Lösungen zu den Übungsaufgaben

Übung 2.1

```
>x<-seq(0,3,0.1)
>plot(x,log(x,base=2),type="l")
>lines(x,log(x,base=3),lty=2,col="Red",lwd=2)
>lines(x,log(x,base=exp(1)),lty=3,col="Blue",lwd=2)
>lines(x,log(x,base=10),lty=4,col="green",lwd=2)
>legend(1.5,-1,c("Basis 2","Basis 3", "Basis e","Basis 10"),
+ lty=1:4,col=c("Black","Red","Blue","green"),lwd=2)
```

Übung 2.2

```
>x<-seq(-3,4,0.1)
>plot(x,dnorm(x,mean=1,sd=1),type="l")
>lines(x,dnorm(x,mean=0,sd=1),lty=2)
>legend(-3,0.4,c("mean=1","mean=0"),lty=c(1,2))
```

Die anderen Grafiken werden analog mit den entsprechenden Parametern erstellt.

Übung 3.1

Bsp.	Merkmalsträger	Merkmal	Merkmalstyp
1.0.1	Kugeln	Durchmesser	stetig
1.0.2	Glühbirne	Lebensdauer	stetig
1.0.3	Rostschutzmittel	Rostanfälligkeit	ordinal
1.0.4	Penicillin	Herstellungsmethode	nominal
		Ertrag	diskret
1.0.5	Zement	Zusammensetzung, Hitze	stetig
1.0.6	Kammgarn	Länge, Amplitude, Gewicht	ordinal
		Anzahl Wicklungen	diskret
1.0.7	Beton	Druck, Rohdichte	stetig
		Orte	nominal
1.0.8	Fahrzeuge	defekt, nicht defekt	nominal

Übung 4.1

```
> druck<-read.csv2("Druckfestigkeit.csv")
> table(druck$S)

   Aachen Darmstadt   Dresden Karlsruhe    Kassel   Leipzig
       12        12        12        11        66        12
> pie(table(druck$S))
```

Übung 4.2

```
> autos<-c(2,2,3,3,2,1,5,6,6,5,4,3,2,1,2)
> plot(table(autos)/length(autos))
```

Übung 4.3

```
> stahl<-scan("STEEL.DAT")
Read 20 items
> linie1<-stahl[1:10]
> linie2<-stahl[11:20]
> par(mfrow=c(1,2))
> hist(linie1,breaks=seq(0.6,2,0.2),include.lowest=T)
> hist(linie2,breaks=seq(0.6,2,0.2),include.lowest=T)
> plot(ecdf(linie1))
> plot(ecdf(linie2))
> hist1<-hist(linie1,breaks=seq(0.6,2,0.2),include.lowest=T,plot=F)
> hist2<-hist(linie2,breaks=seq(0.6,2,0.2),include.lowest=T,plot=F)
> hist1$counts
[1] 1 1 4 1 2 1 0
> hist2$counts
[1] 2 1 0 1 1 4 1
> hist1$counts/length(linie1)
[1] 0.1 0.1 0.4 0.1 0.2 0.1 0.0
> hist2$counts/length(linie2)
```

```
[1] 0.2 0.1 0.0 0.1 0.1 0.4 0.1
> cumsum(hist1$counts/length(linie1))
[1] 0.1 0.2 0.6 0.7 0.9 1.0 1.0
> cumsum(hist2$counts/length(linie2))
[1] 0.2 0.3 0.3 0.4 0.5 0.9 1.0
```

Übung 5.1

x_{mod} =Kassel

Übung 5.2

```
> autos<-c(2,2,3,3,2,1,5,6,6,5,4,3,2,1,2)
> quantile(autos,c(0.25,0.5,0.75))
25% 50% 75%
2.0 3.0 4.5
```

Übung 5.3

```
> #Gesamtdaten
> summary(stahl)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  0.690   1.012   1.305   1.300   1.620   1.960
> #Linie 1
> summary(linie1)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  0.690   1.038   1.185   1.194   1.395   1.620
> #Linie 2
> summary(linie2)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  0.770   1.065   1.530   1.406   1.712   1.960
```

Übung 5.4

```
> cracks<-read.table("Crackcounts.dat")
> cracks0<-cracks[,"T0"]
> cracks5<-cracks[,"T5"]
> cracks10<-cracks[,"T10"]
> cracks18<-cracks[,"T18"]
> par(mfrow=c(2,2))
> hist(cracks0,breaks=seq(0,1400,100),
+ main="Histogramm der Risse in T0")
> hist(cracks5,breaks=seq(0,1400,100),
+ main="Histogramm der Risse in T5")
> hist(cracks10,breaks=seq(0,1400,100),
+ main="Histogramm der Risse in T10")
> hist(cracks18,breaks=seq(0,1400,100),
+ main="Histogramm der Risse in T18")
> cbind(median(cracks0),median(cracks5),median(cracks10),
+ median(cracks18))
     [,1] [,2] [,3]  [,4]
[1,]   40  605  881 919.5
> cbind(mean(cracks0),mean(cracks5),mean(cracks10),mean(cracks18))
         [,1]     [,2]  [,3]     [,4]
[1,] 79.98148 584.2778 841.5 907.6481
```

Übung 6.1

```
> setting<-read.table("SETTING.DAT")
> a<-setting[,1]
> b<-setting[,2]
> c<-setting[,3]
> d<-setting[,4]
> quantile(a,p=c(0.25,0.5,0.75),type=2)
25% 50% 75%
  2   7  11
> mean(a)
[1] 7.461538
> max(a)-min(a)
```

```
[1] 20
> quantile(a,p=0.75,type=2)-quantile(a,p=0.25,type=2)
75%
  9
> mad(a,constant=1)
[1] 4
> sd(a)
[1] 5.882394
> var(a)
[1] 34.60256
> 2*(quantile(a,p=0.75,type=2)-quantile(a,p=0.25,type=2))/
+ (quantile(a,p=0.75,type=2)+quantile(a,p=0.25,type=2))
     75%
1.384615
> sd(a)/mean(a)
[1] 0.7883621
#Für b, c, d analog
```

Übung 6.2

```
> boxplot(list(linie1,linie2),names=c("Linie1","Linie2"))
> max(linie1)-min(linie1)
[1] 0.93
> max(linie2)-min(linie2)
[1] 1.19
> quantile(linie1,p=0.75,type=2)-quantile(linie1,p=0.25,type=2)->Q1
> quantile(linie2,p=0.75,type=2)-quantile(linie2,p=0.25,type=2)->Q2
> mad(linie1,constant=1)
[1] 0.2
> mad(linie2,constant=1)
[1] 0.25
> sd1<-sqrt(1/(length(linie1)-1)*sum((linie1-mean(linie1))^2))
> sd2<-sqrt(1/(length(linie2)-1)*sum((linie2-mean(linie2))^2))
> 1/(length(linie1)-1)*sum((linie1-mean(linie1))^2)
[1] 0.08391556
> 1/(length(linie2)-1)*sum((linie2-mean(linie2))^2)
[1] 0.1834489
> 2*Q1/(quantile(linie1,p=0.75,type=2)+
```

```
+ quantile(linie1,p=0.25,type=2))
      75%
0.3278689
> 2*Q2/(quantile(linie2,p=0.75,type=2)+
+ quantile(linie2,p=0.25,type=2))
      75%
0.5387454
> sd1/mean(linie1)
[1] 0.2426146
> sd2/mean(linie2)
[1] 0.3046297
```

Übung 6.3

```
>boxplot(cracks[,1:15],main="Boxplots der Rissanzahlen",
+ ylab="Anzahl der Risse",names=c("T0","T1","T2","T3","T4","T5"
+ ,"T6","T7","T8","T9","T10","T12","T14","T16","T18"))
```

Übung 7.1

```
>plot(cracks$T5,cracks$T10)
>plot(cracks$T10,cracks$T18)
```

Übung 7.2

```
> rost<-matrix(c(65,103,106,74,85,47),ncol=3,byrow=T)
> chi2<-chisq.test(rost)$statistic
> sqrt(chi2/(chi2+sum(rost))*2/1)
X-squared
0.2519967
```

Übung 7.3

```
> x<-c(4,9,2,8,7)
> y<-c(28,7,5,24,16)
> cor(x,y)
[1] 0.0762274
> cor(rank(x),rank(y))
[1] 0.1
```

Übung 7.4

```
> setting<-read.table("SETTING.DAT")
> b<-setting[,2]
> d<-setting[,4]
> plot(b,d)
> cor(b,d)
[1] -0.972955
> cor(rank(b),rank(d))
[1] -0.9903458
```

Übung 7.5

```
> plot(cracks$T5,cracks$T10)
> cor(cracks$T5,cracks$T10)
[1] 0.582104
> cor(rank(cracks$T5),rank(cracks$T10))
[1] 0.562612
> lsfit(cracks$T5,cracks$T10)$coef
 Intercept          X
454.172324   0.662917
> abline(lsfit(cracks$T5,cracks$T10)$coef,lwd=2,col="red")
```

Genauso für die Zeitpunkte 10 und 18.

Übung 7.6

$r_{xy} = r_{xy}^{Sp} = -1$	$0 < r_{xy} < 1,\ r_{xy}^{Sp} = 1$
$r_{xy} = r_{xy}^{Sp} = 0$	$-1 < r_{xy}, r_{xy}^{Sp} < 0$

Übung 8.1

```
> x100<-rnorm(100,mean=0,sd=1)
> hist(x100)
> lines(seq(-2,2,0.1),dnorm(seq(-2,2,0.1),mean=0,sd=1))
> mean(x100)
[1] 0.1255314
> median(x100)
[1] 0.07157136
> var(x100)
[1] 1.147626
> sd(x100)
[1] 1.071273
```

Genauso für $N = 10000$ und andere Parameter. Vergleich der Kennzahlen mit „realen“ Werten zeigt, dass die Unterschiede mit wachsendem N immer kleiner werden.

Übung 8.2

```
> N<-10000
> x<-rnorm(N,mean=1,sd=1)
> #(a)
> sum(x<=-2)/N
[1] 0.0012
> #(b)
> sum(x>-5)/N
[1] 1
> #(c)
> sum((x>-5) & (x<=-2))/N
[1] 0.0012
```

```
> #(d)
> sum((x>-2) & (x<=4))/N
[1] 0.997
> #(e)
> sum((x<=-2) | (x>4))/N
[1] 0.003
```

Übung 10.1

```
"wuerfel" <-
function (N)
{
# Simuliert N Wuerfelwuerfe und gibt die
# relative Häufigkeit der Sechs aus
u<-runif(N)
u[u<=1/4]<-6
u[1/4<u & u<=1/4+3/20]<-5
u[1/4+3/20<u & u<=1/4+6/20]<-4
u[1/4+6/20<u & u<=1/4+9/20]<-3
u[1/4+9/20<u & u<=1/4+12/20]<-2
u[1/4+12/20<u & u<=1]<-1
list(Wuerfelergebnisse=u,Anteil=sum(u==6)/N)
}
```

Für den fairen Würfel wird das Intervall $[0, 1]$ in gleichgroße Teile der Länge $1/6$ aufgeteilt. Für den verfälschten Würfel wird das Intervall so aufgeteilt, dass für eine Zahl in $[0, 1/4)$ eine 6 ausgegeben wird (entspricht Wahrscheinlichkeit $1/4$). Das restliche Intervall wird nun gleichmäßig in 5 Teilintervalle der Länge $3/4 \cdot 1/5 = 3/20$ aufgeteilt. Damit sind die anderen Zahlen alle gleichwahrscheinlich.

Übung 10.2

(a)$P(A \cup B) = \frac{31}{200}$ Fehler A oder Fehler C treten ein, (b) $P(A \cap \overline{C}) = \frac{2}{25}$ Fehler A und nicht Fehler C tritt auf, (c) $P((\overline{A} \cup \overline{B}) \cap C) = \frac{199}{1000}$ Fehler C und nicht Fehler A oder B, (d) $P(A \cup B \cup C) = \frac{316}{1000}$ Ausfall der Maschine.

Übung 11.1

```
"wuerfel2" <-
function (N)
{
# Simuliert N Wuerfelwuerfe und gibt die
# relative Häufigkeit der Sechs aus
u<-runif(N)
u[u<=1/5]<-6
u[5/25<u & u<=9/25]<-5
u[9/25<u & u<=13/25]<-4
u[13/25<u & u<=17/25]<-3
u[17/25<u & u<=21/25]<-2
u[21/25<u & u<=1]<-1
list(Wuerfelergebnisse=u)
}
plot(ecdf(x2)$Wuerfelergebnisse)
```

Übung 11.2

(a) $P(X < 2) = 0$

(b) $P(X \leq 2) = \frac{1}{2}$

(c) $P(X \leq 3) = \frac{2}{3}$

(d) $P(X > 3) = \frac{1}{3}$

(e) $P(2 < X \leq 3) = \frac{1}{6}$

(f) $P(X \notin (2,3]) = \frac{5}{6}$

(g) $P(2 < X < 3) = 0$

(h) $P(X \in [2,3]) = \frac{2}{3}$

Übung 12.1

```
>N<-1000
>#Poisson
>lam<-1
>dat<-rpois(N,lam)
>plot(table(dat)/N,type="h",ylim=c(0,0.4))
>lines((0:6)+0.05,dpois(0:6,lam),type="h",col="red",lwd=3)
>#genauso für andere Werte lam=lambda
>
>#Exponential
>lam<-1/2
>dat<-rexp(N,lam)
>hist(dat,freq=F)
>lines(seq(0,15,0.1),dexp(seq(0,15,0.1),lam),col="red")
>
>#Weibull
>a<-1
>b<-1
>dat<-rweibull(N,a,b)
>hist(dat,xlim=c(0,10),freq=F,ylim=c(0,1))
>#Hinzufügen der Dichte
>lines(seq(0,14,0.1),dweibull(seq(0,14,0.1),a,b),col="red")
>
>#Gamma
>a<-1
>b<-1
>dat<-rgamma(N,a,1/b)
>hist(dat,xlim=c(0,10),ylim=c(0,1),freq=F)
>lines(seq(0,10,0.1),dgamma(seq(0,10,0.1),a,1/b),col="red")
>
>#Chi^2
>n<-2
>dat<-rchisq(N,df=n)
>hist(dat,xlim=c(0,40),ylim=c(0,0.5),freq=F)
>lines(seq(0,40,0.1),dchisq(seq(0,40,0.1),n),col="red")
>
>#t
>n<-2
```

```
>dat<-rt(N,n)
>hist(dat,freq=F,ylim=c(0,0.5),xlim=c(-10,10))
>lines(seq(-40,40,0.1),dt(seq(-40,40,0.1),n),col="red")
```

Übung 13.1

```
>source("d2norm.asc")
>source("d2norm_plot.asc")
>fix("d2norm_plot.asc")
```

1. Ändere die Funktion wie folgt ab: Für den perspektivischen Plot muss das Zeichen `#` vor `persp` entfernt werden und vor die beiden übrigen Plottypen gesetzt werden. Analog muss für den Konturplot das Kommentarzeichen vor `contour` entfernt und bei den anderen beiden Funktionen eingefügt werden und für `image` wird ebenfalls so verfahren. Da wir alle Plottypen haben möchten können wir auch par(mfrow=c(1,3)) in die Funktion aufnehmen und alle `#` vor den Plottypen entfernen. Bei dieser Änderung ist es sinnvoll den `title` Befehl zu deaktivieren, damit der Titel nicht nur über dem rechten Plot erscheint. Nach den Änderungen wird das Fenster mit der Funktion geschlossen (speichern nicht vergessen) und die Funktion in der R-Console über

```
d2norm.plot(m1=0,m2=1,sd1=1,sd2=1,rho=0)
```

aufgerufen.

2.-5.: Wir ändern die Funktion wieder so ab, dass die Plotfunktionen bis auf `contour` auskommentiert werden. Gegegebenfalls setzen wir `par(mfrow=c(1,1))`. Außerdem ändern wir den Bereich für die y-Achse zu `y<-seq(-5,5,0.1)`. Dann (nach schließen und speichern) wird die Funktion mit den angegebenen Paramtern aufgerufen.

Übung 14.1

Wir benutzen den Satz von der totalen Wahrscheinlichkeit und die Formel von Bayes und erhalten:
(a) Die Wahrscheinlichkeit für ein defektes Teil: $P(D) = 0.15$.
(b) Die Wahrscheinlichkeit, dass das defekte Teil von Firma A stammt: $P(A|D) = 0.2$.

Übung 14.2

$$\mathsf{P}(X \geq 3) = 0.05,\ \mathsf{P}\left(X \geq \frac{1}{2}\right) = 0.61,\ \mathsf{P}(X \geq 1) = 0.37,$$
$$\mathsf{P}\left(X \geq 1 | X \geq \frac{1}{2}\right) = 0.61,\ \mathsf{P}(X \geq 2|X \geq 1) = 0.37,$$
$$\mathsf{P}(X \geq 4|X \geq 1) = 0.05.$$

Übung 14.3

(a)

$\mathsf{P}(X = x_j, Y = y_k)$		y_k 1	2	3	$\mathsf{P}(X = x_j)$
	1	$\frac{1}{20}$	0	0	0.05
	2	$\frac{3}{20}$	$\frac{1}{20}$	$\frac{1}{20}$	0.25
x_j	3	$\frac{3}{20}$	$\frac{1}{10}$	$\frac{1}{20}$	0.3
	4	$\frac{1}{20}$	$\frac{1}{5}$	$\frac{3}{20}$	0.4
$\mathsf{P}(Y = y_k)$		$\frac{2}{5}$	$\frac{7}{20}$	$\frac{1}{4}$	1

(b) X und Y sind nicht stochastisch unabhängig, da $P(X = 1, Y = 1) = \frac{1}{20} \neq \frac{2}{5}\frac{1}{20} = P(X = 1) \cdot P(Y = 1)$

Übung 14.4

Wir definieren $T_i = 0$: das i-te Triebwerk fällt nicht aus, $T_i = 1$: das i-te Triebwerk fällt aus, $i = 1, 2$ bzw. $i = 1, 2, 3, 4$. Damit ist T_i Bernoulli-verteilt und die Summe binomialverteilt (Bin(N,p)). Wir erhalten also

- Wahrscheinlichkeit für Flugfähigkeit bei 2 Triebwerken und Ausfallwahrscheinlichkeit eines Triebwerks 0.5: 0.75
- Wahrscheinlichkeit für Flugfähigkeit bei 4 Triebwerken und Ausfallwahrscheinlichkeit eines Triebwerks 0.5: 0.6875
- Wahrscheinlichkeit für Flugfähigkeit bei 2 Triebwerken und Ausfallwahrscheinlichkeit eines Triebwerks 0.6: 0.64
- Wahrscheinlichkeit für Flugfähigkeit bei 4 Triebwerken und Ausfallwahrscheinlichkeit eines Triebwerks 0.6: 0.5248

Übung 15.1

$\mathsf{E}(X) = 3.05$, $\mathsf{E}(Y) = 1.85$, $\mathsf{var}(X) = 0.8475$, $\mathsf{var}(Y) = 0.6275$

Übung 15.2

Es sei f die Dichte der Dreiecksverteilung. Da f eine stetige Dichte ist, muss gelten: $\int_{-\infty}^{\infty} f(x)dx = 1$, hier entspricht das Integral gerade der Fläche des Dreiecks. Wir nehmen an, dass das Dreieck gleichschenklig ist und die Spitze über dem Punkt $(1, 0)$ liegt. Damit erhalten wir die Höhe h des Dreiecks über $1 = \int_{-\infty}^{\infty} f(x)dx = \frac{1}{2}2 \cdot h$, also $h = 1$. Damit hat f die folgende Form:

$$f(x) = \begin{cases} x, & 0 \leq x \leq 1 \\ -x + 2, & 1 < x \leq 2 \\ 0, & sonst \end{cases}.$$

Und wir berechnen $\mathsf{E}(X) = 1$, $\mathsf{var}(X) = \frac{1}{6}$.

Übung 15.3

```
> w10<-sample(1:6,10,replace=TRUE)
> w100<-sample(1:6,100,replace=TRUE)
> w10000<-sample(1:6,10000,replace=TRUE)
> mean(w10)
[1] 3.2
> mean(w100)
[1] 3.85
> mean(w10000)
[1] 3.5065
```

Der Erwartungswert der Augenzahl beim Würfelwurf ist 3.5.

Übung 16.1

$Q_{0.25}(X) = 2$, $Q_{0.75} = 5$.

Übung 16.2

```
> qnorm(0.05)
[1] -1.644854
> qnorm(0.95)
[1] 1.644854
> qnorm(0.05,2,1)
[1] 0.3551464
> qnorm(0.95,2,1)
[1] 3.644854
> qnorm(0.05,0,2)
[1] -3.289707
> qnorm(0.95,0,2)
[1] 3.289707
```

Übung 16.3

$\mathsf{kov}(X,Y) = \frac{123}{400} = 0.3075$, $\mathsf{korr} = \frac{123}{\sqrt{339 \cdot 251}} = 0.422$.

Übung 17.1

$$r_S = 1 - p_1(1-p_2)(1-p_3) = \begin{cases} 0.9028 & \text{für } p_1 = 0.1 \\ 0.95149 & \text{für } p_1 = 0.05 \end{cases}.$$

Übung 17.2

$$\begin{aligned} \lambda(t) &= \frac{f(t)}{R(t)} = \frac{\frac{\alpha}{\beta^\alpha} t^{\alpha-1} \exp\left(-\left(\frac{t}{\beta}\right)^\alpha\right)}{\exp\left(-\left(\frac{t}{\beta}\right)^\alpha\right)} \\ &= \frac{\alpha}{\beta^\alpha} t^{\alpha-1}. \end{aligned}$$

Übung 18.1

Simulation der Markovketten:

```
> "UeMatFahrzeuge" <-
+ function (p=0.4,k=6)
+ {
+ # Berechnet Uebergangsmatrix mit den
+ # Übergangswahrscheinlichkeiten für das Fahrzeuge Beispiel
+ uemat<-dbinom(0:k,size=k,prob=p)
+ uemat<-c(uemat,uemat)
+ for(j in 2:k){
+  uemat<-c(uemat,rep(0,j-1),dbinom(0:(6-j+1),size=6-j+1,prob=p))
+    }
+ uemat<-matrix(uemat,ncol=(k+1),byrow=F)
+ uemat
+ }
> "MarkovKette" <-
+ function (uemat, N=4, start=1)
+ {
+ # Simuliert eine Markovkette mit der Übergangsmatrix uemat
```

```
+ # für die Zustände 0,1,...,ncol(uemat)-1
+ # Startwert:
+ i<- start
+ mk<-i-1
+ for(n in 1:N){
+   u<-runif(1)
+   for(l in 1:ncol(uemat)){
+     if(i == l){
+        psum<-cumsum(uemat[,l])
+        for(k in 1:nrow(uemat)){
+            if(k==1){
+                 if(u<=psum[1])
+                 {j<-1}
+            }
+            else{
+                 if(psum[k-1]<u & u<=psum[k])
+                 {j<-k}
+            }
+        }
+     }
+   }
+   i<-j
+   mk<-c(mk,i-1)
+ }
+ list(Markovkette=mk)
+ }
>  MarkovKette(UeMatFahrzeuge(0.1),N=20,start=1)
$Markovkette
 [1] 0 1 2 4 3 2 1 0 0 0 2 2 1 1 1 1 0 1 3 2 1

>  MarkovKette(UeMatFahrzeuge(0.3),N=20,start=1)
$Markovkette
 [1] 0 0 3 3 3 4 4 5 5 5 5 4 3 3 4 3 4 5 4 4 4
```

Invariante Verteilung:

```
> "InvariantesP" <-
+ function (UeMat=UeMatFahrzeuge(0.4),epsilon=0.00001,M=100,N=100)
```

```
+ {
+ # Bestimmt die invariante Verteilung der
+ # Markovkette gegen durch die Uebergangsmatrix UeMat
+ # 1. durch Approximation mit maximal M Schritten und
+ #    Genauigkeit epsilon
+ # 2. exakt über Bestiimung des Eigenvektors zum Eigenwert 1
+ # 3. durch Simulation der Markovkette über N Perioden
+ k<-ncol(UeMat)
+ par(mfrow=c(1,3))
+ # Approximation der invarianten Verteilung
+ p<-rep(1,k)/k
+ p1<-UeMat%*%p
+ m<-1
+ while(sum((p-p1)^2)>epsilon & m<M){
+     p<-p1
+     p1<-UeMat%*%p1
+     m<-m+1
+ }
+ cat("Anzahl der Durchlaeufe bis Abbruch: ",m,"\n")
+ p1<-as.vector(p1)
+ names(p1)<-as.character(((1:k)-1))
+ barplot(p1,main="Approx.  invariante Verteilung")
+ # Berechnung der exakten invarianten Verteilung
+ Qeigen<-eigen(UeMat)
+ q<-eigen(UeMat)$vectors[,Qeigen$values<=1+epsilon
+            & Qeigen$values>=1-epsilon]
+ q<-q/sum(q)
+ names(q)<-as.character(((1:k)-1))
+ barplot(q,main="Invariante Verteilung")
+ # Simulation der invarianten Verteilung
+ r<-table(MarkovKette(UeMat,N))/N
+ barplot(r,main="Simulierte invariante Verteilung")
+ list(Approximiert=p1,Exakt=q,Simuliert=r)
+ }
> InvariantesP(UeMatFahrzeuge(0.1),M=100,N=1000)
Anzahl der Durchlaeufe bis Abbruch:  13
$Approximiert
           0            1            2            3            4
4.233084e-01 3.747867e-01 1.559282e-01 3.924438e-02 6.163545e-03
           5            6
```

```
5.483200e-04 2.044907e-05

$Exakt
           0            1            2            3            4
4.252059e-01 3.748940e-01 1.548264e-01 3.855166e-02 5.977630e-03
           5            6
5.249881e-04 1.936782e-05

$Simuliert

    0     1     2     3     4
0.459 0.376 0.133 0.030 0.003

> InvariantesP(UeMatFahrzeuge(0.3),M=100,N=1000)
Anzahl der Durchlaeufe bis Abbruch:  8
$Approximiert
          0           1           2           3           4
0.004487423 0.032285472 0.121725455 0.269837214 0.332188358
          5           6
0.198041882 0.041434197

$Exakt
         0          1          2          3          4          5
0.00402231 0.03016676 0.11794895 0.26817058 0.33541759 0.20183756
         6
0.04243625

$Simuliert

    0     1     2     3     4     5     6
0.007 0.027 0.112 0.249 0.347 0.214 0.045
```

Die Wahrschinlichkeit, dass an einem Tag alle Fahrzeuge defekt sind, ist damit für $p = 0.1$ ungefähr 0 und für $p = 0.3$ ungefähr 0.042.

Die erwartete Anzahl defekter Fahrzeuge berechnet sich für $p = 0.1$ zu ≈ 0.827 mit Streuung ≈ 0.8758 und für $p = 0.3$ erhalten wir den Erwartungswert ≈ 3.676 und die Streuung ≈ 1.1586.

Übung 18.2

Die Übergangsmatrix hat in diesem Fall die folgende Form:

$$P = \begin{pmatrix} 0.2 & 0.2 & 0.5 & 0.2 \\ 0.3 & 0.3 & 0.3 & 0.3 \\ 0.3 & 0.3 & 0.2 & 0.3 \\ 0.2 & 0.2 & 0 & 0.2 \end{pmatrix}.$$

Zur Überprüfung, welche Verteilung π^j die invariante Verteilung am besten annähert bestimme $P\pi^j$ und prüfe ob $P\pi^j \approx \pi^j$, zusätzlich prüfe, ob die Summe der Einträge von π^j gleich Eins ist. Damit bleibt nur π^5 als invariante Verteilung.

Übung 20.1

Mögliche Schätzungen wären $\hat{k}_1(x_1, \ldots, x_N) = x_1$, also das Ergebnis des ersten Wurfs oder $\hat{k}_2(x_1, \ldots, x_N) = 2\overline{x} - 1$. Für das Datenbeispiel erhalten wir $\hat{k}_1(8, 6, 8, 5, 7, 3, 6) = 8$ und $\hat{k}_2(8, 6, 8, 5, 7, 3, 6) = 11.28571$. Die erste Schätzung ist nicht erwartungstreu, da $\mathsf{E}(X_1) = \frac{k-1}{2}$. Die zweite Schätzung hingegen ist erwartungstreu, da $\mathsf{E}(2\overline{X} - 1) = 2\mathsf{E}(X) - 1 = 2\frac{k+1}{2} - 1 = k$.

Übung 20.2

(a) ist eine erwartungstreue und konsistente Schätzung für den Erwartungswert. (b) ist eine nicht erwartungstreue aber konsistente Schätzung für die Varianz. (c) ist eine erwartungstreue und konsistente Schätzung für die Varianz und (d) ist eine erwartungstreue aber nicht konsistente Schätzung für den Erwartungswert.

Übung 22.1

Wir nehmen an, dass die Blechdicken normalverteilt sind und benutzen den t-Test.

```
> BLECH<-c(346,363,360,318,346,268,299,287,310,349,333,
 +365,281,265,344)
```

```
> t.test(BLECH,alternative="two.sided",mu=310)$p.value
[1] 0.1969823
```

Damit können wir H_0 nicht ablehnen.

Simulation des P-Werts für $\sigma = 1$:

```
> N<-length(BLECH)
> mu0<-310
> M<-10000
> sigma<-1
> ts<-numeric(M)
> for (i in (1:M))
+ {
+  testdaten<-rnorm(N,mu0,sigma)
+  ts[i]<-sqrt(N)*abs(mean(testdaten-mu0))/sd(testdaten)
+ }
> TS<-sqrt(N)*abs(mean(BLECH)-310)/sd(BLECH)
> 1/M*sum(ts>=TS)
[1] 0.1993
```

Für $\sigma = 30$ ergibt sich ein simulierter P-Wert von 0.1929. Für größeres σ erhalten wir also einen kleineren P-Wert.

Übung 22.2

1. Lehne H_0 ab, falls die Teststatistik $T(x) > t_{N-1;1-\frac{\alpha}{2}}$.

   ```
   > TS>qt(1-0.05/2,df=N-1)
   [1] FALSE
   ```

2. ```
   >  mu<-300
   > sigma<-1
   > mu0<-310
   > ncp1<-sqrt(N)*(mu-mu0)/sigma
   ```
   ```

```
> pt(qt(1-0.05/2,df=N-1),df=N-1,ncp=ncp1)-
+ pt(qt(1-0.05/2,df=N-1),df=N-1,ncp=ncp1)
[1] 0
```

3.
```
> sigma<-30
> ncp2<-sqrt(N)*(mu-mu0)/sigma
> pt(qt(1-0.05/2,df=N-1),df=N-1,ncp=ncp2)-
+ pt(-qt(1-0.05/2,df=N-1),df=N-1,ncp=ncp2)
[1] 0.7745203
```

4.
```
>ts<-numeric(M)
>sigma<-1
>for (i in (1:M))
+{
+ testdaten<-rnorm(N,mu,sigma)
+ ts[i]<-sqrt(N)*abs(mean(testdaten-mu0))/sd(testdaten)
+}
>1/M*sum(ts<=qt(1-0.05/2,df=N-1))
[1] 0
```

5.
```
> sigma<-30
> ts<-numeric(M)
> for (i in (1:M))
+  {
+  testdaten<-rnorm(N,mu,sigma)
+  ts[i]<-sqrt(N)*abs(mean(testdaten-mu0))/sd(testdaten)
+  }
> 1/M*sum(ts<=qt(1-0.05/2,df=N-1))
[1] 0.7735
```

6.
```
>mu<-seq(305,315,0.1)
>ncps<-sqrt(N)*(mu-mu0)
>plot(mu,pt(qt(1-0.05/2,df=N-1),df=N-1,ncp=ncps)
+ -pt(-qt(1-0.05/2,df=N-1),df=N-1,ncp=ncps),
+ type="l",ylab="",xlab="mu"
+ main="Wahrscheinlichkeit Fehler 2. Art")
>plot(mu,1-pt(qt(1-0.05/2,df=N-1),df=N-1,ncp=ncps)
+ +pt(-qt(1-0.05/2,df=N-1),df=N-1,ncp=ncps)
+ ,type="l",ylab="",ylab="mu",main="Gütefunktion")
```

7.
```
>N<-100
>ncps<-sqrt(N)*(mu-mu0)
>plot(mu,pt(qt(1-0.05/2,df=N-1),df=N-1,ncp=ncps)
+ -pt(-qt(1-0.05/2,df=N-1),df=N-1,ncp=ncps),
+ type="l",ylab="",xlab="mu"
+ main="Wahrscheinlichkeit Fehler 2. Art")
>plot(mu,1-pt(qt(1-0.05/2,df=N-1),df=N-1,ncp=ncps)
+ +pt(-qt(1-0.05/2,df=N-1),df=N-1,ncp=ncps)
+ ,type="l",ylab="",ylab="mu",main="Gütefunktion")
```

Übung 22.3

$\sigma = 1$, dann ist $\delta = 10$:

```
> N<-2
> pt(qt(0.975,df=N-1),df=N-1,ncp=sqrt(N)*10)-pt(-qt(0.975,df=N-1),
+ df=N-1,ncp=sqrt(N)*10)
[1] 0.2671804
> N<-3
> pt(qt(0.975,df=N-1),df=N-1,ncp=sqrt(N)*10)-pt(-qt(0.975,df=N-1),
+ df=N-1,ncp=sqrt(N)*10)
[1] 4.228304e-07
```

Also muss $N \geq 3$ gelten.

$\sigma = 30$, dann ist $\delta = \frac{1}{3}$, hier muss $N \geq 119$ gelten:

```
> N<-118
> pt(qt(0.975,df=N-1),df=N-1,ncp=sqrt(N)*1/3)-pt(-qt(0.975,df=N-1),
+ df=N-1,ncp=sqrt(N)*1/3)
[1] 0.05143629
> N<-119
> pt(qt(0.975,df=N-1),df=N-1,ncp=sqrt(N)*1/3)-pt(-qt(0.975,df=N-1),
+ df=N-1,ncp=sqrt(N)*1/3)
[1] 0.0498281
```

Übung 23.1

1.
```
> t.test(BLECH,alternative="less",mu=mu0)$p.value
[1] 0.9015088
```

2. Lehne H_0 ab, falls $T(x) > t_{N-1;1-\alpha}$:

```
> TS<-sqrt(N)*(mu0-mean(BLECH))/sd(BLECH)
> alpha<-0.05
> qt(1-alpha,df=N-1)
[1] 1.76131
> TS>qt(1-alpha,df=N-1)
[1] FALSE
```

3. H_0 kann nicht verworfen werden.

4.
```
> mu<-seq(200,400,0.5)
> plot(mu,pt(-qt(1-alpha,df=N-1),df=N-1,
+ ncp=sqrt(N)*(mu-mu0)),type="l",
+ main="Gütefunktion",ylab="",lwd=3,ylim=c(0,1))
> abline(h=alpha,lwd=2,lty=2)
```

Übung 23.2

1.
```
> N<-length(BLECH)
> alpha<-0.05
> s0<-30
> TS_var<-(N-1)*sd(BLECH)/s0
> k1<-qchisq(1-alpha/2,df=N-1)
> k2<-qchisq(alpha/2,df=N-1)
> TS_var>k1
[1] FALSE
> TS_var<k2
[1] FALSE
```

2.
```
> TS_var<-(N-1)*sd(BLECH)/s0
> k<-qchisq(alpha,df=N-1)
> TS_var<k
[1] FALSE
```

Übung 23.3

Zur Überprüfung der Anforderungen der Six-Sigma-Qualität teste $H_0^1 : \sigma \geq \frac{1}{6}\min\{USL - \mu_0, \mu_0 - LSL\}$ und $H_0^2 : |\mu - \mu_0| \geq 1.5\sigma$:

```
> N<-length(BLECH)
> sigma0<-1/6*min(c(500-310,310-150))
> TS<-(N-1)*sd(BLECH)/sigma0
> TS
[1] 18.41189
> TS<qchisq(0.05,df=N-1)
[1] FALSE
```

Damit kann H_0^1 nicht abgelehnt werden, d.h. wir können nicht nachweisen, dass $\sigma < \frac{1}{6}\min\{USL-\mu_0, \mu_0-LSL\}$ gilt. Teste trotzdem noch H_0^2. Bestimme dazu zunächst $\tilde{c}_\alpha$ wie in Beispiel 23.5.1:

```
> pt(3.838,df=N-1,ncp=sqrt(N)*1.5)-
+  pt(-3.838,df=N-1,ncp=sqrt(N)*1.5)
[1] 0.04999214
> pt(3.839,df=N-1,ncp=sqrt(N)*1.5)-
+  pt(-3.839,df=N-1,ncp=sqrt(N)*1.5)
[1] 0.05008819
```

Also $\tilde{c}_\alpha = 0.838$. Lehne H_0^2 nicht ab, da für die Teststatistik gilt $T(x) > \tilde{c}_\alpha$:

```
> TS<-sqrt(N)*abs(310-mean(BLECH))/sd(BLECH)
> TS
[1] 1.354669
```

Also kann die Six-Sigma-Qualität nicht nachgewiesen werden.

Übung 23.4

Um den t-Test anzuwenden brauchen wir Normalverteilung der Daten. Bisher haben wir dies für den Datensatz BLECH angenommen. Nun wollen wir dies für beide Datensätze BLECH und BLECH2 überprüfen:

```
> shapiro.test(BLECH)
Shapiro-Wilk normality test
data: BLECH
W = 0.9063, p-value = 0.1189
> shapiro.test(BLECH2)
Shapiro-Wilk normality test
data: BLECH2
W = 0.9679, p-value = 0.8254
```

In beiden Fällen kann die Hypothese, dass die Daten normalverteilt sind nicht abgelehnt werden.

Benutze also den t-Test zum Vergleich der Mittelwerte. Dazu muss zunächst noch die Gleichheit der Varianzen überprüft werden. Benutze hierzu den F-Test:

```
var.test(BLECH,BLECH2)
F test to compare two variances
data: BLECH and BLECH2
F = 2.1144, num df = 14, denom df = 14, p-value = 0.1736
alternative hypothesis: true ratio of variances is not
equal to 1
95 percent confidence interval:
0.7098594 6.2978609
sample estimates:
ratio of variances
2.114378
```

Die Hypothese $H_0 : \sigma_1 = \sigma_2$ kann damit nicht abgelehnt werden. Benutze also den t-Test für zwei Stichproben:

```
> t.test(BLECH,BLECH2,var.equal=T)
Two Sample t-test
data: BLECH and BLECH2
t = 0.2184, df = 28, p-value = 0.8287
alternative hypothesis: true difference in means is not
equal to 0
95 percent confidence interval:
-20.11148 24.91148
sample estimates:
mean of x mean of y
322.2667 319.8667
> t.test(BLECH,BLECH2,alternative="two.sided",var.equal=F)$p.value
[1] 0.8289144
```

Damit unterscheiden sich die beiden Blechdicken weder in der Varianz noch im Mittelwert signifikant. Auch wenn wir den Welch-t-Test benutzen, kommen wir zu diesem Ergebnis.

Benutze nun noch zusätzlich den Wilcoxon-Test zur Überprüfung der Mittelwerte:

```
> wilcox.test(BLECH,BLECH2,alternative="two.sided")
Wilcoxon rank sum test with continuity correction
data: BLECH and BLECH2
W = 121.5, p-value = 0.7243
alternative hypothesis: true location shift
is not equal to 0
```

Auch ohne Annahme der Normalverteilung kann die Hypothese, dass die Blechdicken im Mittel übereinstimmen nicht verworfen werden.

Übung 23.5

Benutze jeweils den Äquivalenz- und den Relevanztest. Ersetze in der Teststatistik $s(x)$ durch $\sigma = 30$. Wir erhalten für den Äquivalenztest die Teststatistiken:

```
> TS<-sqrt(N)*abs(310-mean(BLECH))/30
> TS
[1] 1.58362
> TS2<-sqrt(N)*abs(310-mean(BLECH2))/30
> TS2
[1] 1.273781
```

Vergleichen wir diese mit dem Wert $\tilde{c}_\alpha = 0.838$, den wir schon in der vorletzten Aufgabe bestimmt haben, so zeigt sich, dass wir $H_0 : |\mu - 310| > 1.5 \cdot 30$ nicht ablehnen können. Für den Relevanztest muss zunächst der kritische Wert c_α bestimmt werden. Wir errechnen $c_\alpha = 9.164$:

```
> 1-pt(9.165,df=N-1,ncp=sqrt(N)*d)+pt(-9.165,df=N-1,ncp=sqrt(N)*d)
[1] 0.04999897
> 1-pt(9.164,df=N-1,ncp=sqrt(N)*d)+pt(-9.164,df=N-1,ncp=sqrt(N)*d)
[1] 0.05003987
```

Die Testatistisken für die beiden Reihen sind die gleichen wie für den Äquivalenztest und damit können wir auch die Hypothese $H_0 : |\mu - 310| < 1.5 \cdot 30$ nicht ablehnen. Das könnte an dem relativ kleinen Stichprobenumfang liegen.

Übung 24.1

```
> wilcox.test(BLECH,mu=310,alternative="two.sided")$p.value
[1] 0.1979019
Warnmeldungen:
1:In wilcox.test.default(BLECH,mu = 310,alternative="two.sided"):
  kann bei Bindungen keinen exakten p-Wert Berechnen
2:In wilcox.test.default(BLECH,mu = 310,alternative ="two.sided"):
  kann den exakten p-Wert bei Nullen nicht berechnen
> t.test(BLECH,mu=310,alternative="two.sided")$p.value
[1] 0.1969823
```

Wir erhalten einen P-Wert, der nur minimal größer ist als der P-Wert beim t-Test. In beiden Fällen kann $H_0 : \mu = 310$ nicht abgelehnt werden.

Übung 24.2

```
> beton<-read.csv2("Druckfestigkeit.csv")
> druck<-beton$Druck
> dichte<-beton$Festbetonrohdichte
> shapiro.test(druck)$p.value
[1] 0.01221225
> shapiro.test(dichte)$p.value
[1] 0.003073796
> cor.test(dichte,druck,method="spearman")

        Spearman's rank correlation rho

data:  dichte and druck
S = 211725.2, p-value = 6.456e-05
alternative hypothesis: true rho is not equal to 0
sample estimates:
      rho
0.3495386

Warnmeldung:
In cor.test.default(dichte, druck, method = "spearman") :
  Kann exakte p-Werte bei Bindungen nicht berechnen
```

Wir prüfen zunächst, ob Normalverteilung vorliegt. Dies wird in beiden Fällen abgelehnt. Also bleibt als einziger möglicher Test auf Zusammenhang der beiden Komponenten der Korrelationstest basierend auf dem Rangkorrelationskoeffizienten. Dieser lehnt die Hypothese, dass die beiden Größen unkorreliert sind zum Niveau $\alpha = 0.05$ ab.

Übung 24.3

```
> rost<-matrix(c(65 , 103 , 106 , 74 , 85 , 47),ncol=3,byrow=T)
> rost
```

```
     [,1] [,2] [,3]
[1,]   65  103  106
[2,]   74   85   47
> chisq.test(rost)

        Pearson's Chi-squared test

data:  rost
X-squared = 15.7403, df = 2, p-value = 0.000382
```

Da der P-Wert kleiner als $\alpha = 0.05$ ist, kann geschlossen werden, dass es einen Zusammenhang zwischen Rostschutzmittel und Wirksamkeit gibt.

Übung 25.1

Konfidenzintervalle für die Mittelwerte der Blechdicken, für den Datensatz BLECH erhalten wir $[302.8454, 341.6879]$ und für den Datensatz BLECH2 $[306.5104, 333.2230]$:

```
> t.test(BLECH,alternative="two.sided")$conf.int
[1] 302.8454 341.6879
attr(,"conf.level")
[1] 0.95
> t.test(BLECH2,alternative="two.sided")$conf.int
[1] 306.5104 333.2230
attr(,"conf.level")
[1] 0.95
```

Das 95%-Konfidenzintervall für die Streuung ist für den ersten Datensatz $[25.67588, 55.30929]$ und für den zweiten $[17.65769, 38.03704]$:

```
> sqrt((N-1)/qchisq(0.975,df=N-1)*var(BLECH))
[1] 25.67588
> sqrt((N-1)/qchisq(0.025,df=N-1)*var(BLECH))
[1] 55.30929
```

```
> sqrt((N-1)/qchisq(0.975,df=N-1)*var(BLECH2))
[1] 17.65769
> sqrt((N-1)/qchisq(0.025,df=N-1)*var(BLECH2))
[1] 38.03704
```

Übung 26.1

```
> t<- c(23,261,87,7,120,14,62,47,
+ 225,71,246,21,42,20,5,12,120,
+ 11,3,14,71,11,14,11,16,90,116,52,95)
> lambda<-1/mean(t)
> lambda
[1] 0.01536831
> hist(t,xlab="Ausfallzeit")
> ts<-seq(0,300,1)
> hist(t,xlab="Ausfallzeit",freq=FALSE)
> lines(ts,dexp(ts,lambda),col="red",lwd=2)
> lines(ts,dweibull(ts,0.9230702,62.51505),lwd=2)
```

Der Vergleich der Dichten zeigt, dass auch die Exponentialverteilung eine gute Anpassung liefert. Die Dichten stimmen fast überein, was auch der geschätzte Wert für den Formparameter der Weibull-Verteilung nahe legt, da er sehr nah bei 1 liegt.

Übung 26.2

Wir erstellen den Weibull-Plot und schätzen die Parameter:

```
> t<-c(13.02,1.71,11.60,4.05,5.60,7.66,11.42,16.13,2.75)
> lnt<-log(sort(t))
> Ft<-(1:9-rep(0.5,9))/9
> Ft<-log(-log(1-Ft))
> plot(lnt,Ft,pch=16,xlab="logarithmierte Lebensdauern",
+      ylab=expression(paste(F[n[i]],"*")))
> est<-lm(Ft~lnt)$coeff
```

```
> est
(Intercept)         lnt
  -3.472970    1.554413
> alpha<-est[2]
> beta<-exp(-est[1]/est[2])
> alpha
     lnt
1.554413
> beta
(Intercept)
   9.339615
```

Die Schätzer sind damit $\hat{\alpha} = 1.55$ und $\hat{\beta} = 9.34$. Mithilfe der Maximum-Likelihood-Methode erhalten wir sehr ähnliche Ergebnisse, $\hat{\alpha}_{MLE} = 1.769861$ und $\hat{\beta}_{MLE} = 9.23745$:

```
> n<-length(t)
>  f<-function(a)
+ {
+  s1<-sum(t^a)
+  s2<-sum(t^a*log(t))
+  s3<-sum(log(t))
+  erg<-s2/s1-1/a-1/n*s3
+  erg}
>
> alpha<-uniroot(f,c(0,20))$root
> th<-1/n*sum(t^alpha)
> beta<-th^(1/alpha)
> alpha
[1] 1.769861
> beta
[1] 9.23745
```

Übung 26.3

Die wahren Parameter:

```
> mu<-5
> sd<-0.7
> LSL<-4
> USL<-6.5
> C_p<-(USL-LSL)/(6*sd)
> C_p
[1] 0.5952381
> C_pk<-min(c((USL-mu)/(3*sd),(mu-LSL)/(3*sd)))
> C_pk
[1] 0.4761905
> C_pm<-(USL-LSL)/(6*sqrt(sd^2+(mu-mean(c(USL,LSL)))^2))
> C_pm
[1] 0.5605607
> C_pkm<-min(c((USL-mu)/(3*sqrt(sd^2+(mu-mean(c(USL,LSL)))^2)),
+ (mu-LSL)/(3*sqrt(sd^2+(mu-mean(c(USL,LSL)))^2))))
> C_pkm
[1] 0.4484485
```

Die geschätzten Parameter für die simulierten Daten:

```
> C_p<-numeric(1000)
> C_pk<-numeric(1000)
> C_pm<-numeric(1000)
> C_pkm<-numeric(1000)
> for (i in (1:1000))
+ {
+  x<-rnorm(10,mu,sd)
+  mx<-mean(x)
+  sx<-sd(x)
+  sigmax<-sqrt(mean((x-rep(mu,10))^2))
+  C_p[i]<-(USL-LSL)/(6*sx)
+  C_pk[i]<-min(c((USL-mx)/(3*sx),(mx-LSL)/(3*sx)))
+  C_pm[i]<-(USL-LSL)/(6*sigmax)
+  C_pkm[i]<-min(c((USL-mx)/(3*sigmax),(mx-LSL)/(3*sigmax)))
+ }
> mean(C_p)
[1] 0.650838
```

```
> mean(C_pk)
[1] 0.506397
> mean(C_pm)
[1] 0.646282
> mean(C_pkm)
[1] 0.5052607
```

Übung 26.4

Block j	x_1^j	x_2^j	x_3^j	$\overline{x}_j$
1	31.8	31.4	32.0	31.73
2	30.7	31.3	29.6	30.53
3	28.1	33.6	27.3	29.67
4	29.1	28.5	31.0	29.53
5	34.7	32.3	31.3	32.77
6	34.2	34.1	35.1	34.47

Wir erhalten die folgenden Warn- und Kontrollgrenzen:

Block j	W_u	W_o	K_u	K_o	$\overline{x}_j$
1	31.34	32.12	31.13	32.33	31.73
2	31.34	32.12	31.13	32.33	30.53
3	30.38	31.87	29.99	32.27	29.67
4	28.69	32.60	27.65	33.64	29.53
5	28.48	32.25	27.5	33.25	32.77
6	28.97	32.72	27.98	33.71	34.47

Übung 26.5

```
> M<-2000
> N<-0.08*2000
> L<-0.02*2000
> 1-phyper(0:10,L,M-L,N)
[1] 0.965598347 0.843349275 0.633008959 0.399530819 0.211577035
[6] 0.094618757 0.036064064 0.011825596 0.003364588 0.000836835
[11] 0.000183111
> 1-phyper(5,L,M-L,N)
```

```
[1] 0.09461876
> L<-0.04*2000
> phyper(5,L,M-L,N)
[1] 0.3708482
> c<-5
> L<-seq(0,N)
> plot(L/N,phyper(c,L,M-L,N),type="l",ylim=c(0,1),lwd=2,
+ xlab="p",ylab="OC(p)")
```

Lehne die Lieferung bei 5 oder mehr defekten Teilen ab. Das Konsumentenrisiko beträgt dann immer noch 37%.