MLE01 - Vertiefende statistische Verfahren

1. Übungsblatt SS 2024

Stefan Kolb, Joachim Waltl

Allgemeine Information

Alle Aufgaben sind mit R zu lössen. Die Berechnungen sollen nachvollziehbar und dokumentiert sein. Um die vollständige Punktezahl zu erreichen, müssen alle Ergebnisse und Fragen entsprechend interpretiert bzw. beantwortet werden. Code alleine ist nicht ausreichend! Die Abgabe erfolgt über Moodle entsprechend der Abgaberichtlinien als pdf und Rmd File. Bitte inkludieren Sie namentlich alle beteiligten Gruppenmitglieder sowohl im Bericht als auch im Source Code. Die jeweiligen Datensätze die für diese Übung relevant sind finden Sie ebenfalls in Moodle.

1 Lineare Regressionsanalyse [4P]

Für Menschen, die ihren Blutdruck senken wollen, ist eine häufig empfohlene Vorgehensweise, die Salzaufnahme zu senken. Sie möchten feststellen, ob es eine lineare Beziehung zwischen Salzaufnahme und Blutdruck gibt. Sie nehmen 52 Personen in die Stichprobe auf und messen deren diastolischen Blutdruck (in mmHg) und Natriumausscheidung (mmol/24h). [ref]

[2P] a: Importieren Sie den Datensatz `intersalt.csv`. Erstellen Sie zwei Regressionsmodelle für den diastolische Blutdruck (bp) in Abhängigkeit der Natriumausscheidung (na). Das erste Modell soll alle Datenpunkte verwenden. Für das zweite Modell sollen die vier Datenpunkte mit der geringsten Natriumausscheidung aus dem Datensatz entfernt werden.

```r
# import intersalt data
intersalt <- read.csv("intersalt.csv", sep = ";", dec = ",")

# summary of intersalt data
str(intersalt)
```

```
## 'data.frame':    52 obs. of  4 variables:
##  $ b      : num  0.512 0.226 0.316 0.042 0.086 0.265 0.384 0.501 0.352 0.443 ...
##  $ bp     : num  72 78.2 73.9 61.7 61.4 73.4 79.2 66.6 82.1 75 ...
##  $ na     : num  149.3 133 142.6 5.8 0.2 ...
##  $ country: chr  "Argentina" "Belgium" "Belgium" "Brazil" ...
```

```r
# transform variables from character to numeric
intersalt$b <- as.numeric(intersalt$b)
intersalt$bp <- as.numeric(intersalt$bp)
intersalt$na <- as.numeric(intersalt$na)

# first model (all data points)
model_1 <- lm(bp ~ na, data = intersalt)
```

```r
# sort data by na
intersalt_sorted <- intersalt[order(intersalt$na),]

# new data set without the 4 smallest na values
intersalt_m2 <- intersalt_sorted[-c(1:4),]

# second model (without 4 smallest na values)
model_2 <- lm(bp ~ na, data = intersalt_m2)
```

Führen Sie für beide Modelle eine lineare Regressionsanalyse durch, die folgende Punkte umfasst:

i) Modellgleichung inklusive 95% Konfidenzintervall der Modellparameter
ii) Interpretation des Ergebnisses hinsichtlich Signifikanz und Modellgüte
iii) Grafische Darstellung der Regressionsgeraden inkl. Konfidenzintervall

Modell 1

```r
# summary and confidence interval for model_1
summary(model_1)
```

```
##
## Call:
## lm(formula = bp ~ na, data = intersalt)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.8625 -2.8906  0.0299  3.6470  9.4283
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 67.56245    2.14643  31.477   <2e-16 ***
## na           0.03768    0.01384   2.722   0.0089 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.511 on 50 degrees of freedom
## Multiple R-squared:  0.1291, Adjusted R-squared:  0.1117
## F-statistic: 7.411 on 1 and 50 DF,  p-value: 0.008901
```

```r
confint(model_1)
```

```
##                    2.5 %      97.5 %
## (Intercept) 63.251226427 71.87367736
## na           0.009878513  0.06547863
```

i) Die Modellgleichung für das erste Modell lautet: bp = 67.56 [63.25;71.87] + 0.038 [0.01;0.07] * na.

ii) Interpretation der Signifikanz und Modellgüte:

Intercept: Das 95% Konfidenzintervall für den Intercept liegt zwischen 63.25123 und 71.87368. Das bedeutet, dass der diastolische Blutdruck bei einer Natriumausscheidung von 0 mmol/24h mit einer 95%igen Sicherheit zwischen 63.25 und 71.87 mmHg liegt.

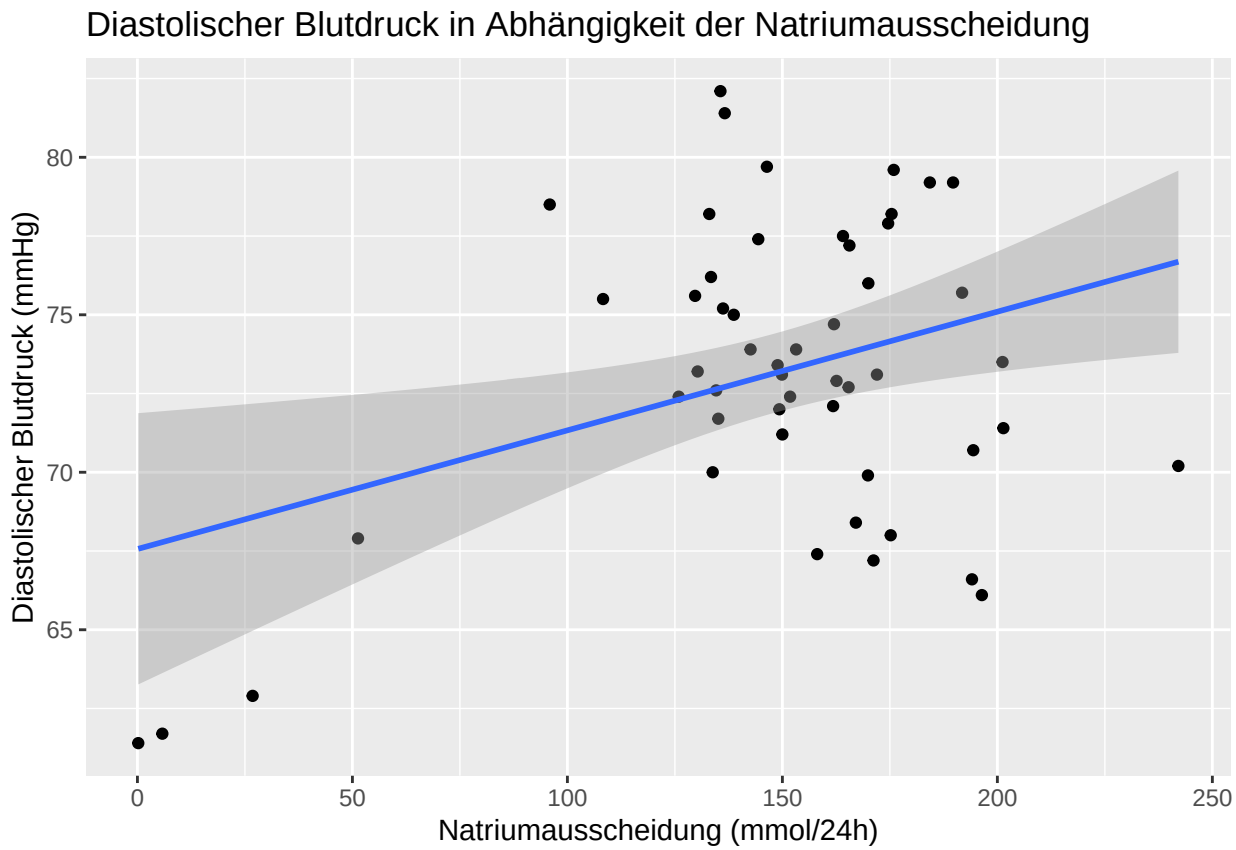
Steigung: Das 95% Konfidenzintervall für die Steigung bezüglich der Natriumausscheidung liegt zwischen 0.00988 und 0.06548. Wir können also mit 95%iger Sicherheit sagen, dass der Anstieg des diastolische Blutdruck zwischen 0.00988 und 0.06548 beträgt, wenn die Natriumausscheidung um 1 mmol/24h steigt.

Signifikanz und Modellgüte: Die Signifikanz des p-Wertes für die Steigung ($p = 0.0089$) deutet auf einen statistisch signifikanten Zusammenhang zwischen der Natriumausscheidung und dem diastolischen Blutdruck hin. Der p-Wert für den Intercept dieses Modells ist sogar als hochsignifikant zu werten. Ein Wert von 0.1291 für das Bestimmtheitsmaß R^2 zeigt jedoch, dass das Modell nur etwa 12.91% der Variabilität im diastolischen Blutdruck mit der Natriumausscheidung erklären kann. Die Erkenntnis daraus ist, dass noch andere Faktor den Blutdruck beeinflussen.

iii) Grafische Darstellung der Regressionsgeraden inkl. Konfidenzintervall

```
# regression plot for model_1
library(ggplot2)
ggplot(intersalt, aes(x = na, y = bp)) +
  geom_point() +
  geom_smooth(method = "lm", se = TRUE) +
  labs(title = "Diastolischer Blutdruck in Abhängigkeit der Natriumausscheidung",
       x = "Natriumausscheidung (mmol/24h)",
       y = "Diastolischer Blutdruck (mmHg)")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



```r
# summary and confidence interval for model_2
summary(model_2)
```

```
##
## Call:
## lm(formula = bp ~ na, data = intersalt_m2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.5966 -2.4042 -0.4884  2.8636  7.0977
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 81.06335    3.30938  24.495   <2e-16 ***
## na          -0.04470    0.02053  -2.177   0.0346 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.807 on 46 degrees of freedom
## Multiple R-squared:  0.09342,    Adjusted R-squared:  0.07371
## F-statistic:  4.74 on 1 and 46 DF,  p-value: 0.03464
```

```r
confint(model_2)
```

```
##                   2.5 %      97.5 %
## (Intercept) 74.40191390 87.72478658
## na          -0.08602363 -0.00337225
```

i) Die Modellgleichung für das zweite Modell lautet: bp = 81.06 [74.40;87.72] - 0.045 [-0.09;0.00] * na.

ii) Interpretation der Signifikanz und Modellgüte:

Intercept: Das 95% Konfidenzintervall für den Intercept liegt zwischen 74.40191 und 87.72479. Das bedeutet, dass der diastolische Blutdruck laut diesem Modell bei einer Natriumausscheidung von 0 mmol/24h mit einer 95%igen Sicherheit zwischen 74.40 und 87.72 mmHg liegt.

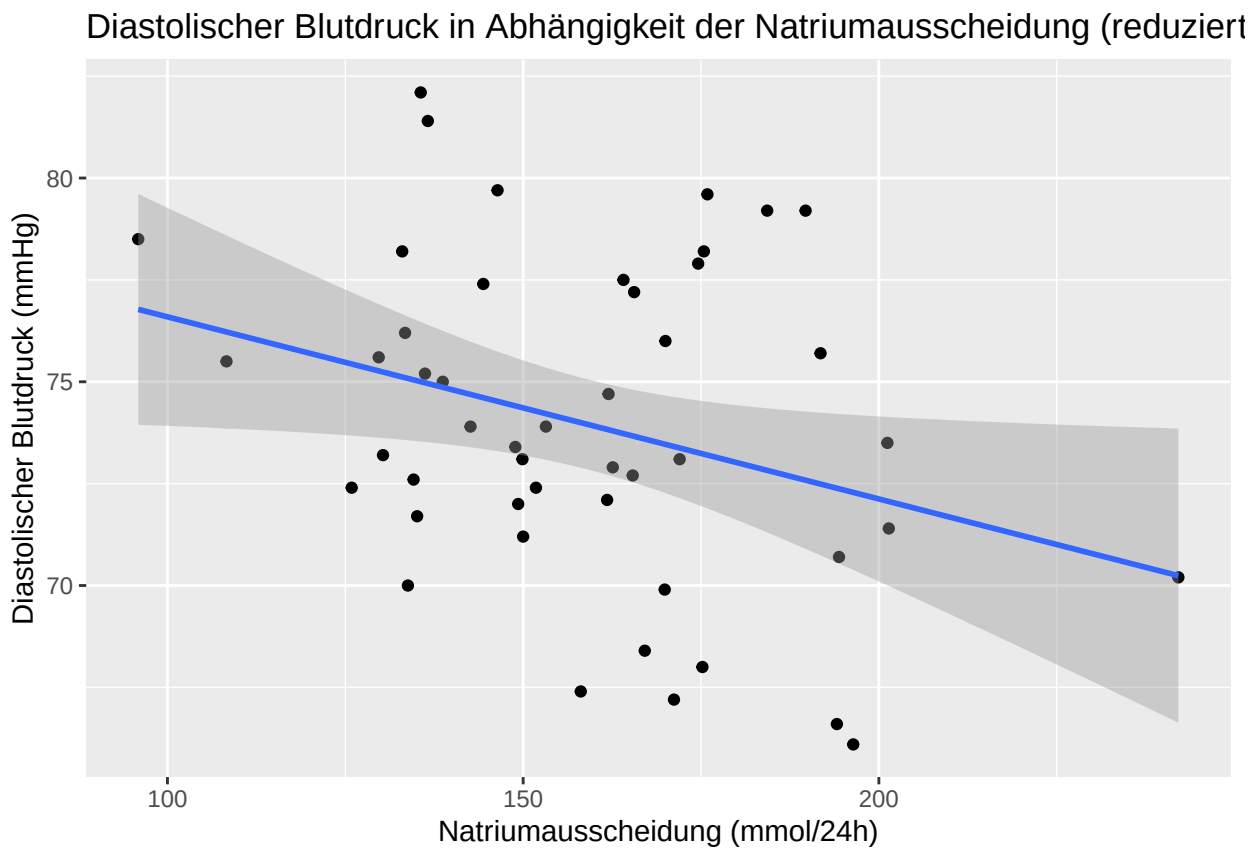
Steigung: Das 95% Konfidenzintervall für die Steigung bezüglich der Natriumausscheidung liegt zwischen -0.08602 und -0.00337. In diesem Fall ist also von einer Abnahme des diastolischen Blutdrucks um 0.00337 bis 0.08602 mmHg auszugehen, wenn die Natriumausscheidung um 1 mmol/24h steigt. Im Gegensatz zum ersten Modell zeigt die Steigung hier also einen negativen Zusammenhang zwischen Natriumausscheidung und diastolischem Blutdruck.

Signifikanz und Modellgüte: Auch bei diesem Modell ist der p-Wert für den Intercept hochsignifikant. Der p-Wert für die Steigung ist mit 0.035 zwar signifikant, aber deutlich weniger als der p-Wert für die Steigung des ersten Modells. Das Bestimmtheitsmaß R^2 beträgt hier 0.09, was bedeutet, dass das Modell nur etwa 9% der Variabilität im diastolischen Blutdruck mit der Natriumausscheidung erklären kann, wobei es beim ersten Modell noch knapp 13% waren.

iii) Grafische Darstellung der Regressionsgeraden inkl. Konfidenzintervall

```r
# regression plot for model_2
ggplot(intersalt_m2, aes(x = na, y = bp)) +
  geom_point() +
  geom_smooth(method = "lm", se = TRUE) +
  labs(title = "Diastolischer Blutdruck in Abhängigkeit der Natriumausscheidung (reduzierte Daten)",
       x = "Natriumausscheidung (mmol/24h)",
       y = "Diastolischer Blutdruck (mmHg)")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



Vergleichen Sie beide Modelle. Was können Sie beobachten?

Die signifikante Änderung der Richtung des Effekts (von positiv zu negativ) nach dem Entfernen der vier Datenpunkte mit der niedrigsten Natriumausscheidung legt nahe, dass diese Datenpunkte einen erheblichen Einfluss auf das Gesamtergebnis des Modells haben.

Ein niedrigerer Wert für R^2 im zweiten Modell könnte bedeuten, dass die vier entfernten Datenpunkte tatsächlich einen wichtigen Beitrag zur Erklärung der Variabilität des diastolischen Blutdrucks leisten.

Trotz der signifikanten Koeffizienten in beiden Modellen bleiben die R^2 Werte relativ niedrig. Somit wird deutlich, dass noch weitere, hier nicht betrachtete Variablen einen Einfluss auf den diastolischen Blutdruck haben.

[2P] b: Lesen Sie den Artikel "The (Political) Science of Salt" und vergleichen Sie damit Ihre Beobachtungen. Gibt es Faktoren die in Ihren Modellen eventuell nicht berücksichtigt wurden? Wie lautet die Schlussfolgerung - führt eine Reduktion der Salzaufnahme zu einer Blutdrucksenkung?

Zusammenfassung des Artikels "The (Political) Science of Salt": Die Intersalt-Studie, eine standardisierte, internationale Untersuchung mit einer großen Stichprobe von über 10000 Teilnehmenden, bietet einen tiefgehenden Einblick in den Zusammenhang zwischen Salzaufnahme und Blutdruck. Die Ergebnisse variieren beträchtlich zwischen verschiedenen geografischen Regionen, wobei extrem niedrige Natriumwerte in einigen isolierten Gemeinschaften mit niedrigem Blutdruck und geringem altersbedingten Anstieg des Blutdrucks korrelierten. Im Gegensatz dazu zeigten andere Studienzentren signifikante Zusammenhänge zwischen Natriumausscheidung und Blutdruckanstieg. Obwohl die Studie einen Trend zu niedrigerem Blutdruck bei geringerer Salzaufnahme andeutet, sind die Ergebnisse durch die Heterogenität der teilnehmenden Populationen und die Vielzahl der berücksichtigten Einflussgrößen wie Body-Mass-Index und Alkoholkonsum komplex.

Modellierung der Daten: Obwohl das lineare Modell signifikante Zusammenhänge zwischen Natriumaufnahme und Blutdruck gefunden hat, zeigen die Ergebnisse, dass der Zusammenhang komplex ist und möglicherweise von weiteren Faktoren beeinflusst wird, die in einfacheren Modellen nicht berücksichtigt werden.

Schlussfolgerung hinsichtlich Salzaufnahme und Blutdrucksenkung: Die Studie legt nahe, dass eine Reduktion der Salzaufnahme zu einer Senkung des Blutdrucks führen kann, besonders über längere Zeiträume und Lebensabschnitte hinweg betrachtet. Es konnte gezeigt werden, dass der Zusammenhang jedoch bei älteren Personen ausgeprägter ist als bei jüngeren. Insgesamt liefern die Ergebnisse der Studie starke Belege dafür, dass eine Reduktion der Salzaufnahme eine effektive Maßnahme zur Blutdrucksenkung darstellt, insbesondere in Bevölkerungsgruppen mit hohem Blutdruck oder erhöhtem Risiko für Herz-Kreislauf-Erkrankungen.

2 Lineare Regressionsanalyse (kategorisch) [3P]

Der Datensatz `infant.csv` enthält Information über die unterschiedliche Kindersterblichkeit zwischen den Kontinenten. Die Variable `infant` enthält die Kindersterblichkeit in Tode pro 1000 Geburten. Unterscheidet sich die Kindersterblichkeit zwischen den Kontinenten?

[2P] a: Führen Sie eine Regressionsanalyse mit Europa als Referenz durch, welche die folgenden Punkte umfasst:

i) Modellgleichung inklusive 95% Konfidenzintervall der Modellparameter
ii) Interpretation des Ergebnisses hinsichtlich Signifikanz
iii) Beurteilung der Modellgüte und Residuenanalyse

```r
# Import the data
data <- read.csv2("infant.csv")

# Convert the region variable to a factor
data$region <- as.factor(data$region)

# Set "Europe" as the reference level for the region variable
data$region <- relevel(data$region, ref = "Europe")

# Fit a linear model with infant as a function of region
model <- lm(infant ~ region, data = data)

# Print the model equation and 95% confidence intervals
summary(model)
```

```
##
## Call:
```

```
## lm(formula = infant ~ region, data = data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -87.29 -37.52  -5.76  16.91 553.83
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)       19.26      18.75   1.027  0.30696
## regionAfrica     123.04      23.19   5.306 7.07e-07 ***
## regionAmericas    35.87      25.28   1.419  0.15918
## regionAsia        76.91      24.20   3.178  0.00199 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 79.54 on 97 degrees of freedom
##   (4 Beobachtungen als fehlend gelöscht)
## Multiple R-squared:  0.2556, Adjusted R-squared:  0.2326
## F-statistic:  11.1 on 3 and 97 DF,  p-value: 2.494e-06
```

```
confint(model, level=0.95)
```

```
##                    2.5 %    97.5 %
## (Intercept)    -17.95542  56.46653
## regionAfrica    77.01704 169.05421
## regionAmericas -14.30810  86.04244
## regionAsia      28.87565 124.95398
```

Der p-Wert der F-Stastik liegt unter 5%&, was bedeuet, dass mindestens ein Koeffizient signifikant ist.

Der p-Wert der erwarteten Kindersterblichkeit für Europa "(Intercept)" liegt mit 31% über dem Signifikanzniveau von 5%. Dementsprechend ist 0 im Konfidenzintervall eingeschlossen. D.h. es gibt keinen signfikanten Unterschied zu Kindersterblichkeit gleich 0.

Der p-Wert der erwarteten Differenz in der Kindersterblichkeit zwischen Europa und Amerika "regionAmericas" liegt auch über 5%. D.h. es gibt keinen signifikanten Unterschied in der Kindersterblichkeit zwischen Euorpa und Amerika.

Die p-Werte der erwareten Differenzen in der Kindersterblichkeit zwischen den übrigen Regionen und Europa liegen unter 5%. Hier gibt es also, basierend auf den Daten dieser Stichprobe einen signfikanten Unterschied in der Kindersterblichkeit zu Europa.

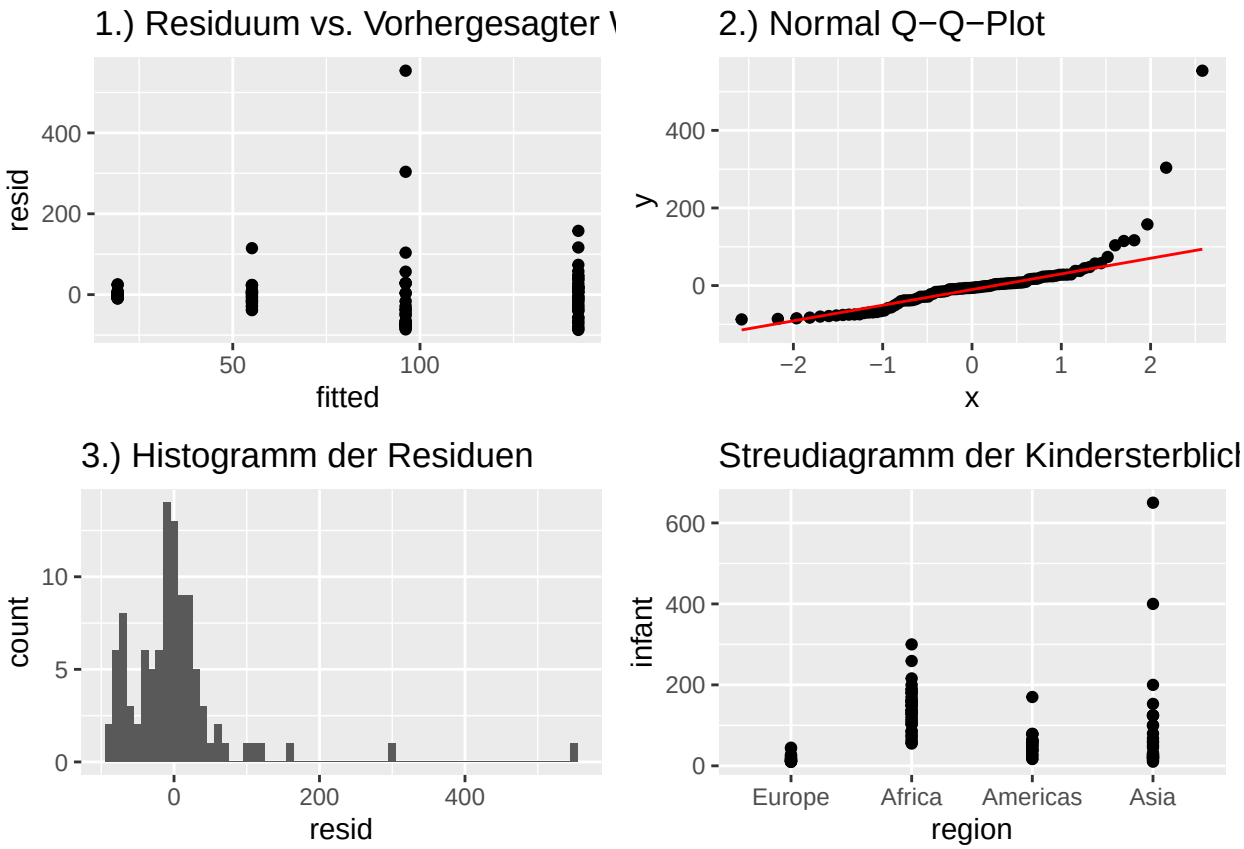
Der adjusted R^2 beträgt 23%. Somit können 23% der Varianz in der (relativen) Kindersterblichkeit durch eine Änderung in den (binären) Prädiktor-Variablen bestimmt werden.

```
# Interpret the significance of the results

# The p-value for each term tests the null hypothesis that the coefficient is equal to zero (no effect)
# A low p-value (< 0.05) indicates that you can reject the null hypothesis.

# Assess the quality of the model and perform a residual analysis
#par(mfrow=c(2,2))
#plot(model)

# Convert model's data to a data frame
```

```r
model_data <- data[complete.cases(data), ]  # remove rows with missing values
df <- data.frame(resid = resid(model), fitted = fitted(model),
                 region = model_data$region, infant = model_data$infant)

# Load ggplot2
library(ggplot2)

# plots of residual analysis
# 1. Residuals vs Fitted
p1 <- ggplot(df, aes(fitted, resid)) +
  geom_point() +
  #geom_smooth(se = FALSE) +
  ggtitle("1.) Residuum vs. Vorhergesagter Wert")

# 2. Normal Q-Q
p2 <- ggplot(df, aes(sample = resid)) +
  stat_qq() +
  stat_qq_line(colour = "red") +
  ggtitle("2.) Normal Q-Q-Plot")

# 3. Histogram of residuals
p3 <- ggplot(df, aes(x=resid)) +
  geom_histogram(binwidth = 10) +
  ggtitle("3.) Histogramm der Residuen")

# 4. Scatter Plot
p4 <- ggplot(df, aes(x=region, y=infant)) +
  geom_jitter(width=0.0) +
  ggtitle("Streudiagramm der Kindersterblichkeit nach Region")

# Load gridExtra
library(gridExtra)
```

```
## Warning: Paket 'gridExtra' wurde unter R Version 4.3.2 erstellt
```

```r
# Arrange the plots in a 4x1 grid
grid.arrange(p1, p2, p3, p4, nrow = 2)
```



Die Region Asien zeigt 3 Ausreißer nach oben, die besonders ins Auge stechen. Wenn man sie ignoriert, kann man eine annährende Normalverteilung der Residuen beispielsweise im Q-Q-Plot und Histogramm erkennen.

Links oben: Die Residuen streuen normalverteilt um den Nullpunkt, allerdings scheint die Standardabweichung der Residuen nach Region unterschiedlich zu sein.

Insgesamt entsteht aber der Eindruck, dass eine Normalverteilung der Residuen vorliegt und ihre Varianz in etwa gleich bleibt (d.h. Homoskedastizität gegeben ist).

[1P] b: Wie hoch ist die Kindersterblichkeit in Europa und wie hoch in Afrika (inkl. Unsicherheit)?

```
# Get the model coefficients and their confidence intervals
coef <- coef(model)
conf_int <- confint(model)
names(coef(model))
```

```
## [1] "(Intercept)"    "regionAfrica"   "regionAmericas" "regionAsia"
```

```
# Calculate the infant mortality rate for Europe
europe_coef <- coef["(Intercept)"]
europe_conf_int <- conf_int["(Intercept)", ]

# Calculate the infant mortality rate for Africa
africa_coef <- coef["regionAfrica"]
africa_conf_int <- conf_int["regionAfrica", ]
```

```r
# Print the results
cat("Europa:\n")
```

```
## Europa:
```

```r
cat("Anzahl: ", europe_coef, "\n")
```

```
## Anzahl:  19.25556
```

```r
cat("95% KI: ", europe_conf_int, "\n")
```

```
## 95% KI:  -17.95542 56.46653
```

```r
cat("Afrika:\n")
```

```
## Afrika:
```

```r
cat("Anzahl: ", africa_coef+europe_coef, "\n")
```

```
## Anzahl:  142.2912
```

```r
cat("95% KI: ", africa_conf_int+europe_coef, "\n")
```

```
## 95% KI:  96.27259 188.3098
```

3 Regressionsanalyse [3P]

Die Daten `wtloss.xlsx` enthalten den Gewichtsverlauf eines adipösen Patienten im Zuge einer Diät. Sie als betreuender Mediziner und passionierter Freizeit Data Scientist möchten ein geeignetes Regressionsmodell erstellen, um den Verlauf der Diät besser steuern zu können. Das ideale Zielgewicht bezogen auf die Größe des Patienten wäre bei 80 kg. Importieren Sie den Datensatz mit Hilfe der `read_excel()` Funktion aus dem `library(readxl)` Paket.

[**2P**] **a:** Die Regressionsanalyse sollte folgende Punkte inkludieren:

i) Modellgleichung inklusive 95% Konfidenzintervall der Modellparameter
ii) Interpretation des Ergebnisses hinsichtlich Signifikanz
iii) Beurteilung der Modellgüte und Residuenanalyse
iv) Grafische Darstellung der Regressionsgeraden inkl. Konfidenz-und Vorhersageinterval

```r
library(readxl)
# import the data
data <- read_excel("wtloss.xlsx")

# fit a linear model
model <- lm(Weight ~ Days, data = data)

# print the model equation and 95% confidence intervals
summary(model)
```

```
##
## Call:
## lm(formula = Weight ~ Days, data = data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -4.522 -2.891 -1.331  3.096  7.460
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 176.890244   0.987644   179.1   <2e-16 ***
## Days         -0.290735   0.007125   -40.8   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.639 on 50 degrees of freedom
## Multiple R-squared:  0.9708, Adjusted R-squared:  0.9703
## F-statistic:  1665 on 1 and 50 DF,  p-value: < 2.2e-16
```

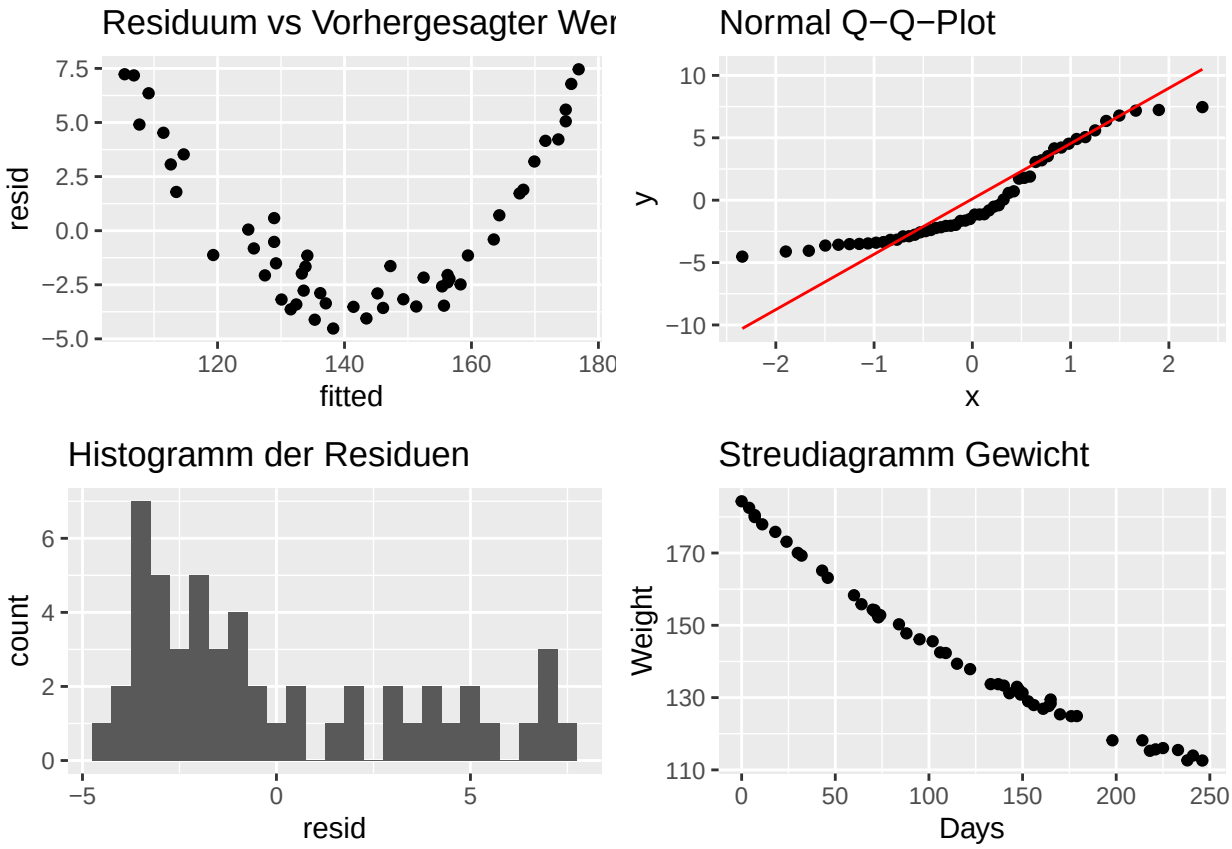
```
confint(model, level=0.95)
```

```
##                  2.5 %      97.5 %
## (Intercept) 174.906503 178.8739857
## Days         -0.305046  -0.2764236
```

Laut der p-Werte (p<0.01) und zugehörigen Konfidenzintervalle liefert das lineare Regressionsmodell hochsignfikante Ergebnisse für die Konstante und den Koeffizienten in Weight = (Intercept) + Days * x

```
# Convert model's data to a data frame
model_data <- data[complete.cases(data), ]  # remove rows with missing values
df <- data.frame(resid = resid(model), fitted = fitted(model),
                 Weight = model_data$Weight, Days = model_data$Days)

# Load ggplot2
library(ggplot2)

# Create the four diagnostic plots
# 1. Residuals vs Fitted
p1 <- ggplot(df, aes(fitted, resid)) +
  geom_point() +
  #geom_smooth(se = FALSE) +
  ggtitle("Residuum vs Vorhergesagter Wert")

# 2. Normal Q-Q
p2 <- ggplot(df, aes(sample = resid)) +
  stat_qq() +
  stat_qq_line(colour = "red") +
  ggtitle("Normal Q-Q-Plot")

# 3. Histogram of residuals
p3 <- ggplot(df, aes(x=resid)) +
  geom_histogram(binwidth = 0.5) +
  ggtitle("Histogramm der Residuen")
```

```
# 4. Scatter Plot
p4 <- ggplot(df, aes(x=Days, y=Weight)) +
  geom_jitter(width=0.0) +
  ggtitle("Streudiagramm Gewicht")

# Load gridExtra
library(gridExtra)

# Arrange the plots in a 4x1 grid
grid.arrange(p1, p2, p3, p4, nrow = 2)
```



Die Grafiken zeigen, dass die Residuen nicht normalverteilt sind. Sie zeigen ein Muster. D.h. das linerare Modell ist nicht sehr gut geeignet für den Datensatz. Zur weiteren Veranschaulichung erfolgt eine Darstellung des linearen Modells inklusive Konfidenz- und Vorhersage-Intervall.

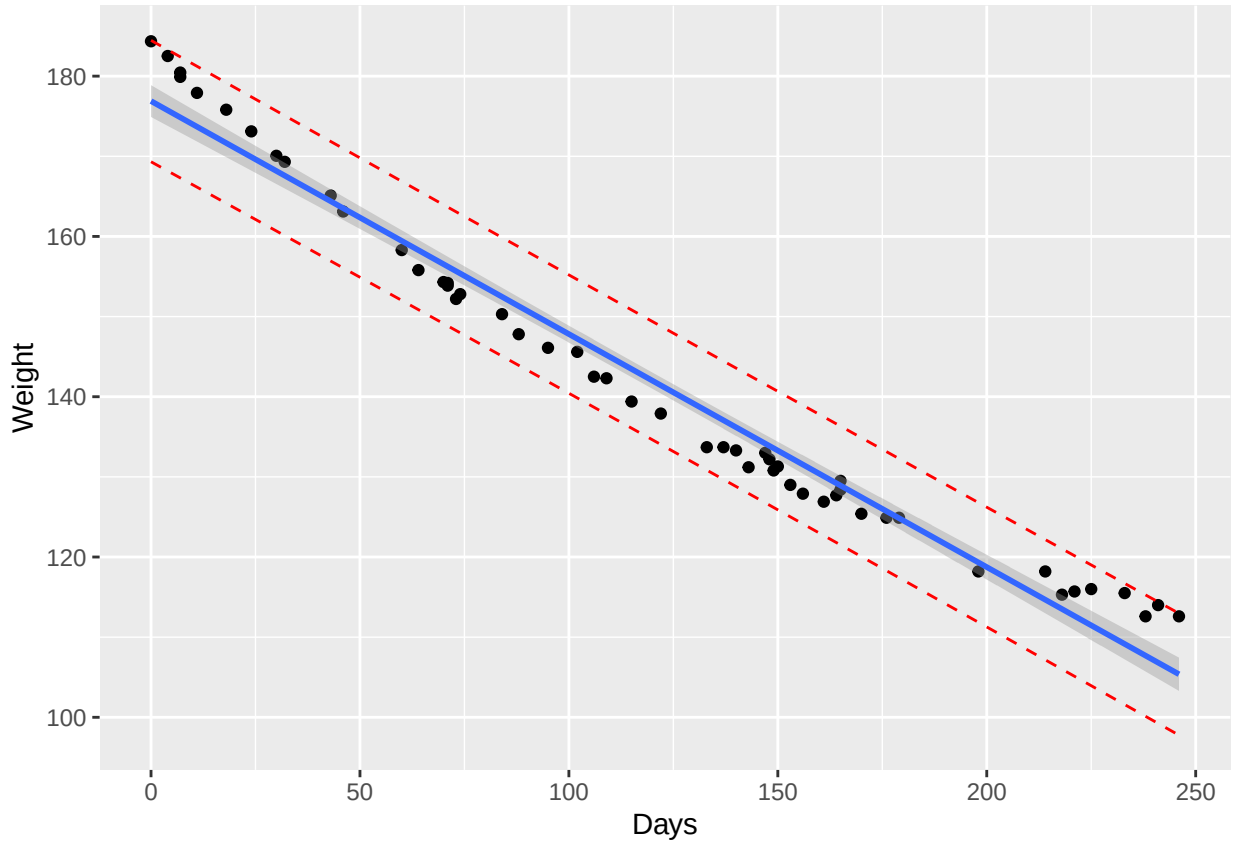
```
# 1. Add predictions
pred.int <- predict(model, interval = "prediction")
```

```
## Warning in predict.lm(model, interval = "prediction"): Vorhersagen auf aktuellen Daten beziehen sich
```

```
mydata <- cbind(data, pred.int)
# 2. Regression line + confidence intervals
library("ggplot2")
p <- ggplot(mydata, aes(x=Days, y=Weight)) +
  geom_point() +
```

```
  stat_smooth(method = lm)
# 3. Add prediction intervals
p + geom_line(aes(y = lwr), color = "red", linetype = "dashed")+
    geom_line(aes(y = upr), color = "red", linetype = "dashed")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



Um ein besseres Regressionsmodell zu finden, wird R^2 and RMSE für das lineare Modell, ein logarithmisches, ein quadratisches, kubisches und ein Polynom 5. Grades berechnet.

```
# Load necessary library
library(caret)
```

```
## Warning: Paket 'caret' wurde unter R Version 4.3.3 erstellt
```

```
## Lade nötiges Paket: lattice
```

```
# Drop the first row to avoid singularity in logarithmic model
new_data <- data[-1, ]

# Fit different types of models
linear_model <- lm(Weight ~ Days, data = data)
logarithmic_model <- lm(Weight ~ log(Days), data = new_data)
quadratic_model <- lm(Weight ~ poly(Days, 2), data = data)
```

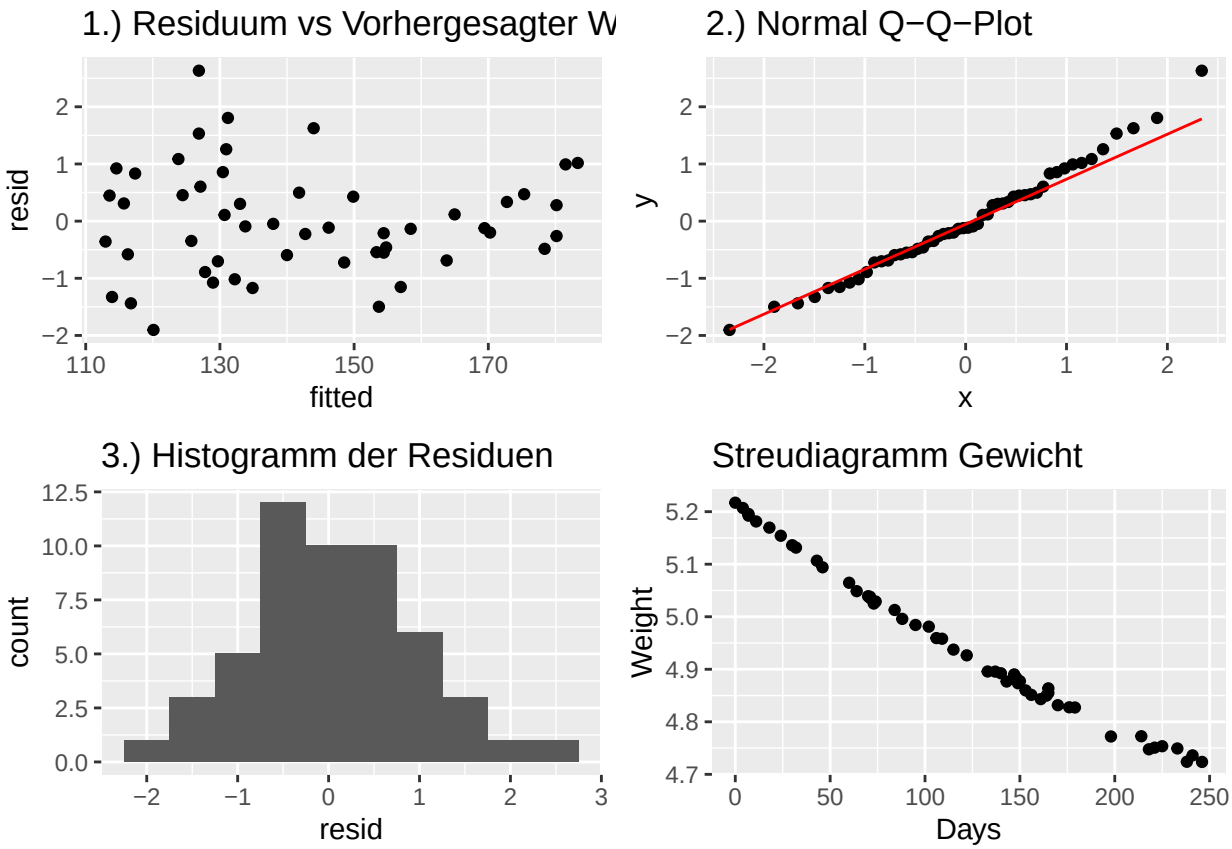
```r
cubic_model <- lm(Weight ~ poly(Days, 3), data = data)
fifth_order_model <- lm(Weight ~ poly(Days, 5), data = data)

# List of models
models <- list(linear_model, logarithmic_model, quadratic_model, cubic_model, fifth_order_model)

# Calculate R^2 and RMSE for each model
for (model in models) {
  # output title of model
  cat(paste("Model: ", deparse(model$call), "\n"))
  print(summary(model)$r.squared) # R^2
  print(summary(model)$adj.r.squared) # Adjusted R^2
  print(sqrt(mean(resid(model)^2))) # RMSE
  cat("\n")
}
```

```
## Model:  lm(formula = Weight ~ Days, data = data)
## [1] 0.9708454
## [1] 0.9702623
## [1] 3.568399
##
## Model:  lm(formula = Weight ~ log(Days), data = new_data)
## [1] 0.8902471
## [1] 0.8880072
## [1] 6.707146
##
## Model:  lm(formula = Weight ~ poly(Days, 2), data = data)
## [1] 0.9980994
## [1] 0.9980218
## [1] 0.911098
##
## Model:  lm(formula = Weight ~ poly(Days, 3), data = data)
## [1] 0.9982725
## [1] 0.9981645
## [1] 0.8686237
##
## Model:  lm(formula = Weight ~ poly(Days, 5), data = data)
## [1] 0.9983244
## [1] 0.9981422
## [1] 0.8554825
```

Das quadratische Modell stellt eine hinreichende Verbesserung im Vergleich zum linearen dar. Polynome höhrer Ordnung verbessern R^2 und RMSE nur geringfügig. Um die Güte des quadratischen Modells zu beurteilen, erfolgt die Residualanalyse.

```r
# Calculate the logarithm of a column and overwrite the old values
log_data = data
log_data$Weight <- log(data$Weight)

# Convert model's data to a data frame
model_data <- data[complete.cases(data), ]  # remove rows with missing values
df <- data.frame(resid = resid(models[[3]]), fitted = fitted(models[[3]]),
                 Weight = log_data$Weight, Days = log_data$Days)
```

```r
# Load ggplot2
library(ggplot2)

# Create the four diagnostic plots
# 1. Residuals vs Fitted
p1 <- ggplot(df, aes(fitted, resid)) +
  geom_point() +
  #geom_smooth(se = FALSE) +
  ggtitle("1.) Residuum vs Vorhergesagter Wert")

# 2. Normal Q-Q
p2 <- ggplot(df, aes(sample = resid)) +
  stat_qq() +
  stat_qq_line(colour = "red") +
  ggtitle("2.) Normal Q-Q-Plot")

# 3. Histogram of residuals
p3 <- ggplot(df, aes(x=resid)) +
  geom_histogram(binwidth = 0.5) +
  ggtitle("3.) Histogramm der Residuen")

# 4. Scatter Plot
p4 <- ggplot(df, aes(x=Days, y=Weight)) +
  geom_jitter(width=0.0) +
  ggtitle("Streudiagramm Gewicht")

# Load gridExtra
library(gridExtra)

# Arrange the plots in a 4x1 grid
grid.arrange(p1, p2, p3, p4, nrow = 2)
```



Die Grafiken zeigen, dass die Residuen normalverteilt sind und kein eindeutiges Muster erkennen lassen. Das quadratische Modell ist daher hinreichend gut geeignet zur Modellierung dieser Daten.

[**1P**] **b:** Welches Gewicht hat der Patient nach 30 Tagen bzw. nach 200 Tagen Diät?

```
# Define new data frame with the single x value
new_data <- data.frame(Days = c(30,200))

# Calculate predicted y value with confidence interval
confidence_interval <- predict(models[[1]], newdata = new_data, interval = "confidence")

# Calculate predicted y value with prediction interval
prediction_interval <- predict(models[[1]], newdata = new_data, interval = "prediction")

# Print the results
print(confidence_interval)
```

```
##        fit      lwr      upr
## 1 168.1682 166.5387 169.7977
## 2 118.7433 117.2051 120.2815
```

```
print(prediction_interval)
```

```
##        fit      lwr      upr
## 1 168.1682 160.6795 175.6569
## 2 118.7433 111.2739 126.2127
```

Laut dem linearen Modell beträgt das Gewicht des Patienten nach 30 Tagen 168.17 kg (Konfidenzintervall [166.54,169.80], Vorhersageintervall [160.68,175.66]) und nach 200 Tagen 118.74 kg (Konfidenzintervall [117.21,120.28], Vorhersageintervall [111.27,126.21]).

```r
# Define new data frame with the single x value
new_data <- data.frame(Days = c(30,200))

# Calculate predicted y value with confidence interval
confidence_interval <- predict(models[[3]], newdata = new_data, interval = "confidence")

# Calculate predicted y value with prediction interval
prediction_interval <- predict(models[[3]], newdata = new_data, interval = "prediction")

# Print the results
print(confidence_interval)
```

```
##        fit      lwr      upr
## 1 170.2600 169.8106 170.7094
## 2 119.7419 119.3378 120.1460
```

```r
print(prediction_interval)
```

```
##        fit     lwr      upr
## 1 170.2600 168.321 172.1989
## 2 119.7419 117.813 121.6708
```

Laut dem quadratischen Modell beträgt das Gewicht des Patienten nach 30 Tagen 170.26 kg (Konfidenzintervall [169.81,170.71], Vorhersageintervall [168.32,172.20]) und nach 200 Tagen 119.74 kg (Konfidenzintervall [119.34,120.15], Vorhersageintervall [117.81,121.67]).

4 Multiple Regressionsanalyse [3P]

Die Framingham-Herz-Studie war ein Wendepunkt bei der Identifizierung von Risikofaktoren für koronare Herzkrankheiten und ist eine der wichtigsten epidemiologischen Studien die je durchgeführt wurden. Ein großer Teil unseres heutigen Verständnisses von Herz-Kreislauf-Erkrankungen ist auf diese Studie zurückzuführen. Der Datensatz `Framingham.sav` enthält Varibalen hinsichtlich Demographie, Verhaltensweise, Krankengeschichte und Risikofaktoren. Finden Sie ein geeignetes Modell, dass den systolischen Blutdruck (`sysbp`) beschreibt. Vermeiden Sie nicht relevante bzw. redundante Variablen (z.B. "Incident" Variablen). Achten Sie auf Ausreißer und fehlende Daten (`NaN, NA's`).

[2P] a: Die Regressionsanalyse sollte folgende Punkte inkludieren:

i) Modellgleichung inklusive 95% Konfidenzintervall der Modellparameter
ii) Interpretation des Ergebnisses hinsichtlich Signifikanz
iii) Beurteilung der Modellgüte und Residuenanalyse

Die folgende Sektion ist nur eine experimentelle Vorarbeit und kein Teil der Antwort.

```r
# Import the data
# Load necessary library
library(haven)

# Read in .sav file
data <- read_spss("Framingham.sav")

# Fit a multivariate linear regression model with all variables
max_model <- lm(sysbp ~ ., data = data)

summary(max_model)
```

```
##
## Call:
## lm(formula = sysbp ~ ., data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -45.524  -7.783  -1.043   6.384  83.556
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.062e+01  2.668e+00  -3.982 6.95e-05 ***
## randid      -3.301e-08  6.746e-08  -0.489 0.624602
## sex          3.935e+00  4.394e-01   8.955  < 2e-16 ***
## age          4.520e-01  2.747e-02  16.455  < 2e-16 ***
## educ        -6.646e-01  1.968e-01  -3.377 0.000739 ***
## cursmoke     4.036e-01  6.320e-01   0.639 0.523095
## cigpday      2.869e-03  2.716e-02   0.106 0.915868
## bpmeds       1.016e+01  1.127e+00   9.017  < 2e-16 ***
## totchol      3.612e-03  4.731e-03   0.764 0.445177
## diabp        1.219e+00  1.970e-02  61.885  < 2e-16 ***
## bmi          1.571e-02  5.373e-02   0.292 0.769987
## diabetes     2.730e-01  1.507e+00   0.181 0.856260
## heartrte     5.233e-02  1.701e-02   3.076 0.002112 **
## glucose      4.262e-02  1.028e-02   4.148 3.43e-05 ***
## angina      -6.561e-01  5.869e-01  -1.118 0.263659
## hospmi       4.539e-01  1.011e+00   0.449 0.653453
## mi_fchd      2.150e+00  1.122e+00   1.917 0.055323 .
## stroke       1.345e+00  9.491e-01   1.417 0.156481
## cvd         -7.537e-01  8.938e-01  -0.843 0.399123
## hyperten     5.988e+00  5.015e-01  11.941  < 2e-16 ***
## death        3.302e+00  5.077e-01   6.504 8.84e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.15 on 3805 degrees of freedom
##   (608 Beobachtungen als fehlend gelöscht)
## Multiple R-squared:  0.7089, Adjusted R-squared:  0.7074
## F-statistic: 463.3 on 20 and 3805 DF,  p-value: < 2.2e-16
```

```r
# Calculate confidence intervals for each variable
#conf_intervals <- confint(model,level=0.95)
```

```r
#print(conf_intervals)

# Extract coefficients
coefficients <- coef(max_model)

# Calculate confidence intervals
conf_intervals <- confint(max_model, level = 0.95)

# Print coefficients and their confidence intervals
for (i in 1:length(coefficients)) {
  cat(names(coefficients)[i], ": ", coefficients[i], ", CI: [", conf_intervals[i, 1], ", ", conf_interva
}
```

```
## (Intercept) :  -10.62419 , CI: [ -15.85471 ,  -5.393663 ]
## randid :  -3.301337e-08 , CI: [ -1.652742e-07 ,  9.924746e-08 ]
## sex :  3.934823 , CI: [ 3.073317 ,  4.796328 ]
## age :  0.452006 , CI: [ 0.3981518 ,  0.5058602 ]
## educ :  -0.6646026 , CI: [ -1.050401 ,  -0.2788038 ]
## cursmoke :  0.4036274 , CI: [ -0.8354829 ,  1.642738 ]
## cigpday :  0.002868857 , CI: [ -0.0503714 ,  0.05610911 ]
## bpmeds :  10.16483 , CI: [ 7.954712 ,  12.37494 ]
## totchol :  0.003612468 , CI: [ -0.005663249 ,  0.01288819 ]
## diabp :  1.219386 , CI: [ 1.180755 ,  1.258018 ]
## bmi :  0.01571155 , CI: [ -0.08963225 ,  0.1210554 ]
## diabetes :  0.2729894 , CI: [ -2.681584 ,  3.227563 ]
## heartrte :  0.05232853 , CI: [ 0.01897606 ,  0.085681 ]
## glucose :  0.04262199 , CI: [ 0.02247679 ,  0.06276718 ]
## angina :  -0.656128 , CI: [ -1.806802 ,  0.4945459 ]
## hospmi :  0.4538921 , CI: [ -1.52803 ,  2.435814 ]
## mi_fchd :  2.150185 , CI: [ -0.0489783 ,  4.349348 ]
## stroke :  1.345169 , CI: [ -0.515655 ,  3.205993 ]
## cvd :  -0.7537323 , CI: [ -2.50612 ,  0.998655 ]
## hyperten :  5.987937 , CI: [ 5.004764 ,  6.971111 ]
## death :  3.302001 , CI: [ 2.306621 ,  4.297381 ]
```

Für das multiple Regressionsmodell wurden folgende Variablen mit eingeschlossen: Geschlecht, Alter, Bildungsstand, Raucherstatus, Zigaretten pro Tag, die Einnahme blutdrucksenkender Medikamente, Cholesterol, diastolischer Blutdruck, Body Mass Index, Diagnose Diabetes, Herzrate und Glukosewert. Davon sind Geschlecht, Bildungsstand, Raucherstatus, die Einnahme blutdrucksenkender Medikamente und die Diagnose Diabetes kategorialen Variablen.

```r
# Import the data
# Load necessary library
library(haven)

# Read in .sav file
data <- read_spss("Framingham.sav")

# Convert the educ variable to a factor
data$educ <- as.factor(data$educ)

# Set "1" as the reference level for the educ variable
data$educ <- relevel(data$educ, ref = "1")
```

```r
# Convert the sex variable to a factor
data$sex <- as.factor(data$sex)

# Set "1" as the reference level for the educ variable
data$sex <- relevel(data$sex, ref = "1")

# Convert the cursmoke variable to a factor
data$cursmoke <- as.factor(data$cursmoke)

# Set "0" as the reference level for the cursmoke variable
data$cursmoke <- relevel(data$cursmoke, ref = "0")

# Convert the bpmeds variable to a factor
data$bpmeds <- as.factor(data$bpmeds)

# Set "0" as the reference level for the bpmeds variable
data$bpmeds <- relevel(data$bpmeds, ref = "0")

# Convert the diabetes variable to a factor
data$diabetes <- as.factor(data$diabetes)

# Set "0" as the reference level for the diabetes variable
data$diabetes <- relevel(data$diabetes, ref = "0")

model <- lm(sysbp ~ sex + age + educ + cursmoke + cigpday + bpmeds + totchol + diabp + bmi + diabetes +

summary(model)
```

```
##
## Call:
## lm(formula = sysbp ~ sex + age + educ + cursmoke + cigpday +
##     bpmeds + totchol + diabp + bmi + diabetes + heartrte + glucose,
##     data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -45.570  -8.033  -1.175   6.507  82.972
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -17.760540   2.493534  -7.123 1.26e-12 ***
## sex2          3.489549   0.443050   7.876 4.37e-15 ***
## age           0.566722   0.025888  21.891  < 2e-16 ***
## educ2        -0.533886   0.502344  -1.063 0.287944
## educ3        -1.840224   0.597055  -3.082 0.002070 **
## educ4        -2.546924   0.677995  -3.757 0.000175 ***
## cursmoke1     0.263111   0.646714   0.407 0.684146
## cigpday       0.019903   0.027807   0.716 0.474198
## bpmeds1      10.958899   1.148423   9.543  < 2e-16 ***
## totchol       0.005496   0.004801   1.145 0.252410
## diabp         1.312215   0.018950  69.246  < 2e-16 ***
## bmi           0.069886   0.055025   1.270 0.204136
## diabetes1     1.095047   1.538794   0.712 0.476739
```

```
## heartrte      0.058471   0.017402   3.360 0.000787 ***
## glucose       0.048529   0.010519   4.613 4.09e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.47 on 3811 degrees of freedom
##   (608 Beobachtungen als fehlend gelöscht)
## Multiple R-squared:  0.693,  Adjusted R-squared:  0.6919
## F-statistic: 614.5 on 14 and 3811 DF,  p-value: < 2.2e-16
```

```r
# Extract coefficients
coefficients <- coef(model)

# Calculate confidence intervals
conf_intervals <- confint(model, level = 0.95)

# Print coefficients and their confidence intervals
for (i in 1:length(coefficients)) {
  cat(names(coefficients)[i], ": ", coefficients[i], ", CI: [", conf_intervals[i, 1], ", ", conf_interva
}
```

```
## (Intercept) :  -17.76054 , CI: [ -22.64933 ,  -12.87175 ]
## sex2 :  3.489549 , CI: [ 2.620911 ,  4.358187 ]
## age :  0.5667222 , CI: [ 0.5159667 ,  0.6174778 ]
## educ2 :  -0.5338864 , CI: [ -1.518775 ,  0.4510022 ]
## educ3 :  -1.840224 , CI: [ -3.010801 ,  -0.6696461 ]
## educ4 :  -2.546924 , CI: [ -3.876191 ,  -1.217656 ]
## cursmoke1 :  0.2631113 , CI: [ -1.004827 ,  1.53105 ]
## cigpday :  0.01990275 , CI: [ -0.03461599 ,  0.07442148 ]
## bpmeds1 :  10.9589 , CI: [ 8.707316 ,  13.21048 ]
## totchol :  0.005495798 , CI: [ -0.003917228 ,  0.01490882 ]
## diabp :  1.312215 , CI: [ 1.275062 ,  1.349369 ]
## bmi :  0.06988565 , CI: [ -0.03799537 ,  0.1777667 ]
## diabetes1 :  1.095047 , CI: [ -1.921892 ,  4.111986 ]
## heartrte :  0.05847142 , CI: [ 0.02435292 ,  0.09258992 ]
## glucose :  0.04852854 , CI: [ 0.0279048 ,  0.06915229 ]
```

6 von 15 Modellparametern sind nicht signifikant. Diese sind educ2 (High School) cursmoke1 (positiver Raucherstatus), cigpday (Zigaretten pro Tag), totchol (Cholesterol), bmi (Body Mass Index) und diabetes1 (positive Diabetes Diagnose).

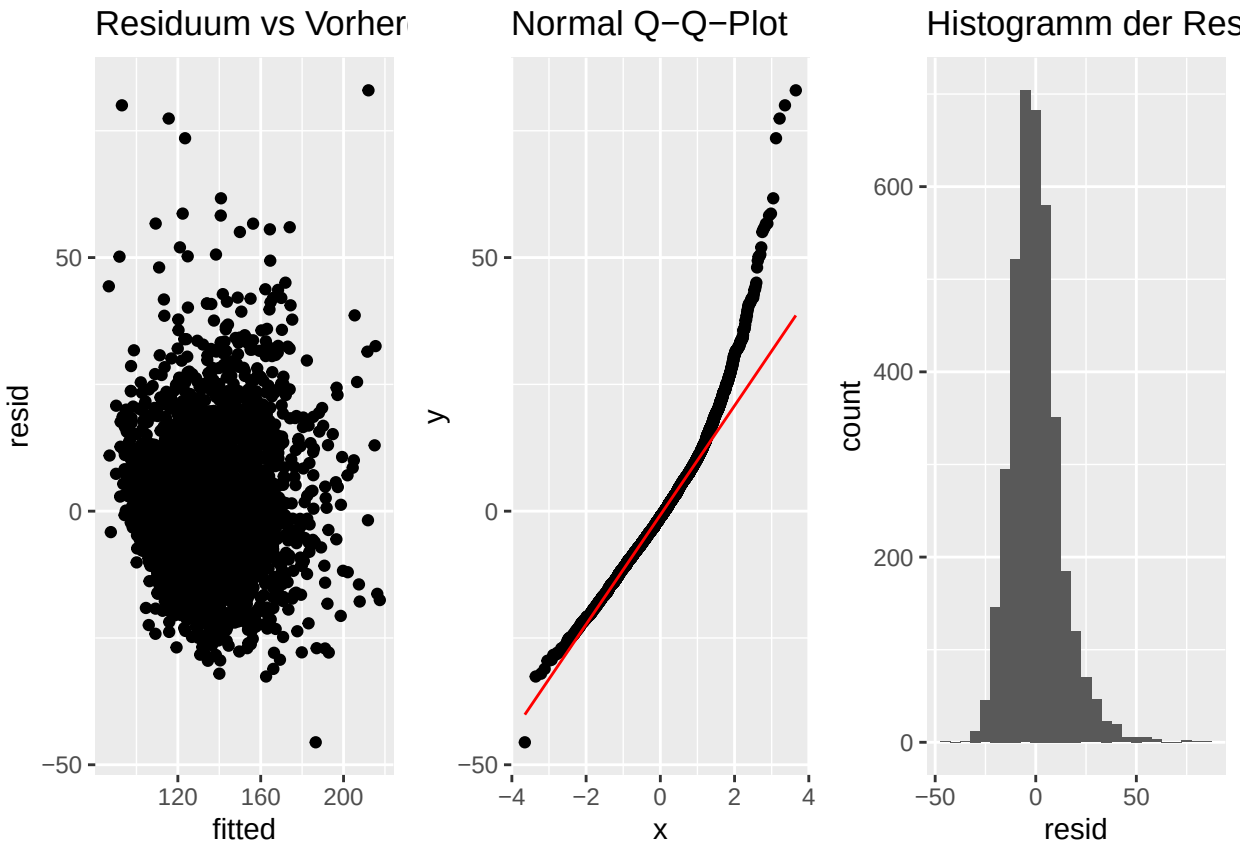
Nicht erklärbar ist der positive Wert von bpmeds1 (vorhandene Einnahme von Blutdrucksenkern). Dieses hochsignifikatne Ergebnis besagt, dass Blutdrucksenker den systolischen Blutdruck erhöhen!

```r
# Convert model's data to a data frame
model_data <- data[complete.cases(data), ]  # remove rows with missing values
df <- data.frame(resid = resid(model), fitted = fitted(model),
                 sysbp = model_data$sysbp, sex = model_data$sex, age = model_data$age, educ = model_data

# Load ggplot2
library(ggplot2)

# Create a function to generate scatter plots
scatter_plot <- function(df, x_var, y_var = "sysbp", title) {
```

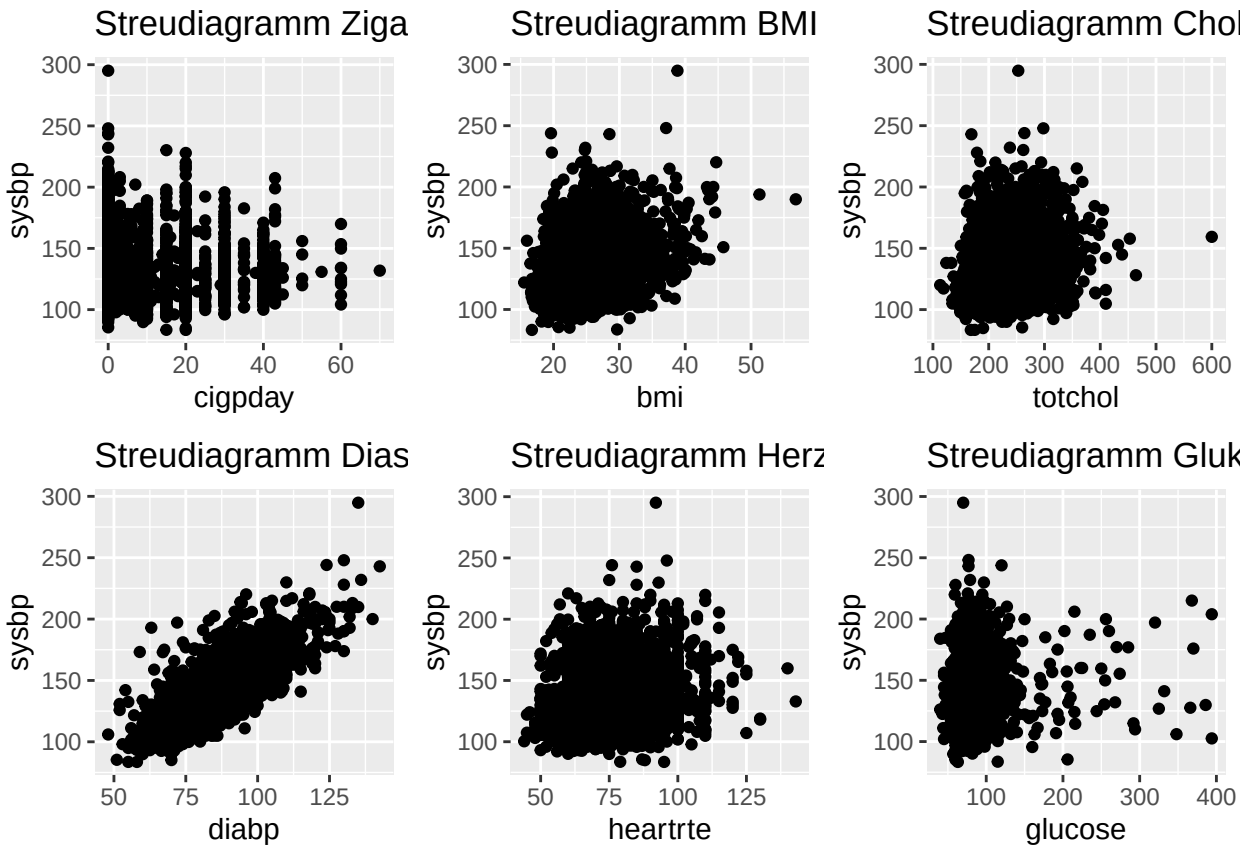
```
  ggplot(df, aes_string(x = x_var, y = y_var)) +
    geom_jitter(width = 0.0) +
    ggtitle(title)
}

# Create the four diagnostic plots
p1 <- ggplot(df, aes(fitted, resid)) +
  geom_point() +
  ggtitle("Residuum vs Vorhergesagter Wert")

p2 <- ggplot(df, aes(sample = resid)) +
  stat_qq() +
  stat_qq_line(colour = "red") +
  ggtitle("Normal Q-Q-Plot")

p3 <- ggplot(df, aes(x=resid)) +
  geom_histogram(binwidth = 5) +
  ggtitle("Histogramm der Residuen")

# Create scatter plots using the function
p4 <- scatter_plot(df, "sex", title = "Streudiagramm Geschlecht")
```

```
## Warning: `aes_string()` was deprecated in ggplot2 3.0.0.
## i Please use tidy evaluation idioms with `aes()`.
## i See also `vignette("ggplot2-in-packages")` for more information.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

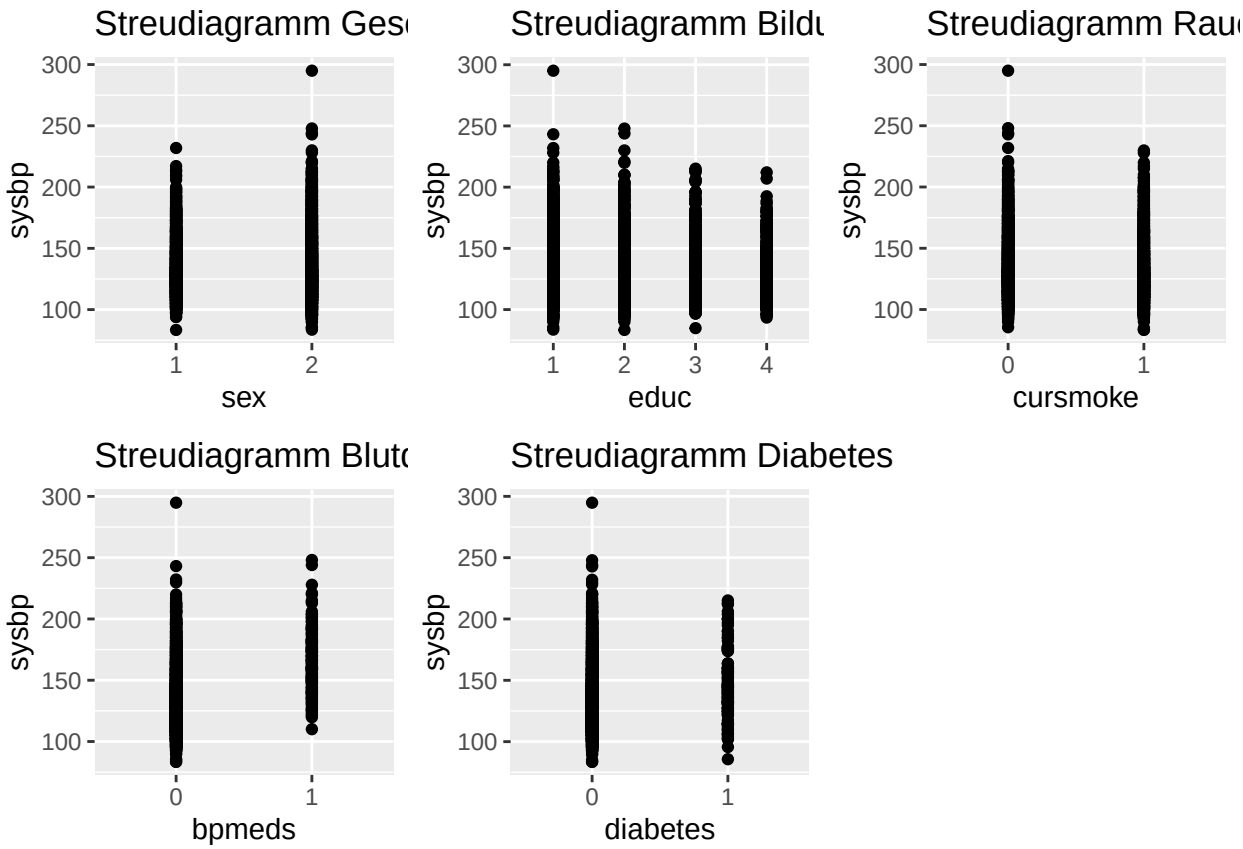
```
p5 <- scatter_plot(df, "educ", title = "Streudiagramm Bildung")
p6 <- scatter_plot(df, "cursmoke", title = "Streudiagramm Raucher")
p7 <- scatter_plot(df, "cigpday", title = "Streudiagramm Zigaretten")
p8 <- scatter_plot(df, "bmi", title = "Streudiagramm BMI")
p9 <- scatter_plot(df, "bpmeds", title = "Streudiagramm Blutdrucksenker")
p10 <- scatter_plot(df, "totchol", title = "Streudiagramm Cholesterol")
p11 <- scatter_plot(df, "diabp", title = "Streudiagramm Diast. Blutdruck")
p12 <- scatter_plot(df, "diabetes", title = "Streudiagramm Diabetes")
p13 <- scatter_plot(df, "heartrte", title = "Streudiagramm Herzrate")
p14 <- scatter_plot(df, "glucose", title = "Streudiagramm Glukose")

# Arrange the plots
grid.arrange(p1, p2, p3, nrow = 1)
```




```
grid.arrange(p7, p8, p10, p11, p13, p14, nrow = 2)
```




```
grid.arrange(p4, p5, p6, p9, p12, nrow = 2)
```



Selbst wenn man die 4 stärksten Ausreißer unter den Residuen größer 55 entfernt, bleibt ersichtlich, dass die Verteilung der Residuen rechtsschief ist. Die Güte des Modells ist daher fragwürdig.

[1P] b: Welchen systolischen Blutdruck hat eine Person mit folgendem Profil:

Frau, 50 Jahre, High School, Raucher, 8 Zig/Tag, keine Blutdruck senkenden Medikamente, 220 mg/dl Serum Cholesterol, 85 mmHg diastolischer Blutdruck, BMI von 30, kein Diabetes, 90 bpm Herzrate und Glukoselevel von 90 mg/dl.

```r
# Create a new data frame that contains the person's profile
new_data <- data.frame(
  sex = factor("2", levels = c("1","2")), # Female
  age = 50,
  educ = factor("2", levels = c("1","2","3","4")), # High School
  cursmoke = factor("1", levels = c("0","1")), # Yes
  cigpday = 8,
  bpmeds = factor("0", levels = c("0","1")), # No
  totchol = 220,
  diabp = 85,
  bmi = 30,
  diabetes = factor("0", levels = c("0","1")), # No
  heartrte = 90,
  glucose = 90
)

# Predict the systolic blood pressure
predicted_sysbp <- predict(model, newdata = new_data)
```

```r
print(predicted_sysbp)
```

```
##        1
## 138.4275
```

```r
# Get the confidence interval for the predicted value
predicted_sysbp_conf_int <- predict(model, newdata = new_data, interval = "confidence")
print(predicted_sysbp_conf_int)
```

```
##        fit      lwr      upr
## 1 138.4275 137.1283 139.7267
```

```r
# Get the prediction interval for the predicted value
predicted_sysbp_pred_int <- predict(model, newdata = new_data, interval = "prediction")
print(predicted_sysbp_pred_int)
```

```
##        fit      lwr      upr
## 1 138.4275 113.9516 162.9034
```

Der systolische Blutdruck der Person mit dem gegebenen Profil ist 138.4 mmHg (Konfidenzintervall [137.1,139.7], Vorhersageintervall [114.0,162.9]).