```
#############################################################
#
#    Reinhard Hoepfner -- Statistik mit Rechneruebungen -- SoSe19
#
#    log-likelihoodflaechen im normalverteilungsmodell
#
#    am beispiel von normalverteilten datensaetzen
#
#    16.05.19
#
#############################################################3



###################################
#
#    ein kleiner datensatz und ein grobes gitter
#
###################################


  # waehle einen satz normalverteilter ZV :
m0 <- 0.513 ;
s0 <- 1.458 ;
  # beachte: R will eingabe des skalenparameters
N <- 23 ;
# N <- 5 ;
# N <- 275 ;
gesicherterdatensatz <- rnorm( N, m0, s0 ) ;
gesicherterdatensatz ;


  # funktionsdefinition zur berechnung der loglikelihood
  # im normalverteilungsmodell bei  n  iid beobachtungen
bildeloglikelihood <- function(datensatz, m, s){
  dat <- datensatz ;
  hilf <- log( dnorm( dat , m, s ) ) ;
  max ( sum(hilf), -1e12 ) ;
  } ; # funktion berechnet die log-likelihoodratio an der stelle (m,s)
  # fuer die gegebenen daten 'datensatz', trunkiert 'nahe' minus unendlich
  # fuer das normalverteilungsmodell als lokations- und skalenmodell


 # waehle gitter von werten
 # fuer den lokations- und den skalenparameter :
mgitter <- seq( m0-2, m0+2, length.out=25) ;
sgitter <- seq( 0.5*s0, 3.5*s0, length.out=25 )  ;



###############################################
#
#   gemeinsame legenden fuer plots
#
###############################################


xleg <- "mittelwert als lokationsparameter" ;
yleg <- "standardabweichung als skalenparameter" ;
```

```
zleg <- "log-likelihood" ;
mainleg <- "log-likelihood-flaeche des normalverteilungsmodells:\n  iid beobachtungen unter m=&&&&
und s=%%%%" ;
mainleg <- gsub("", N, mainleg) ;
mainleg <- gsub("&&&&", round(m0,3), mainleg) ;
mainleg <- gsub("%%%%", round(s0,3), mainleg) ;
mainleg ;



###################################
#
#    log likelihood flaeche
#
#    bei vorliegender beobachtung 'gesichertedaten'
#
#    im zweiparametrigen normalverteilungsmodell
#    auf dem durch 'mgitter' und 'sgitter'
#    definierten raster berechnen
#
###################################


loglikelihoodflaeche <- array( 0, dim=c(length(mgitter),length(sgitter)) ) ;
loglikelihoodflaeche ;
  # matrix mit 'length(mgitter)' zeilen und 'length(sgitter)' spalten
  # als startwert zum ueberschreiben

for( i in 1:length(mgitter) )
  for( j in 1:length(sgitter) )
     loglikelihoodflaeche[i,j] <-
     bildeloglikelihood(gesicherterdatensatz, mgitter[i], sgitter[j]) ;
  # jetzt ist die loglikelihoodflaeche berechnet und abgespeichert
loglikelihoodflaeche ;



###################################
#
#   contour plots
#
###################################


contour( mgitter, sgitter, loglikelihoodflaeche,
 xlab=xleg, ylab=yleg, main=mainleg,
 nlevels=100 ) ;

 # vorsicht : sehr grobe gitterstruktur vorgegeben
for( i in 1:length(mgitter) ) abline( v = mgitter[i], col=5, lty=2 ) ;
for( i in 1:length(sgitter) ) abline( h = sgitter[i], col=5, lty=2 ) ;

abline( v = m0, col=2, lty=2 ) ;
abline( h = s0, col=2, lty=2 ) ;

points( m0, s0, pch=19, col=2, cex=1.5 ) ;
c( m0, s0 ) ;

m1 <- mean(gesicherterdatensatz) ;
n1 <- length(gesicherterdatensatz) ;
s1 <- sqrt( (n1-1)*var(gesicherterdatensatz)/n1 ) ;
```

```
  # beachte: MLE im normalverteilungsmodell hat 1/n normierung
points( m1, s1, pch=18, col=2, cex=2.5 ) ;
c( m1, s1 )  ;



filled.contour( mgitter, sgitter, loglikelihoodflaeche,
 xlab=xleg, ylab=yleg, main=mainleg,
 nlevels=100, col=terrain.colors(12) ) ;

filled.contour( mgitter, sgitter, loglikelihoodflaeche,
 xlab=xleg, ylab=yleg, main=mainleg,
 nlevels=100, col=terrain.colors(24) ) ;

filled.contour( mgitter, sgitter, loglikelihoodflaeche,
 xlab=xleg, ylab=yleg, main=mainleg,
 nlevels=100, col=terrain.colors(36) ) ;


 # help(terrain.colors) ;
 # aehnlich mit topo.colors(), heat colors()
 # cm.colors(), rainbow()


 # beispiele, in denen zahl der levels
 # und zahl der farben zueinander passen :


filled.contour( mgitter, sgitter, loglikelihoodflaeche,
 xlab=xleg, ylab=yleg, main=mainleg,
 nlevels=36, col=heat.colors(36) ) ;


 # mit zlim=c( . , . ) kann man diese flaeche trunkieren
 # vorsicht : das haengt vom jeweiligen wertebereich ab
filled.contour( mgitter, sgitter, loglikelihoodflaeche,
 xlab=xleg, ylab=yleg, main=mainleg,
 zlim=c(-500,-860 ),
 nlevels=36, col=heat.colors(36) ) ;


 # aehnlich hat man
image( mgitter, sgitter, loglikelihoodflaeche,
 xlab=xleg, ylab=yleg, main=mainleg,
 nlevels=36, col=heat.colors(36)  ) ;


 # sehr flexibel erscheint
filled.contour( mgitter, sgitter, loglikelihoodflaeche,
 xlab=xleg, ylab=yleg, main=mainleg,
 nlevels=100,col=rainbow(100) ) ;
 # help(rainbow) ;


####################################
#
#   persp plots
#
####################################
```

```
persp( mgitter, sgitter, loglikelihoodflaeche ) ;
  # see 'help(persp)'


persp( mgitter, sgitter, loglikelihoodflaeche,
  ticktype="detailed", nticks=10, cex.axis=0.5 ) ;
  # skalierungen auf den achsen


persp( mgitter, sgitter, loglikelihoodflaeche,
  xlab=xleg, ylab=yleg, zlab=zleg,
  ticktype="detailed", nticks=10, cex.axis=0.5,
  col=7, shade=0.3, main=mainleg ) ;
  # mit skalen, farbe und schattierung



  # man kann persp plots raeumlich drehen

winkel <- 15 ;
thetaneu <- 0 ;  thetaalt <- 0 ;
phineu<- 0 ; phialt <- 0 ;

  # in der ebene drehen : wiederhole
thetaneu <- thetaalt + winkel ;
persp( mgitter, sgitter, loglikelihoodflaeche,
  theta=thetaneu, phi=-10, col=7, shade=0.3  ) ;
thetaalt <- thetaneu  ;
  # bei leichter untersicht

  # zum zenith drehen und weiter : wiederhole
phineu <- phialt + winkel ;
persp( mgitter, sgitter, loglikelihoodflaeche,
  theta=0, phi=phineu, col=7, shade=0.3 ) ;
phialt <- phineu ;


   # man kann die box etwas groesser waehlen
   # als die zu zeigende flaeche :
persp( mgitter, sgitter, loglikelihoodflaeche,
  xlab=xleg, ylab=yleg, zlab=zleg,
  theta=30, phi=30, col=7, shade=0.5,
  xlim=1.5*range(mgitter),
  ylim=range(sgitter), zlim=range(loglikelihoodflaeche),
  ticktype="detailed", nticks=10, cex.axis=0.5, main=mainleg ) ;


  # oder die box ganz weglassen (fuer aestethen) :
persp( mgitter, sgitter, loglikelihoodflaeche,
  theta=30, phi=30, col=7, shade=0.3,
  box=F, main=mainleg ) ;


############### ende  16.05.19 ####################
```