Beispiel: Regressionsmodelle in R

Statistical Aspects (09-202-2413)

Janne Pott

Last compiled on 06 Oktober, 2022

Session Setup

```
rm(list = ls())
time0<-Sys.time()

source("../sourceFile.R")
setwd(pathToExample)

knitr::opts_chunk$set(echo = TRUE)
```

Lineare Regression

In dem ersten Teil dieser Übung beschäftigen wir uns mit einfacher linearer Regression.

Wiederholung aus der VL

- Regresswerte = Vorhergesagte Werte: $\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i$
- Residuen = Abweichung Vorhersage - Beobachtung: $\hat{y}_i - y_i$
- Residual Sum of Square (RSS) = Summe der quadratischen Fehler: $\sum_{i=1}^{n} (\hat{y}_i - y_i)^2$
- Mean Square Error = Residuale Varianz: $\frac{1}{n} \sum_{i=1}^{n} (\hat{y}_i - y_i)^2$
- Lineares Regressionsmodell: $Y = \beta_0 + \beta_1 X_1 + ... + \beta_n X_n + \epsilon, \epsilon \sim N(0, \sigma^2)$

Das heißt, bei der linaren Regression versucht man eine beobachtete abhängige Variable y durch eine oder mehrere unabhängige Variablen x zu erklären, wobei die RSS minimal werden soll. Im Falle einer einfachen linearen Regression mit je einer unabhängigen und einer abhängigen Variablen muss der Term

$$RSS(\beta_0, \beta_1) = \sum_{i=1}^{n} [(\beta_0 + \beta_1 x_i) - y_i]^2$$

partiell für β_0 und β_1 abgeleitet werden.

In R ist diese Rechnung in der Funktion *lm()* implementiert. Wir nutzen hier wieder den *iris* Datensatz als Beispiel.

```
data(iris)

# Modell 1: Sepal.Length ~ Sepal.Width
```

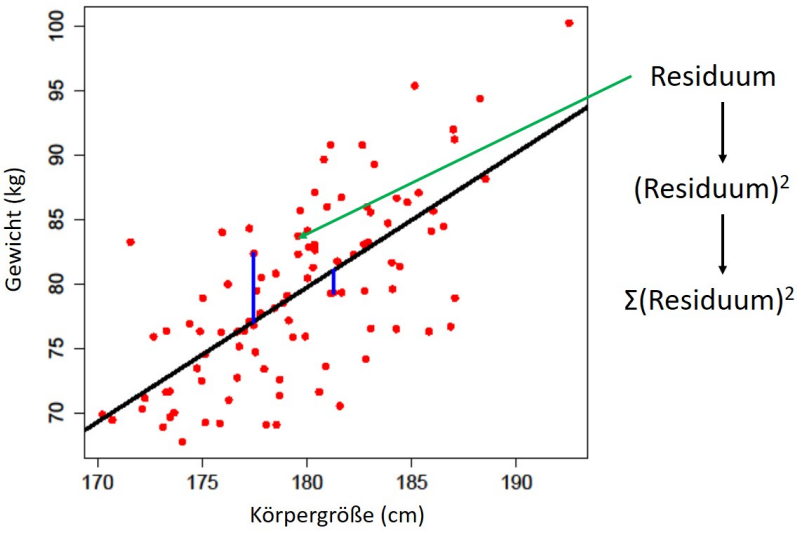


Figure 1: Lineare Regression. Quelle: Vorlesung

```
mod1 = lm(Sepal.Length ~ Sepal.Width,data = iris)
summary(mod1)
```

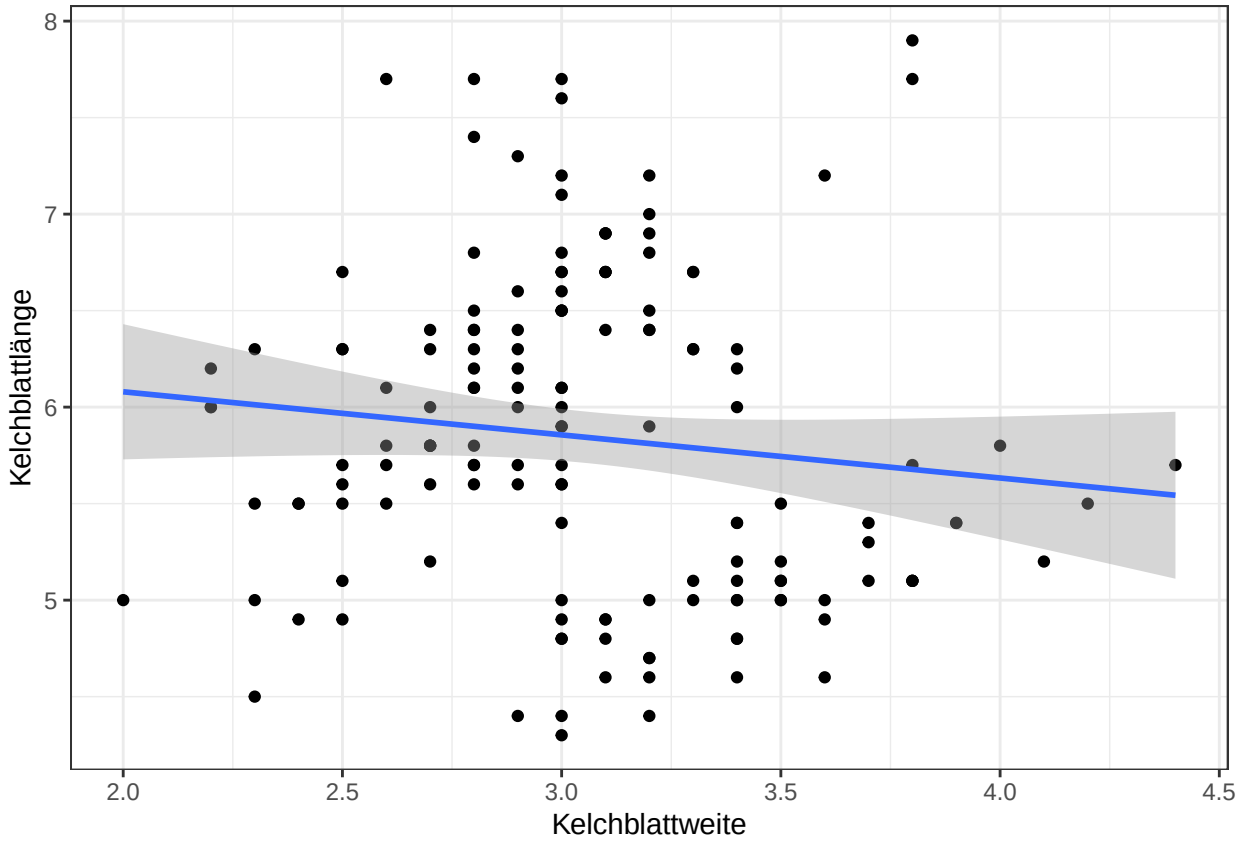
```
##
## Call:
## lm(formula = Sepal.Length ~ Sepal.Width, data = iris)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.5561 -0.6333 -0.1120  0.5579  2.2226
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   6.5262     0.4789   13.63   <2e-16 ***
## Sepal.Width  -0.2234     0.1551   -1.44    0.152
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8251 on 148 degrees of freedom
## Multiple R-squared:  0.01382,    Adjusted R-squared:  0.007159
## F-statistic: 2.074 on 1 and 148 DF,  p-value: 0.1519
```

```
summary(mod1)$coeff
```

```
##               Estimate Std. Error   t value     Pr(>|t|)
## (Intercept)  6.5262226  0.4788963 13.627631 6.469702e-28
## Sepal.Width -0.2233611  0.1550809 -1.440287 1.518983e-01
```

```
ggplot(iris,aes(x=Sepal.Width,y=Sepal.Length)) +
  geom_point() +
  geom_smooth(method = 'lm',formula = 'y ~ x') +
  theme_bw() +
```

```
  scale_x_continuous('Kelchblattweite') +
  scale_y_continuous ('Kelchblattlänge')
```



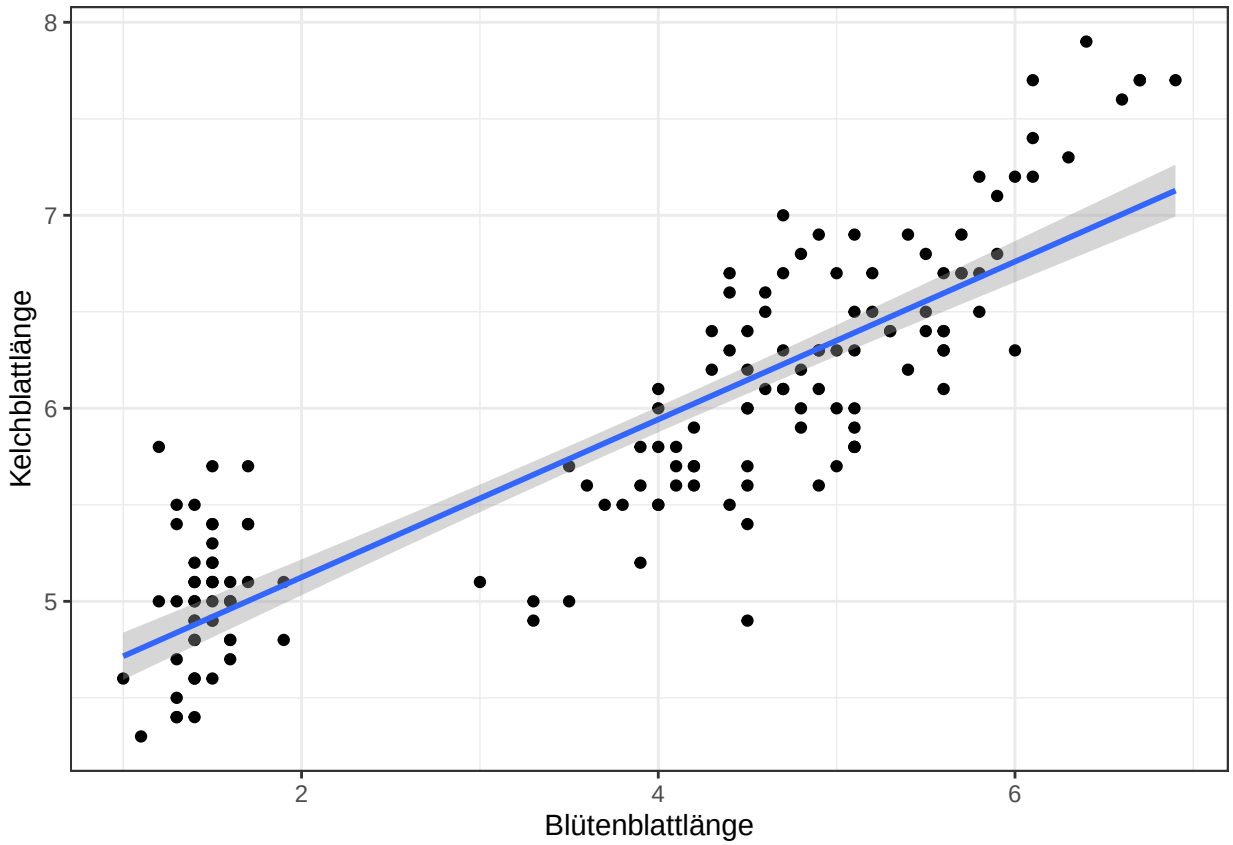

```
# Modell 2: Sepal.Length ~ Petal.Length
mod2 = lm(Sepal.Length ~ Petal.Length,data = iris)
summary(mod2)
```

```
##
## Call:
## lm(formula = Sepal.Length ~ Petal.Length, data = iris)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.24675 -0.29657 -0.01515  0.27676  1.00269
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   4.30660    0.07839   54.94   <2e-16 ***
## Petal.Length  0.40892    0.01889   21.65   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4071 on 148 degrees of freedom
## Multiple R-squared:   0.76,  Adjusted R-squared:  0.7583
## F-statistic: 468.6 on 1 and 148 DF,  p-value: < 2.2e-16
```

```
summary(mod2)$coeff
```

```
##                Estimate Std. Error  t value      Pr(>|t|)
## (Intercept)  4.3066034 0.07838896 54.93890 2.426713e-100
## Petal.Length 0.4089223 0.01889134 21.64602  1.038667e-47
```

```
ggplot(iris,aes(x=Petal.Length,y=Sepal.Length)) +
  geom_point() +
  geom_smooth(method = 'lm',formula = 'y ~ x') +
  theme_bw() +
  scale_x_continuous('Blütenblattlänge') +
  scale_y_continuous ('Kelchblattlänge')
```



Wir sehen also, dass es einen signifikanten Zusammenhang zwischen der Kelch- und Blütenblattlänge gibt ($p = 1.04x10^{-47}$), aber keinen zwischen Kelchlänge und -weite ($p = 0.152$).

Das lineare Modell von Kelch- und Blütenblattlänge lässt sich schreiben als

$$Sepal.Lenght = \beta_0 + \beta_1 Petal.Length + \epsilon = 4.31 + 0.41 * Petal.Length + \epsilon, \epsilon \sim N(0, \sigma^2)$$

Dieses Modell erklärt 76% der Varianz der Kelchblattlänge (r^2 aus dem *summary* Aufruf).

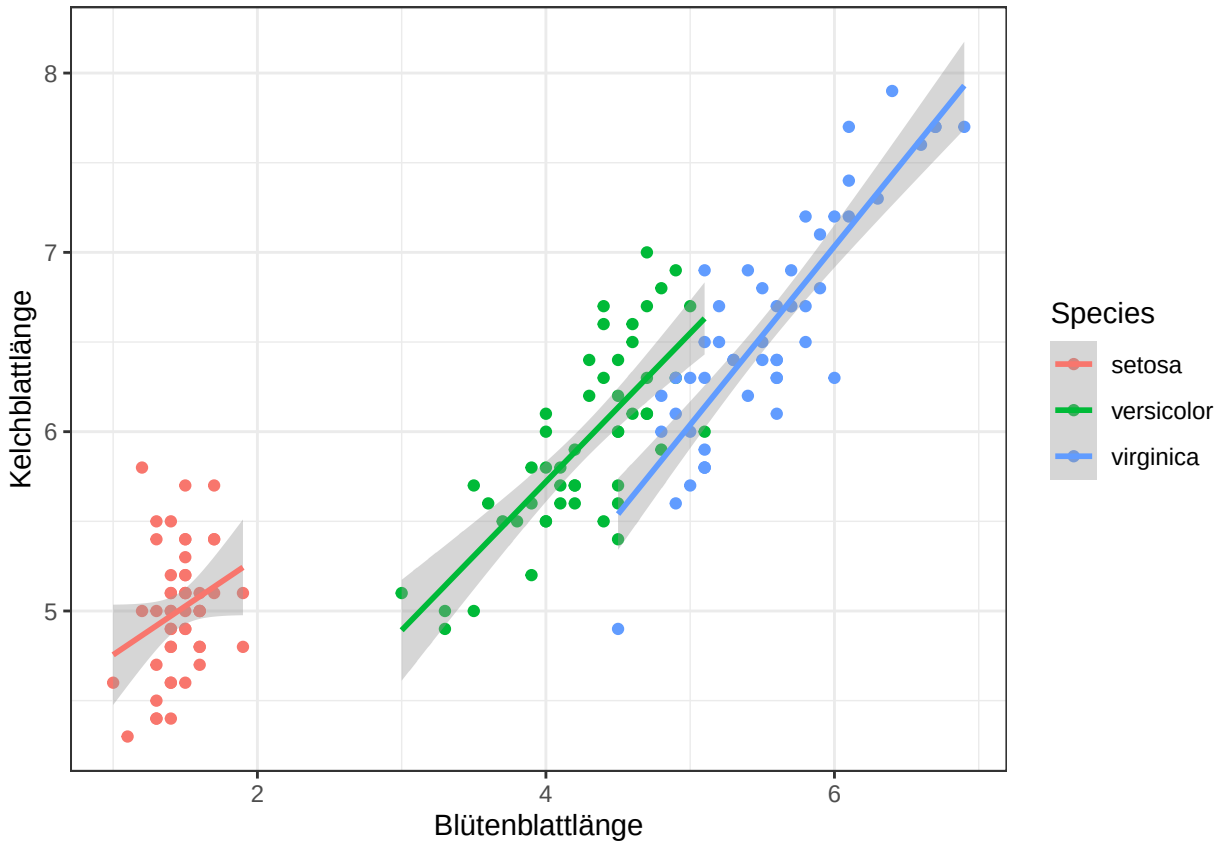
In diesem Datensatz werden jedoch 3 Spezies auf einmal betrachtet, und die Spezies hat ebenfalls einen Effekt auf die Kelchblattlänge. Um beide Variablen gleichzeitig zu analysieren, gibt es verschiedene Möglichkeiten:

- Multiple lineare Regression: $Sepal.Lenght = \beta_0 + \beta_1 Petal.Length + \beta_2 Species + \epsilon$
- Stratifizierte Analyse: $Sepal.Lenght_{Species} = \beta_{0,Species} + \beta_{1,Species} Petal.Length + \epsilon$ pro Species
- Interaktionsanalyse: $Sepal.Lenght = \beta_0 + \beta_1 Petal.Length + \beta_2 Species + \beta_3 Petal.Length * Species + \epsilon$

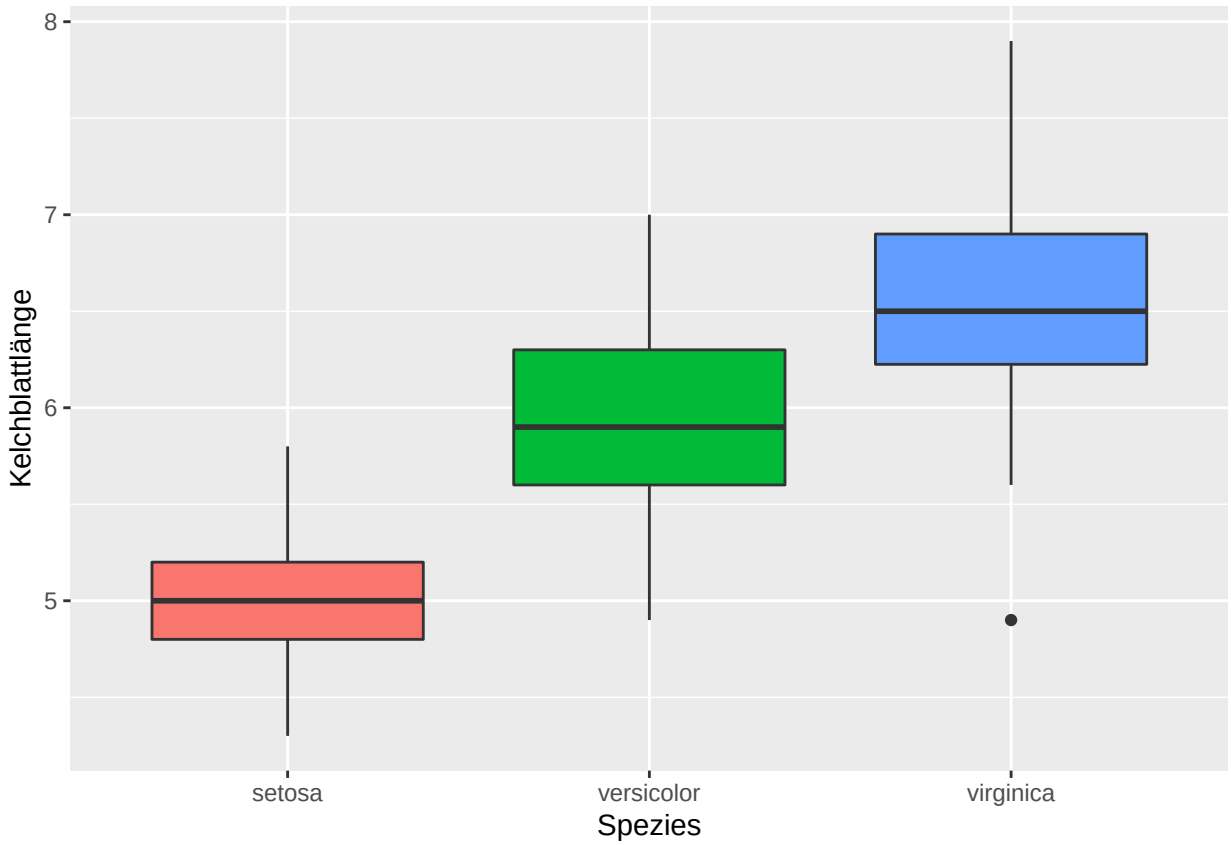
Sowohl mit der stratifizierten als auch mit der Interaktionsanalyse können Interaktionseffekte zwischen *Petal.Length* und *Species* berücksichtigt werden. Bei der stratifizierten Analyse kann die Differenz der

jeweiligen Schätzer verglichen werden, und bei der Interaktionsanalyse kann der Schätze β_3 mittels t-Test gegen 0 getestet werden.

```
ggplot(iris,aes(x=Petal.Length,y=Sepal.Length,col = Species)) +
  geom_point() +
  geom_smooth(method = 'lm',formula = 'y ~ x') +
  theme_bw() +
  scale_x_continuous('Blütenblattlänge') +
  scale_y_continuous ('Kelchblattlänge')
```




```
ggplot(iris,aes(x=Species,y=Sepal.Length,fill = Species)) +
  geom_boxplot() +
  theme(legend.position="none") +
  scale_x_discrete('Spezies') +
  scale_y_continuous ('Kelchblattlänge')
```




```
mod3 = lm(Sepal.Length ~ Petal.Length + Species,data = iris)
mod4a = lm(Sepal.Length ~ Petal.Length,data = iris,subset = Species == "setosa")
mod4b = lm(Sepal.Length ~ Petal.Length,data = iris,subset = Species == "versicolor")
mod4c = lm(Sepal.Length ~ Petal.Length,data = iris,subset = Species == "virginica")
mod5 = lm(Sepal.Length ~ Petal.Length*Species,data = iris)

summary(mod3)$coeff
```

```
##                     Estimate Std. Error   t value     Pr(>|t|)
## (Intercept)        3.6835266 0.10609608 34.718780 1.968671e-72
## Petal.Length       0.9045646 0.06478559 13.962436 1.121002e-28
## Speciesversicolor -1.6009717 0.19346616 -8.275203 7.371529e-14
## Speciesvirginica  -2.1176692 0.27346121 -7.743947 1.480296e-12
```

```
summary(mod4a)$coeff
```

```
##              Estimate Std. Error   t value     Pr(>|t|)
## (Intercept)  4.2131682  0.4155888 10.137830 1.614927e-13
## Petal.Length 0.5422926  0.2823153  1.920876 6.069778e-02
```

```
summary(mod4b)$coeff
```

```
##              Estimate Std. Error  t value     Pr(>|t|)
## (Intercept)  2.407523  0.4462583 5.394909 2.075294e-06
## Petal.Length 0.828281  0.1041364 7.953806 2.586190e-10
```

```
summary(mod4c)$coeff
```

```
##              Estimate Std. Error   t value     Pr(>|t|)
```

```
## (Intercept)  1.0596591 0.46676645  2.270213 2.772289e-02
## Petal.Length 0.9957386 0.08366764 11.901120 6.297786e-16
```

```
summary(mod5)$coeff
```

```
##                                  Estimate Std. Error   t value     Pr(>|t|)
## (Intercept)                     4.2131682  0.4074209 10.341071 4.331619e-19
## Petal.Length                    0.5422926  0.2767667  1.959385 5.199902e-02
## Speciesversicolor              -1.8056451  0.5984284 -3.017312 3.016413e-03
## Speciesvirginica               -3.1535091  0.6340741 -4.973408 1.846894e-06
## Petal.Length:Speciesversicolor  0.2859884  0.2950624  0.969247 3.340471e-01
## Petal.Length:Speciesvirginica   0.4534460  0.2901455  1.562823 1.202893e-01
```

Die Modelle 4a-c testen seperat pro Spezies den Effekt, während Modell 5 alle Spezies gemeinsam analysiert und auch auf Interaktion testet. Die Schätzer für Intercept und Steigung pro *Petal.Length* sind jedoch die gleichen:

$$\begin{pmatrix} y_{setosa} \\ y_{versicolor} \\ y_{virginica} \end{pmatrix} = 4.21 + \begin{pmatrix} 0 \\ -1.81 \\ -3.15 \end{pmatrix} + (0.54 + \begin{pmatrix} 0 \\ 0.29 \\ 0.45 \end{pmatrix}) * x + \epsilon$$

Die letzten beiden Zeilen der Koeffizientenmatrix von Model 5 gibt die Interaktionseffekte an. Beide sind nicht signifikant. D.h. man kann die Nullhypothese von keiner Interaktion nicht ablehnen.

Auch wenn man die stratifizierten Schätzer verwendet, gibt es keine signifikante Interaktion:

```
interactionTest = function (mean1, se1, mean2, se2) {
    meandiff_se = sqrt(se1^2 + se2^2)
    meandiff = mean2 - mean1
    meandiff_cilow = meandiff - 1.96 * meandiff_se
    meandiff_cihigh = meandiff + 1.96 * meandiff_se
    meandiff_z = meandiff/meandiff_se
    meandiff_p = stats::pnorm(abs(meandiff_z), lower.tail = F) * 2
    if (meandiff_p > 1)
        meandiff_p = 1
    data.table::data.table(mean1, se1, mean2, se2, meandiff,
        meandiff_se, meandiff_cilow, meandiff_cihigh, meandiff_z,
        meandiff_p)
}
interactionTest(mean1 = summary(mod4a)$coeff[2,1],se1 = summary(mod4a)$coeff[2,2],
                mean2 = summary(mod4b)$coeff[2,1],se2 = summary(mod4b)$coeff[2,2])
```

```
##        mean1       se1    mean2       se2  meandiff meandiff_se meandiff_cilow
## 1: 0.5422926 0.2823153 0.828281 0.1041364 0.2859884   0.3009091     -0.3037935
##    meandiff_cihigh meandiff_z meandiff_p
## 1:       0.8757702  0.9504144  0.3419017
```

```
interactionTest(mean1 = summary(mod4a)$coeff[2,1],se1 = summary(mod4a)$coeff[2,2],
                mean2 = summary(mod4c)$coeff[2,1],se2 = summary(mod4c)$coeff[2,2])
```

```
##        mean1       se1     mean2        se2 meandiff meandiff_se meandiff_cilow
## 1: 0.5422926 0.2823153 0.9957386 0.08366764 0.453446   0.2944523     -0.1236805
##    meandiff_cihigh meandiff_z meandiff_p
## 1:        1.030573   1.539964  0.1235691
```

```
interactionTest(mean1 = summary(mod4c)$coeff[2,1],se1 = summary(mod4c)$coeff[2,2],
                mean2 = summary(mod4b)$coeff[2,1],se2 = summary(mod4b)$coeff[2,2])
```

```
##        mean1        se1    mean2       se2   meandiff meandiff_se
## 1: 0.9957386 0.08366764 0.828281 0.1041364 -0.1674577   0.1335839
##    meandiff_cilow meandiff_cihigh meandiff_z meandiff_p
## 1:     -0.4292822      0.09436686  -1.253576   0.209996
```

Logistische / Proportional Odds Regression

Bei der **logistischen Regression** (=logit Regression) werden binäre abhängige Variablen betrachtet, z.B. Erkrankung ja/nein. In der Regel werden diese beiden Ausprägungen mit 0 und 1 kodiert, sodass man das Ergebnis der logistischen Regression als Wahrscheinlichkeit auffassen kann.

Die Odds geben das Verhältnis von Ereigniswahrscheinlichkeit $P(Y = 1)$ zur Gegenwahrscheinlichkeit $P(Y = 0)$ an. Diese werden im logit Modell logarithmiert, sodass die Werte zwischen - und + unendlich liegen. Diese kontinuierlichen logits koppeln die Wahrscheinlichkeit mit der linearen Prädiktion.

$$Logit(Y_{1/0}) := ln(Odds(Y_{1/0}) = ln(\frac{P(Y_i = 1)}{1 - P(Y_i = 1)}$$

$$Logit(Y_{1/0}|X_i = x_i) = \beta_0 + \beta_1 x_{i,1} + ... + \beta_n x_{i,n}$$

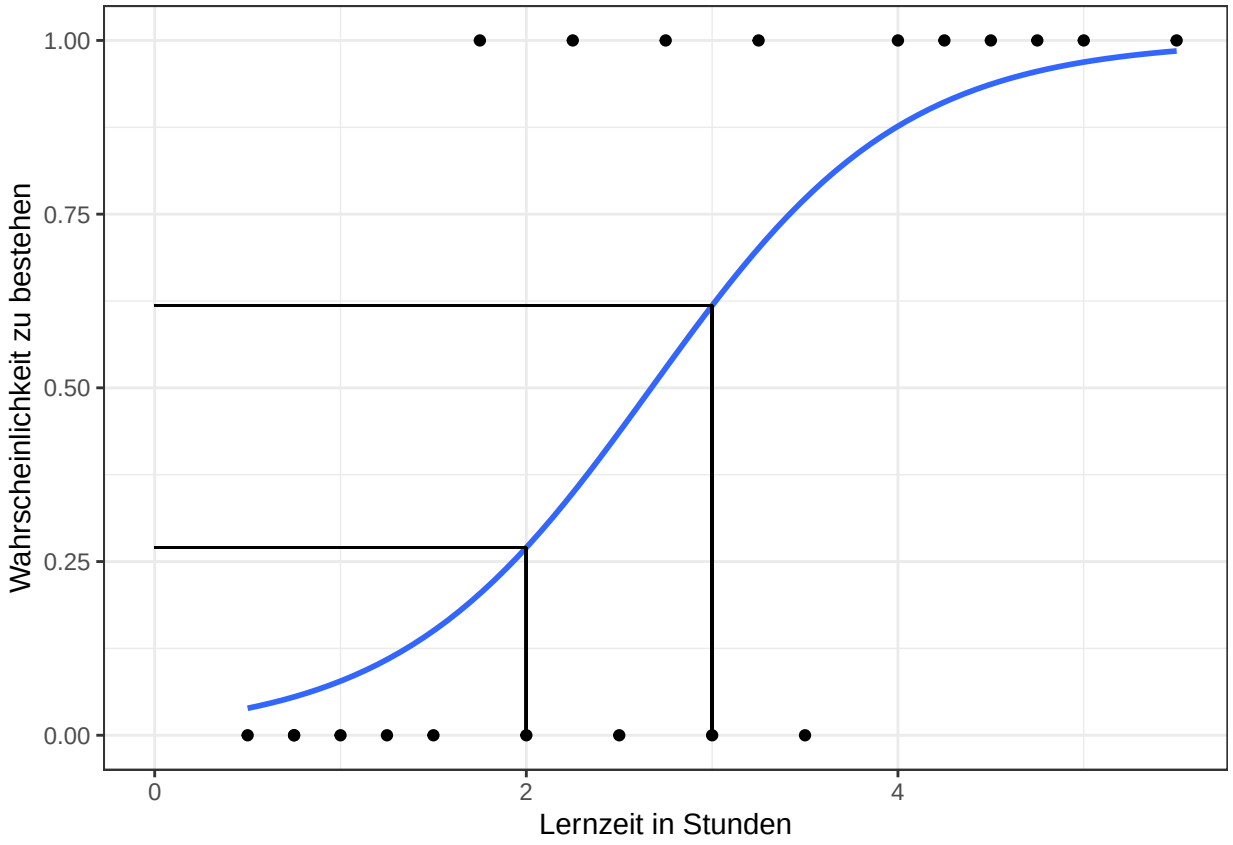
In diesem Beispiel testen wir, ob die Lernzeit einen Effekt auf das Bestehen einer Prüfung hat.

```
myTab = data.table(learning = c(0.50,0.75,1.00,1.25,1.50,0.75,1.75,2.00,2.25,2.50,
                                2.75,3.00,3.25,3.50,4.00,4.25,4.50,4.75,5.00,5.50),
                   pass = c(0,0,0,0,0,0,1,0,1,0,1,0,1,0,1,1,1,1,1,1),
                   grade = c(4,4,4,4,4,4,4,3,3,3,3,3,2,2,2,2,1,1,1,1))
mod6 = glm(pass ~ learning,data = myTab,family = "binomial")
summary(mod6)
```

```
##
## Call:
## glm(formula = pass ~ learning, family = "binomial", data = myTab)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.72039  -0.50297  -0.05338   0.45167   1.78369
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -3.9474     1.7158  -2.301   0.0214 *
## learning      1.4768     0.6118   2.414   0.0158 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 27.726  on 19  degrees of freedom
## Residual deviance: 15.728  on 18  degrees of freedom
## AIC: 19.728
##
## Number of Fisher Scoring iterations: 5
```

```r
# learning = 2 and 3 hours
x1 = exp(summary(mod6)$coeff[1,1] + summary(mod6)$coeff[2,1]*2) / (1+exp(summary(mod6)$coeff[1,1] + summ
x2 = exp(summary(mod6)$coeff[1,1] + summary(mod6)$coeff[2,1]*3) / (1+exp(summary(mod6)$coeff[1,1] + summ

ggplot(myTab,aes(x=learning,y=pass)) +
   geom_point() +
  geom_smooth(method = "glm",formula = 'y ~ x', method.args = list(family = "binomial"), se = FALSE) +
  theme_bw() +
  scale_x_continuous('Lernzeit in Stunden') +
  scale_y_continuous ('Wahrscheinlichkeit zu bestehen')+
  geom_segment(aes(x = 2, y = 0, xend = 2, yend = x1))+
  geom_segment(aes(x = 0, y = x1, xend = 2, yend = x1))+
  geom_segment(aes(x = 3, y = 0, xend = 3, yend = x2))+
  geom_segment(aes(x = 0, y = x2, xend = 3, yend = x2))
```



$$Logit(Y_{1/0}|X_i = x_i) = \beta_0 + \beta_1 x_i = -3.95 + 1.48 * x$$

D.h. jede Stunde Lernzeit erhöht die log-odds um 1.48, und bei etwa 2.7 Stunden Lernzeit hat man eine 50% Chance zu bestehen.

$$P(Y = 1|X = 2) = \frac{exp(-3.95 + 1.48 * 2)}{1 + exp(-3.95 + 1.48 * 2)} = 0.27$$

$$P(Y = 1|X = 3) = \frac{exp(-3.95 + 1.48 * 3)}{1 + exp(-3.95 + 1.48 * 3)} = 0.62$$

Bei der **Proportional Odds Regression** hat man nicht mehr binäre Ereignisse, sondern ordinale Kategorien (Noten 1-5), mit der Fragestellung, wie der Lernaufwand die Wahrscheinlichkeit die Prüfung mit Note 1, 2, 3, 4 oder nicht zu bestehen beeinflusst. Es gilt die Annahme: "equal slope", d.h. die logistische Funktion für die Wahrscheinlichkeit, mindestens Note j zu erreichen, verläuft für jede Note parallel verschoben, aber mit der gleichen Steigung.

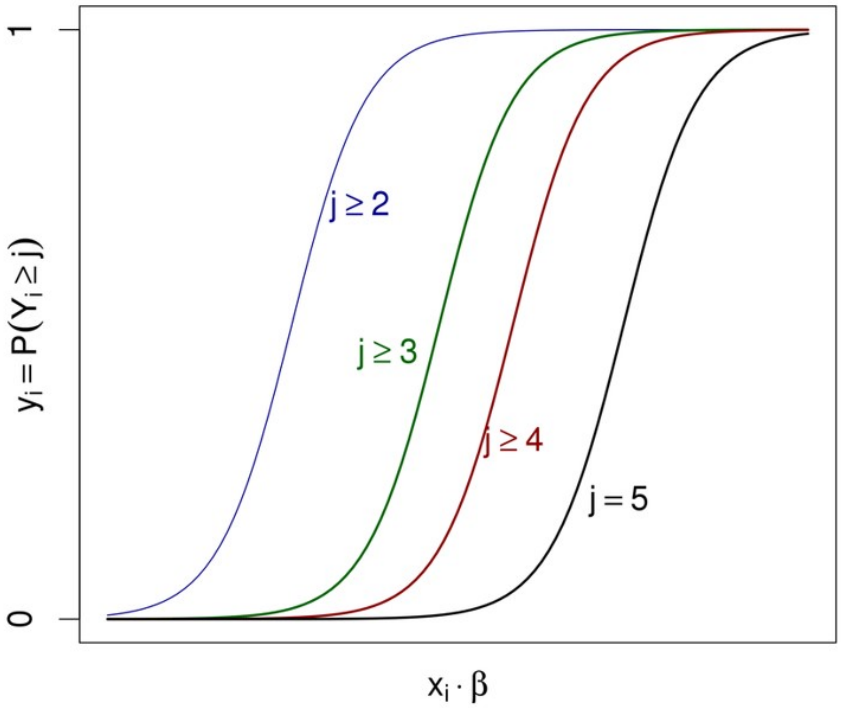


Figure 2: Proportional Odds Regression. Quelle: Schlarmann, Galatsch, 2014

```
setorder(myTab,grade)
myTab[,grade2 := grade-1]
factor_temperature_vector <- factor(myTab$grade, order = TRUE,
                                    levels = c("1", "2", "3","4","5"))
mod7<-polr(as.factor(grade)~learning,data = myTab,Hess = T)
summary(mod7)
```

Nichtlineare Regression

Bei einer **nichtlinearen Regression** können die Daten an jede Gleichung der Form $y = f(\alpha)$, also Kurven, angepasst werden. Im Allgemeinen ergibt sich bei nichtlinearen Modellfunktionen ein Problem der Form

$$min||f(\alpha) - y||_2$$

mit einer nichtlinearen Funktion f. Ein Beispiel dafür ist die Enzymkinetik, die im Aufgabenblatt gestellt ist.

Session Information

```
sessionInfo()
```

```
## R version 4.1.0 (2021-05-18)
## Platform: x86_64-suse-linux-gnu (64-bit)
## Running under: openSUSE Leap 15.3
##
## Matrix products: default
## BLAS:   /usr/lib64/R/lib/libRblas.so
## LAPACK: /usr/lib64/R/lib/libRlapack.so
##
## locale:
##  [1] LC_CTYPE=de_DE.UTF-8       LC_NUMERIC=C
##  [3] LC_TIME=de_DE.UTF-8        LC_COLLATE=de_DE.UTF-8
##  [5] LC_MONETARY=de_DE.UTF-8    LC_MESSAGES=de_DE.UTF-8
##  [7] LC_PAPER=de_DE.UTF-8       LC_NAME=C
##  [9] LC_ADDRESS=C               LC_TELEPHONE=C
## [11] LC_MEASUREMENT=de_DE.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] parallel  stats     graphics  grDevices utils     datasets  methods
## [8] base
##
## other attached packages:
##  [1] corrplot_0.92     ivpack_1.2        AER_1.2-9         sandwich_3.0-1
##  [5] lmtest_0.9-39     car_3.0-11        carData_3.0-5     qqman_0.1.8
##  [9] meta_5.1-1        nlme_3.1-155      Hmisc_4.4-2       ggplot2_3.3.5
## [13] Formula_1.2-4     survival_3.2-13   lattice_0.20-45   MASS_7.3-55
## [17] vioplot_0.3.7     zoo_1.8-9         sm_2.2-5.7        lubridate_1.8.0
## [21] readxl_1.3.1      data.table_1.14.2 doMC_1.3.7        doParallel_1.0.16
## [25] iterators_1.0.13  foreach_1.5.1     rmarkdown_2.11
##
## loaded via a namespace (and not attached):
##  [1] RColorBrewer_1.1-2  tools_4.1.0         backports_1.4.1
##  [4] utf8_1.2.2          R6_2.5.1            metafor_3.0-2
##  [7] rpart_4.1-15        mgcv_1.8-38         DBI_1.1.2
## [10] colorspace_2.0-2    nnet_7.3-17         withr_2.4.3
## [13] tidyselect_1.1.1    gridExtra_2.3       curl_4.3
## [16] compiler_4.1.0      htmlTable_2.4.0     xml2_1.3.2
## [19] labeling_0.4.2      scales_1.1.1        checkmate_2.0.0
## [22] stringr_1.4.0       digest_0.6.29       foreign_0.8-82
## [25] minqa_1.2.4         rio_0.5.27          base64enc_0.1-3
## [28] jpeg_0.1-9          pkgconfig_2.0.3     htmltools_0.5.2
## [31] lme4_1.1-27.1       highr_0.9           fastmap_1.1.0
## [34] htmlwidgets_1.5.4   rlang_0.4.12        rstudioapi_0.13
## [37] farver_2.1.0        generics_0.1.1      dplyr_1.0.7
## [40] zip_2.2.0           magrittr_2.0.1      Matrix_1.4-0
## [43] Rcpp_1.0.8          munsell_0.5.0       fansi_1.0.0
## [46] abind_1.4-5         lifecycle_1.0.1     stringi_1.7.6
## [49] yaml_2.2.1          CompQuadForm_1.4.3  mathjaxr_1.4-0
## [52] grid_4.1.0          forcats_0.5.1       crayon_1.4.2
## [55] haven_2.3.1         splines_4.1.0       hms_1.1.1
```

```
## [58] knitr_1.37          pillar_1.6.4        boot_1.3-28
## [61] codetools_0.2-18    glue_1.6.0          evaluate_0.14
## [64] calibrate_1.7.7     latticeExtra_0.6-29 png_0.1-7
## [67] vctrs_0.3.8         nloptr_1.2.2.3      cellranger_1.1.0
## [70] gtable_0.3.0        purrr_0.3.4         assertthat_0.2.1
## [73] xfun_0.29           openxlsx_4.2.5      tibble_3.1.6
## [76] cluster_2.1.2       ellipsis_0.3.2
```

```
message("\nTOTAL TIME : " ,round(difftime(Sys.time(),time0,units = "mins"),3)," minutes")
```

```
##
## TOTAL TIME : 0.192 minutes
```