

```

#####
#
#   Schaetzer und Tests, SoSe12
#   R. Hoepfner
#
#   Anleitung Aufgaben 1.2+1.3
#   08.05.12
#
#####

# laplacetransformierte einer exponentialverteilung
# mit parameter lambda > 0 :
exponential_LT <- function( lambda, t ) lambda/(t+lambda) ;
# ende funktion

#schoenes testbild
tgitter <- seq(0,10,0.01) ;
mm <- 0.99*max(tgitter) ;
plot( range(tgitter), c(0,1), type="n", ylab="", xlab="", main="laplace-transformierte
einer schar von exponentialverteilungen" ) ;
lines( tgitter, exponential_LT(0.01,tgitter), col=1 ) ;
lines( tgitter, exponential_LT(0.04,tgitter), col=1, lty=2 ) ;
lines( tgitter, exponential_LT(0.1,tgitter), col=2 ) ;
legend( mm, min(exponential_LT(0.1,tgitter)), cex=0.6, legend="0.1", bg=2 ) ;
lines( tgitter, exponential_LT(0.17,tgitter), col=2, lty=2 ) ;
lines( tgitter, exponential_LT(0.25,tgitter), col=3 ) ;
lines( tgitter, exponential_LT(0.37,tgitter), col=3, lty=2 ) ;
lines( tgitter, exponential_LT(0.5,tgitter), col=4 ) ;
lines( tgitter, exponential_LT(0.7,tgitter), col=4, lty=2 ) ;
lines( tgitter, exponential_LT(1.0,tgitter), col=5 ) ;
legend( mm, min(exponential_LT(1.0,tgitter)), cex=0.6, legend="1.0", bg=5 ) ;
lines( tgitter, exponential_LT(1.4,tgitter), col=5, lty=2 ) ;
lines( tgitter, exponential_LT(2.0,tgitter), col=6 ) ;
legend( mm, min(exponential_LT(2.0,tgitter)), cex=0.6, legend="2.0", bg=6 ) ;
lines( tgitter, exponential_LT(2.9,tgitter), col=6, lty=2 ) ;
lines( tgitter, exponential_LT(4.0,tgitter), col=8 ) ;
legend( mm, min(exponential_LT(4.0,tgitter)), cex=0.6, legend="4.0", bg=8 ) ;
lines( tgitter, exponential_LT(5.0,tgitter), col=8, lty=2 ) ;
# ende testbild

# berechnen einer empirischen LT
emp_LT <- function( daten , t ) {
  l <- length(daten) ;
  hilf <- rep( 0, l ) ;
  for (i in 1:l) hilf[i] <- exp( -t*daten[i] ) ;
  mean(hilf) ;
} ; # funktion berechnet empirische LT an der stelle t\ge 0 zum datensatz daten

# testbild variieren
lambda <- 1.5 ;
N <- 100 ;
daten <- rexp( N, lambda) ;
tgitter <- seq(0,10,0.1) ;
yy <- rep( 0, length(tgitter) ) ;
for (i in 1:length(tgitter) ) yy[i] <- emp_LT( daten, tgitter[i] ) ;
plot( range(tgitter), c(0,1), type="n", ylab="", xlab="", main="eine empirische laplace
transformierte" ) ;
lines( tgitter, yy, lwd=2 ) ;
lines( tgitter, exponential_LT(lambda-0.1,tgitter), col=2 ) ;
lines( tgitter, exponential_LT(lambda+0.1,tgitter), col=4 ) ;
# ende testbild

```

```

# punktweise an einer stelle t\ge 0
# quadratischer abstand zwischen einer LT mit parameter lambda
# und einer empirischen LT zum datensatz daten :
dist_LT_empLT <- function( daten, t, lambda ) ( emp_LT( daten , t ) -
exponential_LT( lambda, t ) )^2 ;
# ende funktion

# finde zuerst ein gutes gitter tgitter von argumenten t fuer die LT
# und fuer die empirischen LT
# mit haefung von gitterpunkten nahe der 0
# (z.b.: mit einer geometrischer folge gegen 0 gehen)
geometrische_rate_tgitter <- 0.85 ;
punktzahl_tgitter <- 50 ;
start_tgitter <- 10 ;
tgitter <- sort( start_tgitter * geometrische_rate_tgitter^(1:punktzahl_tgitter) ) ;
range(tgitter) ;
# pruefe, ob man nahe genug (aber auch nicht zu nahe) an die 0 herankommt

# scharparameter zur definition der schar der LT, an die angepasst werden soll,
# in ein gitter lambdagitter einschreiben
lambdagitter <- sort( 5 * 0.95^(1:100) ) ;
# lambdagitter <- seq( 0.05, 5, 0.01 ) ;
range(lambdagitter) ;
# pruefe, ob man nahe genug (aber auch nicht zu nahe) an die 0 herankommt

# graphische MDE schaeztung
# durch abstandsberechnung zwischen einer empirischen LT zum datensatz daten
# und einer (durch die eintraege in lambdagitter definierten) schar von LT's von
exponentialverteilungen
# wobei der abstand zwischen zwei funktionen in einem L^2 sinn auf einem
geometrischen,
# sich zur 0 hin verdichtenden gitter tgitter
# von nichtnegativen argumenten berechnet wird
# lambdagitter und tgitter muessen fuer die funktionsdefinition bereits
vordefiniert sein

MDEschaetzung <- function(daten){
  abstaende <- rep( 0, length(lambdagitter) ) ;
  for (i in 1:length(lambdagitter) ) {
    la <- lambdagitter[i] ;
    hilf <- rep( 0, length(tgitter) ) ;
    for (j in 1:length(tgitter) ) hilf[j] <- dist_LT_empLT( daten, tgitter[j], la ) ;
    # das ist ein punktweise an den stellen t berechneter abstand
    abstaende[i] <- mean( hilf ) ;
  } ; # i - schleife berechnet fuer ein gegebenes tgitter und eine gegebenes
lambdagitter
  # den in t gemittelten abstand
  # zwischen empirischer LT zum datensatz daten
  # und der schar der LT zu exponentialverteilungen mit parameterwerten aus
lambdagitter
  # und speichert diesen auf einem vektor abstaende von gleicher laenge wie
lambdagitter ab
  hauptleg <- "anpassung eines datensatzes an exponentialverteilungen: MDE basierend
auf laplacetransformierten" ;
  yleg <- "mittlerer quadratischer abstand berechnet auf t-gitter %%% * &&&
^( 1 : )" ;
  yleg <- gsub( "&&&", round( geometrische_rate_tgitter, 2 ), yleg ) ;
  yleg <- gsub( "", round( punktzahl_tgitter ), yleg ) ;
  yleg <- gsub( "%%%", round( start_tgitter, 2 ), yleg ) ; yleg ;
  plot( range(lambdagitter), range(abstaende), type="n", main= hauptleg,
xlab="wertebereich von lambda", ylab=yleg ) ;

```

```

points( lambdagitter, abstaende, col=2 ) ;
dstern <- min(abstaende) ;
lambdastern <- lambdagitter[ abstaende == dstern ] ;
lambdastern ;
abline( v=lambdastern, col=2, lwd=2 ) ;
ergebnisleg <- "bestapproximation in lambda = ; verbleibender mittlerer abstand zur
bestapproximation = %%%%" ;
ergebnisleg <- gsub( "", round(lambdastern,3), ergebnisleg ) ;
ergebnisleg <- gsub( "%%%%", round(dstern,6), ergebnisleg ) ;
legend( lambdastern, 0.95*max(abstaende), bg=2, legend=ergebnisleg ) ;
} ; # ende funktionsdefinition zur graphischen MDE schaeztung

```

```

# testbild
MDEschaetzung(daten) ;

```

```

# jetzt wiederholt laufen lassen und variieren
lambda <- 0.25 ;
N <- 100 ;
daten <- rexp( N, lambda ) ;
MDEschaetzung(daten) ;
# verschiedene parameterwerte ausprobieren

```

```

#####
#
#   Aufgabe 1.3
#
#####

```

```

daten <- c(
1.1410640891, 1.2559212310, 1.7091496592, 0.5791973571, 0.2350347566,
0.2797354238, 0.0835205310, 1.1497028098, 1.6077323918, 0.8691289156,
0.2975460053, 0.1229902146, 0.4172616919, 1.0820273828, 0.6185261721,
0.0587241488, 0.0306683251, 0.1879248143, 0.9258516930, 0.7601731704,
1.0013452960, 0.4099094245, 0.9833031763, 0.5648066155, 0.8742712731,
1.0682496939, 0.2342384149, 1.2652041352, 0.0271478180, 0.0510763275,
0.2498786364, 0.3913555323, 0.9585606049, 1.7270911803, 0.7146648342,
0.0666528586, 0.2451770579, 2.4354740832, 0.3013140305, 0.5157648562,
0.3676634537, 0.2083851608, 0.2653654185, 2.8224802932, 0.4735276058,
0.2012072008, 1.1979086031, 0.1357195497, 0.8904302237, 0.3454855155,
2.7100823856, 0.5170517103, 0.0557705792, 0.1832603964, 0.4347159039,
0.7251092801, 0.4508100262, 0.0186552066, 0.3026104625, 0.4651722613,
0.5632647058, 0.6903335809, 0.9993758649, 3.0855387167, 0.7227570080,
0.3589278033, 0.0765612000, 0.7773538943, 0.6006757129, 0.4373124590,
0.8409083739, 0.1219266178, 0.0935296307, 1.5909156979, 1.5226498811,
0.9043596768, 0.4571208089, 0.1851323845, 1.3467421755, 0.6719887818,
0.4104998578, 1.9006858873, 0.9281802718, 0.0541537790, 0.3712284950,
0.3690936434, 0.1254227464, 0.5987409923, 0.1315650561, 0.3457894025,
0.1164121585, 1.2771764272, 1.1435623516, 0.1471229599, 2.2297406951,
0.3517080239, 0.0075890638, 0.1670633350, 1.2889276169, 0.1613457024,
0.6031800357, 1.8473422345, 0.8048775842, 0.0005469626, 0.3179824175,
0.3239016738, 1.9120465657, 0.3077701541, 0.7779872968, 1.2986388503,
0.2596163644, 2.3730390232, 0.1840455486, 1.0670601096, 1.1644639827,
0.4506231078, 0.0752116703, 0.1822424552, 0.0375124535, 0.8625324956,
2.0103077110, 0.4287275294, 0.6951164363, 0.1497795853, 0.2240365814,
0.1460687458, 1.0281269655, 0.1141387834, 0.3854087629, 0.1287396701,
0.0611926581, 0.8346027094, 1.0433865511, 0.5442047671
) ; # ende datensatz aus aufgabe 1.3

```

```

#####
#
#   ende datenblatt : die daten aus Aufgabe 1.3
#
#####

```