```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)
```

```r
### 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)
my = mean(y)
mx = mean(x)
n = length(y)
covxy = cov(y,x)
# Korrelation gleicht die  Kovarianz
# dividiert durch  (sd von y Mal sd von x)
r = covxy / (sd (y) * sd(x))
# Korrelationskoeffizient mit der cor() Funktion bestätigen

# Regressionssteigung: r mal sd von y dividiert durch die sd von
x
b = r * sd(y) / sd(x)
# Intercept: Mittelwert von y - (b mal Mittelwert von x)
k = my - b * mx
# Eingeschaetze Werte:
yhut = b * x + k

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

# SSE: sum-of-squares (Error)
SSE = sum(error ^ 2)

# SSR: sum-of-squares (Regression)
SSR = sum((yhut - my)^2)

# SSY: sum-of-squares (Total)
SSY = sum((y - my)^2)

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

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

## R^2 mit der cor() Funktion berechnen
rsquared2 = cor(x, y)^2

## 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 = r / rsb

# Die F-statistik ist tstat hoch 2
fstat = tstat^2

# 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
#
# (R-squared = 0.88, F[1,18 = 128.9, p < 0.001) )

# Die Regressionlinie berechnen mit lm()
reg = lm(y ~ x)

# x, y Werte abbilden und die Regressionslinie überlagern
abline(reg)

pfun = function(x, y, ...)
{
    panel.xyplot(x, y, ...)
    panel.lmline(x, y)
    panel.points(x, yhut, col = "red")
}


xyplot(y ~ x, panel = pfun)

# Die  Quantitaeten tstat, fstat, SSR/SSY
# hier idenfizieren
summary(reg)

# Folgen die Residuals der Normalverteilung?
# Ja.
shapiro.test(resid(reg))

# Konstante Varianz der Residuals?
# Ja.
plot(resid(reg))
abline( h = 0)

# Keine Autokorrelation?
```

```r
# Keine Autokorrelation
acf(resid(reg))

# Wert von y vorhersagen, wenn x = 0.8
p = b * 0.8 + k
p = predict(reg, data.frame(x = 0.8))

# Die Abbildung mit dem vorhergesagten Wert und Regressionslinie
neu malen
xlim = c(0.7, 1.6)
ylim = c(85, 100)
plot(y ~ x, xlim = xlim, ylim = ylim)
abline(reg)
points(0.8, p, col = "red")


# 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
zeit = c(1910, 1920, 1930, 1940, 1950, 1959, 1969, 1978, 1987,
1997)
form = c(139, 149, 157, 175, 216, 303, 390, 449, 462, 487)

# Entweder
plot(form ~ zeit)
reg = lm(form ~ zeit)
abline(reg)

# Oder

pfun = function(x, y, ...)
{
    panel.xyplot(x, y, ...)
    panel.lmline(x, y)
    # panel.points(x, yhut, col = "red")
}

xyplot(form ~ zeit, panel=pfun)
```

```r
summary(reg)
shapiro.test(resid(reg))
plot(resid(reg))
acf(resid(reg))
p = predict(reg, data.frame(zeit = 2012))
ylim = c(100, 600); xlim = c(1910, 2015)
plot(form ~ zeit, xlim = xlim, ylim = ylim)
abline(reg)
points(2012, p, col = "red")



# 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?
jahr = seq(1987, 1996, by = 1)
f0 = c(137.0,   131.2,   127.1,   123.4 ,  119.2 ,  114.6,
109.6,   104.5 ,  99.4,    95.3)
plot(f0 ~ jahr)
reg = lm(f0 ~ jahr)
abline(reg)
summary(reg)
shapiro.test(resid(reg))
plot(resid(reg))
acf(resid(reg))
p = predict(reg, data.frame(jahr = 2000))
xlim = c(1985, 2005)
ylim = c(70, 140)
plot(jahr, f0, xlim = xlim, ylim = ylim)
abline(reg)
points(2000, p, col = 2)


# 5. Inwiefern kann die Intensität (dB) aus der Dauer
vorhergesagt werden,
# in den folgenden Vokalen produziert von 15 Sprechern:
dba = read.table(file.path(pfadu, "dba.txt"))
plot(dB ~ Dauer, data = dba)
reg = lm(dB ~ Dauer, data = dba)
abline(reg)
summary(reg)
shapiro.test(resid(reg))
plot(resid(reg))
acf(resid(reg))
```