

(1) Einführung (Beispiel: Einstichproben- T^2 -Tests)

Ziel des Seminars ist es, anhand von in **R** geschriebenem Quellcode die in der Vorlesung behandelten Datenanalysemethoden mathematisch nachzuvollziehen.

In dieser Sitzung werden Einstichproben- T^2 -Tests benutzt (die in diesem Semester nicht in der Vorlesung behandelt werden), um exemplarisch zu zeigen, wie die Analysen im Seminar dargestellt werden sollen. Dabei werden anhand einer Simulation der Verteilung der Einstichproben- T^2 -Teststatistik zunächst die Grundlagen Frequentistischer Inferenz veranschaulicht und anschließend ein T^2 -Test im Anwendungsszenario durchzuführen.

Simulation der Einstichproben- T^2 -Teststatistik

Folgender **R**-Code implementiert eine Simulation der Verteilung der Einstichproben- T^2 -Teststatistik auf Grundlage des wiederholten identischen und unabhängigen Realisierens von multivariat normalverteilten zweidimensionalen Zufallsvariablen.

```
# Modellparameter
m      = 2                                # Dimensionalität
n      = 15                               # Anzahl der Datenpunkte
mu_0   = matrix(c(1,1), nrow = 2)        # H0-Hypothesenparameter
mu     = matrix(c(2,2), nrow = 2)        # w.a.u. Erwartungswertparameter
Sigma  = matrix(c(0.5,0.3,0.3,0.5), nrow = 2, byrow = TRUE) # w.a.u. Kovarianzmatrixparameter

# Simulation
library(MASS)                             # multivariate Normalverteilungen
nsim   = 1e4                             # Anzahl Datensatzrealisierungen
Yb     = matrix(rep(NaN,m*nsim), nrow = 2) # Stichprobenmittelarray
T2     = rep(NaN,nsim)                   # T^2-Statistik-Array
j_n    = matrix(rep(1,n), nrow = n)      # 1_n
I_n    = diag(n)                        # I_n
J_n    = matrix(rep(1,n^2), nrow = n)    # 1_{nn}
for(s in 1:nsim){
  Y      = t(mvrnorm(n,mu,Sigma))        # Simulationsiterationen
  Y_bar  = (1/n)*(Y %*% j_n)              # y_i \sim N(\mu,\Sigma)
  C      = (1/(n-1))*(Y %*% (I_n-(1/n)*J_n) %*% t(Y)) # Stichprobenmittel
  T2[s]  = n*t(Y_bar - mu_0) %*% solve(C) %*% (Y_bar - mu_0) # Stichprobenkovarianzmatrix
  Yb[,s] = Y_bar                         # T^2-Statistik
}                                         # Stichprobenmittelarray
```

Mit folgendem **R**-Code wird obige Simulation in Abbildung 1 visualisiert.

```
# Abbildungsparameter
library(latex2exp)
par(
  family = "sans",
  mfcol  = c(2,2),
  pty    = "s",
  bty    = "l",
  lwd    = 1,
  las    = 1,
  mgp    = c(2,1,0),
```

```

xaxs      = "i",
yaxs      = "i",
font.main = 1,
cex.main  = 1)

# Stichprobenmittel per Simulation
plot(Yb[1,], Yb[2,],
     xlim    = c(1,3),
     ylim    = c(1,3),
     xlab    = TeX("$\\bar{y}_1$"),
     ylab    = TeX("$\\bar{y}_2$"),
     pch     = 21,
     col     = "white",
     bg      = "gray60",
     main    = TeX("$\\bar{y}$ Simulationen"))

# Normalisierte T^2-Teststatistik-WDF
t2_min     = 0
t2_max     = 200
t2_res     = 1e3
t2         = seq(t2_min, t2_max, len = t2_res)
nu         = (n-m)/(m*(n-1))
delta      = n*t(mu - mu_0) %*% solve(Sigma) %*% (mu - mu_0)
p_T2       = nu*df(nu*t2,m,n-m,delta)
hist(T2,
     breaks = 100,
     col    = "gray90",
     prob   = TRUE,
     xlim   = c(t2_min, t2_max),
     ylim   = c(0,0.03),
     xlab   = TeX("$t^2$"),
     ylab   = "",
     main   = TeX("$p(t^2) = \\nu \\, , f(\\nu \\, , t^2; \\, , \\delta, m, n-m)$"))
lines(t2, p_T2,
     lwd   = 2,
     col   = "darkorange")

# Unnormalisierte T^2-Teststatistik WDF
t2_min     = 0
t2_max     = 50
t2_res     = 1e3
t2         = seq(t2_min, t2_max, len = t2_res)
delta      = n*t(mu-mu_0) %*% solve(Sigma) %*%(mu-mu_0)
p_nm_T2    = df(t2,m,n-m,delta)
hist(((n-m)/m)/(n-1)*T2,
     breaks = 100,
     col    = "gray90",
     prob   = TRUE,
     xlim   = c(t2_min, t2_max),
     ylim   = c(0,0.06),
     xlab   = TeX("$\\nu \\, , t^2$"),
     ylab   = "",
     main   = TeX("$\\nu \\, , T^2 \\sim f(\\delta, m, n-m)$"))
lines(t2, p_nm_T2,
     lwd   = 2,
     col   = "darkorange")

# Normalisierte T^2-Teststatistik-KVF
t2_min     = 0
t2_max     = 200

```

```

t2_res      = 1e3
t2          = seq(t2_min, t2_max, len = t2_res)
nu          = (n-m)/(m*(n-1))
P_T2        = pf(nu*t2, m, n-m, delta)
ECDF_T2     = ecdf(T2)
plot(ECDF_T2,
     verticals = TRUE,
     lwd       = 6,
     xlim      = c(t2_min, t2_max),
     col       = "gray90",
     xlab      = TeX("$t^2$"),
     ylab      = "",
     main      = TeX("$P(t^2) = F(\\nu t^2; \\delta, m, n-m)$"))
lines(t2, P_T2,
      lwd     = 2,
      col     = "darkorange")
dev.copy2pdf(
  file       = "./Abbildungen/einstichproben_t2_teststatistik.pdf",
  width      = 6,
  height     = 6)
dev.off()

```

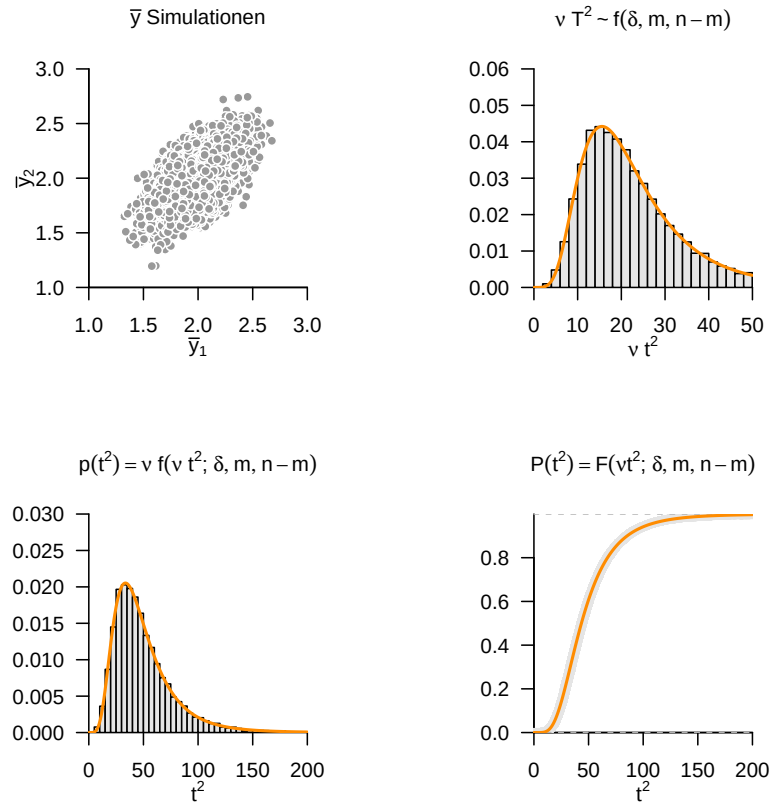


Abbildung 1. Simulation der Frequentistischen Verteilung der T^2 -Teststatistik.

Anwendungsbeispiel und Deskriptivstatistik

Als Anwendungsbeispiel betrachten wir die Frage nach der mit der Abweichung eines zweidimensionalen Gruppenmittelwertes von einem Therapieerfolgsnormwert $\mu_0 = (30, 3.5)$ assoziierten Unsicherheit anhand der in Tabelle 1 dargestellten Daten.

Tabelle 1. Differenzwerte für BDI-II-Score und Glukokortikoidplasmalevel prä/post Intervention von $n = 20$ Patient:innen

dBDI	dGLU
35	6.1
25	4.0
20	1.7
29	2.6
29	1.9
17	0.9
33	2.0
28	4.1
26	3.9
31	3.8
14	2.1
18	2.0
19	5.0
28	2.6
20	2.1
35	4.4
28	4.0
32	3.9
32	1.0
25	1.9

Mithilfe folgenden **R**-Codes berechnen wir zunächst das Stichprobenmittel, die Stichprobenkovarianzmatrix und die Mahalanobis-Distanz des Stichprobenmittels vom Normwert.

```
# Deskriptivstatistik
D = read.csv("Einstichproben_T2_Tests.csv")      # Daten einlesen
Y = rbind(D$y_1i, D$y_2i)                        # Datenmatrix
mu_0 = matrix(c(30,3.5), nrow = 2)              # Normwert
n = ncol(Y)                                       # Anzahl Datenpunkte
j_n = matrix(rep(1,n), nrow = n)                # 1_n
I_n = diag(n)                                    # I_n
J_n = matrix(rep(1,n^2), nrow = n)              # 1_{nn}
Y_bar = (1/n)*(Y %*% j_n)                        # Stichprobenmittel
C = (1/(n-1))*(Y %*% (I_n-(1/n)*J_n) %*% t(Y))  # Stichprobenkovarianzmatrix
D = t(Y_bar - mu_0) %*% solve(C) %*% (Y_bar - mu_0) # Mahalanobis-Distanz
cat( "Y_bar =", Y_bar,                          # Ausgabe
     "\nD      =", D, "\n")
```

```
Y_bar = 26.25615 2.991039
D      = 0.3773184
```

Mithilfe folgenden **R**-Codes visualisieren wir den Datensatz wie in Abbildung 2 gezeigt.

```
# Abbildungsparameter
library(ellipse)
library(latex2exp)
par(
  family      = "sans",
  mfcol       = c(1,1),
  pty         = "s",
  bty         = "l",
  lwd         = 1,
  las         = 1,
  mgp         = c(2,1,0),
  xaxs        = "i",
  yaxs        = "i",
  font.main   = 1,
  cex         = 1,
  cex.main    = 1)

# Datenpunkte
plot(Y[1,], Y[2,],
     col = "Gray70",
     bg  = "Gray70",
     pch = 21,
     xlab = TeX("\\Delta{}BDI"),
     ylab = TeX("\\Delta{}GLU"),
     xlim = c(10,40),
     ylim = c(0,7))
points(Y_bar[1], Y_bar[2],
       col = "White",
       bg  = "gray70",
       pch = 24)
lines(ellipse(C, level = 0.5, centre = Y_bar),
      type = "l",
      col = "Gray70")

# Normwert
points(mu_0[1], mu_0[2],
       col = "White",
       bg  = "Gray30",
       pch = 24)

# Legende
legend("topleft", c("Datenpunkte", "Stichprobenmittel", "Stichprobenkovarianz", "Normwert"),
      lty = c(NA,NA,1,NA),
      pch = c(19,17,NA,17),
      col = c("gray70","gray70","gray70","gray30"),
      bty = "n",
      cex = 1,
      y.intersp = 3)

# Speichern
dev.copy2pdf(
  file      = "./Abbildungen/einstichproben_t2_deskription.pdf",
  width     = 4.5,
  height    = 4.5)
dev.off()
```

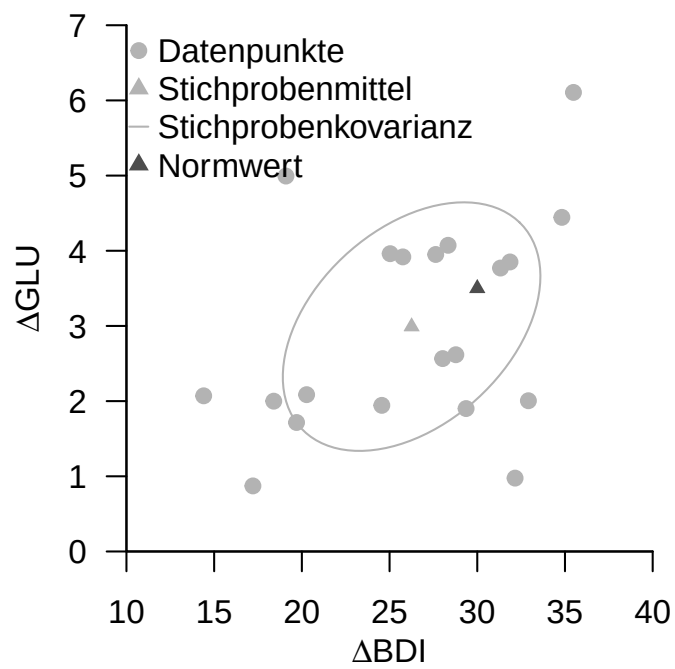


Abbildung 2. Deskriptivstatistik des Beispieldatensatzes.

Inferenzstatistik mit dem Einstichproben-T²-Test

Mit folgendem **R**-Code führen wir schließlich einen Einstichproben-T²-Test mit Signifikanzlevel $\alpha_0 = 0.05$ durch.

```
# Datenbereitstellung
D      = read.csv("Einstichproben_T2_Tests.csv")      # Datensatzeinlesen
Y      = rbind(D$y_1i, D$y_2i)                      # Datenselektion

# Testparameter
m      = nrow(Y)                                     # Dimensionalität der Daten
n      = ncol(Y)                                     # Anzahl der Datenpunkte
nu     = (n-m)/(m*(n-1))                             # Parameter
mu_0   = matrix(c(30,3.5), nrow = 2)                # H0-Parameter ("Normwert")
alpha_0 = 0.05                                       # Signifikanzlevel
k_alpha_0 = (1/nu)*qf(1-alpha_0, m, n-m)            # kritischer Wert

# Testevaluation
j_n    = matrix(rep(1,n), nrow = n)                 # 1_n
I_n    = diag(n)                                    # I_n
J_n    = matrix(rep(1,n^2), nrow = n)                # 1_{nn}
Y_bar  = (1/n)*(Y %*% j_n)                           # Stichprobenmittel
C      = (1/(n-1))*(Y %*% (I_n-(1/n)*J_n) %*% t(Y))   # Stichprobenkovarianzmatrix
T2     = n*t(Y_bar - mu_0) %*% solve(C) %*% (Y_bar - mu_0) # T^2-Statistik
if(T2 > k_alpha_0){                                  # Test 1_{T^2 >= k_alpha_0}
  phi = 1                                           # H0 ablehnen
} else {
  phi = 0                                           # H0 nicht ablehnen
}
p      = 1 - pf(nu*T2, m, n-m)                     # p-Wert

# Ausgabe
cat( "Y_bar   = ", Y_bar,
     "\nC     = ", C,
     "\nT^2    = ", T2,
     "\nalpha_0 = ", alpha_0,
     "\nk      = ", k_alpha_0,
     "\nphi    = ", phi,
     "\np      = ", p, "\n")

Y_bar   = 26.25615 2.991039
C        = 38.8981 3.549813 3.549813 1.972143
T^2      = 7.546368
alpha_0  = 0.05
k        = 7.504065
phi      = 1
p        = 0.04928746
```

Im vorliegenden Fall würden wir also die Nullhypothese eines wahren, aber unbekannten Erwartungswertparameters $\mu_0 = (30, 3.5)$ anhand der beobachteten Daten ablehnen.

Zuletzt führen wir den gleichen Einstichproben- T^2 -Test noch einmal mithilfe des R-Pakets `MVTests`, speziell der Funktion `MVTests::OneSampleHT2()` im Sinne eines Black-Box-Verfahrens durch.

```
# Datenbereitstellung
D      = read.csv("Einstichproben_T2_Tests.csv") # Daten einlesen
Y      = rbind(D$y_1i, D$y_2i)                  # Datenselektion

# Einstichproben- $T^2$ -Test
library(MVTests)                                # R-Paket laden
phi    = OneSampleHT2(t(Y), mu_0, alpha_0)       # Einstichproben- $T^2$ -Test
cat(   "Y_bar    = ", phi$Descriptive[2,],       # Ausgabe
      "\nT^2     = ", phi$HT2,
      "\nalpha_0 = ", phi$alpha,
      "\np       = ", phi$p.value, "\n")
```

```
Y_bar    = 26.25615 2.991039
T^2       = 7.546368
alpha_0   = 0.05
p         = 0.04928746
```