die weiteren Argumente sollten wir im Normalfall auf den Standardwerten belassen.

Die Nullhypothese des Breusch-Pagan Tests ist Homoskedastie. Wir führen zunächst den Test für den homoskedastischen Fall aus:

```
bptest(schaetzung_homo)
```

```
#>
#>  studentized Breusch-Pagan test
#>
#> data:  schaetzung_homo
#> BP = 0.0067387, df = 1, p-value = 0.9346
```

Wir können H_0 (also die Hypothese der Homoskedastie) nicht ablehnen da $p > 0.05$. Nun führen wir den Test für den heteroskedastischen Fall aus:

```
bptest(schaetzung_hetero)
```

```
#>
#>  studentized Breusch-Pagan test
#>
#> data:  schaetzung_hetero
#> BP = 88.513, df = 1, p-value < 2.2e-16
```

Wir können H_0 (also die Hypothese der Homoskedastie) hier klar ablehnen.

Ein ebenfalls häufig verwendeter Test ist der **Goldfeld-Quandt Test**. Dieser wird mit der Funktion `gqtest()` durchgeführt und hat mehr Freiheitsgrade als der Breusch-Pagan Test: hier testen wir die Hypothese ob die Fehlervarianz in einem Bereich der Daten größer oder kleiner ist als in einem anderen Bereich. Standardmäßig wird der Datensatz dabei in zwei gleich große Teile geteilt, aber der Trennpunkt kann mit dem Argument `point` theoretisch beliebig gewählt werden, genauso wie der Anteil der Daten um den Trennpunkt, die ausgeschlossen werden sollen (Argument `fraction`). Zudem können wir über das Argument `alternative` wählen ob für steigende, sinkende oder andere Varianz getestet werden soll. Diese Wahlmöglichkeiten erhöhen die Power des Tests - wenn wir denn theoretisch gut begründete Werte wählen können. Ansonsten ist es im besten die Standardwerte zu verwenden und den Test mit anderen Tests und grafischen Methoden zu ergänzen.

Wir verwenden zunächst den Test mit der Standardspezifikation:

```
gqtest(schaetzung_homo)
```

```
#>
#>  Goldfeld-Quandt test
#>
#> data:  schaetzung_homo
#> GQ = 0.98576, df1 = 248, df2 = 248, p-value = 0.5449
#> alternative hypothesis: variance increases from segment 1 to 2
```

Für den homoskedastischen Fall kann H_0 (Homoskedastie) also nicht abgelehnt werden.

```
gqtest(schaetzung_hetero)
```

```
#>
#>  Goldfeld-Quandt test
#>
#> data:  schaetzung_hetero
#> GQ = 0.79606, df1 = 248, df2 = 248, p-value = 0.9635
#> alternative hypothesis: variance increases from segment 1 to 2
```

Dagegen muss H_0 für den heteroskedastischen Fall verworfen werden. Aus der grafischen Analyse oben haben wir bereits gesehen, dass die Alternativhypothese von einer steigenden Varianz auszugehen hier viel Sinn macht. Hätten wir aber für sinkende Varianz getestet hätte H_0 *nicht* abgelehnt werden können:

```
gqtest(schaetzung_hetero, alternative = "less")
```

```
#>
#>  Goldfeld-Quandt test
#>
```

```
#> data:  schaetzung_hetero
#> GQ = 0.79606, df1 = 248, df2 = 248, p-value = 0.03655
#> alternative hypothesis: variance decreases from segment 1 to 2
```

Das zeigt die potenzielle Schwäche des GQ-Tests. Wenn wir uns nicht sicher sind ob wir für steigende oder sinkende Varianz testen sollen bietet sich natürlich immer auch der zweiseitige Test an, der aber über eine verminderte Power vefügt, im vorliegenden Falle aber dennoch das richtige Ergebnis liefert:

```
gqtest(schaetzung_hetero, alternative = "two.sided")
```

```
#>
#>  Goldfeld-Quandt test
#>
#> data:  schaetzung_hetero
#> GQ = 0.79606, df1 = 248, df2 = 248, p-value = 0.0731
#> alternative hypothesis: variance changes from segment 1 to 2
```

Wir lernen aus diesen Ergebnissen, dass wir immer mit verschiedenen Methoden auf Heteroskedastie testen sollten und immer sowohl grafische als auch quantitative Tests verwenden sollten. Für den Fall, dass unsere Daten Heteroskedastie aufweisen sollte dann eine der im folgenden beschriebenen Strategien als Reaktion auf Heteroskedastie umgesetzt werden.

1.2.2 Reaktionen auf Heteroskedastie

Aus unseren Vorüberlegungen können wir folgendes festhalten:

1. Der OLS-Schätzer ist auch unter Heteroskedastie erwartungstreu
2. Der OLS-Schätzer ist weiterhin konsistent
3. Die Varianz des OLS-Schätzers ist unter Heteroskedastie größer und der Schätzer ist nicht mehr effizient
4. Die Standardfehler unter Heteroskedastie sind nicht mehr korrekt.

Daraus ergibt sich, dass wir in jedem Fall die Standardfehler korrigieren müssen. Darüber hinaus können wir uns überlegen ob wie die geschätzten Werte des Standard OLS-Schätzers weiterhin verwenden, da der Schätzer ja weiterhin erwartungstreu und konsistent ist, oder ob wir ein alternatives Schätzverfahren implementieren um die Effizienz des Schätzers zu steigern.

Für den ersten Fall korrigieren wir 'einfach' die Standardfehler des OLS-Schätzers, verwenden aber die alten geschätzten Koeffizienten weiter. Im zweiten Fall verwenden wir die Schätzmethode der *Generalized Least Squares* um nicht nur die Standardfehler zu korrigieren sondern auch die Parameter neu zu schätzen. Im folgenden fokussieren wir uns auf die erste Strategie, da die GLS Methode mit neuen Schwierigkeiten einhergeht und nicht ganz einfach zu implementieren ist.

Denn wie gesagt ist der OLS Schätzer weiterhin konsistent. Das bedeutet, dass wir in großen Stichproben eigentlich kein Problem haben. In kleinen Stichproben kann die Verwendung dagegen Effizienzverluste mit sich bringen - aber keinen Verlust der Erwartungstreue. Beim GLS Verfahren schätzen wir die Varianzstruktur. Das funktioniert gut, wenn wir große Stichproben haben. Gerade da ist aber die Verwendung der OLS Schätzers aufgrund seiner Konsistenz aber gar kein Problem. In kleinen Stichproben ist die Schätzung der Varianz dagegen problematisch, solange wir keine theoretischen Restriktionen einführen können. Insofern ist die sinnvolle Anwendung von GLS eher gering, weswegen wir uns im Folgenden darauf beschränken robuste Standardfehler einzuführen.

Die am weitesten verbreitete Korrektur der Standardfehler sind *White's robuste Standardfehler*.[3] Um diese in R zu berechnen bedarf es zweier Schritte. Zunächst verwenden wir die Funktion `vcovHC()` aus dem Paket sandwich (Zeileis, 2004) um eine korrigiert Varianz-Kovarianz-Matrix zu berechnen. Diese Funktion nimmt als notwendiges Argument das Regressionsobjekt. Darüber hinaus können wir über das Argument `type` die genaue Berechnungsmethode festlegen. Mehr Infos dazu findet sich z.B. in der Hilfefunktion. Hier verwenden wir die am häufigsten verwendetete Verion `"HC1"`:

```
var_covar_matrix <- vcovHC(schaetzung_hetero, type = "HC1")
var_covar_matrix
```

```
#>              (Intercept)          x1
#> (Intercept)  0.018596906 -0.004287583
#> x1          -0.004287583  0.001130885
```

Dann können wir die Funktion `coeftest()` aus dem Paket `lmtest` verwenden um die korrigierten Standardfehler zu erhalten:

```
coeftest(schaetzung_hetero, vcov. = var_covar_matrix)
```

```
#>
#> t test of coefficients:
#>
#>             Estimate Std. Error t value  Pr(>|t|)
#> (Intercept) 2.047718   0.136370  15.016 < 2.2e-16 ***
#> x1          0.482333   0.033629  14.343 < 2.2e-16 ***
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Diese unterscheiden sich offensichtlich von den nicht-korrigierten Standardfehlern:

```
summary(schaetzung_hetero)
```

```
#>
#> Call:
#> lm(formula = schaetzgleichung, data = stichprobe_hetero)
#>
#> Residuals:
#>     Min      1Q  Median      3Q     Max
#> -6.8628 -0.8143 -0.0055  0.7927  5.7683
#>
#> Coefficients:
#>             Estimate Std. Error t value Pr(>|t|)
#> (Intercept)   2.0477     0.1753   11.68   <2e-16 ***
#> x1            0.4823     0.0291   16.58   <2e-16 ***
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> Residual standard error: 1.657 on 498 degrees of freedom
```

[3] Die mathematischen Grundlagen behandeln wir hier nicht, sie werden aber in der weiterführenden Literatur erläutert, z.B. in Kapitel 4 von Greene (2018).

```
#> Multiple R-squared:  0.3556, Adjusted R-squared:  0.3543
#> F-statistic: 274.8 on 1 and 498 DF,  p-value: < 2.2e-16
```

Beachten Sie, dass die korrigierten Standardfehler zwar häufig größer sind, dies aber nicht notwendigerweise der Fall sein muss!

1.3 Autokorrelation

Wir sprechen von Autokorrelation wenn die Fehlerterme in der Regression untereinander korreliert sind. Wie bei der Heteroskedastizität ist die Varianz-Kovarianz Matrix eine andere als ursprünglich angenommen: im Falle der Heteroskedastizität lag die Abweichung auf der Hauptdiagonale, also der Varianz der einzelnen Fehlerterme, die nicht wie laut A4 konstant ist. Im Falle der Autokorrelation liegt das Problem abseits der Hauptdiagonale, bei den Kovarianzen der einzelnen Fehler. Standardmäßig nehmen wir an, dass diese Kovarianz gleich Null ist, in der Praxis ist diese Annahme möglicherweise nicht erfüllt.

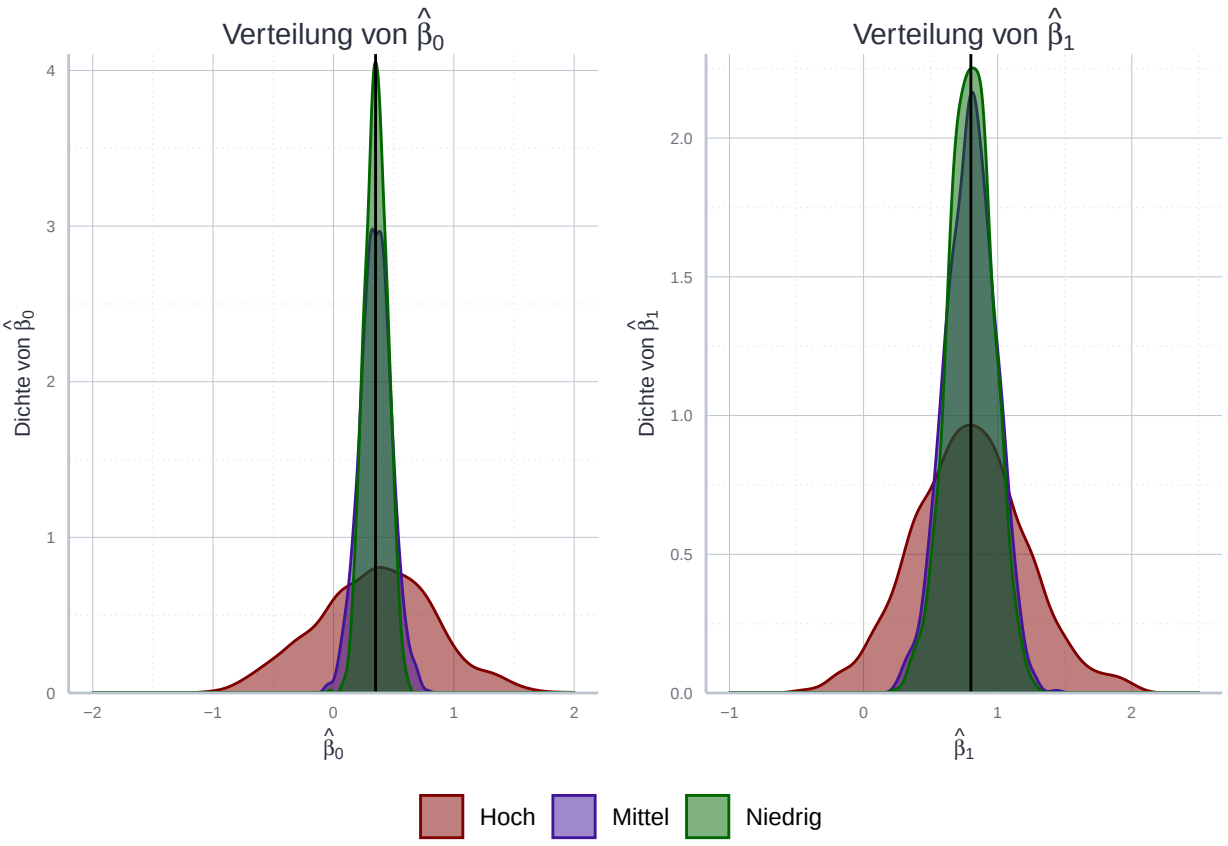
Besonders häufig tritt Autokorrelation auf, wenn wir mit Zeitreihendaten arbeiten. Denn dann ist es sogar sehr plausibel, dass die Fehler einer Beobachtung in t mit denen aus der Vorperiode $t-1$ zusammenhängen. Entsprechend groß ist die Literatur zur Autokorrelation in der Zeitreihenanalyse und Panel-Schätzung. Diese Themenbereich sind jedoch erst viel später unser Thema. Nichtdestotrotz macht es Sinn sich die Folgen von Autokorrelation auch im Querschnittsfall anzusehen.

1.3.1 Folgen von Autokorrelation

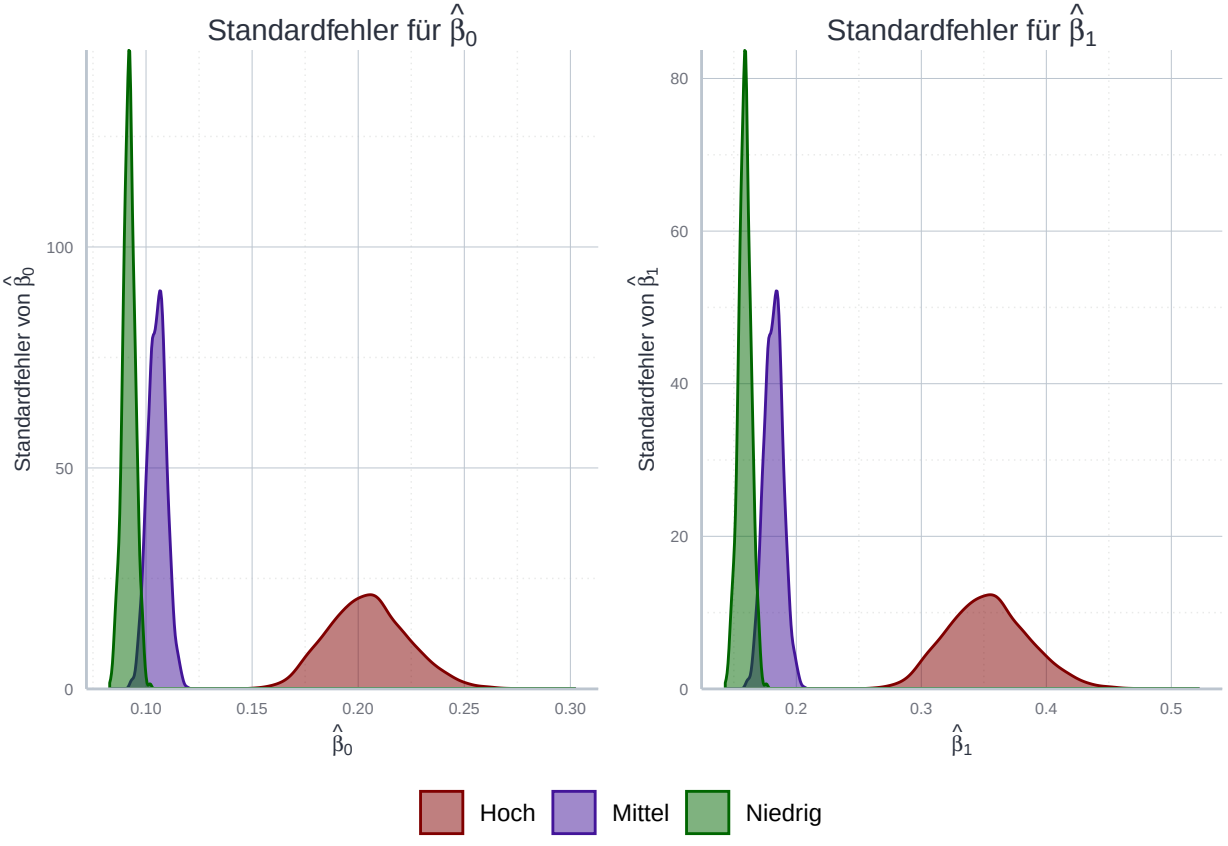
Wir wissen zwar von der Herleitung des OLS-Schätzers bereits, dass Autokorrelation keinen Einfluss auf die Erwartungstreue des Schätzers hat, wir wollen aber dennoch die Folgen von Autokorrelation durch eine kleine MCS illustrieren.

Dazu erstellen wir einen künstlichen Datensatz in dem die Fehler unterschiedlich stark miteinander korreliert sind.

Um die Variablen mit vorher spezifizierter Korrelation zu erstellen verwenden wir wieder die Funktion `mvrnorm` aus dem Paket MASS (Venables and Ripley, 2002). Eine genauere Erläuterung findet sich hier.



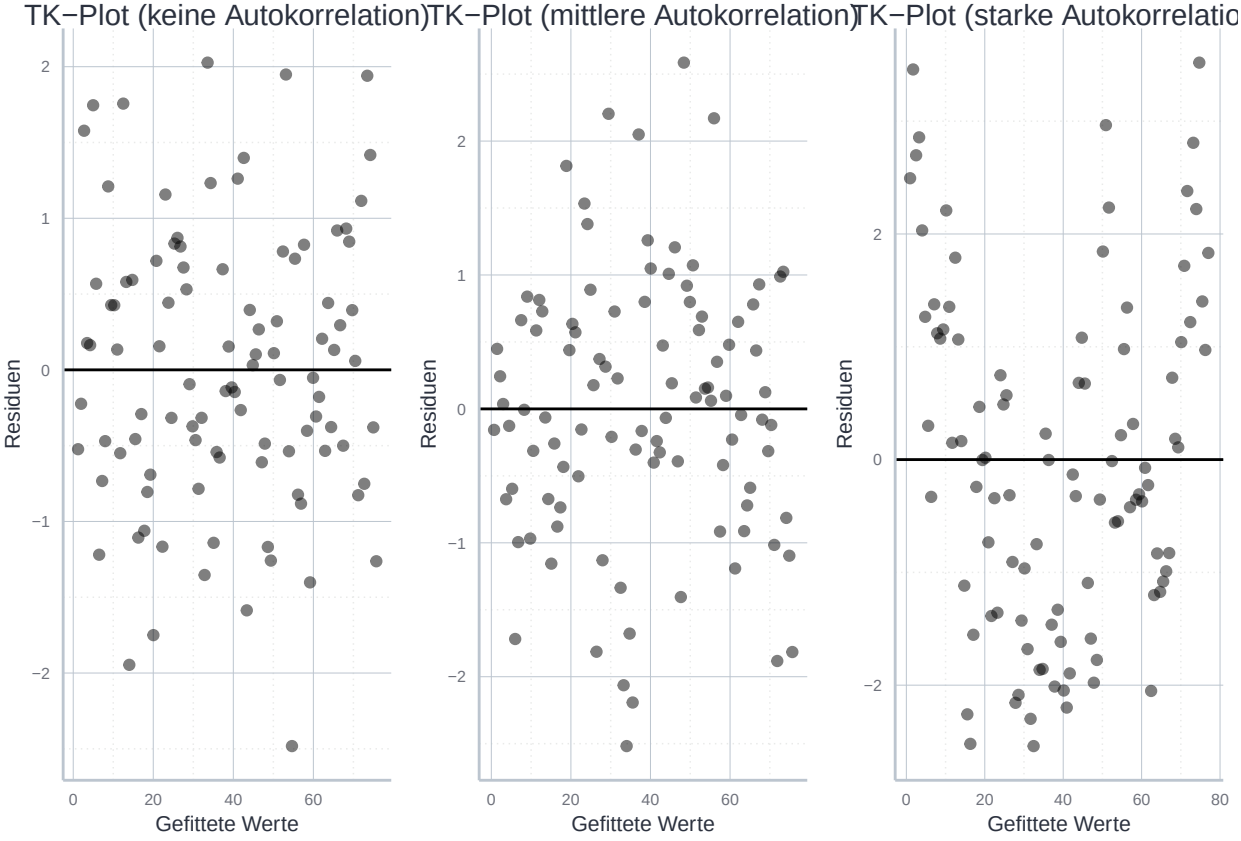
Wie erwartet bleiben die Schätzer erwartungstreu, büßen aber deutlich an Effizienz ein wenn die Autokorrelation größer wird. Betrachten wir nun noch die geschätzten Standardfehler:



Wie bei der Heteroskedastie hat Autokorrelation einen großen Einfluss auf die geschätzten Standardfehler. Da auch hier geschätzten Standardfehler falsch sind müssen wir entsprechend kontrollieren.

1.3.2 Testen auf Autokorrelation

Wie bei der Heteroskedastie sollten wir auch beim Testen auf Autokorrelation grafische und quantitative Tests kombinieren. Für die grafische Analyse verwenden wir wie vorher den Tukey-Anscombe Plot der Residuen. Die Idee ist, dass wenn in den 'echten' Fehlern Autokorrelation vorherrscht wir das auch in den Residuen beobachten können. Die folgende Abbildung verdeutlicht wie wir Autokorrelation in den entsprechenden Abbildungen erkennen können:



Gerade bei Anwendungen außerhalb der Zeitreihenökonometrie ist Autokorrelation aber grafisch nicht so einfach zu identifizieren. Dennoch ist gerade bei der starken Autokorrelation offensichtlich, dass die Fehler in t von denen in $t-1$ abhängen.

> **Die vielen Arten von Autokorrelation** Das Problem beim Testen auf Autokorrelation ist, dass die Fehler natürlich auf sehr viele Arten und Weiten miteinander korreliert sein können. In Zeitreihen beobachten wir häufig einen so genannten *autoregressiven Prozess*, bei dem die Fehler in t folgendermaßen bestimmt sind: $\epsilon_t = \rho\epsilon_{t-1} + u$, wobei $u \propto \mathcal{N}(0, \sigma^2)$. Es sind aber natürlich viele weitere Möglichkeiten denkbar, was es schwierig macht *allgemeine* Tests für Autokorrelation zu entwickeln. Wenn wir aufgrund von theoretischen Überlegungen eine bestimmte Struktur der Autokorrelation vermuten, können wir spezialisierte Tests verwenden, die über deutlich größere Power verfügen als allgemeine Tests. Dieses Thema wird im Kurs zur Zeitreihenökonometrie in größerem Umfang behandelt.

Es gibt diverse Tests für Autokorrelation, die für jeweils unterschiedliche Settings besonders gut oder weniger

gut geeignet sind. Insofern macht es Sinn sich für den konkreten Anwendungsfall die am besten passenden Tests herauszusuchen und immer mehr als einen Test zu verwenden. Im Folgenden werden einige prominente Tests vorgestellt.

Häufig verwendet wird der *Box–Pierce*, bzw. *Ljung–Box* Test, welche die H_0 keiner Autokorrelation testen. Sie unterscheiden sich in der genauen Berechnung der Teststatistik und können als Alternativhypothese eine Autokorrelation von unterschiedlichen Graden testen. Mit unterschiedlichen Graden meinen wir die Anzahl der Lags zwischen den Beobachtungen, deren Fehler noch miteinander korreliert sind. Standardmäßig testen wir gegen eine Autokorrelation mit Grad 1, allerdings können je nach Anwendungsfall auch höhere Grade sinnvoll sein.

Die Funktion `Box.test()` kann verwendet werden um diese Tests durchzuführen. Das erste Argument sind immer die Residuen der zu untersuchenden Regression, mit dem Argument `type` wird dann der Test (`"Box-Pierce"` oder `"Ljung-Box"`) ausgewählt und mit `lag` der Grad der Autokorrelation. Entsprechend testen wir folgendermaßen auf eine Autokorrelation mit Grad 1:

```
Box.test(mid_acl$residuals, lag = 1, type = "Box-Pierce")
```

```
#>
#>  Box-Pierce test
#>
#> data:  mid_acl$residuals
#> X-squared = 9.5899, df = 1, p-value = 0.001957
```

bzw.:

```
Box.test(mid_acl$residuals, lag = 1, type = "Ljung-Box")
```

```
#>
#>  Box-Ljung test
#>
#> data:  mid_acl$residuals
#> X-squared = 9.8805, df = 1, p-value = 0.00167
```

In beiden Fällen muss H_0 abgelehnt werden. Wir müssen also von Autokorrelation ausgehen!

Ein anderer bekannter Test auf Autokorrelation ist der *Durbin-Watson Test*, der allerdings nicht besonders robust ist. Wir können diesen Test mit der Funktion `dwtest()` aus dem Paket `lmtest` implementieren. Dazu übergeben wir als erstes Argument das Schätzobjekt der zu überprüfenden Schätzung:

```
dwtest(small_acl)
```

```
#>
#>  Durbin-Watson test
#>
#> data:  small_acl
#> DW = 1.9569, p-value = 0.3741
#> alternative hypothesis: true autocorrelation is greater than 0
```

H_0 des DW-Tests ist keine Autokorrelation. Im aktuellen Fall können wir H_0 (keine Autokorrelation) nicht ablehnen und wir brauchen uns keine Gedanken über Autokorrelation machen. Allerdings können wir die Alternativhypothese des Tests selbst über das Argument `alternative` festlegen. Wir haben dabei die Wahl zwischen verschiedenen Strukturen der Autokorrelation, nämlich ob die Fehler in zukünftigen Beobachtungen *positive* (`alternative="greater"`)

oder *negativ* (`alternative="less"`) von dem Fehler in der aktuellen Beobachtung abhängen. Sind wir uns unsicher wählen wir am besten einen zweiseitigen Test (`alternative="two.sided"`). Wie immer ist die Power des Tests größer wenn wir H_1 restriktiver wählen.

Im folgenden Beispiel ist die tatsächliche Autokorrelation positiv. Die Rolle der gewählten H_1 wird so deutlich:

```
dwtest(mid_acl, alternative = "greater")
```

```
#>
#>  Durbin-Watson test
#>
#> data:  mid_acl
#> DW = 1.3469, p-value = 0.0003333
#> alternative hypothesis: true autocorrelation is greater than 0
```

Hier gibt der Test also korrektermaßen Autokorrelation an. Testen wir dagegen gegen die 'falsche' H_1:

```
dwtest(mid_acl, alternative = "less")
```

```
#>
#>  Durbin-Watson test
#>
#> data:  mid_acl
#> DW = 1.3469, p-value = 0.9997
#> alternative hypothesis: true autocorrelation is less than 0
```

In diesem Fall wird keine entsprechende Autokorrelation gefunden. Im Zweifel ist daher der zweiseitige Test vorzuziehen:

```
dwtest(mid_acl, alternative = "two.sided")
```

```
#>
#>  Durbin-Watson test
#>
#> data:  mid_acl
#> DW = 1.3469, p-value = 0.0006667
#> alternative hypothesis: true autocorrelation is not 0
```

Hier wird H_0 wieder korrektermaßen verworfen.

Zuletzt wollen wir noch den *Breusch-Godfrey Test* einführen, der als relativ robust und breit anwendbar gilt. Er wird mit der Funktion `bgtest()` aus dem Pakelt `lmtest` durchgeführt. Hier wird als erstes Argument wieder das Regressionsobjekt übergeben. Als Spezifikationsalternativen können wir wiederum den höchsten Grad der zu testenden Autokorrelation (Argument `order`) und die Art der Teststatisik (Argument `type`) auswählen.

Zum Beispiel:

```
bgtest(mid_acl, order = 1, type = "F")
```

```
#>
#>  Breusch-Godfrey test for serial correlation of order up to 1
#>
#> data:  mid_acl
```

```
#> LM test = 10.702, df1 = 1, df2 = 97, p-value = 0.001483
```

Oder:

```
bgtest(mid_acl, order = 1, type = "Chisq")
```

```
#>
#>  Breusch-Godfrey test for serial correlation of order up to 1
#>
#> data:  mid_acl
#> LM test = 9.9367, df = 1, p-value = 0.00162
```

Insgesamt bedard die richtige Wahl des Tests einige theoretische Überlegunden für den Anwendungsfall und wir sollten uns nicht auf das Ergebnis eines einzelnen Tests verlassen!

1.3.3 Reaktionen auf Autokorrelation

Falls wir Autokorrelation in den Residuen finden sollten wir aktiv werden und die Standardfehler unserer Schätzung äquivalent zur Heteroskedastie korrigieren. Da der Schätzer selbt weiterhin erwartungstreu ist können wir die OLS-Schätzer als solche weiterverwenden. Effizienzgewinne sind durch alternative Schätzverfahren möglich, werden hier aber nicht weiter verfolgt.

Das Vorgehen ist dabei quasi äquivalent zum Fall der Heteroskedastie. Wir berechnen wieder zunächst eine robuste Varianz-Kovarianzmatrix mit der Funktion `vcovHAC()` aus dem Paket `sandwich` und korrigieren dann die Standardfehler mit der Funktion `coeftest()` and dem Paket `lmtest`. Beachten Sie, dass die resultierenden Standardfehler robust sowohl gegen Heteroskedastie als auch Autokorrelation sind.

```
var_covar_matrix <- vcovHC(large_acl, type = "HC0")
coeftest(large_acl, vcov. = var_covar_matrix)
```

```
#>
#> t test of coefficients:
#>
#>             Estimate Std. Error  t value Pr(>|t|)
#> (Intercept) 0.186245   0.343932   0.5415   0.5894
#> x           0.768353   0.005885 130.5620   <2e-16 ***
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Diese unterscheiden sich offensichtlich von den nicht-korrigierten Standardfehlern:

```
summary(large_acl)
```

```
#>
#> Call:
#> lm(formula = y ~ x, data = dgp_acl(0.5, 0.75, 1:100, 0.85))
#>
#> Residuals:
#>     Min      1Q  Median      3Q     Max
#> -2.5399 -1.1799 -0.1028  1.0744  3.5191
```

```
#>
#> Coefficients:
#>             Estimate Std. Error t value Pr(>|t|)
#> (Intercept) 0.186245   0.304383   0.612    0.542
#> x           0.768353   0.005233 146.833   <2e-16 ***
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> Residual standard error: 1.511 on 98 degrees of freedom
#> Multiple R-squared:  0.9955, Adjusted R-squared:  0.9954
#> F-statistic: 2.156e+04 on 1 and 98 DF,  p-value: < 2.2e-16
```

1.4 Multikollinearität

In den OLS-Annahmen schließen wir lediglich *perfekte Multikollinearität* aus. Diese läge vor, wenn eine erklärende Variable eine lineare Funktion einer anderen erklärenden Variable wäre. Da wir in diesem Falle die Matrix $\boldsymbol{X}$ in der Formel für $\hat{\boldsymbol{\beta}}$ nicht invertierbar, wir können $\boldsymbol{X'X}\hat{\boldsymbol{\beta}} = \boldsymbol{X'Y}$ also nicht berechnen und der OLS Schätzer ist überhaupt nicht identifizierbar.

Es zeigt sich jedoch, dass bereits die Existenz von moderater Multikollinearität wichtige Implikationen für den OLS Schätzer hat. Wir sprechen von moderater Multikollinearität wenn zwei oder mehrere erklärende Variablen miteinander korrelieren. Wie wir sehen werden nimmt in diesem Falle die Schätzgenauigkeit ab, weswegen man die Inklusion stark miteinander korrlierter erklärenden Variablen vermeiden sollte.

1.4.1 Folgen von Multikollinearität

Gehen wir einmal von folgender Regressionsgleichung aus:

$$Y_i = \hat{\beta}_0 + \hat{\beta}_1 x_{i1} + \hat{\beta}_2 x_{i2} + e$$

Zunächst wollen wir den Effekt von Multikollinearität per Monte Carlo Simulation ergründen. Zu diesem Zweck erstellen wir drei Datensätze: einen mit wenig, einen mit mittel und einen mit stark korrelierten erklärenden Variablen. Davon abgesehen bleiben die OLS Annahmen erfüllt. Um die Variablen mit vorher spezifizierter Korrelation zu erstellen verwenden wir wieder die Funktion `mvrnorm` aus dem Paket MASS (Venables and Ripley, 2002). Eine genauere Erläuterung findet sich hier.

```
set.seed("123")
stichprobengroesse <- 500
r_small <- 0.0
r_mid <- 0.4
r_large <- 0.9

data_small = mvrnorm(n=stichprobengroesse, mu=c(0, 0),
                     Sigma=matrix(c(1, r_small, r_small, 1),
                                  nrow=2), empirical=TRUE)
```

```
data_mid = mvrnorm(n=stichprobengroesse, mu=c(0, 0),
                   Sigma=matrix(c(1, r_mid, r_mid, 1),
                                nrow=2), empirical=TRUE)

data_large = mvrnorm(n=stichprobengroesse, mu=c(0, 0),
                     Sigma=matrix(c(1, r_large, r_large, 1),
                                  nrow=2), empirical=TRUE)

x_1_small = data_small[, 1]
x_1_mid = data_mid[, 1]
x_1_large = data_large[, 1]

x_2_small = data_small[, 2]
x_2_mid = data_mid[, 2]
x_2_large = data_large[, 2]

cor(x_1_small, x_2_small)  # Test
```

```
#> [1] -1.929638e-16
```

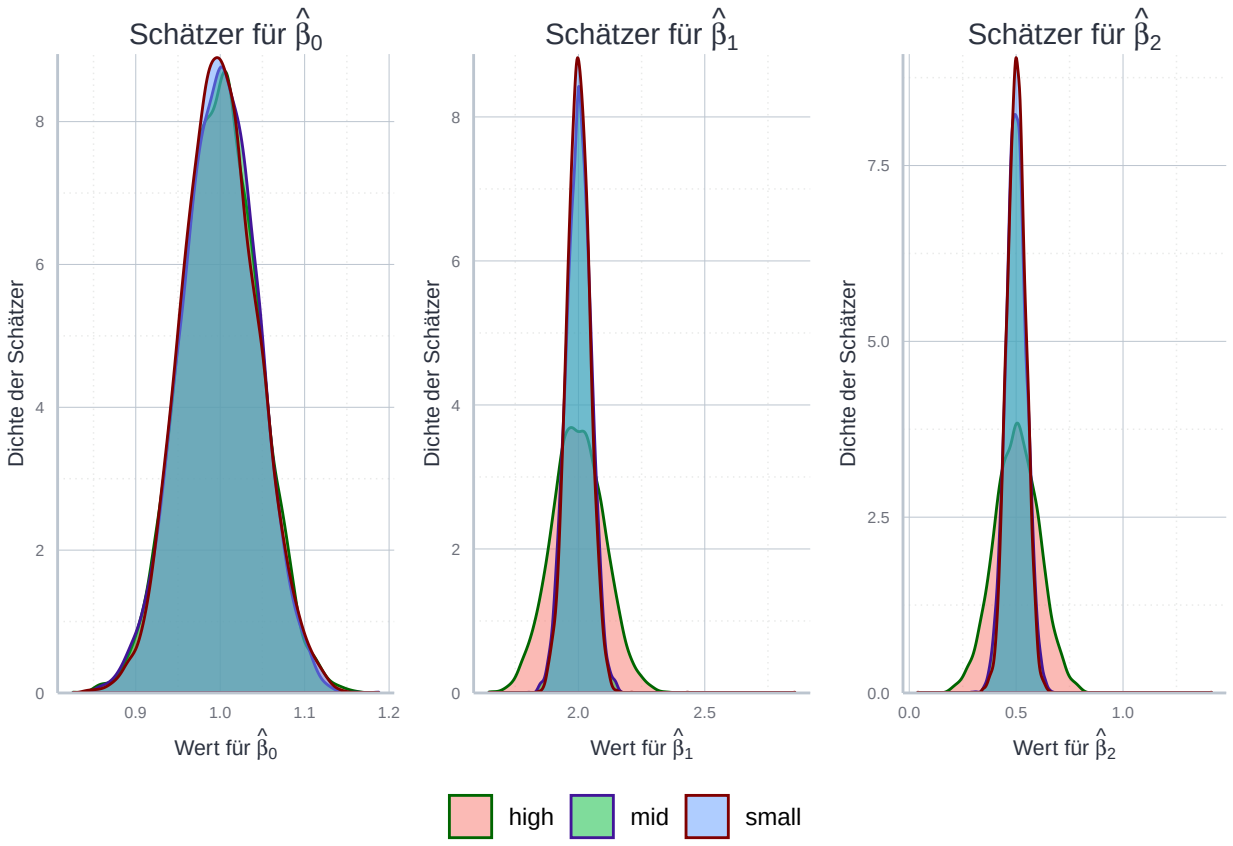
```
cor(x_1_mid, x_2_mid)  # Test
```

```
#> [1] 0.4
```

```
cor(x_1_large, x_2_large)  # Test
```

```
#> [1] 0.9
```

Analog zum Vorgehen oben führen wir nun eine Monte Carlo Simulation durch, in der wir wiederholt Stichproben aus einem künstlich generierten Datensatz ziehen und das oben beschriebene Modell mit Hilfe von OLS schätzen. Dies führt zu folgender Verteilung der Schätzer:



Wie wir sehen wird die Schätzgenauigkeit für die Schätzer von β_1 und β_2 deutlich reduziert! Auf den Schätzer des Achsenabschnitts hat Multikollinearität dagegen keinen Einfluss.

Auch analytisch kann der Effekt von Multikollinearität gezeigt werden. Betrachten wir dazu die folgenden *Hilfsregressionen*:

$$x_{i1} = \hat{\beta}_0^a + \hat{\beta}_3^a x_{i3} + e^a \tag{1.2}$$

$$x_{i2} = \hat{\beta}_0^a + \hat{\beta}_2^a x_{i1} + e^a \tag{1.3}$$

$$\tag{1.4}$$

Bei k erklärenden Variablen ergeben sich die $k-1$ Hilfsregressionen durch eine Umstellung bei der wir eine erklärenden Variable auf die LHS der Regressionsgleichung ziehen und alle weiteren erklärenden Variablen auf der RHS belassen. Im folgenden Bezeichnen wir mit R_h^2 das Bestimmtheitsmaß der h-ten Hilfsregression (also der Hilfsregression mit x_{ih} als abhängiger Variable).

Es kann nun gezeigt werden, dass für die Varianz des Schätzers $\hat{\boldsymbol{\beta}}$ folgendes gilt (siehe Greene (2018) für Details):

$$Var(\beta_h) = \frac{\sigma^2}{(1 - R_h^2) \sum_{i=1}^{n} (x_{ih} - \bar{x_h})^2}$$

Hieraus wird unmittelbar ersichtlich, dass die Varianz des Schätzers steigt je größer die Bestimmtheitsmaße der Hilfsregressionen ist!

Gleichzeitig wissen wir aus den Herleitungen oben auch, dass Multikollinearität keinen Einfluss auf die Er-

wartungstreue oder Effizienz des OLS-Schätzers hat.[4]

1.4.2 Testen auf Multikollinearität

Da der Begriff der Multikollinearität nicht exakt definiert ist gibt es natürlich auch keinen exakten Test. Die Frage welches Ausmaß an Korrelation zwischen den erklärenden Variablen akzeptabel ist ist auch immer eine individuelle Entscheidung. Es haben sich jedoch einige Faustregeln herausgebildet die zumindest hilfreich sind um festzustellen ob Multikollinearität die Größe der Standardfehler in unserer Regression erklären kann.

Zu diesem Zweck führen wir wieder die *Hilfsregressionen* von oben durch durch. Die Bestimmtheitsmaße R^2 dieser Hilfsregressionen geben uns einen Hinweis auf das Ausmaß der Korrelation zwischen den erklärenden Variablen. Ist eines der Bestimmtheitsmaße ähnlich groß wie das Bestimmtheitsmaß der 'originalen' Regression macht es Sinn sich über Multikollinearität Gedanken zu machen.

Alternativ können wir uns natürlich auch die paarweisen Korrelationen der erklärenden Variablen anschauen, allerdings berücksichtigt das nicht die Korrelation mehrerer Variablen untereinander - die Hilfsregressionen sind da der bessere Weg!

1.4.3 Reaktionen auf Multikollinearität

Grundsätzlich sollten Sie es vermeiden, stark miteinander korrlierte Variablen gemeinsam als erklärende Variablen in einer Regression zu verwenden. Gleichzeitig werden wir weiter unten sehen, dass das Weglassen von Variablen schwerwiegende Konsequenzen für die Erwartungstreue des OLS-Schätzers haben kann (Stichtwort *omitted variable bias*, siehe unten). Insofern müssen wir immer sehr gut überlegen ob wir eine Variable aus der Schätzgleichung eliminieren können.

Manchmal können wir die Daten transformieren um die Multikollinearität zu senken oder alternative Variablen erheben, häufig bleibt uns aber auch nichts anderes übrig als uns zu ärgern und die Kröte der Multikollinearität zu schlucken.

1.5 Vergessene Variablen

Stellen wir uns vor der 'wahre' Datengenerierende Prozess sie folgendermaßen aus:

$$\boldsymbol{y} = \beta_0 + \beta_1 \boldsymbol{x}_1 + \beta_2 \boldsymbol{x}_2 + \boldsymbol{\epsilon}$$

Aufgrund geistiger Umnachtung haben wir in unserem Modell $\boldsymbol{x}_2$ aber nicht berücksichtigt. Unser geschätztes Modell ist also:

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

Wir haben also eine erklärende Variable vergessen. Dies ist ein praktisch hochrelevantes Problem, denn häufig hat man relevante Variablen nicht auf dem Schirm oder es gibt zu uns relevant erscheinenden Variablen keine Daten.

[4] Beachten Sie, dass wir den Begriff *Effizienz* hier immer relativ verwenden: unter Multikollinearität wird der OLS-Schätzer weniger genau, aber er bleibt dennoch der genauste Schätzer, den wir zur Verfügung haben.

Die Frage, die sich nun stellt: was sind die Implikationen vergessener Variablen? Die Antwort ist recht unbequem, da wir hier nicht so glimpflich wie bisher davon kommen: im Falle vergessener Variablen ist Annahme A2 nicht mehr erfüllt und unser Schätzer $\hat{\boldsymbol{\beta}}$ ist nun weder erwartungstreu noch konsistent - und zwar für alle unabhänigen Variablen in der Regression!

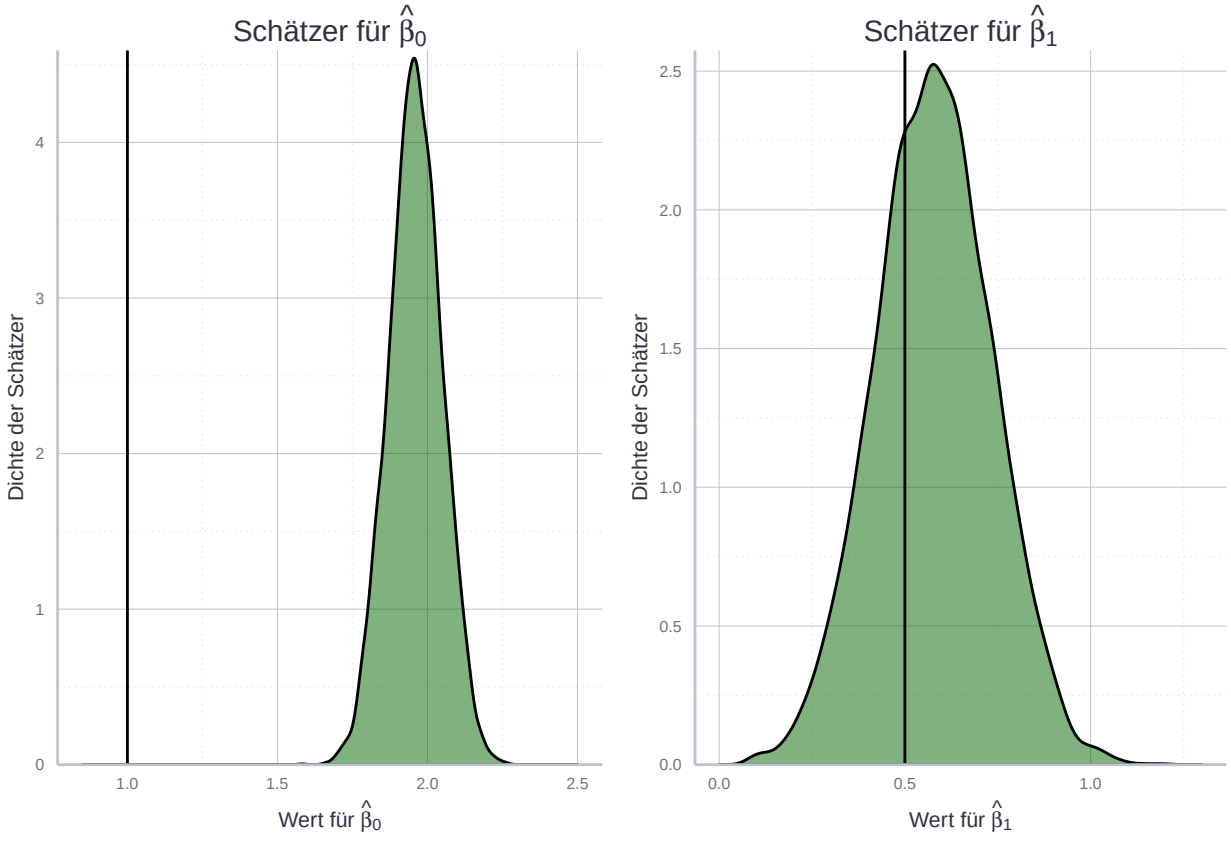
1.5.1 Folgen vergessener Variablen

Zunächst werden wir die Effekt von einer vergessenen Variable per Monte Carlo Simulation illustrieren. Zu diesem Zweck erzeugen wir Daten gemäß des Modells

$$\boldsymbol{y} = \beta_0 + \beta_1 \boldsymbol{x}_1 + \beta_2 \boldsymbol{x}_2 + \boldsymbol{\epsilon}$$

schätzen aber nur folgende Spezifikation:

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



Wir sehen also, dass unser OLS-Schätzer nun nicht mehr erwartungstreu sind! Dies können wir auch recht einfach analytisch zeigen. Nehmen wir generell an, das korrekte Modell ist gegeben durch:

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{z}\gamma + \boldsymbol{\epsilon}$$

wobei $\boldsymbol{z}$ hier eine unabhängige Variable ist, die wir normalerweise in $\boldsymbol{X}$ inkludiert hätten, hier zu Illustrationszwecken jedoch separat angeben um zu zeigen, was passiert wenn wir diese Variable vergessen. γ ist der zugehörige zu

schätzende Parameter.

Wenn wir diese Gleichung nun schätzen ohne $\boldsymbol{z}$ zu berücksichtigen bekommen wir folgenden Schätzer:

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{X'X})^{-1}\boldsymbol{X'y} = \boldsymbol{\beta} + (\boldsymbol{X'X})^{-1}\boldsymbol{X'z\gamma} + (\boldsymbol{X'X})^{-1}\boldsymbol{X'\epsilon}$$

Daraus resultiert, dass:

$$\mathbb{E}(\hat{\boldsymbol{\beta}}|\boldsymbol{X}, \boldsymbol{z}) = \boldsymbol{\beta} + (\boldsymbol{X'X})^{-1}\boldsymbol{X'z\gamma}$$

Das bedeutet, dass $\hat{\boldsymbol{\beta}}$ nicht erwartungstreu ist, es sei denn (1) $\gamma = 0$ oder (2) $\boldsymbol{X'z} = 0$. Fall (1) würde bedeuten, dass $\boldsymbol{z}$ für die Analyse unserer abhängigen Variable gar nicht relevant wäre. Das würde bedeuten, wir hätten die Variable nicht 'vergessen', sondern zu Recht nicht inkludiert. Fall (2) würde bedeutetn, dass $\boldsymbol{z}$ mit keiner der anderen erklärenden Variablen korreliert. Es ist sehr unwahrscheinlich, dass dies der Fall ist sollte $\boldsymbol{z}$ tatsächlich relevant für die Erklärung von $\boldsymbol{y}$ sein.

Das Vergessen relevanter Variablen führt also zu einer Korrelation der andren unabhängigen Variablen mit dem Fehlerterm, da der Effekt von $\boldsymbol{z}$ dann im Fehlerterm steckt und dieser dann mit den anderen unabhängigen Variablen korreliert. Zudem gilt, dass $\mathbb{E}(\epsilon) \neq 0$. Das alles geht mit einem Verlust der Erwartungstreue und auch der Konsistenz des Schätzers einher. Daher können wir die Verzerrung auch durch eine Vergößerung der Stichprobe nicht beheben.

1.5.2 Testen auf vergessene Variablen

Da wir den wahren datenerzeugenden Prozess nicht kennen ist es unmöglich direkt zu testen ob wir eine relevante Variable vergessen haben. Es gibt einen möglichen Test, der die Verwendung von *Instrumentenvariablen* einschließt - ein Thema, das wir später behandeln werden - allerdings basiert auch dieser Test dann wiederum auch nicht zu testenden Annahmen. Insgesamt müssen wir uns hier also vor allem auf unsere theoretischen Überlegungen verlassen: wir müssen überlegen welche Variablen einen Einfluss auf unsere zu erklärende Variable haben könnten und diese Variablen müssen dann auf die eine oder andere Weise in der Regression berücksichtigt werden!

1.5.3 Reaktion auf vergessene Variablen

Das ist diesmal einfach: fügen Sie 'einfach' die relevanten Variablen zu ihrer Regression hinzu. Wenn Sie dazu keine Daten haben hilft Ihnen allerhöchstens die Verwendung von *Instrumentenvariablen*, einem Thema, das wir später in der Vorlesung behandeln werden.

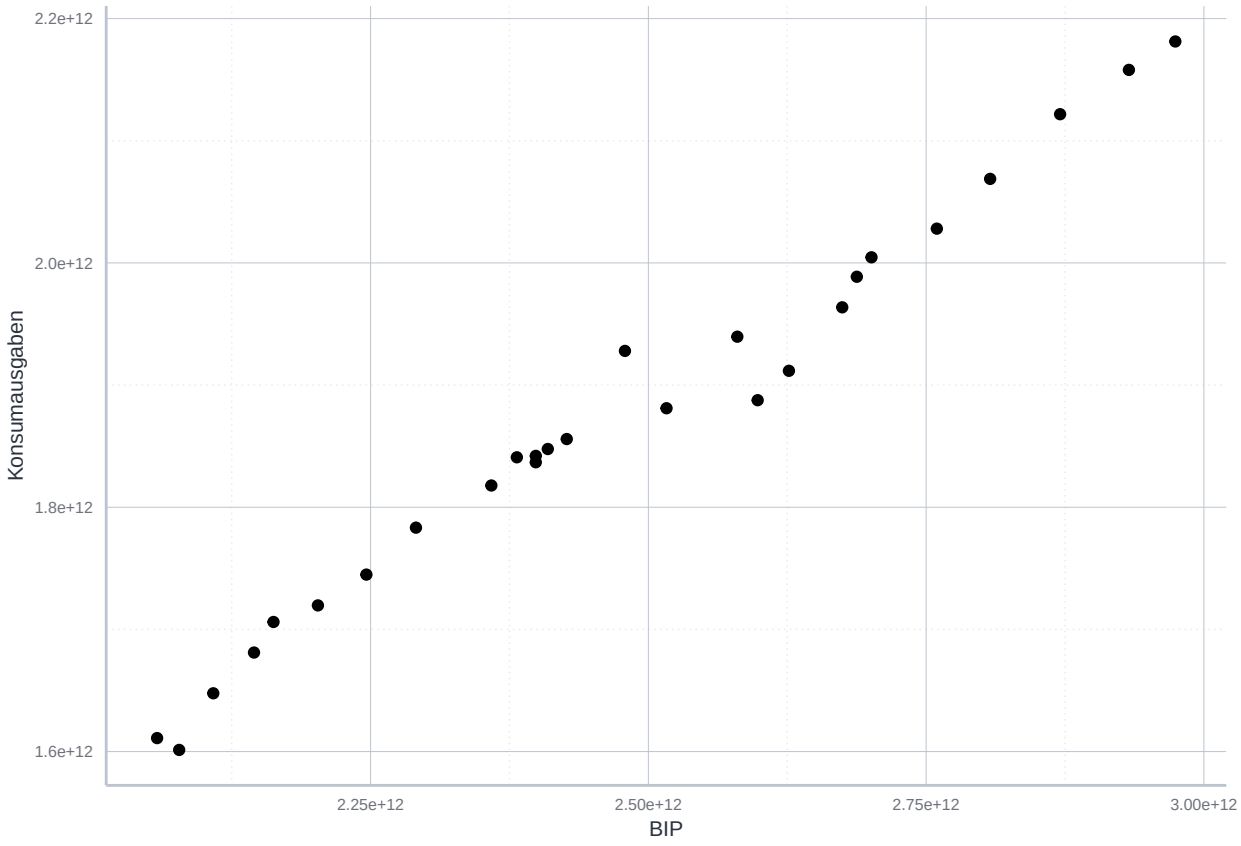
1.6 Falsche funktionale Form

Eine zentrale Annahme des linearen Regressionsmodells ist die Linearität des datenerzeugenden Prozesses (A1). Wenn diese Annahme verletzt ist wäre unser Schätzer weder erwartungstreu noch konsistent.

Wir haben aber auch gelernt, dass die Annahme der Linearität sich nur auf die *Parameter* bezieht. Das bedeutet, dass bestimmte nicht-lineare Zusammenhänge durchaus mit OLS geschätzt werden können, wenn wir die Daten entsprechend transformieren. Dies geschieht durch die Wahl der funktionalen Form. Am besten wir illustrieren dies durch ein univariates Beispiel.

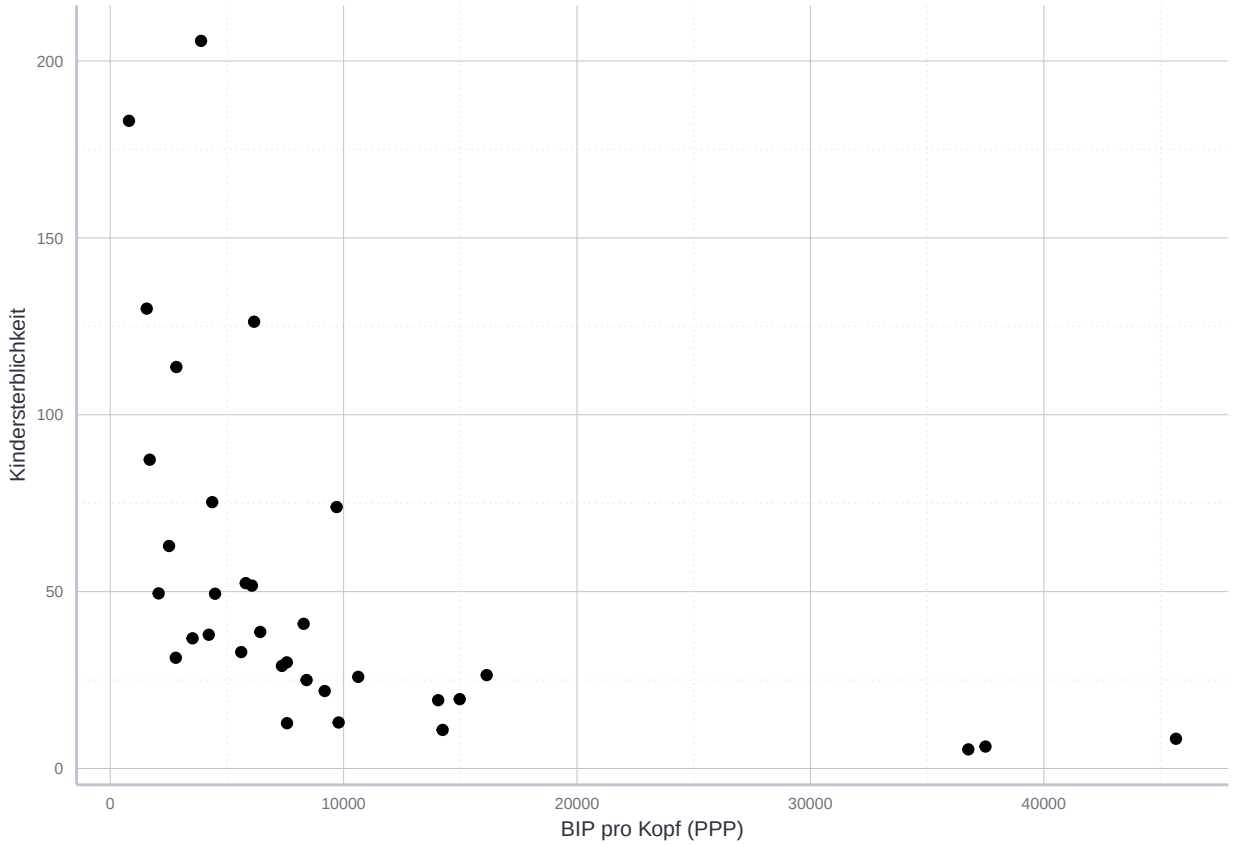
So ist auf den ersten Blick ersichtlich, dass der Zusammenhang zwischen BIP und Konsumausgaben direkt linear ist:

```
bipkonsum <- fread(here("data/tidy/BIPKonsum.csv"),
                   colClasses = rep("double", 3))
ggplot(data = bipkonsum, aes(x=BIP, y=Konsumausgaben)) +
  geom_point() + theme_icae()
```



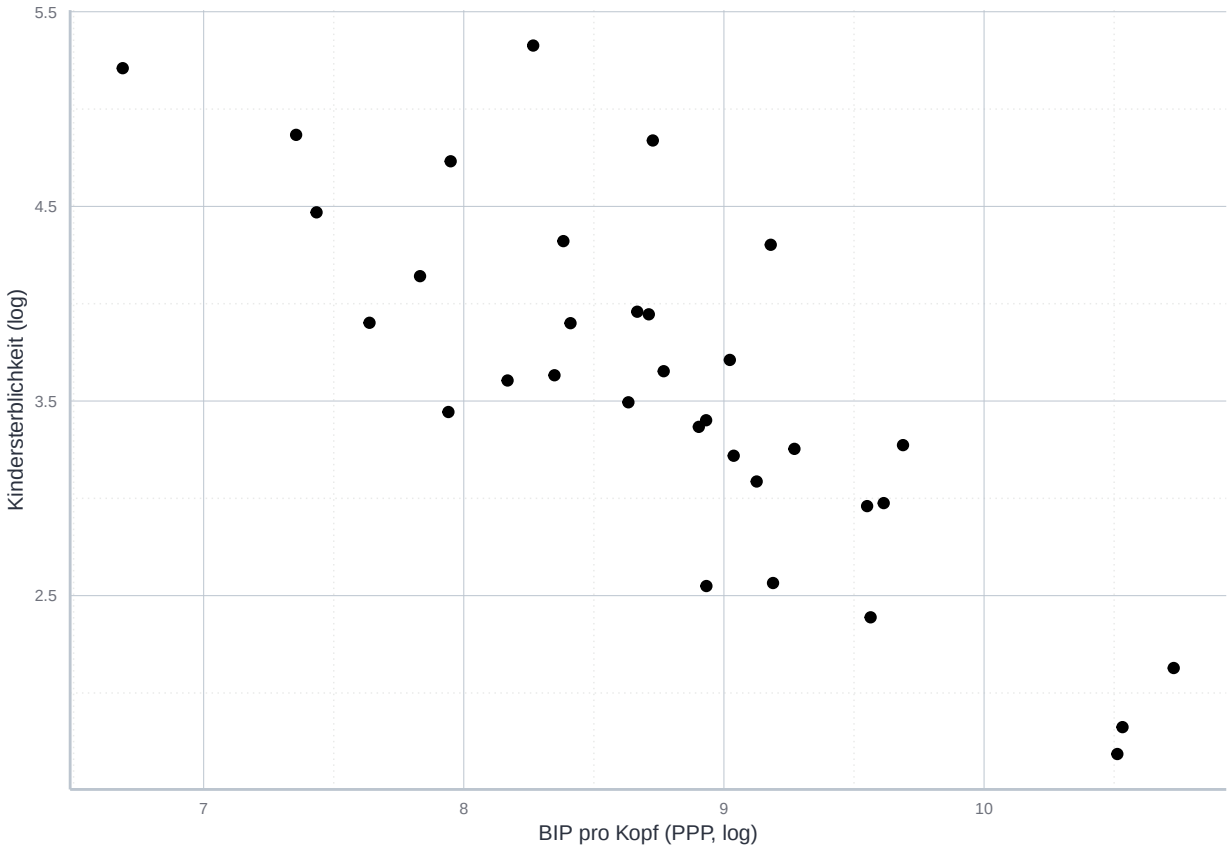
Wir könnten den Zusammenhang also unmittelbar mit OLS schätzen ohne gegen Annahme A1 zu verstoßen.

Der Zusammenhang zwischen BIP pro Kopf und Kindersterblichkeit im Jahr 2000 erscheint dagegen nicht linear zu sein:



Wenn wir diesen Zusammenhang mit OLS schätzen würden würden wir klar gegen Annahme A1 verstoßen. Die Konsequenz wäre, dass unser Schätzer weder erwartungstreu, noch konsistent noch effizient wäre.

Gleichzeitig können wir durch Wahl einer alternativen funktionalen Form den Zusammenhang linearisieren. Dazu nehmen wir einfach den Logarithmus:



Diesen Zusammenhang können wir nun mit OLS schätzen ohne gegen A1 zu verstoßen! Das zeigt, dass die falsche Wahl der funktionalen Form, also die nicht korrekte Transformation der Variablen, große Implikationen für die Eigenschaften unserer Schätzer haben kann!

1.6.1 Folgen einer falschen funktionalen Form

Wie bereits erwähnt bezieht sich die Wahl der funktionalen Form direkt auf Annahme A1. Wir wir oben gesehen haben ist diese Annahme wichtig um die Konsistenz und Erwartungstreue des OLS-Schätzers herzuleiten. Mit anderen Worten: ist A1 nicht erfüllt, z.B. durch die Wahl einer falschen funktionalen Form, ist der OLS-Schätzer nicht mehr erwartungstreu und konsistent. Wir müssen also entweder die funktionale Form ändern oder ein anderes Schätzverfahren wählen.

1.6.2 Testen auf die richtige funktionale Form

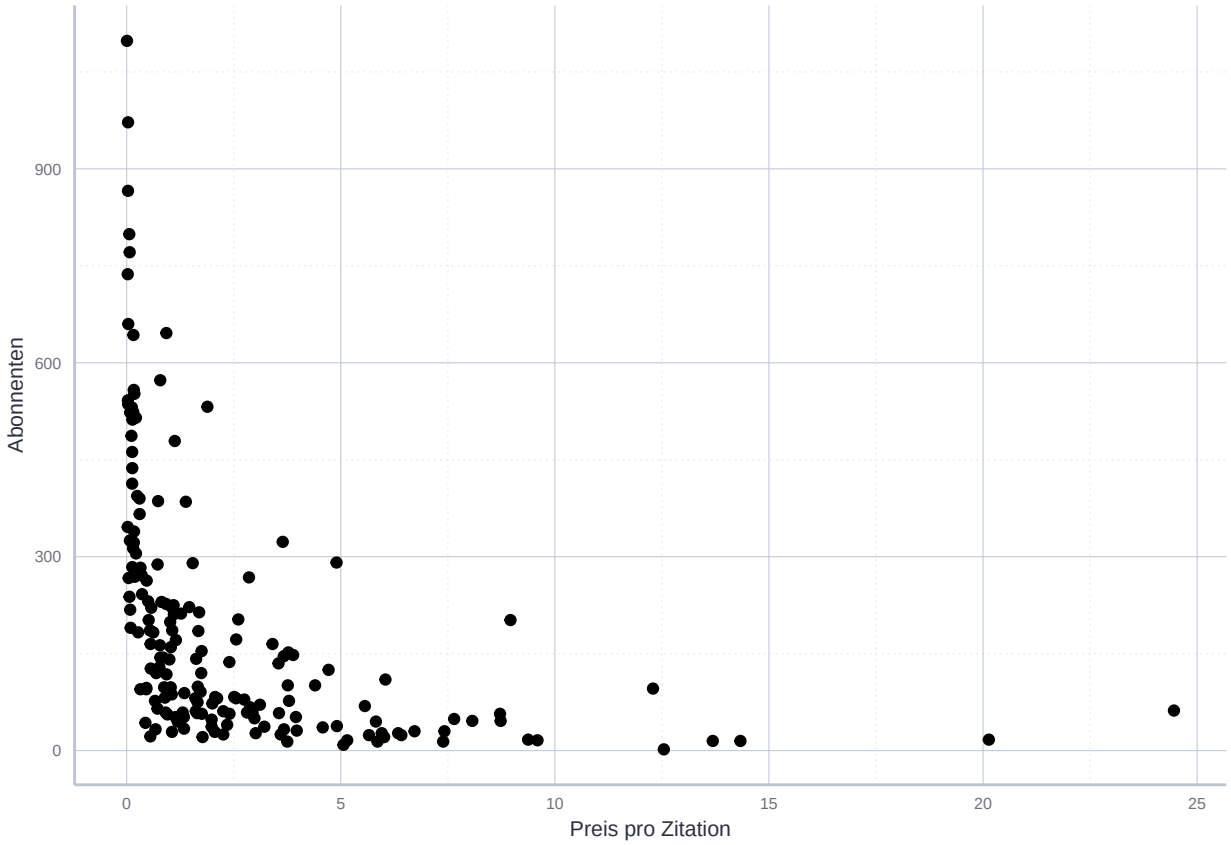
Bei der Wahl der funktionalen Form spielen vor allem theoretische Überlegungen eine wichtige Rolle. Auch eine Inspektion der paarweisen Beziehungen zwischen abhängiger und unabhängigen Variablen ist hilfreich.

Eine wirksame Methode zur Überprüfung unserer funktionalen Form ist dagegen die Inspektion des Tukey-Anscombe Plots. Haben wir die richtige Form gewählt werden wir hier keine Struktur erkennen können. Zeigen die Residuen jedoch eine klare Struktur auf ist das ein Signal, dass wir eine andere funktionale Form ausprobieren sollten. Natürlich kann die Struktur der Residuen auch andere Gründe haben, z.B. Heteroskedastie. Für diese Gründe gibt es jedoch zusätzlich noch statistische Tests sodass wir durch sukszessives Testen und Ausprobieren eine angemessene funktionale Form identifizieren können.

Es gibt auch einige Tests, die manchmal verwendet werden um die richtige Wahr der funktionalen Form zu überprüfen. Der bekannteste Test ist dabei der so genannte *RESET Test*. *RESET* steht dabei für *REgression Specification Error Test*. Dieser Test wird mit der Funktion `resettest()` durchgeführt und testes die H_0, dass wir die richtige funktionale Form gewählt haben.

Wir illustrieren den Test anhand folgenden Beispiels, in dem wir den uns bereits bekannten Datensatz zu Journaldaten analysieren.

Wir betrachten den Zusammenhang zwischen Abonnenten und dem Preis pro Zitation. Wie wir hier sehen ist dieser Zusammenhang alles andere linear:



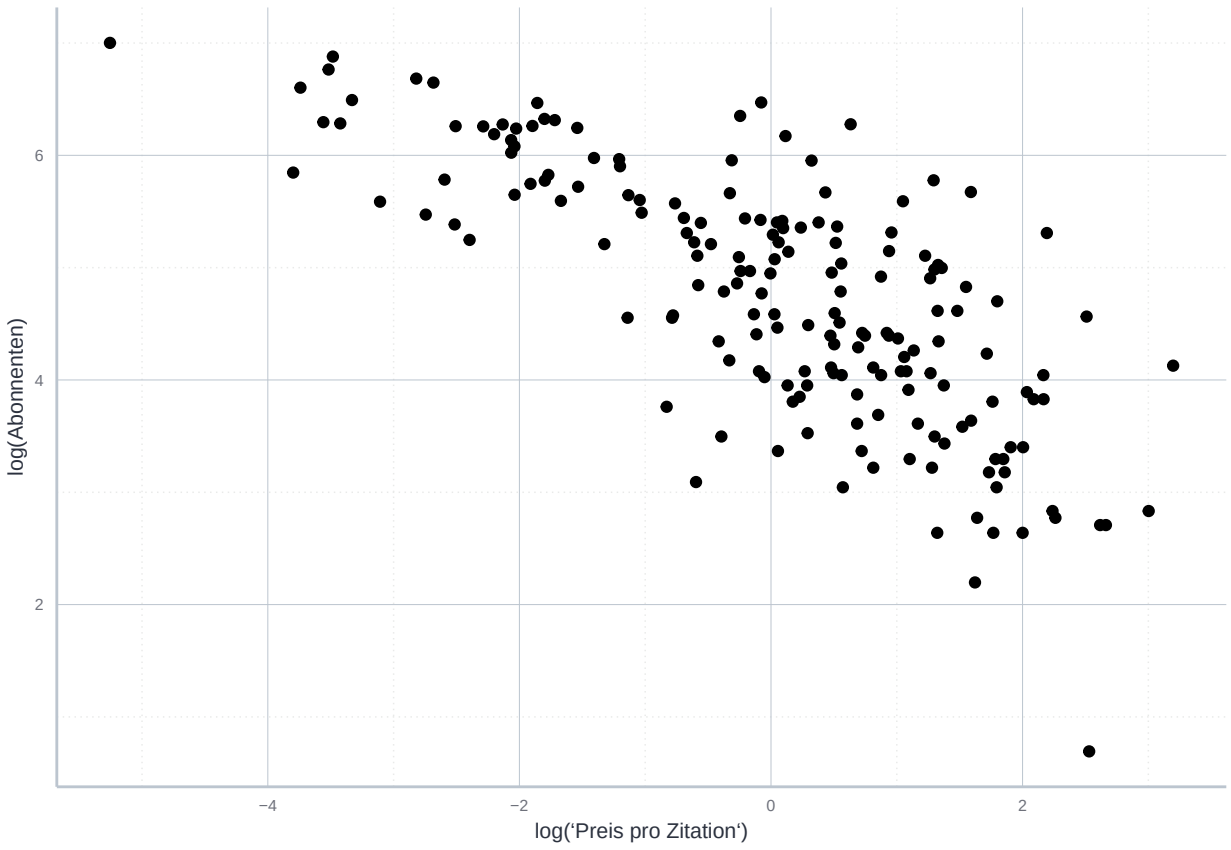
Für die folgende Spezifikation wäre der OLS-Schätzer also weder konsistent noch erwartungstreu, da hier ein klarer Verstoß gegen A1 vorliegen würde. Die folgende Schätzung ist entsprechend nicht zu gebrauchen:

$$\text{Abonnenten} = \beta_0 + \beta_1 \text{Zitationspreis} + \epsilon$$

```
lin_mod <- lm(Abonnenten~`Preis pro Zitation`, data=journal_daten)
```

Wenn wir aber beide Größen logarithmieren würden wäre der Zusammenhang schon ziemlich linear:

```
ggplot(data = journal_daten,
       aes(x=log(UQ(as.name("Preis pro Zitation"))),
           y=log(Abonnenten))) +
  geom_point() + theme_icae()
```



Die folgende Gleichung wäre also nicht unbedingt mit einem Verstoß gegen A1 verbunden:

$$\ln(\text{Abonnenten}) = \beta_0 + \beta_1 \ln(\text{Zitationspreis}) + \epsilon$$

```
log_mod <- lm(log(Abonnenten)~log(`Preis pro Zitation`), data=journal_daten)
```

Wir verwenden die Funktion `resettest()` um diese Intuition zu überprüfen. Zunächst testen wir auf eine Misspezifikation im linearen Modell, indem wir der Funktion `resettest()` das Schätzobjekt übergeben:

```
resettest(lin_mod)
```

```
#>
#>  RESET test
#>
#> data:  lin_mod
#> RESET = 28.99, df1 = 2, df2 = 176, p-value = 1.31e-11
```

Wenig überraschend müssen wir die H_0 des korrekt spezifizierten Modells klar ablehnen. Wie sieht es mit dem Log-Lin Modell aus?

```
resettest(log_mod)
```

```
#>
#>  RESET test
#>
#> data:  log_mod
```

```
#> RESET = 1.4409, df1 = 2, df2 = 176, p-value = 0.2395
```

Hier kann H_0 nicht abgelehnt werden.

Beachten Sie aber, dass der RESET Test keine abschließende Sicherheit bieten kann. Sie werden immer wieder Situationen erlegebn in denen der RESET Test ein Modell ablehnt, das sie aufgrund empirischer und theoretischer Überlegungen gut verteidigen könnten und umgekehrt. Daher sollte er immer mit Theorie und Beobachtung kombiniert werden.

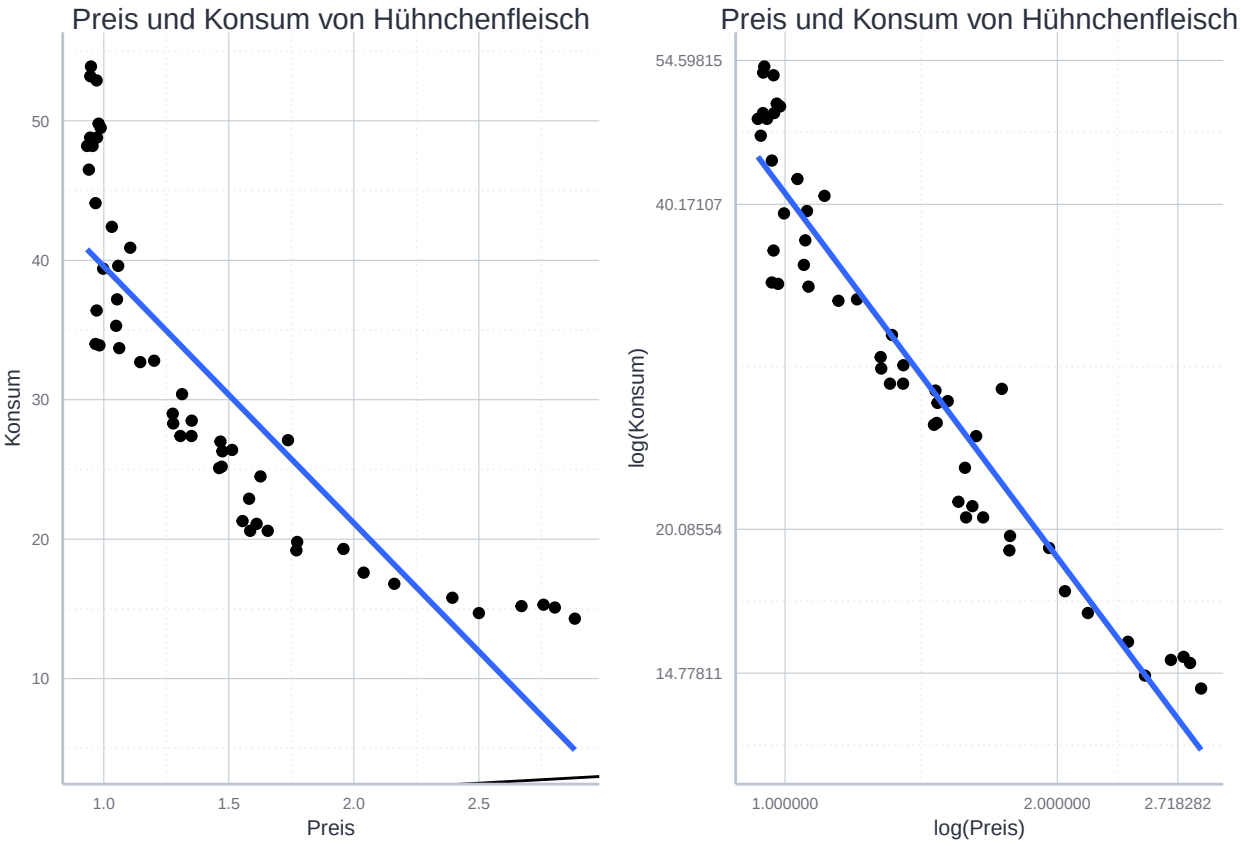
1.6.3 Wahl der funktionalen Form

Die Wahl der funktionalen Form hat nicht nur das Ziel Annahme 1 zu erfüllen. Da auch die Interpretation der geschätzten Koeffizienten je nach funktionaler Form eine andere ist, kann die Wahl einer bestimmten funktionalen Form auch theoretisch motiviert sein. Gerade die so genannten 'log-log-Modelle' sind häufig auch theoretisch sehr interessant, da wir hier Elastizitäten direkt schätzen können. Die folgende Tabelle gibt einen Überblick über häufig gewählte Spezifikationen und ihre Interpretation für das einfache lineare Regressionsmodell. Für das Modell mit mehreren unabhängigen Variablen ist die Interpretation äquivalent:

Modellart	**Schätzgleichung**	**Interpretation der Koeffizienten**
Level-Level	$y = \beta_0 + \beta_1 x_1 + \epsilon$	Ändert sich x_1 um 1 ändert sich y um β_1
Log-Level	$\ln(y) = \beta_0 + \beta_1 x_1 + \epsilon$	Ändert sich x_1 um 1 ändert sich y c.p. um ca. $\beta_1 \cdot 100\%$
Level-Log	$y = \beta_0 + \beta_1 \ln(x_1) + \epsilon$	Ändert sich x_1 um ca. 1% ändert sich y c.p. um ca. $\beta_1/100$
Log-Log	$\ln(y) = \beta_0 + \beta_1 \ln(x_1) + \epsilon$	Ändert sich x_1 um ca. 1% ändert sich y c.p. um ca. $\beta_1 \cdot 100\%$

Illustrieren wir die Wahl der funktionalen Form an folgendem Beispiel. Die Daten kommen von Epple and McCallum (2006) und enthalten Information zum Preis und zum Konsum von Hähnchenfleisch.

Wie wir sehen werden ist dieser Zusammenhang an sich nicht linear, kann aber durch Logarithmieren in eine lineare Form gebracht werden:



Die folgende Gleichung ist also konsistent mit A1 und kann entsprechend mit OLS geschätzt werden:

$$\ln(q) = \beta_0 + \beta_1 \ln(p) + \epsilon$$

```
log_model <- lm(log(q)~log(p), data = chicken_daten)
```

Diese Form ist dann linear und konsistent mit A1. Entsprechend macht es Sinn den Output zu interpretieren.

```
summary(log_model)
```

```
#>
#> Call:
#> lm(formula = log(q) ~ log(p), data = chicken_daten)
#>
#> Residuals:
#>       Min        1Q    Median        3Q       Max
#> -0.228363 -0.080077 -0.007662  0.106041  0.218679
#>
#> Coefficients:
#>             Estimate Std. Error t value Pr(>|t|)
#> (Intercept)  3.71694    0.02236   166.2   <2e-16 ***
#> log(p)      -1.12136    0.04876   -23.0   <2e-16 ***
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
```

```
#> Residual standard error: 0.118 on 50 degrees of freedom
#> Multiple R-squared:  0.9136, Adjusted R-squared:  0.9119
#> F-statistic:   529 on 1 and 50 DF,  p-value: < 2.2e-16
```

Wir würden den geschätzten Koeffizienten von β_1 folgendermaßen interpretieren: wenn der Preis von Hühnerfleisch um 1% steigt wird der Konsum um ca. 1.12% zurückgehen.

1.7 Anhang: Übersicht über die Testverfahren

Problem	**Mögliche Tests**	**Implikationen**	**Reaktion**
Heteroskedastie	Tukey-Anscombe Plot, Breusch-Pagan (`bptest()`), Goldfeld-Quandt (`gqtest`)	Reduzierte Effizienz, falsche Standardfehler	Robuste Standardfehler
Autokorrelation	Tukey-Anscombe Plot, Box–Pierce/Ljung–Box (`Box.test`), Durbin-Watson (`dwtest`), Breusch-Godfrey (`bgtest()`)	Reduzierte Effizienz, falsche Standardfehler	Robuste Standardfehler
Multikollinearität	Hilfsregressionen	Größere Standardfehler	Ggf. alternative unabh. Variablen verwenden
Falsche funktionale Form	Theorie, RESET-Test, Tukey-Anscombe Plot	Verzerrter und ineffizienter Schätzer	Funktionale Form anpassen
Vergessene Variablen	Theorie, Tukey-Anscombe Plot	Verzerrter und ineffizienter Schätzer	Variablen ergänzen

Chapter 2

Ausgewählte nichtlineare Schätzverfahren

Eine der zentralsten und gleichzeitig restriktivsten Annahmen des OLS Modells ist die Annahme eines linearen Zusammenhangs zwischen der abhängigen und den unabhängigen Variablen. Auch wenn wir im letzten Kapitel gesehen haben wie wir manche nicht-lineare Zusammenhänge durch angemessene Datentransformationen und der Verwendung cleverer funktionaler Formen mit OLS konsistent schätzen können bleiben zahlreiche interessante Zusammenhänge außen vor.

In diesem Kapitel werden wir uns beispielhaft mit dem Fall beschäftigen, in dem unsere abhängige Variable binär ist. Ein typisches Beispiel ist die Analyse von Arbeitslosigkeit. Stellen wir uns vor wir möchten untersuchen unter welchen Umständen Menschen arbeitslos werden. Unsere abhängige Variable $\boldsymbol{y}$ ist dabei eine binäre Varianble, die entweder den Wert 0 annimmt wenn eine Person nicht arbeitslos ist oder den Wert 1 annimmt wenn eine Person arbeitslos ist. Unsere Matrix $\boldsymbol{X}$ enthält dann Informationen über Variablen, die die Arbeitslosigkeit beeinfluss könnten, z.B. Ausbildungsniveau oder Alter. Wir möchten untersuchen wie Variation in den erklärenden Variablen die Wahrscheinlichkeit bestimmt, dass jemand arbeitslos ist, also $\mathbb{P}(\boldsymbol{y} = \mathbf{1}|\boldsymbol{X})$.

Dieser Zusammenhang kann unmöglich als linear aufgefasst werden: es ist unmöglich, dass $y < 0$ oder $y > 1$ und der Zusammenhang im Intervall $[0, 1]$ ist quasi nie linear. Daher ist der herkömmliche OLS Schätzer für solche Fälle ungeeignet, denn A1 ist klar verletzt. In diesem Kapitel lernen wir dabei logit- und probit-Modelle als alternative Schätzverfahren kennen.

Dabei werden die folgenden Pakete verwendet:

```
library(tidyverse)
library(data.table)
library(here)
library(icaeDesign)
```

2.1 Binäre abhängige Variablen: Logit- und Probit-Modelle

Das folgende Beispiel verwendet angepasste Daten aus Kleiber and Zeileis (2008) zur Beschäftigunssituation von Frauen aus der Schweiz:

```
schweiz_al <- fread(here("data/tidy/nonlinmodels_schweizer-arbeit.csv"),
                    colClasses = c("double", rep("double", 5), "factor"))
head(schweiz_al)
```

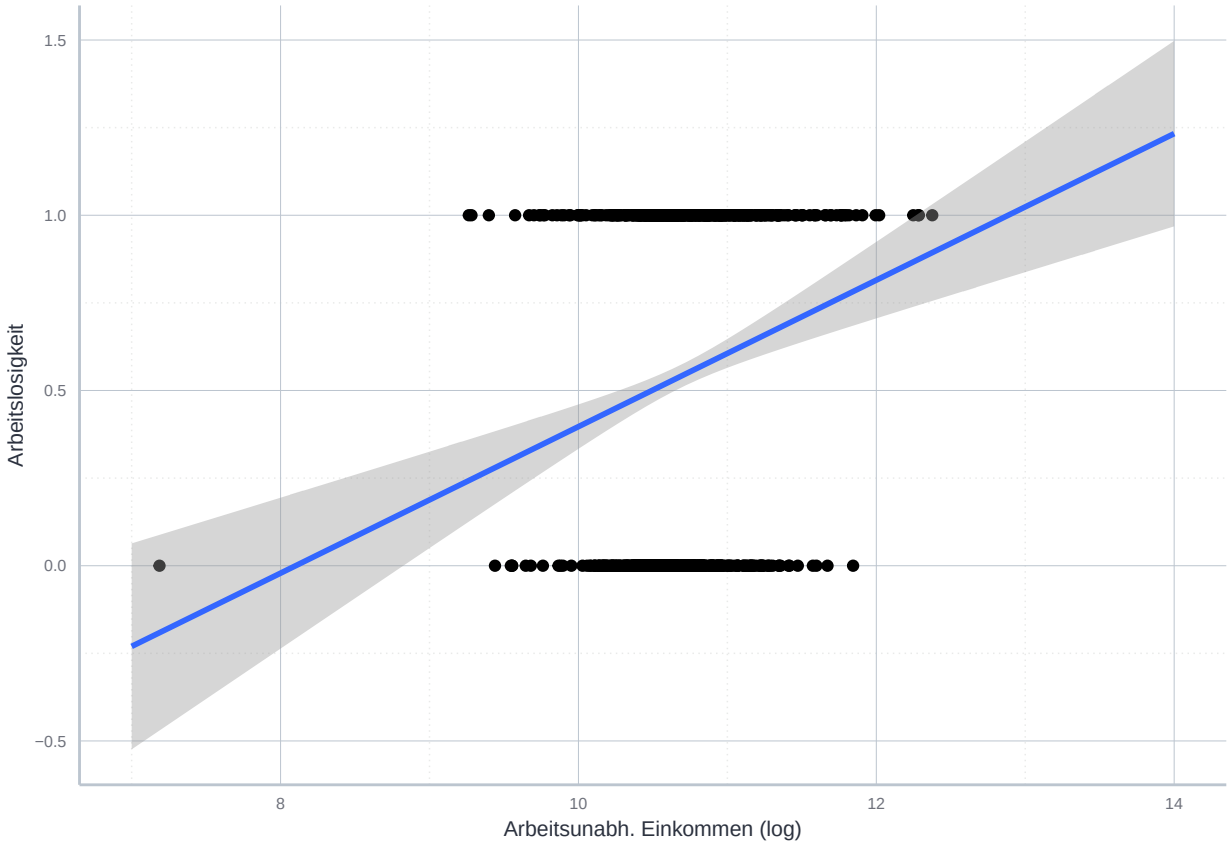
```
##    Arbeitslos Einkommen_log Alter Ausbildung_Jahre Kinder_jung Kinder_alt
## 1:          1      10.78750    30                8           1          1
## 2:          0      10.52425    45                8           0          1
## 3:          1      10.96858    46                9           0          0
## 4:          1      11.10500    31               11           2          0
## 5:          1      11.10847    44               12           0          2
## 6:          0      11.02825    42               12           0          1
##    Auslaender
## 1:          0
## 2:          0
## 3:          0
## 4:          0
## 5:          0
## 6:          0
```

Wir sind interessiert welchen Einfluss die erklärenden Variablen auf die Wahrscheinlichkeit haben, dass eine Frau Arbeitslos ist, also die Variable `Arbeitslos` den Wert `1` annimmt.

2.1.1 Warum nicht OLS?

Wir könnten natürlich zunächst einmal unser bekanntes und geliebtes OLS Modell verwenden um den Zusammenhang zu schätzen. Um die Probleme zu illustrieren schätzen wir einmal nur den bivariaten Zusammenhang zwischen `Arbeitslos` und `Einkommen_log`:

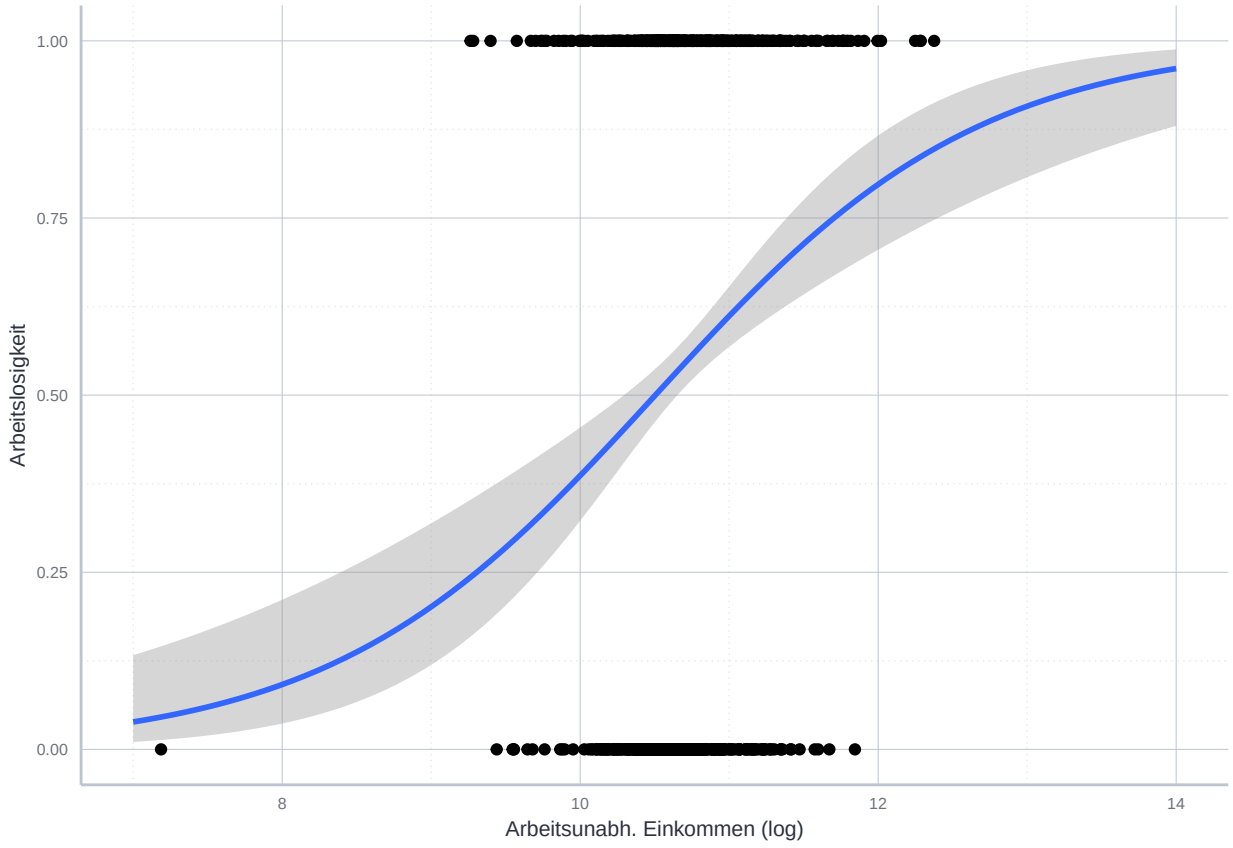
```
ggplot(
  data = schweiz_al,
  mapping = aes(x=Einkommen_log, y=Arbeitslos, group=1)) +
  scale_x_continuous(limits = c(7, 14)) +
  ylab("Arbeitslosigkeit") + xlab("Arbeitsunabh. Einkommen (log)") +
  geom_point() + geom_smooth(method = "lm", fullrange=TRUE) + theme_icae()
```



Unser Modell würde für bestimmte Levels an arbeitsunabhängigem Einkommen Werte außerhalb des Intervalls $0, 1$ vorhersagen - also Werte, die y gar nicht annehmen kann und die, da wir die Werte für y später als Wahrscheinlichkeiten interpretieren wollen, auch gar keinen Sinn ergeben wollen.

Unser Ziel ist da eher ein funktionaler Zusammenhang wie in folgender Abbildung zu sehen:

```
ggplot(
  data = schweiz_al,
  mapping = aes(x=Einkommen_log, y=Arbeitslos, group=1)) +
  scale_x_continuous(limits = c(7, 14)) +
  ylab("Arbeitslosigkeit") + xlab("Arbeitsunabh. Einkommen (log)") +
  geom_point() + geom_smooth(aes(y=Arbeitslos), method = "glm",
                             method.args = list(family = "binomial"),
                             fullrange=TRUE, se = TRUE) + theme_icae()
```



Dieser Zusammenhang ist jedoch nicht linear und damit inkonsisten mit A1 des OLS Modells.

2.1.2 Logit und Probit: theoretische Grundidee

Wir sind interessiert an $\mathbb{P}(y = 1|\boldsymbol{x})$, also der Wahrscheinlichkeit, dass y den Wert 1 annimmt, gegeben die unabhängigen Variablen $\boldsymbol{x}$.

Eine Möglichkeit $\mathbb{P}(y = 1|\boldsymbol{x})$ auf das Intervall $[0, 1]$ zu beschränken ist folgende Transformation:

$$\mathbb{P}(y = 1|\boldsymbol{x}) = \frac{\exp(\boldsymbol{X\beta})}{1 + \exp(\boldsymbol{X\beta})}$$

Diesen Ausdruck können wir dann folgendermaßen umformen:

$$\frac{\mathbb{P}(y = 1|\boldsymbol{x})}{1 - \mathbb{P}(y = 1|\boldsymbol{x})} = \frac{\frac{\exp(\boldsymbol{X\beta})}{1+\exp(\boldsymbol{X\beta})}}{1 - \frac{\exp(\boldsymbol{X\beta})}{1+\exp(\boldsymbol{X\beta})}}$$

Hier haben wir nun die so genannten *odds*: das Verhältnist dass $\mathbb{P}(y = 1|\boldsymbol{x})$ und $\mathbb{P}(y \neq 0|\boldsymbol{x})$. Wir multiplizieren nun den linken Teil der Gleichung mit $1 = \frac{\exp(\boldsymbol{X\beta})}{\exp(\boldsymbol{X\beta})}$ um den Zähler durch Kürzen zu vereinfachen:

$$\begin{aligned}\frac{\mathbb{P}(y=1|\boldsymbol{x})}{1-\mathbb{P}(y=1|\boldsymbol{x})} &= \frac{\exp(\boldsymbol{X\beta})}{\exp(\boldsymbol{X\beta})\cdot\left(\frac{1+\exp(\boldsymbol{X\beta}}{1+\exp(\boldsymbol{X\beta}} - \frac{\exp(\boldsymbol{X\beta})}{1+\exp(\boldsymbol{X\beta})}\right)} \\ &= \frac{\exp(\boldsymbol{X\beta})}{(1+\exp(\boldsymbol{X\beta}))\cdot\frac{1}{1+\exp(\boldsymbol{X\beta})}} \\ &= \exp(\boldsymbol{X\beta}) \end{aligned} \tag{2.1}$$

Nun können wir durch logarithmieren eine brauchbare Schätzgleichung herleiten:

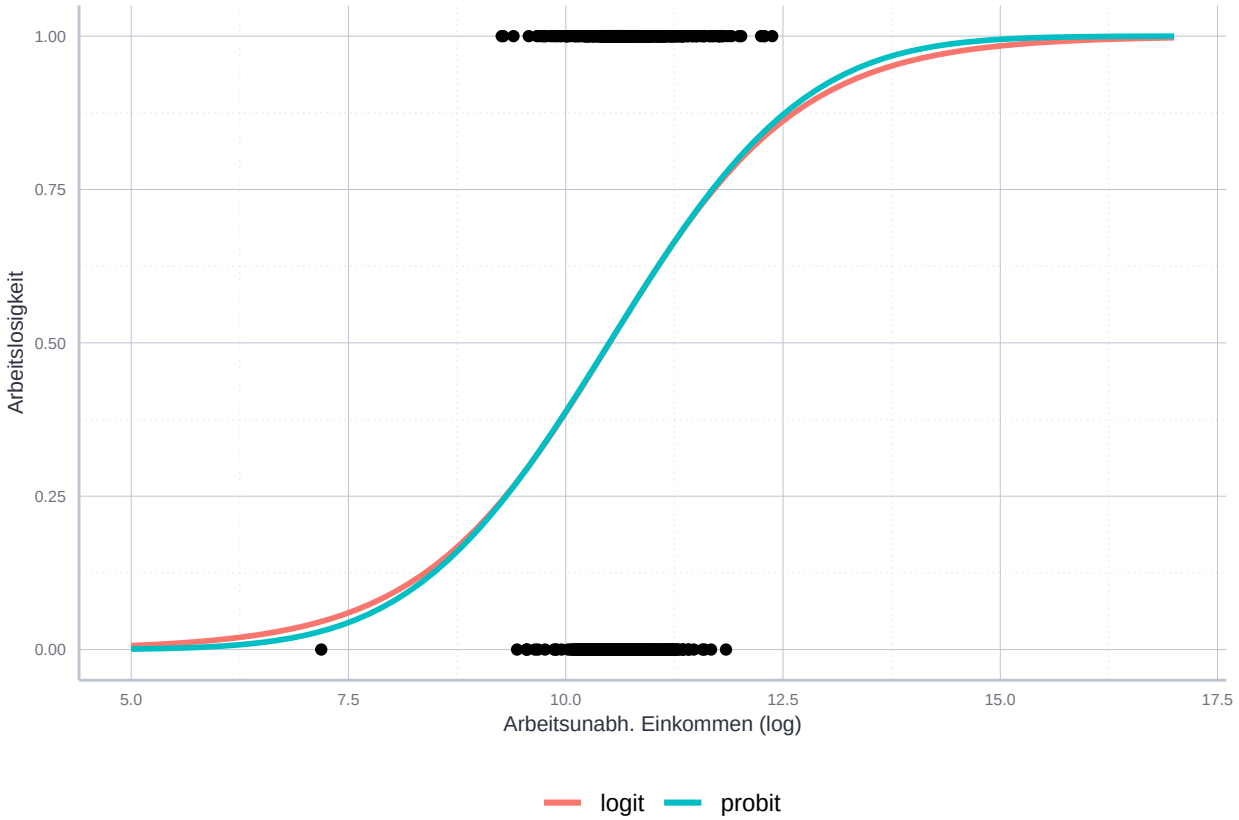
$$\begin{aligned}\ln\left(\frac{\mathbb{P}(y=1|\boldsymbol{x})}{1-\mathbb{P}(y=1|\boldsymbol{x})}\right) &= \ln\left(\exp(\boldsymbol{X\beta})\right) \\ \ln\left(\frac{\mathbb{P}(y=1|\boldsymbol{x})}{1-\mathbb{P}(y=1|\boldsymbol{x})}\right) &= \boldsymbol{X\beta}\end{aligned} \tag{2.2}$$

Wir sprechen hier von dem so genannten *logit* Modell, da wir hier auf der linken Seite den *Logarithmus* der *Odds* haben. Diesen Zusammenhang können wir nun auch ohne Probleme mit unserem OLS-Schätzer schätzen, denn hier haben wir einen klaren linearen Zusammenhang. Nur die anhängige Variable ist auf den ersten Blick ein wenig merkwürdig: der logarithmus der *Odds* des interessierenden Events. Aber das ist kein unlösbares Problem wie wir später sehen werden.

probit Modelle funktionieren auf eine sehr ähnliche Art und Weise, verwenden aber eine andere Transformation über die kumulierte Wahrscheinlichkeitsverteilung der Normalverteilung. Hier wird im Endeffekt folgende Regressionsgleichung geschätzt:

$$\mathbb{P}(y=1|\boldsymbol{x}) = \phi(\boldsymbol{X\beta})$$

wobei $\Phi(\cdot)$ die CDF der Normalverteilung ist. Wie sie in folgender Abbildung sehen, die sich wieder auf das Einführungsbeispiel bezieht, sind die funktionalen Formen bei der beiden Modelle sehr ähnlich:



Wir werden der Einfachheit halber im folgenden in der Regel das *logit* Modell verwenden, aber die Implementierung in R ist wirklich sehr ähnlich.

2.1.3 Logit und Probit: Implementierung in R

Da *logit* und *probit* Modell zu den so genannten *generalisierten Modellen* gehören verwenden wir die Funktion `glm` um die Modelle zu schätzen. Die Spezifikation ist dabei sehr ähnlich zu den linearen Modellen, die wir mit `lm()` geschätzt haben.

Nehmen wir einmal an wir wollen mit unserem Datensatz von Schweizerinnen die Effekt von Alter und arbeitsunabhänigem Einkommen auf die Wahrscheinlichkeit der Arbeitslosigkeit schätzen.

Als erstes Argument `formula` übergeben wir wieder die Schätzgleichung. In unserem Falle wäre das also `Arbeitslos ~ Alter + Einkommen_log`.

Als zweites Argument (`family`) müssen wir die Schätzart spezifizieren. Für *logit* Modelle schreiben wir `family = binomial(link = "logit")`, für *probit* Modelle entsprechend `family = binomial(link = "probit")`.

Das letzte Argument ist dann `data`. Insgesamt erhalten wir also für das *logit*-Modell:

```
arbeitslogit_test <- glm(
  Arbeitslos ~ Einkommen_log + Alter,
  family = binomial(link = "logit"),
  data = schweiz_al)
```

Und das *probit*-Modell:

```
arbeitsprobit_test <- glm(
  Arbeitslos ~ Einkommen_log + Alter,
  family = binomial(link = "probit"),
  data = schweiz_al)
```

Für die Schätzergebnisse können wir wie bilang die Funktion `summary()` verwenden:

```
summary(arbeitslogit_test)
```

```
##
## Call:
## glm(formula = Arbeitslos ~ Einkommen_log + Alter, family = binomial(link = "logit"),
##     data = schweiz_al)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.7448  -1.1855   0.8128   1.1017   1.8279
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -10.381739   2.003223  -5.183 2.19e-07 ***
## Einkommen_log   0.920045   0.185414   4.962 6.97e-07 ***
## Alter           0.018013   0.006612   2.724  0.00645 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1203.2  on 871  degrees of freedom
## Residual deviance: 1168.5  on 869  degrees of freedom
## AIC: 1174.5
##
## Number of Fisher Scoring iterations: 4
```

Aber wie sollen wir das interpretieren? Da das ein wenig schwieriger ist beschäftigen wir uns damit im nächsten Abschnitt.

2.1.4 Logit und Probit: Interpretation der Ergebnisse

Wie wir oben gesehen haben ist die abhängige Variable in der Logit-Regression der Logarithmus der *Odds Ratio*. Das ist nicht ganz einfach zu interpretieren. So bedeutet der Koeffizient für `Auslaender1` in folgender Ergebnistabelle, dass sich die logarithmierte *Odds Ratio ceteris paribus* um 1.3 Prozent reduziert, wenn die betroffene Person Ausländerin ist:

```
arbeitslogit <- glm(
  Arbeitslos ~ Einkommen_log + Alter + Ausbildung_Jahre + Kinder_jung +
    Kinder_alt + Auslaender,
```

```
  family = binomial(link = "logit"),
  data = schweiz_al)
summary(arbeitslogit)
```

```
##
## Call:
## glm(formula = Arbeitslos ~ Einkommen_log + Alter + Ausbildung_Jahre +
##     Kinder_jung + Kinder_alt + Auslaender, family = binomial(link = "logit"),
##     data = schweiz_al)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2681  -1.0675   0.5383   0.9727   1.9384
##
## Coefficients:
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -10.374346   2.166852  -4.788 1.69e-06 ***
## Einkommen_log      0.815041   0.205501   3.966 7.31e-05 ***
## Alter              0.051033   0.009052   5.638 1.72e-08 ***
## Ausbildung_Jahre  -0.031728   0.029036  -1.093    0.275
## Kinder_jung        1.330724   0.180170   7.386 1.51e-13 ***
## Kinder_alt         0.021986   0.073766   0.298    0.766
## Auslaender1       -1.310405   0.199758  -6.560 5.38e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1203.2  on 871  degrees of freedom
## Residual deviance: 1052.8  on 865  degrees of freedom
## AIC: 1066.8
##
## Number of Fisher Scoring iterations: 4
```

Es wäre ja deutlich schöner wenn wir Änderungen in den unabhängigen Variablen als Änderungen in $\mathbb{P}(y = 1|\boldsymbol{x})$ interpretieren könnten. In unserem Beispiel also: um wie viel Prozent würde die Wahrscheinlichkeit für Arbeitslosigkeit steigen wenn es sich bei der betroffenen Person um eine Ausländerin handelt? Um dieses Ergebnis zu bekommen bedarf es aber einiger weniger Umformungen.

Da der Zusammenhang zwischen $\mathbb{P}(y = 1|\boldsymbol{x})$ und den unabhängigen Variablen nicht-linear ist müssen wir für die Vergleiche der Wahrscheinlichkeiten konkrete Werte angeben.

In einem ersten Schritt verwenden wir die Funktion `predict`, der wir als erstes Argument `object` unser geschätztes Modell übergeben. Als zweites Argument übergeben wir einen `data.frame`, in dem wir die relevanten Änderungen und den zu betrachtenden Bereich angeben. Je nach Anzahl der abhängigen Variablen kann diese Tabelle recht groß werden, sie ist aber notwendig, da der Zusammenhang zwischen abhängigen und unabhängiger Variable ja nicht-linear ist.

Als drittes Argument müssen wir noch `type = "response"` übergeben damit wir die Vorhersagen auf der Skala der zugrundeliegenden abhänigigen Variable bekommen, also direkt als Wahrscheinlichkeiten:

```
predicted_probs <- predict(object = arbeitslogit,
        newdata = data.frame(
          "Einkommen_log" = c(10, 10),
          "Alter"=c(30, 30),
          "Ausbildung_Jahre" = c(5, 5),
          "Kinder_alt" = c(0, 0),
          "Kinder_jung"= c(1, 2),
          "Auslaender" = factor(c(0, 0))
          ),
        type = "response")
predicted_probs
```

```
##         1         2
## 0.6175431 0.8593445
```

Das erste Element ist die Wahrscheinlichkeit arbeitslos zu sein für eine dreißigjährige Frau mit einem arbeitsunabhänigen Einkommen von $\exp(10) = 22025$, fünfjähiger Ausbildung, keinen alten Kindern, einem jungen Kind und mit schweizerischer Staatsangehörigkeit. Die zweite Wahrscheinlichkeit gilt für eine Frau mit den gleichen Eigenschaften aber zwei jungen Kindern. Mit `diff()` bekommen wir gleich den entsprechenden Effekt eines weiteren jungen Kindes auf die Wahrscheinlichkeit arbeitslos zu sein:

```
diff(predicted_probs)
```

```
##         2
## 0.2418014
```

Die Wahrscheinlichkeit ist also nach dem Modell ca. 25% größer! Wenn wir wissen wollen ob der Effekt für Ausländerinnen ähnlich ist rechnen wir:

```
diff(
  predict(object = arbeitslogit,
        newdata = data.frame(
          "Einkommen_log" = c(10, 10),
          "Alter"=c(30, 30),
          "Ausbildung_Jahre" = c(5, 5),
          "Kinder_alt" = c(0, 0),
          "Kinder_jung"= c(1, 2),
          "Auslaender" = factor(c(1, 1))
          ),
        type = "response")
)
```

```
##         2
## 0.3189543
```

Hier ist der Effekt mit ca. 32% also noch größer!

Bibliography

Epple, D. and McCallum, B. T. (2006). Simultaneous equation econometrics: The missing example. *Economic Inquiry*, 44(2).

Greene, W. H. (2018). *Econometric Analysis*. Pearson, New York, NY. ISBN: 978-0-13-446136-6.

Kleiber, C. and Zeileis, A. (2008). *Applied Econometrics with R*. Springer-Verlag, New York. ISBN 978-0-387-77316-2.

R Core Team (2018). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Venables, W. N. and Ripley, B. D. (2002). *Modern Applied Statistics with S*. Springer, New York, fourth edition. ISBN 0-387-95457-0.

Zeileis, A. (2004). Econometric computing with HC and HAC covariance matrix estimators. *Journal of Statistical Software*, 11(10):1–17.

Zeileis, A. and Hothorn, T. (2002). Diagnostic checking in regression relationships. *R News*, 2(3):7–10.