TUTORIAL ON PRINCIPAL COMPONENT ANALYSIS, WITH APPLICATIONS IN R

HENK VAN ELST*

parcIT GmbH, Erftstraße 15, 50672 Köln, Germany

December 6, 2020

Abstract

This tutorial reviews the main steps of the principal component analysis of a multivariate data set and its subsequent dimensional reduction on the grounds of identified dominant principal components. The underlying computations are demonstrated and performed by means of a script written in the statistical software package R.

1 Introduction

This tutorial demonstrates the main theoretical and practical steps of the **principal component analysis** of a metrically scaled **multivariate data set**, and a subsequently performed **dimensional reduction** of the data set considered. The tutorial is based on a transparent exposition of all relevant computations by means of a script written with the statistical software package **R**; cf. R Core Team (2020) [15].

The methodological foundations of the **principal component analysis** and an associated **dimensional reduction** were laid in particular in the works by Pearson (1901) [14], Hotelling (1933) [8] and Kaiser (1960) [12]; see also Hatzinger *et al* (2014) [7], Hair *et al* (2010) [6] or Jolliffe (2002) [11].

The discussion to follow is divided into three parts. First, in Secs. 2 to 5, a **trivariate example data set** with **measured values** for three metrically scaled **variables** is loaded and then characterised and visualised with standard methods of **Descriptive Statistics**. Then, in Secs. 6 to 10, the central tool from **Linear Algebra** needed to perfom a **principal component analysis** is reviewed: this is the **eigenvalue analysis** of symmetrical quadratic matrixes and their **diagonalisation** by means of **rotational transformations** constructed from the matrixes' **eigenvectors**. In **Analytic Geometry** this method is also known as principal axes transformation; cf. Bronstein *et al* (2005) [2]. Lastly, in Sec. 11, the procedure of a **dimensional reduction** of a **multivariate data set** based on an **eigenvalue analysis** of its **sample correlation matrix** is outlined in the context of the given **trivariate example data set**. The tutorial ends with a conclusion in Sec. 12.

The results to be presented have been generated with R Version 3.6.3.

2 Loading of required R packages

The following R packages and self-written script are loaded into an R session in order to perform all the calculations involved in this tutorial and to generate helpful visualisations of the distributions observed in the analysed data:

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

```
library(tidyverse)

## - Attaching packages --------------------------------- tidyverse
1.3.0 -
## v ggplot2 3.3.2     v purrr   0.3.3
## v tibble  2.1.3     v dplyr   1.0.2
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0
## - Conflicts ----------------------------------
tidyverse_conflicts() -
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()

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(e1071)
library(GGally)

## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2

source("descripStats.R")
```

3 Loading of trivariate example data set (matrix X)

The example employed in this tutorial for illustrative purposes is given by a **trivariate data set** that contains **measured values** for the three metrically scaled **variables** height [cm], mass [kg] and age [yr] from a sample of $n = 187$ adult women. This **trivariate data set** is part of a larger data set according to

Howell (2001) [9], which needs to be loaded into the R session.[1]

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

## '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 ...
```

The **trivariate data set** that comprises **measured values** for the three **variables** height [cm], mass [kg] and age [yr] for a sample of women aged 18 yr or more is obtained via adequate filtering.

```
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 ...
```

The **data matrix** $\boldsymbol{X}$ constitutes the raw data matrix for the theoretical and practical considerations outlined in this tutorial.

3.1 Visualisation of data in $\boldsymbol{X}$ via 3D scatter plot

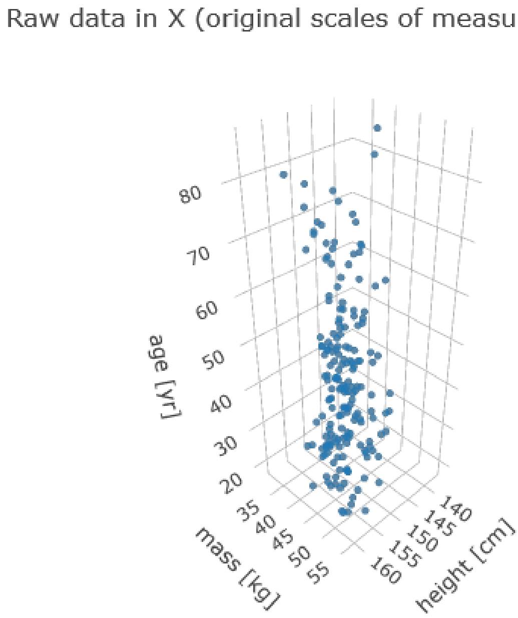
To begin with, the **trivariate data set** in $\boldsymbol{X}$ is first visualised by means of a **3D scatter plot**. This is realised by using the function `plot_ly()` from the package `plotly`.[2]

```
fig1 <- plot_ly(
  data = as.data.frame(X),
  type = "scatter3d",
  x = X[, 1],
  y = X[, 2],
  z = X[, 3],
  mode = "markers",
  size = 1
) %>%
  layout(
    title = "Raw data in X (original scales of measurement)",
    scene = list(
      xaxis = list(title = "height [cm]"),
      yaxis = list(title = "mass [kg]"),
      zaxis = list(title = "age [yr]")
```

[1]The complete original data set may be obtained from the URL tspace.library.utoronto.ca/handle/1807/10395.

[2]Note that the orientation of the scale of measurement along the "x"-axis of this 3D scatter plot does not conform to the mathematical convention; the resultant reference frame does *not* constitute a right-handed oriented reference frame.

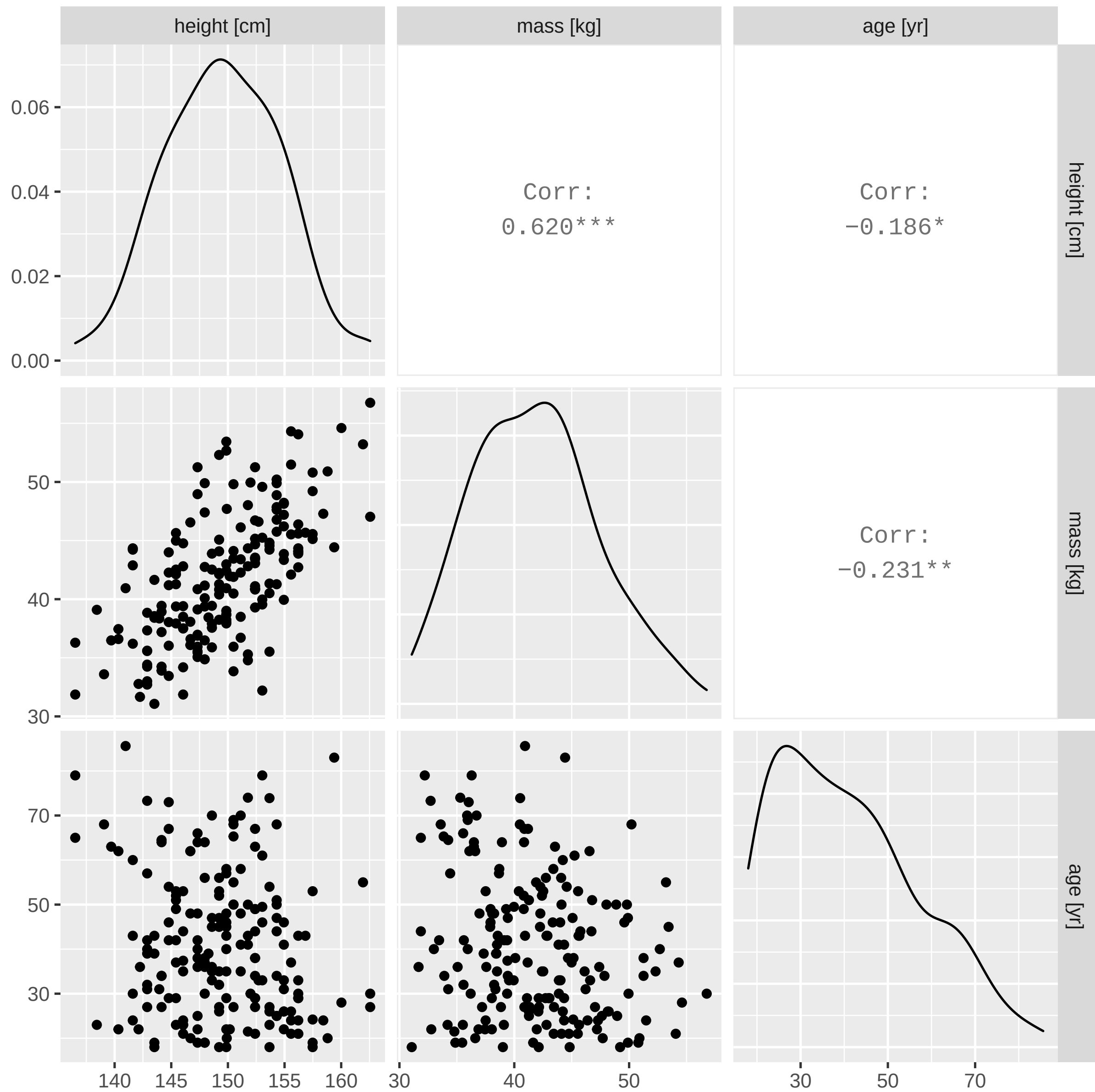
```
  ))
fig1
```



3.2 Visualisation of data in X via scatter plot matrix

In addition, the **trivariate data set** in X is now visualised by means of a **scatter plot matrix**, thus providing projections of the cloud of data points in the **3D scatter plot** onto 2D horizontal and vertical slices. For this purpose the function `ggpairs()` from the R package `GGally` is used.

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



Note that the plots of the univariate distributions of the three **variables** observed in the present sample, displayed along the diagonal of the **scatter plot matrix**, suggest qualitatively that the measured values for height [cm] and mass [kg] appear **approximately normally distributed**, whereas this property does not apply to age [yr].

3.3 Descriptive statistics for data in X

For the **measured values** of each of the three **variables** in the **data matrix** X, the following descriptive statistical measures are computed that characterise their respective observed univariate distributions:

1. **Mean** and **standard deviation**:

```r
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. **Standardised skewness** and **standardised excess kurtosis**; cf. Joanes and Gill (1998) [10] and van Elst (2019) [4]:

```
standSkewness(X[, 1:3])

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

standKurtosis(X[, 1:3])

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

As long as the computed values for *both* the standardised skewness and the standardised excess kurtosis range inside an interval with boundaries $Q_{0.025} = -1.96$ and $Q_{0.975} = +1.96$, corresponding to the central 95 $\%$ probability interval for a standard normally distributed variable, one may assume according to statistical convention that one is dealing with **approximately normally distributed** univariate metrically scaled data; cf. Hair *et al* (2010) [6]. As the results obtained for the **trivariate example data set** in $\boldsymbol{X}$ show, this property applies presently to the variables height [cm] and mass [kg], but *not* to the variable age [yr], the distribution of which classifies as right-skewed; cf. the scatter plot matrix discussed in Subsec. 3.2.

3. Counts of **outliers**, **extremal values** and **6-sigma-events**; cf. Toutenburg (2004) [16]:

```
outliers(X[, 1:3])

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

extremalValues(X[, 1:3])

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

sixSigmaEvents(X[, 1:3])

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

4 Standardisation of trivariate data set (matrix Z)

In a next step, the raw data in $\boldsymbol{X}$ needs to be transformed onto a common **dimensionless scale of measurement**, with repect to which **measured values** for univariate metrically scaled **variables** are expressed as **deviations from the mean in multiples of the standard deviation**. This kind of **transformation** is referred to as **standardisation**.

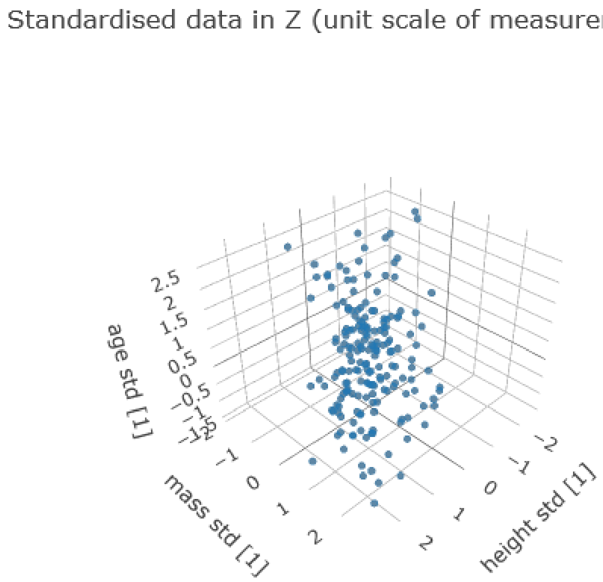
```
Z <-
  scale(x = X[, c("height [cm]", "mass [kg]", "age [yr]")],
        center = TRUE,
        scale = TRUE)
colnames(Z) <- c("height_std [1]", "mass_std [1]", "age_std [1]")
dim(Z)

## [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
```

In consequence of **standardisation**, the univariate data for each of the three **variables** in the resultant **data matrix Z** exhibit a **mean** of 0 and a **standard deviation** of 1. This property will be displayed shortly.

The **data matrix Z** of **standardised measured values** ("z-scores") constitutes the basis of all subsequent steps of **statistical data analysis**.

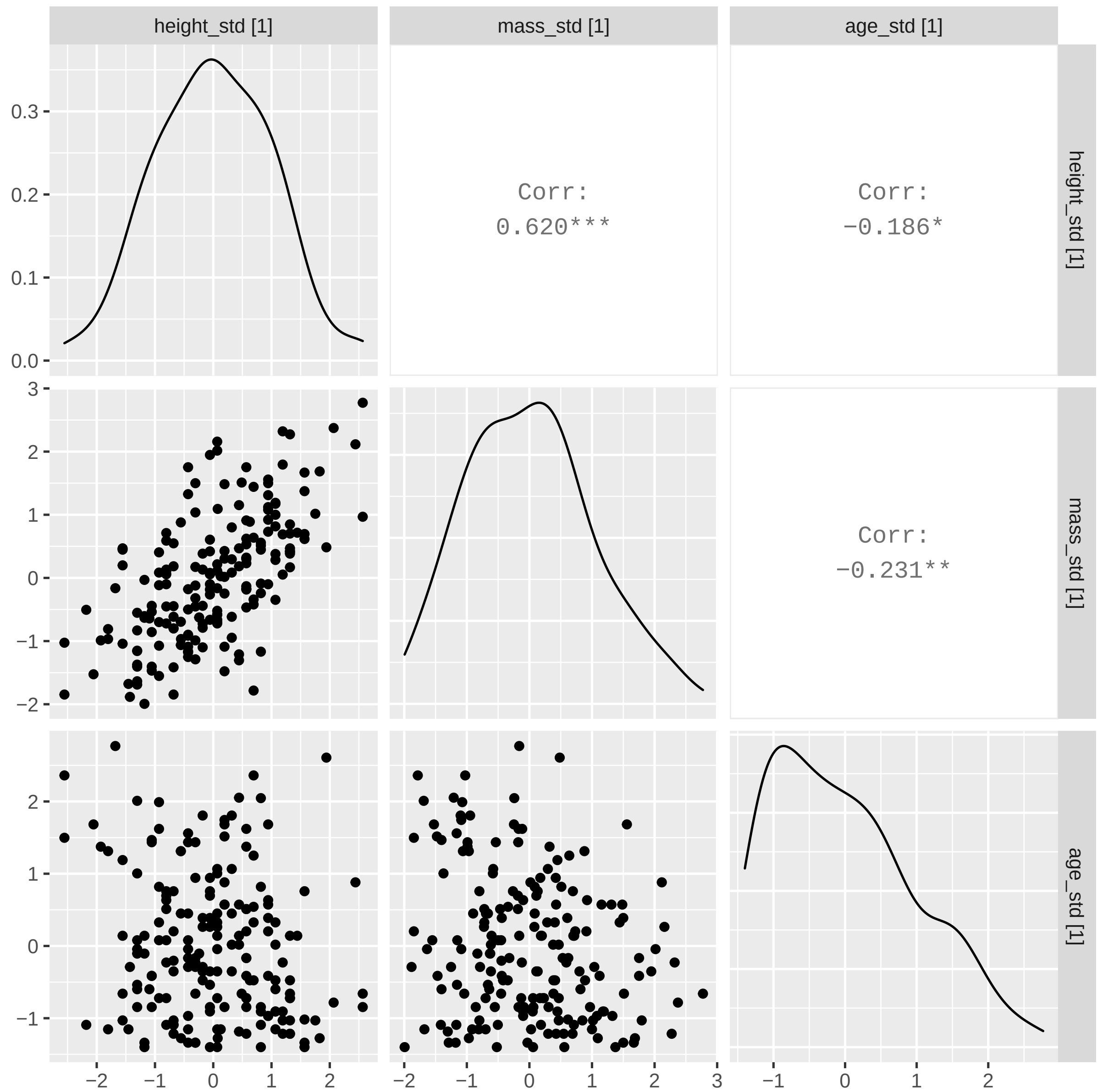
4.1 Visualisation of data in Z via 3D scatter plot

```
fig2 <- plot_ly(
  data = as.data.frame(Z),
  type = "scatter3d",
  x = Z[, 1],
  y = Z[, 2],
  z = Z[, 3],
  mode = "markers",
  size = 1
) %>%
  layout(
    title = "Standardised data in Z (unit scale of measurement)",
    scene = list(
      xaxis = list(title = "height std [1]"),
      yaxis = list(title = "mass std [1]"),
      zaxis = list(title = "age std [1]")
  ))
fig2
```



4.2 Visualisation of data in Z via scatter plot matrix

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



As the comparison of this **scatter plot matrix** with the one given in Subsec. 4.2 for the original **measured values** shows qualitatively, **standardisation** has preserved the observed uni- and bivariate (and trivariate) distributional properties of the **trivariate data set**. Otherwise, a **transformation** of this type would have to be rejected as an illegitimate procedure. The preservation of the distributional properties is demonstrated from the univariate perspective by the next step of **statistical data analysis**.

4.3 Descriptive statistics for data in Z

Again, descriptive statistical measures are computed that characterise observed univariate distributions, this time for the data in $\boldsymbol{Z}$ ("z-scores"):

1. **Mean** and **standard deviation**:

```
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. **Standardised skewness** and **standardised excess kurtosis**; cf. Joanes and Gill (1998) [10] and van Elst (2019) [4]:

```
standSkewness(Z)

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

standKurtosis(Z)

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

5 Sampling adequacy of trivariate data set in $\boldsymbol{Z}$ for principal component analysis

The **sampling adequacy** for a **principal component analysis** of the present **trivariate data set** is evaluated by employing Bartlett's (1951) [1] **test of sphericity**, as well as the standardised **KMO** and **MSA measures** according to Kaiser, Meyer und Olkin (KMO); cf. Kaiser (1970) [12], Guttman (1953) [5] and Hatzinger *et al* (2014) [7]. For this purpose, one may use the functions `bart_spher()` and `KMO()` from the R package `REdaS`.

Bartlett's (1951) [1] frequentist **null hypothesis significance test** subjects the assumption of sphericity of the **envelope** of the **cloud of data points** (defined via the matrix $\boldsymbol{Z}$) in, presently, Euclidian space $\mathbb{R}^3$ to an empirical check. For the given **trivariate data set** this yields

```
bart_spher(x = Z)

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

In view of the calculated p-value, the null hypothesis can be rejected at a significance level of $\alpha = 0.01$. Most likely the empirically attested deformation of the **envelope** is not due to chance. Non-sphericity of the **envelope** may thus be assumed and so supports the intention of performing a **principal component analysis** on the **trivariate data set** in $\boldsymbol{Z}$.[3]

The standardised **KMO** and **MSA measures**, which both take values in the interval $[0; 1]$, assume for the **trivariate data set** in $\boldsymbol{Z}$ the values

[3]The non-sphericity of the envelope of the cloud of data points defined via $\boldsymbol{Z}$ is conspicuous in the 3D scatter plot displayed in Subsec. 4.1.

```
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
```

The **KMO measure** quantifies the **sampling adequacy** of the entire **data set** in $\boldsymbol{Z}$, whereas, in contrast, the **MSA measure** individually quantifies the **sampling adequacy** of the **measured values** for every single **variable**. According to, e.g., Hatzinger *et al* (2014) [7] or Hair *et al* (2010) [6], a good **sampling adequacy** for the **data set** in $\boldsymbol{Z}$ is given when *both* standardised measures range between 0.8 und 1.0. Of course, in this respect, the presently considered **trivariate data set** in $\boldsymbol{Z}$ constitutes a *negative example*. However, as it proves very useful for demonstrating the main theoretical and practical steps of the **principal component analysis** of a metrically scaled **multivariate data set**, and is also accessible to the reader's imagination, this tutorial continuous to employ it in the steps of **statistical data analysis** taken in the following sections.

6 Calculation of sample correlation matrix $\boldsymbol{R}$ and its inverse $\boldsymbol{R}^{-1}$

The **sample correlation matrix** $\boldsymbol{R}$ of the considered **trivariate data set** in $\boldsymbol{X}$ is defined in terms of algebraic projections of "z-scores" onto themselves by

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

viz.

```
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
```

The **sample correlation matrix** $\boldsymbol{R}$ exhibits a non-zero value for its **determinant** and therefore classifies as regular. It follows that there exists an **inverse**, $\boldsymbol{R}^{-1}$, which is given here for completeness:

```
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
```

The **trace** of the **sample correlation matrix** $\boldsymbol{R}$ (presently) amounts to

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

## [1] 3
```

This value is equal to the number of **variables** in the considered **trivariate data set** in $\boldsymbol{X}$.

7 Eigenvalues and eigenvectors: orthonormal eigenbasis of sample correlation matrix

In the present case, the three **eigenvalues** of the **sample correlation matrix** $\boldsymbol{R}$ are

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

## [1] 1.7382412 0.8838105 0.3779482

sum(evAnaCor$values)

## [1] 3
```

and they sum up to the very number of **variables** in the **trivariate data set** in $\boldsymbol{X}$. The three mutually orthogonal, normalised **eigenvectors** of the **sample correlation matrix** $\boldsymbol{R}$ are (columnwise, from left to right)[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
```

[4]Regrettably R does not return the components of the three identified eigenvectors according to the usual mathematical convention, i.e., so that the eigenvectors form a right-handed oriented orthonormal basis.

These are also referred to as the **principal components** of the **trivariate data set** in $\boldsymbol{X}$. The **eigenvectors** span the **orthonormal eigenbasis** of the **sample correlation matrix** $\boldsymbol{R}$ in Euclidian space $\mathbb{R}^3$; cf. Bronstein *et al* (2005) [2].

With regard to Kaiser's (1960) [12] **eigenvalue criterion** it is noted that in the present example only one of the three **eigenvalues** of the **sample correlation matrix** $\boldsymbol{R}$ is greater than 1, implying that presently the **trivariate data set** in $\boldsymbol{X}$ possesses only a single **dominant principal component**.

The **proportions of total variance** of the **trivariate data set** in $\boldsymbol{Z}$ explained by each of the three **principal components** amount to

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

## [1] 0.5794 0.2946 0.1260
```

i.e., 57.94 %, 29.46 % and 12.60 %, and in cumulative terms

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

## [1] 0.5794 0.8740 1.0000
```

An interpretation for the **eigenvalues** of the **sample correlation matrix** $\boldsymbol{R}$ will be given in the next section.

8 Rotation matrix $\boldsymbol{V}$, diagonal eigenvalue matrix $\boldsymbol{\Lambda}$ and inverse diagonal eigenvalue matrix $\boldsymbol{\Lambda}^{-1}$

From the three eigenvectors of the **sample correlation matrix** $\boldsymbol{R}$ one constructs an orthogonal **rotation matrix** $\boldsymbol{V}$, by means of which one can perform (presently in Euclidian space $\mathbb{R}^3$) transformations to the *right-handed oriented* **orthonormal eigenbasis** of the **sample correlation matrix** $\boldsymbol{R}$. The **determinant** of the **rotation matrix** $\boldsymbol{V}$ has the value 1, i.e.,

```
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
```

implying that transformations with the **rotation matrix** $\boldsymbol{V}$ *preserve volumes*.

Per construction the **rotation matrix** $\boldsymbol{V}$ satisfies the following two **tests of orthogonality**

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

viz.

```
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
```

By means of **diagonalisation** of the **sample correlation matrix** $\boldsymbol{R}$ via transformation with the **rotation matrix** $\boldsymbol{V}$, one obtains the **diagonal eigenvalue matrix** $\boldsymbol{\Lambda}$ as

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

viz.

```
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
```

The **diagonal eigenvalue matrix** $\boldsymbol{\Lambda}$ is nothing but the representation of the **sample correlation matrix** $\boldsymbol{R}$ with respect to its **orthonormal eigenbasis** in Euclidian space $\mathbb{R}^3$.

The **diagonal eigenvalue matrix** $\boldsymbol{\Lambda}$ obtained above may be subjected to the **consistency check**

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

viz.

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

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

The **inverse**, $\boldsymbol{\Lambda}^{-1}$, of the **diagonal eigenvalue matrix** $\boldsymbol{\Lambda}$ is computed by

```
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
```

The three **eigenvalues** of the **sample correlation matrix** $\boldsymbol{R}$ amount to the **variances** of the data in $\boldsymbol{Z}$ along the three directions in Euclidian space $\mathbb{R}^3$ defined by the **orthonormal eigenbasis** of the **sample correlation matrix** $\boldsymbol{R}$. The following consideration makes this fact explicit. **Transformation** of the data in $\boldsymbol{Z}$ to the **orthonormal eigenbasis** of the **sample correlation matrix** $\boldsymbol{R}$ yields

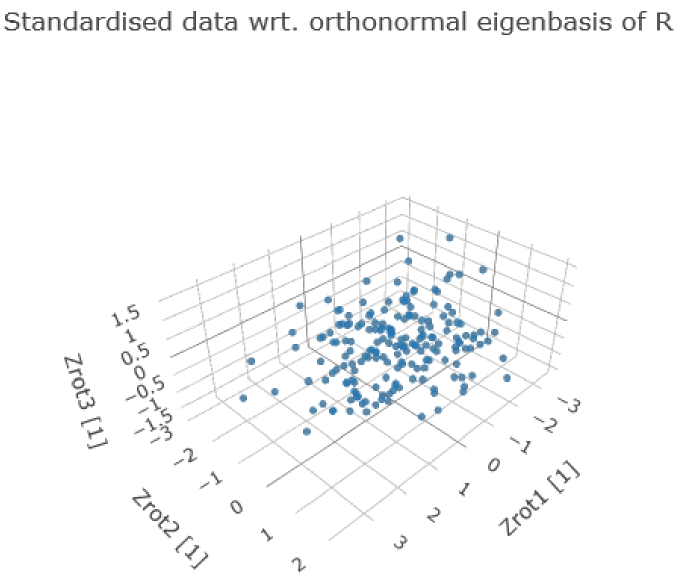
$$\boldsymbol{Z}_{\text{rot}} = \boldsymbol{Z}\boldsymbol{V}$$

viz.

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

Visualisation of the resultant data in $\boldsymbol{Z}_{\text{rot}}$ by means of a **3D scatter plot** gives

```
fig3 <- plot_ly(
  data = as.data.frame(Zrot),
  type = "scatter3d",
  x = Zrot[, 1],
  y = Zrot[, 2],
  z = Zrot[, 3],
  mode = "markers",
  size = 1
) %>%
  layout(
    title = "Standardised data wrt. orthonormal eigenbasis of R",
    scene = list(
      xaxis = list(title = "Zrot1 [1]"),
      yaxis = list(title = "Zrot2 [1]"),
      zaxis = list(title = "Zrot3 [1]")
  ))
fig3
```



Subsequent computation of the **variances** of the data in $\boldsymbol{Z}_{\text{rot}}$ obtains

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

## [1] 1.7382412 0.8838105 0.3779482
```

i.e., the claimed result. Note that the data in the columns of $\boldsymbol{Z}_{\mathrm{rot}}$ is pairwise *uncorrelated*

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

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

9 Principal component loadings matrix $\boldsymbol{A}$

The **principal component loadings matrix** $\boldsymbol{A}$, defined in terms of the orthogonal **rotation matrix** $\boldsymbol{V}$ and the **diagonal eigenvalue matrix** $\boldsymbol{\Lambda}$ by

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

provides the answer to the question: how strong are the (presently) three **original variables** height, mass and age correlated with the identified three **principal components** of the **trivariate data set** considered?[5]

```
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
```

The fact that the **principal component loadings matrix** $\boldsymbol{A}$ may indeed be interpreted as a (formal) correlation matrix will become apparent later on.

The **principal component loadings matrix** $\boldsymbol{A}$ satisfies two **consistency checks**:

1. The **sample correlation matrix** $\boldsymbol{R}$ can be factorised by means of the **principal component loadings matrix** $\boldsymbol{A}$ as

$$\boldsymbol{0} = \boldsymbol{R} - \boldsymbol{A}\boldsymbol{A}^{\top}$$

viz.

```
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
```

[5]Generally: how strong are the m original variables of a given metrically scaled multivariate data set correlated with the identified m principal components of this data set?

2. The **diagonal eigenvalue matrix** $\boldsymbol{\Lambda}$ can be factorised by means of the **principal component loadings matrix** $\boldsymbol{A}$ as
$$\mathbf{0} = \boldsymbol{\Lambda} - \boldsymbol{A}^{\top}\boldsymbol{A}$$
viz.

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

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

10 Standardised data set in orthonormal eigenbasis of sample correlation matrix (matrix $\boldsymbol{F}$)

Finally, a **transformation** needs to be performed of the standardised **trivariate data set** in $\boldsymbol{Z}$ to the **orthonormal eigenbasis** of the **sample correlation matrix** $\boldsymbol{R}$, while respecting the conventional requirement that the resultant data shall likewise be standardised. This is realised in terms of a volume preserving **rotation** of the original reference frame with the **rotation matrix** $\boldsymbol{V}$,[6] followed by a volume changing but directions preserving **rescaling** of the axes of the rotated reference frame with the (square root of the) **inverse** $\boldsymbol{\Lambda}^{-1}$ of the **diagonal eigenvalue matrix** $\boldsymbol{\Lambda}$. These two **transformations** performed in combination define the matrix $\boldsymbol{F}$,[7]
$$\boldsymbol{F} := \boldsymbol{Z}\boldsymbol{V}\boldsymbol{\Lambda}^{-1/2}$$
viz.

```
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
```

which contains standardised and, by construction, mutually *uncorrelated* so-called "f-scores" in its columns.

[6] The effect of a pure transformation of the data in $\boldsymbol{Z}$ with the rotation matrix $\boldsymbol{V}$ was described and visualised before in Sec. 8.

[7] Equivalently, the matrix $\boldsymbol{F}$ can also be computed from the trivariate data set in $\boldsymbol{Z}$ by employing the principal component loadings matrix $\boldsymbol{A}$ and the inverse $\boldsymbol{\Lambda}^{-1}$ as $\boldsymbol{F} = \boldsymbol{Z}\boldsymbol{A}\boldsymbol{\Lambda}^{-1}$.

10.1 Consistency checks for matrix $\boldsymbol{F}$

The following **consistency checks** need to be satisfied by the matrix $\boldsymbol{F}$:

1. The "f-scores" are standardised and mutually uncorrelated:

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

viz.

```
proxy <- (1 / (nrow(FmatCor) - 1)) * t(FmatCor) %*% FmatCor
round(diag(rep(1, nrow(proxy))) - proxy, 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. The elements of the **principal component loadings matrix** $\boldsymbol{A}$ represent, as algebraic projections of standardised and uncorrelated "z-scores" onto standardised and uncorrelated "f-scores", bivariate correlations between the (presently three) **original variables** and the (presently three) **principal components** of the **multivariate data set** considered; cf. the remarks at the beginning of Sec. 9:

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

viz.

```
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. The "z-scores" may be perceived as linear combinations of the "f-scores":

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

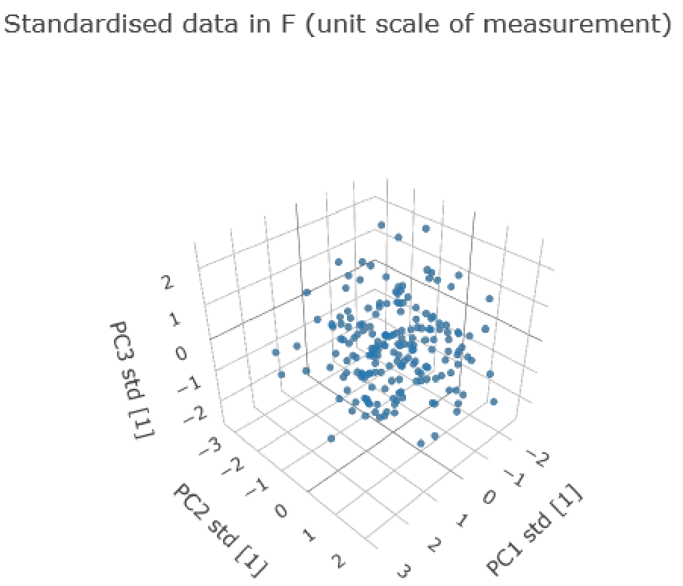
viz.

```
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
```

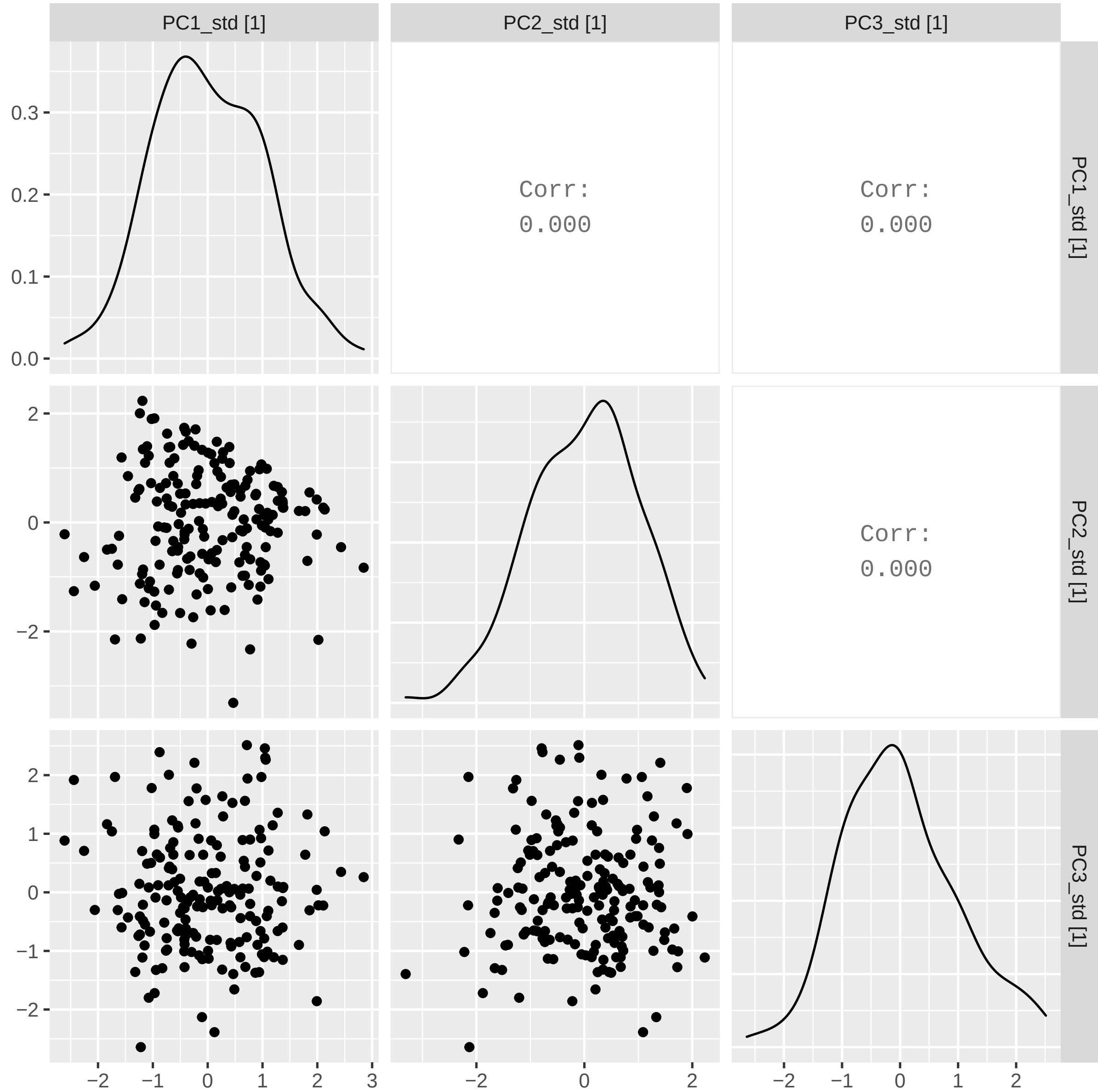
10.2 Visualisation of data in *F* via 3D scatter plot

```
fig4 <- plot_ly(
  data = as.data.frame(FmatCor),
  type = "scatter3d",
  x = FmatCor[, 1],
  y = FmatCor[ ,2],
  z = FmatCor[, 3],
  mode = "markers",
  size = 1
) %>%
  layout(
    title = "Standardised data in F (unit scale of measurement)",
    scene = list(
      xaxis = list(title = "PC1 std [1]"),
      yaxis = list(title = "PC2 std [1]"),
      zaxis = list(title = "PC3 std [1]")
  ))
fig4
```



10.3 Visualisation of data in *F* via scatter plot matrix

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



Die **scatter plot matrix** verdeutlicht ebenfalls das paarweise Nichtkorreliertsein der den einzelnen **principal components** zugeordneten "f-Werten".

11 Dimensional reduction: extraction of single dominant principal component

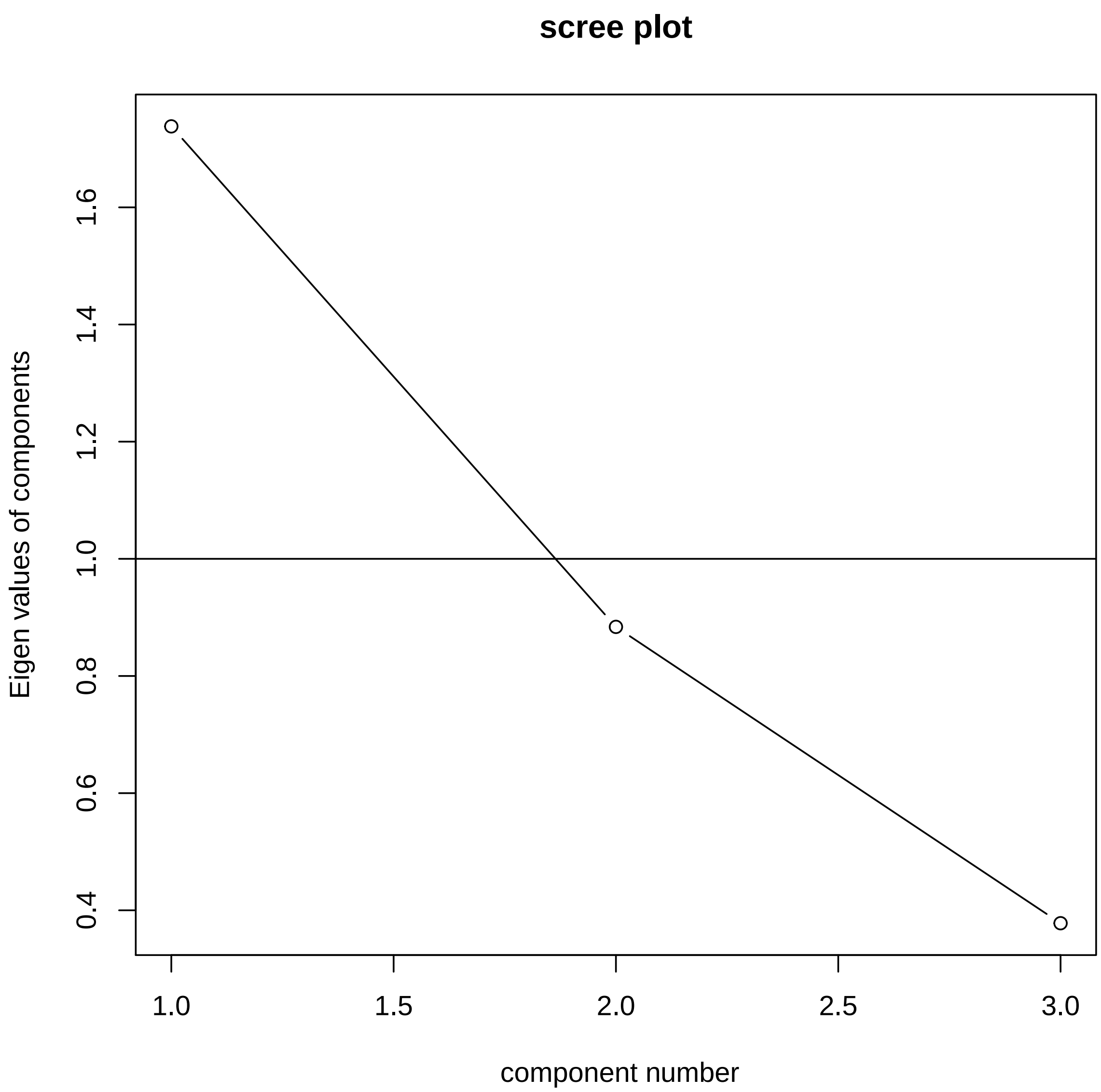
Secs. 4 to 10 featured in some detail the linear-algebraic methodology on which the **principal component analysis** of a metrically scaled **multivariate data set** is grounded. The discussion now turns to describe the necessary steps involved in a **dimensional reduction** of such a **data set**, given that for the case considered the procedure proves conceptually and practically meaningful.

The **dimensional reduction** of the **multivariate data set** begins in the **orthonormal eigenbasis** of the **sample correlation matrix** $\boldsymbol{R}$. In short, one only keeps those "f-scores" which are associated with the **dominant principal components**; the remaining ones are being discarded. The loss of information incurred in this way is generally perceived as being compensated by an overall reduction of complexity for the **multivariate data set** being analysed. The remaining "f-scores" will be transformed back to the **original orthonormal basis** and then expressed with respect to the **original scales of measurement**. The **dimensionally reduced measured values** so obtained may form the starting point of further **statistical data analysis**.[8]

11.1 Qualitative criterion for extraction

The R package `psych` provides the function `VSS.scree()` for generating **scree plots** according to Cattell (1966) [3].

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



Based on numerous applications in practice, it is recommended to extract as many **principal components** fom a given **multivariate data set** as correspond to the number of **eigenvalues** to be found *left* of the

[8]The performance of a dimensional reduction of a given multivariate data set, when meaningful, does not really require computation of the matrix $\boldsymbol{F}$, respectively the "f-scores". For this purpose it fully suffices to transform the "z-scores" from the original orthonormal basis to the orthonormal eigenbasis of the sample correlation matrix $\boldsymbol{R}$ and then keep only those values (columns) associated with the dominant principal components. Standardisation of values in the orthonormal eigenbasis merely corresponds to a convenient convention.

"ellbow" in a **scree plot**. In the present example this is just one, and thus proves consistent with Kaiser's (1960) [12] **eigenvalue criterion**, which recommends **extraction** of all **principal components** that are associated with an **eigenvalue** greater than 1.

The procedure of a **dimensional reduction** of a given **multivariate data set** is now exemplified for the **trivariate data set** in $\boldsymbol{X}$. It is demonstrated on the basis of **extraction** of the single **dominant principal component** identified in Sec. 7.

11.2 Dimensionally reduced matrixes

The procedure of a **dimensional reduction** of a given multivariate data set is reflected in the first place in the matrixes of the aforementioned linear-algebraic methodology.

1. Dimensionally reduced rotation matrix $\boldsymbol{V}_{\text{red}}$ — only those eigenvectors of a sample correlation matrix $\boldsymbol{R}$ that are associated with eigenvalues greater than 1 will be employed in the construction of a rotation matrix:

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

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

2. Dimensionally reduced diagonal eigenvalue matrix $\boldsymbol{\Lambda}_{\text{red}}$ — generated from the sample correlation matrix $\boldsymbol{R}$ via transformation with the dimensionally reduced rotation matrix:

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

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

3. Inverse of dimensionally reduced diagonal eigenvalue matrix, $\boldsymbol{\Lambda}_{\text{red}}^{-1}$

```
LambdaCorRedInv <- solve(LambdaCorRed)
LambdaCorRedInv

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

4. Dimensionally reduced principal component loadings matrix $\boldsymbol{A}_{\text{red}}$

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

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

5. Dimensionally reduced matrix $\boldsymbol{F}$, $\boldsymbol{F}_{\text{red}}$

```
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 Comparison of trivariate example data set with its dimensionally reduced variant

The "f-scores" remaining in the dimensionally reduced matrix $\boldsymbol{F}_{\text{red}}$ need to be transformed back to the **original orthonormal basis**, and then expressed with respect to the **original scales of measurement**. For illustrative purposes, a sample of the corresponding dimensionally reduced data will be contrasted with the original "z-scores," and with the original measured values for the three **variables** height, mass and age.

Standardised scale of measurement

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

Original scales of measurement

This requires a **backward transformation** (de-standardisation) of the data in $\boldsymbol{Z}_{\text{red}}$ respectively in $\boldsymbol{Z}$ to the **original scales of measurement** used for the three variables height, mass and age:

```
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])
```

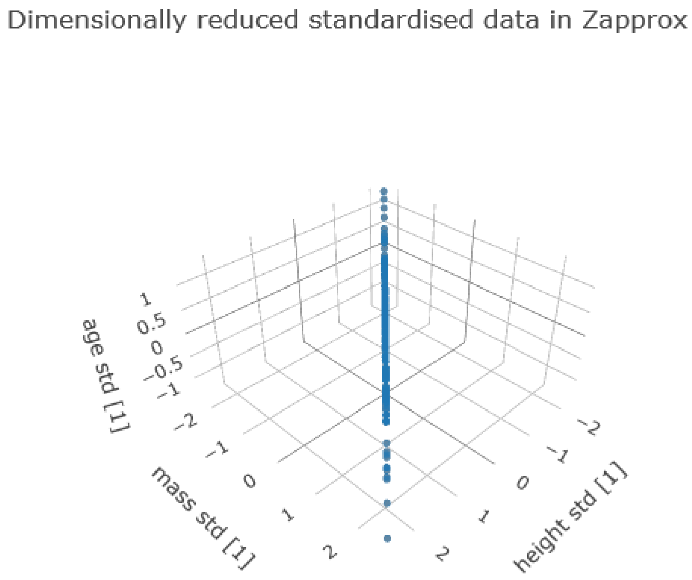
```
##   height [cm] mass [kg] age [yr]
## 1     139.700  36.48581       63
## 2     136.525  31.86484       65
## 3     145.415  41.27687       51
## 4     149.225  38.24348       32
## 5     147.955  34.86988       19
## 6     154.305  49.89512       47
```

```
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 Visualisation of dimensionally reduced data via 3D scatter plot

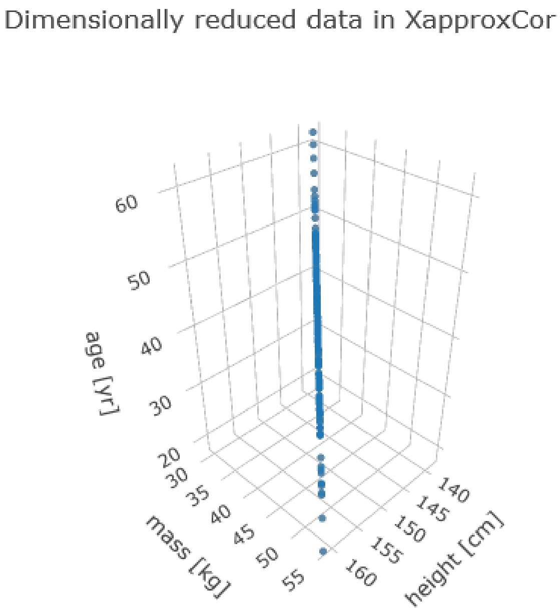
Standardised scale of measurement

```
fig5 <- plot_ly(
  data = as.data.frame(Zapprox),
  type = "scatter3d",
  x = Zapprox[, 1],
  y = Zapprox[, 2],
  z = Zapprox[, 3],
  mode = "markers",
  size = 1
) %>%
  layout(
    title = "Dimensionally reduced standardised data in Zapprox",
    scene = list(
      xaxis = list(title = "height std [1]"),
      yaxis = list(title = "mass std [1]"),
      zaxis = list(title = "age std [1]")
  ))
fig5
```



Original scales of measurement

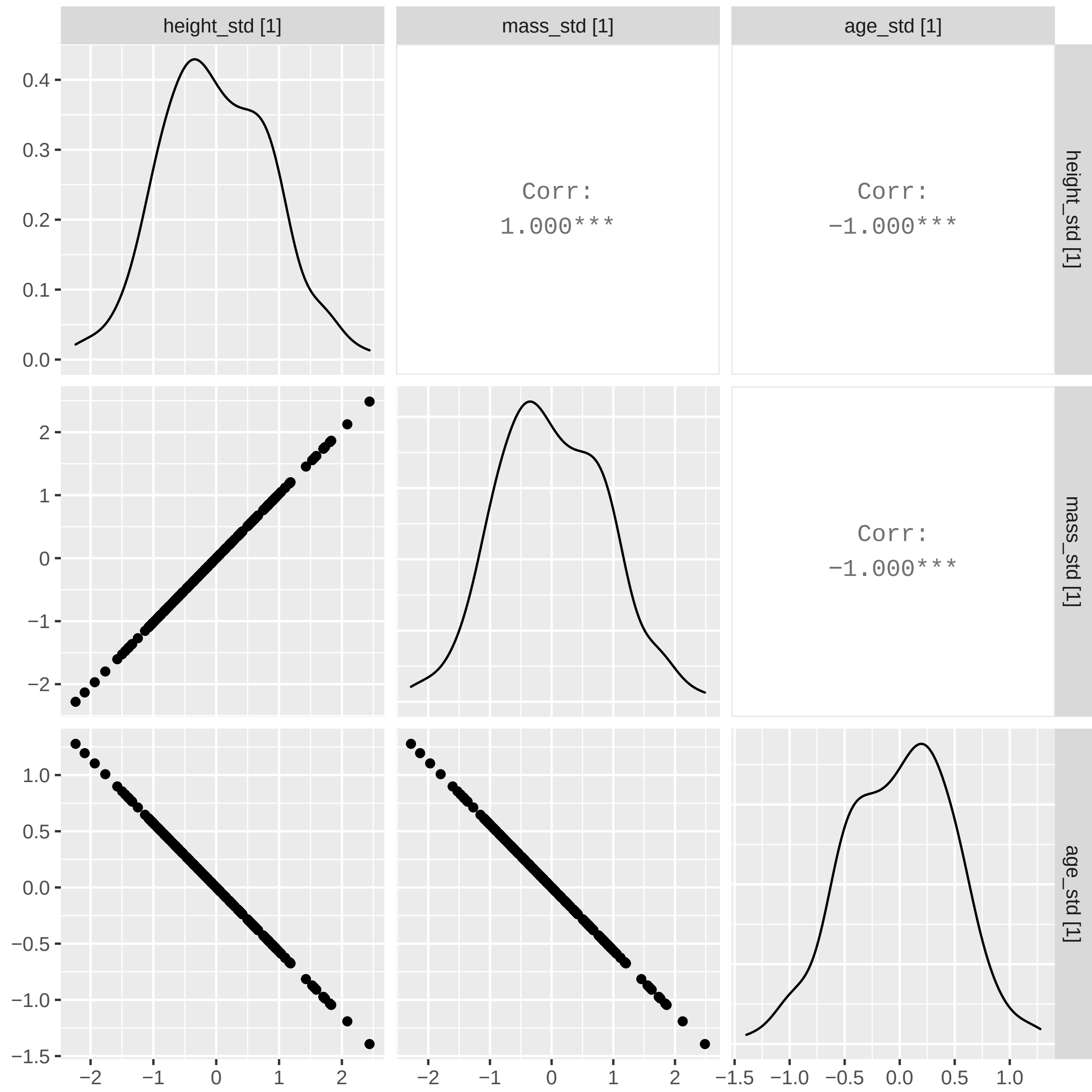
```
fig6 <- plot_ly(
  data = as.data.frame(XapproxCor),
  type = "scatter3d",
  x = XapproxCor[, 1],
  y = XapproxCor[, 2],
  z = XapproxCor[, 3],
  mode = "markers",
  size = 1
) %>%
  layout(
    title = "Dimensionally reduced data in XapproxCor",
    scene = list(
      xaxis = list(title = "height [cm]"),
      yaxis = list(title = "mass [kg]"),
      zaxis = list(title = "age [yr]")
  ))
fig6
```



11.5 Visualisation of dimensionally reduced data via scatter plot matrix

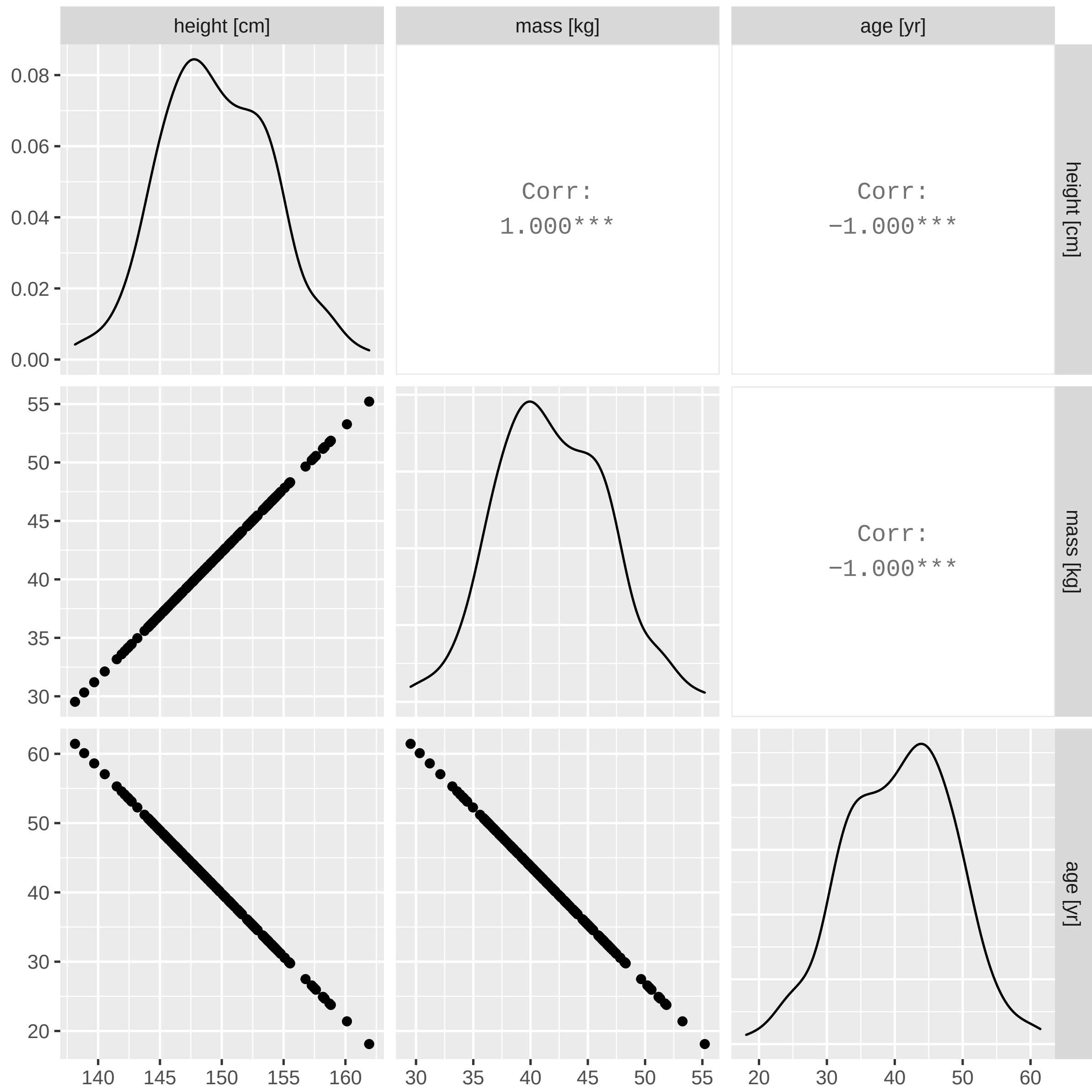
Standardised scale of measurement

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



Original scales of measurement

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



12 Conclusion

In the example considered, the **dimensional reduction** performed results in an **extremal case**: the **trivariate data set** in $\boldsymbol{X}$ was effectively reduced to a univariate data set, which can explain $57.94\ \%$ of the total variance of the original data set. The dimensionally reduced data set exhibits the maximal (minimal) possible values for the bivariate correlations between any pair of the three original variables height, mass and age.

Acknowledgements

Constructive comments by Jana Orthey and Laurens van der Woude have helped to focus this tutorial on the specific needs of the targeted audience.

References

[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 and H Mühlig (2005) *Taschenbuch der Mathematik* 6th Edition (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 and M J Maier (2014) *R — Einführung durch angewandte Statistik* 2nd Edition (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* 4th Edition (Berlin: Springer) ISBN–10: 3540222332