Section 10.1 | Exploratory Multivariate Analysis -PCA

Mohammad Saqib Ansari

2023-12-21

Exploratory Multivariate Analysis (EMA) is a statistical approach used to analyze and understand relationships between multiple variables simultaneously. In R, there are several packages like `stats`, `ggplot2`, `caret`, and `FactoMineR`, among others, that facilitate multivariate analysis.

Here is a theoretical explanation of exploratory multivariate analysis:

Exploratory Multivariate Analysis

What is Multivariate Analysis?

Multivariate analysis involves examining data that consists of observations on multiple variables simultaneously. Unlike univariate (single variable) or bivariate (two variables) analyses, multivariate analysis deals with the interrelationships among several variables.

Purpose of Exploratory Multivariate Analysis

The primary goal of exploratory multivariate analysis is to:

1. **Discover Patterns:** Identify underlying patterns, structures, or relationships within complex datasets.
2. **Reduce Dimensionality:** Reduce the number of variables while preserving important information.
3. **Visualize Relationships:** Use visualizations and statistical techniques to understand the connections between variables.
4. **Identify Outliers:** Detect outliers or anomalies that might impact the analysis.

Techniques in Exploratory Multivariate Analysis

1. **Principal Component Analysis (PCA):** PCA is used for dimensionality reduction. It transforms original variables into a smaller set of uncorrelated variables called principal components, capturing most of the variation in the data.

2. **Factor Analysis:** Similar to PCA, factor analysis explores underlying factors that explain observed correlations among variables. It helps identify latent variables.

3. **Cluster Analysis:** This technique groups similar observations into clusters based on their characteristics or distances between data points.

4. **Multidimensional Scaling (MDS):** MDS visualizes the level of similarity or dissimilarity between objects or cases by representing them in a lower-dimensional space.

5. **Correspondence Analysis:** It visualizes relationships among categorical variables, displaying them in a low-dimensional space.

R for Exploratory Multivariate Analysis

In R, several packages provide functionalities for multivariate analysis:

- `stats`: Basic statistical functions and models for multivariate analysis.
- `ggplot2`: Offers versatile and powerful visualization capabilities.
- `caret`: Assists in training and evaluating machine learning models, often used in multivariate analysis.
- `FactoMineR`, `cluster`, `ade4`: Specialized packages for factor analysis, clustering, and multivariate analysis.

Exploratory multivariate analysis is a crucial step in understanding complex datasets, revealing hidden structures, and aiding in decision-making processes. R provides a wide array of tools and packages to perform various multivariate analyses and visualize relationships among multiple variables.

When conducting exploratory multivariate analysis, it's important to select appropriate techniques based on the nature of the data and the specific objectives of the analysis.

Principal Component Analysis (PCA)

Theory

Principal Component Analysis (PCA) is a dimensionality reduction technique that transforms high-dimensional data into a new coordinate system. It identifies a set of linearly uncorrelated variables known as principal components, capturing the maximum variance within the data.

Assumptions

1. **Linearity:** PCA assumes linear relationships between variables.
2. **Statistical Independence:** Variables should ideally be statistically independent, although it can still be effective in identifying patterns even if this assumption is not entirely met.
3. **Homoscedasticity:** Variables should have roughly equal variance.

Applications

1. **Dimensionality Reduction:** PCA reduces the number of variables while preserving most of the information, aiding in data simplification and visualization.
2. **Feature Extraction:** Extracting meaningful features or components from high-dimensional data.
3. **Pattern Recognition:** Identifying patterns or structures within complex datasets.
4. **Noise Reduction:** PCA can filter out noise or irrelevant information, focusing on the most significant aspects of the data.
5. **Data Visualization:** Visualizing high-dimensional data in a lower-dimensional space for easier interpretation.

Disadvantages

1. **Linearity Assumption:** If relationships among variables are highly nonlinear, PCA might not effectively capture important patterns.
2. **Variance Explained:** Interpreting the variance explained by each principal component might be challenging, requiring careful consideration.
3. **Interpretability:** Extracted components might not be easily interpretable in terms of original variables, making it complex for non-experts.

4. **Information Loss:** Dimensionality reduction can lead to information loss, impacting predictive modeling or analysis if critical information is discarded.
5. **Sensitive to Outliers:** PCA can be sensitive to outliers, potentially affecting the results of the analysis.

PCA is a widely-used technique for dimensionality reduction and identifying patterns within high-dimensional data. While it offers significant advantages, careful attention to assumptions, interpretation challenges, and potential limitations is necessary for its effective application.

Step 1 | Reading The data

```
# Install the 'FactoMineR' package if not installed
#install.packages("FactoMineR")
# Load the necessary library
library(FactoMineR)

# Read the CSV file 'decathalon.csv' into the 'datd' #dataframe

datd <- read.csv("decathlon.csv")

# Check the dimensions (rows and columns) of the dataframe
dim(datd)
```

```
## [1] 41 14
```

```
# Obtain a summary of the dataset, showing basic statistics #for each variable
summary(datd)
```

```
##    Athlets              X100m         Long.jump
##  Length:41          Min.   :10.44   Min.   :6.61
##  Class :character   1st Qu.:10.85   1st Qu.:7.03
##  Mode  :character   Median :10.98   Median :7.30
##                     Mean   :11.00   Mean   :7.26
##                     3rd Qu.:11.14   3rd Qu.:7.48
##                     Max.   :11.64   Max.   :7.96
##     Shot.put       High.jump         X400m
##  Min.   :12.68   Min.   :1.850   Min.   :46.81
##  1st Qu.:13.88   1st Qu.:1.920   1st Qu.:48.93
##  Median :14.57   Median :1.950   Median :49.40
##  Mean   :14.48   Mean   :1.977   Mean   :49.62
##  3rd Qu.:14.97   3rd Qu.:2.040   3rd Qu.:50.30
##  Max.   :16.36   Max.   :2.150   Max.   :53.20
##   X110m.hurdle       Discus        Pole.vault
##  Min.   :13.97   Min.   :37.92   Min.   :4.200
##  1st Qu.:14.21   1st Qu.:41.90   1st Qu.:4.500
##  Median :14.48   Median :44.41   Median :4.800
##  Mean   :14.61   Mean   :44.33   Mean   :4.762
##  3rd Qu.:14.98   3rd Qu.:46.07   3rd Qu.:4.920
##  Max.   :15.67   Max.   :51.65   Max.   :5.400
##     Javeline         X1500m           Rank
##  Min.   :50.31   Min.   :262.1   Min.   : 1.00
```

```
##  1st Qu.:55.27   1st Qu.:271.0   1st Qu.: 6.00
##  Median :58.36   Median :278.1   Median :11.00
##  Mean   :58.32   Mean   :279.0   Mean   :12.12
##  3rd Qu.:60.89   3rd Qu.:285.1   3rd Qu.:18.00
##  Max.   :70.52   Max.   :317.0   Max.   :28.00
##      Points     Competition
##  Min.   :7313   Length:41
##  1st Qu.:7802   Class :character
##  Median :8021   Mode  :character
##  Mean   :8005
##  3rd Qu.:8122
##  Max.   :8893
```

In this step, the code loads the 'FactoMineR' library, reads the CSV file named 'decathalon.csv' into the 'datd' dataframe, and provides an overview of the dimensions (number of rows and columns) of the dataset, as well as a summary of its variables.

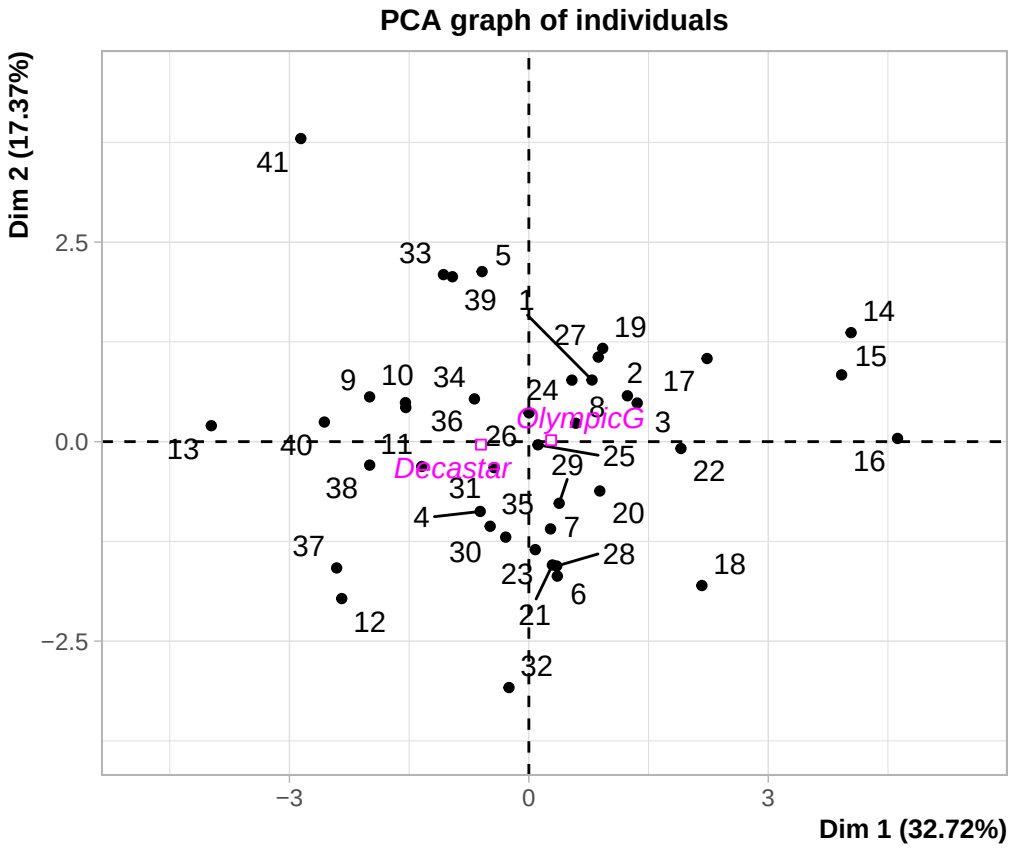
Step 2 | Choosing the active individual or variable

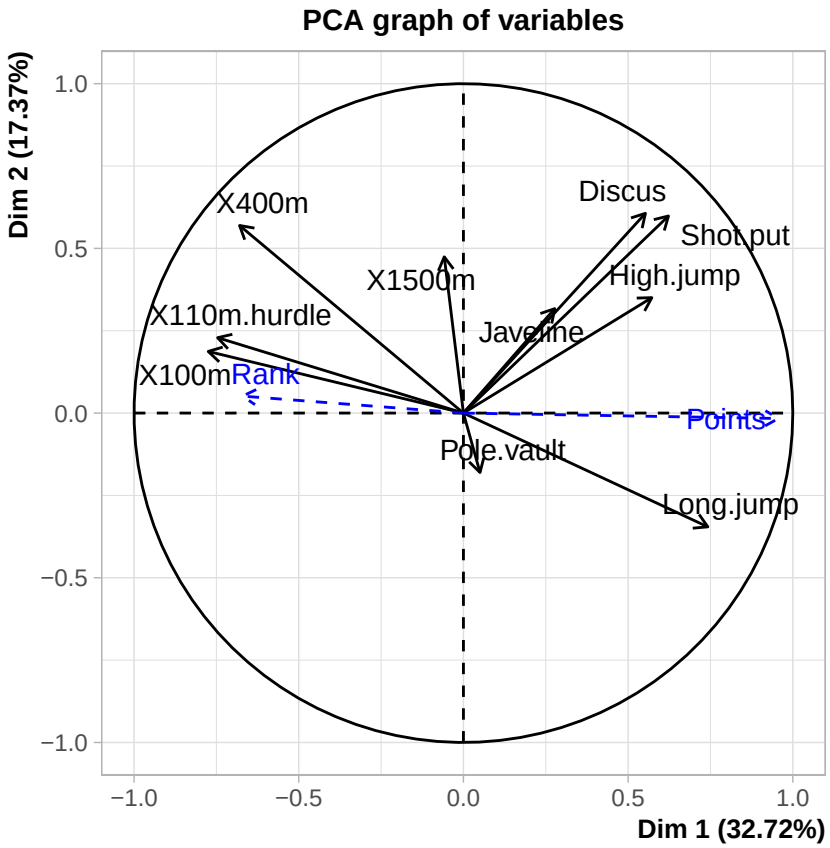
This step is not explicitly coded in the provided section. Typically, in PCA, the choice of the active individuals (rows) or variables (columns) is crucial for the analysis.

Step 3 | Choosing if the variable needs to be standardized

```r
# Remove non-quantitative variable (assuming it's the first column)
datd_numeric <- datd[, -1]  # This removes the first column assuming it's 'Athlets'

# Now perform PCA on the modified dataset
# 'quanti.sup' specifies quantitative supplementary variables, and 'quali.sup' specifies qualitative su
res.pca <- PCA(datd_numeric, quanti.sup = 11:12, quali.sup = 13)
```


PCA graph of individuals
Dim 2 (17.37%)
Dim 1 (32.72%)
2.5
0.0
−2.5
−3
0
3
41
33
5
39
1
27
19
14
15
9
10
34
24
2
17
8
OlympicG
3
36
26
40
11
13
Decastar
29
25
22
16
38
31
35
4
7
20
37
30
28
18
23
6
12
21
32



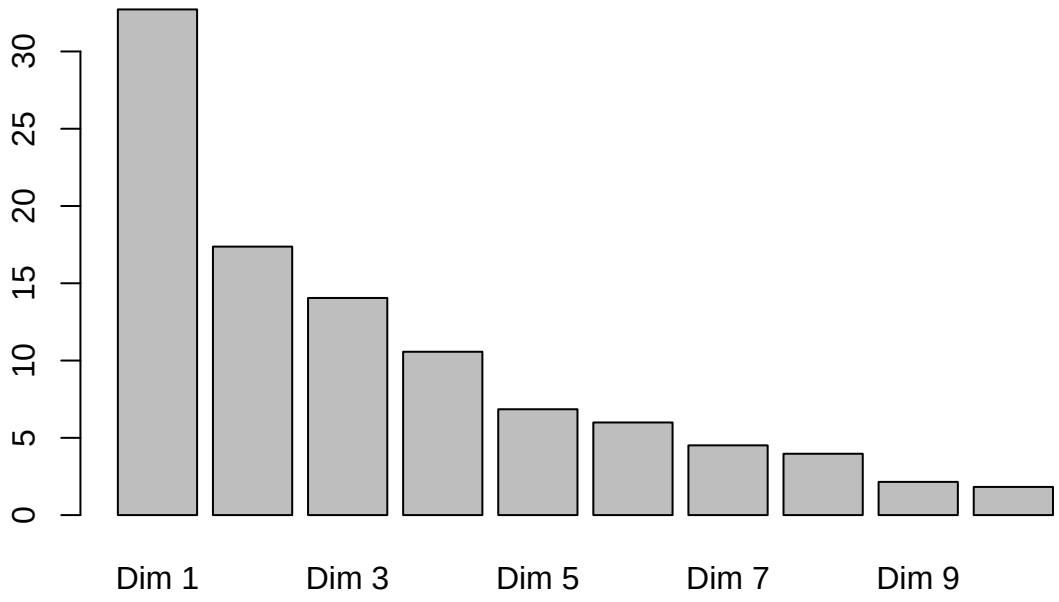

```
# Obtain the names of the results obtained from PCA
names(res.pca)
```

```
## [1] "eig"        "var"        "ind"        "svd"
## [5] "quanti.sup" "quali.sup"  "call"
```

This section of code installs and loads the 'FactoMineR' package if not already installed, removes the assumed non-quantitative variable (assuming it's the first column named 'Athlets'), and performs PCA on the modified dataset. Additionally, it specifies quantitative ('quanti.sup') and qualitative ('quali.sup') supplementary variables (columns 11, 12, and 13 in this case) to be considered during the analysis. Finally, it displays the names of the results obtained from PCA.

Step 4 | Choosing The Number of Dimensions

```
# Bar plot to visualize the eigenvalues
barplot(res.pca$eig[,2], names = paste("Dim", 1:nrow(res.pca$eig)))
```




```
# Display the first four eigenvalues
round(res.pca$eig[1:4, ], 2)
```

```
##        eigenvalue percentage of variance
## comp 1       3.27                 32.72
## comp 2       1.74                 17.37
## comp 3       1.40                 14.05
## comp 4       1.06                 10.57
##        cumulative percentage of variance
## comp 1                             32.72
## comp 2                             50.09
## comp 3                             64.14
## comp 4                             74.71
```

In this step, a bar plot is created to visualize the eigenvalues obtained from the PCA analysis using `res.pca$eig[,2]`. Additionally, the first four eigenvalues are displayed, helping to determine the number of significant dimensions in the dataset.

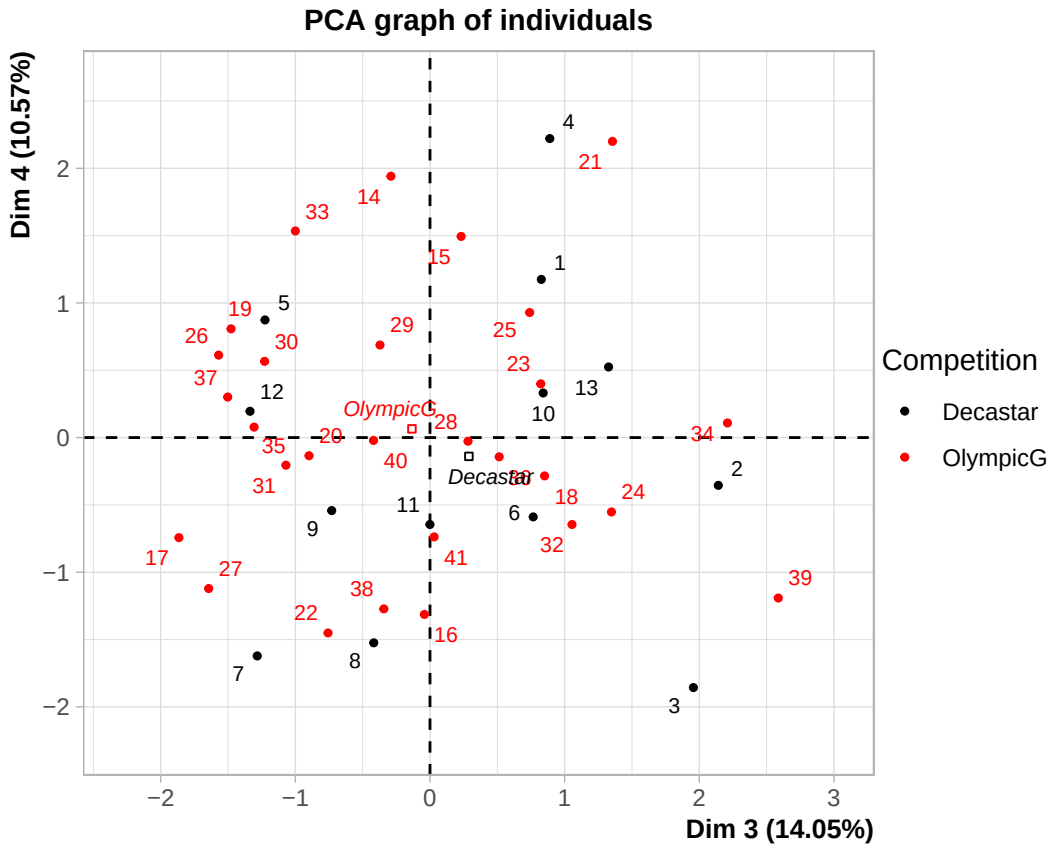
Step 5 | Analyzing the Result

```
# Displaying coordinates, squared cosines, and contributions for the first three individuals
round(cbind(res.pca$ind$coord[,1:3], res.pca$ind$cos2[,1:3],
            res.pca$ind$contrib[,1:3]), digits=2)
```

```
##    Dim.1 Dim.2 Dim.3 Dim.1 Dim.2 Dim.3 Dim.1 Dim.2 Dim.3
## 1   0.79  0.77  0.83  0.11  0.11  0.12  0.47  0.84  1.19
## 2   1.23  0.57  2.14  0.12  0.03  0.37  1.14  0.46  7.96
## 3   1.36  0.48  1.96  0.16  0.02  0.33  1.38  0.33  6.64
## 4  -0.61 -0.87  0.89  0.05  0.10  0.10  0.28  1.07  1.37
## 5  -0.59  2.13 -1.23  0.04  0.50  0.16  0.26  6.38  2.61
## 6   0.36 -1.68  0.77  0.02  0.48  0.10  0.09  3.99  1.02
## 7   0.27 -1.09 -1.28  0.01  0.18  0.25  0.06  1.68  2.86
## 8   0.59  0.23 -0.42  0.05  0.01  0.03  0.26  0.07  0.30
## 9  -2.00  0.56 -0.73  0.28  0.02  0.04  2.97  0.44  0.93
## 10 -1.55  0.49  0.84  0.31  0.03  0.09  1.78  0.33  1.23
## 11 -1.34 -0.31  0.00  0.47  0.03  0.00  1.34  0.14  0.00
## 12 -2.34 -1.97 -1.34  0.39  0.28  0.13  4.10  5.43  3.10
## 13 -3.98  0.20  1.33  0.86  0.00  0.10 11.80  0.06  3.05
## 14  4.04  1.37 -0.29  0.70  0.08  0.00 12.16  2.62  0.15
## 15  3.92  0.84  0.23  0.71  0.03  0.00 11.45  0.98  0.09
## 16  4.62  0.04 -0.04  0.85  0.00  0.00 15.91  0.00  0.00
## 17  2.23  1.04 -1.86  0.42  0.09  0.29  3.72  1.52  6.03
## 18  2.17 -1.80  0.85  0.53  0.37  0.08  3.51  4.57  1.26
## 19  0.93  1.17 -1.48  0.13  0.21  0.33  0.64  1.92  3.79
## 20  0.89 -0.62 -0.90  0.24  0.11  0.24  0.59  0.54  1.40
## 21  0.30 -1.55  1.36  0.01  0.25  0.19  0.07  3.35  3.19
## 22  1.91 -0.09 -0.76  0.45  0.00  0.07  2.71  0.01  1.00
## 23  0.08 -1.35  0.82  0.00  0.47  0.17  0.00  2.57  1.17
## 24  0.54  0.77  1.35  0.05  0.10  0.32  0.22  0.83  3.15
## 25  0.11 -0.04  0.74  0.00  0.00  0.17  0.01  0.00  0.95
## 26  0.00  0.36 -1.57  0.00  0.03  0.50  0.00  0.18  4.28
## 27  0.87  1.06 -1.64  0.06  0.09  0.22  0.56  1.58  4.69
## 28  0.35 -1.56  0.28  0.02  0.38  0.01  0.09  3.41  0.14
## 29  0.38 -0.77 -0.37  0.03  0.11  0.03  0.11  0.84  0.24
## 30 -0.48 -1.06 -1.23  0.06  0.28  0.37  0.17  1.58  2.62
## 31 -0.43 -0.33 -1.07  0.06  0.03  0.37  0.14  0.15  1.99
## 32 -0.25 -3.08  1.05  0.01  0.81  0.09  0.05 13.33  1.93
## 33 -1.07  2.09 -1.00  0.09  0.36  0.08  0.85  6.15  1.74
## 34 -0.68  0.54  2.21  0.04  0.03  0.45  0.35  0.40  8.47
## 35 -0.29 -1.20 -1.31  0.01  0.21  0.25  0.06  2.01  2.96
## 36 -1.54  0.43  0.51  0.25  0.02  0.03  1.77  0.26  0.46
## 37 -2.41 -1.58 -1.50  0.47  0.20  0.18  4.32  3.52  3.92
## 38 -1.99 -0.29 -0.34  0.54  0.01  0.02  2.97  0.12  0.20
## 39 -0.96  2.07  2.59  0.06  0.27  0.42  0.68  6.00 11.61
## 40 -2.56  0.25 -0.42  0.76  0.01  0.02  4.89  0.08  0.30
## 41 -2.86  3.80  0.03  0.34  0.60  0.00  6.09 20.25  0.00
```

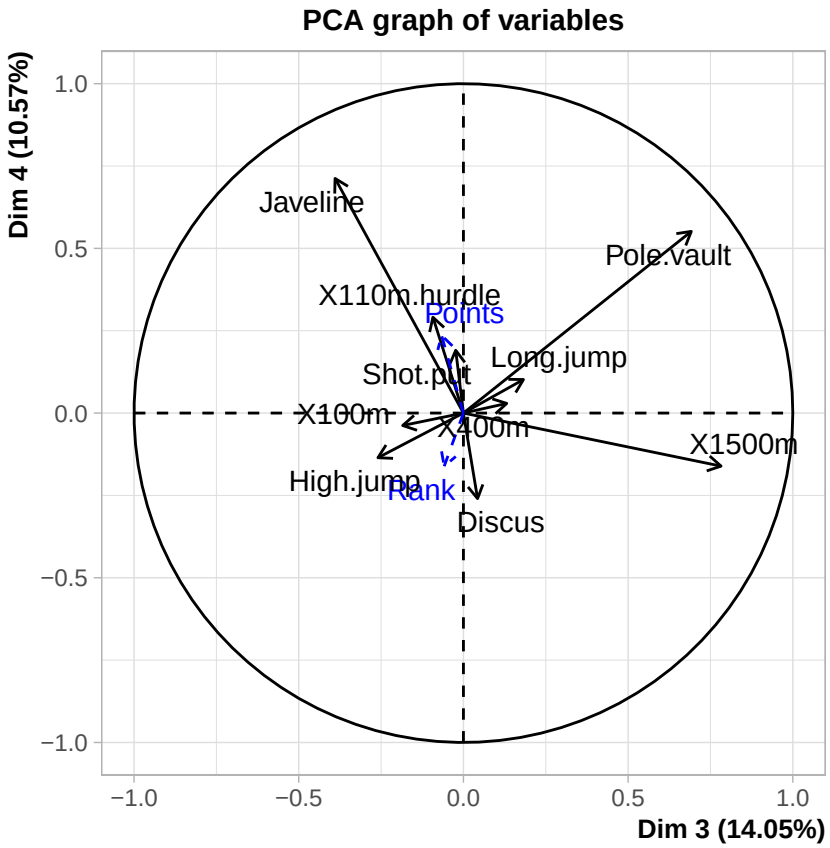
```r
# Calculating loadings adjusted by the square root of eigenvalues for the first five dimensions
loadings <- sweep(res.pca$var$coord, 2, sqrt(res.pca$eig[1:5, 1]), FUN="/")

# Creating scatter plots of individuals and variables on the third and fourth axes
plot(res.pca, choix = "ind", habillage = 13, axes = 3:4, cex = 0.7)
```




```
plot(res.pca, choix = "var", habillage = 13, axes = 3:4)
```

```
## Warning in plot.PCA(res.pca, choix = "var", habillage = 13,
## axes = 3:4): Habillage must be in c('contrib','cos2','none')
```



This section of code provides detailed analysis of PCA results: - Displays the coordinates, squared cosines, and contributions for the first three individuals. - Calculates loadings adjusted by the square root of eigenvalues for the first five dimensions and creates scatter plots of individuals and variables on axes three and four.

```
# To save the graph in PDF format (example)
# pdf("mypath/mygraph.pdf")
# plot(res.pca, choix = "var", habillage = 13, axes = 3:4, newplot = FALSE)
# dev.off()
```

The commented lines demonstrate how to save the plotted graph in a PDF file.

Step 6 | Automatically Describing the Dimensions of Variability

```
# Describing dimensions of variability with default probability
dimdesc(res.pca)
```

```
## $Dim.1
##
## Link between the variable and the continuous variables (R-square)
## ================================================================================
##              correlation      p.value
## Points         0.9561543 2.099191e-22
```

```
## Long.jump      0.7418997 2.849886e-08
## Shot.put       0.6225026 1.388321e-05
## High.jump      0.5719453 9.362285e-05
## Discus         0.5524665 1.802220e-04
## Rank          -0.6705104 1.616348e-06
## X400m         -0.6796099 1.028175e-06
## X110m.hurdle  -0.7462453 2.136962e-08
## X100m         -0.7747198 2.778467e-09
##
## $Dim.2
##
## Link between the variable and the continuous variables (R-square)
## ================================================================================
##           correlation      p.value
## Discus      0.6063134 2.650745e-05
## Shot.put    0.5983033 3.603567e-05
## X400m       0.5694378 1.020941e-04
## X1500m      0.4742238 1.734405e-03
## High.jump   0.3502936 2.475025e-02
## Javeline    0.3169891 4.344974e-02
## Long.jump  -0.3454213 2.696969e-02
##
## $Dim.3
##
## Link between the variable and the continuous variables (R-square)
## ================================================================================
##            correlation      p.value
## X1500m       0.7821428 1.554450e-09
## Pole.vault   0.6917567 5.480172e-07
## Javeline    -0.3896554 1.179331e-02
```

```
# Describing dimensions of variability with specified probability (e.g., 20%)
dimdesc(res.pca, proba = 0.2)
```

```
## $Dim.1
##
## Link between the variable and the continuous variables (R-square)
## ================================================================================
##              correlation      p.value
## Points         0.9561543 2.099191e-22
## Long.jump      0.7418997 2.849886e-08
## Shot.put       0.6225026 1.388321e-05
## High.jump      0.5719453 9.362285e-05
## Discus         0.5524665 1.802220e-04
## Javeline       0.2771108 7.942460e-02
## Rank          -0.6705104 1.616348e-06
## X400m         -0.6796099 1.028175e-06
## X110m.hurdle  -0.7462453 2.136962e-08
## X100m         -0.7747198 2.778467e-09
##
## Link between the variable and the categorical variable (1-way anova)
## ============================================
##                     R2   p.value
## Competition 0.05110487 0.1552515
```

```
##
## Link between variable and the categories of the categorical variables
## ========================================================================
##                         Estimate   p.value
## Competition=OlympicG  0.4393744 0.1552515
## Competition=Decastar -0.4393744 0.1552515
##
## $Dim.2
##
## Link between the variable and the continuous variables (R-square)
## ================================================================================
##              correlation      p.value
## Discus         0.6063134 2.650745e-05
## Shot.put       0.5983033 3.603567e-05
## X400m          0.5694378 1.020941e-04
## X1500m         0.4742238 1.734405e-03
## High.jump      0.3502936 2.475025e-02
## Javeline       0.3169891 4.344974e-02
## X110m.hurdle   0.2287933 1.501925e-01
## Long.jump     -0.3454213 2.696969e-02
##
## $Dim.3
##
## Link between the variable and the continuous variables (R-square)
## ================================================================================
##            correlation      p.value
## X1500m       0.7821428 1.554450e-09
## Pole.vault   0.6917567 5.480172e-07
## High.jump   -0.2595119 1.013160e-01
## Javeline    -0.3896554 1.179331e-02
```

These lines execute the `dimdesc` function to automatically describe the dimensions of variability in the PCA analysis, providing insights into the contribution of variables to each dimension.

Step 7 | Going back to Raw Data

```r
# Scale and round the first 12 columns of 'datd'
scaled_rounded_data <- round(scale(datd[, 2:13]), 2)

# Display the first few rows of the scaled and rounded data
head(scaled_rounded_data)
```

```
##      X100m Long.jump Shot.put High.jump X400m X110m.hurdle
## [1,]  0.16      1.01     0.43      1.05  0.17         0.18
## [2,] -0.91      0.44    -0.26     -1.31 -0.21        -1.18
## [3,]  0.08      0.13     0.36      0.71 -1.08        -1.09
## [4,]  0.08     -0.09    -0.28     -0.64 -0.60         0.81
## [5,]  1.30     -0.54     0.86      1.38  0.70         1.49
## [6,]  0.43      1.07    -0.20      0.04 -0.81        -0.80
##      Discus Pole.vault Javeline X1500m  Rank Points
## [1,]  -0.17       0.93     1.01   1.09 -1.40   0.62
## [2,]   1.89       0.57     0.38   1.93 -1.28   0.34
```

```
## [3,]   1.37       0.57    -1.66   1.81 -1.15   0.27
## [4,]  -1.02       2.01     0.92   0.09 -1.03   0.18
## [5,]   0.57      -0.15     1.06  -0.22 -0.90   0.09
## [6,]  -0.95       0.57    -1.36  -0.08 -0.77   0.07
```

```
# Display the correlation matrix for columns 2 to 13 of 'datd'
round(cor(datd[, 2:13]), 2)
```

```
##              X100m Long.jump Shot.put High.jump X400m
## X100m         1.00     -0.60    -0.36     -0.25  0.52
## Long.jump    -0.60      1.00     0.18      0.29 -0.60
## Shot.put     -0.36      0.18     1.00      0.49 -0.14
## High.jump    -0.25      0.29     0.49      1.00 -0.19
## X400m         0.52     -0.60    -0.14     -0.19  1.00
## X110m.hurdle  0.58     -0.51    -0.25     -0.28  0.55
## Discus       -0.22      0.19     0.62      0.37 -0.12
## Pole.vault   -0.08      0.20     0.06     -0.16 -0.08
## Javeline     -0.16      0.12     0.37      0.17  0.00
## X1500m       -0.06     -0.03     0.12     -0.04  0.41
## Rank          0.30     -0.60    -0.37     -0.49  0.56
## Points       -0.68      0.73     0.63      0.58 -0.67
##              X110m.hurdle Discus Pole.vault Javeline X1500m
## X100m                0.58  -0.22      -0.08    -0.16  -0.06
## Long.jump           -0.51   0.19       0.20     0.12  -0.03
## Shot.put            -0.25   0.62       0.06     0.37   0.12
## High.jump           -0.28   0.37      -0.16     0.17  -0.04
## X400m                0.55  -0.12      -0.08     0.00   0.41
## X110m.hurdle         1.00  -0.33       0.00     0.01   0.04
## Discus              -0.33   1.00      -0.15     0.16   0.26
## Pole.vault           0.00  -0.15       1.00    -0.03   0.25
## Javeline             0.01   0.16      -0.03     1.00  -0.18
## X1500m               0.04   0.26       0.25    -0.18   1.00
## Rank                 0.44  -0.39      -0.32    -0.21   0.09
## Points              -0.64   0.48       0.20     0.42  -0.19
##               Rank Points
## X100m         0.30  -0.68
## Long.jump    -0.60   0.73
## Shot.put     -0.37   0.63
## High.jump    -0.49   0.58
## X400m         0.56  -0.67
## X110m.hurdle  0.44  -0.64
## Discus       -0.39   0.48
## Pole.vault   -0.32   0.20
## Javeline     -0.21   0.42
## X1500m        0.09  -0.19
## Rank          1.00  -0.74
## Points       -0.74   1.00
```

This segment scales and rounds the first 12 columns of 'datd' (excluding the assumed non-quantitative first column) and displays the scaled and rounded data, along with the correlation matrix of columns 2 to 13.