```r
library(lattice)
lmdat = read.table(file.path(pfadu, "lmdat.txt"))

# Für den vorhandenen Data-Frame 'trees' prüfen Sie
# inwiefern Height aus Volume  vorhersagbar ist. Schätzen Sie
# Height ein bei einem Volumen von 110.

head(trees)
plot(Height ~ Volume, data = trees)

# Regression
trees.lm = lm(Height ~ Volume, data = trees)

# Weichen die Residuals von einer Normalverteilung ab?
shapiro.test(resid(trees.lm))
# Nein
# W = 0.9822, p-value = 0.8707

# Gleichmäßige Verteilung um die 0-Linie?
# Mehr oder weniger, ja.
plot(resid(trees.lm))

# Autokorrelation?
# Nein - die meisten Werte - und vor allem den zweiten Wert -
# liegen innherhalb den blauen Linien
acf(resid(trees.lm))

# Daher können wir das Problem lösen
# Abbildung mit Regressionslinie
plot(Height ~ Volume, data = trees)
abline(trees.lm)

# Statistik
summary(trees.lm)

# Es gibt eine signifikante lineare Beziehung zwischen Height und
Volume
# (R^2 = 0.36, F[1,29] = 16.2, p < 0.001).

# Die eingeschätzte Höhe bei einem Volumen von 100:
predict.lm(trees.lm, data.frame(Volume = 100))

# ggf. Bild neu malen mit diesem Wert
xlim = c(10, 110)
ylim = c(60, 100)
plot(Height ~ Volume, data = trees, xlim=xlim, ylim = ylim)
abline(trees.lm)
points(100, 92.19335, col = 2)
abline(v = 100, h = 92.19335, lty=2, col=2)

### 2. Führen Sie die folgenden Berechnungen für diese Daten durch
#

y = lmdat$y
x = lmdat$x
```

```
# Mittelwert von y, Mittelwert von x; Anzahl der Werte in x (oder y)
mx =
my =
n =

# Kovarianz berechnen:
# Die Summe von dy Mal dx. Dann  dividiert durch n-1
# Abweichung vom Mittelwert für y
dy =
# Für x
dx =
# Kovarianz

# Bestätigen

covxy = cov(y,x)

# Korrelation (r) gleicht die  Kovarianz
# dividiert durch  (sd von y Mal sd von x)

r =
# Korrelationskoeffizient mit der cor() Funktion bestätigen


# Regressionssteigung: r mal sd von y dividiert durch die sd von x
b =

# Intercept: Mittelwert von y - (b mal Mittelwert von x)
k =

# Eingeschaetze Werte:
yhut =

# Error: Der Unterschied zwischen den tatsaechlichen und eingeschaetzen
# Werte
error =

# SSE: sum-of-squares (Error)
SSE =

# SSR: sum-of-squares (Regression)
SSR =

# SSY: sum-of-squares (Total)
SSY =

# Bestaetigung: SSY = SSR + SSE  (ja/nein?)

##  R^2 (R-squared) aus SSY und SSE berechnen
rsquared1 =

## R^2 mit der cor() Funktion berechnen
rsquared2 =
```

```
## Pruefen ob es eine eine signifikante lineare Beziehung
## zwischen x und y gibt (ob rsquared signifikant von 0 abweicht).
## critical ratio (tstat): r dividiert durch die Standardabweichung von
r
## die Standardabweichung von r ist
rsb = sqrt( (1 - r^2)/(n-2))
tstat

# Die F-statistik ist tstat hoch 2
fstat

# Die Wahrscheinlichkeit, dass die Werte nicht durch die
# Regressionslinie modelliert werden können
1 - pf(fstat, 1, n-2)

# Ergebnis
# Es gibt eine signifikante lineare Beziehung zwischen
# y und x
#
#

# Die Regressionlinie berechnen mit lm()


# x, y Werte abbilden und die Regressionslinie überlagern


# Die  Quantitaeten tstat, fstat, SSR/SSY
# hier idenfizieren


# Folgen die Residuals der Normalverteilung?



# Konstante Varianz der Residuals?


# Keine Autokorrelation?


# Wert von y vorhersagen, wenn x = 0.8

# Die Abbildung mit dem vorhergesagten Wert und Regressionslinie neu
malen



# 3. Für diese Daten wurde F2 - F1 (Hz) in einem Vokal
# zwischen 1910 und 1997 gemessen. Ändert sich F2-F1 mit der Zeit?
# Wenn ja, schätzen Sie den Wert von F2-F1 ein im Jahr 2012.
# Jahr F2-F1
# 1910    139
# 1920    149
# 1930    157
```

```
# 1940    175
# 1950    216
# 1959    303
# 1969    390
# 1978    449
# 1987    462
# 1997    487

# 4. Die Grundfrequenz wurde in der selben Person
# in einem Zeitraum von 10 Jahren gemessen.
# Der erste Werte ist aus 1987, der letzte aus 1996
# 137.0   131.2   127.1   123.4   119.2   114.6   109.6   104.5   99.4
95.3
# Ändert sich die Grundfrequenz mit der Zeit?
# Wenn ja, welchen Wert müsste f0 im Jahr 2000 gehabt haben?

# 5. Inwiefern kann die Intensität (dB) aus der Dauer vorhergesagt
werden,
# in den folgenden Vokalen produziert von 15 Sprechern:
```