Computational Statistics

Robin Baudisch
Merlin Kopfmann
Maximilian Neudert

17. Juni 2019

Inhaltsverzeichnis

1 Einführung

Alle Aufgaben werden mit folgendem Seed bearbeitet:

```
set.seed(42)
```

Und es werden folgende Libraries benutzt:

```
usepackage = function(name) {
  x = deparse(substitute(name))
  if (!require(x,character.only = TRUE)) {
    install.packages(x,dep=TRUE)
    if(!require(x,character.only = TRUE)) stop("Package not found")
  }
}
usepackage(knitr)
usepackage(ggplot2)
usepackage(tidyr)
usepackage(lm.beta)
usepackage(car)
usepackage(magrittr)
usepackage(boot)
```

1.1 A1

```r
# Lade Daten
load(file='res/Donald.RData')
data <- Donald_1

# Fitte die Regression
fit <- lm(
  Trump ~ Alter + Geschlecht + Minderheit + Fremdenfeindlich + IQ,
  data = data
)

# Ergebnis
summary(fit)
#>
#> Call:
#> lm(formula = Trump ~ Alter + Geschlecht + Minderheit + Fremdenfeindlich +
#>     IQ, data = data)
#>
#> Residuals:
#>      Min       1Q   Median       3Q      Max
#> -15.6835  -4.5286  -0.0023   4.1231  13.0974
#>
#> Coefficients:
#>                  Estimate Std. Error t value Pr(>|t|)
#> (Intercept)      29.73834    3.60973   8.238 9.73e-14 ***
#> Alter             0.18153    0.03049   5.954 1.91e-08 ***
#> Geschlecht        5.75572    0.99977   5.757 4.97e-08 ***
#> Minderheit       -6.57586    1.82843  -3.596 0.000443 ***
#> Fremdenfeindlich  9.34984    0.16325  57.272  < 2e-16 ***
#> IQ               -0.40135    0.03016 -13.309  < 2e-16 ***
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> Residual standard error: 6.065 on 144 degrees of freedom
#> Multiple R-squared:  0.9713, Adjusted R-squared:  0.9703
#> F-statistic: 973.6 on 5 and 144 DF,  p-value: < 2.2e-16
```

1.2 A2

```
# standardisiere die Parameter des Regressionsmodells
fit.beta <- lm.beta(fit)
print(fit.beta)
#>
#> Call:
#> lm(formula = Trump ~ Alter + Geschlecht + Minderheit + Fremdenfeindlich +
#>     IQ, data = data)
#>
#> Standardized Coefficients::
#>      (Intercept)            Alter       Geschlecht       Minderheit
#>       0.00000000       0.08513441       0.08190695      -0.05790182
#> Fremdenfeindlich               IQ
#>       0.91808061      -0.19120337
```

Der Parameter "Fremdenfeindlich" hat den größten Effekt auf die abhängige Variable. Je höher der Parameterwert, desto größer die Zustimmung zu Trump (in %). Der Parameter "IQ" hat einen moderaten negativen Effekt auf die Ausprägung der abhängigen Variable. Je höher der Parameterwert, desto geringer die Zustimmung zu Trump (in %). DIe Parameter "Geschlecht", "Minderheit" und "Alter" haben jeweils einen geringen Effekt auf die Ausprägung der abhängigen Variable.

1.3 A3

```
KI <- confint(object = fit, level = 0.95)
kable(KI, format = 'pandoc', align = 'c', digits = 3)
```

	2.5 %	97.5 %
(Intercept)	22.603	36.873
Alter	0.121	0.242
Geschlecht	3.780	7.732
Minderheit	-10.190	-2.962
Fremdenfeindlich	9.027	9.673
IQ	-0.461	-0.342

In der Ausgabe sehen wir die Intervallgrenzen für das 95%-Konfidenzintervall.

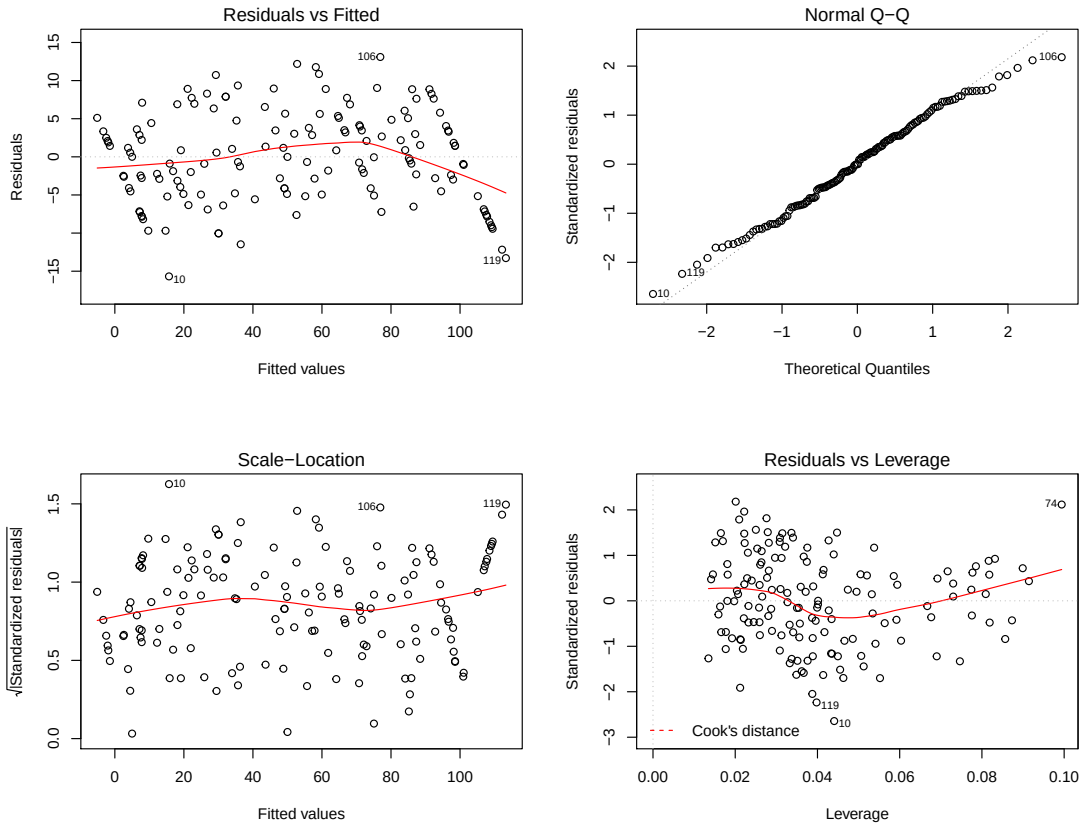
1.4 A4

```
vif_fit <- vif(mod = fit)
vif_fit
#>            Alter       Geschlecht       Minderheit Fremdenfeindlich
#>         1.024821         1.014460         1.299054         1.287851
#>               IQ
#>         1.034409
```

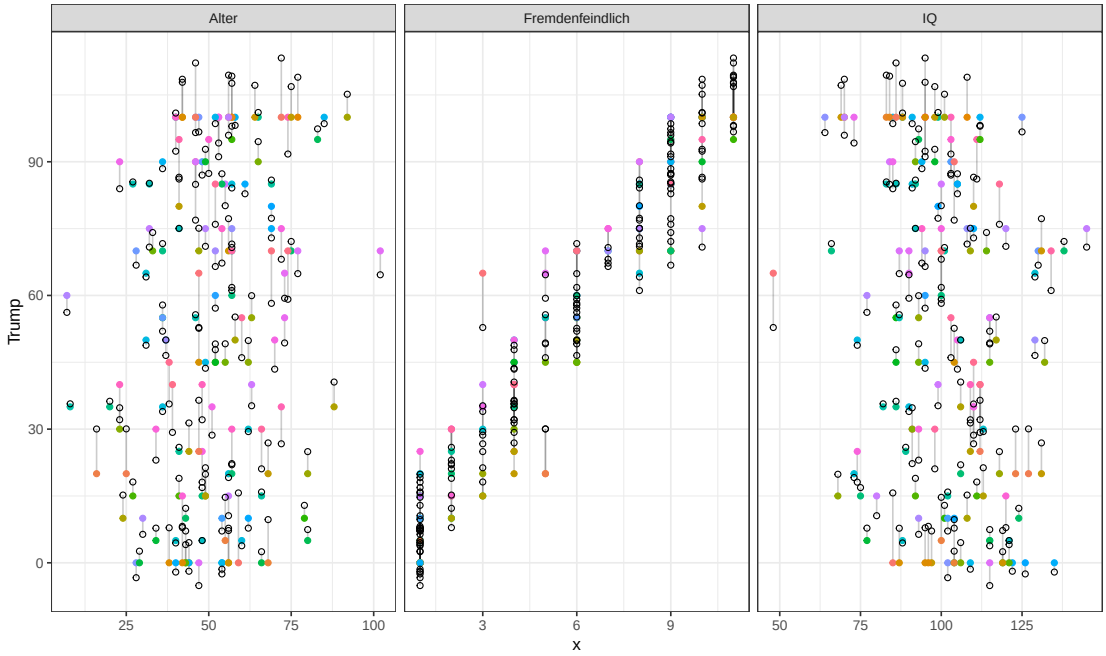
Die VIF-Werte liegen alle deutlich unter 5(10), es liegen also keine Hinweise für Multikolliniarit?t zwischen den Modellparametern vor.

1.5 A5

```
par(mfrow = c(2, 2))
plot(fit)
```



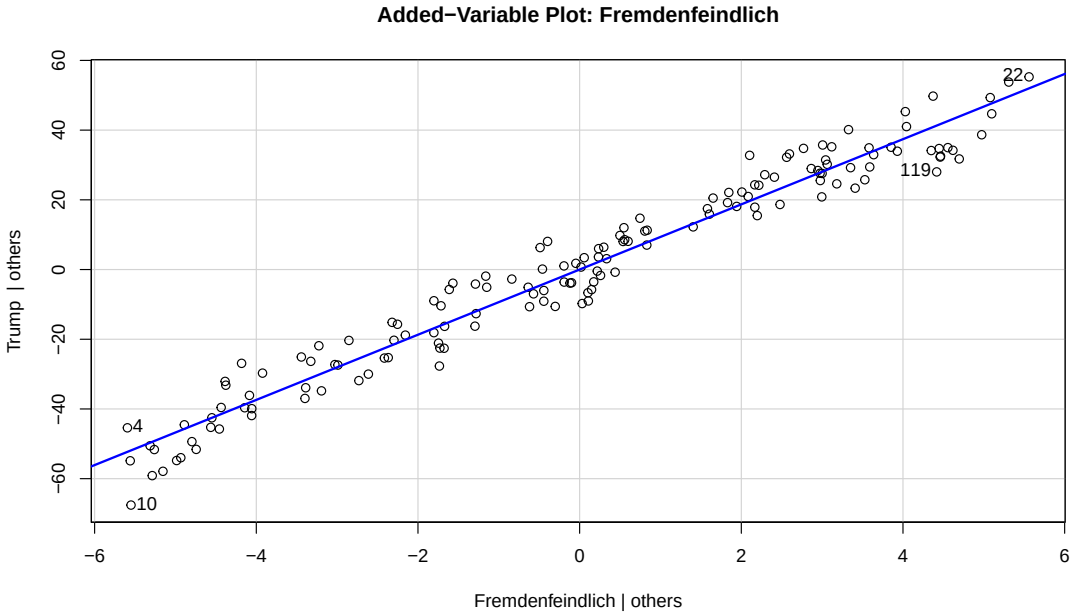

```
data$predicted <- predict(fit)
data$residuals <- residuals(fit)

data %>%
  gather(key = "iv", value = "x", -Trump, -predicted, -residuals, -Minderheit, -Geschlech
  ggplot(aes(x = x, y = Trump)) +
  geom_segment(aes(xend = x, yend = predicted), alpha = .2) +
  geom_point(aes(color = factor(residuals))) +
  guides(color = FALSE) +
  geom_point(aes(y = predicted), shape = 1) +
  facet_grid(~ iv, scales = 'free_x') +
  theme_bw()
```



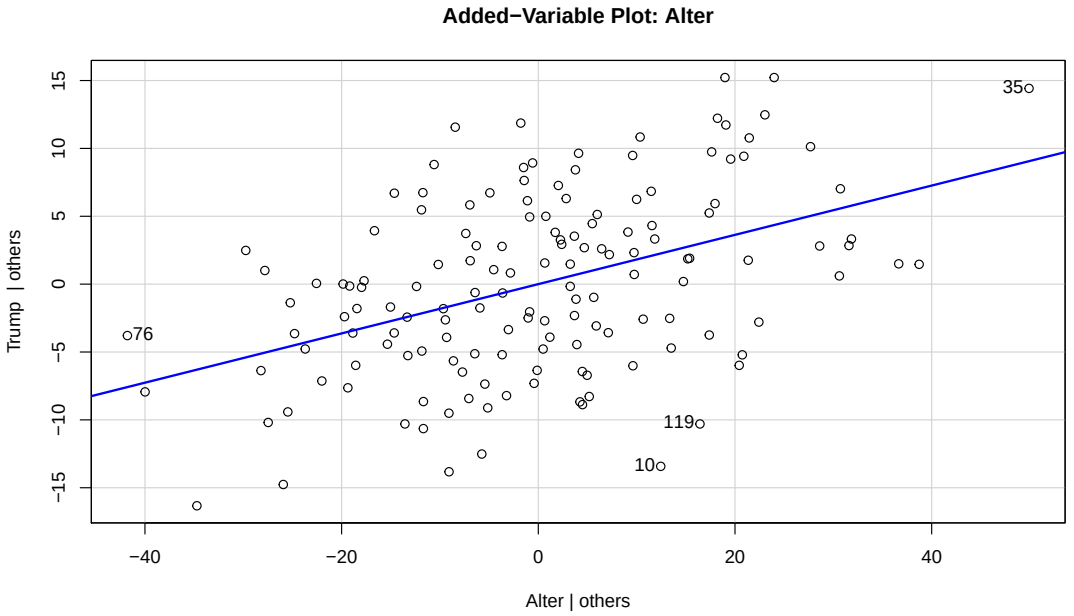
Die Residuenanalyse ergibt, dass die Residuen näherungsweise normalverteilt sind und es keine klar erkennbaren Muster in den Residuenplots gibt.

Zudem wurden die Ausprägungen der (nicht-binären) unabhängigen Parameter den Ausprägungen der abhängigen Variablen durch Scatterplots gegenübergestellt. Lediglich der Parameter "Fremdenfeindlich" weist einen klar erkennbaren linearen Zusammenhang zur abhängigen Variabel auf. Dies bestätigt das Ergebnis aus 2.

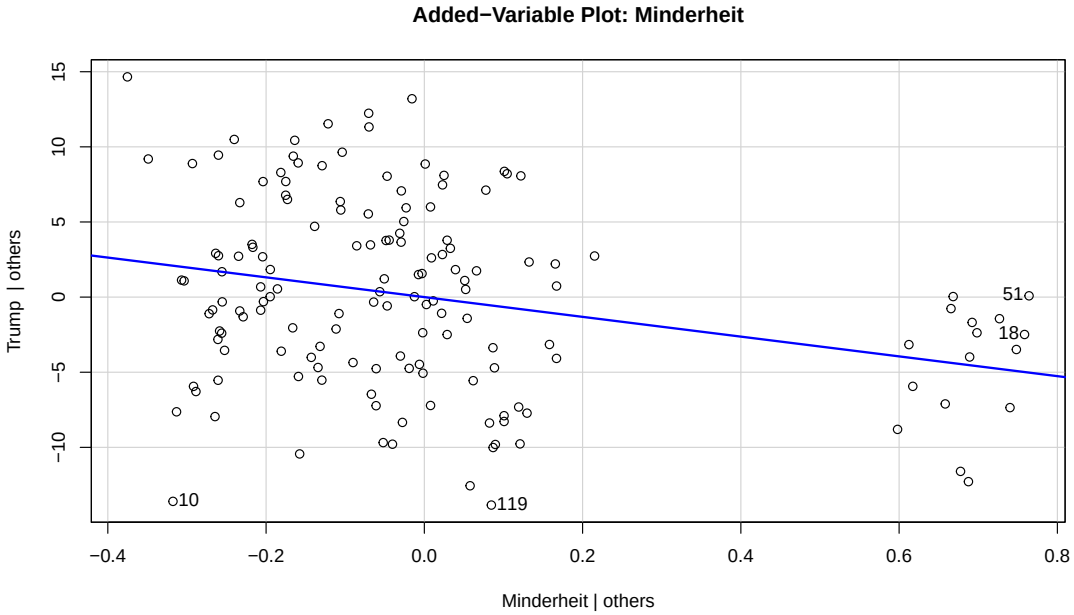
```
avPlot(fit, "Fremdenfeindlich")
```



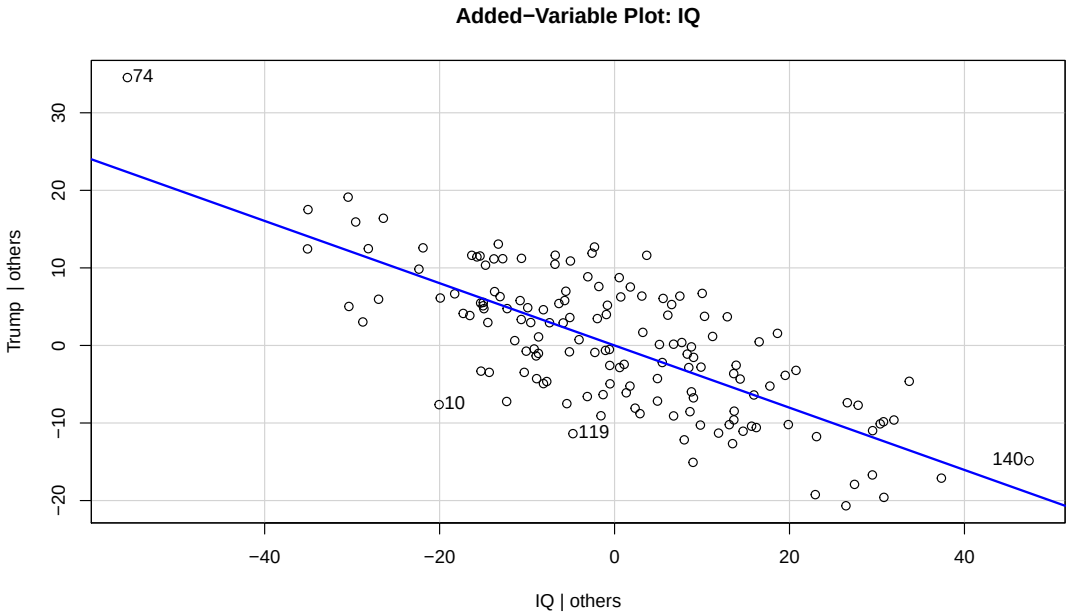

```
avPlot(fit, "Alter")
```



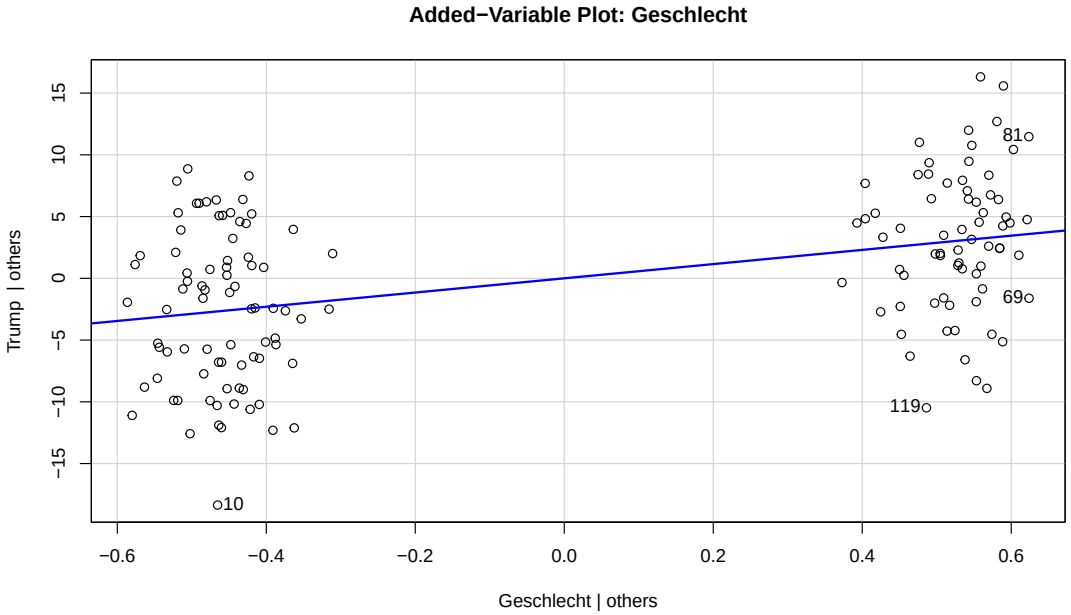

```
avPlot(fit, "Minderheit")
```




```
avPlot(fit, "IQ")
```




```
avPlot(fit, "Geschlecht")
```



An den Added-Variable Plot wird nochmals deutlich, dass “Fremdenfeindlich”, “IQ” und “Alter” einen linearen Trend aufweisen. Die beiden binären Variablen “Minderheit” und “Geschlecht” kann man nicht linear erklären.

1.6 A6

```
my_data <- data.frame("Geschlecht" = 1, "Alter" = 24,
                      "Minderheit" = 0, "Fremdenfeindlich" = 5, "IQ" = 100)
my_data$predicted <- predict(fit, newdata =  my_data)


my_data2 <- data.frame("Geschlecht" = 1, "Alter" = 27,
                      "Minderheit" = 1, "Fremdenfeindlich" = 3, "IQ" = 100)
my_data2$predicted <- predict(fit, newdata =  my_data2)


my_data3 <- data.frame("Geschlecht" = 1, "Alter" = 29,
                      "Minderheit" = 0, "Fremdenfeindlich" = 8, "IQ" = 90)
my_data3$predicted <- predict(fit, newdata =  my_data3)

pred_sum <- rbind(my_data, my_data2, my_data3)
pred_sum
#>   Geschlecht Alter Minderheit Fremdenfeindlich  IQ predicted
```

```
#> 1          1    24          0                5 100  46.46492
#> 2          1    27          1                3 100  21.73396
#> 3          1    29          0                8  90  79.43557
```

Das Modell prognostiziert uns unterschiedliche Ergebnisse. Man erkennt klar, dass “Fremdenfeindlich” die größte Auswirkung auf die Zustimmungsrate hat.

2 Lineare Regression

2.1 A1

```
load(file = "res/Donald.RData")

## Vergleich der Variablenverteilung zwischen Trainings- und Testdatensatz/Lineare Regres
for (i in 1:3){
smp_size <- 100
train_ind <- sample(seq_len(nrow(Donald_1)), size = smp_size)

train <- Donald_1[train_ind, ]
test <- Donald_1[-train_ind, ]
print(i)
cat("train: \n\n")
print(summary(train))
cat("\n")
cat("test: \n\n")
print(summary(test))
linreg <- lm(Trump ~ Alter + Geschlecht + Minderheit + Fremdenfeindlich + IQ,
             data = train)
pred <- predict.lm(linreg, test)
cat("\n")
cat("MSE: \n")
print(mean((test$Trump - pred) ^ 2))
cat("_____________________________ \n\n")
}
#> [1] 1
#> train:
#>
#>    Geschlecht       Alter         Minderheit   Fremdenfeindlich
#>  Min.   :0.00   Min.   : 7.00   Min.   :0.00   Min.   : 1.00
#>  1st Qu.:0.00   1st Qu.:40.00   1st Qu.:0.00   1st Qu.: 2.00
#>  Median :0.00   Median :50.50   Median :0.00   Median : 5.00
#>  Mean   :0.43   Mean   :49.79   Mean   :0.12   Mean   : 5.35
#>  3rd Qu.:1.00   3rd Qu.:58.50   3rd Qu.:0.00   3rd Qu.: 9.00
#>  Max.   :1.00   Max.   :92.00   Max.   :1.00   Max.   :11.00
#>        IQ            Trump
#>  Min.   : 66.0   Min.   :  0.00
#>  1st Qu.: 92.0   1st Qu.: 15.00
#>  Median :103.5   Median : 45.00
#>  Mean   :102.9   Mean   : 49.35
```

```
#>  3rd Qu.:112.0   3rd Qu.: 81.25
#>  Max.   :145.0   Max.   :100.00
#>
#> test:
#>
#>    Geschlecht       Alter          Minderheit   Fremdenfeindlich
#>  Min.   :0.00   Min.   : 23.00   Min.   :0.00   Min.   : 1.00
#>  1st Qu.:0.00   1st Qu.: 42.00   1st Qu.:0.00   1st Qu.: 3.00
#>  Median :1.00   Median : 53.00   Median :0.00   Median : 6.00
#>  Mean   :0.54   Mean   : 54.14   Mean   :0.08   Mean   : 5.82
#>  3rd Qu.:1.00   3rd Qu.: 66.50   3rd Qu.:0.00   3rd Qu.: 9.00
#>  Max.   :1.00   Max.   :102.00   Max.   :1.00   Max.   :11.00
#>        IQ             Trump
#>  Min.   : 48.00   Min.   :  0.00
#>  1st Qu.: 87.25   1st Qu.: 21.25
#>  Median : 95.00   Median : 70.00
#>  Mean   : 96.68   Mean   : 57.40
#>  3rd Qu.:106.00   3rd Qu.: 88.75
#>  Max.   :138.00   Max.   :100.00
#>
#> MSE:
#> [1] 35.98927
#> ------------------------------
#>
#> [1] 2
#> train:
#>
#>    Geschlecht       Alter          Minderheit  Fremdenfeindlich
#>  Min.   :0.00   Min.   :  7.00   Min.   :0.0   Min.   : 1.00
#>  1st Qu.:0.00   1st Qu.: 41.00   1st Qu.:0.0   1st Qu.: 2.00
#>  Median :0.00   Median : 52.00   Median :0.0   Median : 6.00
#>  Mean   :0.42   Mean   : 51.61   Mean   :0.1   Mean   : 5.58
#>  3rd Qu.:1.00   3rd Qu.: 60.50   3rd Qu.:0.0   3rd Qu.: 9.00
#>  Max.   :1.00   Max.   :102.00   Max.   :1.0   Max.   :11.00
#>        IQ             Trump
#>  Min.   : 48.00   Min.   :  0.00
#>  1st Qu.: 89.75   1st Qu.: 20.00
#>  Median :100.00   Median : 57.50
#>  Mean   :100.26   Mean   : 53.15
#>  3rd Qu.:111.25   3rd Qu.: 85.00
#>  Max.   :145.00   Max.   :100.00
#>
#> test:
```

```
#>
#>    Geschlecht       Alter         Minderheit   Fremdenfeindlich
#>  Min.   :0.00   Min.   :16.00   Min.   :0.00   Min.   : 1.00
#>  1st Qu.:0.00   1st Qu.:41.25   1st Qu.:0.00   1st Qu.: 1.25
#>  Median :1.00   Median :49.50   Median :0.00   Median : 5.00
#>  Mean   :0.56   Mean   :50.50   Mean   :0.12   Mean   : 5.36
#>  3rd Qu.:1.00   3rd Qu.:60.75   3rd Qu.:0.00   3rd Qu.: 9.00
#>  Max.   :1.00   Max.   :85.00   Max.   :1.00   Max.   :11.00
#>        IQ          Trump
#>  Min.   : 66   Min.   :  0.00
#>  1st Qu.: 92   1st Qu.: 15.00
#>  Median :102   Median : 47.50
#>  Mean   :102   Mean   : 49.80
#>  3rd Qu.:112   3rd Qu.: 88.75
#>  Max.   :135   Max.   :100.00
#>
#> MSE:
#> [1] 40.43078
#> ------------------------------
#>
#> [1] 3
#> train:
#>
#>    Geschlecht       Alter         Minderheit   Fremdenfeindlich
#>  Min.   :0.00   Min.   : 7.00   Min.   :0.00   Min.   : 1.00
#>  1st Qu.:0.00   1st Qu.:42.00   1st Qu.:0.00   1st Qu.: 1.00
#>  Median :0.00   Median :52.50   Median :0.00   Median : 4.50
#>  Mean   :0.46   Mean   :51.76   Mean   :0.13   Mean   : 5.02
#>  3rd Qu.:1.00   3rd Qu.:62.25   3rd Qu.:0.00   3rd Qu.: 8.00
#>  Max.   :1.00   Max.   :83.00   Max.   :1.00   Max.   :11.00
#>        IQ            Trump
#>  Min.   : 48.0   Min.   :  0.0
#>  1st Qu.: 93.0   1st Qu.: 15.0
#>  Median :102.0   Median : 47.5
#>  Mean   :102.4   Mean   : 47.4
#>  3rd Qu.:113.0   3rd Qu.: 75.0
#>  Max.   :135.0   Max.   :100.0
#>
#> test:
#>
#>    Geschlecht       Alter         Minderheit   Fremdenfeindlich
#>  Min.   :0.00   Min.   : 16.0   Min.   :0.00   Min.   : 1.00
#>  1st Qu.:0.00   1st Qu.: 36.0   1st Qu.:0.00   1st Qu.: 3.25
```

```
#>  Median :0.00   Median : 48.0   Median :0.00   Median : 7.00
#>  Mean   :0.48   Mean   : 50.2   Mean   :0.06   Mean   : 6.48
#>  3rd Qu.:1.00   3rd Qu.: 57.0   3rd Qu.:0.00   3rd Qu.: 9.00
#>  Max.   :1.00   Max.   :102.0   Max.   :1.00   Max.   :11.00
#>        IQ            Trump
#>  Min.   : 66.0   Min.   :  0.00
#>  1st Qu.: 86.0   1st Qu.: 31.25
#>  Median : 94.5   Median : 70.00
#>  Mean   : 97.7   Mean   : 61.30
#>  3rd Qu.:108.8   3rd Qu.: 95.00
#>  Max.   :145.0   Max.   :100.00
#>
#> MSE:
#> [1] 43.26809
#> ______________________________
```

Durch die zufällige Aufteilung der Datenpunkte in zwei disjunkte Datensätze (train, test) entstehen Diskrepanzen zwischen den Verteilungen der Variablen. Diese Schwankungen wirken sich dann auch auf die Modellgüte aus. Wir wissen aus der vergangenen Übung, dass das Merkmal "Fremdenfeinlich" die Kovariate mit dem stärksten Einfluss auf die abhängige Variable ist. Vergleichen wir die Verteilung dieses Merkmals zwischen Train- und Testdatensatz in 1 mit dem aus 3, fällt auf, dass die Diskrepanzen zwischen train und test in 3 eindeutig stärker ausfallen, als in 1. Dies spiegelt sich im jeweiligen MSE wieder: Das Modell, welches mit den Daten aus 1 trainiert und getestet wurde, weist einen deutlich niedrigeren MSE auf, als das Modell aus 3.

2.2 A2

a)

```
mse <- c()
for (i in 1:nrow(Donald_1)){
  lou <- lm(Trump ~ Alter + Geschlecht + Minderheit + Fremdenfeindlich + IQ,
            data = Donald_1[-i,])
  pred <- predict.lm(lou, Donald_1[i,])
  mse[i] <- mean((Donald_1[i,]$Trump - pred) ^ 2)
}
mse <- sum(mse)/nrow(Donald_1)
kable(mse, format = 'pandoc', align = 'c', digits = 3)
```

x
38.191

Der MSE für die Leave-one-out Cross-Validation liegt zwischen den berechneten MSE's aus Aufgabe 1. Dies ist zu erwarten, da wir hier untersuchen, wie gut das lineare Regressionsmodell auf "ungesehenen" Daten performt. In dem wir jeweils nur einen Datenpunkt als "Test"-Datensatz verwenden, bekommen wir so viele MSE-Werte, wie Datenpunkte im Datensatz. Diese MSE-Werte werden gemittelt, um eine "robustere" Einschätzung über die Generalisierbarkeit des Modells auf unsere Daten zu erhalten.

b)

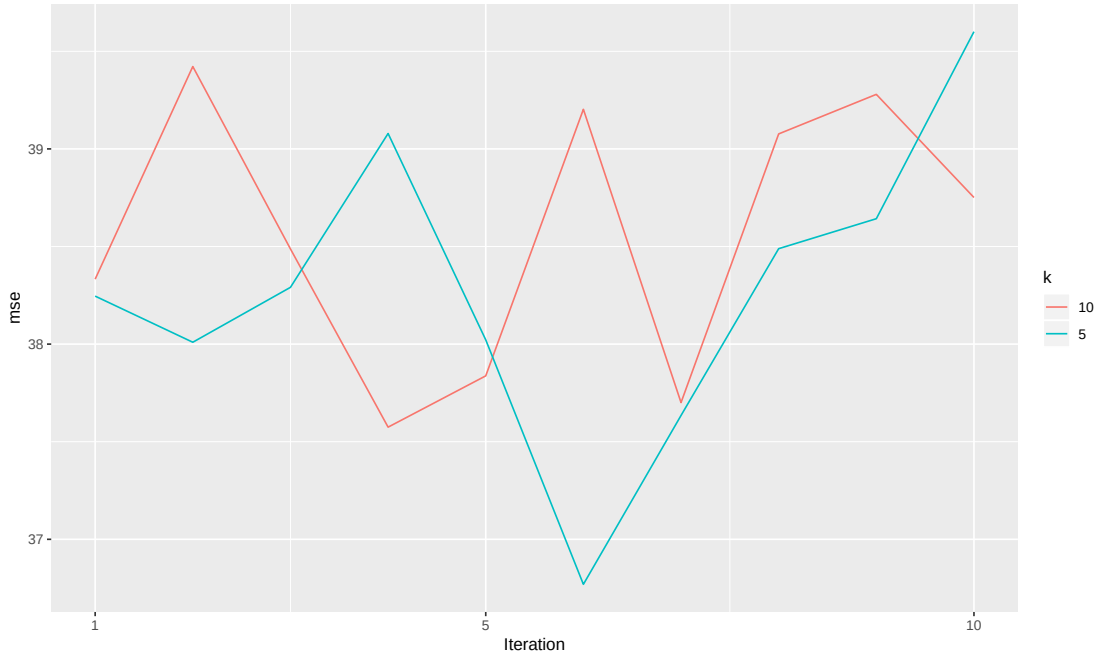
```
model <- glm(
  Trump ~ Alter + Geschlecht + Minderheit + Fremdenfeindlich + IQ,
  data = Donald_1
)

cv_model <- cv.glm(Donald_1, model, K = nrow(Donald_1))
mse1 <- cv_model$delta
kable(mse1, format = 'pandoc', align = 'c', digits = 3)
```

x
38.191
38.181

Die Implementierung des “Leave-one-out cross-validation”-Verfahrens durch cv.glm gibt zwei Modellgütemetriken zurück. Der erste Wert ist der durschnittliche MSE über die Zeilen. Dieser ist identisch mit dem Wert aus a). Man kann vermuten, dass die Implementierung des “Leave-one-out cross-validation”-Verfahrens durch cv.glm identisch mit unserer manuellen Implementierung aus a) ist. Der zweite Wert ist laut Dokumentation ein Bias-korrigierter MSE.

2.3 A3

```
mse <- matrix(NA, nrow = 20, ncol = 3)
for (k in c(5,10)){
  for (i in 1:10){
    mse[ifelse(k == 5, i, i + 10),] <- c(i, cv.glm(Donald_1, model, K = k)$delta[1], k)
  }
}

mse <- data.frame(mse)
names(mse) <- c("index", "mse", "k")
mse$k <- as.character(mse$k)

gg = ggplot(
data = mse,
  mapping = aes(
    x = index,
    y = mse,
    color = k
  )
)
gg = gg + xlab("Iteration")
gg = gg + scale_x_continuous(breaks = c(1,5,10))
gg + geom_line()
```



Bei der k-fachen Kreuzvalidierung wird die ursprüngliche Stichprobe zufällig in k gleich große Teilstichproben aufgeteilt. Von den k Teilstichproben wird eine einzige Teilstichprobe als Validierungsdaten für den Test des Modells aufbewahrt und die restlichen $k - 1$ Teilstichproben werden als Trainingsdaten verwendet. Der Kreuzvalidierungsprozess wird dann k-mal wiederholt, wobei jede der k Teilstichproben genau einmal als Validierungsdaten verwendet wird. Die k-Ergebnisse können dann gemittelt werden, um eine einzige Schätzung zu erhalten. Der Vorteil dieser Methode gegenüber des wiederholt zufälligen Subsampling besteht darin, dass alle Beobachtungen sowohl für das Training als auch für die Validierung verwendet werden und jede Beobachtung genau einmal für die Validierung verwendet wird. Die k-fache Kreuzvalidierung mit $k = 10$ wird häufig verwendet, aber im Allgemeinen bleibt k ein nicht fixierter Parameter.

Untereinander schwanken die MSE-Werte zufällig. Als Trend ist zu beobachten, dass für $k = 5$ die MSE-Werte niedriger sind. Alles in allem variieren die MSE-Werte um einen Wert von 38. Dies deckt sich mit den Ergebnissen aus Aufgabe 2. Die Schwankungen sind ebenfalls im Bereich der generierten MSE-Werte aus aus Aufgabe 1. Interessant ist, dass wir in Aufgabe 1 mit MSE ~ 35 einen, im Vergleich, relativ niedrigen MSE-Wert erzielt haben.

3 Cross Validation