

## (6) Einfaktorielle Varianzanalyse

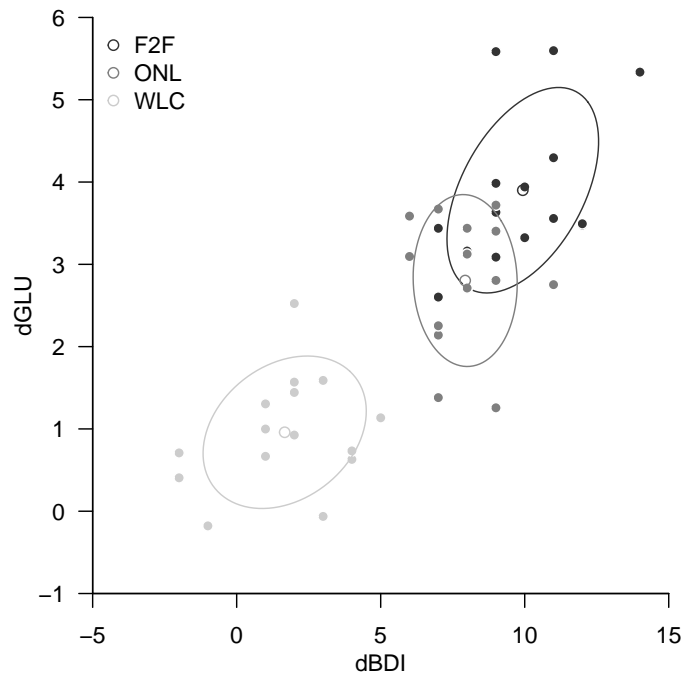
Ziel dieses Seminar ist es, die Durchführung einer einfaktoriellen multivariaten Varianzanalyse an einem Anwendungsbeispiel zu demonstrieren. Dazu betrachten wir die Analyse von Prä-Post-Interventions-BDI-Score und Prä-Post-Interventions- Glukokortikoidplasmalevel Differenzenwerten von drei Gruppen von jeweils 15 Patient:innen, die unterschiedliche Psychotherapiesettings (Face-to-face und Online) bzw. eine Wartelistenkontrollbedingung durchlaufen haben. Wir stellen dazu in Tabelle 1 einen simulierten Beispieldatensatz dar. Die erste Spalte von Tabelle 1 (COND) listet das spezifische Therapiesetting (F2F: Face-to-face, ONL: online, WLC: waitlist control) der Patient:innen auf. Die zweite Spalte (dBDI) listet die entsprechenden BDI-Score-Differenzwerte und die dritte Spalte (dGLU) die entsprechenden Glukokortikoidplasmalevel-Differenzwerte auf. In beiden Fällen sollen positive Werte eine Abnahme der Depressionssymptomatik, negative Werte dagegen einer Zunahme der Depressionssymptomatik anzeigen.

**Tabelle 1.** Prä-Post-Interventions-BDI-II-Score und -Glukokortikoidplasmalevel Differenzenwerte von drei Studiengruppen (F2F: Face-to-face, ONL: online, WLC: waitlist control) jeweils 15 Patient:innen

COND	dBDI	dGLU
F2F	11	4.3
F2F	10	3.9
F2F	12	3.5
F2F	7	2.6
F2F	10	3.3
F2F	12	3.5
F2F	9	3.1
F2F	9	3.6
F2F	9	5.6
F2F	11	3.6
F2F	7	3.4
F2F	9	4.0
F2F	11	5.6
F2F	14	5.3
F2F	8	3.2
ONL	6	3.1
ONL	8	2.7
ONL	7	2.1
ONL	8	3.1
ONL	11	2.8
ONL	9	2.8
ONL	9	3.7
ONL	8	2.7
ONL	6	3.6
ONL	7	1.4
ONL	9	1.3
ONL	8	3.4
ONL	7	3.7
ONL	7	2.3
ONL	9	3.4
WLC	-2	0.7
WLC	2	1.4
WLC	1	1.0
WLC	2	0.9
WLC	3	1.6
WLC	2	1.6
WLC	5	1.1
WLC	-1	-0.2
WLC	2	2.5
WLC	-2	0.4
WLC	1	0.7
WLC	3	-0.1
WLC	1	1.3
WLC	4	0.6
WLC	4	0.7

Folgender **R** Code demonstriert die Auswertung gruppenspezifischer Deskriptivstatistiken für diesen Datensatz, die wir in [Abbildung 1](#) visualisieren.

```
# Studiengruppenspezifische Deskriptivstatistiken
D = read.csv("./6_Daten_S/6_Einfaktorielle_Varianzanalyse.csv") # Dateneinlesen
m = 2 # Datendimension von Interesse
p = 3 # Anzahl Gruppen
k = 15 # Anzahl Datenpunkte pro Gruppe
Y = array(dim = c(m,k,p)) # Datenarrayinitialisierung
Y[, ,1] = rbind(D$dBDI[D$COND == "F2F"], # F2F dBDI Werte
               D$dGLU[D$COND == "F2F"]) # F2F dGLU Werte
Y[, ,2] = rbind(D$dBDI[D$COND == "ONL"], # ONL dBDI Werte
               D$dGLU[D$COND == "ONL"]) # ONL dGLU Werte
Y[, ,3] = rbind(D$dBDI[D$COND == "WLC"], # WLC dBDI Werte
               D$dGLU[D$COND == "WLC"]) # WLC dGLU Werte
y_bar_i = array(dim = c(m,p)) # Stichprobenmittelarray
C_i = array(dim = c(m,m,p)) # Stichprobenkovarianzmatrizenarray
j_k = matrix(rep(1,k), nrow = k) # 1_{1}
I_k = diag(k) # Einheitsmatrix I_1
J_k = matrix(rep(1,k^2), nrow = k) # 1_{11}
for (i in 1:p){ # Gruppeniterationen
  y_bar_i[,i] = (1/k)*(Y[, ,i] %*% j_k) # Stichprobenmittel \bar{\upsilon}_i
  C_i[, ,i] = (1/(k-1))*(Y[, ,i] %*% (I_k - (1/k)*J_k) %*% t(Y[, ,i])) # Stichprobenkovarianzmatrix C_i
}
```



**Abbildung 1.** Deskriptivstatistiken der dBDI, dGLU Daten des Beispieldatensatzes. Jeder Punkt visualisiert die Daten einer Patient:in.

Zur Parameterschätzung nutzen wir folgende **R** Funktion.

```
estimate = function(Y){
# Diese Funktion evaluiert die Parameterschätzer einer einfaktoriellen
# multivariaten Varianzanalyse basierend auf einem m x l x p Datensatz Y.
#
# Input
#   Y           : m x k x p Datenarray
#
# Output
#   $mu_hat     : m x p \mu_i Parameterschätzer
#   $Sigma_hat  : m x m \Sigma Parameterschätzer
# -----
# Dimensionsparameter
d   = dim(Y)                # Datensatzdimensionen
m   = d[1]                  # Datendimension
k   = d[2]                  # Anzahl Datenpunkte pro Gruppe
p   = d[3]                  # Anzahl Gruppen

# Erwartungswertparameterschätzer
mu_hat_i = matrix(apply(Y,3,rowMeans), nrow = m)

# Kovarianzmatrixparameterschätzer
Sigma_hat = matrix(rep(0,m*m), nrow = m)
for(i in 1:p){
  for(j in 1:k){
    Sigma_hat = Sigma_hat + (1/(k*p-p))*(Y[,j,i] - mu_hat_i[,i]) %*% t(Y[,j,i] - mu_hat_i[,i])}
}

# Outputspezifikation
return(list(mu_hat_i = mu_hat_i, Sigma_hat = Sigma_hat))}

```

Zur Kreuzproduktsammenmatrixzerlegung nutzen wir folgende **R** Funktion.

```
sos = function(Y){
# Diese Funktion evaluiert die Kreuzproduktsammenmatrizen T,B,W einer
# einfaktoriellen Varianzanalyse basierend auf einem m x k x p Datensatz Y.
#
# Input
#   Y           : m x k x p Datenarray
#
# Output
#   $y_bar      : m x 1 Gesamtmittelwert
#   $y_bar_i    : m x p Gruppenmittelwerte
#   $T          : m x m Total Sum of Squares Matrix
#   $B          : m x m Between-Group Sum-of-Squares Matrix
#   $W          : m x m Within Group Sum of Squares Matrix
# -----
d   = dim(Y)                # Datensatzdimensionen
m   = d[1]                  # Datendimension
k   = d[2]                  # Anzahl Datenpunkte pro Gruppe
p   = d[3]                  # Anzahl Gruppen

# Mittelwerte
y_bar_i = matrix(apply(Y,3,rowMeans), nrow = m) # Gruppenstichprobenmittel
y_bar   = matrix(rowMeans(y_bar_i), nrow = m)   # Gesamtstichprobenmittel

# Totale Sum-of-Squares Matrix
T = matrix(rep(0,m*m), nrow = m)
for(i in 1:p){
  for(j in 1:k){
    T = T + (Y[,j,i] - y_bar) %*% t(Y[,j,i] - y_bar)}
}

# Between Sum of Squares Matrix
B = matrix(rep(0,m*m), nrow = m)
for(i in 1:p){
  B = B + k*(y_bar_i[,i] - y_bar) %*% t(y_bar_i[,i] - y_bar)}

# Within Sum of Squares Matrix
W = matrix(rep(0,m*m), nrow = m)
for(i in 1:p){
  for(j in 1:k){
    W = W + (Y[,j,i] - y_bar_i[,i]) %*% t(Y[,j,i] - y_bar_i[,i])}
}

# Outputspezifikation
return(list(y_bar_i = y_bar_i, y_bar = y_bar, T = T, B = B, W = W))}

```

Zur Durchführung der einfaktoriellen Varianzanalyse nutzen wir folgenden **R** Code.

```
# Einlesen und Präprozessierung des Datensatzes
D = read.csv("./6_Daten_S/6_Einfaktorielle_Varianzanalyse.csv") # Dateneinlesen
m = 2 # Datendimension von Interesse
p = 3 # Anzahl Gruppen
k = 15 # Anzahl Datenpunkte pro Gruppe
n = p*k # Gesamtanzahl Datenpunkte
Y = array(dim = c(m,k,p)) # Datenarrayinitialisierung
Y[, ,1] = rbind(D$dBDI[D$COND == "F2F"], # F2F dBDI Werte
               D$dGLU[D$COND == "F2F"]) # F2F dGLU Werte
Y[, ,2] = rbind(D$dBDI[D$COND == "ONL"], # ONL dBDI Werte
               D$dGLU[D$COND == "ONL"]) # ONL dGLU Werte
Y[, ,3] = rbind(D$dBDI[D$COND == "WLC"], # WLC dBDI Werte
               D$dGLU[D$COND == "WLC"]) # WLC dGLU Werte

# Einfaktorielle Varianzanalyse
alpha_0 = 0.05 # Signifikanzlevel
S = sos(Y) # Sum of Squares Matrizen
L = det(S$W)/det(S$W + S$B) # Wilks' Lambda
w = n-1-(1/2)*(m+p) # w
t = sqrt((m^2*(p-1)^2-4)/(m^2+(p-1)^2-5)) # t
nu_1 = m*(p-1) # \nu_1
nu_2 = w*t-(1/2)*(m*(p-1)-2) # \nu_2
tau = ((1-L^(1/t))/L^(1/t))*(nu_2/nu_1) # Teststatistik
k_alpha_0 = qf(1-alpha_0,nu_1,nu_2) # kritischer Wert
phi = as.numeric(tau > k_alpha_0) # Nullhypothesentest
P = 1-pf(tau,nu_1,nu_2) # Überschreitungswahrscheinlichkeit

# Ausgabe
cat("Wilks' Lambda      ", L,
    "\ntau              ", tau,
    "\nnu_1              ", nu_1,
    "\nnu_2              ", nu_2,
    "\nphi              ", phi,
    "\nP(tau > tau_tilde)", P)
```

```
Wilks' Lambda      0.1569049
tau                31.25301
nu_1                4
nu_2                82
phi                1
P(tau > tau_tilde) 8.881784e-16
```

Im vorliegenden Fall wird die Nullhypothese identischer Gruppenerwartungswertparameter also verworfen. Schließlich validieren wir obige Analyse im Sinne eines Black-Box-Verfahrens mithilfe der **R** `lm()` und `Manova()` Funktionen.

```
library(car)
D = read.csv("./6_Daten_S/6_Einfaktorielle_Varianzanalyse.csv") # Dateneinlesen
model = lm(cbind(D$dBDI,D$dGLU) ~ D$COND, D) # Modellspezifikation
Manova(model, test.statistic = "Wilks") # Einfaktorielle Varianzanalyse
```

```
Type II MANOVA Tests: Wilks test statistic
      Df test stat approx F num Df den Df Pr(>F)
D$COND 2 0.1569 31.253 4 82 8.346e-16 ***
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```