

# Skript zur Vorlesung 5: Das lineare Regressionsmodell

Claudius Gräbner

KW 48 2020

## Contents

<b>1</b>	<b>Einleitung</b>	<b>1</b>
<b>2</b>	<b>Einführungsbeispiel zur Konsumfunktion</b>	<b>2</b>
2.1	Komsum und BIP . . . . .	2
2.2	Marketingeffekte . . . . .	3
<b>3</b>	<b>Illustration der OLS-Schätzung</b>	<b>4</b>
<b>4</b>	<b>Das Bestimmtheitsmaß <math>R^2</math></b>	<b>5</b>
<b>5</b>	<b>Simulation der Schätzerverteilung</b>	<b>12</b>
<b>6</b>	<b>Preise von Journalen</b>	<b>13</b>
<b>7</b>	<b>Vergleich der multiplen und einfachen linearen Regression</b>	<b>14</b>

## 1 Einleitung

In diesem Dokument werden alle Abbildungen und Tabellen aus der vierten Vorlesung repliziert. Dabei gebe ich der Info wegen *allen* R Code. Entsprechend sind bestimmt auch einige Befehle dabei, die Sie jetzt noch nicht kennen.

Im Code wird davon ausgegangen, dass ausgehend vom Arbeitsverzeichnis das File `data/tidy/journaldaten.csv` existiert, also im Arbeitsverzeichnis ein Ordner `data` mit einem Unterordner `T5` existiert, in dem dann die Datei `journaldaten.csv` liegt. Diese Datei kann separat aus dem Moodle heruntergeladen werden. Um die Abbildungen zu replizieren müssen Sie den Pfad ggf. an Ihre eigene Ordnerstruktur anpassen.

Folgende Pakete wurden verwendet:

```
library(tidyverse)
library(data.table)
library(ggpubr)
library(latex2exp)
library(icaeDesign)
library(here)
library(WDI)
library(datarium)
library(plot3D)
```

Beachten Sie, dass das Paket `icaeDesign` nicht über die zentrale Paketverwaltung verfügbar ist. Es muss folgendermaßen installiert werden:

```
devtools::install_github("graebner/icaeDesign")
```

## 2 Einführungsbeispiel zur Konsumfunktion

### 2.1 Konsum und BIP

Die Daten stammen von der Weltbank und wurden mit folgendem Code heruntergeladen:

```
bip_daten <- WDI(  
  country = "DE", start = 2000, end=2019,  
  indicator = c("NE.CON.TOTL.CN", "NY.GDP.MKTP.CN")) %>%  
  rename(Konsum=NE.CON.TOTL.CN,  
         BIP=NY.GDP.MKTP.CN) %>%  
  dplyr::mutate(Konsum=Konsum/1000000000000,  
               BIP=BIP/1000000000000)
```

```
fwrite(bip_daten, file = pfad_bip_daten)
```

```
schaetzung_bip <- lm(Konsum~BIP, data = bip_daten)  
schaetzung_bip
```

```
##  
## Call:  
## lm(formula = Konsum ~ BIP, data = bip_daten)  
##  
## Coefficients:  
## (Intercept)          BIP  
##      0.1902      0.6655  
  
beta_0 <- schaeztung_bip[["coefficients"]][1]  
beta_1 <- schaeztung_bip[["coefficients"]][2]  
unnane(beta_0 + beta_1*4)
```

```
## [1] 2.852296
```

Die Abbildungen wurden folgendermaßen erstellt:

```
bip_plot <- ggplot(  
  data = bip_daten, mapping = aes(x=BIP, y=Konsum)  
) +  
  geom_point() +  
  scale_x_continuous(  
    limits = c(1.95, 3.6),  
    expand = expansion(c(0, 0)),  
    breaks = seq(2.0, 4.0, by = 0.5)) +  
  labs(  
    title = "Konsumausgaben und BIP (2000-2019)",  
    caption = "Daten: Weltbank; Werte in Billionen Euro.") +  
  theme_icae() +  
  theme(  
    axis.title = element_text(size=13),  
    plot.title = element_text(size=14),  
    axis.text = element_text(size=12)  
  )  
  
ggsave(plot = bip_plot,
```

```
filename = here("figures/T5/01_C-BIP.pdf"),
width = 5, height = 4)
```

Hier die Abbildung mit Regressionsgrade:

```
bip_plot_reg <- bip_plot +
  geom_smooth(method = "lm", color="black", se = FALSE) +
  annotate(geom = "text", x = 2.6, y = 1.9,
    label=TeX("Steigung der Geraden:  $\hat{\beta}_1=0.665$ "),
    size=4, hjust=0)
ggsave(plot = bip_plot_reg,
  filename = here("figures/T5/01_C-BIP-reg.pdf"),
  width = 5, height = 4)
```

## 2.2 Marketingeffekte

Hier haben wir den Datensatz `marketing` aus dem Paket `datarium` verwendet. Er enthält Informationen über die Ausgaben für drei Werbeplattformen (Youtube, Facebook und Zeitungen) und entsprechende Umsätze:

```
library(datarium)
marketing_data <- datarium::marketing
head(marketing_data)
```

```
##  youtube facebook newspaper sales
## 1  276.12    45.36    83.04 26.52
## 2   53.40    47.16    54.12 12.48
## 3   20.64    55.08    83.16 11.16
## 4  181.80    49.56    70.20 22.20
## 5  216.96    12.96    70.08 15.48
## 6   10.44    58.68    90.00  8.64
```

Die Daten werden zunächst vorbereitet:

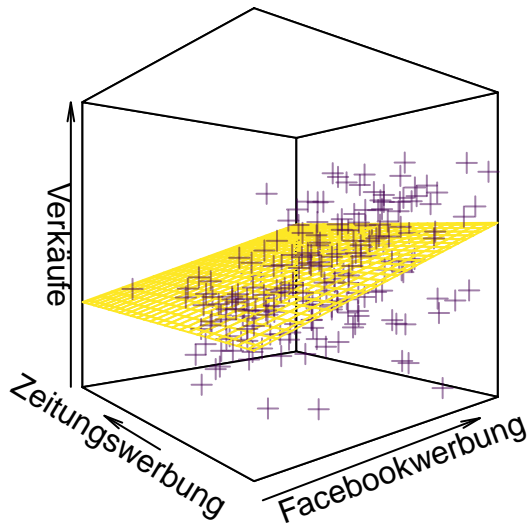
```
x = marketing_data$facebook
y = marketing_data$newspaper
z = marketing_data$sales
fit <- lm(z ~ x + y)
grid.lines = 25
x.pred = seq(min(x), max(x), length.out = grid.lines)
y.pred = seq(min(y), max(y), length.out = grid.lines)
xy = expand.grid(x = x.pred, y = y.pred)
z.pred = matrix(predict(fit, newdata = xy),
  nrow = grid.lines, ncol = grid.lines)
fitpoints = predict(fit)

cols <- viridis::viridis(2)
```

Da `ggplot2` keine 3d-Darstellungen kann wird das `base` Paket verwendet:

```
scatter3D(x, y, z,
  pch = 3, cex = 1,
  col = cols[1], alpha=0.5, lighting = TRUE,
  theta = -40, phi = 0, # viewing angles
  #ticktype = "detailed",
  xlab = "Facebookwerbung", ylab = "Zeitungswerbung", zlab = "Verkäufe",
  zlim = c(0, 40), clim = c(0, 40),
```

```
surf = list(x = x.pred, y = y.pred, z = z.pred,
            facets = NA, col=cols[2]))
```



```
## pdf
## 2
```

Ich habe bei der resultierenden Abbildung noch manuell den weißen Rand mit der UNIX-Funktion `pdfcrop` entfernt.

### 3 Illustration der OLS-Schätzung

Datensatz erstellen:

```
set.seed(123)
wahres_b0 <- 3
wahres_b1 <- 1.4

stichproben_n <- 5
x <- 1:stichproben_n * 0.1
fehler <- rnorm(stichproben_n, mean = 0, sd = 1)
y <- rep(NA, stichproben_n)

for (i in 1:stichproben_n){
  y[i] <- wahres_b0 + wahres_b1*x[i] + fehler[i]
}
datensatz <- data.frame(
  x = x,
  y = round(y, 2)
)
```

```
datensatz
```

```
##      x      y
## 1 0.1 2.58
## 2 0.2 3.05
## 3 0.3 4.98
## 4 0.4 3.63
## 5 0.5 3.83
```

Schätzung durchführen:

```
schaetzung <- lm(y~x, data = datensatz)
schaetzung
```

```
##
## Call:
## lm(formula = y ~ x, data = datensatz)
##
## Coefficients:
## (Intercept)          x
##          2.69          3.08
schaetzung[["coefficients"]]
```

```
## (Intercept)          x
##          2.69          3.08
```

## 4 Das Bestimmtheitsmaß $R^2$

$$TSS = \sum_{i=1}^n (Y_i - \bar{Y})^2$$

```
tss <- sum((datensatz$y - mean(datensatz$y))**2)
tss
```

```
## [1] 3.30012
```

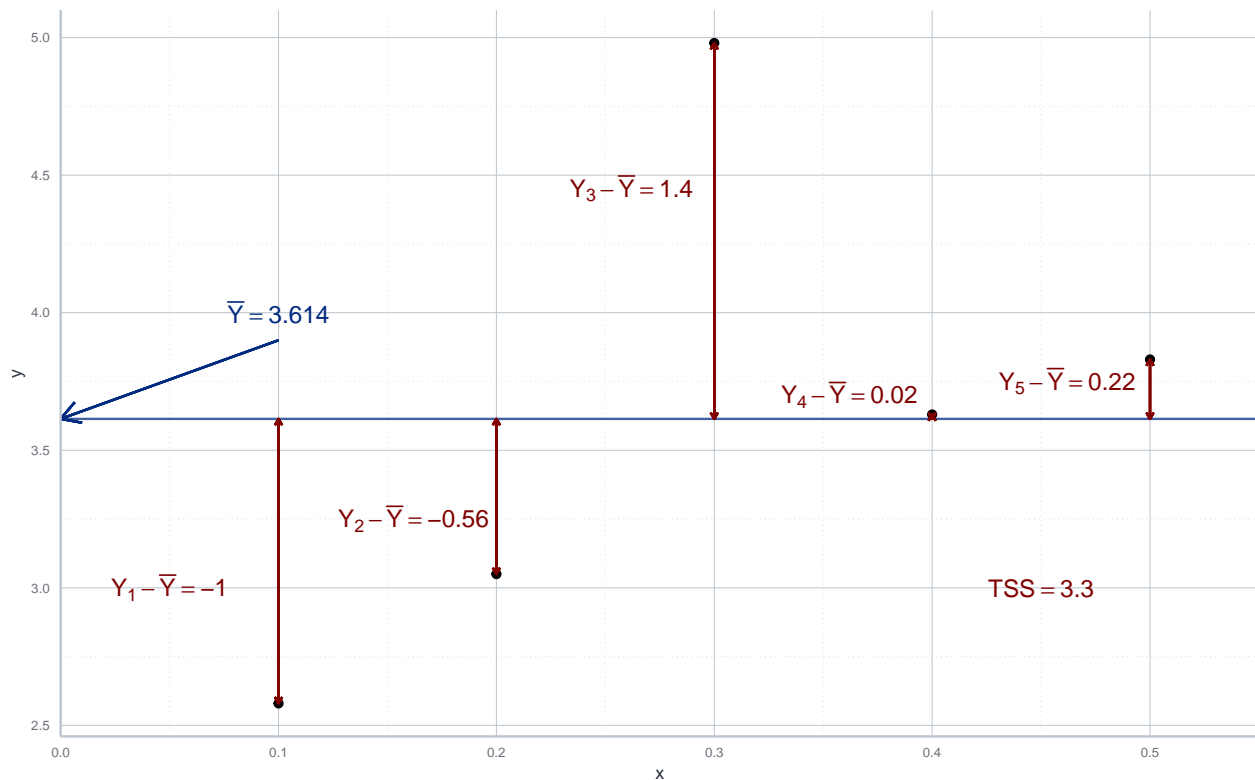
Grafische Darstellung:

```
ggplot(datensatz, aes(x=x, y=y)) +
  geom_point() +
  geom_abline(aes(color="Geschätzte Werte",
                  intercept = mean(datensatz$y),
                  slope = 0),
              alpha=0.75, show.legend = TRUE, color=get_icae_colors("dark blue")) +
  scale_x_continuous(limits = c(0, 0.55), expand = c(0, 0)) +
  geom_segment(aes(x = 0.1, y = 3.9,
                  xend = 0, yend=mean(datensatz$y)),
              arrow = arrow(length = unit(0.03, "npc")),
              colour = get_icae_colors("dark blue")) +
  geom_segment(aes(
    x = datensatz$x[1], y = datensatz$y[1],
    xend = datensatz$x[1], yend=mean(datensatz$y)
  ),
  arrow = arrow(length = unit(0.01, "npc"), ends = "both"),
  colour = get_icae_colors("dark red")
) +
  geom_segment(aes(
    x = datensatz$x[2], y = datensatz$y[2],
    xend = datensatz$x[2], yend=mean(datensatz$y)
  ),
  arrow = arrow(length = unit(0.01, "npc"), ends = "both"),
  colour = get_icae_colors("dark red")
) +
  geom_segment(aes(
```

```

x = datensatz$x[3], y = datensatz$y[3],
xend = datensatz$x[3], yend=mean(datensatz$y)
),
arrow = arrow(length = unit(0.01, "npc"), ends = "both"),
colour = get_icae_colors("dark red")
) +
geom_segment(aes(
x = datensatz$x[4], y = datensatz$y[4],
xend = datensatz$x[4], yend=mean(datensatz$y)
),
arrow = arrow(length = unit(0.01, "npc"), ends = "both"),
colour = get_icae_colors("dark red")
) +
geom_segment(aes(
x = datensatz$x[5], y = datensatz$y[5],
xend = datensatz$x[5], yend=mean(datensatz$y)
),
arrow = arrow(length = unit(0.01, "npc"), ends = "both"),
colour = get_icae_colors("dark red")
) +
annotate(geom = "text",
label=TeX("$\\bar{Y} = 3.614$"),
x = datensatz$x[1], y = 4,
colour = get_icae_colors("dark blue")) +
annotate(geom = "text",
label=TeX("$Y_1-\\bar{Y} = -1$"),
x = datensatz$x[1]-0.05, y = 3,
colour = get_icae_colors("dark red")) +
annotate(geom = "text",
label=TeX("$Y_2-\\bar{Y} = -0.56$"),
x = datensatz$x[2]-0.038, y = 3.25,
colour = get_icae_colors("dark red")) +
annotate(geom = "text",
label=TeX("$Y_3-\\bar{Y} = 1.4$"),
x = datensatz$x[3]-0.038, y = 4.45,
colour = get_icae_colors("dark red")) +
annotate(geom = "text",
label=TeX("$Y_4-\\bar{Y} = 0.02$"),
x = datensatz$x[4]-0.038, y = 3.7,
colour = get_icae_colors("dark red")) +
annotate(geom = "text",
label=TeX("$Y_5-\\bar{Y} = 0.22$"),
x = datensatz$x[5]-0.038, y = 3.75,
colour = get_icae_colors("dark red")) +
annotate(geom = "text",
label=TeX("$TSS = 3.3$"),
x = 0.45, y = 3.0,
colour = get_icae_colors("dark red")) +
theme_icae() +
theme(legend.position = "none")

```



$$RSS = \sum_i^n \epsilon_i^2$$

```
rss <- sum(schaetzung[["residuals"]]**2)
rss
```

```
## [1] 2.35148
```

Grafische Darstellung:

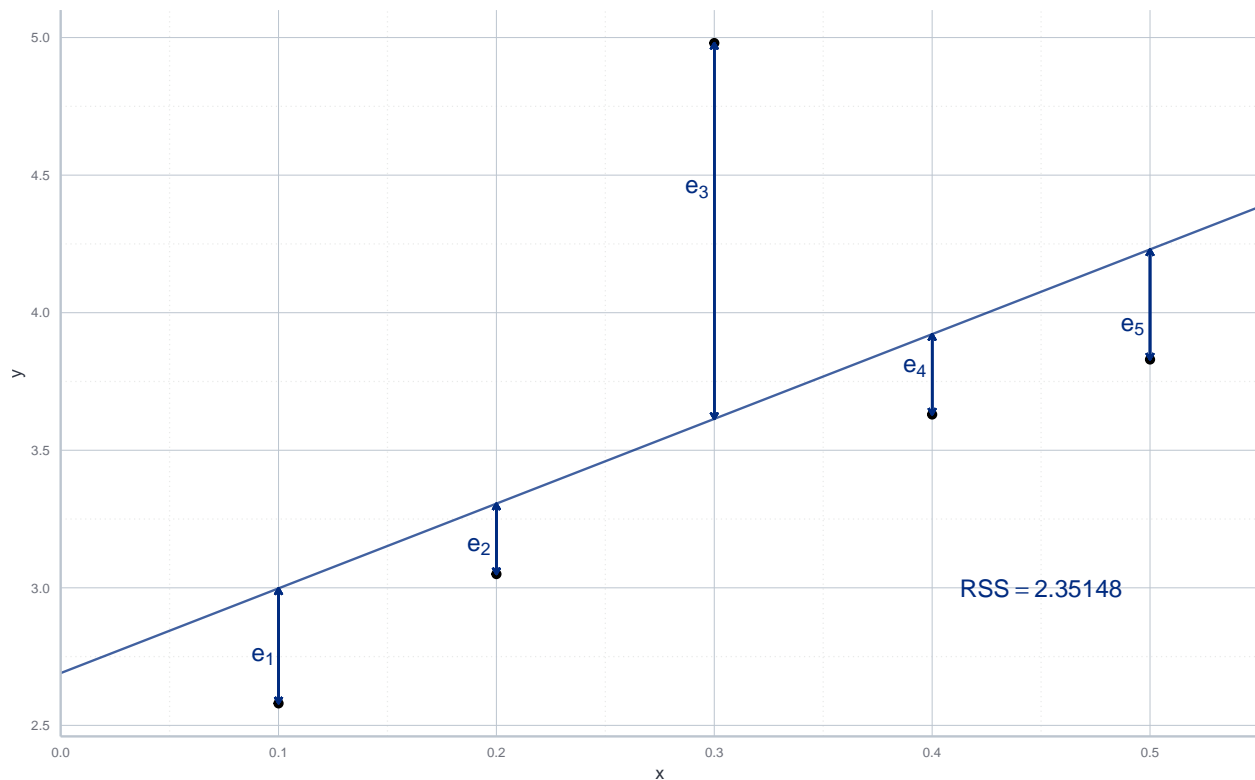
```
ggplot(datensatz, aes(x=x, y=y)) +
  geom_point() +
  geom_abline(aes(color="Geschätzte Werte",
    intercept = schaeztung[["coefficients"]][1],
    slope = schaeztung[["coefficients"]][2]),
    alpha=0.75, show.legend = TRUE, color=get_icae_colors("dark blue")) +
  scale_x_continuous(limits = c(0, 0.55), expand = c(0, 0)) +
  geom_segment(aes(
    x = datensatz$x[1], y = datensatz$y[1],
    xend = datensatz$x[1], yend=schaeztung$fitted.values[1]
  ),
  arrow = arrow(length = unit(0.01, "npc"), ends = "both"),
  colour = get_icae_colors("dark blue")
  ) +
  geom_segment(aes(
    x = datensatz$x[2], y = datensatz$y[2],
    xend = datensatz$x[2], yend=schaeztung$fitted.values[2]
  ),
  arrow = arrow(length = unit(0.01, "npc"), ends = "both"),
```

```

    colour = get_icae_colors("dark blue")
  ) +
  geom_segment(aes(
    x = datensatz$x[3], y = datensatz$y[3],
    xend = datensatz$x[3], yend=schaetzung$fitted.values[3]
  ),
  arrow = arrow(length = unit(0.01, "npc"), ends = "both"),
  colour = get_icae_colors("dark blue")
  ) +
  geom_segment(aes(
    x = datensatz$x[4], y = datensatz$y[4],
    xend = datensatz$x[4], yend=schaetzung$fitted.values[4]
  ),
  arrow = arrow(length = unit(0.01, "npc"), ends = "both"),
  colour = get_icae_colors("dark blue")
  ) +
  geom_segment(aes(
    x = datensatz$x[5], y = datensatz$y[5],
    xend = datensatz$x[5], yend=schaetzung$fitted.values[5]
  ),
  arrow = arrow(length = unit(0.01, "npc"), ends = "both"),
  colour = get_icae_colors("dark blue")
  ) +
  annotate(geom = "text",
    label=TeX("$e_1$"),
    x = datensatz$x[1]-0.007, y = 2.75,
    colour = get_icae_colors("dark blue")) +
  annotate(geom = "text",
    label=TeX("$e_2$"),
    x = datensatz$x[2]-0.008, y = 3.15,
    colour = get_icae_colors("dark blue")) +
  annotate(geom = "text",
    label=TeX("$e_3$"),
    x = datensatz$x[3]-0.008, y = 4.45,
    colour = get_icae_colors("dark blue")) +
  annotate(geom = "text",
    label=TeX("$e_4$"),
    x = datensatz$x[4]-0.008, y = 3.8,
    colour = get_icae_colors("dark blue")) +
  annotate(geom = "text",
    label=TeX("$e_5$"),
    x = datensatz$x[5]-0.008, y = 3.95,
    colour = get_icae_colors("dark blue")) +
  annotate(geom = "text",
    label=TeX("$RSS = 2.35148$"),
    x = 0.45, y = 3.0,
    colour = get_icae_colors("dark blue")) +
  theme_icae() +
  theme(legend.position = "none")

```





$$ESS = \sum_{i=1}^n (\hat{Y}_i - \bar{Y})^2$$

```
ess <- sum((schaetzung[["fitted.values"]] - mean(datensatz$y))**2)
ess
```

```
## [1] 0.94864
```

Und grafisch:

```
ggplot(datensatz, aes(x=x, y=y)) +
  geom_point() +
  geom_abline(aes(color="Geschätzte Werte",
    intercept = schaezung[["coefficients"]][1],
    slope = schaezung[["coefficients"]][2]),
    alpha=0.75, show.legend = TRUE, color=get_icae_colors("dark blue")) +
  geom_abline(aes(color="Geschätzte Werte",
    intercept = mean(datensatz$y),
    slope = 0),
    alpha=0.75, show.legend = TRUE, color=get_icae_colors("dark red")) +
  geom_segment(aes(x = 0.1, y = 3.9,
    xend = 0, yend=mean(datensatz$y)),
    arrow = arrow(length = unit(0.03, "npc")),
    colour = get_icae_colors("dark red")) +
  annotate(geom = "text",
    label=TeX("$\\bar{Y} = 3.614$"),
    x = datensatz$x[1], y = 4,
    colour = get_icae_colors("dark red")) +
  scale_x_continuous(limits = c(0, 0.55), expand = c(0, 0)) +
```

```

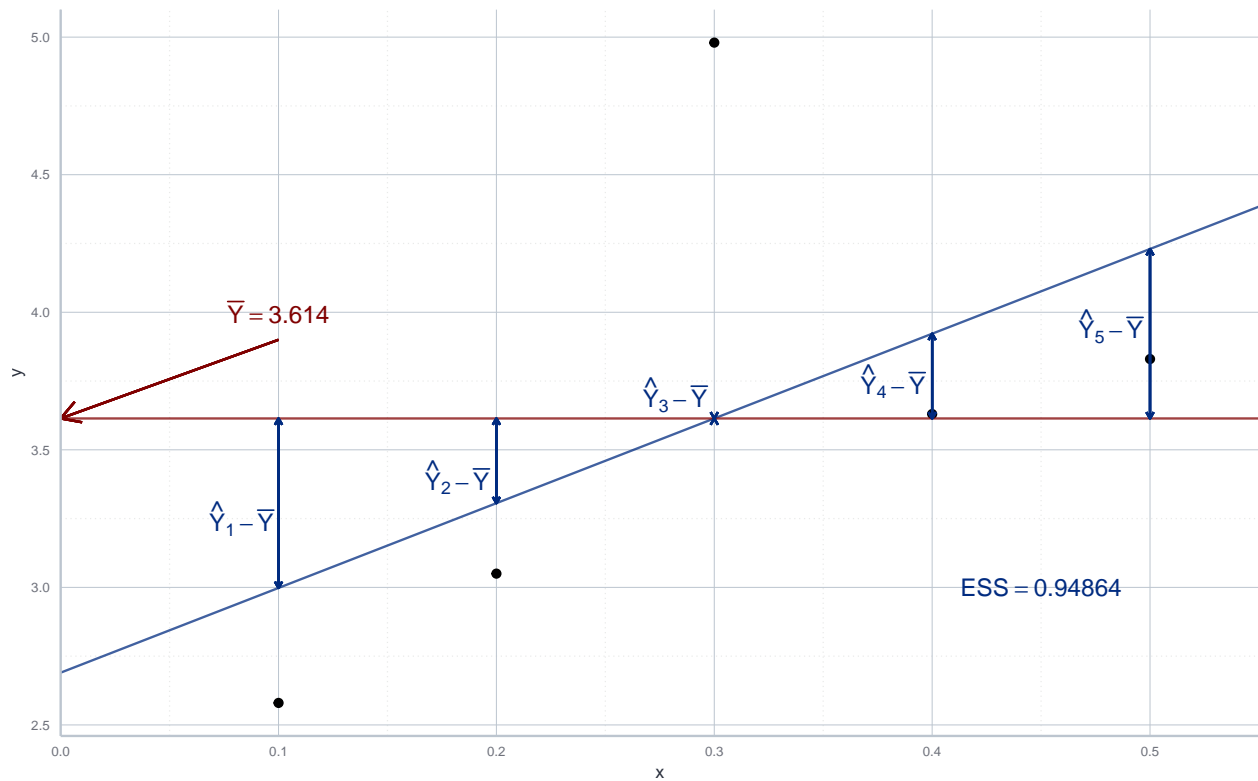
geom_segment(aes(
  x = datensatz$x[1], y = schaeztung$fitted.values[1],
  xend = datensatz$x[1], yend = mean(datensatz$y)
),
arrow = arrow(length = unit(0.01, "npc"), ends = "both"),
colour = get_icae_colors("dark blue")
) +
geom_segment(aes(
  x = datensatz$x[2], y = schaeztung$fitted.values[2],
  xend = datensatz$x[2], yend = mean(datensatz$y)
),
arrow = arrow(length = unit(0.01, "npc"), ends = "both"),
colour = get_icae_colors("dark blue")
) +
geom_segment(aes(
  x = datensatz$x[3], y = schaeztung$fitted.values[3],
  xend = datensatz$x[3], yend = mean(datensatz$y)
),
arrow = arrow(length = unit(0.01, "npc"), ends = "both"),
colour = get_icae_colors("dark blue")
) +
geom_segment(aes(
  x = datensatz$x[4], y = schaeztung$fitted.values[4],
  xend = datensatz$x[4], yend = mean(datensatz$y)
),
arrow = arrow(length = unit(0.01, "npc"), ends = "both"),
colour = get_icae_colors("dark blue")
) +
geom_segment(aes(
  x = datensatz$x[5], y = schaeztung$fitted.values[5],
  xend = datensatz$x[5], yend = mean(datensatz$y)
),
arrow = arrow(length = unit(0.01, "npc"), ends = "both"),
colour = get_icae_colors("dark blue")
) +
annotate(geom = "text",
  label=TeX("$\\hat{Y}_1-\\bar{Y}$"),
  x = datensatz$x[1]-0.017, y = 3.25,
  colour = get_icae_colors("dark blue")) +
annotate(geom = "text",
  label=TeX("$\\hat{Y}_2-\\bar{Y}$"),
  x = datensatz$x[2]-0.018, y = 3.4,
  colour = get_icae_colors("dark blue")) +
annotate(geom = "text",
  label=TeX("$\\hat{Y}_3-\\bar{Y}$"),
  x = datensatz$x[3]-0.018, y = 3.7,
  colour = get_icae_colors("dark blue")) +
annotate(geom = "text",
  label=TeX("$\\hat{Y}_4-\\bar{Y}$"),
  x = datensatz$x[4]-0.018, y = 3.75,
  colour = get_icae_colors("dark blue")) +
annotate(geom = "text",
  label=TeX("$\\hat{Y}_5-\\bar{Y}$"),

```

```

x = datensatz$x[5]-0.018, y = 3.95,
colour = get_icae_colors("dark blue")) +
annotate(geom = "text",
label=TeX("$ESS = 0.94864$"),
x = 0.45, y = 3.0,
colour = get_icae_colors("dark blue")) +
theme_icae() +
theme(legend.position = "none")

```



$$TSS = ESS + RSS$$

$$R^2 = \frac{ESS}{TSS} = 1 - \frac{RSS}{TSS}$$

Händische Berechnung:

```

r_sq_manual <- ess / tss
r_sq_manual

```

```
## [1] 0.2874562
```

Auslesen aus dem Regressionsobjekt:

```

info_schaetzung <- summary(schaetzung)
info_schaetzung[["r.squared"]]

```

```
## [1] 0.2874562
```

## 5 Simulation der Schätzerverteilung

Grundstruktur des ‘wahren’ Modells:

```
set.seed(123)
true_DGP <- function(x, b0, b1){
  y <- b0 + b1*x + rnorm(length(x), 0, 5)
  return(y)
}
beta_0_wahr <- 3
beta_1_wahr <- 2
sample_size <- 100
x <- runif(sample_size, 0, 10)
```

Simulieren von 1000 Datensätzen und Schätzung von 1000 Regressionen:

```
set.seed(123)
n_datensaetze <- 1000
beta_0_estimates <- rep(NA, n_datensaetze)
beta_1_estimates <- rep(NA, n_datensaetze)

for (i in 1:n_datensaetze){
  daten_satz <- data.frame(
    x = x,
    y = true_DGP(x, beta_0_wahr, beta_1_wahr)
  )
  schaeztung_2 <- lm(y~x, data = daten_satz)
  beta_0_estimates[i] <- schaeztung_2[["coefficients"]][1]
  beta_1_estimates[i] <- schaeztung_2[["coefficients"]][2]
}
```

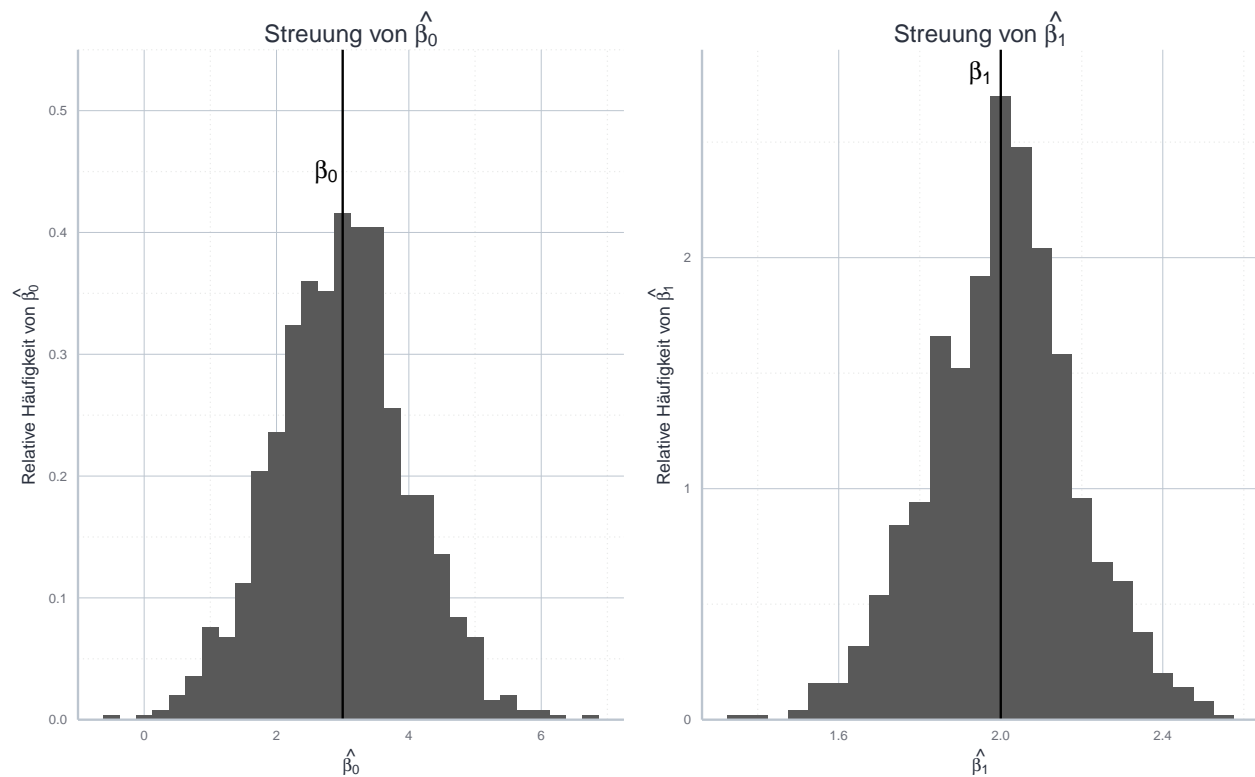
Abbildung der Ergebnisse:

```
beta_0_plot <- ggplot(
  data.frame(x=beta_0_estimates), aes(x=x)
) +
  geom_histogram(binwidth = 0.25, aes(y=..density..)) +
  geom_vline(xintercept = beta_0_wahr) +
  scale_y_continuous(expand = expand_scale(c(0, 0), c(0, 0.1))) +
  ylab(TeX("Relative Häufigkeit von  $\hat{\beta}_0$ ")) +
  xlab(TeX(" $\hat{\beta}_0$ ")) +
  ggtitle(TeX("Streuung von  $\hat{\beta}_0$ ")) +
  annotate(geom = "text", label=TeX(" $\beta_0$ "), x = beta_0_wahr-0.25, y=0.45) +
  theme_icae()

beta_1_plot <- ggplot(
  data.frame(x=beta_1_estimates), aes(x=x)
) +
  geom_histogram(binwidth = 0.05, aes(y=..density..)) +
  geom_vline(xintercept = beta_1_wahr) +
  scale_y_continuous(expand = expand_scale(c(0, 0), c(0, 0.1))) +
  ylab(TeX("Relative Häufigkeit von  $\hat{\beta}_1$ ")) +
  xlab(TeX(" $\hat{\beta}_1$ ")) +
  ggtitle(TeX("Streuung von  $\hat{\beta}_1$ ")) +
  annotate(geom = "text", label=TeX(" $\beta_1$ "),
    x = beta_1_wahr-0.05, y=2.8) +
```

```
theme_icae()

ggarrange(beta_0_plot, beta_1_plot, ncol = 2)
```



## 6 Preise von Journalen

Einlesen der Daten:

```
journale <- fread(here("data/T5/journaldaten.csv")) %>%
  select(-Titel, -Verlag, -Society, -Bereich, -Buchstaben_pS)
head(journale)
```

```
##      Kuerzel Preis Seitenanzahl Zitationen Gruendung Abonnenten
## 1:   APEL   123         440         21    1986         14
## 2:  SAJoEH    20         309         22    1986         59
## 3:    CE   443         567         22    1987         17
## 4:  MEPiTE  276         520         22    1991          2
## 5:   JoSE  295         791         24    1972         96
## 6:  LabEc  344         609         24    1994         15
```

Schätzung:

```
schaetzung_j <- lm(Preis~Seitenanzahl, data = journale)
summary(schaetzung_j)
```

```
##
## Call:
## lm(formula = Preis ~ Seitenanzahl, data = journale)
##
```

```
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1157.56  -190.54   -40.72   179.59  1329.30
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  56.74315   53.85199   1.054    0.293
## Seitenanzahl  0.43610    0.05757   7.575 1.89e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 336.5 on 178 degrees of freedom
## Multiple R-squared:  0.2438, Adjusted R-squared:  0.2395
## F-statistic: 57.38 on 1 and 178 DF,  p-value: 1.888e-12
```

## 7 Vergleich der multiplen und einfachen linearen Regression

Zunächst die Schätzung von

$$PREIS = \beta_0 + \beta_1 SEITEN + \epsilon_i$$

```
reg_einfach <- lm(Preis~Seitenanzahl, data = journale)
summary(reg_einfach)
```

```
##
## Call:
## lm(formula = Preis ~ Seitenanzahl, data = journale)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1157.56  -190.54   -40.72   179.59  1329.30
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  56.74315   53.85199   1.054    0.293
## Seitenanzahl  0.43610    0.05757   7.575 1.89e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 336.5 on 178 degrees of freedom
## Multiple R-squared:  0.2438, Adjusted R-squared:  0.2395
## F-statistic: 57.38 on 1 and 178 DF,  p-value: 1.888e-12
```

Und die Schätzung von

```
reg_mult <- lm(Preis~Seitenanzahl + Zitationen, data = journale)
summary(reg_mult)
```

```
##
## Call:
## lm(formula = Preis ~ Seitenanzahl + Zitationen, data = journale)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
```

```
## -1346.70 -173.48 -38.83 138.32 1259.00
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.72002   52.80969  -0.070    0.944
## Seitenanzahl  0.59413    0.06477   9.173 < 2e-16 ***
## Zitationen   -0.10872    0.02393  -4.544 1.02e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 319.3 on 177 degrees of freedom
## Multiple R-squared:  0.3228, Adjusted R-squared:  0.3151
## F-statistic: 42.18 on 2 and 177 DF, p-value: 1.049e-15
```

Man beachte der veränderten Koeffizienten von `Seitenanzahl` im Vergleich zu der einfachen Regression.