

```

#####
#
# Reinhard Hoepfner, Statistik mit Rechneruebungen, SoSe19
#
# empirische WEF, empirische VF, empirische LT,
# Histogramme
#
# 01.05.19
#
#####

# datensaetze der folgenden art sollen betrachte werden :
s2_null <- 1.45 ;
norm_dat <- rnorm( 87, 0, sqrt(s2_null) ) ;

la_null <- 1.75 ;
pois_dat <- rpois( 65, la_null ) ;

la_null <- 1.75 ;
exp_dat <- rexp( 93, la_null ) ;

porositaet <- c(
  22.1, 23.5, 25.3, 26.6, 23.9, 26.0, 22.8, 22.3, 23.1, 23.0, 21.0, 21.8,
  22.0, 22.2, 22.3, 22.4, 22.4, 22.4, 22.3, 21.6, 22.1, 22.6, 22.1, 21.9,
  22.3, 23.9, 23.2, 22.5, 23.7, 23.3, 24.4, 22.6, 23.9, 24.2, 27.6, 27.9,
  25.2, 21.7, 20.0, 19.8, 21.5, 25.6, 25.3, 24.1, 28.6, 23.7, 24.0, 21.8,
  24.9, 24.2, 25.0, 23.7, 27.3, 23.0, 23.8, 21.2, 21.1
) ;
por <- porositaet ;
por_zs_dat <- ( por - mean(por)*rep(1,length(por)) )/sqrt( var(por) ) ;

#####
#
# WEF und empirische WEF (poissonverteilungen)
#
#####

# zwei funktionsdefinitionen :

WEF <- function(lambda,s){ exp( -lambda*(1-s) ) } ;
# help(dpois) ;

emp_WEF <- function(daten,s) mean( s^daten ) ;

# parameter waehlen :
n_null <- 65 ;
# n_null <- 36 ;
# n_null <- 1797 ;

la_null <- 1.75 ;
# la_null <- 0.63 ;

# legenden und argumente
leg <- "poissonvert mit parameter : WEF (rot) und empirische WEF zu %%% simulierten poisson
daten" ;
leg <- gsub( "", la_null, leg ) ;
leg <- gsub( "%%%", n_null, leg ) ;

```

```

s_liste <- seq( 0, 1, 0.001 ) ;
y_liste <- rep(0,length(s_liste)) ;
z_liste <- rep(0,length(s_liste)) ;

# WEF
for( i in 1:length(s_liste) ) y_liste[i] <- WEF(la_null,s_liste[i]) ;
plot( c(0,1), c(0,1), type="n", xlab="", ylab="", main=leg ) ;
lines( s_liste, y_liste, lwd=3, lty=6, col=2 ) ;

# empirische WEF
dat <- rpois( 65, la_null ) ;
for( i in 1:length(s_liste) ) z_liste[i] <- emp_WEF(dat,s_liste[i]) ;
lines( s_liste, z_liste, lwd=2 ) ;

#####
#
#   VF und empirische VF (zentrierte normalverteilungen)
#
#####

# funktionsdefinition :

emp_DF <- function(daten,y){ length(daten[daten <= y]) / length(daten) } ;

# parameter waehlen :
n_null <- 55 ;
# n_null <- 385 ;
s2_null <- 1.85 ;
# s2_null <- 0.45 ;

# n_null <- length(por_zs_dat) ; s2_null <- 1 ;

# legenden und argumente
leg <- "zentrierte normalvert mit varianz : DF (rot) und empirische DF zu %%% simulierten
daten" ;
leg <- gsub( " ", s2_null, leg ) ;
leg <- gsub( "%%%", n_null, leg ) ;

skal <- 4 * sqrt(s2_null) ;
x_liste <- seq( -skal, skal, 0.001 ) ;
y_liste <- rep(0,length(x_liste)) ;
z_liste <- rep(0,length(x_liste)) ;

# DF
for( i in 1:length(x_liste) ) y_liste[i] <- pnorm( x_liste[i], 0, sqrt(s2_null) ) ;
plot( c(-skal,skal), c(0,1), type="n", xlab="", ylab="", main=leg ) ;
lines( x_liste, y_liste, lwd=3, lty=6, col=2 ) ;

# empirische DF
dat <- rnorm( n_null, 0, sqrt(s2_null) ) ;
for( i in 1:length(x_liste) ) z_liste[i] <- emp_DF(dat,x_liste[i]) ;
lines( x_liste, z_liste, lwd=2 ) ;

#####
#
#   anwendung auf den datensatz sandsteinporositaet
#   zentriert und standardisiert

```

```

#
#####

# legenden und argumente
dat <- por_zs_dat ;

leg <- "'sandsteinporositaet' zentr+stand: emp. VF vs. standardnormalVF (rot)" ;

skal <- 3 ;
x_liste <- seq( -skal, skal, 0.001 ) ;
y_liste <- rep(0,length(x_liste)) ;
z_liste <- rep(0,length(x_liste)) ;

# DF
for( i in 1:length(x_liste) ) y_liste[i] <- pnorm( x_liste[i], 0, 1 ) ;
plot( c(-skal,skal), c(0,1), type="n", xlab="", ylab="", main=leg ) ;
lines( x_liste, y_liste, lwd=3, lty=6, col=2 ) ;

# empirische DF
dat <- rnorm( n_null, 0, sqrt(s2_null) ) ;
for( i in 1:length(x_liste) ) z_liste[i] <- emp_DF(dat,x_liste[i]) ;
lines( x_liste, z_liste, lwd=2 ) ;

#####
#
# LT und empirische LT (exponentialverteilungen)
#
#####

# funktionsdefinition :

emp_LT <- function(daten,t){ mean( exp( (-t)*daten ) ) } ;

LT <- function(t,lambda){ lambda / ( t + lambda ) } ;
# help(pexp) ;

# parameter waehlen :
n_null <- 78 ;
# n_null <- 378 ;
la_null <- 1.55 ;
# la_null <- 0.85 ;

# legenden und argumente
leg <- "exponentialverteilung mit parameter : LT (rot) und empirische LT zu %%% simulierten
daten" ;
leg <- gsub( "", la_null, leg ) ;
leg <- gsub( "%%%", n_null, leg ) ;

skal <- 10*la_null ;
t_liste <- seq( 0, skal, 0.001 ) ;
y_liste <- rep(0,length(t_liste)) ;
z_liste <- rep(0,length(t_liste)) ;

# LT
for( i in 1:length(t_liste) ) y_liste[i] <- LT( t_liste[i], la_null ) ;
plot( c(0,skal), c(0,1), type="n", xlab="", ylab="", main=leg ) ;
lines( t_liste, y_liste, lwd=3, lty=6, col=2 ) ;

```

```

# empirische LT
dat <- rexp( n_null, la_null ) ;
for( i in 1:length(t_liste) ) z_liste[i] <- emp_LT( dat, t_liste[i] ) ;
lines( t_liste, z_liste, lwd=2 ) ;

#####
#
#   CF und empirische CF (zentrierte normalverteilungen)
#
#####

# funktionsdefinition :

emp_CF <- function(daten,t){ mean( exp( li*t*daten ) ) } ;
# li ;
# help(Re) ; help(Im) ;
# erklart die verwendung komplexer zahlen in R

CF <- function(t,s2_null){ exp( -0.5 * s2_null * t^2 ) } ;
# help(pnorm) ;

# parameter waehlen :
n_null <- 65 ;
# n_null <- 378 ;
s2_null <- 1.55 ;
# s2_null <- 0.85 ;

# legenden und argumente
leg1 <- "zentrierte normalvert. mit var. : Re(CF) (rot) und Re(emp. CF) zu %%% simulierten
daten" ;
leg1 <- gsub( "", s2_null, leg1 ) ;
leg1 <- gsub( "%%%", n_null, leg1 ) ;

leg2 <- "zentrierte normalvert. mit var. : Im(CF) (rot) und Im(emp. CF) zu %%% simulierten
daten" ;
leg2 <- gsub( "", s2_null, leg2 ) ;
leg2 <- gsub( "%%%", n_null, leg2 ) ;

skal <- 10*s2_null ;
t_liste <- seq( -skal, skal, 0.001 ) ;
y_liste <- rep( 0 + 1i ,length(t_liste)) ;
z_liste <- rep( 0 + 1i ,length(t_liste)) ;
# startwert zum ueberschreiben
for( i in 1:length(t_liste) ) y_liste[i] <- CF( t_liste[i], s2_null ) ;

par( mfrow=c(2,1) ) ;
dat <- rnorm( n_null, 0, sqrt(s2_null) ) ;
for( i in 1:length(t_liste) ) z_liste[i] <- emp_CF( dat, t_liste[i] ) ;
#
plot( range(t_liste), c(-1,1), type="n", xlab="", ylab="", main=leg1 ) ;
lines( t_liste, Re(y_liste), lwd=2, col=2 ) ;
lines( t_liste, Re(z_liste), lwd=2 ) ;
#
plot( range(t_liste), c(-1,1), type="n", xlab="", ylab="", main=leg2 ) ;
lines( t_liste, Im(y_liste), lwd=2, col=2 ) ;
lines( t_liste, Im(z_liste), lwd=2 ) ;
#
par( mfrow=c(1,1) ) ;

```

```

#####
#
#   histogramme
#
#####

# datensatz :
dat <- por ;
# dat <- exp_dat ;
# dat <- norm_dat ;
# dat <- pois_dat ;

hist( dat ) ;
# help(hist) ;
rug( dat, col=2, lwd=2 ) ;

hist( dat, nclass=10, main="sandsteinporositaet" ) ;
# zellhaeufigkeiten
hist( dat, probability=T, nclass=10, main="sandsteinporositaet", xlab="" ) ;
# flaeche unter histogramm auf 1 normiert

hist( dat, probability=T, nclass=20, xlab="sandsteinporositaet", xlab="" ) ;

# hist( dat, probability=T, nclass=5, xlab="sandsteinporositaet", xlab="" ) ; # zu grob
# hist( dat, probability=T, nclass=50, xlab="sandsteinporositaet", xlab="" ) ; # zu fein

# selbstdefinierte zellgrenzen (vermeiden dass gitterstruktur
# des datensatzes mit zellgrenzen zusammenfaellt)

meinebreaks <- seq( min(dat)-1.05, max(dat)+1.05, 0.6 ) ;
# meinebreaks <- seq( min(dat)-1.05, max(dat)+1.05, 1.2 ) ; # zu grob
# meinebreaks <- seq( min(dat)-1.05, max(dat)+1.05, 0.3 ) ;
# meinebreaks <- seq( min(dat)-1.05, max(dat)+1.05, 0.1 ) ; # zu fein

hist( dat, probability=T, breaks=meinebreaks, col=7, main="sandsteinporositaet", xlab="" ) ;

# man kann das histogramm auch als reinen datensatz auswerten
hist( dat, breaks=meinebreaks, plot=F ) ;

abline( v=median(dat), lty=2, col=3, lwd=2 ) ;
abline( v=quantile(dat,0.25), lty=2, col=3, lwd=2 ) ;
abline( v=quantile(dat,0.75), lty=2, col=3, lwd=2 ) ;

abline( v=mean(dat), lty=2, col=2, lwd=2 ) ;

# vergleiche: normaldichte mit geschaetzten parametern
m <- mean(dat) ;
s <- sqrt( var(dat) ) ;
xx <- seq( min(dat)-1.05, max(dat)+1.05, 0.01 ) ;
yy <- dnorm( xx, m, s ) ;
lines( xx, yy, lwd=2, lty=6, col=2 ) ;
abline( v=mean(dat), lty=2, col=2, lwd=2 ) ;
abline( v=qnorm(0.25,m,s), lty=2, col=2, lwd=2 ) ;
abline( v=qnorm(0.75,m,s), lty=2, col=2, lwd=2 ) ;

#####
#
#   ende 01.05.19
#
#####

```