set.seed(42)

# Aufgabe 1

## a)

```
library(MASS)
sig = 6.8
R = 0.73
tgeld = c(5.1,15.6,28.2,11.1,4.0,31.5,19.5)
y = tgeld
time = c(18,19.5,20,20.5,21.25,21.5,22)
gender = c(0,1,1,0,0,1,1)
eins = rep(1,7)

X = matrix(
  c(eins, time, gender),
  nrow = 7,
  ncol = 3,
  byrow = FALSE
)

XT = t(X)

XTX = XT %*% X

iXTX = ginv(XTX)

b = iXTX %*% XT %*% y
b = round(b, 2)

model <- lm(y~time+gender)
model$coefficients
#> (Intercept)        time      gender
#> -11.8775934   0.9344398  16.1879668
```

## b)

Wir erhalten dann eine Sch??tzung der Regressionskoeffizienten mittels:

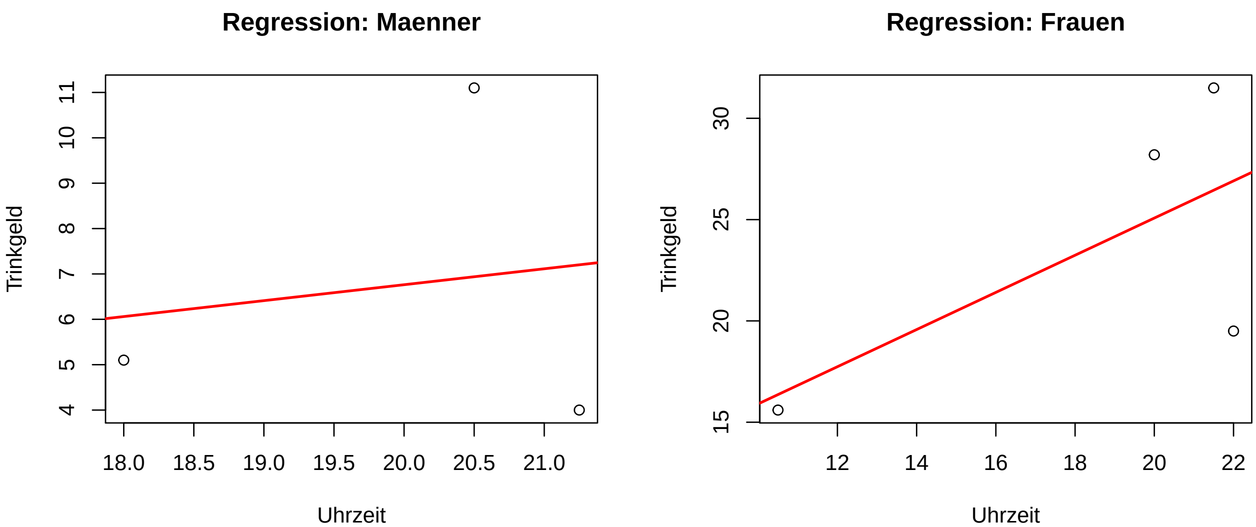
$$\hat{b} = (X^T X)^{-1} X^T y \approx (-11.88, 0.93, 16.19)$$

Sowohl Uhrzeit als auch Geschlecht (Frauen) haben einen positiven Einfluss auf die abh??ngige Variable Trinkgeld. Dabei ist das Geschlecht deutlich gr????er gewichtet als die Uhrzeit.

## c)

```
y_m <- c(5.1, 11.1, 4)
X_m <- matrix(c(1, 1, 1, 18, 20.5, 21.25), nrow = 3, ncol = 2)
```

```
XTX_m <- solve(t(X_m)%*%X_m)
b_hat_m <- XTX_m%*%t(X_m)%*%y_m

y_w <- c(15.6, 28.2, 31.5, 19.5)
X_w <- matrix(c(1, 1, 1, 1, 10.5, 20, 21.5, 22), nrow = 4, ncol = 2)
XTX_w <- solve(t(X_w)%*%X_w)
b_hat_w <- XTX_w%*%t(X_w)%*%y_w

par(mfrow=c(1,2))

plot(c(18, 20.5, 21.25), y_m, xlab = 'Uhrzeit', ylab = 'Trinkgeld', main = 'Regression: Maenner')
abline(b_hat_m, lw=2, col='red')

plot(c(10.5, 20, 21.5, 22), y_w, xlab = 'Uhrzeit', ylab = 'Trinkgeld', main = 'Regression: Frauen')
abline(b_hat_w, lw=2, col='red')
```




```
f = t(b) %*% c(0,1,1)
m = t(b) %*% c(0,1,0)
```

Die ??nderung des zu erwartenden Trinkgelds pro Stunde ist die Steigung der Regression, das hei??t:

Trinkgeld pro Stunde M??nner: $y = \hat{b}_1 + 1 \cdot \hat{b}_2 = 0.93$

Trinkgeld pro Stunde Frauen: $y = \hat{b}_1 + 1 \cdot \hat{b}_2 = 17.12$

## d)

H0: "Es besteht kein signfikanter Unterschied der H??he des Trinkgelds unter den Geschlechtern." H1: "Es besteht ein signfikanter Unterschied der H??he des Trinkgelds unter den Geschlechtern."

```
n <- 7
p <- 3
beta <- b[3]

test <- beta/(sig*sqrt(iXTX[3,3]))

t_value <-  qt(c(0.025, 0.975), df=n-p)
```

```
if (t_value[1] < test & test < t_value[2]){
  print('H0 wird nicht verworfen!')
  } else {
    print('H0 wird verworfen!')
  }
#> [1] "H0 wird verworfen!"
```

Wir erhalten einen Testwert von 2.95 und dieser liegt im Ablehnungsbereich -2.78, 2.78. Folglich lehnen wir die Nullhypothese ab.

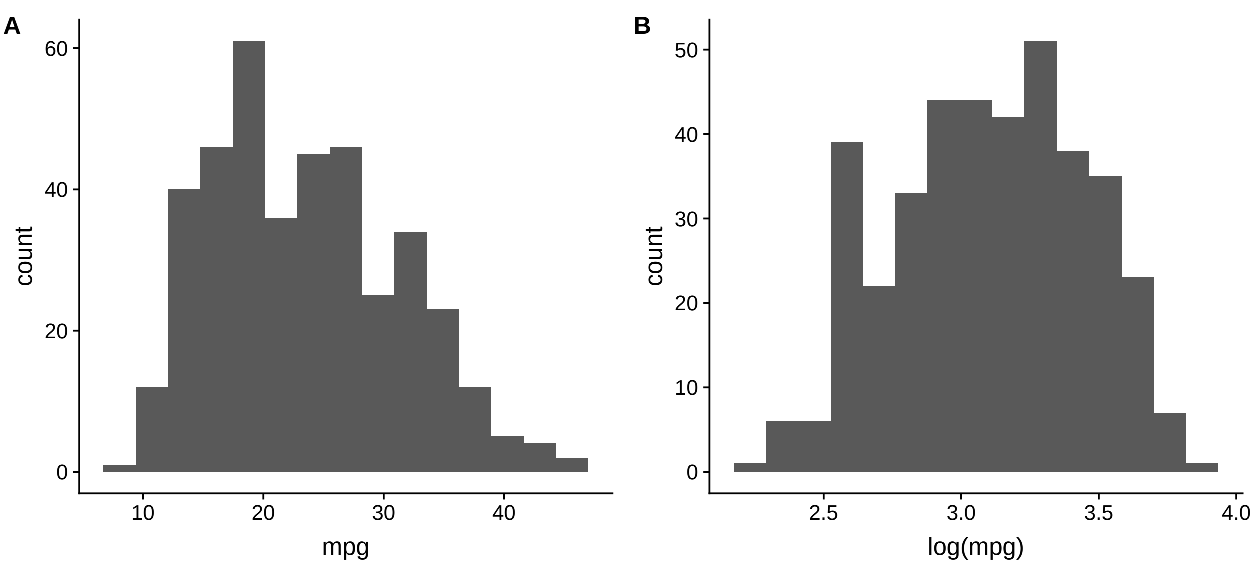
### e)

Der SSE ist die Quadratsumme der Residuen und damit unmittelbar abh??ngig von der Skalierung der Zielwerte, deswegen kann man daraus nicht folgern, dass die absolute Trinkgeldh??he schlechter erkl??rt wird, als die prozentuale. Als Beispiel kann man die Werte hier betrachten, diese liegen im unteren 2-stelligen Bereicht. Nimmt man nun das absolute Trinkgeld in Yen, so befinden sich die Werte im h??heren 3-stelligen oder sogar im 4-stelligen Bereich folglich w??ren bei gleicher Modellg??te die Betr??ge der Residuen deutlich gr????er.

## Aufgabe 2

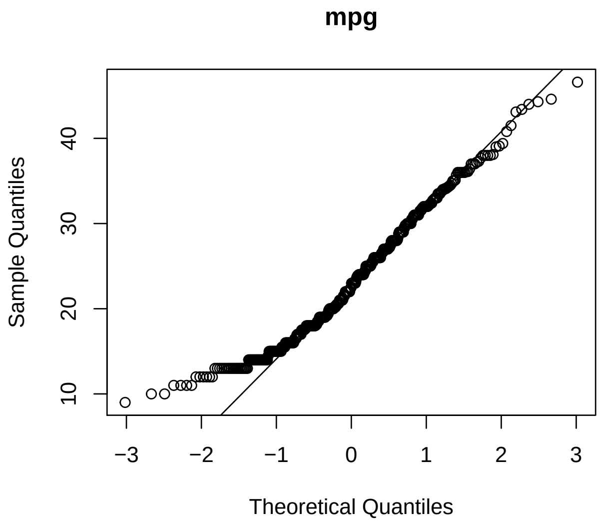
### a)

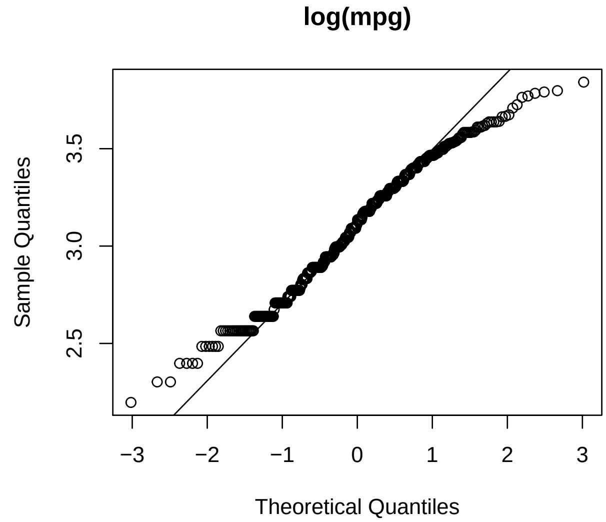
```
library(ISLR)
library(ggplot2)
library(cowplot)

mpg_hist <- ggplot(Auto, aes(x=mpg)) + geom_histogram(bins = 15)
log_mpg_hist <- ggplot(Auto, aes(x=log(mpg))) + geom_histogram(bins = 15)
plot_grid(mpg_hist, log_mpg_hist, labels = "AUTO")
```




```
par(mfrow=c(1,2))
```

```
qqnorm(Auto$mpg, main='mpg')
qqline(Auto$mpg)

qqnorm(log(Auto$mpg), main='log(mpg)')
qqline(log(Auto$mpg))
```

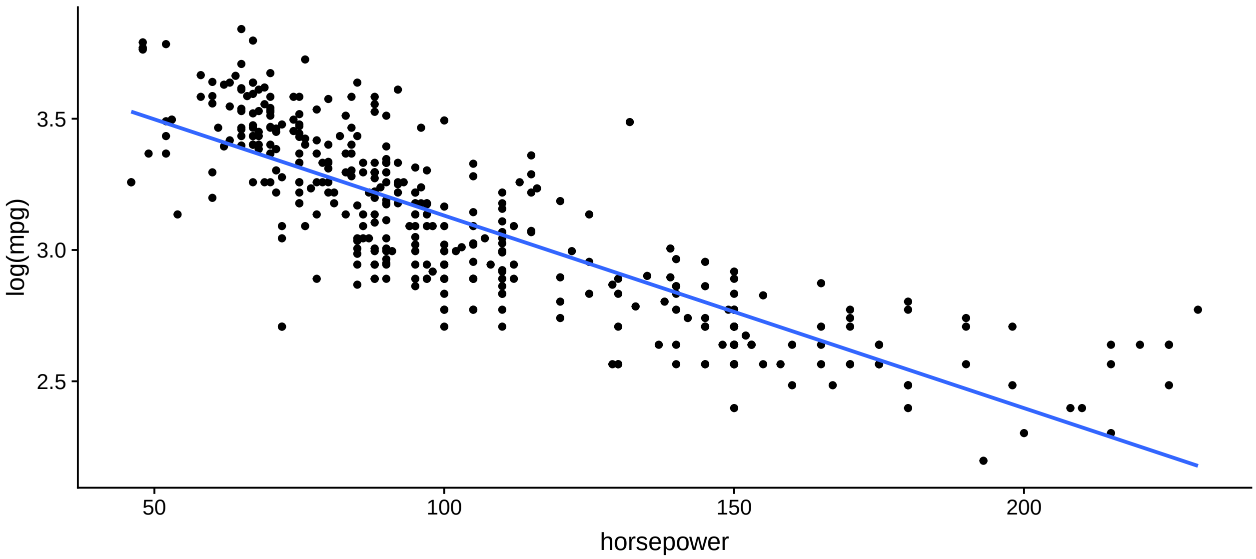



Laut Histogramm gleicht mpg einer rechtsschiefen Verteilung, log(mpg) kommt laut Histogramm einer Normalverteilung n??her. Die QQ-Plots sind schwieriger auszuwerten, beide Variablen weichen an den R??ndern deutlich von der Geraden ab. Allerdings weichen die Punkte von log(mpg) an den R??ndern "symmetrischer" von der Geraden ab, als bei mpg.

## b)

```
X <- matrix(NA, nrow = nrow(Auto), ncol = 2)
X[,1] <- rep(1, nrow(Auto))
X[,2] <- Auto$horsepower
XTX <- t(X) %*% X

iXTX = ginv(XTX)

b <- iXTX %*% t(X) %*% log(Auto$mpg)

ggplot(Auto, aes(x=horsepower, y=log(mpg))) + geom_point() + geom_smooth(method = 'lm', se = FALSE)
```




Auf den ersten Blick deutet ein beta von $-0.0073338$ auf keinen Zusammenhang zwischen Pr??diktor und Zielgr????e hin. Plottet man Zielgr????e auf Pr??diktor, ist ein klarer negativer linearer Zusammenhang erkennbar. Dies liegt daran, dass horsepower und log(mpg) unterschiedlich skaliert sind und die Steigung der Regression deswegen sehr klein wird.

```
x = Auto$horsepower
y = log(Auto$mpg)
fit = aov(y~x)
summary(fit)
#>              Df Sum Sq Mean Sq F value Pr(>F)
#> x             1  31.16  31.157   864.7 <2e-16 ***
#> Residuals   390  14.05   0.036
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
logval = t(b) %*% c(1, 98)
val = exp(logval)
```

$p < 0.001$ und somit ist der Zusammenhang signifikant zum 5% Niveau.

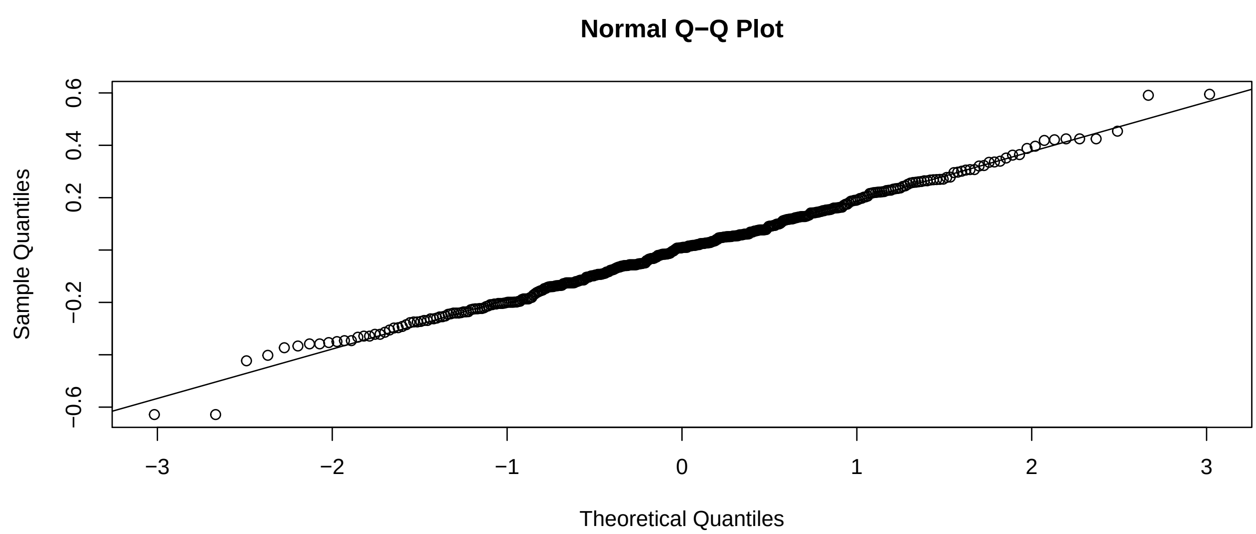
F??r ein Auto mit 98PS w??rden man anhand des Modells 23.24 an mpg erwarten.

```
logval = abs(t(b) %*% c(0, 20))
```

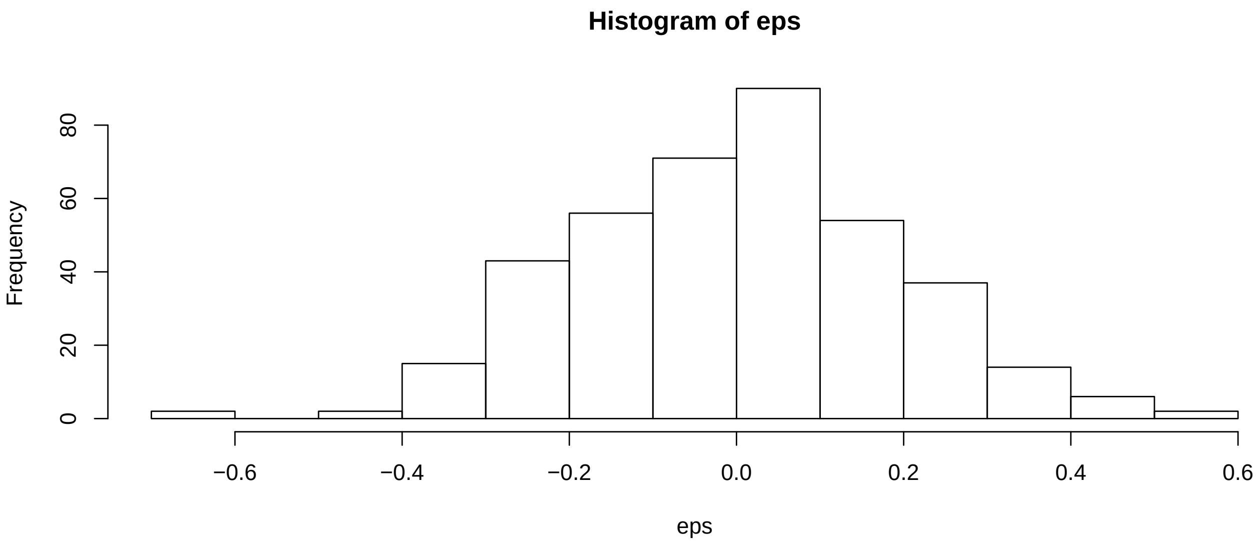
Der logarithmierte Verbrauch unterscheidet sich f??r 20PS erwartungsgem???? um 0.1466753.

```
X_log <- matrix(NA, nrow = nrow(Auto), ncol = 2)
X_log[,1] <- rep(1, nrow(Auto))
X_log[,2] <- log(Auto$horsepower)
XTX_log <- solve(t(X_log)%*%X_log)
b_hat_log <- XTX_log%*%t(X_log)%*%log(Auto$mpg)
b_hat_log[2,]
#> [1] -0.841847

n <- nrow(Auto)
p <- 3
beta <- b[2,]
sig_hat <- sd(log(Auto$mpg))
```

```
test <- beta/(sig_hat*sqrt(XTX[2,2]))
test
#> [1] -9.785675e-06

t_value <-  qt(c(0.025, 0.975), df=n-p)

if (t_value[1] < test & test < t_value[2]){
  print('H0 wird nicht verworfen!')
} else {
  print('H0 wird verworfen!')
}
#> [1] "H0 wird nicht verworfen!"

mpg98 <- exp(b[1,] + b[2,]*98)
mpg98
#> [1] 23.23728

diff_20 <- abs(20*b[2,])
diff_20
#> [1] 0.1466753

y_hat <- c()
for(i in 1:nrow(Auto)){
  y_hat[i] <- (b[1,] + b[2,]*Auto$horsepower[i])
}

eps <- log(Auto$mpg) - y_hat
qqnorm(eps)
qqline(eps)
```




```
hist(eps)
```




```
shapiro.test(eps)
#>
#>  Shapiro-Wilk normality test
#>
#> data:  eps
#> W = 0.99607, p-value = 0.4421
```

Laut Shapiro-Wilk-Test wird H0 ("Die Residuen sind normalverteilt.") nicht verworfen. Wir k??nnen von einer Normalverteilung ausgehen. Die Plots best??tigen unsere Vermutung.

### c)

#### A)

```
# preds
x1 = Auto$horsepower
x2 = Auto$year
x3 = as.factor(Auto$year)
y = log(Auto$mpg)
# manual linear regression
X = matrix(
  c(
    rep(1, nrow(Auto)),
    x1,
    x2,
    x3
  ),
  nrow = nrow(Auto),
  ncol = 4
)
XTX = t(X) %*% X
iXTX = ginv(XTX)
b = iXTX %*% t(X) %*% y
# anova
```

```
fit = aov(y~x1+x2+x3)
summary(fit)
#>              Df Sum Sq Mean Sq  F value   Pr(>F)    
#> x1            1 31.157  31.157 1310.530  < 2e-16 ***
#> x2            1  2.934   2.934  123.424  < 2e-16 ***
#> x3           11  2.132   0.194    8.153 7.85e-13 ***
#> Residuals   378  8.987   0.024                      
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Nach anova haben PS, Jahr und Ursprungsland jeweils ein $p < 0.001$ und weisen somit einen statistisch signifikanten Zusammenhang mit der Zielvariable auf.

### B)

```
# preds
x1 = Auto$cylinders
x2 = Auto$displacement
x3 = Auto$horsepower
x4 = Auto$weight
x5 = Auto$acceleration
x6 = Auto$year
x7 = as.factor(Auto$year)
y = log(Auto$mpg)
# manual linear regression
X = matrix(
  c(
    rep(1, nrow(Auto)),
    x1,
    x2,
    x3,
    x4,
    x5,
    x6,
    x7
  ),
  nrow = nrow(Auto),
  ncol = ncol(Auto) + 1
)
#> Warning in matrix(c(rep(1, nrow(Auto)), x1, x2, x3, x4, x5, x6, x7), nrow
#> = nrow(Auto), : data length [3136] is not a sub-multiple or multiple of the
#> number of columns [10]
XTX = t(X) %*% X
iXTX = ginv(XTX)
b = iXTX %*% t(X) %*% y
# anova
fit = aov(y~x1+x2+x3+x4+x5+x6+x7)
summary(fit)
#>              Df Sum Sq Mean Sq  F value   Pr(>F)    
#> x1            1 30.907  30.907 2404.831  < 2e-16 ***
#> x2            1  2.149   2.149  167.201  < 2e-16 ***
#> x3            1  1.009   1.009   78.503  < 2e-16 ***
```

```
#> x4           1  1.676   1.676  130.416  < 2e-16 ***
#> x5           1  0.037   0.037    2.899   0.0895 .
#> x6           1  3.747   3.747  291.542  < 2e-16 ***
#> x7          11  0.878   0.080    6.209 2.08e-09 ***
#> Residuals   374  4.807   0.013
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```