

## (7) Kanonische Korrelationsanalyse

Wir erinnern zunächst an die Bestimmung eines Korrelationskoeffizienten bei univariaten unabhängigen und abhängigen Variablen.

```
# Laden des Beispieldatensatzes
D      = read.csv("../7_Daten_S/7_Kanonische_Korrelationsanalyse.csv")
x_i    = D$DUR
y_i    = D$dBDI
n      = length(x_i)

# "Manuelle" Berechnung der Stichprobenkorrelation
x_bar  = (1/n)*sum(x_i)
y_bar  = (1/n)*sum(y_i)
s_x    = sqrt(1/(n-1)*sum((x_i - x_bar)^2))
s_y    = sqrt(1/(n-1)*sum((y_i - y_bar)^2))
c_xy   = 1/(n-1) * sum((x_i - x_bar) * (y_i - y_bar))
r_xy   = c_xy/(s_x * s_y)
print(r_xy)
```

```
[1] 0.8829308
```

```
# Automatische Berechnung mit cor()
r_xy   = cor(x_i,y_i)
print(r_xy)
```

```
[1] 0.8829308
```

Wir führen dann eine kanonische Korrelationsanalyse an dem in der Vorlesung betrachteten Anwendungsbeispiel durch.

```

# R Paket
library(expm)

# Datenpräprozessierung
D      = read.csv("../Daten_S/7_Kanonische_Korrelationsanalyse.csv")
x      = as.matrix(cbind(D$DUR , D$EXP))
y      = as.matrix(cbind(D$dBDI, D$dGLU))
n      = nrow(x)
m_x    = ncol(x)
m_y    = ncol(y)
Y      = t(cbind(x,y))

# Stichprobenkovarianzmatrixpartition
I_n    = diag(n)
J_n    = matrix(rep(1,n^2), nrow = n)
C      = (1/(n-1))*(Y %%% (I_n-(1/n)*J_n) %%% t(Y))
C_xx   = C[1:m_x,1:m_x]
C_xy   = C[1:m_x,(m_x+1):(m_x+m_y)]
C_yx   = C[(m_x+1):(m_x+m_y),1:m_x]
C_yy   = C[(m_x+1):(m_x+m_y),(m_x+1):(m_x+m_y)]

# Kanonische Korrelationsanalyse
K_hat  = sqrtm(solve(C_xx)) %%% C_xy %%% sqrtm(solve(C_yy))
ALB_hat = svd(K_hat)
A_hat  = ALB_hat$u
Lambda_hat = ALB_hat$d
B_hat  = ALB_hat$v
a_hat  = sqrtm(solve(C_xx)) %%% A_hat
b_hat  = sqrtm(solve(C_yy)) %%% B_hat
rho_hat = as.matrix(Lambda_hat)

# Ausgabe
cat("rho_hat_1 : " , rho_hat[1],
    "\na_hat_1   : ", a_hat[,1],
    "\nb_hat_1   : ", b_hat[,1],
    "\nrho_hat_2 : ", rho_hat[2],
    "\na_hat_2   : ", a_hat[,2],
    "\nb_hat_2   : ", b_hat[,2])

rho_hat_1 : 0.9950575
a_hat_1   : -0.1623409 -0.173979
b_hat_1   : -0.1554175 -0.05025419
rho_hat_2 : 0.5010358
a_hat_2   : -0.06026274 0.3118808
b_hat_2   : -0.08128072 0.7773036

```

Die gleiche Analyse können wir auch im Sinne eines Black-Box-Verfahrens mit **R**'s' `cancor()` Funktion durchführen.

```
# Datenpräprozessierung
D      = read.csv("../7_Daten_S/7_Kanonische_Korrelationsanalyse.csv")
x      = as.matrix(cbind(D$DUR , D$EXP))
y      = as.matrix(cbind(D$dBDI, D$dGLU))
cca    = cancor(x,y)
```

```
rho_hat_1 : 0.9950575
```

```
rho_hat_2 : 0.5010358
```