

```

#####
#
#   R. Hoepfner -- Statistik mit Rechneruebungen -- SoSe19
#   Blatt 5A -- Praesenzaufgabe
#
#   Berechne per Simulation die Guetefunktionen
#   von Rangtests in Zweistichprobenmodellen:
#   Wilcoxon, Median, van der Waerden
#
#   Vergleich auf verschiedenen Verteilungstypen
#
#   07.07.19
#
#####

m <- 10 ; N <- 2*m ;
# stichprobenumfang fuer die drei tests
alpha <- 0.05 ;
# gewuenshtes niveau fuer alle drei tests

# bestimme zu (m,N) und alpha kritische werte

# cc und gg

# so dass gilt:

#  $P(S > cc) + gg P(S = cc) = \alpha$ 

# der einseitige rangtest zum Niveau alpha ist dann
# mit einer nichtfallenden funktion a(.) und der
# summe der gewichteten rangzahlen der ersten teilstichprobe

#  $S = a(R_1) + \dots + a(R_m)$ 

# von form

#  $\phi \leftarrow \text{ifelse}( S > cc , 1 , 0 ) ;$ 
#  $\text{if}( S = cc ) \phi \leftarrow gg ;$ 

#####
#
# bestimme die kritischen werte fuer v-d-waerden
#
#####

# test phil : v-d-waerden test :

# v-d-waerden scores  $a(.)=a_N(.)$  :
aal <- function(x){ qnorm(x/(N+1),0,1) } ;
plot( 1:N, aal(1:N), main="v-d-waerden-scores" ) ;

# kritische werte v-d-waerden zum niveau 0.05
# entnimmt man der tabelle in hajek (1969) s. 167 :
cc1 <- 3.3 ;
gg1 <- 0 ;

```

```

#####
#
# bestimme die kritischen werte fuer mediantest
#
#####

# test phi2 mediantest :

# mediantest scores a(.)=a_N(.) :
aa2 <- function(x){ ifelse( x > (N+1)/2 , 1 , 0 ) };
plot( 1:N, aa2(1:N), main="mediantest-scores" );

# kritische werte erhaelt man aus der hypergeometrischen vert.
# ' urne mit N kugeln, m davon rot, man entnimmt stichprobe der
# der laenge m : wieviele kugeln in der stichprobe sind rot ? '
# help(dhyper)

cc2 <- qhyper( 0.95, m, N-m, m ) ;
cc2 ;
# [1] 7
# umgebung des vordefinierten quantils genau anschauen :
plot( 1:m, phyper( 1:m, m, N-m, m ) ) ;
abline( h=0.95, lty=2 ) ;
abline(v = cc2, lty=2 ) ;

# dann damit weiter
hilf2 <- 1 - phyper( cc2, m, N-m, m ) ;
gg2 <- ( 0.05 - hilf2 ) / dhyper( cc2, m, N-m, m ) ;
gg2 ;
# [1] 0.493875

#####
#
# bestimme die kritischen werte fuer wilcoxon-test
#
#####

# test phi3 wilcoxon-test

# wilcoxon-scores fuer (m,N) zum niveau alpha = 0.05
aa3 <- function(x){ x } ;
# die funktion a(.) fuer rangstatistik S
plot( 1:N, aa3(1:N), main="wilcoxon-scores" ) ;

# kritische werte erhaelt man aus
# help(qwilcox)
# R gibt die U-form der rangsstatistik, nicht die S-form
#  $S = U + m*(m+1)/2$ 
xx3 <- seq( 1 , (N-m)*m, 1 ) ;
yy3 <- pwilcox( xx3, m, N-m ) ;
plot( xx3, yy3, main="verteilungsfunktion des wilcoxon-U auf dem rand der hypothesen" ) ;

abline( h=0.95, lty=2 ) ;
cc3 <- qwilcox( 0.95, m, N-m ) ;
abline( v=cc3, lty=2 ) ;
cc3 ;
# [1] 72

```

```

# genauer ansehen mit
hilf_1 <- pwilcox( cc3-1, m, N-m, m ) ;
hilf_2 <- pwilcox( cc3, m, N-m, m ) ;
hilf_3 <- pwilcox( cc3+1, m, N-m, m ) ;
c( hilf_1, hilf_2, hilf_3 ) ;

# man setzt dann
gg3 <- ( 0.05 - (1-pwilcox( cc3, m, N-m, m)) ) / dwilcox( cc3, m, N-m ) ;
gg3 ;
# [1] 0.6780952

# soweit wurde in der U-form gerechnet, brauche S-form :
cc3 <- cc3 + m*(m+1)/2 ;
cc3 ;
# [1] 127 ;

#####
#
# definitionen fuer das simulationsprogramm :
#
#####

# delta-shift auf erste teilstichprobe anwenden
shiftedaten <- function(dat,d) { dat + c( rep(d,m), rep(0,N-m) ) } ;

# gitter der moeglichen delta-werte
delta_gitter <- seq( -1.5, 3.5, 0.1 ) ;

# addiere werte der testfunktionen in eine matrix
guete <- matrix( 0, ncol=length(delta_gitter), nrow = 3 ) ;
guetealt <- guete ;
# zeilen entsprechen den delta -werten,
# jede spalte ist fuer einen der drei tests vorgesehen

# funktionsdefinition: zahl der simulationslaeufe, verteilungstyp
simuliere <- function(schritte, typ){
  guete <- matrix( 0, nrow = 3, ncol=length(delta_gitter) ) ;
  guetealt <- guete ; # startwerte zum ueberschreiben
  # spalten entsprechen den delta-werten, zeilen den drei tests
  rangzahlen <- rep( 0, length(delta_gitter) ) ;
  R1T <- rep( 0, m ) ; # rangzahlen erste teilstichprobe
  # zum ueberschreiben
  for (zaehl in 1:schritte) {
    if (typ == 1) iidat <- rnorm(N,0,1) ;
    if (typ == 2) iidat <- runif(N,0,1) ;
    if (typ == 3) iidat <- rcauchy(N) ;
    if (typ == 4) { u <- rbinom(N,1,0.5) ;
      iidat <- u * rexp(N) - (rep(1,N)-u) * rexp(N) ;
    } ; # erzeugung doppel exponentialverteilung
    # in jedem simulationslauf werden N iid zufallsvariablen erzeugt
    # die naechsten schleife: delta-shifts auf erste teilstichprobe
    for ( j in 1:length(delta_gitter) ) {
      guetealt <- guete[,j] ;
      # die bisherige j-te spalte der matrix guete sichern
      rangzahlen <- rank( shiftedaten( iidat, delta_gitter[j] ) ) ;
      R1T <- rangzahlen[1:m] ;
      # test phi1
      phi1 <- ifelse( sum( aa1(R1T) ) > cc1 , 1 , 0 ) ;
      if( round( sum(aa1(R1T)),2 ) == round(cc1,2) ) phi1 <- gg1 ;
      # test phi2
      phi2 <- ifelse( sum( aa2(R1T) ) > cc2 , 1 , 0 ) ;
      if( round( sum(aa2(R1T)),2 ) == round(cc2,2) ) phi2 <- gg2 ;
      # test phi3

```

```

    phi3 <- ifelse( sum( aa3(R1T) ) > cc3 , 1 , 0 ) ;
    if( round( sum(aa3(R1T)),2 ) == round(cc3,2) ) phi3 <- gg3 ;
    # jetzt j-te spalte der matrix guete updaten
    guete[,j] <- guetealt + c( phil, phi2 , phi3 ) ;
  } ; # schleife addiert werte der simulierten testfunktionen
  # unter shift delta fuer erste teilstichprobe
  # zu der delta entsprechenden spalte der matrix guete dazu
} ; # ende der zaehlschleife
(1/schritte)*guete ; # ausgabe :
# empirisches mittel der simulierten testfunktionen
# als approximation an die wahre guetefunktion
} ; # funktion simuliert
# die guetefunktionen von drei tests auf delta_gitter

schritte <- 5000 ;
typ <- 1 ;
# typ <- 2 ;
# typ <- 3 ;
# typ <- 4 ;
guetematrix <- simuliere(schritte,typ) ;

# (*) ab hier zum malen der graphiken
if (typ == 1) leg1 <- "2-stichproben-normalverteilungsmodell: sim guetefktn von vdwaerden-,
median- und wilcoxontest" ;
if (typ == 2) leg1 <- "2-stichproben-gleichverteilungsmodell: sim guetefktn von vdwaerden-,
median- und wilcoxontest" ;
if (typ == 3) leg1 <- "2-stichproben-cauchymodell: sim guetefunktionen von vdwaerden-, median-
und wilcoxontest" ;
if (typ == 4) leg1 <- "2-stichproben-doppelexp.vert.modell: sim guetefktn von vdwaerden-, median-
und wilcoxontest" ;
leg2 <- "tuerkis = vdwaerden, magenta = median, blau = wilcoxon; m = N-m = %%%, alpha = 0.05,
sim laeufe" ;
leg2 <- gsub( "%%%", m, leg2 ) ;
leg2 <- gsub( "", schritte, leg2 ) ;

plot( range(delta_gitter), c(0,1), type="n", bty="l", main=leg1, xlab=leg2, ylab="" ) ;
abline( h = 0, lty=2 ) ; abline( h=1, lty=2 ) ;
abline( h = 0.05, col=2, lty=2 ) ; abline( v=0, col=2, lty=2 ) ;
lines( delta_gitter, guetematrix[1,], col=5, lwd=2.5 ) ; # vdwaerden farbe tuerkis
lines( delta_gitter, guetematrix[2,], col=6, lwd=2.5 ) ; # median farbe magenta
lines( delta_gitter, guetematrix[3,], col=4, lwd=2.5 ) ; # wilcoxon farbe blau

##### 07.07.19 #####

```