```r
library(lattice)
nex = read.table(file.path(pfadu, "normexample.txt"))
source(file.path(pfadu, "lattice.normal.R"))


#############################################################
# 1. Der Populationsmittelwert
#############################################################
# Erstes Beispiel
# 100 Stück Papier nummeriert 0, 1, 2, …99
# Wir ziehen 10 Stück und berechnen den Mittelwert
# Was ist der wahrscheinlichste Mittelwert dieser Zahlen?


# Dieser theoretischer Mittelwert, µ,
# (griechisch, `mu`) ist der Populations-Mittelwert
# Für den obigen Fall µ =

# Zweites Beispiel
# Wir werfen einen Würfel k Mal
# (oder k Würfel gleichzeitig ein Mal).
# Wir berechnen den Mittelwert der k Zahlen. Was ist µ?

#############################################################
# 2. Der Stichprobenmittelwert
#############################################################
# Wenn wir den obigen Vorgang durchführen, bekommen wir
# 10 Zufallswerte

# Einen Würfel werfen
sample(1:6, 1, replace=T)
# 5

# 10 Würfel werfen
sample(1:6, 10, replace=T)
# 6 2 5 4 2 3 5 1 1 3

# Der Mittelwert dieser Zufallswerte nennen wir
# den Stichprobenmittelwert und verwenden das Symbol m
#
mean(sample(1:6, 10, replace=T))
#
# Für diesen obigen Fall: m = , µ = 3.5
# d.h. m und µ werden fast immer voneinander abweichen

#############################################################
# 3. Beziehung zwischen Stichproben- und Populationsmittelwert
#############################################################

# Wir werfen k (10) Würfel N (20) Mal. Und berechnen
# jedes Mal den Mittelwert der 10 Zahlen
#
```

```r
# Wieviele Würfel werden geworfen?
k = 10
# Wie oft wird geworfen?
N = 20

wuerfel <- NULL
for(j in 1:N){
ergebnis = mean(sample(1:6, k, replace=T))
wuerfel = c(wuerfel, ergebnis)
}

wuerfel

# Der Mittelwert dieser 20 Mittelwerte nähert sich an μ
mean(wuerfel)

# Je höher N, umso näher an μ. 20 Würfel werfen. Den Mittelwert
# davon berechnen. 5000 Mal wiederholen. Daher 5000 Mittelwerte
N = 5000
wuerfel <- NULL
for(j in 1:N){
    ergebnis = mean(sample(1:6, k, replace=T))
    wuerfel = c(wuerfel, ergebnis)
}
# Der Mittelwert dieser 5000 Mittelwerte
mean(wuerfel)

# Nur wenn wir diesen Vorgang unendlich viel Mal wiederholen
# könnten, dann bekämen wir exakt μ = 3.5 durch mean(wuerfel)

###################################################################
# 4. Verteiltung der Stichprobenmittelwerte; Wahrscheinlichkeitsdichte
###################################################################

## Allgemeine Funktion mit den folgenden Defaults:
# Wir werfen k Würfel, berechnen den Mittelwert
# davon, wiederholen diesen Vorgang 50 Mal - und
# bekommen daher 50 Mittelwerte
#
#
proben <- function(unten=1, oben = 6, k = 10, N = 50)
{
# default: wir werfen 10 Wuerfel 50 Mal
alle <- NULL
for(j in 1:N){
ergebnis = mean(sample(unten:oben, k, replace=T))
alle = c(alle, ergebnis)
}
alle
}

# Funktion durchführen mit den obigen Defaults
proben()
```

```
# 100 Stück Papier nummeriert 0, 1, ... 99 in einem Hut.
# Wir ziehen 10 Stück Papier, berechnen den Mittelwert,
# (legen die 10 Stück wieder in den Hut hinein)
# wiederholen diesen Vorgang 50 Mal, bekommen 50 Mittelwerte
o = proben(0, 99, 10, 50)
# Ein Histogramm dieser 50 Mittelwerte
histogram(~ o)
# oder
histogram(o, data = NULL)

# Wahrscheinlichkeitsdichte: eine Umsetzung der vertikalen
# Achse, sodass die Flächensumme der Balken = 1
histogram(~ o,  type= "density")

# Beide Abbildungen nebeneinander
h = histogram(~ o)
hden = histogram(~ o, type="density")
# c(1, 1, 2, 1): Reihe 1, Spalte 1 in einer Abbildung von 2 Reihen und
# einer Spalte
plot(h, split = c(1, 1, 2, 1), more=T)
plot(hden, split = c(2, 1, 2, 1))
# Zur Info: density = pc/(Balkenbreite * 100)

###################################################################
# 5. Annäherung an die Normalverteilung
###################################################################
# Die Normalverteilung ist ein Histogramm, das unter zwei
# Bedingungen erstellt wird:

# (a) nicht N = 50 sondern unendlich viele Mittelwerte
#
# (b) wir lassen mit zunehmenden Stichproben
# die Balkenbreite immer kleiner werden,
# sodass im unendlichen Fall
# die Balkenbreite unendlich klein (= 0)  ist
# (also wird  die Balkenfläche  zu einer Linie).
#
# Daher haben wir keine Stufen mehr
# (von einem Balken zum nächsten) sondern eine glatte Kurve.

# Hut mit 100 Stück Papier; wir ziehen 10 berechnen den Mittelwert
# Wir wiederholen diesen Vorgang 50000 Mal (kann länger dauern!)
osehrviele = proben(0, 99, 10, 50000)
# Histogramm erstellen mit 200 Intervallen
histogram(~ osehrviele, nint=200, type = "density")

###################################################################
# 6. Die Standardabweichung
###################################################################
#
############# i. Die Stichproben-Standardabweichung
####################################################
```

```
# Die Standardabweichung misst die Streuungsgröße einer Verteilung
# um den Mittelwert, z.B:
#
nex = read.table(file.path(pfadu, "normexample.txt"))
bwplot(werte ~ Verteilung, data = nex)
# Die Standardabweichungen dieser Verteilungen

aggregate(werte ~ Verteilung, sd, data = nex)

############# ii. Die Populations-Standardabweichung,σ
#####################################################
# ist die theoretische Standardabweichung unendlich vieler Stichproben.
# z.B.
#
# Ich werfe einen Würfel 20 Mal

#Stichproben-Standardabweichung:
o = proben(k = 1, N = 20)
sd(o)

# Ich werfe den Würfel unendlich viel Mal.

# Populationsstandardabweichung, σ (griechisch, `sigma`)
sd(1:6) * sqrt(5/6)

# Annäherung an die Populations-Standardabweichung:

# Ich werfe einen Würfel 50000 Mal
o = proben(k = 1, N = 50000)
# Stichproben-Standardabweichung (müsste ziemlich nah
# an σ sein)
sd(o)

############# iii. Der Standard-Error
####################################################
# ist die Populations-Standardabweichung von Mittelwerten.
# z.B.
# ich werfe 4 Würfel, berechne den Mittelwert, wiederhole
# diesen Vorgang unendlich viel Mal,
# bekomme unendlich viele Mittelwerte.
# Der Standard-Error (SE) ist die
# Standard-Abweichung dieser Mittelwerte
# und ist in diesem theoretischen Fall σ/√k
# (k ist die Anzahl der Würfel).
# Für diesen Fall (Mittelwerte von  4 Würfeln) ist SE:

sd(1:6) * sqrt(5/6) / sqrt(4)

# Hut mit 100 Zahlen nummeriert 0, 1, 2, ..99.
# Wir ziehen 10 Zahlen, berechnen den Mittelwert
# (tun die Zahlen wieder in den Hut hinein),
# ziehen wieder 10 Zahlen, berechnen den Mittelwert,
# wiederholen diesen Vorgang unendlich viel Mal
```

```
# Was ist SE?
sd(0:99) * sqrt(99/100) / sqrt(10)
# Wir müssten diesen Wert mit ziemlich vielen Wiederholungen
# annähern
o = proben(0, 99, 10, 5000)
sd(o)

####################################################################
# 7. Überlagerung der theoretischen Normalverteilung auf eine Stichprobe
####################################################################
# Wenn μ und SE im theoretischen Fall
# (= unendlich vielen Wiederholungen) bekannt sind, dann kann
# eine theoretische Normalverteilung auf die Stichprobe uberlagert werden.

# Hut-Spiel: wir ziehen 10 Zahlen, berechnen den Mittelwert,
# wiederhole diesen Vorgang N Mal, bekomme N Mittelwerte.
# Hier für N = 50
hut = proben(0, 99, 10, 50)
mu.hut = mean(0:99)
SE.hut = sd(0:99) * sqrt(99/100) / sqrt(10)
histogram(~hut, type="density", mu = mu.hut, SE = SE.hut, panel = lattice.normal)

# je mehr Stichproben, umso besser die Anpassung an die theoretische
# Normalverteilung. ... hier für N = 5000 Wiederholungen
hut = proben(0, 99, 10, 5000)
mu.hut = mean(0:99)
SE.hut = sd(0:99) * sqrt(99/100) / sqrt(10)
histogram(~hut, type="density", mu = mu.hut, SE = SE.hut, panel = lattice.normal)

####################################################################
# 8. pnorm(): Die proportionale Fläche unter der Normalverteilung
####################################################################
# Dies wird benötigt, um Wahrscheinlichkeiten aus der Normalverteilung zu berechnen.
# Wenn ich 7 Stück Papier aus einem Hut mit
# Zahlen 0-99 ziehe, was ist die Wahrscheinlichkeit,
# dass der Mittelwert unter 38 liegt?

# mu, SE müssen berechnet werden
mu = mean(0:99)
SE = sd(0:99) * sqrt(99/100) / sqrt(7)

# Die Antwort graphisch darstellen
# (a) die entsprechende Normalverteilung
xlim = c(20, 80)
curve(dnorm(x, mu, SE), xlim = xlim, ylab = "")
#
# Unter 38: die Fläche unter der Normalverteilung
# zwischen -∞ (minus unendlich) und 38.
# graphisch (so gut wie es geht!)
#
x = seq(15, 38, length=1000)
y = dnorm(x, mu, SE)
lines(x, y, type="h", col="turquoise")
```

```
pnorm(38, mu, SE)
# [1] 0.1459311
# bedeutet: Wenn wir 7 Stück Papier aus einem
# Hut mit Zahlen 0, 1, ...99 ziehen, und davon
# den Mittelwert berechnen,
# in ca. 14-15% der Fälle ist der Mittelwert
# unter 38.

# Die Wahrscheinlichkeit, dass der Mittelwert über 68 liegt
curve(dnorm(x, mu, SE), xlim = xlim, ylab = "")
x = seq(68, 100, length=1000)
y = dnorm(x, mu, SE)
lines(x, y, type="h", col="turquoise")
# Da die Fläche unter Normalverteilung 1 (eins) ist, und da
# pnorm() immer die Fläche zwischen -∞ und einen Wert berechnet:
1 - pnorm(68, mu, SE)

# zwischen 45 und 55?
curve(dnorm(x, mu, SE), xlim = xlim, ylab = "")
x = seq(45, 55, length=1000)
y = dnorm(x, mu, SE)
lines(x, y, type="h", col="turquoise")


pnorm(55, mu, SE) - pnorm(45, mu, SE)

####################################################################
# 9. qnorm() und ein Konfidenzintervall
####################################################################
# Hut mit Zahlen 0, 1, 2, ... 99. Wir ziehen 7 Zahlen
# und berechnen den Mittelwert.
# In welchem Bereich liegt dieser Mittelwert (was ist
# unterste, was ist der oberste Wert) mit einer
# Wahrscheinlichkeit von z.B. 95%? z.B.

# Für das obige Beispiel:
# Hut mit Zahlen 0, 1, ...99. Wir ziehen
# 7 Stück Papier, berechnen den Mittelwert
# davon. In welchem Bereich liegt der
# Mittelwert mit einer Wahrscheinlichkeit von 95%?
curve(dnorm(x, mu, SE), xlim = xlim, ylab = "")

# Ein Wert berechnen, sodass die Fläche zwischen  -∞ und a gleicht 0.025
a = qnorm(0.025, mu, SE)
# Ein Wert berechnen, sodass die Fläche zwischen b und  +∞ und a gleicht 0.025
b = qnorm(0.975, mu, SE)

# graphisch
x = seq(10, a, length=1000)
y = dnorm(x, mu, SE)
curve(dnorm(x, mu, SE), xlim = xlim, ylab = "")
lines(x, y, type="h", col="turquoise")
```

```
x = seq(b, 90, length=1000)
y = dnorm(x, mu, SE)
lines(x, y, type="h", col="turquoise")
text(25,  0.001, "0.025", cex=1.2)
text(75,  0.001, "0.025", cex=1.2)
text(50, .015, "Fläche = 0.95", cex=2)

# d.h.
a
[1] 28.11611
b
[1] 70.88389

# Wir ziehen 7 Stück Papier aus einem Hut mit
# Zahlen 0-99, und berechnen den Mittelwert.
# Der Mittelwert liegt zwischen a (28.1) und b (70.9)
# mit einer Wahrscheinlichkeit von 0.95 (95%).
#
# Oder: der 95% Konfidenzintervall für den Mittelwert ,m,  ist:
# 28.11611 ≤ m  ≤ 70.88389
# (liest: Der Stichprobenmittelwert ist größer
# als oder gleicht 28.11611 und ist weniger als oder gleicht
# 70.88389 mit einer Wahrscheinlichkeit von 95%).
#
# 99 % Konfidenz-Intervall?
```