TUTORIUM ZUR HAUPTKOMPONENTENANALYSE UNTER ANWENDUNG VON R

HENK VAN ELST*
parcIT GmbH, Erftstraße 15, 50672 Köln, Germany

21. November 2020

1 Einführung

Dieses Tutorium demonstriert die wesentlichen theoretischen und praktischen Schritte der **Hauptkomponentenanalyse** eines multivariaten, metrisch skalierten Datenssatzes, und einer aus dieser gegebenenfalls abgeleiteten **Dimensionsreduktion** des betrachteten Datensatzes. Das Tutorium basiert auf einer transparenten Darstellung aller relevanten Berechnungen mithilfe eines **R-Skripts**; vgl. R Core Team (2020) [15].

Die Grundüberlegungen zur **Hauptkomponentenanalyse** und der **Dimensionsreduktion** wurden im Wesentlichen von Pearson (1901) [14], Hotelling (1933) [8] und Kaiser (1960) [12] angestellt; siehe auch Hatzinger *et al* (2014) [7], Hair *et al* (2010) [6] oder Jolliffe (2002) [11].

Die nachfolgenden Ausführungen gliedern sich in drei Teile. Zunächst wird ein **trivariater Beispieldatensatz** mit **Messwerten** zu drei metrisch skalierten **Variablen** geladen und mit Standardmethoden der **Beschreibenden Statistik** quantitativ charakterisiert und visualisiert (Abschnitte 2 bis 5). Dann wird die in der **Linearen Algebra** angesiedelte Methodik einer **Hauptkomponentenanalyse** bereitgestellt. Es handelt sich hierbei um die **Eigenwertanalyse** symmetrischer quadratischer Matrizen und deren Diagonalisierung durch bestimmte, aus den **Eigenvektoren** konstruierten **Rotationstransformationen** (Abschnitte 6 bis 10). (In der Analytischen Geometrie ist die hier angewendte Methodik als Hauptachsentransformation bekannt; vgl. Bronstein *et al* (2005) [2].) Schließlich wird das auf der **Eigenwertanalyse** der **Korrelationsmatrix** aufbauende Verfahren der **Dimensionsreduktion** eines multivariaten Datensatzes dargestellt, hier für den Fall des **trivariaten Beispieldatensatzes** (Abschnitt 11).

Die nachfolgenden Ergebnisse werden erstellt mit R Version 4.0.2.

2 Laden der benötigten R-Pakete

Für die Durchführung der Berechnungen und die Erstellung der Grafiken in diesem Tutorium werden die folgenden R-Pakete geladen:

```
library(tidyverse)

## - Attaching packages --------------------------------- tidyverse
1.3.0 -
## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.3     v dplyr   1.0.0
```

*ePost: Henk.van.Elst@parcIT.de

```
## v tidyr   1.1.0     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0
## - Conflicts ----------------------------------
tidyverse_conflicts() -
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(zinsszenarien)
library(plotly)
```

```
##
## Attaching package: 'plotly'
## The following object is masked from 'package:ggplot2':
##
##     last_plot
## The following object is masked from 'package:stats':
##
##     filter
## The following object is masked from 'package:graphics':
##
##     layout
```

```
library(psych)
```

```
##
## Attaching package: 'psych'
## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha
```

```
library(REdaS)
```

```
## Loading required package: grid
```

```
library(GGally)
```

```
## Warning: package 'GGally' was built under R version 4.0.3
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

3 Laden des trivariaten Beispieldatensatzes (X-Matrix)

Der in diesem Tutorium als konkretes Beispiel herangezogene **trivariate Datensatz** umfasst **Messwerte** zu den drei metrisch skalierten **Variablen** Körpergröße [cm], Masse [kg] und Alter [yr] für eine Stichprobe von 187 volljährigen Frauen umfasst. Diese Information befindet sich in einem größeren Datensatz nach Howell (2001) [9], der nun geladen wird.[1]

```
load("testData.RData")
str(object = testData)
```

[1] Der vollständige Originaldatensatz ist unter der URL tspace.library.utoronto.ca/handle/1807/10395 verfügbar.

```
## 'data.frame': 544 obs. of  4 variables:
##  $ height: num  152 140 137 157 145 ...
##  $ weight: num  47.8 36.5 31.9 53 41.3 ...
##  $ age   : num  63 63 65 41 51 35 32 27 19 54 ...
##  $ male  : int  1 0 0 1 0 1 0 1 0 1 ...
```

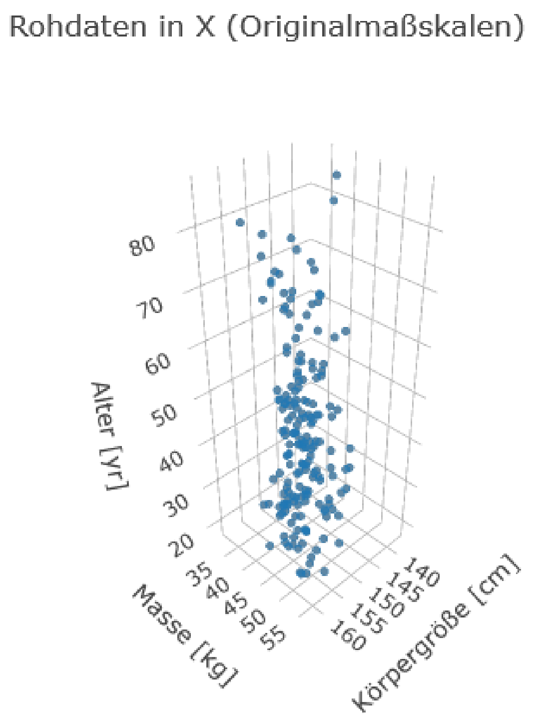
Es folgt das Ausfiltern des eigentlichen **trivariaten Datensatzes**, den **Messwerten** zu den drei **Variablen** Körpergröße [cm], Masse [kg] und Alter [yr] für eine Stichprobe von Frauen ab 18 Jahren.

```
X <- testData %>% filter(.data = ., age >= 18 & male == 0)
colnames(X) <- c("height [cm]", "mass [kg]", "age [yr]", "male")
str(object = X)

## 'data.frame': 187 obs. of  4 variables:
##  $ height [cm]: num  140 137 145 149 148 ...
##  $ mass [kg]  : num  36.5 31.9 41.3 38.2 34.9 ...
##  $ age [yr]   : num  63 65 51 32 19 47 73 20 65.3 31 ...
##  $ male       : int  0 0 0 0 0 0 0 0 0 0 ...
```

Die **Datenmatrix** $\boldsymbol{X}$ ist die Rohdatenmatrix der in diesem Tutorium beschriebenen theoretischen und praktischen Betrachtung.

3.1 Visualisieren der Daten in $\boldsymbol{X}$ per 3D-Streudiagramm

Der **trivariate Datensatz** in $\boldsymbol{X}$ wird zunächst über ein **3D-Streudiagramm** visualisiert. Dies wird ermöglicht durch die Funktion `plot_ly()` aus dem Paket `plotly`.[2]

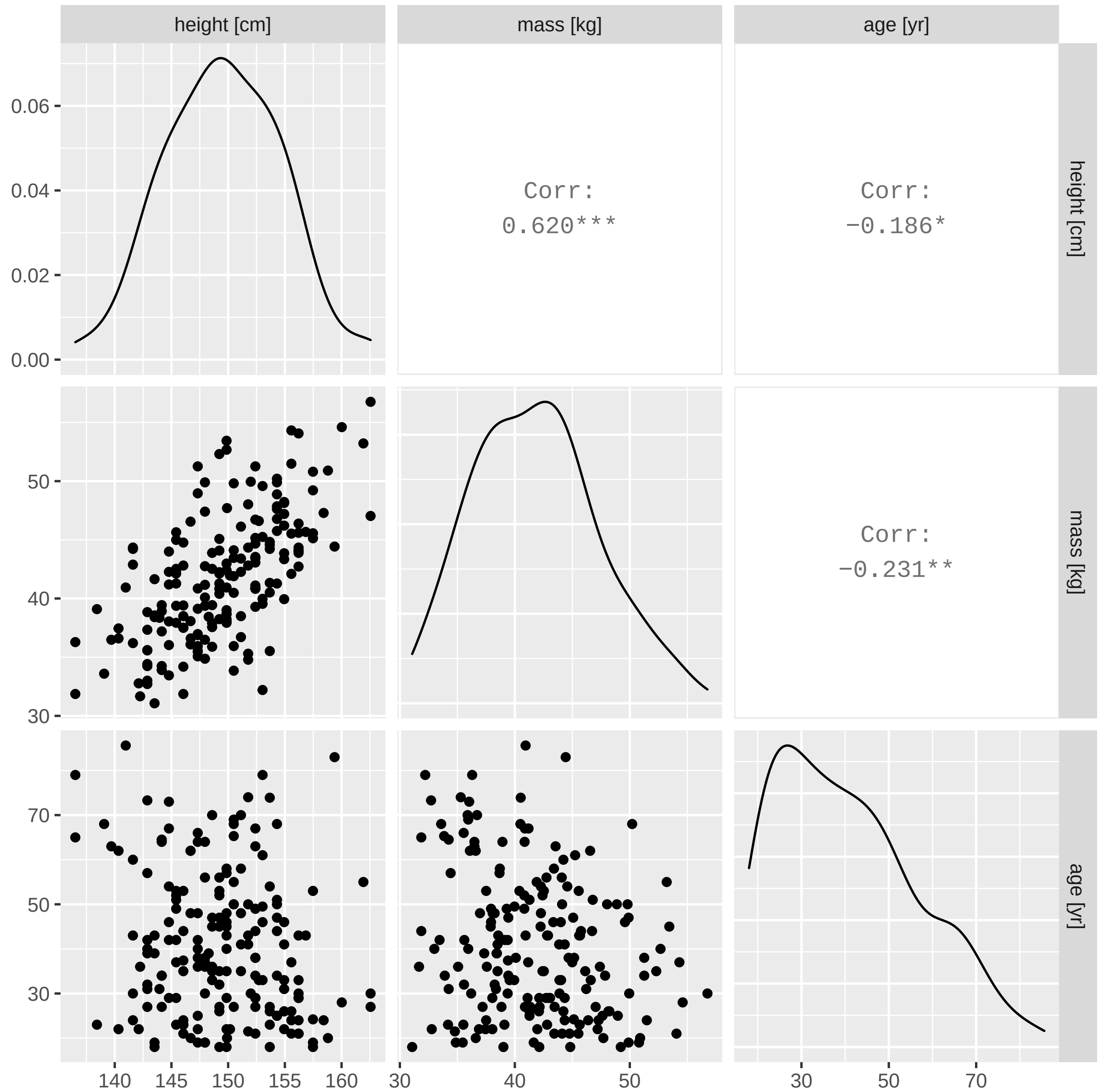


[2]Zu beachten ist hier die Tatsache einer nicht der mathematischen Konvention entsprechenden Orientierung der Maßskala entlang der "x"-Achse; das resultierende Koordinatensystem ist *nicht* rechtshändisch orientiert.

3.2 Visualisieren der Daten in X per Streudiagrammmatrix

Der **trivariate Datensatz** in X wird mithilfe einer **Streudiagrammmatrix** visualisiert. Die einzelnen **Streudiagramme** entsprechen Projektionen der Punktewolke im **3D-Streudiagramm** auf horizontal und vertikal orientierte 2D Schnittebenen. Hierfür wird die Funktion `ggpairs()` aus dem Paket `GGally` verwendet.

```
ggpairs(data = X[, 1:3])
```



Wie die Diagramme auf der Diagonalen der **Streudiagrammmatrix** qualitativ suggerieren, erscheinen die Messwerte zu den **Variablen** Körpergröße [cm] und Masse [kg] in der Stichprobe **näherungsweise normalverteilt** zu sein, jene zu der **Variablen** Alter [yr] hingegen nicht.

3.3 Beschreibende Statistik und Ausreißerbetrachtung für die Daten in X

Für die **Messwerte** zu jeder der drei **Variablen** in X werden die folgenden beschreibenden statistischen Kennzahlen berechnet:

1. **Mittelwert** und **Standardabweichung**:

```
apply(X = X[, 1:3], MARGIN = 2, FUN = mean)

## height [cm]   mass [kg]    age [yr]
##   149.51352    41.81419    40.71230

apply(X = X[, 1:3], MARGIN = 2, FUN = sd)

## height [cm]   mass [kg]    age [yr]
##    5.084577    5.387917   16.219897
```

2. Standardisierte **Schiefe-** und **Wölbungsmaße**; vgl. Joanes und Gill (1998) [10] und van Elst (2019) [4]:

```
zinsszenarien::stand.schiefe(X[, 1:3])

## height [cm]   mass [kg]    age [yr]
##   0.0205191   1.7939789   3.3003240

zinsszenarien::stand.woelbung(X[, 1:3])

## height [cm]   mass [kg]    age [yr]
##  -0.6688236  -0.9719034  -1.3598265
```

Sofern sowohl das standardisierte Schiefe- als auch das standardisierte Wölbungsmaß einen Wert innerhalb eines Intervalls mit den Grenzen $Q_{0,025} = -1,96$ und $Q_{0,975} = +1,96$ gemäß dem zentralen 95 %-Wahrscheinlichkeitsbereich für eine standardnormalverteilte Größe annehmen, kann entsprechend einer in der Statistik gepflegten Konvention von **näherungsweise normalverteilten** univariaten Daten ausgegangen werden; vgl. Hair *et al* (2010) [6]. Wie die erhaltenen Ergebnisse zeigen, trifft das im betrachteten **trivariaten Datensatz** für die Variablen Körpergröße [cm] und Masse [kg] zu, nicht aber für die Variable Alter [yr], deren beobachtete Verteilung als rechtsschief zu klassifizieren ist; vgl. die Streudiagrammmatrix oben.

3. Anzahlen von **Ausreißern**, **Extremwerten** und **6-sigma-Ereignissen**; vgl. Toutenburg (2004) [16]:

```
zinsszenarien::ausreisser(X[, 1:3])

## height [cm]   mass [kg]    age [yr]
##           0           1           0

zinsszenarien::extremwerte(X[, 1:3])

## height [cm]   mass [kg]    age [yr]
##           0           0           0

zinsszenarien::sechs_sigma_ereignisse(X[, 1:3])

## height [cm]   mass [kg]    age [yr]
##           0           0           0
```

4 Standardisieren des trivariaten Datensatzes (Z-Matrix)

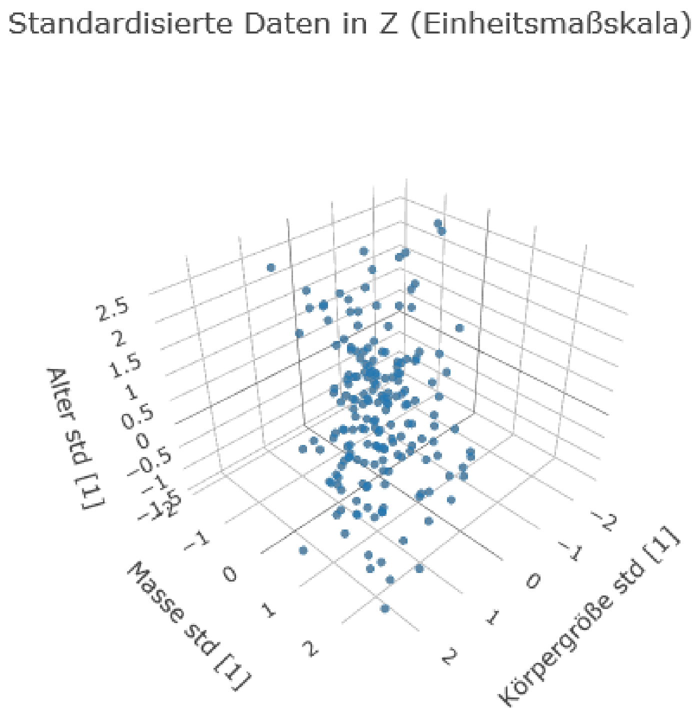
Es folgt die **Transformation** der Rohdaten in $\boldsymbol{X}$ auf eine gemeinsame **dimensionslose Maßskala**, auf welcher Messwerte als **Abweichungen vom Mittelwert in Vielfachen der Standardabweichung** kodiert werden.

```
Z <- scale(x = X[, c("height [cm]", "mass [kg]", "age [yr]")], center = TRUE, sc
colnames(Z) <- c("height_std [1]", "mass_std [1]", "age_std [1]")
dim(Z)

## [1] 187   3

head(x = Z)

##      height_std [1] mass_std [1] age_std [1]
## [1,]     -1.9300562   -0.9889505   1.3740963
## [2,]     -2.5544936   -1.8466045   1.4974016
## [3,]     -0.8060688   -0.0997265   0.6342642
## [4,]     -0.0567439   -0.6627263  -0.5371365
## [5,]     -0.3065189   -1.2888663  -1.3386213
## [6,]      0.9423560    1.4998244   0.3876535
```

Als Folge des **Standardisierens** haben die univariaten Daten zu jeder der drei **Variablen** in der **Datenmatrix** $\boldsymbol{Z}$ einen **Mittelwert** von 0 und eine **Standardabweichung** von 1. Dieser Sachverhalt wird in Kürze explizit betrachtet werden.

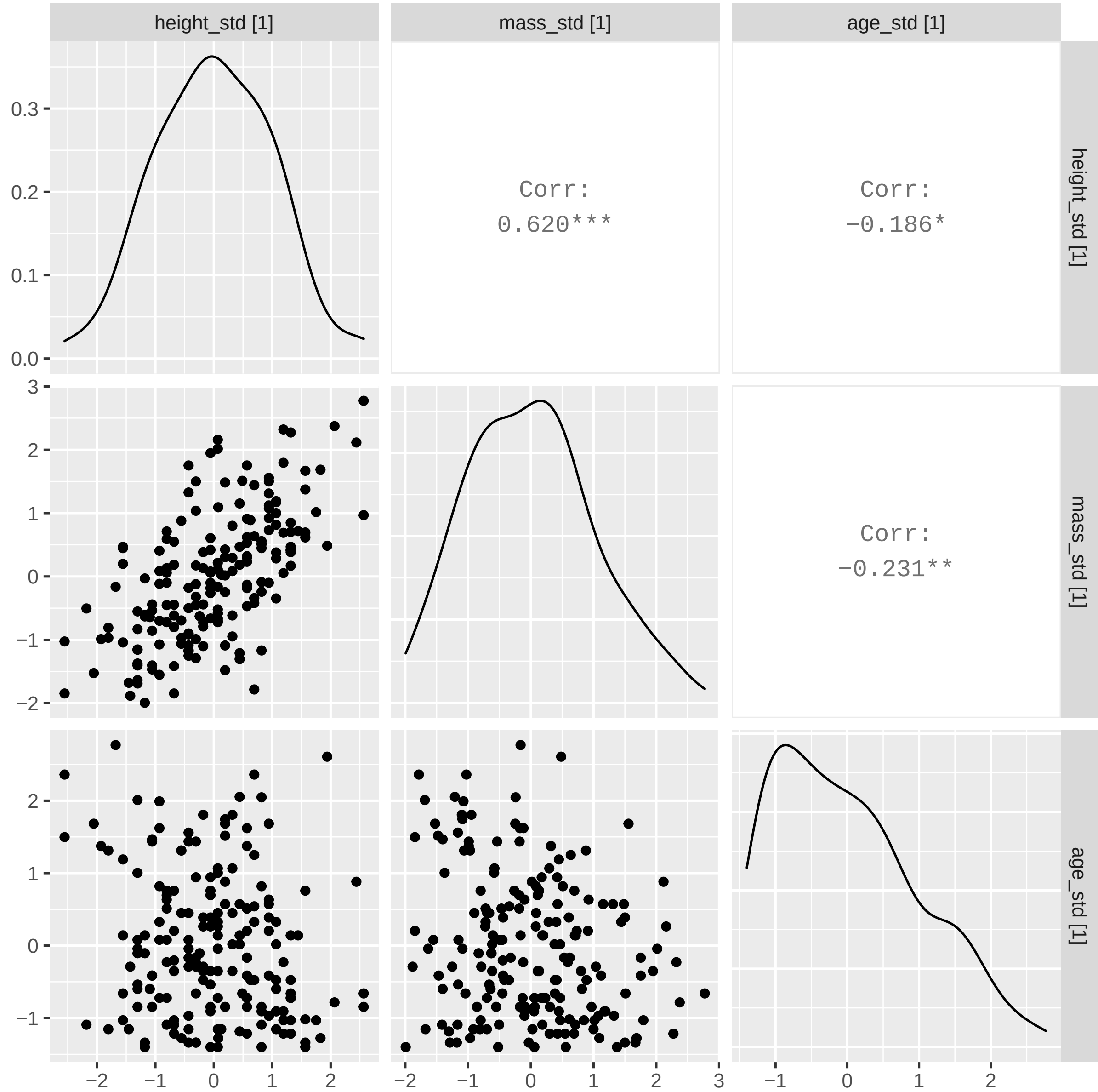
Die **Datenmatrix** $\boldsymbol{Z}$ der **standardisierten Messwerte** ("z-Werte") bildet die Grundlage der nachfolgenden Analyseschritte.

4.1 Visualisieren der Daten in $\boldsymbol{Z}$ per 3D-Streudiagramm



4.2 Visualisieren der Daten in Z per Streudiagrammmatrix

```
ggpairs(data = as.data.frame(Z))
```



Wie diese **Streudiagrammmatrix** im Vergleich zu jener oben für die ursprünglichen **Messwerte** qualitativ zeigt, sind durch das **Standardisieren** deren zentrale uni- und bivariate (und trivariate) Verteilungseigenschaften unverändert geblieben. (Andernfalls würde es sich um einen illegitimen Analyseschritt handeln.) Diesen Sachverhalt belegt auch der nächste Analyseschritt.

4.3 Beschreibende Statistik für die Daten in Z

Erneut werden beschreibende statistische Kennzahlen berechnet, nun für die Daten in Z ("z-Werte"):

1. **Mittelwert** und Standardabweichung:

```
apply(X = Z, MARGIN = 2, FUN = mean) %>% round(x = ., digits = 4)

## height_std [1]   mass_std [1]    age_std [1]
##              0              0              0
```

```
apply(X = Z, MARGIN = 2, FUN = sd)

## height_std [1]   mass_std [1]    age_std [1]
##              1              1              1
```

2. Standardisierte **Schiefe-** und **Wölbungsmaße**; vgl. Joanes und Gill (1998) [10] und van Elst (2019) [4]:

```
zinsszenarien::stand.schiefe(Z)

## height_std [1]   mass_std [1]    age_std [1]
##      0.0205191      1.7939789      3.3003240

zinsszenarien::stand.woelbung(Z)

## height_std [1]   mass_std [1]    age_std [1]
##     -0.6688236     -0.9719034     -1.3598265
```

5 Untersuchen der Eignung des trivariaten Datensatzes in $\boldsymbol{Z}$ für eine Hauptkomponentenanalyse

Die Eignung des **trivariaten Datensatzes** für eine **Hauptkomponentenanalyse** wird mit Bartletts (1951) [1] **Test auf Sphärizität** sowie mit den standardisierten **KMO-** und **MSA-Maßen** nach Kaiser, Meyer und Olkin (KMO) untersucht; vgl. Kaiser (1970) [12], Guttman (1953) [5] und Hatzinger *et al* (2014) [7]. Hierfür werden die Funktionen `bart_spher()` und `KMO()` aus dem Paket `REdaS` verwendet.

Bartletts (1951) [1] frequentistischer **Nullhypothesentest** unterzieht die Grundannahme der Sphärizität der **Einhüllenden** der durch die Daten in $\boldsymbol{Z}$ gegebenen **Punktewolke** im Euklidischen Raum $\mathbb{R}^3$ einer empirischen Überprüfung. Im vorliegenden Fall liefert dies das folgende Resultat:

```
bart_spher(x = Z)

##  Bartlett's Test of Sphericity
##
## Call: bart_spher(x = Z)
##
##      X2 = 100.12
##      df = 3
## p-value < 2.22e-16
```

Entsprechend dem erhaltenen p-Wert kann die Nullhypothese zu einem Signifikanzniveau von $\alpha = 0,01$ verworfen werden. Mit großer Wahrscheinlichkeit ist die empirisch attestierte Deformation der **Einhüllenden** nicht zufälliger Natur. Von Nichtspherizität der **Einhüllenden** kann ausgegangen werden, was in diesem Beispiel für die grundsätzlich Sinnhaftigkeit der Durchführung einer **Hauptkomponentenanalyse** spricht.[3]

Die standardisierten **KMO-** und **MSA-Maße** nehmen für die Daten in $\boldsymbol{Z}$ folgende Werte an:

[3]Die Nichtspherizität der Einhüllenden der durch die Daten in $\boldsymbol{Z}$ gegebenen Punktewolke ließ sich bereits im oben erstellten 3D-Streudiagramm erkennen.

```
kmoZ <- KMOS(x = Z)
print(x = kmoZ, stats = "KMO")

##
## Kaiser-Meyer-Olkin Statistic
## Call: KMOS(x = Z)
##
## KMO-Criterion: 0.5478232

print(x = kmoZ, stats = "MSA", sort = TRUE, digits = 7, show = 1:3)

##
## Kaiser-Meyer-Olkin Statistics
##
## Call: KMOS(x = Z)
##
## Measures of Sampling Adequacy (MSA):
##   mass_std [1] height_std [1]    age_std [1]
##      0.5309161      0.5327821      0.7749174
```

Das **KMO-Maß** bezieht sich auf den gesamten (hier trivariaten) **Datensatz**, das **MSA-Maß** individuell jede der beteiligten **Variablen**. Empfohlen werden für die standardisierten **KMO-** und **MSA-Maße**, deren Wertespektra sich über das Intervall $[0;1]$ erstrecken, Werte zwischen $0,8$ und $1,0$; vgl. Hatzinger *et al* (2014) [7] und Hair *et al* (2010) [6]. In dieser Hinsicht stellt der hier betrachtete **trivariate Datensatz** in $\boldsymbol{Z}$ ein *Negativbeispiel* bzgl. der Eignung für eine **Hauptkomponentenanalyse** dar.

6 Berechnen der Korrelationsmatrix $\boldsymbol{R}$ und ihrer Inversen $\boldsymbol{R}^{-1}$

Die **Korrelationsmatrix** $\boldsymbol{R}$ des betrachteten **trivariaten Datensatzes** in $\boldsymbol{X}$ ist gegeben durch

$$\boldsymbol{R} = \frac{1}{n-1}\,\boldsymbol{Z}^{\top}\boldsymbol{Z}$$

also

```
Rmat <- (1/(nrow(Z) - 1)) * t(Z) %*% Z
Rmat

##                  height_std [1] mass_std [1] age_std [1]
## height_std [1]       1.0000000    0.6202596  -0.1863417
## mass_std [1]         0.6202596    1.0000000  -0.2308225
## age_std [1]         -0.1863417   -0.2308225   1.0000000
```

Die **Korrelationsmatrix** $\boldsymbol{R}$ besitzt eine von Null verschiedene **Determinante** und ist somit regulär. Folglich existiert eine **Inverse**, $\boldsymbol{R}^{-1}$, die hier der Vollständigkeit halber angegeben wird.

```
det(x = Rmat)

## [1] 0.5806328

RmatInv <- solve(Rmat)
RmatInv
```

```
##                  height_std [1] mass_std [1] age_std [1]
## height_std [1]      1.63049873   -0.9941702  0.07435314
## mass_std [1]       -0.99417018    1.6624566  0.19847692
## age_std [1]         0.07435314    0.1984769  1.05966802
```

Die **Spur** der **Korrelationsmatrix** $\boldsymbol{R}$ beträgt

```
sum(diag(x = Rmat))

## [1] 3
```

Sie entspricht somit genau der Anzahl von **Variablen** im betrachteten **trivariaten Datensatz** in $\boldsymbol{X}$.

7 Eigenwerte und Eigenvektoren: Eigenorthonormalbasis der Korrelationsmatrix

Die drei **Eigenwerte** der **Korrelationsmatrix** $\boldsymbol{R}$ sind

```
evAnaCor <- eigen(x = Rmat, symmetric = "TRUE")
evAnaCor$values

## [1] 1.7382412 0.8838105 0.3779482

sum(evAnaCor$values)

## [1] 3
```

und summieren sich genau zu der Anzahl von **Variablen** im betrachteten **trivariaten Datensatzes** in $\boldsymbol{X}$. Die drei paarweise orthogonalen, normierten **Eigenvektoren** der **Korrelationsmatrix** $\boldsymbol{R}$ sind (spaltenweise von links nach rechts)[4]

```
evAnaCor$vectors

##            [,1]      [,2]        [,3]
## [1,] -0.6504363 0.3033698  0.69634715
## [2,] -0.6625952 0.2215906 -0.71544756
## [3,]  0.3713492 0.9267494 -0.05688085
```

Diese werden auch die **Hauptkomponenten** der **Korrelationsmatrix** $\boldsymbol{R}$ genannt. Sie spannen die **Eigenorthonormalbasis** der **Korrelationsmatrix** $\boldsymbol{R}$ im Euklidischen Raum $\mathbb{R}^3$ auf; vgl. Bronstein *et al* (2005) [2].

Mit Bezug auf Kaisers (1960) [12] **Eigenwertkriterium** wird festgestellt, dass im vorliegenden Beispiel nur einer der drei **Eigenwerte** der **Korrelationsmatrix** $\boldsymbol{R}$ größer 1 ist, dementsprechend also nur eine **dominante Hauptkomponente** der **Korrelationsmatrix** $\boldsymbol{R}$ vorliegt.

Der **Erklärungswert** der einzelnen **Eigenwerte** (Hauptkomponenten) an der **Gesamtvarianz** des betrachteten standardisierten **trivariaten Datensatzes** in $\boldsymbol{Z}$ beläuft sich auf

[4] Bedauerlicherweise spukt R hier die Komponenten der drei Eigenvektoren nicht der mathematischen Konvention entsprechend als rechtshändisch orientierte Orthonormalbasis aus.

```r
round(evAnaCor$values/sum(evAnaCor$values), 4)

## [1] 0.5794 0.2946 0.1260
```

also $57,94\ \%$, $29,46\ \%$ und $12,60\ \%$, und kumuliert auf

```r
round(cumsum(evAnaCor$values)/sum(evAnaCor$values), 4)

## [1] 0.5794 0.8740 1.0000
```

Auf die Interpretation dieser **Eigenwerte** wird im nachfolgenden Abschnitt eingegangen.

8 Rotationsmatrix V, Eigenwertdiagonalmatrix Λ und inverse Eigenwertdiagonalmatrix Λ^{-1}

Aus den drei Eigenvektoren der **Korrelationsmatrix** $\boldsymbol{R}$ wird eine orthogonale **Rotationsmatrix** $\boldsymbol{V}$ gebildet, mithilfe welcher (im betrachteten Beispiel im Euklidischen Raum $\mathbb{R}^3$) Transformationen in die *rechtshändisch* orientierte **Eigenorthonormalbasis** der **Korrelationsmatrix** $\boldsymbol{R}$ vorgenommen werden können. Die **Determinante** der **Rotationsmatrix** $\boldsymbol{V}$ hat den Wert 1

```r
rotMatCor <- (-1) * evAnaCor$vectors  # scaling by a factor of (-1)
rotMatCor

##            [,1]       [,2]        [,3]
## [1,]  0.6504363 -0.3033698 -0.69634715
## [2,]  0.6625952 -0.2215906  0.71544756
## [3,] -0.3713492 -0.9267494  0.05688085

det(x = rotMatCor)

## [1] 1
```

Transformationen mit der **Rotationsmatrix** $\boldsymbol{V}$ sind folglich *Volumen erhaltend*.

Per Konstruktion genügt die **Rotationsmatrix** $\boldsymbol{V}$ den beiden **Orthogonalitätstests**

$$\mathbf{1} = \boldsymbol{V}^\top \boldsymbol{V} = \boldsymbol{V}\boldsymbol{V}^\top$$

also

```r
round(t(rotMatCor) %*% rotMatCor, 4)

##      [,1] [,2] [,3]
## [1,]    1    0    0
## [2,]    0    1    0
## [3,]    0    0    1

round(rotMatCor %*% t(rotMatCor), 4)

##      [,1] [,2] [,3]
## [1,]    1    0    0
## [2,]    0    1    0
## [3,]    0    0    1
```

Durch **Diagonalisieren** der **Korrelationsmatrix** $\boldsymbol{R}$ über Transformation mit der **Rotationsmatrix** $\boldsymbol{V}$ gelangt man zur **Eigenwertdiagonalmatrix** $\boldsymbol{\Lambda}$ gemäß

$$\boldsymbol{\Lambda} = \boldsymbol{V}^{\top}\boldsymbol{R}\boldsymbol{V}$$

also

```
LambdaCor <- t(rotMatCor) %*% Rmat %*% rotMatCor
round(LambdaCor, 7)

##          [,1]      [,2]      [,3]
## [1,] 1.738241 0.0000000 0.0000000
## [2,] 0.000000 0.8838105 0.0000000
## [3,] 0.000000 0.0000000 0.3779482
```

Die **Eigenwertdiagonalmatrix** $\boldsymbol{\Lambda}$ ist nichts anderes als die Darstellung der **Korrelationsmatrix** $\boldsymbol{R}$ bzgl. ihrer **Eigenorthonormalbasis** im Euklidischen Raum $\mathbb{R}^3$.

Es folgt ein **Konsistenztest** für die **Eigenwertdiagonalmatrix** $\boldsymbol{\Lambda}$,

$$\boldsymbol{0} = \boldsymbol{\Lambda} - \operatorname{diag}(\lambda_1, \ldots, \lambda_m)$$

also

```
round(LambdaCor - diag(x = evAnaCor$values), 4)

##      [,1] [,2] [,3]
## [1,]    0    0    0
## [2,]    0    0    0
## [3,]    0    0    0
```

Die **Inverse**, $\boldsymbol{\Lambda}^{-1}$, der **Eigenwertdiagonalmatrix** $\boldsymbol{\Lambda}$ wird durch

```
LambdaCorInv <- diag(x = (1/evAnaCor$values))
LambdaCorInv

##           [,1]     [,2]     [,3]
## [1,] 0.5752941 0.000000 0.000000
## [2,] 0.0000000 1.131464 0.000000
## [3,] 0.0000000 0.000000 2.645865
```

berechnet.

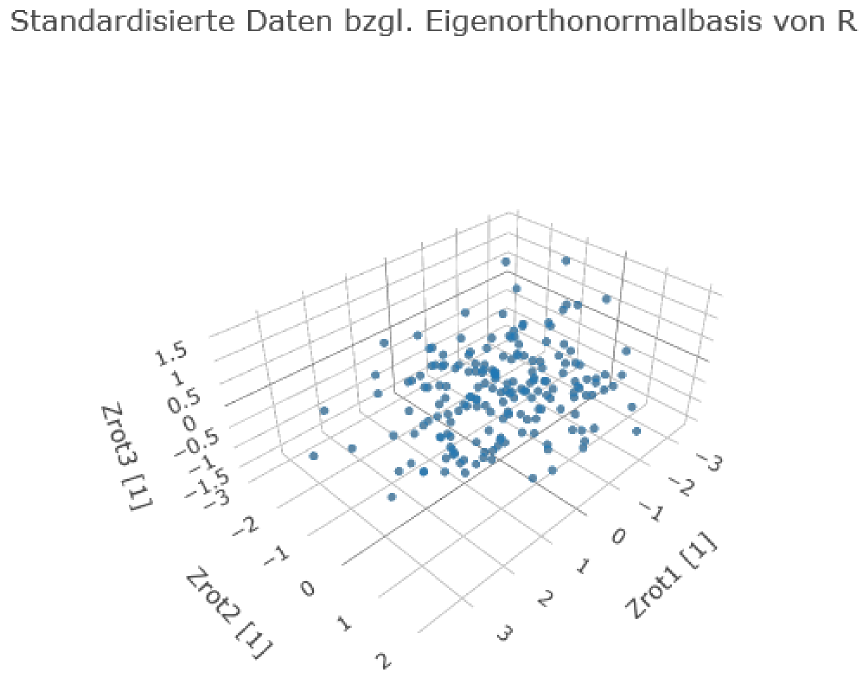
Die drei **Eigenwerte** der **Korrelationsmatrix** $\boldsymbol{R}$ entsprechen den **Varianzen** der Daten in $\boldsymbol{Z}$ entlang jeder der durch die **Eigenorthonormalbasis** der **Korrelationsmatrix** $\boldsymbol{R}$ vorgegebenen drei Richtungen im Euklidischen Raum $\mathbb{R}^3$. Die nächste Betrachtung verdeutlicht diesen Sachverhalt. Transformation der Daten in $\boldsymbol{Z}$ in die **Eigenorthonormalbasis** der **Korrelationsmatrix** $\boldsymbol{R}$ führt zur Datenmatrix

$$\boldsymbol{Z}_{\text{rot}} = \boldsymbol{Z}\boldsymbol{V}$$

also

```
Zrot <- Z %*% rotMatCor
```

und eine daran anschließende Visualisierung der resultierenden Daten in $\boldsymbol{Z}_{\text{rot}}$ per **3D-Streudiagramm** liefert



Die nun ausgeführte Berechnung der **Varianzen** der Daten in $\boldsymbol{Z}_{\text{rot}}$ führt zu

```
apply(X = Zrot, MARGIN = 2, FUN = var)

## [1] 1.7382412 0.8838105 0.3779482
```

also zu dem behaupteten Ergebnis. Beachte: Die Daten in den Spalten von $\boldsymbol{Z}_{\text{rot}}$ sind paarweise *unkorreliert*

```
round(cor(x = Zrot), 4)

##      [,1] [,2] [,3]
## [1,]    1    0    0
## [2,]    0    1    0
## [3,]    0    0    1
```

9 Hauptkomponentenladungsmatrix $\boldsymbol{A}$

Die **Hauptkomponentenladungsmatrix** $\boldsymbol{A}$, definiert über die orthogonale **Rotationsmatrix** $\boldsymbol{V}$ und die **Eigenwertdiagonalmatrix** $\boldsymbol{\Lambda}$ durch

$$\boldsymbol{A} := \boldsymbol{V}\boldsymbol{\Lambda}^{1/2}$$

liefert die Antwort auf die Frage: Wie stark korrelieren die (hier) drei **Ausgangsvariablen** Körpergröße, Masse und Alter mit den gerade bestimmten drei **Hauptkomponenten** (Eigenvektoren) der **Korrelationsmatrix** $\boldsymbol{R}$?

```
AmatCor <- rotMatCor %*% LambdaCor^(1/2)
rownames(AmatCor) <- c("height", "mass", "age")
colnames(AmatCor) <- c("PC1", "PC2", "PC3")
AmatCor

##                PC1        PC2         PC3
## height  0.8575507 -0.2852016 -0.42809679
## mass    0.8735813 -0.2083199  0.43983925
## age    -0.4895956 -0.8712482  0.03496889
```

Dass die **Hauptkomponentenladungsmatrix** $\boldsymbol{A}$ in der Tat formal als eine Korrelationsmatrix interpretiert werden, wird weiter unten deutlich werden.

Die **Hauptkomponentenladungsmatrix** $\boldsymbol{A}$ erfüllt zwei **Konsistenztests**:

1. Die **Korrelationsmatrix** $\boldsymbol{R}$ lässt sich mithilfe der **Hauptkomponentenladungsmatrix** $\boldsymbol{A}$ faktorisieren,
$$\mathbf{0} = \boldsymbol{R} - \boldsymbol{A}\boldsymbol{A}^\top$$
also

```
round(Rmat - AmatCor %*% t(AmatCor), 4)

##                 height_std [1] mass_std [1] age_std [1]
## height_std [1]              0            0           0
## mass_std [1]                0            0           0
## age_std [1]                 0            0           0
```

2. Die **Eigenwertdiagonalmatrix** $\boldsymbol{\Lambda}$ lässt sich mithilfe der **Hauptkomponentenladungsmatrix** $\boldsymbol{A}$ faktorisieren,
$$\mathbf{0} = \boldsymbol{\Lambda} - \boldsymbol{A}^\top\boldsymbol{A}$$
also

```
round(LambdaCor - t(AmatCor) %*% AmatCor, 4)

##     PC1 PC2 PC3
## PC1   0   0   0
## PC2   0   0   0
## PC3   0   0   0
```

10 Standardisierter Datensatz in der Eigenorthonormalbasis der Korrelationsmatrix ($\boldsymbol{F}$-Matrix)

Es folgt die **Transformation** des standardisierten **trivariaten Datensatzes** in $\boldsymbol{Z}$ in die **Eigenorthonormalbasis** der **Korrelationsmatrix** $\boldsymbol{R}$ mit der einer Konvention der Statistik entsprechenden Maßgabe, dass die resultierenden Daten gleichfalls standardisiert sind. Für die Umsetzung dieses Anliegens wird eine durch die **Rotationsmatrix** $\boldsymbol{V}$ beschriebene, Volumen erhaltende **Drehung** des Ausgangskoordinatensystems[5] mit einer Volumen verändernden **Reskalierung** der Koordinatenachsen mithilfe der (Wurzel

[5]Die Auswirkungen alleine dieser Rotationstransformation auf die Daten in $\boldsymbol{Z}$ waren zuvor in Abschnitt 8 beschrieben und visualisiert worden.

der) **Inversen** $\boldsymbol{\Lambda}^{-1}$ der **Eigenwertdiagonalmatrix** $\boldsymbol{\Lambda}$ kombiniert. Diese kombinierte **Transformation** definiert die $\boldsymbol{F}$-Matrix,[6]

$$\boldsymbol{F} = \boldsymbol{Z}\boldsymbol{V}\boldsymbol{\Lambda}^{-1/2}$$

also

```
FmatCor <- Z %*% rotMatCor %*% LambdaCorInv^(1/2)
colnames(FmatCor) <- c("PC1_std [1]", "PC2_std [1]", "PC3_std [1]")
dim(FmatCor)

## [1] 187   3

head(x = FmatCor)

##      PC1_std [1] PC2_std [1] PC3_std [1]
## [1,]  -1.8362245  -0.4986426   1.1623874
## [2,]  -2.6100451  -0.2165376   0.8829885
## [3,]  -0.6264361  -0.3416280   0.8556499
## [4,]  -0.2097674   0.7040217  -0.7566757
## [5,]  -0.4219218   1.7223008  -1.2765885
## [6,]   1.1094795  -1.0397559   0.7139017
```

die standardisierte und per Konstruktion paarweise *unkorrelierte*, so genannte "f-Werteënthält.

10.1 Konsistenztests für die $\boldsymbol{F}$-Matrix

Folgende **Konsistenztests** können für die $\boldsymbol{F}$-Matrix durchgeführt werden:

1. Die "f-Werteßind standardisiert und paarweise unkorreliert,

$$\mathbf{0} = \mathbf{1} - \frac{1}{n-1}\boldsymbol{F}^\top\boldsymbol{F}$$

also

```
proxy4 <- (1/(nrow(FmatCor) - 1)) * t(FmatCor) %*% FmatCor
round(diag(rep(1, nrow(proxy4))) - proxy4, 4)

##             PC1_std [1] PC2_std [1] PC3_std [1]
## PC1_std [1]           0           0           0
## PC2_std [1]           0           0           0
## PC3_std [1]           0           0           0
```

2. Die Elemente der **Hauptkomponentenladungsmatrix** $\boldsymbol{A}$ repräsentieren, als algebraische Projektionen von standardisierten "z-Wertenäuf standardisierte "f-Werte", bivariate Korrelationen zwischen den **Ausgangsvariablen** und den **Hauptkomponenten** (vgl. die Bemerkungen zu Beginn von Abschnitt 9),

$$\mathbf{0} = \boldsymbol{A} - \frac{1}{n-1}\boldsymbol{Z}^\top\boldsymbol{F}$$

also

[6] Äquivalent kann die $\boldsymbol{F}$-Matrix auch unter Einbinden der Hauptkomponentenladungsmatrix $\boldsymbol{A}$ über $\boldsymbol{F} = \boldsymbol{Z}\boldsymbol{A}\boldsymbol{\Lambda}^{-1}$ berechnet werden.

```
round(AmatCor - (1/(nrow(Z) - 1)) * t(Z) %*% FmatCor, 4)

##        PC1 PC2 PC3
## height   0   0   0
## mass     0   0   0
## age      0   0   0
```

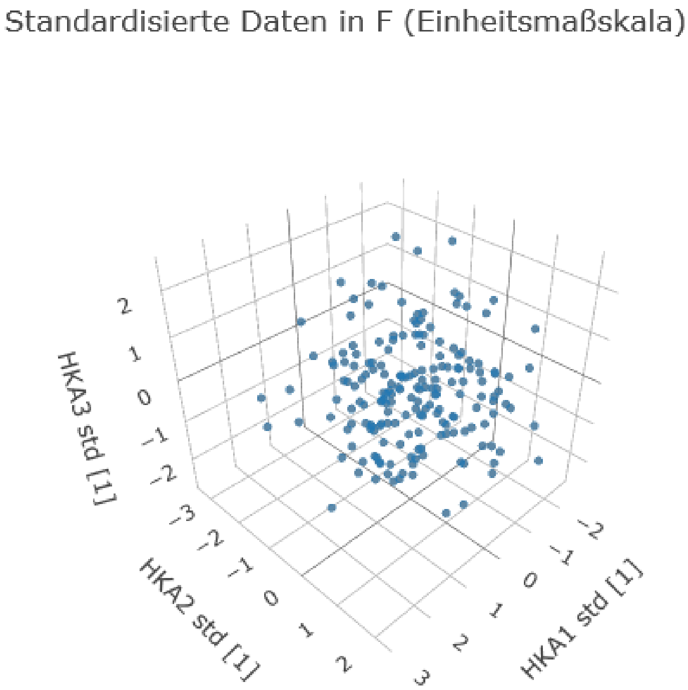
3. Die "z-Werte"können als Linearkombinationen von "f-Wertenäufgefasst werden,

$$\mathbf{0} = \boldsymbol{Z} - \boldsymbol{F}\boldsymbol{A}^\top$$

also

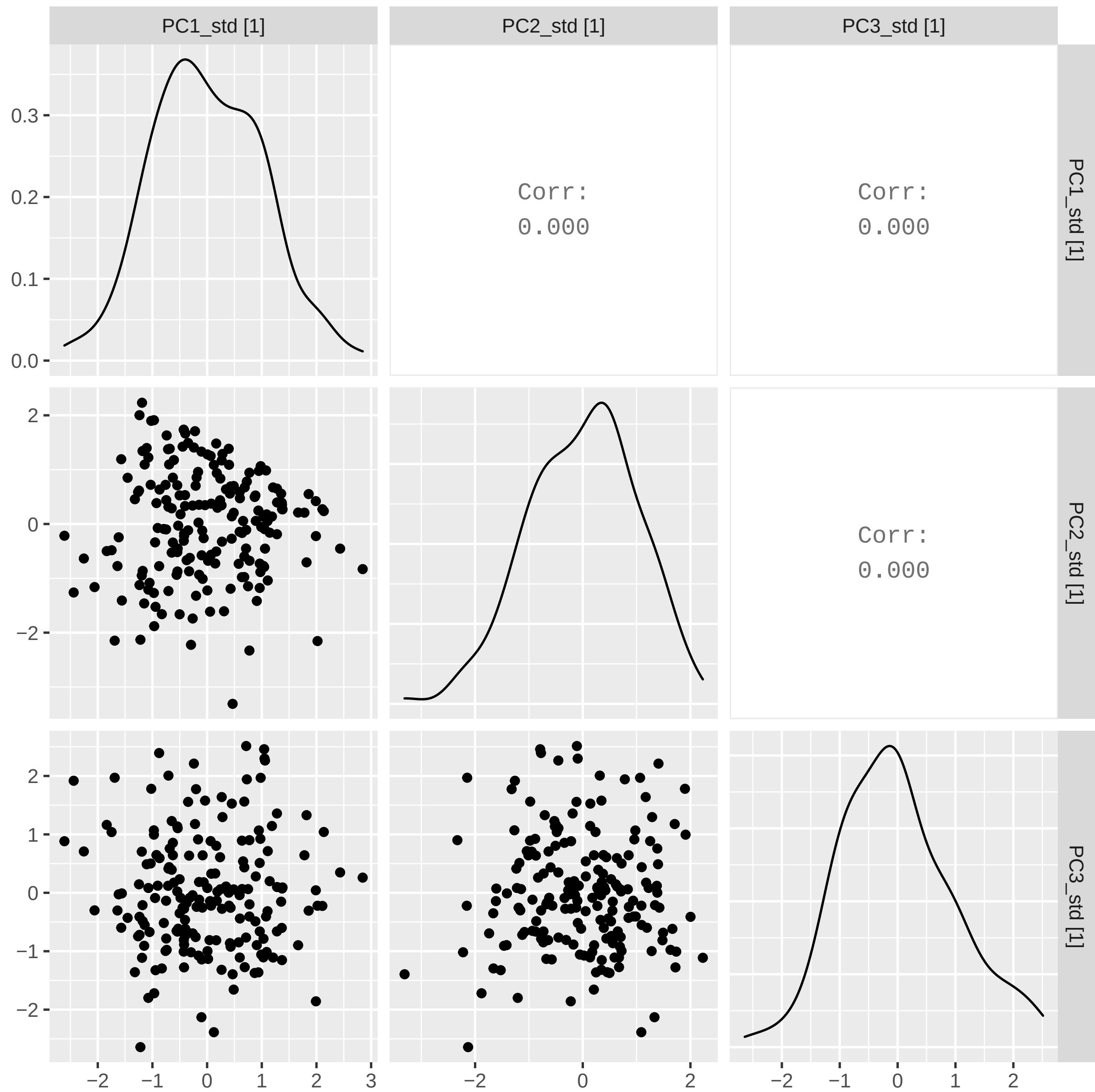
```
head(x = round(Z - FmatCor %*% t(AmatCor), 4))

##      height_std [1] mass_std [1] age_std [1]
## [1,]              0            0           0
## [2,]              0            0           0
## [3,]              0            0           0
## [4,]              0            0           0
## [5,]              0            0           0
## [6,]              0            0           0

tail(x = round(Z - FmatCor %*% t(AmatCor), 4))

##        height_std [1] mass_std [1] age_std [1]
## [182,]              0            0           0
## [183,]              0            0           0
## [184,]              0            0           0
## [185,]              0            0           0
## [186,]              0            0           0
## [187,]              0            0           0
```

10.2 Visualisieren der Daten in F per 3D-Streudiagramm



10.3 Visualisieren der Daten in F per Streudiagrammmatrix

```
ggpairs(data = as.data.frame(FmatCor))
```



Die **Streudiagrammmatrix** verdeutlicht ebenfalls das paarweise Nichtkorreliertsein der den einzelnen **Hauptkomponenten** zugeordneten "f-Werten".

11 Dimensionsreduktion: Extraktion der einzigen dominanten Hauptkomponente

Im Anschluss an die umfangreiche Diskussion der für eine **Hauptkomponentenanalyse** eines multivariaten metrisch skalierten Datensatzes benötigten linear-algebraischen Methodik, wird nun abschließend das Vorgehen bei einer **Dimensionsreduktion** vorgestellt.

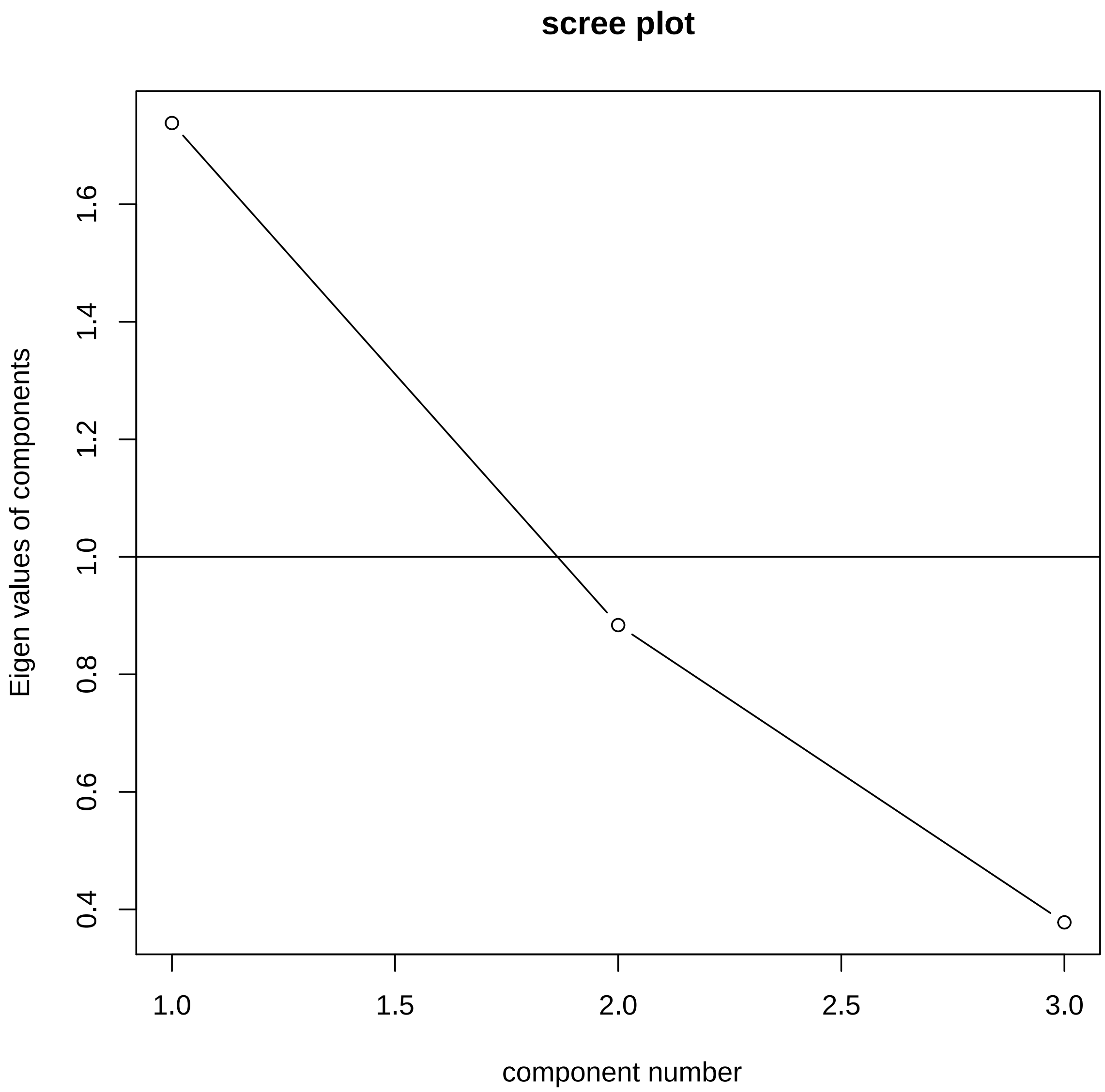
Die **Dimensionsreduktion** wird in der Eigenorthonormalbasis der **Korrelationsmatrix** $\boldsymbol{R}$ vorgenommen. Verkürzt dargestellt, werden von den "f-Werten"nur jene beibehalten, welche den **dominanten Hauptkomponenten** zugeordnet sind. Alle Restlichen werden verworfen; ein damit verbundener Informationsverlust wird zugunsten einer geringeren Komplexität des gegebenen Sachverhalts in Kauf genommen. Die verbleibenden "f-Werte"werden in die **Ausgangsorthonormalbasis** rücktransformiert und dann bezüglich der **Originalmaßskalen** dargestellt. Die so erhaltenen **dimensionsreduzierten Mess-**

werte bilden den Ausgangspunkt möglicher weiterer statistischer Anwendungen.[7]

11.1 Qualitatives Extraktionskriterium

Das R-Paket `psych` stellt die Funktion `VSS.scree()` zur Erstellung eines **Gerölldiagramms** nach Cattell (1966) [3] bereit.

```
VSS.scree(rx = Z)
```



Zu extrahieren ist nach erfahrungsbasierter Empfehlung jene Anzahl von **Hauptkomponenten**, deren **Eigenwerte** im **Gerölldiagramm** *links* des Ëllbogensßu liegen kommen; im betrachteten Beipiel also einer. Dieses Resultat ist im vorliegeneden Fall konsistent mit Kaisers (1960) [12] **Eigenwertkriterium**, welches die **Extraktion** aller **Hauptkomponenten** der **Korrelationsmatrix** $\boldsymbol{R}$ mit einem **Eigenwert** größer 1 empfiehlt.

[7] Für die Umsetzung einer Dimensionsreduktion werden die $\boldsymbol{F}$ -Matrix bzw. die "f-Werte"nicht wirklich benötigt. Die in die Eigenorthonormalbasis der Korrelationsmatrix $\boldsymbol{R}$ transformierten "z-Werteßind für dieses Anliegen vollkommen ausreichend. Das Standardisieren entspricht lediglich einer in der Statistik gepflegten, da häufig nützlichen, Konvention.

Beispielhaft wird im Rahmen der weiteren Diskussion eine **Dimensionsreduktion** für den gegebenen **trivariaten Datensatz** in $\boldsymbol{X}$ auf Basis der **Extraktion** der einzigen **dominanten Hauptkomponente** der **Korrelationsmatrix** $\boldsymbol{R}$ demonstriert.

11.2 Dimensionsreduzierte Matrizen

Das Vorgehen bei einer **Dimensionsreduktion** spiegelt sich insbesondere in den Matrizen der beschriebenen linear-algebraischen Methodik wieder.

1. Dimensionsreduzierte Rotationsmatrix $\boldsymbol{V}_{\text{red}}$ — nur die Eigenvektoren der Korrelationsmatrix $\boldsymbol{R}$ mit Eigenwerten größer 1 werden zur Konstruktion der Rotationsmatrix verwendet:

```
rotMatCorRed <- as.matrix((-1) * evAnaCor$vectors[, 1])
rotMatCorRed

##            [,1]
## [1,]  0.6504363
## [2,]  0.6625952
## [3,] -0.3713492
```

2. Dimensionsreduzierte Eigenwertdiagonalmatrix $\boldsymbol{\Lambda}_{\text{red}}$— erzeugt aus der Korrelationsmatrix $\boldsymbol{R}$ durch Transformation mit der dimensionsreduzierten Rotationsmatrix:

```
LambdaCorRed <- t(rotMatCorRed) %*% Rmat %*% rotMatCorRed
LambdaCorRed

##          [,1]
## [1,] 1.738241
```

3. Inverse der dimensionsreduzierten Eigenwertdiagonalmatrix, $\boldsymbol{\Lambda}_{\text{red}}^{-1}$

```
LambdaCorRedInv <- solve(LambdaCorRed)
LambdaCorRedInv

##           [,1]
## [1,] 0.5752941
```

4. Dimensionsreduzierte Hauptkomponentenladungsmatrix $\boldsymbol{A}_{\text{red}}$

```
AmatCorRed <- rotMatCorRed %*% LambdaCorRed^(1/2)
AmatCorRed

##            [,1]
## [1,]  0.8575507
## [2,]  0.8735813
## [3,] -0.4895956
```

5. Dimensionsreduzierte $\boldsymbol{F}$-Matrix, $\boldsymbol{F}_{\text{red}}$

```
FmatCorRed <- Z %*% AmatCorRed %*% LambdaCorRedInv
dim(FmatCorRed)

## [1] 187   1

head(x = FmatCorRed)

##            [,1]
## [1,] -1.8362245
## [2,] -2.6100451
## [3,] -0.6264361
## [4,] -0.2097674
## [5,] -0.4219218
## [6,]  1.1094795

(1/(nrow(FmatCorRed) - 1)) * t(FmatCorRed) %*% FmatCorRed

##      [,1]
## [1,]    1
```

11.3 Vergleich des trivariaten Beispieldatensatzes mit seiner dimensionsreduzierten Variante

Die in der dimensionsreduzierten $\boldsymbol{F}$-Matrix, $\boldsymbol{F}_{\text{red}}$, verbliebenen "$f$-Werte"werden in die **Ausgangsorthonormalbasis** rücktransformiert und bezüglich der **Originalmaßskalen** ausgedrückt. Zu illustrativen Zwecken werden sie hier stichprobenartig mit den nichtdimensionsreduzierten "z-Werten"bzw. den ursprünglichen Messwerten verglichen.

Standardisierte Maßskala

$\boldsymbol{Z}_{\text{red}}$ vs $\boldsymbol{Z}$

```
Zapprox <- FmatCorRed %*% t(AmatCorRed)
colnames(Zapprox) <- c("height_std [1]", "mass_std [1]", "age_std [1]")
```

```
head(x = Z)

##      height_std [1] mass_std [1] age_std [1]
## [1,]     -1.9300562   -0.9889505   1.3740963
## [2,]     -2.5544936   -1.8466045   1.4974016
## [3,]     -0.8060688   -0.0997265   0.6342642
## [4,]     -0.0567439   -0.6627263  -0.5371365
## [5,]     -0.3065189   -1.2888663  -1.3386213
## [6,]      0.9423560    1.4998244   0.3876535

head(x = Zapprox)

##      height_std [1] mass_std [1] age_std [1]
## [1,]     -1.5746556   -1.6040913   0.8990074
```

```
## [2,]      -2.2382460   -2.2800865   1.2778665
## [3,]      -0.5372007   -0.5472428   0.3067003
## [4,]      -0.1798862   -0.1832489   0.1027012
## [5,]      -0.3618193   -0.3685830   0.2065711
## [6,]       0.9514349    0.9692205  -0.5431963

tail(x = Z)

##         height_std [1] mass_std [1] age_std [1]
## [182,]       1.3170185    0.4106565  -0.4754839
## [183,]      -0.6811813   -0.4469974  -0.2042121
## [184,]       0.5676935   -0.1839134   0.5109589
## [185,]       2.5658933    0.9683947  -0.8453999
## [186,]      -1.3056188   -1.4046233  -0.5987892
## [187,]       1.3170185    2.2732915  -1.2153159

tail(x = Zapprox)

##         height_std [1] mass_std [1] age_std [1]
## [182,]      0.84901854   0.86488963 -0.48472437
## [183,]     -0.43150557  -0.43957190  0.24635654
## [184,]      0.03749388   0.03819477 -0.02140612
## [185,]      1.70709764   1.73900920 -0.97462164
## [186,]     -1.01309268  -1.03203088  0.57839810
## [187,]      1.83046773   1.86468550 -1.04505648
```

Originalmaßskalen

Dies erfordert eine **Rücktransformation** (Destandardisierung) der Daten in $\boldsymbol{Z}_{\text{red}}$ bzw. $\boldsymbol{Z}$ auf die ursprünglich für die drei Variablen Körpergröße, Masse und Alter verwendeten **Maßskalen**:

```
b <- attr(x = Z, "scaled:scale")
a <- attr(x = Z, "scaled:center")
Xapp_int <- Zapprox * rep(b, each = nrow(Zapprox)) + rep(a, each = nrow(Zapprox)
XapproxCor <- data.frame(Xapp_int)
colnames(XapproxCor) <- c("height [cm]", "mass [kg]", "age [yr]")
```

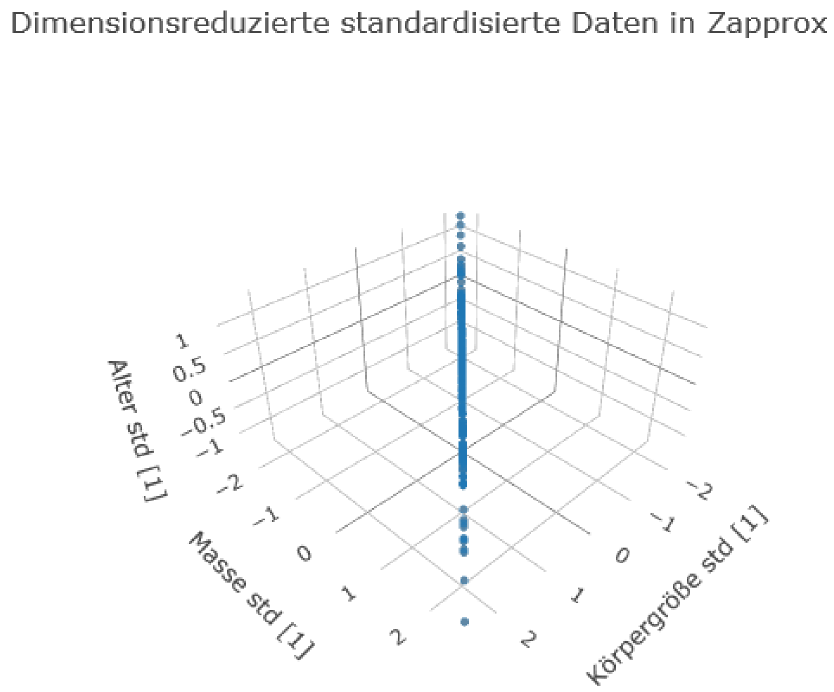
$\boldsymbol{X}_{\text{red}}$ vs $\boldsymbol{X}$

```
head(x = X[, 1:3])

##   height [cm] mass [kg] age [yr]
## 1     139.700  36.48581       63
## 2     136.525  31.86484       65
## 3     145.415  41.27687       51
## 4     149.225  38.24348       32
## 5     147.955  34.86988       19
## 6     154.305  49.89512       47

head(x = XapproxCor)
```

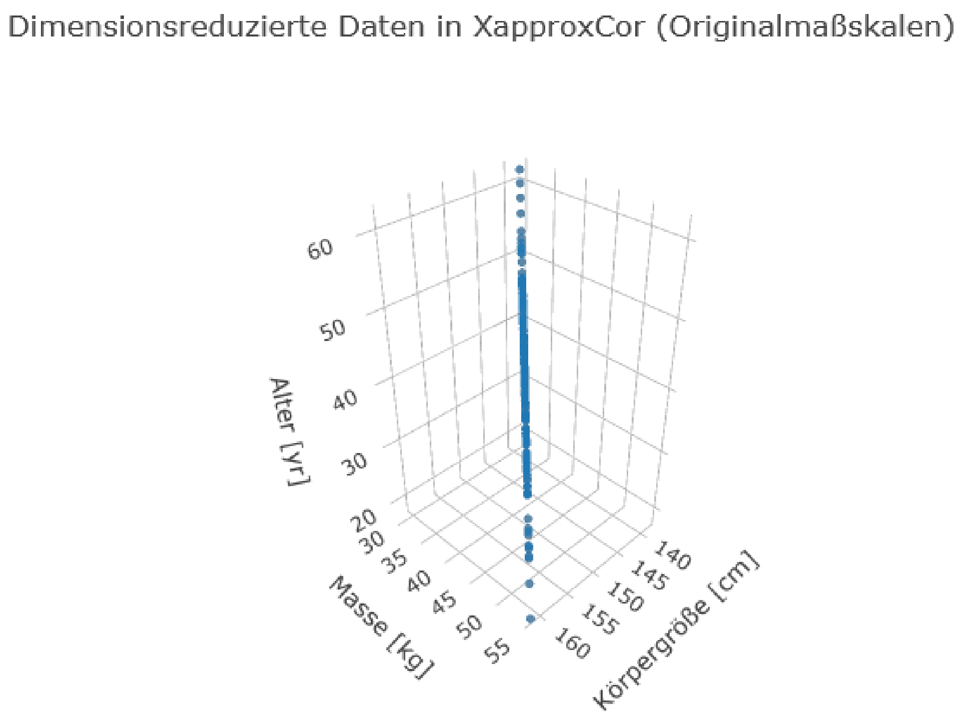
```
##   height [cm] mass [kg] age [yr]
## 1    141.5071  33.17148 55.29411
## 2    138.1330  29.52927 61.43916
## 3    146.7821  38.86569 45.68695
## 4    148.5989  40.82686 42.37810
## 5    147.6738  39.82830 44.06286
## 6    154.3512  47.03627 31.90171

tail(x = X[, 1:3])

##     height [cm] mass [kg] age [yr]
## 182     156.210  44.02677     33.0
## 183     146.050  39.40581     37.4
## 184     152.400  40.82328     49.0
## 185     162.560  47.03182     27.0
## 186     142.875  34.24620     31.0
## 187     156.210  54.06250     21.0

tail(x = XapproxCor)

##     height [cm] mass [kg] age [yr]
## 182    153.8304  46.47414 32.85012
## 183    147.3195  39.44581 44.70818
## 184    149.7042  42.01998 40.36509
## 185    158.1934  51.18383 24.90404
## 186    144.3624  36.25369 50.09386
## 187    158.8207  51.86096 23.76159
```

11.4 Visualisieren der dimensionsreduzierten Daten per 3D-Streudiagramm

Standardisierte Maßskala



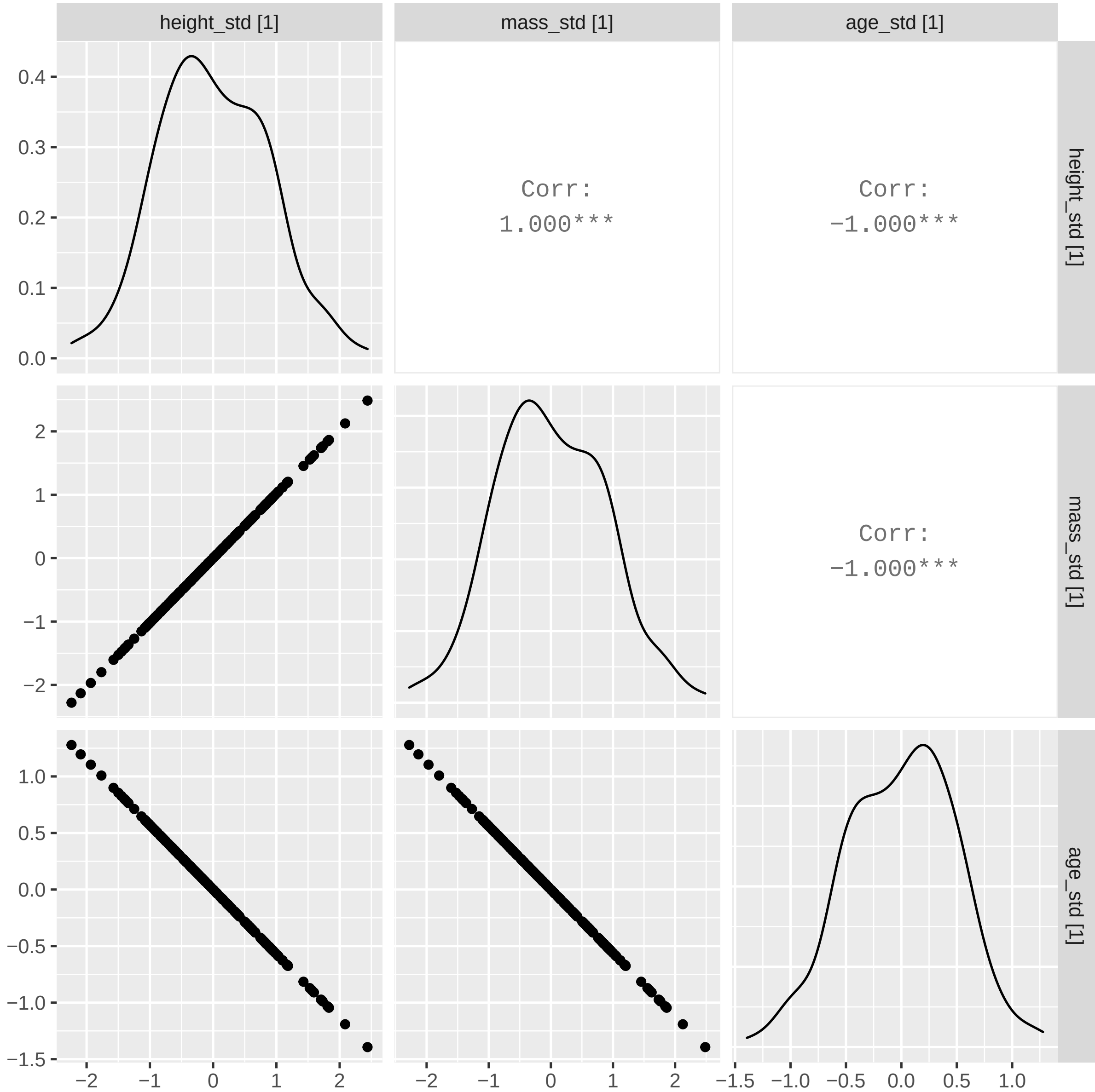
Originalmaßskalen



11.5 Visualisieren der dimensionsreduzierten Daten per Streudiagrammmatrix

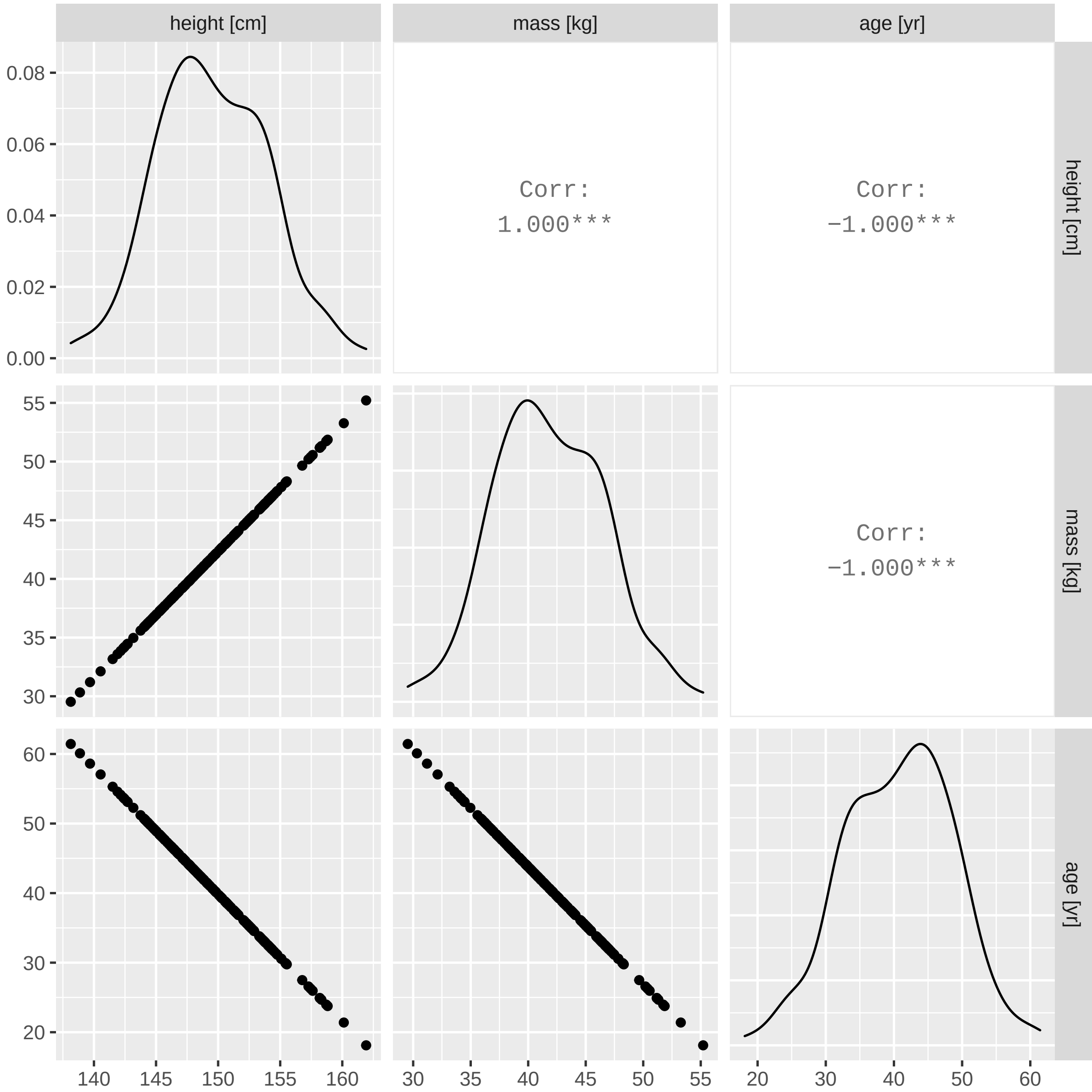
Standardisierte Maßskala

```
ggpairs(data = as.data.frame(Zapprox))
```



Originalmaßskalen

```
ggpairs(data = XapproxCor)
```



12 Fazit

Im vorliegenden Beispiel führt die **Dimensionsreduktion** zu einem **Extremfall**; der betrachtete trivariate Datensatz in $\boldsymbol{X}$ wurde zu einem effektiv univariaten Datensatz reduziert, welcher $57,94\ \%$ der Gesamtvarianz des Ausgangsdatensatzes zu erklären vermag. Im dimensionsreduzierten Datensatz liegen maximal (minimal) mögliche Werte für die bivariaten Korrelationen zwischen den drei Ausgangsvariablen vor.

Danksagung

Hilfreiche Kommentare von Jana Orthey und Laurens van der Woude haben dazu beigetragen, dieses Tutorium adressatengerecht zu gestalten.

Literatur

[1] M S Bartlett (1951) The effect of standardization on a chi square approximation in factor analysis *Biometrika* **38** 337–344

[2] I N Bronstein, K A Semedjajew, G Musiol und H Mühlig (2005) *Taschenbuch der Mathematik* 6. Aufl. (Frankfurt (Main): Harri Deutsch) ISBN–10: 3817120060

[3] R B Cattell (1966) The scree test for the number of factors *Multivariate Behavioral Research* **1** 629–637

[4] H van Elst (2019) Foundations of descriptive and inferential statistics (version 4) *Preprint* arXiv:1302.2525v4 [stat.AP]

[5] L Guttman (1953) Image theory for the structure of quantitative variates *Psychometrika* **18** 277–296

[6] J F Hair jr, W C Black, B J Babin and R E Anderson (2010) *Multivariate Data Analysis* 7th Edition (Upper Saddle River, NJ: Pearson) ISBN–13: 9780135153093

[7] R Hatzinger, K Hornik, H Nagel und M J Maier (2014) *R — Einführung durch angewandte Statistik* 2. Aufl. (München: Pearson Studium) ISBN–13: 9783868942507

[8] H Hotelling (1933) Analysis of a complex of statistical variables into principal components *Journal of Educational Psychology* **24** 417–441

[9] N Howell (2001) *Demography of the Dobe !Kung* 2nd Edition (Abingdon, Oxon: Routledge) ISBN–13: 9780202306490

[10] D N Joanes and C A Gill (1998) Comparing measures of sample skewness and kurtosis *The Statistician* **47** 183–189

[11] I T Jolliffe (2002) *Principal Component Analysis* 2nd Edition (New York: Springer) ISBN–10: 0387954422

[12] H F Kaiser (1960) The application of electronic computers to factor analysis *Educational and Psychological Measurement* **20** 141–151

[13] H F Kaiser (1970) A second generation little jiffy *Psychometrika* **35** 401–415

[14] K Pearson (1901) LIII. On lines and planes of closest fit to systems of points in space *Philosophical Magazine Series 6* **2** 559–572

[15] R Core Team (2020) *R: A language and environment for statistical computing* (Wien: R Foundation for Statistical Computing) URL (cited on December 5, 2020): https://www.R-project.org/

[16] H Toutenburg (2004) *Deskriptive Statistik* 4. Aufl. (Berlin: Springer) ISBN–10: 3540222332