

Übung 6

Jan Rabenseifner

2024-01-18

```
r = getOption("repos")
r["CRAN"] = "http://cran.us.r-project.org"
options(repos = r)

install.packages("rmarkdown")

## Installiere Paket nach 'C:/Users/janra/AppData/Local/R/win-library/4.3'
## (da 'lib' nicht spezifiziert)

## Paket 'rmarkdown' erfolgreich ausgepackt und MD5 Summen abgeglichen
##
## Die heruntergeladenen Binärpakete sind in
## C:\Users\janra\AppData\Local\Temp\RtmpE1N03g\downloaded_packages

library(rmarkdown)

## Warning: Paket 'rmarkdown' wurde unter R Version 4.3.2 erstellt

install.packages("matlib" )

## Installiere Paket nach 'C:/Users/janra/AppData/Local/R/win-library/4.3'
## (da 'lib' nicht spezifiziert)

## Paket 'matlib' erfolgreich ausgepackt und MD5 Summen abgeglichen
##
## Die heruntergeladenen Binärpakete sind in
## C:\Users\janra\AppData\Local\Temp\RtmpE1N03g\downloaded_packages

library(matlib)

## Warning: Paket 'matlib' wurde unter R Version 4.3.2 erstellt

# Übungsaufgabe 2

Jahr<-c(1,2,3,4,5,6,7,8)
Kurs<-c(20,25,36,48,64,86,114,168)

# Anpassung des nichtlinearen Regressionsmodells
NL.Modell<-nls(Kurs ~ beta.1*exp(beta.2 * Jahr), start = list(beta.1=15, beta.2=0.25),trace = TRUE)

## 4592.919      (7.63e+00): par = (15 0.25)
## 218.4818      (1.61e+00): par = (12.45084 0.3302673)
## 60.36954      (3.91e-02): par = (13.10317 0.3161399)
## 60.27457      (1.34e-03): par = (13.18495 0.3152461)
## 60.27446      (3.84e-05): par = (13.1877 0.3152186)
## 60.27446      (1.21e-06): par = (13.18778 0.3152177)
```

```
summary(NL.Modell)
```

```
##
## Formula: Kurs ~ beta.1 * exp(beta.2 * Jahr)
##
## Parameters:
##      Estimate Std. Error t value Pr(>|t|)
## beta.1 13.187785   0.868144   15.19 5.14e-06 ***
## beta.2  0.315218   0.009327   33.80 4.47e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.17 on 6 degrees of freedom
##
## Number of iterations to convergence: 5
## Achieved convergence tolerance: 1.209e-06
```

```
# Test different starting point
```

```
NL.Modell<-nls(Kurs ~ beta.1*exp(beta.2 * Jahr), start = list(beta.1=-10, beta.2=0.9), trace = TRUE)
```

```
## 2.212814e+08 (2.83e+02): par = (-10 0.9)
##      2785146. (3.11e+01): par = (1.158405 0.9129727)
## 318399.4      (1.26e+01): par = (1.071118 0.809125)
## 10639.30      (3.96e+00): par = (1.977111 0.6074596)
## 2708.560      (3.81e+00): par = (3.833851 0.4672136)
## 2276.133      (4.53e+00): par = (5.354344 0.4180664)
## 1950.974      (5.37e+00): par = (8.370608 0.352595)
## 438.1874      (2.50e+00): par = (12.84113 0.3069191)
## 60.92133      (1.04e-01): par = (13.19873 0.3155775)
## 60.27447      (5.53e-04): par = (13.18672 0.3152302)
## 60.27446      (1.69e-05): par = (13.18775 0.3152181)
## 60.27446      (5.18e-07): par = (13.18779 0.3152177)
```

```
summary(NL.Modell)
```

```
##
## Formula: Kurs ~ beta.1 * exp(beta.2 * Jahr)
##
## Parameters:
##      Estimate Std. Error t value Pr(>|t|)
## beta.1 13.187786   0.868144   15.19 5.14e-06 ***
## beta.2  0.315218   0.009327   33.80 4.47e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.17 on 6 degrees of freedom
##
## Number of iterations to convergence: 11
## Achieved convergence tolerance: 5.179e-07
```

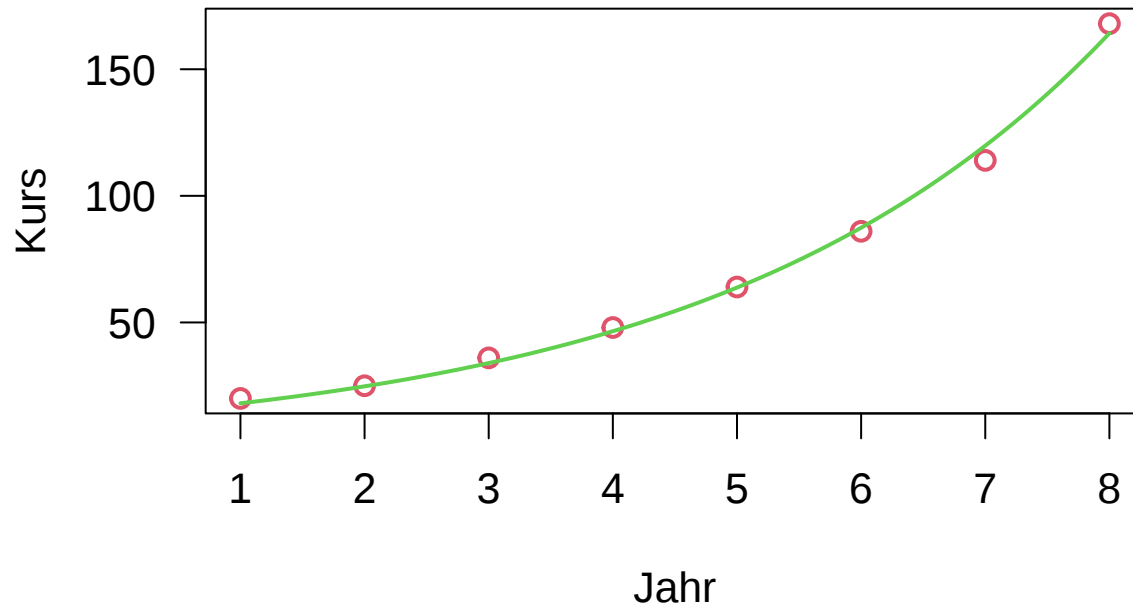
```
# Streudiagramm für die Daten und die angepasste Funktion
```

```
par(mfrow=c(1,1),cex=1.3)
```

```
plot(Jahr, Kurs, type='p', xlab=expression(Jahr), ylab=expression(Kurs), col=2, lwd=2, las=1)
```

```
x<-seq(1, 8, length=100)
```

```
lines(x, coef(NL.Modell)[1]*exp(coef(NL.Modell)[2] * x), col=3, lwd=2)
```



```
# Schätzung des Varianzparameters
y_hat<-coef(NL.Modell)[1]*exp(coef(NL.Modell)[2] * Jahr)
Varianz<-1/(length(Jahr)-2)*sum((y_hat-Kurs)^2)
sigma<-sqrt(Varianz)

library(matlib)
# Schätzung für die asymptotische Varianz-Kovarianzmatrix
a<-sum(exp(2*coef(NL.Modell)[2]*Jahr))
b<--coef(NL.Modell)[1]*sum(Jahr*exp(2*coef(NL.Modell)[2]*Jahr))
c<-coef(NL.Modell)[1]^2*sum(Jahr^2*exp(2*coef(NL.Modell)[2]*Jahr))

A <- matrix( c(a,b,
               b,c), nrow=2, byrow=TRUE)
AI<-inv(A)
Sigma<-Varianz*AI

printmat <- function(mat) {
  out <- capture.output(mat)
  out[1] <- paste0("Kovarianz:", out[1])
  cat(paste(out, collapse = "\n"))
}
printmat(Sigma)

## Kovarianz:          [,1]          [,2]
## [1,] 0.753673699 7.931717e-03
## [2,] 0.007931717 8.699613e-05
```

```
mm <- list(Kovarianz = Sigma)
class(mm) <- "listof"
mm
```

```
## Kovarianz :
##           [,1]      [,2]
## [1,] 0.753673699 7.931717e-03
## [2,] 0.007931717 8.699613e-05
```

```
# Least Squares starting points from linearized modell
LS.modell<-lm(log(Kurs)~Jahr)
summary(LS.modell)
```

```
##
## Call:
## lm(formula = log(Kurs) ~ Jahr)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.043476 -0.021872 -0.002798  0.023123  0.048440
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.657961    0.027811   95.57 8.84e-11 ***
## Jahr         0.302195    0.005507   54.87 2.46e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.03569 on 6 degrees of freedom
## Multiple R-squared:  0.998, Adjusted R-squared:  0.9977
## F-statistic: 3011 on 1 and 6 DF, p-value: 2.46e-09
```

```
# Back-transformed intercept
beta.1.ls <- exp(coef(LS.modell)[1])
beta.2.ls <- coef(LS.modell)[2]
beta.1.ls
```

```
## (Intercept)
##      14.26717
beta.2.ls
```

```
##      Jahr
## 0.3021953
```

```
# Anpassung des nichtlinearen Regressionsmodells
```

```
NL.Modell<-nls(Kurs ~ beta.1*exp(beta.2 * Jahr), start = list(beta.1=beta.1.ls, beta.2=beta.2.ls),trace=
```

```
## 86.46763    (6.48e-01): par = (14.26717 0.3021953)
## 60.73467    (8.73e-02): par = (13.19281 0.3147624)
## 60.27448    (5.85e-04): par = (13.18907 0.3152042)
## 60.27446    (1.82e-05): par = (13.18783 0.3152173)
## 60.27446    (5.93e-07): par = (13.18779 0.3152177)
```

```
summary(NL.Modell)
```

```
##
## Formula: Kurs ~ beta.1 * exp(beta.2 * Jahr)
```

```
##
## Parameters:
##      Estimate Std. Error t value Pr(>|t|)
## beta.1 13.187788   0.868144   15.19 5.14e-06 ***
## beta.2  0.315218   0.009327   33.80 4.47e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.17 on 6 degrees of freedom
##
## Number of iterations to convergence: 4
## Achieved convergence tolerance: 5.928e-07
```