Multivariate Datenanalyse

MSc Psychologie WiSe 2021/22

Prof. Dr. Dirk Ostwald

(11) Einfaktorielle Varianzanalyse

Anwendungsszenario und Datendeskription

Modellformulierung und Modellschätzung

Modellevaluation

- Wilks' Λ
- Pillai's V

Selbstkontrollfragen

Anwendungsszenario und Datendeskription

Modellformulierung und Modellschätzung

Modellevaluation mit

- Wilks' Λ
- Pillai's V

Selbstkontrollfragen

Anwendungsszenario und Datendeskription

Anwendungsszenario

- Zwei oder mehr Stichproben (oft Gruppen genannt) experimenteller Einheiten.
- Annahme der unabhängigen und identischen Normalverteilung $N(\mu_i, \Sigma)$ der Daten.
- $\mu_i \in \mathbb{R}^m, i = 1, ..., k$ und $\Sigma \in \mathbb{R}^{m \times m}$ p.d. unbekannt.
- Absicht des inferentiellen Testens der Nullhypothese identischer Gruppenerwartungswerte.

Generalisierung des Zweistichproben-T^2-Tests bei unabhängigen Stichproben mit einfacher Nullhypothese für mehr als zwei Stichproben

Anwendungsbeispiele

- BDI/Glukokortikoid Analyse von drei Gruppen psychiatrischer Patient:innen nach PA, CBT,ST
  - Inferentielle Evidenz für Gruppenerwartungswertunterschiede?
  - Evidenz für unterschiedliche Therapiewirksamkeit?
- Gruppenvergleich von Testdaten bei gutem, befriedigendem, ungenügenden Studienabschluss
  - Inferentielle Evidenz für Gruppenerwartungswertunterschiede?
  - Evidenz für Studienerfolgsprädiktivität der Testdaten?

Anwendungsbeispiel

Nach Rudolf and Buse (2020) Kapitel 4, vgl. Einheiten zur Prädiktiven Modellierung

Datensatz zum Verhältnis psychologischer Testdiagnostik und Studienerfolg

Faktor Studienerfolg mit $k = 3$ Leveln (*Ungenügend, Befriedigend, Gut*)

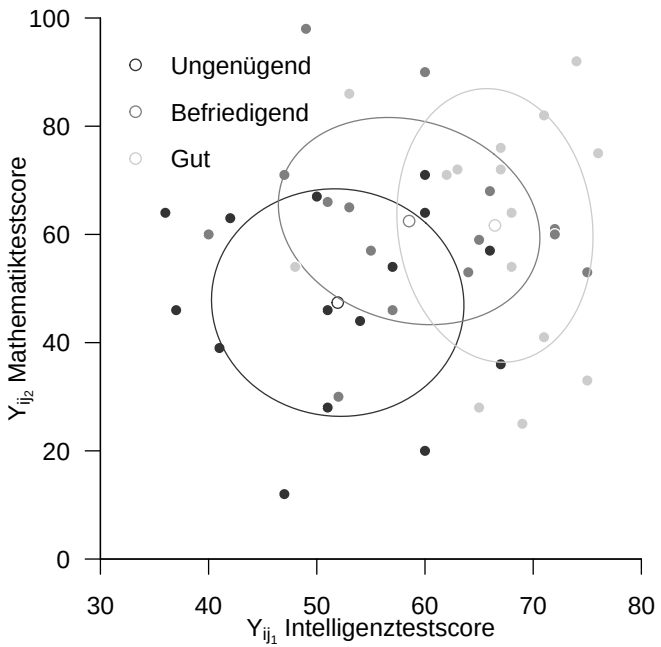
Datendimension $m = 2$ mit $y_{ij} \in \mathbb{R}^2$ (y_{ij_1} *Intelligenztestscore*, y_{ij_2} *Mathematiktestscore*)

```
# Einlesen des Datensatzes aus Rudolf & Buse (2019) Multivariate Verfahren
library("foreign")
S       = read.spss(file.path(getwd(), "11_Daten", "studienerfolg.sav"), to.data.frame = T)

# Datenpräprozessierung
m       = 2                                     # Datendimension von Interesse
k       = 3                                     # Anzahl Gruppen
l       = 15                                    # Anzahl Datenpunkte pro Gruppe
Y       = array(dim = c(m,l,k))                 # Datenarrayinitialisierung
Y[,,1]  = rbind(S$X1[S$Gruppe == "ungenügend"],   # Y_{1j_1} Intelligenztestscore
                S$X2[S$Gruppe == "ungenügend"])   # Y_{1j_2} Mathematiktestscore
Y[,,2]  = rbind(S$X1[S$Gruppe == "befriedigend"], # Y_{2j_1} Intelligenztestscore
                S$X2[S$Gruppe == "befriedigend"]) # Y_{2j_2} Mathematiktestscore
Y[,,3]  = rbind(S$X1[S$Gruppe == "gut"],          # Y_{3j_1} Intelligenztestscore
                S$X2[S$Gruppe == "gut"])          # Y_{3j_2} Mathematiktestscore
```

Anwendungsbeispiel

Datendeskription

```
Y_bar_i = array(dim = c(m,k))                                                    # Stichprobenmittelarray
C_i     = array(dim = c(m,m,k))                                                  # Stichprobenkovarianzmatrizenarray
j_l     = matrix(rep(1,l), nrow = l)                                             # 1_{l}
I_l     = diag(l)                                                                # Einheitsmatrix I_l
J_l     = matrix(rep(1,l^2), nrow = l)                                           # 1_{ll}
for (i in 1:k){                                                                  # Gruppeniterationen
    Y_bar_i[,i] = (1/l)*(Y[,,i] %*% j_l)                                         # Stichprobenmittel \bar{Y}_i
    C_i[,,i]    = (1/(l-1))*(Y[,,i] %*% (I_l-(1/l)*J_l) %*% t(Y[,,i]))}          # Stichprobenkovarianzmatrix C_i
```



Anwendungsszenario und Datendeskription

Modellformulierung und Modellschätzung

Modellevaluation

- Wilks' Λ
- Pillai's V

Selbstkontrollfragen

Definition (Modell der einfaktoriellen Varianzanalyse)

Für $i = 1, \ldots, k$ sei

$$Y_{i1}, \ldots, Y_{il} \sim N(\mu_i, \Sigma) \tag{1}$$

eine Stichprobe eines multivariaten Normalverteilungsmodells von Größe l mit unbekanntem Erwartungswertparameter $\mu_i \in \mathbb{R}^m$ und unbekannten Kovarianzmatrixparameter $\Sigma \in \mathbb{R}^{m \times m}$ p.d. Dann heißt (1) *Klassisches Modell der einfaktoriellen Varianzanalyse*. Die Gesamtstichprobengröße bezeichnen wir mit $n := kl$. Äquivalent sei

$$Y_{ij} = \mu_i + \varepsilon_{ij} \text{ mit } \varepsilon_{ij} \sim N(0_m, \Sigma) \text{ für } i = 1, \ldots, k \text{ und } j = 1, \ldots, l, \tag{2}$$

wobei

- i die Stichproben und j die experimentellen Einheiten indizieren,
- l die Stichprobengrößen und $n := lk$ die Gesamtstichprobengröße sind,
- Y_{ij} beobachtbare Zufallsvektoren sind,
- $\mu_i \in \mathbb{R}^m$ feste Erwartungswertparameter der Stichprobenvariablen sind, und
- ε_{ij} unabhängige normalverteilte nicht-beobachtbare Zufallsvariablen mit $\Sigma \in \mathbb{R}^{m \times m}$ p.d. sind.

Dann heißt (2) *Generatives Modell der einfaktoriellen Varianzanalyse.*

Bemerkungen

- Der Einfachheit halber setzen wir hier identische Stichprobengrößen voraus.
- Die Kovarianzmatrixparameter aller Stichproben werden als identisch vorausgesetzt.
- Wir verzichten auf einen Beweis der Äquivalenz von (1) und (2).
- Die Gesamtheit aller Stichprobenzufallsvektoren bezeichen wir mit $Y := \{Y_{ij}\}_{i=1,\ldots,k, j=1,\ldots,l}$.

Theorem (Parameterschätzer der einfaktoriellen Varianzanalyse)

Für $i = 1, \ldots, k$ sei mit

$$Y_{i1}, \ldots, Y_{il} \sim N(\mu_i, \Sigma) \tag{3}$$

für $\mu_i \in \mathbb{R}^m$ und $\Sigma \in \mathbb{R}^{m \times m}$ p.d. das Modell der einfaktoriellen Varianzanalyse gegeben. Dann ist für $i = 1, \ldots, k$

$$\hat{\mu}_i := \frac{1}{l} \sum_{j=1}^{l} Y_{ij} \tag{4}$$

ein unverzerrte Schätzer des Erwartungswertparameters μ_i und

$$\hat{\Sigma} := \frac{1}{lk - k} \sum_{i=1}^{k} \sum_{j=1}^{l} \left(Y_{ij} - \hat{\mu}_i \right) \left(Y_{ij} - \hat{\mu}_i \right)^T \tag{5}$$

ein unverzerrter Schätzer des Kovarianzmatrixparameters Σ.

Bemerkungen

- $\hat{\mu}_i$ ist das Stichprobenmittel der $i$ten Gruppe.
- $\hat{\Sigma}$ ist die mit $(lk - k)$ skalierte Within Group Sum of Squares Matrix (siehe unten).
- Anstelle eines Beweis validieren wir die Aussage des Theorems mithilfe einer Simulation.

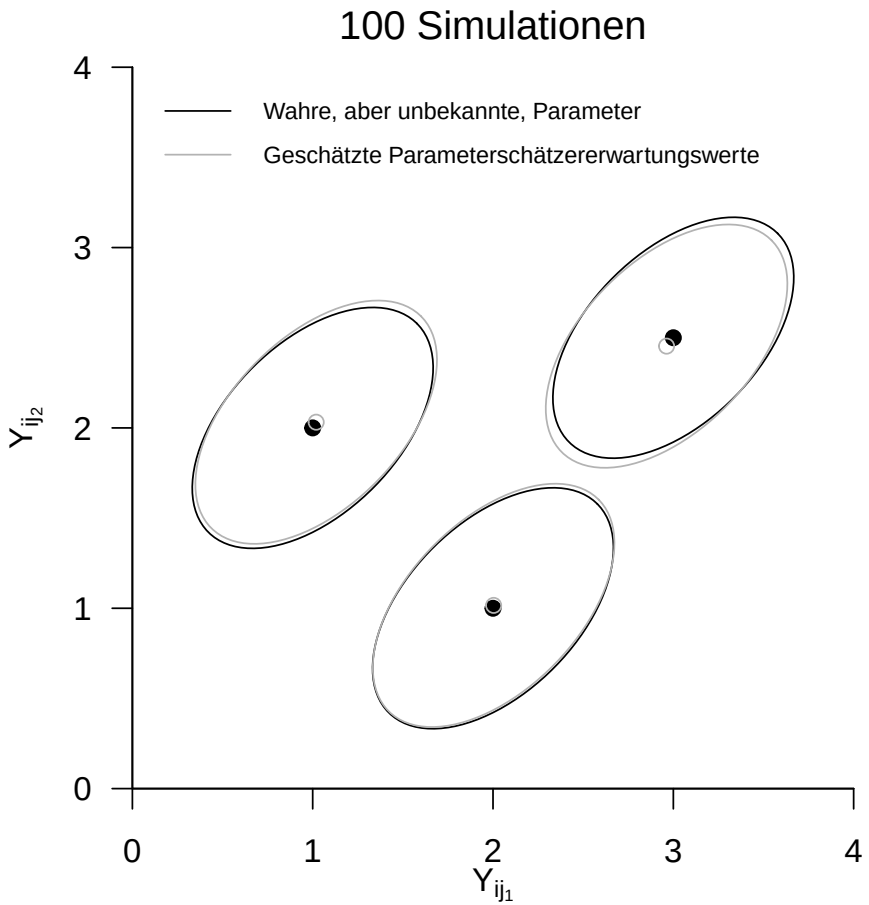
Parameterschätzer der einfaktoriellen Varianzanalyse

```
estimate = function(Y){

  # Diese Funktion evaluiert die Parameterschätzer einer einfaktoriellen
  # Varianzanalyse basierend auf einen m x l x k Datensatz Y.
  #
  # Input
  #     Y          : m x l x k Datenarray
  #
  # Output
  #     $mu_hat    : m x k \mu_i Parameterschätzer
  #     $Sigma_hat : m x m \Sigma Parametschätzer
  # ------------------------------------------------------------------------
  # Dimensionsparameter
  d         = dim(Y)                                    # Datensatzdimensionen
  m         = d[1]                                      # Datendimension
  l         = d[2]                                      # Anzahl Datenpunkte pro Gruppe
  k         = 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:k){
      for(j in 1:l){
          Sigma_hat = Sigma_hat + (1/(l*k-k))*(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))}
```

Parameterschätzer der einfaktoriellen Varianzanalyse

```r
# R Pakete
library(MASS)                                       # multivariate Normalverteilungen
library(matlib)                                     # Matrixalgebra


# Modellparameter
k           = 3                                     # Anzahl Gruppen
l           = 15                                    # Anzahl Datenpunkte pro Gruppe
m           = 2                                     # Datendimensionalität
mu_i        = matrix(c(1,2,2,1,3,2.5), ncol = k)    # Erwartungswertparameter
Sigma       = matrix(c(1,.5,.5,1)   , ncol = m)     # Kovarianzmatrixparameter

# Simulationsparameter und Arrays
nsm         = 1e2                                   # Anzahl Simulation
mu_hat_is   = array(dim = c(m,k,nsm))               # \hat{\mu}_i Array
Sigma_hats  = array(dim = c(m,m,nsm))               # \hat{\Sigma} Array

# Simulationen
for(s in 1:nsm){

    # Datengeneration
    Y               = array(dim = c(m,l,k))       # Datenarray
    for(i in 1:k){
        Y[,,i] = t(mvrnorm(l,mu_i[,i],Sigma))     # Datengeneration
    }
    S               = estimate(Y)                 # Parameterschätzung
    mu_hat_is[,,s]  = S$mu_hat_i                  # \hat{\mu}_i
    Sigma_hats[,,s] = S$Sigma_hat                 # \hat{\Sigma}
}

# Erwartungswertschätzung
E_hat_mu_i_hat  = apply(mu_hat_is , c(1,2), mean)
E_hat_Sigma_hat = apply(Sigma_hats, c(1,2), mean)
```

Parameterschätzer der einfaktoriellen Varianzanalyse



Anwendungsbeispiel

Parameterschätzung

```
# Parameterschätzung
S = estimate(Y)

# Ausgabe
print(S$mu_hat_i)
```

```
>      [,1] [,2] [,3]
> [1,] 51.9 58.5 66.5
> [2,] 47.4 62.5 61.7
```

```
print(S$Sigma_hat)
```

```
>       [,1]  [,2]
> [1,]  87.6 -14.9
> [2,] -14.9 348.3
```

Anwendungsszenario und Datendeskription

Modellformulierung und Modellschätzung

Modellevaluation

- Wilks' Λ
- Pillai's V

Selbstkontrollfragen

Überblick

Primäres Ziel der einfaktoriellen Varianzanalyse ist zumeist das Testen der Nullhypothese

$$H_0 : \mu_1 = \mu_2 = \cdots = \mu_k. \tag{6}$$

Die Nullhypothese gesagt also, dass zwischen den Gruppen keine Erwartungswertparameterunterschiede bestehen. Die Alternativhypothese lautet somit

$$H_1 : \mu_{i_c} \neq \mu_{j_c} \text{ für mindestens ein Paar } (i, j) \text{ mit } i \neq j \text{ und mindestens ein } c \text{ mit } 1 \leq c \leq m. \tag{7}$$

Die Alternativhypothese besagt also, dass sich mindestens zwei Erwartungswertparameter in mindestens einer ihrer Komponenten unterscheiden.

Kritische Wert-basierte Tests der Nullhypothesen können mit verschiedenen Teststatistiken (*Wilks'* Λ*, Pillai Statisik*) konstruiert werden. Diesen Teststatistiken ist gemein, dass sie auf eine Generalisierung der vom univariaten Fall bekannten Quadratsummenzerlegung zurück gehen. Wir führen also zunächst diese sogenannte *Kreuzproduktsummenmatrizenzerlegung* ein und betrachten dann die durch die Wilks' Λ und die Bartlett-Pillai-Spur induzierten Tests anhand der Gliederung (1) Teststatistik und Test, (2) Analyse der Teststatistik und (3) Testumfangkontrolle.

Einen Überblick über die Modellevaluation bei der univariaten einfaktoriellen Varianzanalyse gibt (12) Varianzanalysen.

Theorem (Kreuzproduktsummenmatrizenzerlegung)

Für $i = 1, \ldots, k$ und $j = 1, \ldots, l$ bezeichne Y_{ij} den $j$ten Stichprobenvektor der $i$ten Stichprobengruppe eines einfaktoriellen Varianzanalysemodells. Weiterhin seien

$$\bar{Y} := \frac{1}{n} \sum_{i=1}^{k} \sum_{j=1}^{l} Y_{ij} \text{ und } \bar{Y}_i := \frac{1}{l} \sum_{j=1}^{l} Y_{ij} \tag{8}$$

das *Gesamtstichprobenmittel* und das i*te Gruppenstichprobenmittel*, respektive. Schließlich seien

$T := \sum_{i=1}^{k} \sum_{j=1}^{l} \left(Y_{ij} - \bar{Y}\right) \left(Y_{ij} - \bar{Y}\right)^T$ die *Totale Sum of Squares Matrix*

$B := \sum_{i=1}^{k} l \left(\bar{Y}_i - \bar{Y}\right) \left(\bar{Y}_i - \bar{Y}\right)^T$ die *Between Group Sum of Squares Matrix*

$W := \sum_{i=1}^{k} \sum_{j=1}^{l} \left(Y_{ij} - \bar{Y}_i\right) \left(Y_{ij} - \bar{Y}_i\right)^T$ die *Within Group Sum of Squares Matrix*.

Dann gilt

$$T = B + W. \tag{9}$$

Bemerkungen

- $T \in \mathbb{R}^{m \times m}$ repräsentiert die totale Variabilität der Datenvektoren um das Gesamtstichprobenmittel.
- $B \in \mathbb{R}^{m \times m}$ repräsentiert die Variabilität der Gruppenstichprobenmittel um das Gesamtstichprobenmittel.
- $W \in \mathbb{R}^{m \times m}$ repräsentiert die Variabilität der Datenvektoren um ihre jeweiligen Gruppenstichprobenmittel.
- Die totale Variabilität wird hier also in zwei unabhängige Beiträge von Variabilität zerlegt.
- W heißt auch *Residualvariabiliät*, weil sie die verbleibende Variabilität nach Schätzung der Gruppenerwartungswertparameter quantifiziert und gilt das

$$W = (lk - k)\hat{\Sigma}. \tag{10}$$

- Die Kreuzproduktsummenmatrixzerlegung ergibt sich anhand von algebraischen Identitäten wie unten gezeigt.

Beweis

$$
\begin{aligned}
T &= \sum_{i=1}^{k}\sum_{j=1}^{l}(Y_{ij}-\bar{Y})(Y_{ij}-\bar{Y})^T \\
&= \sum_{i=1}^{k}\sum_{j=1}^{l}(Y_{ij}-\bar{Y}_i+\bar{Y}_i-\bar{Y})(Y_{ij}-\bar{Y}_i+\bar{Y}_i-\bar{Y})^T \\
&= \sum_{i=1}^{k}\sum_{j=1}^{l}\Big((Y_{ij}-\bar{Y}_i)+(\bar{Y}_i-\bar{Y})\Big)\Big((Y_{ij}-\bar{Y}_i)+(\bar{Y}_i-\bar{Y})\Big)^T \\
&= \sum_{i=1}^{k}\sum_{j=1}^{l}\Big((Y_{ij}-\bar{Y}_i)(Y_{ij}-\bar{Y}_i)^T+2(Y_{ij}-\bar{Y}_i)(\bar{Y}_i-\bar{Y})^T+(\bar{Y}_i-\bar{Y})(\bar{Y}_i-\bar{Y})^T\Big) \\
&= \sum_{i=1}^{k}\left(\sum_{j=1}^{l}(Y_{ij}-\bar{Y}_i)(Y_{ij}-\bar{Y}_i)^T+\sum_{j=1}^{l}2(Y_{ij}-\bar{Y}_i)(\bar{Y}_i-\bar{Y})^T+\sum_{j=1}^{l}(\bar{Y}_i-\bar{Y})(\bar{Y}_i-\bar{Y})^T\right)
\end{aligned}
$$

Beweis (fortgeführt)

$$
\begin{aligned}
&= \sum_{i=1}^{k} \left(\sum_{j=1}^{l} (Y_{ij} - \bar{Y}_i)(Y_{ij} - \bar{Y}_i)^T + 2 \left(\sum_{j=1}^{l} (Y_{ij} - \bar{Y}_i) \right) (\bar{Y}_i - \bar{Y})^T + l(\bar{Y}_i - \bar{Y})(\bar{Y}_i - \bar{Y})^T \right) \\
&= \sum_{i=1}^{k} \left(\sum_{j=1}^{l} (Y_{ij} - \bar{Y}_i)(Y_{ij} - \bar{Y}_i)^T + 2 \left(\sum_{j=1}^{l} \left(Y_{ij} - \frac{1}{l} \sum_{j=1}^{l} Y_{ij} \right) \right) (\bar{Y}_i - \bar{Y})^T + l(\bar{Y}_i - \bar{Y})(\bar{Y}_i - \bar{Y})^T \right) \\
&= \sum_{i=1}^{k} \left(\sum_{j=1}^{l} (Y_{ij} - \bar{Y}_i)(Y_{ij} - \bar{Y}_i)^T + 2 \left(\sum_{j=1}^{l} Y_{ij} - \sum_{j=1}^{l} Y_{ij} \right) (\bar{Y}_i - \bar{Y})^T + l(\bar{Y}_i - \bar{Y})(\bar{Y}_i - \bar{Y})^T \right) \\
&= \sum_{i=1}^{k} \left(\sum_{j=1}^{l} (Y_{ij} - \bar{Y}_i)(Y_{ij} - \bar{Y}_i)^T + l(\bar{Y}_i - \bar{Y})(\bar{Y}_i - \bar{Y})^T \right) \\
&= \sum_{i=1}^{k} l(\bar{Y}_i - \bar{Y})(\bar{Y}_i - \bar{Y})^T + \sum_{i=1}^{k} \sum_{j=1}^{l} (Y_{ij} - \bar{Y}_i)(Y_{ij} - \bar{Y}_i)^T \\
&= B + W.
\end{aligned}
$$

Modellevaluation

```r
sos = function(Y){

  # Diese Funktion evaluiert die Kreuzproduktsummenmatrizen T,B,W einer
  # einfaktoriellen Varianzanalyse basierend auf einen m x l x k Datensatz Y.
  #
  # Input
  #     Y        : m x l x k Datenarray
  #
  # Output
  #     $Y_bar   : m x 1 Gesamtmittelwert
  #     $Y_bar_i : m x k 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
  l        = d[2]                                      # Anzahl Datenpunkte pro Gruppe
  k        = 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:k){
      for(j in 1:l){
          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:k){
    B = B + l*(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:k){
      for(j in 1:l){
          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))}
```

Überblick zu Teststatistiken und ihren Verteilungen

Basierend auf der Kreuzproduktsummenmatrixzerlegung $T = B + W$ wurden eine Reihe von Teststatisken für Hypothesentests der Nullhypothese $H_0 : \mu_1 = \cdots \mu_k$ vorgeschlagen. Wir betrachten hier (nur)

$$\text{Wilks' } \Lambda = \frac{|W|}{|B+W|} \text{ und Pillai's } V := \text{tr}\left((B+W)^{-1}B\right) \tag{11}$$

Im Gegensatz zur T^2-Teststatistik und zur F-Teststatistik der univariaten Varianzanalyse sind die Verteilungen von Λ und V nur für bestimmte Anwendungsszenarien, d.h. kleine Werte der Datendimensionalität m und die Anzahl der Gruppen k, analytisch exakt beschreibbar und führen, wie im Fall der T^2-Teststatistik auf f-Verteilungen.

Für Anwendungsszenarien mit größeren Werten von m und oder k existieren lediglich Approximationen der Verteilungen von Λ und V, die für unendlich große Stichprobenumfänge $n \to \infty$ exakt sind und wiederum durch f-Verteilungen gegeben sind.

Anderson (2003), Kapitel 8 gibt eine Einführung in die Approximationstheorie für multivariate Modelle, das Wissen um die exakten Verteilungen der Teststatistiken um 1970 wird von Rao (1972) zusammengefasst. Im Sinne der Anwendung unterscheiden sich Testentscheidungen basierend auf exakten oder approximativen Verteilungen von Wilk's Λ und Pillai's V nicht. Zur Absicherung dieser Aussage mögen im konkreten Fall von m und k Simulationen wie im Folgenden diskutiert helfen, Unterschiede zwischen Λ und V und der Approximation ihrer Verteilungen abzuschätzen.

Anwendungsszenario und Datendeskription

Modellformulierung und Modellschätzung

Modellevaluation

- **Wilks' Λ**
- Pillai's V

Selbstkontrollfragen

Definition (Wilks' Λ)

Es seien das Modell der einfaktoriellen Varianzanalyse sowie die Between Sum of Squares Matrix B und die Within Sum of Squares Matrix W definiert wie oben. Dann ist die Wilks' Λ Teststatistik definiert als

$$\Lambda := \frac{|W|}{|B+W|}, \tag{12}$$

wobei $|\cdot|$ die Determinante bezeichnet.

Bemerkungen

- Intuitiv misst Λ das Verhältnis von Residualvariabilität und Gesamtvariabilität.
- Ohne Beweis halten wir fest, dass $\Lambda \in [0,1]$
- Für $\bar{Y}_1 = \cdots = \bar{Y}_p = \bar{Y}$ gilt $B = 0_{mm}$ und damit $\Lambda = 1$.
- Für steigende Unterschiede zwischen den $\bar{Y}_i$ nimmt $|B+W|$ gegenüber $|W|$ zu, Λ also ab.
- Kleine Werte von Λ sprechen also für eine Abweichung von der Nullhypothese.

Theorem (Eigenwertform von Wilks' Λ)

Es seien das Modell der einfaktoriellen Varianzanalyse, die Between Sum of Squares Matrix B, die Within Sum of Squares Matrix W und Wilks' Λ definiert wie oben. Weiterhin seien $\lambda_1, \ldots, \lambda_s$ die Eigenwerte von $W^{-1}B$. Dann gilt

$$\Lambda = \prod_{i=1}^{s} \frac{1}{1+\lambda_i} \tag{13}$$

Bemerkungen

- Wir verzichten auf einen Beweis.
- Die Matrix $W^{-1}B$ ist das multivariate Analogon zu $\frac{\text{SQB}}{\text{SQB}}$.

Theorem (Spezielle H_0 Verteilungen von Wilks' Λ Transformationen)

Es seien das Modell der einfaktoriellen Varianzanalyse und Wilks' Λ definiert wie oben und es gelte außerdem

$$H_0 : \mu_1 = \mu_2 = \cdots = \mu_p \text{ bzw. } H_0 : \alpha_1 = \alpha_2 = \cdots = \alpha_p = 0_m \tag{14}$$

Dann sind für die in den ersten beiden Tabellenspalten aufgeführten Spezialfällen die in der dritten Tabellenspalte aufgeführten Statistiken f-Zufallsvariablen und zwar mit den in der vierten Tabellenspalte aufgeführten Parametern.

Datendimension m	Gruppenanzahl k	Statistik	f-Verteilungsparameter
Beliebig	2	$\frac{1-\Lambda}{\Lambda}\,\frac{n-k-m+1}{m}$	$m, n-k-m+1$
Beliebig	3	$\frac{1-\sqrt{\Lambda}}{\sqrt{\Lambda}}\,\frac{n-k-m+1}{m}$	$2m, 2(n-k-m+1)$
1	Beliebig	$\frac{1-\Lambda}{\Lambda}\,\frac{n-k}{k-1}$	$k-1, n-k$
2	Beliebig	$\frac{1-\sqrt{\Lambda}}{\sqrt{\Lambda}}\,\frac{n-k-1}{k-1}$	$2(k-1), 2(n-k-1)$

Bemerkungen

- Die Verteilungen gehen zurück auf Wilks (1932).

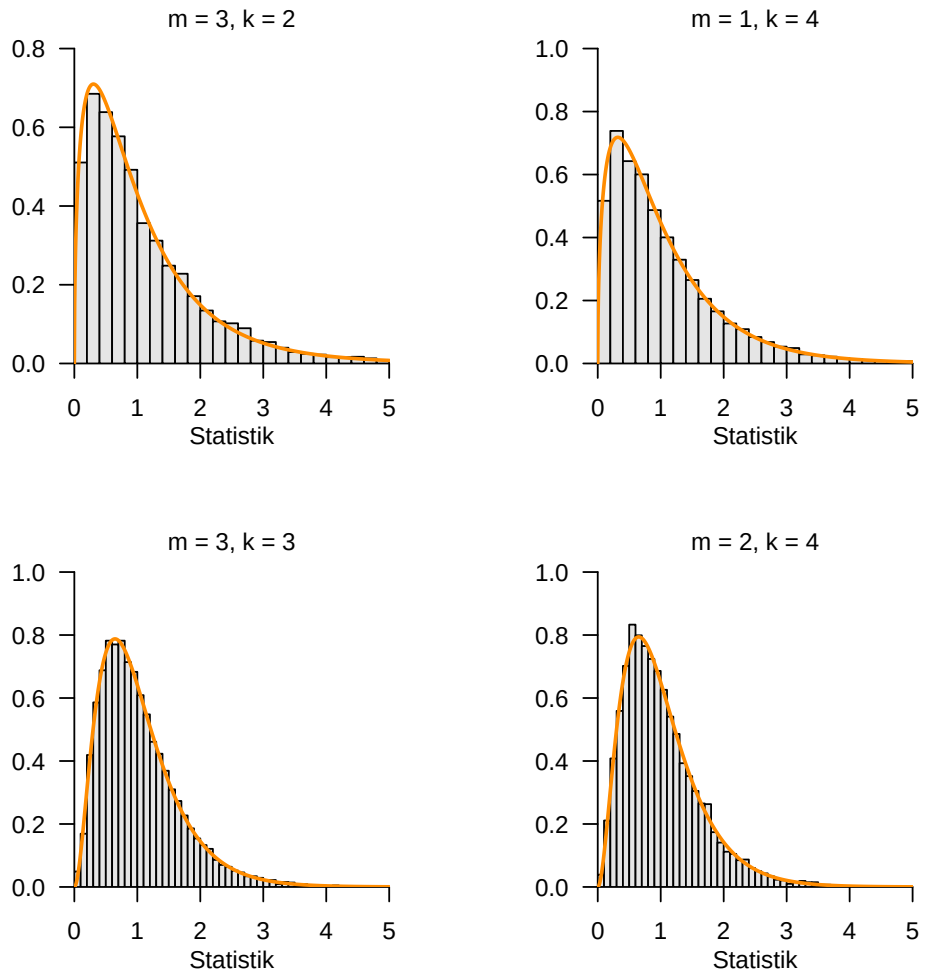
Modellevaluation mit der Wilks' Λ Statistik

Simulation spezieller H_0 Verteilungen von Wilks' Λ Transformationen

```
# Szenarioparameter
nsm = 1e4                                          # Datensimulationsanzahl
M   = c(3,3,1,2)                                   # Datendimension
K   = c(2,3,4,4)                                   # Gruppenanzahl
l   = 15                                           # Datenpunkte pro Gruppe
N   = l*K                                          # Gesamtanzahl Datenpunkte

# Szenariensimulationen
library(MASS)                                      # R Paket für multivariate Normalverteilungen
nsc = length(M)                                    # Szenarienanzahl
P   = matrix(rep(NaN,nsm*nsc), ncol = nsc)         # Statistik ("Prüfgröße") Array
for(sc in 1:nsc){                                  # Szenarioiterationen

    # Modellparameter
    m     = M[sc]                                  # Datendimension
    k     = K[sc]                                  # Gruppenanzahl
    n     = N[sc]                                  # Gesamtanzahl Datenpunkte
    mu_i  = matrix(rep(1,m), nrow = m)             # Identische Gruppenerwartungswertparameter bei H_0
    Sigma = diag(m)                                # Identische Gruppenkovarianzmatrixparameter

    # Datensimulationen
    for(sm in 1:nsm){

        # Datengeneration
        Y     = array(dim = c(m,l,k))              # Datenanarrayinitialisierung
        for(i in 1:k){                             # Gruppeniterationen
            Y[,,i] =  t(mvrnorm(l,mu_i,Sigma))}    # Datensimulation

        # Analyse
        S        = sos(Y)                          # Stichprobenmittel und Sum of Squares Matrizen
        L        = det(S$W)/det(S$W + S$B)         # Wilks' Lambda

        # Szenarioabhängige Statistik ("Prüfgröße")
        if     (sc == 1){p = ((1-L)/L)*((n-k-m+1)/m)}
        else if(sc == 2){p = ((1-sqrt(L))/sqrt(L))*((n-k-m+1)/m)}
        else if(sc == 3){p = ((1-L)/L)*((n-k)/(k-1))}
        else if(sc == 4){p = ((1-sqrt(L))/sqrt(L))*((n-k-1)/(k-1))}

        # Statistikrealisation
        P[sm,sc] = p }}
```

Modellevaluation mit der Wilks' Λ Statistik

Simulation spezieller H_0 Verteilungen von Wilks' Λ Transformationen



Theorem (Approximative H_0 Verteilungen von Wilks' Λ Transformationen)

Es seien das Modell der einfaktoriellen Varianzanalyse und Wilks' Λ definiert wie oben und es gelte außerdem die Nullhypothese

$$H_0 : \mu_1 = \mu_2 = \cdots = \mu_p \text{ bzw. } H_0 : \alpha_1 = \alpha_2 = \cdots = \alpha_p = 0_m \tag{15}$$

Dann ist die Statistik

$$\tau := \frac{1 - \Lambda^{1/t}}{\Lambda^{1/t}} \frac{\nu_2}{\nu_1} \tag{16}$$

mit

$$\nu_1 := m(k-1) \text{ und } \nu_2 := wt - \frac{1}{2}(m(k-1) - 2) \tag{17}$$

sowie

$$w := n - 1 - \frac{1}{2}(m + k) \text{ und } t := \sqrt{\frac{m^2(k-1)^2 - 4}{m^2 + (k-1)^2 - 5}} \tag{18}$$

approximativ f-verteilt mit Freiheitsgradparametern ν_1 und ν_2.

Bemerkungen

- Die Approximation geht zurück auf Rao (1951).

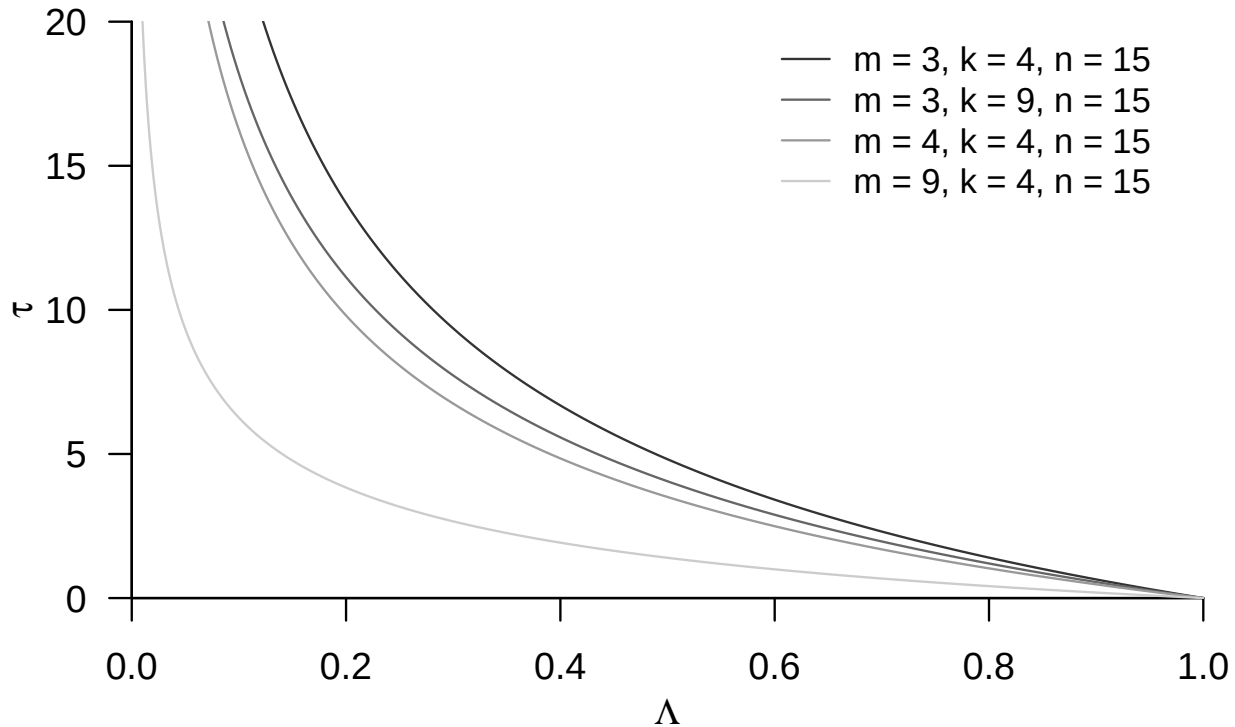
Modellevaluation mit der Wilks' Λ Statistik

Simulation approximativer H_0 Verteilungen von Wilks' Λ Transformationen

```
# Szenarioparameter
library(MASS)                                          # # R Paket für multivariate Normalverteilungen
nsm = 1e4                                              # Datensimulationsanzahl
M   = c(3,3,4,9)                                       # Datendimension
K   = c(4,9,4,4)                                       # Gruppenanzahl
l   = 15                                               # Datenpunkte pro Gruppe
N   = l*K                                              # Gesamtanzahl Datenpunkte
nsc = length(M)                                        # Szenarienanzahl
TAU = matrix(rep(NaN,nsm*nsc), ncol = nsc)             # Statistik Array
NU  = matrix(rep(NaN,2*nsc)  , ncol = nsc)             # Parameter Array
WL  = seq(0,1,len = 1e3)                               # Wilk's Lambda Values
TL  = matrix(rep(NaN,length(WL)*nsc), nrow = nsc)      # \tau(\Lambda) Array
for(sc in 1:nsc){                                      # Szenarioiterationen

    # Modellparameter
    m       = M[sc]                                    # Datendimension
    k       = K[sc]                                    # Gruppenanzahl
    n       = N[sc]                                    # Gesamtanzahl Datenpunkte
    mu_i    = matrix(rep(1,m), nrow = m)               # Identische Gruppenerwartungswertparameter bei H_0
    Sigma   = diag(m)                                  # Identische Gruppenkovarianzmatrixparameter

    # Varianzanalyse Parameter
    w       = n-1-(1/2)*(m+k)                          # w
    t       = sqrt((m^2*(k-1)^2-4)/(m^2+(k-1)^2-5))    # t
    nu_1    = m*(k-1)                                  # \nu_1
    nu_2    = w*t-(1/2)*(m*(k-1)-2)                    # \nu_2
    TL[sc,] = ((1-WL^(1/t))/WL^(1/t))*(nu_2/nu_1)      # \tau(\Lambda)

    # Datensimulationen
    for(sm in 1:nsm){

        # Datengeneration
        Y     = array(dim = c(m,l,k))                  # Datenanarrayinitialisierung
        for(i in 1:k){                                 # Gruppeniterationen
            Y[,,i] =  t(mvrnorm(l,mu_i,Sigma))}        # Datensimulation

        # Varianzanalyse
        S          = sos(Y)                            # Stichprobenmittel und Sum of Squares Matrizen
        L          = det(S$W)/det(S$W + S$B)           # Wilks' Lambda
        tau        = ((1-L^(1/t))/L^(1/t))*(nu_2/nu_1) # Statistik
        TAU[sm,sc] = tau                               # Statistik
        NU[1,sc]   = nu_1                              # \nu_1
        NU[2,sc]   = nu_2}}                            # \nu_2
```

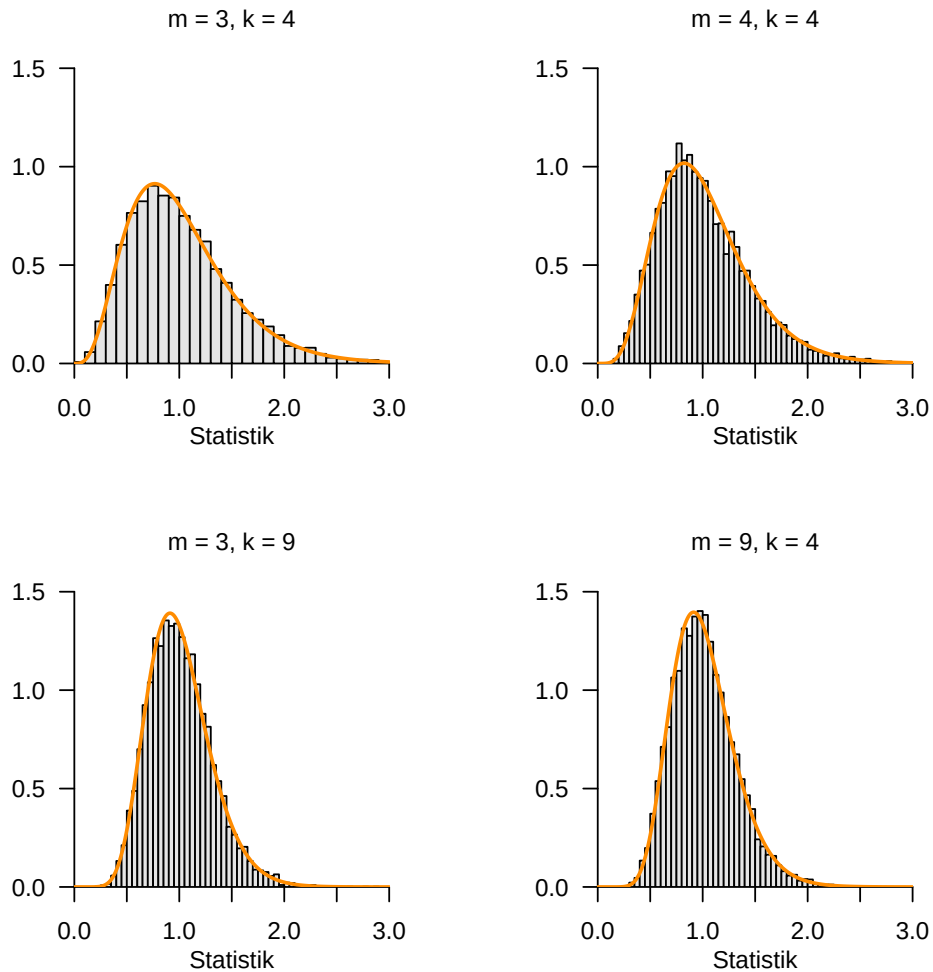
Modellevaluation mit der Wilks' Λ Statistik

τ als Funktion von Λ : $\Lambda \downarrow \Rightarrow \tau \uparrow$



Modellevaluation mit der Wilks' Λ Statistik

Simulation approximativer H_0 Verteilungen von Wilks' Λ Transformationen



Definition (Wilks' Λ-basierter Test, Testumfangkontrolle, p-Wert)

Es seien das Modell der einfaktoriellen Varianzanalyse und die Wilks' Λ basierte Teststatistik τ mit Verteilungsparametern ν_1, ν_2 wie oben definiert. Weiterhin sei der kritische Wert-basierte Test

$$\phi(Y) := 1_{\{\tau > k\}} := \begin{cases} 1 & \tau > k \\ 0 & \tau \leq k \end{cases} \tag{19}$$

definiert. ϕ ist genau dann ein Level-α_0-Test mit Testumfang α, wenn

$$k := k_{\alpha_0} := F^{-1}(1 - \alpha_0; \nu_1, \nu_2) \tag{20}$$

ist und der p-Wert einer realisiertern τ-Teststatistik $\tilde{\tau}$ ergibt sich zu

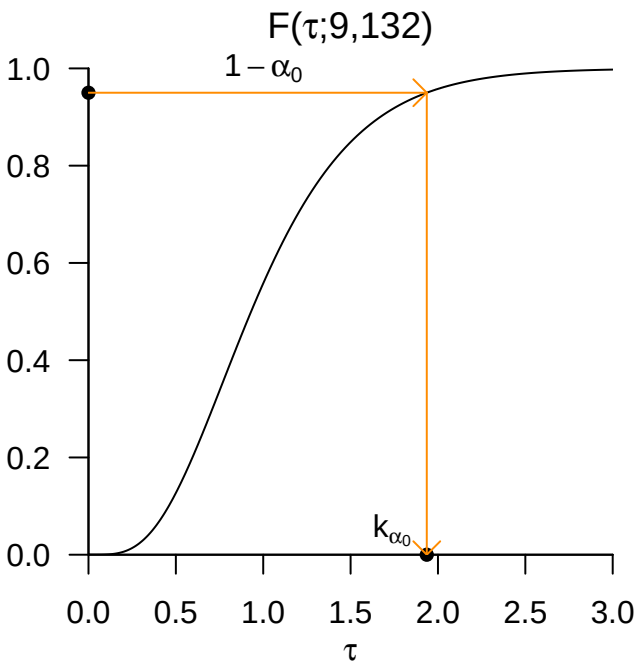
$$\text{p-Wert} = \mathbb{P}(\tau \geq \tilde{\tau}) = 1 - F(\tilde{\tau}; \nu_1, \nu_2) \tag{21}$$

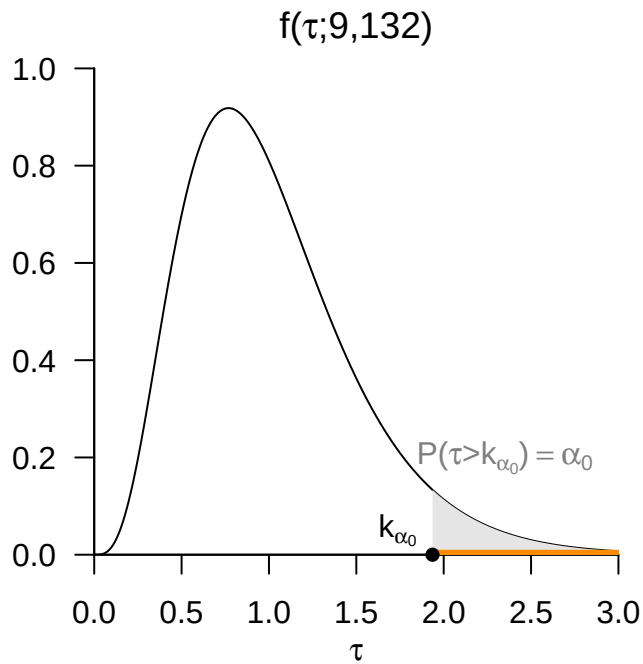
Bemerkungen

- Ein Beweis kann in Analogie zum Einstichproben-T^2-Test Fall geführt werden.
- Wir validieren die Testumfangkontrolle mithilfe von k_{α_0} in untenstehender Simulation.

Testumfangkontrolle

Wahl von k_{α_0} bei $m = 3, k = 4, n = 15 \Rightarrow \nu_1 = 9, \nu_2 = 132$ und $\alpha_0 = 0.05$.





Modellevaluation mit der Wilks' Λ Statistik

Testumfangkontrolle

```
# Szenarioparameter
nsm       = 1e4                                       # Datensimulationsanzahl
M         = c(3,3,4,9)                                # Datendimension
K         = c(4,9,4,4)                                # Gruppenanzahl
l         = 15                                        # Datenpunkte pro Gruppe
N         = l*K                                       # Gesamtanzahl Datenpunkte
alpha_0   = 0.05                                      # \alpha_0
nsc       = length(M)                                  # Szenarienanzahl
TAU       = matrix(rep(NaN,nsm*nsc), ncol = nsc)      # Statistik Array
NU        = matrix(rep(NaN,2*nsc)  , ncol = nsc)      # Parameter Array
KA        = rep(NaN, nsc)                             # Kritische Werte
PHI       = matrix(rep(0,nsm*nsc)  , ncol = nsc)      # Testarray

# Simulationen
for(sc in 1:nsc){                                     # Szenarioiterationen

    # Modellparameter
    m         = M[sc]                                 # Datendimension
    k         = K[sc]                                 # Gruppenanzahl
    n         = N[sc]                                 # Gesamtanzahl Datenpunkte
    mu_i      = matrix(rep(1,m), nrow = m)            # Identische Gruppenerwartungswertparameter bei H_0
    Sigma     = diag(m)                               # Identische Gruppenkovarianzmatrixparameter

    # Varianzanalyse Parameter
    w         = n-1-(1/2)*(m+k)                       # w
    t         = sqrt((m^2*(k-1)^2-4)/(m^2+(k-1)^2-5)) # t
    nu_1      = m*(k-1)                               # \nu_1
    nu_2      = w*t-(1/2)*(m*(k-1)-2)                 # \nu_2
    KA[sc]    = qf(1-alpha_0,nu_1,nu_2)               # kritischer Wert

    # Datensimulationen
    for(sm in 1:nsm){
        Y     = array(dim = c(m,l,k))                 # Datenanarrayinitialisierung
        for(i in 1:k){                                # Gruppeniterationen
            Y[,,i] =  t(mvrnorm(l,mu_i,Sigma))}       # Datensimulation
        S         = sos(Y)                            # Stichprobenmittel und Sum of Squares Matrizen
        L         = det(S$W)/det(S$W + S$B)           # Wilks' Lambda
        tau       = ((1-L^(1/t))/L^(1/t))*(nu_2/nu_1) # Statistik
        PHI[sm,sc] = tau > KA[sc]}}                   # Test
```

```
> Kritische Werte       :  1.95 1.55 1.82 1.57
> Geschätzte Testumfänge:  0.0526 0.0477 0.0513 0.0502
```

Praktisches Vorgehen

- Man nimmt an, dass ein vorliegender Datensatz von k Gruppendatensätzen $y_{11}, ..., y_{1l}$, $y_{21}, ..., y_{2l}$, $\cdots$, $y_{k1}, ..., y_{kl}$ Realisationen von $Y_{ij} \sim N(\mu_i, \Sigma)$ mit unbekannten Parametern $\mu_i \in \mathbb{R}^m, i = 1, ..., k$ und $\Sigma \in \mathbb{R}^{m \times m}$ p.d. sind.
- Man möchte entscheiden ob $H_0 : \mu_1 = \cdots = \mu_k$ eher zutrifft oder eher nicht.
- Man wählt ein Signifikanzniveau α_0 und bestimmt den zugehörigen Freiheitsgradparameter-abhängigen kritischen Wert k_{α_0}. Zum Beispiel gilt bei Wahl von $\alpha_0 := 0.05, m = 3, k = 4, l = 15$ und somit $n = 60$ sowie $\nu_1 = 9, \nu_2 = 132$, dass $k_{\alpha_0} = F^{-1}(1 - 0.05; 9, 132) \approx 1.95$ ist.
- Anhand der Wilk's Λ Statistik sowie m, k und n berechnet man man den realisierten Wert der τ-Teststatistik, den wir hier mit $\tilde{\tau}$ bezeichnen.
- Wenn $\tilde{\tau}$ größer als k_{α_0} ist, lehnt man die Nullhypothese ab, andernfalls nicht.
- Die oben entwickelte Theorie garantiert dann, dass man im Mittel in höchstens $\alpha_0 \cdot 100$ von 100 Fällen die Nullhypothese fälschlicherweise ablehnt.
- Schließlich ergibt sich der assoziierte p-Wert der realisiertern τ-Teststatistik $\tilde{\tau}$ zu

$$\text{p-Wert} = \mathbb{P}(\tau \geq \tilde{\tau}) = 1 - F(\tilde{\tau}; \nu_1, \nu_2) \tag{22}$$

Anwendungsbeispiel

Einfaktorielle Varianzanalyse mit R's `lm()` und `Manova()`

```
# Einlesen des Datensatzes aus Rudolf & Buse (2019) Multivariate Verfahren
library(foreign)
library(car)
D        = read.spss(file.path(getwd(), "11_Daten", "studienerfolg.sav"), to.data.frame = T)
model    = lm(cbind(D$X1,D$X2) ~ D$Gruppe, 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$Gruppe  2      0.61     5.76      4     82 0.00039 ***
> ---
> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Anwendungsbeispiel

Einfaktorielle Varianzanalyse mit sos()

```
# Einlesen des Datensatzes aus Rudolf & Buse (2019) Multivariate Verfahren
library("foreign")
D      = read.spss(file.path(getwd(), "11_Daten", "studienerfolg.sav"), to.data.frame = T)

# Datepräprozessierung
m      = 2                                          # Datendimension von Interesse
k      = 3                                          # Anzahl Gruppen
l      = 15                                         # Anzahl Datenpunkte pro Gruppe
n      = k*l                                        # Gesamtdatenpunktanzahl
Y      = array(dim = c(m,l,k))                      # Datenarrayinitialisierung
Y[,,1] = rbind(D$X1[D$Gruppe == "ungenügend"],      # Y_{1j_1} Intelligenztestscore
               D$X2[D$Gruppe == "ungenügend"])      # Y_{1j_2} Mathematiktestscore
Y[,,2] = rbind(D$X1[D$Gruppe == "befriedigend"],    # Y_{2j_1} Intelligenztestscore
               D$X2[D$Gruppe == "befriedigend"])    # Y_{2j_2} Mathematiktestscore
Y[,,3] = rbind(D$X1[D$Gruppe == "gut"],             # Y_{3j_1} Intelligenztestscore
               D$X2[D$Gruppe == "gut"])             # Y_{3j_2} Mathematiktestscore

# Einfaktorielle Varianzanalyse
S      = sos(Y)                                     # Sum of Squares Matrizen
L      = det(S$W)/det(S$W + S$B)                    # Wilks' Lambda
w      = n-1-(1/2)*(m+k)                            # w
t      = sqrt((m^2*(k-1)^2-4)/(m^2+(k-1)^2-5))      # t
nu_1   = m*(k-1)                                    # \nu_1
nu_2   = w*t-(1/2)*(m*(k-1)-2)                      # \nu_2
tau    = ((1-L^(1/t))/L^(1/t))*(nu_2/nu_1)          # Teststatistik
P      = 1-pf(tau,nu_1,nu_2)                        # Überschreitungswahrscheinlichkeit

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

```
> Wilks' Lambda      0.61
> tau                5.76
> nu_1               4
> nu_2               82
> P(tau > tau_tilde) 0.000392
```

Anwendungsszenario und Datendeskription

Modellformulierung und Modellschätzung

Modellevaluation

- Wilks' Λ
- **Pillai's** V

Selbstkontrollfragen

Definition (Pillai's V Statistik)

Es seien das Modell der einfaktoriellen Varianzanalyse sowie die Between Sum of Squares Matrix B und die Within Sum of Squares Matrix W definiert wie oben. Dann ist die *Pillai's V Statistik* definiert als

$$V := \text{tr}\left((B+W)^{-1}B\right) \tag{23}$$

wobei tr$(\cdot)$ die Spur, also die Summe der Diagonalelemente, einer Matrix bezeichnet.

Bemerkungen

- Die Pillai's V Statistik betrachtet die Diagonalelemente von $T^{-1}B$
- Ein hoher Wert von V spricht also für einen großen Anteil der Between-Varianz an der Gesamtvarianz.
- Für einen hohen Wert von V würde man also die Nullhypothese $B = 0_{mm}$ verwerfen.

Theorem (Eigenwertform der Pillai's V Statistik)

Es seien das Modell der einfaktoriellen Varianzanalyse, die Between Sum of Squares Matrix B, die Within Sum of Squares Matrix W und die Pillai's V Statistik definiert wie oben. Weiterhin seien $\lambda_1, ..., \lambda_s$ die Eigenwerte von $W^{-1}B$. Dann gilt

$$V = \sum_{i=1}^{s} \frac{\lambda_i}{1 + \lambda_i} \tag{24}$$

footnotesize Bemerkungen

- Wir verzichten auf einen Beweis.
- Die Matrix $W^{-1}B$ ist das multivariate Analogon zu $\frac{\text{SQB}}{\text{SQW}}$

Theorem (Approximative H_0 Verteilungen von Pillai's V Transformationen)

Es seien das Modell der einfaktoriellen Varianzanalyse und Pillai's V definiert wie oben und es gelte außerdem die Nullhypothese

$$H_0 : \mu_1 = \mu_2 = \cdots = \mu_p \text{ bzw. } H_0 : \alpha_1 = \alpha_2 = \cdots = \alpha_p = 0_m. \tag{25}$$

Weiterhin seien die Parameter

$$s := \min(k-1, m), t := \frac{1}{2}(|k-1-m| - 1) \text{ und } w := \frac{1}{2}(n-k-m-1) \tag{26}$$

definiert. Dann ist

$$\tau = \frac{(2w+s+1)V}{(2t+s+1)(s-V)} \tag{27}$$

approximativ f-verteilt mit Freiheitsgradparametern

$$\nu_1 := s(2t+s+1) \text{ und } \nu_2 := s(2w+s+1) \tag{28}$$

Bemerkungen

- Die Approximation geht zurück auf Pillai (1955).

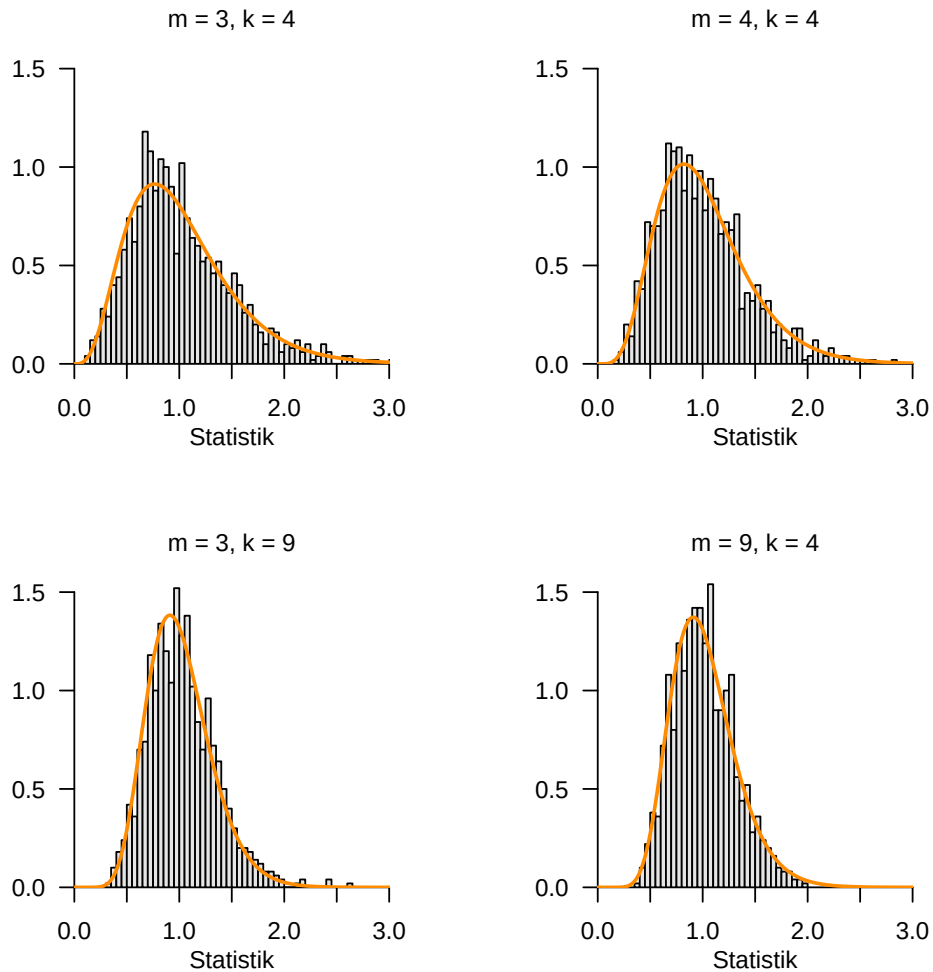
Modellevaluation mit der Pillai's V Statistik

Simulation approximativer H_0 Verteilungen von Pillai's V Transformatione

```
# Szenarioparameter
library(MASS)                                          # R Paket für multivariate Normalverteilungen
library(matlib)                                        # R Paket für Matrixalgebra
nsm = 1e1                                              # Datensimulationsanzahl
M   = c(3,3,4,9)                                       # Datendimension
K   = c(4,9,4,4)                                       # Gruppenanzahl
l   = 15                                               # Datenpunkte pro Gruppe
N   = l*K                                              # Gesamtanzahl Datenpunkte
nsc = length(M)                                        # Szenarienanzahl
TAU = matrix(rep(NaN,nsm*nsc), ncol = nsc)             # Statistik Array
NU  = matrix(rep(NaN,2*nsc)  , ncol = nsc)             # Parameter Array
for(sc in 1:nsc){                                      # Szenarioiterationen


    # Modellparameter
    m     = M[sc]                                      # Datendimension
    k     = K[sc]                                      # Gruppenanzahl
    n     = N[sc]                                      # Gesamtanzahl Datenpunkte
    mu_i  = matrix(rep(1,m), nrow = m)                 # Identische Gruppenerwartungswertparameter bei H_0
    Sigma = diag(m)                                    # Identische Gruppenkovarianzmatrixparameter

    # Varianzanalyseparameter
    s     = min(k-1,m)                                 # s
    t     = (1/2)*(abs(k-1-m)-1)                       # t
    w     = (1/2)*(n-k-m-1)                            # w
    nu_1  = s*(2*t+s+1)                                # \nu_1
    nu_2  = s*(2*w+s+1)                                # \nu_2

    # Datensimulationen
    for(sm in 1:nsm){

        # Datengeneration
        Y     = array(dim = c(m,l,k))                  # Datenanarrayinitialisierung
        for(i in 1:k){                                 # Gruppeniterationen
            Y[,,i] =  t(mvrnorm(l,mu_i,Sigma))}        # Datensimulation

        # Varianzanalyse
        S          = sos(Y)                            # Stichprobenmittel und Sum of Squares Matrizen
        V          = sum(diag(inv(S$B+S$W) %*% S$B))   # Pillai's V
        tau        = ((2*w+s+1)*V)/((2*t+s+1)*(s-V))   # Statistik
        TAU[sm,sc] = tau                               # Statistik
        NU[1,sc]   = nu_1                              # \nu_1
        NU[2,sc]   = nu_2}}                            # \nu_2
```

Modellevaluation mit der Pillai's V Statistik

Simulation approximativer H_0 Verteilungen von Pillai's V Transformationen



Definition (Pillai's V -basierter Test, Testumfangkontrolle, p-Wert)

Es seien das Modell der einfaktoriellen Varianzanalyse und die Pillai's V basierte Teststatistik τ mit Verteilungsparametern ν_1, ν_2 wie oben definiert. Weiterhin sei der kritische Wert-basierte Test

$$\phi(Y) := 1_{\{\tau > k\}} := \begin{cases} 1 & \tau > k \\ 0 & \tau \leq k \end{cases} \tag{29}$$

definiert. ϕ ist genau dann ein Level-α_0-Test mit Testumfang α, wenn

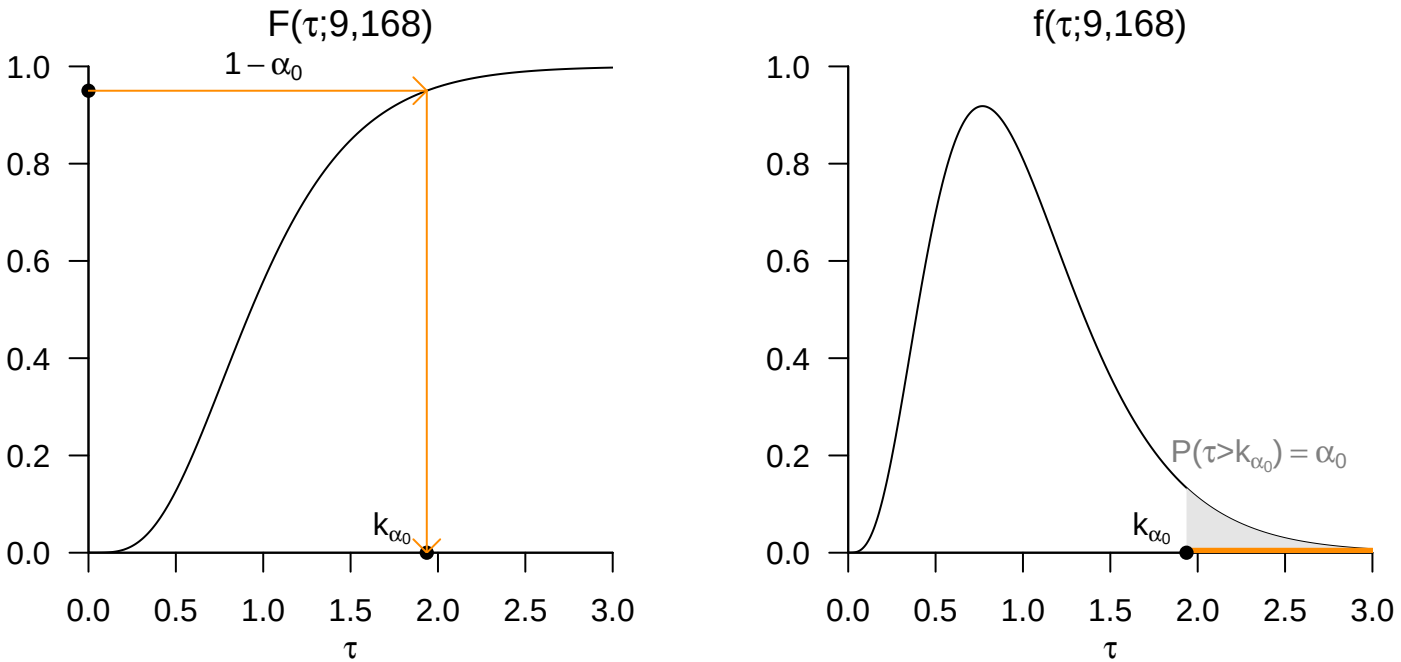
$$k := k_{\alpha_0} := F^{-1}(1 - \alpha_0; \nu_1, \nu_2) \tag{30}$$

ist und der p-Wert einer realisiertern τ-Teststatistik $\tilde{\tau}$ ergibt sich zu

$$\text{p-Wert} = \mathbb{P}(\tau \geq \tilde{\tau}) = 1 - F(\tilde{\tau}; \nu_1, \nu_2) \tag{31}$$

Testumfangkontrolle

Wahl von k_{α_0} bei $m = 3, k = 4, n = 15 \Rightarrow \nu_1 = 9, \nu_2 = 168$ und $\alpha_0 = 0.05$.



Modellevaluation mit der Pillai's V Statistik

Testumfangkontrolle

```
# Szenarioparameter
library(MASS)                                          # R Paket für multivariate Normalverteilungen
library(matlib)                                        # R Paket für Matrixalgebra
nsm     = 1e3                                          # Datensimulationsanzahl
M       = c(3,3,4,9)                                   # Datendimension
K       = c(4,9,4,4)                                   # Gruppenanzahl
l       = 15                                           # Datenpunkte pro Gruppe
N       = l*K                                          # Gesamtanzahl Datenpunkte
alpha_0 = 0.05                                         # Signifikanzlevel
nsc     = length(M)                                    # Szenarienanzahl
KA      = rep(NaN, nsc)                                # kritische Werte
PHI     = matrix(rep(0,nsm*nsc), ncol = nsc)           # Testarray

# Szenariosimulationen
for(sc in 1:nsc){                                      # Szenarioiterationen

    # Modell- und Varianzanalyseparameter
    m      = M[sc]                                     # Datendimension
    k      = K[sc]                                     # Gruppenanzahl
    n      = N[sc]                                     # Gesamtanzahl Datenpunkte
    mu_i   = matrix(rep(1,m), nrow = m)                # Identische Gruppenerwartungswertparameter bei H_0
    Sigma  = diag(m)                                   # Identische Gruppenkovarianzmatrixparameter
    s      = min(k-1,m)                                # s
    t      = (1/2)*(abs(k-1-m)-1)                      # t
    w      = (1/2)*(n-k-m-1)                           # w
    nu_1   = s*(2*t+s+1)                               # \nu_1
    nu_2   = s*(2*w+s+1)                               # \nu_2
    KA[sc] = qf(1-alpha_0,nu_1,nu_2)                   # kritischer Wert

    # Datensimulationen
    for(sm in 1:nsm){

        # Datengeneration und Varianzanalyse
        Y     = array(dim = c(m,l,k))                  # Datenanarrayinitialisierung
        for(i in 1:k){                                 # Gruppeniterationen
            Y[,,i] =  t(mvrnorm(l,mu_i,Sigma))}        # Datensimulation
        S          = sos(Y)                            # Stichprobenmittel und Sum of Squares Matrizen
        V          = sum(diag(inv(S$B+S$W) %*% S$B))   # Pillai's V
        tau        = ((2*w+s+1)*V)/((2*t+s+1)*(s-V))   # Statistik
        PHI[sm,sc] = tau > KA[sc]}}                    # Test
```

```
> Kritische Werte       :  1.95 1.55 1.82 1.57
> Geschätzte Testumfänge:  0.0526 0.0477 0.0513 0.0502
```

Praktisches Vorgehen

- Man nimmt an, dass ein vorliegender Datensatz von k Gruppendatensätzen $y_{11}, ..., y_{1l}$, $y_{21}, ..., y_{2l}$, $\cdots$, $y_{k1}, ..., y_{kl}$ Realisationen von $Y_{ij} \sim N(\mu_i, \Sigma)$ mit unbekannten Parametern $\mu_i \in \mathbb{R}^m, i = 1, ..., k$ und $\Sigma \in \mathbb{R}^{m \times m}$ p.d. sind.
- Man möchte entscheiden ob $H_0 : \mu_1 = \cdots = \mu_k$ eher zutrifft oder eher nicht.
- Man wählt ein Signifikanzniveau α_0 und bestimmt den zugehörigen Freiheitsgradparameter-abhängigen kritischen Wert k_{α_0}. Zum Beispiel gilt bei Wahl von $\alpha_0 := 0.05, m = 3, k = 4, l = 15$ und somit $n = 60$ sowie $\nu_1 = 9, \nu_2 = 168$, dass $k_{\alpha_0} = F^{-1}(1 - 0.05; 9, 168) \approx 1.94$ ist.
- Anhand der Pillai's V Statistik sowie m, k und n berechnet man man den realisierten Wert der τ-Teststatistik, den wir hier mit $\tilde{\tau}$ bezeichnen.
- Wenn $\tilde{\tau}$ größer als k_{α_0} ist, lehnt man die Nullhypothese ab, andernfalls nicht.
- Die oben entwickelte Theorie garantiert dann, dass man im Mittel in höchstens $\alpha_0 \cdot 100$ von 100 Fällen die Nullhypothese fälschlicherweise ablehnt.
- Schließlich ergibt sich der assoziierte p-Wert der realisiertern τ-Teststatistik $\tilde{\tau}$ zu

$$\text{p-Wert} = \mathbb{P}(\tau \geq \tilde{\tau}) = 1 - F(\tilde{\tau}; \nu_1, \nu_2) \tag{32}$$

Anwendungsbeispiel

Einfaktorielle Varianzanalyse mit R's `lm()` und `Manova()`

```
# Einlesen des Datensatzes aus Rudolf & Buse (2019) Multivariate Verfahren
library(foreign)
library(car)
D        = read.spss(file.path(getwd(), "11_Daten", "studienerfolg.sav"), to.data.frame = T)
model    = lm(cbind(D$X1,D$X2) ~ D$Gruppe, D)      # Modellspezifikation
Manova(model, test.statistic = "Pillai")           # Einfaktorielle Varianzanalyse
```

```
>
> Type II MANOVA Tests: Pillai test statistic
>          Df test stat approx F num Df den Df  Pr(>F)
> D$Gruppe  2     0.404     5.31      4     84 0.00073 ***
> ---
> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Modellevaluation mit der Pillai's V Statistik

Anwendungsbeispiel

Einfaktorielle Varianzanalyse mit sos()

```
# Einlesen des Datensatzes aus Rudolf & Buse (2019) Multivariate Verfahren
library(foreign)
library(matlib)
D      = read.spss(file.path(getwd(), "11_Daten", "studienerfolg.sav"), to.data.frame = T)

# Datepräprozessierung
m      = 2                                          # Datendimension von Interesse
k      = 3                                          # Anzahl Gruppen
l      = 15                                         # Anzahl Datenpunkte pro Gruppe
n      = k*l                                        # Gesamtdatenpunktanzahl
Y      = array(dim = c(m,l,k))                      # Datenarrayinitialisierung
Y[,,1] = rbind(D$X1[D$Gruppe == "ungenügend"],      # Y_{1j_1} Intelligenztestscore
               D$X2[D$Gruppe == "ungenügend"])      # Y_{1j_2} Mathematiktestscore
Y[,,2] = rbind(D$X1[D$Gruppe == "befriedigend"],    # Y_{2j_1} Intelligenztestscore
               D$X2[D$Gruppe == "befriedigend"])    # Y_{2j_2} Mathematiktestscore
Y[,,3] = rbind(D$X1[D$Gruppe == "gut"],             # Y_{3j_1} Intelligenztestscore
               D$X2[D$Gruppe == "gut"])             # Y_{3j_2} Mathematiktestscore

# Einfaktorielle Varianzanalyse
S      = sos(Y)                                     # Sum of Squares Matrizen
V      = sum(diag(inv(S$B+S$W) %*% S$B))            # Pillai's V
s      = min(k-1,m)                                 # s
t      = (1/2)*(abs(k-1-m)-1)                       # t
w      = (1/2)*(n-k-m-1)                            # w
nu_1   = s*(2*t+s+1)                                # \nu_1
nu_2   = s*(2*w+s+1)                                # \nu_2
tau    = ((2*w+s+1)*V)/((2*t+s+1)*(s-V))            # Statistik
P      = 1-pf(tau,nu_1,nu_2)                        # Überschreitungswahrscheinlichkeit

# Ausgabe
cat("Pillai's  V        "  , V,
    "\ntau                ", tau,
    "\nnu_1               ", nu_1,
    "\nnu_2               ", nu_2,
    "\nP(tau > tau_tilde) ", P)
```

```
> Pillai's  V         0.404
> tau                 5.31
> nu_1                4
> nu_2                84
> P(tau > tau_tilde)  0.000732
```

Anwendungsszenario und Datendeskription

Modellformulierung und Modellschätzung

Modellevaluation

- Wilks' Λ
- Pillai's V

Selbstkontrollfragen

Selbskontrolfragen

1. Erläutern Sie das Anwendungsszenario einer multivariaten einfaktoriellen Varianzanalyse.
2. Definieren Sie das Klassische Modell der einfaktoriellen Varianzanalyse.
3. Geben Sie das Theorem zur Kreuzproduktsummenmatrixzerlegung wieder.
4. Erläutern Sie die intuitive Bedeutung der T, B und W Matrizen der Kreuzproduktsummenmatrixzerlegung.
5. Geben Sie einen Überblick zu den Teststatistiken der einfaktoriellen Varianzanalyse und ihren Verteilungen.
6. Definieren Sie die Wilk's Λ Teststatistik.
7. Erläutern Sie die intuitive Bedeutung der Wilk's Λ Teststatistik.
8. Definieren Sie die Pillai's V Teststatistik.
9. Erläutern Sie die intuitive Bedeutung der Pillai's V Teststatistik

References

Anderson, T. W. 2003. *An Introduction to Multivariate Statistical Analysis*. 3rd ed. Wiley Series in Probability and Statistics. Hoboken, N.J: Wiley-Interscience.

Pillai, K. C. S. 1955. "Some New Test Criteria in Multivariate Analysis." *The Annals of Mathematical Statistics* 26 (1): 117–21. https://doi.org/10.1214/aoms/1177728599.

Rao, C Radhakrishna. 1951. "An Asymptotic Expansion of the Distribution of Wilk's Criterion." *Bulletin of the International Statistical Institute* 33 (2): 177–80.

———. 1972. "Recent Trends of Research Work in Multivariate Analysis." *Biometrics* 28 (1): 21.

Rudolf, Matthias, and Johannes Buse. 2020. *Multivariate Verfahren*. Göttingen: Hogrefe.

Wilks, S. S. 1932. "Certain Generalizations in the Analysis of Variance." *Biometrika* 24 (3-4): 471–94. https://doi.org/10.1093/biomet/24.3-4.471.