

```

#####
#
#   Reinhard Hoepfner -- Schaetzer und Tests -- Sommer 12
#
#   Uebungsblatt 3 - Aufgabe 2
#
#   Dichten und Erzeugung von Zufallszahlen
#   fuer verschiedene Lokations- und Skalenmodelle
#
#       a) erzeugt von Doppelexponentialverteilung
#
#       b) erzeugt von einer Mischung von Doppelexponentialverteilungen
#
#       c) erzeugt von Cauchyverteilung
#
#   (Beispiel fuer eindeutige globale Maxima, f"ur Annahme desselben
#   globalen Maximums auf Teilmengen des Beobachtungsraums, fuer
#   Existenz verschiedener lokaler Maxima)
#
#   Nachbemerkung zur Erzeugung von Zufallszahlen fuer das von LeCam (1990)
#   betrachtete
#   kontaminierte Normalverteilungsmodell
#
#   30.05.12
#
#####3

#####
#
#   funktionsdefinitionen
#   fuer die lokations- und skalenmodelle aus aufgabe 3.2
#   und visualisierung dieser verteilungen
#
#####

# zum checken der folgenden funktionsdefinitionen setze fest
m0 <- 0.5 ; m <- m0 ;
s0 <- 1.5 ; s <- s0 ;
x1 <- sort( c( m0+0, m0-0.2, m0+0.2, seq(-5*s0 + m0, 5*s0 + m0, length.out=1000)
) ) ;

# eine funktion zum einzeichnen von quantilen in ein histogramm
meine_quantile <- function( daten, farbe ){
  liste <- c( 0.01, 0.025, 0.05, 0.1 ) ;
  for(i in 1:length(liste) ) abline( v=quantile( daten, liste[i]), col=farbe,
lty=i+1 ) ;
  for(i in 1:length(liste) ) abline( v=quantile( daten, 1-liste[i]), col=farbe,
lty=i+1 ) ;
} ; # funktion berechnet die empirischen quantile des datensatzes
# und zeichnet diese in einen existierenden plot ein

# bekannt: normalverteilungen und ihre quantile
zv <- rnorm( 5000, m, s ) ;
hist( zv, probability=T, nclass=100, col=8 ) ;
lines( x1, dnorm(x1,m,s), col=2 ) ;
meine_quantile( zv, 8 ) ;

# bekannt: cauchydichten und ihre quantile
zv <- rcauchy(5000, location=m, scale=s) ;

```

```

trunkiere <- qcauchy( 0.975, location=m, scale=s ) ;
zv1 <- pmax( -trunkiere, pmin( zv, trunkiere ) ) ;
hist( zv1, probability=T, nclass=250, col=8 ) ;
lines( x1, dcauchy(x1,m,s), col=2 ) ;
meine_quantile( zv, 8 ) ;

# doppel exponentialverteilungen und ihre quantile
# symmetrisiere exponentialverteilungen (fertig in R) zu
doppel exponentialverteilungen
bilde_doppel exp_dichte <- function(x, m, s){ dexp( abs( (x-m)/s ) ) / (2*s) } ;
bilde_doppel exp_zv <- function(N, m, s){
  yy <- rexp( N, rate=1/s ) ;
  vz <- 2*( rbinom(N,1,0.5) - 0.5 ) ; # vorzeichen plus minus
  vz*yy + m*rep(1,N) ;
} ; # funktion simuliert doppel exponentialverteilte zv mit lokationsparameter
m und skalenparameter s

zv <- bilde_doppel exp_zv( 5000, m, s ) ;
hist( zv, probability=T, nclass=150, col=8, ylim=c(0,1.1/(2*s)) ) ;
lines( x1, bilde_doppel exp_dichte( x1, m, s ), col=2 ) ;
meine_quantile( zv, 8 ) ;

# mischungen von doppel exponentialverteilungen :
bilde_mischdoppel exp_dichte <- function(x, m, s){
  0.5*bilde_doppel exp_dichte(x, m+0.2, s) + 0.5*bilde_doppel exp_dichte(x, m-
0.2, s)
} ; # die mischung von doppel exponentialverteilungen aus aufgabe 3.2 b)
bilde_mischdoppel exp_zv <- function(N, m, s){ 0.2*2*(rbinom(N,1,0.5)-0.5) +
bilde_doppel exp_zv(N,m,s) } ;

zv <- bilde_mischdoppel exp_zv( 5000, m, s ) ;
hist( zv, probability=T, nclass=140, col=8, ylim=c(0,1.1/(2*s)) ) ;
lines( x1, bilde_mischdoppel exp_dichte( x1, m, s ), col=2 ) ;
meine_quantile( zv, 8 ) ;

#####
#
#   ende der funktionsdefinitionen
#   fuer die lokations- und skalenmodelle aus aufgabe 3.2
#
#####

#####
#
#   NACHTRAG (fuer Aufgabe 4.1) LeCam's boesartiges beispiel :
#   kontamination im normaldichten- lokations- und skalenmodell
#
#   im produktmodell bekommen dann loglikelihoodflaechen singularitaeten fuer s
--> 0
#   an allen stellen m , die mit einer der beobachtungen zusammenfallen
#
#   (dies ist noch ein lokations-, aber kein skalenmodell mehr)
#
#####

bilde_lecam_dichte <- function(x, m, s, alpha){ (1-alpha)*dnorm(x,m,s) +
alpha*dnorm(x,m,1) } ;
# idee : alpha ist klein

```

```
bilde_lecam_zv <- function(N, m, s, alpha){ (1-alpha)*rnorm(N,m,s) +
alpha*rnorm(N,m,1) } ;
  # funktion simuliert lecam's zv

zv <- bilde_lecam_zv( 5000, m, s, alpha=0.1 ) ;
hist( zv, probability=T, nclass=100, col=8 ) ;
lines( x1, bilde_lecam_dichte( x1, m, s, alpha=0.1 ), col=2 ) ;
lines( x1, dnorm( x1, m, s ), col=2, lty=2 ) ; # der mehrheitsbestandteil
lines( x1, dnorm( x1, m, 1 ), col=2, lty=4 ) ; # der minderheitsbestandteil
meine_quantile( zv, 8 ) ;
```

```
##### ende 30.05.12 #####
```