Final Assignment Arthur WEHBE

Arthur WEHBE

2024-05-19

Introduction

This assignment involves two tasks: clustering and principal components analysis (PCA). The purpose of this assignment is to demonstrate the ability to analyze data using these techniques and to present the findings in a clear and detailed report.

Question 1: Clustering

Dataset Overview

The `pottery.csv` dataset contains the chemical composition of Romano-British pottery, with measurements for nine different oxides and the location (kiln) where each piece of pottery was found.

1. Explore the Dataset

```
getwd()
```

```
## [1] "C:/Devoir Arthur/Dorset College/DS"
```

```
library(ggplot2)
```

```
## Warning: le package 'ggplot2' a été compilé avec la version R 4.3.3
```

```
library(cluster)
library(factoextra)
```

```
## Warning: le package 'factoextra' a été compilé avec la version R 4.3.3

## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```
library(dendextend)
```

```
## Warning: le package 'dendextend' a été compilé avec la version R 4.3.3
```

```
##
## ---------------------
## Welcome to dendextend version 1.17.1
## Type citation('dendextend') for how to cite the package.
##
## Type browseVignettes(package = 'dendextend') for the package vignette.
## The github page is: https://github.com/talgalili/dendextend/
##
## Suggestions and bug-reports can be submitted at: https://github.com/talgalili/dendextend/issues
## You may ask questions at stackoverflow, use the r and dendextend tags:
##   https://stackoverflow.com/questions/tagged/dendextend
##
##  To suppress this message use:  suppressPackageStartupMessages(library(dendextend))
## ---------------------
```

```
##
## Attachement du package : 'dendextend'
```

```
## L'objet suivant est masqué depuis 'package:stats':
##
##     cutree
```

```r
pottery <- read.csv('pottery.csv')

head(pottery)
```

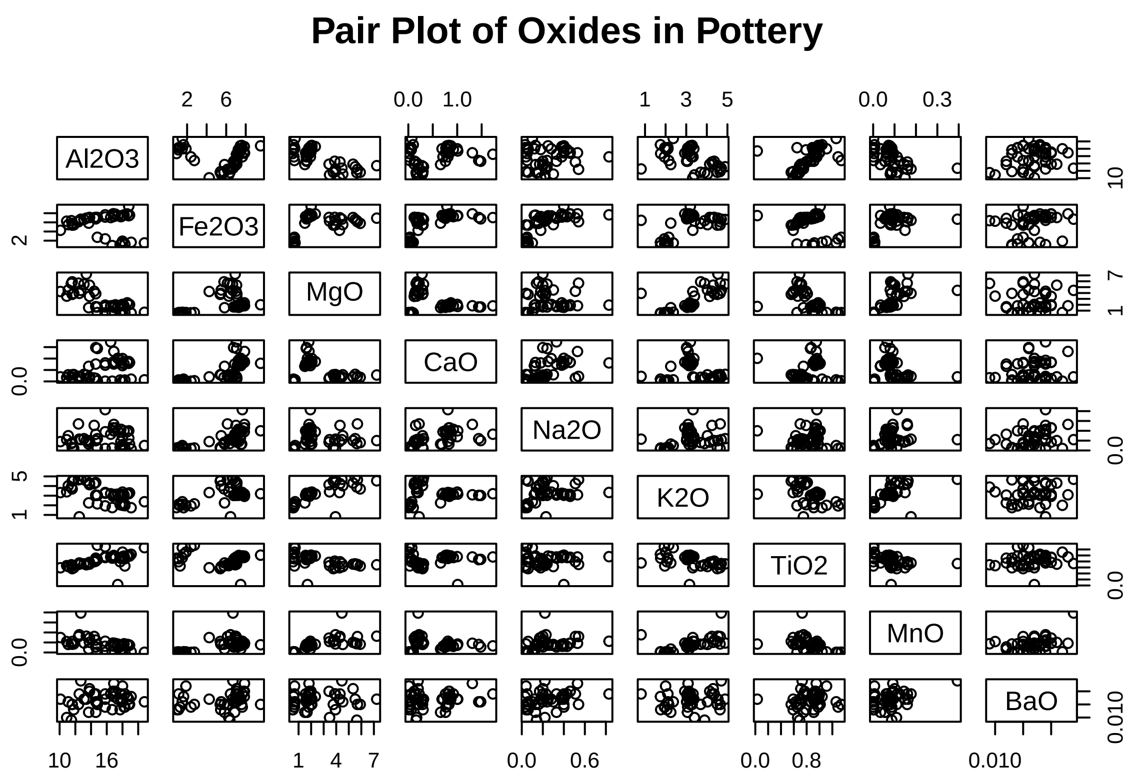
```
##         kiln Al2O3 Fe2O3  MgO  CaO Na2O  K2O TiO2   MnO   BaO
## 1 Gloucester  18.8  9.52 2.00 0.79 0.40 3.20 1.01 0.077 0.015
## 2 Gloucester  16.9  7.33 1.65 0.84 0.40 3.05 0.99 0.067 0.018
## 3 Gloucester  18.2  7.64 1.82 0.77 0.40 3.07 0.98 0.087 0.014
## 4 Gloucester  17.4  7.48 1.71 1.01 0.40 3.16 0.03 0.084 0.017
## 5 Gloucester  16.9  7.29 1.56 0.76 0.40 3.05 1.00 0.063 0.019
## 6 Gloucester  17.8  7.24 1.83 0.92 0.43 3.12 0.93 0.061 0.019
```

```r
summary(pottery)
```

```
##      kiln               Al2O3           Fe2O3             MgO
##  Length:48          Min.   :10.10   Min.   :0.920   Min.   :0.530
##  Class :character   1st Qu.:13.62   1st Qu.:5.428   1st Qu.:1.605
##  Mode  :character   Median :16.15   Median :6.895   Median :1.930
##                     Mean   :15.61   Mean   :5.826   Mean   :2.543
##                     3rd Qu.:18.00   3rd Qu.:7.353   3rd Qu.:3.895
##                     Max.   :20.80   Max.   :9.520   Max.   :7.230
##       CaO              Na2O              K2O             TiO2
##  Min.   :0.0100   Min.   :0.0300   Min.   :0.810   Min.   :0.0300
##  1st Qu.:0.1450   1st Qu.:0.1150   1st Qu.:2.790   1st Qu.:0.7175
##  Median :0.2950   Median :0.2100   Median :3.155   Median :0.9050
##  Mean   :0.5112   Mean   :0.2454   Mean   :3.181   Mean   :0.8533
##  3rd Qu.:0.8300   3rd Qu.:0.3850   3rd Qu.:3.748   3rd Qu.:0.9650
##  Max.   :1.7300   Max.   :0.8300   Max.   :4.890   Max.   :1.3400
##       MnO                BaO
##  Min.   :0.00100   Min.   :0.00900
```

```
## 1st Qu.:0.05000   1st Qu.:0.01500
## Median :0.07850   Median :0.01700
## Mean   :0.07975   Mean   :0.01673
## 3rd Qu.:0.09575   3rd Qu.:0.01900
## Max.   :0.39400   Max.   :0.02400
```

```
pairs(pottery[,2:10], main="Pair Plot of Oxides in Pottery")
```



2. Data Preparation

HEre we use standardisation because varaible like Al203 and Fe203 have different ranges which can disproportionate the further analysis and the clustering. After standartisation all data will have amean of 0 and a standard deviation of 1 which will make comparison and analysis relevant and accurate.

```
pottery_data <- pottery[, -1]

pottery_scaled <- scale(pottery_data)

head(pottery_scaled)
```

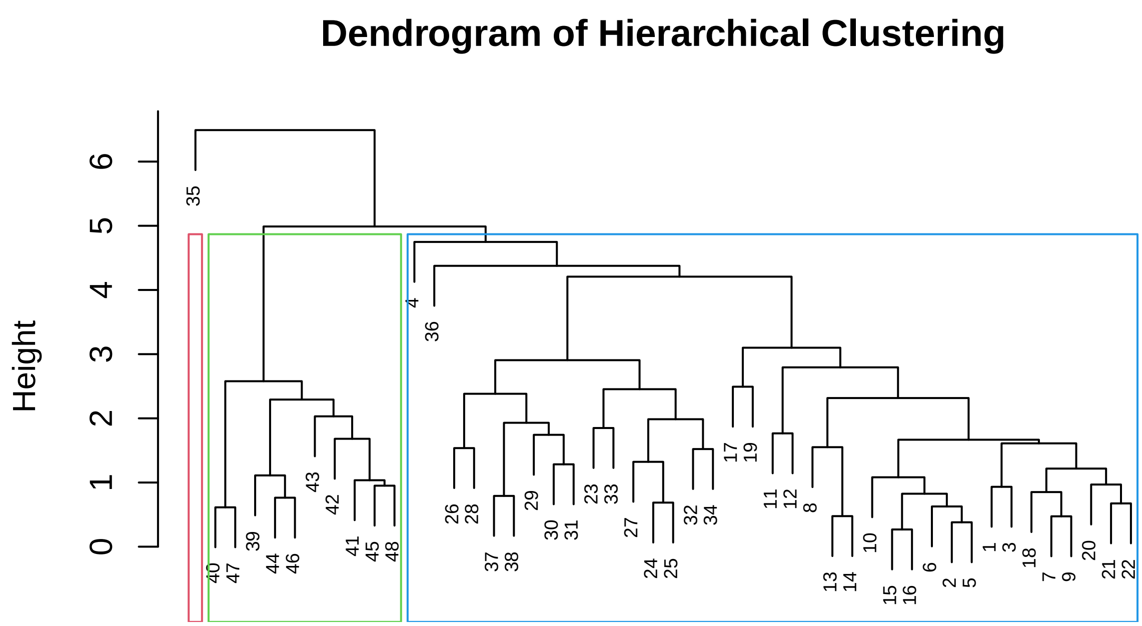
```
##          Al2O3     Fe2O3        MgO       CaO      Na2O         K2O      TiO2
## [1,] 1.1787012 1.5744454 -0.3148639 0.6196585 0.8884038  0.02101780 0.7323502
## [2,] 0.4756433 0.6410724 -0.5176903 0.7308080 0.8884038 -0.14170065 0.6388587
## [3,] 0.9566829 0.7731937 -0.4191746 0.5751987 0.8884038 -0.12000486 0.5921129
```

```
## [4,] 0.6606585 0.7050021 -0.4829200 1.1087164 0.8884038 -0.02237379 -3.8487339
## [5,] 0.4756433 0.6240245 -0.5698457 0.5529688 0.8884038 -0.14170065  0.6856044
## [6,] 0.8086707 0.6027146 -0.4133796 0.9086472 1.0608164 -0.06576537  0.3583841
##              MnO         BaO
## [1,] -0.04129390 -0.55791357
## [2,] -0.19145353  0.41003287
## [3,]  0.10886573 -0.88056239
## [4,]  0.06381784  0.08738405
## [5,] -0.25151738  0.73268168
## [6,] -0.28154931  0.73268168
```

3. Hierarchical Clustering

There seems to have 3 cluster in the dataset from analysing the dendogram.

```r
hclust_res <- hclust(dist(pottery_scaled), method = "average")
plot(hclust_res, main = "Dendrogram of Hierarchical Clustering", xlab = "", sub = "", cex = 0.6)

rect.hclust(hclust_res, k = 3, border = 2:4)  # Cutting dendrogram at 3 clusters
```



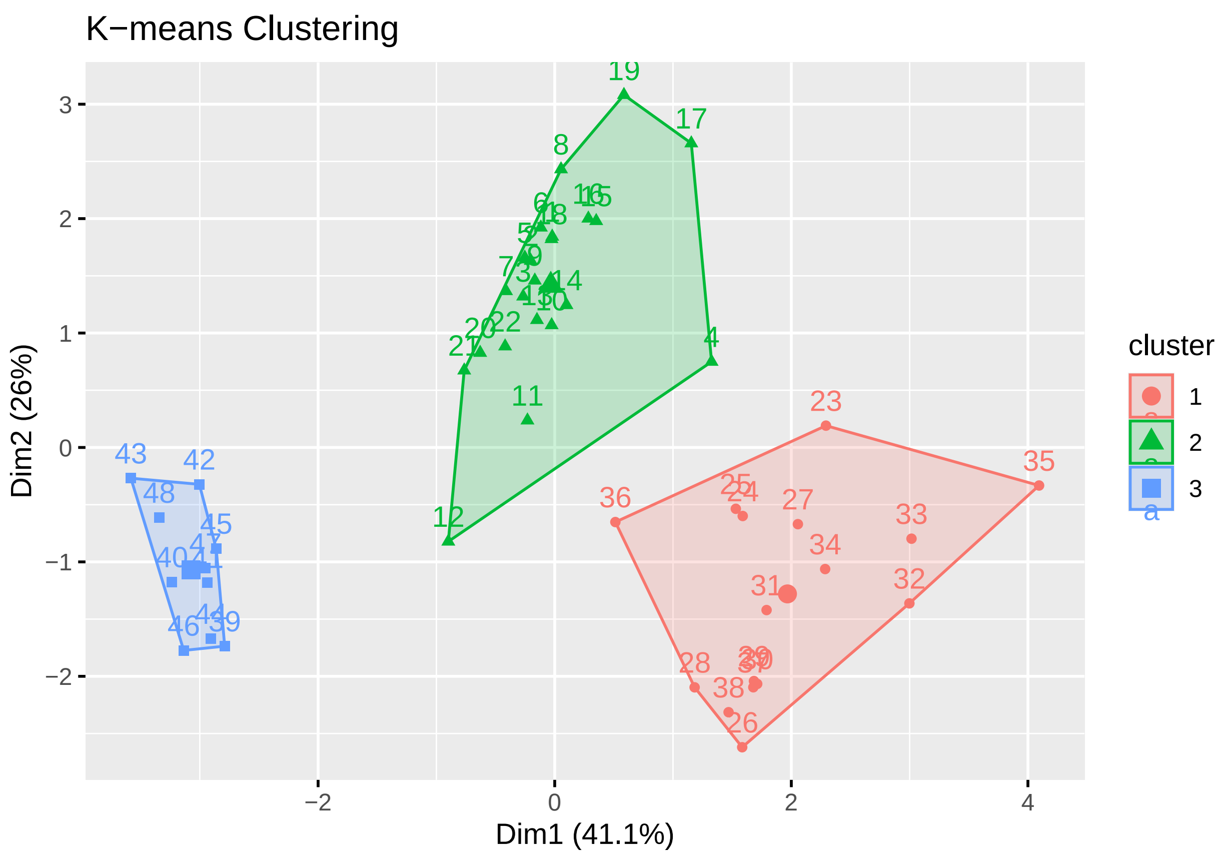

```r
clusters_hierarchical <- cutree(hclust_res, k = 3)
pottery$Cluster_Hierarchical <- clusters_hierarchical

head(pottery)
```

```
##         kiln Al2O3 Fe2O3  MgO  CaO Na2O  K2O TiO2   MnO   BaO
## 1 Gloucester  18.8  9.52 2.00 0.79 0.40 3.20 1.01 0.077 0.015
## 2 Gloucester  16.9  7.33 1.65 0.84 0.40 3.05 0.99 0.067 0.018
## 3 Gloucester  18.2  7.64 1.82 0.77 0.40 3.07 0.98 0.087 0.014
## 4 Gloucester  17.4  7.48 1.71 1.01 0.40 3.16 0.03 0.084 0.017
## 5 Gloucester  16.9  7.29 1.56 0.76 0.40 3.05 1.00 0.063 0.019
## 6 Gloucester  17.8  7.24 1.83 0.92 0.43 3.12 0.93 0.061 0.019
##   Cluster_Hierarchical
## 1                    1
## 2                    1
## 3                    1
## 4                    1
## 5                    1
## 6                    1
```

4. K-means Clustering

With this type of clustering we can also see that there is 3 clusters in the dataset. The decision for k = 3 comes from the elbow method displaying clear "elbow" at k = 3.

```r
set.seed(123)

kmeans_res <- kmeans(pottery_scaled, centers = 3, nstart = 25)
pottery$Cluster_KMeans <- kmeans_res$cluster

fviz_cluster(kmeans_res, data = pottery_scaled, main = "K-means Clustering")
```



5. Comparing clustering

The ageement between the hierarchical and K-means clustering solutions is pretty high, as shown by the results of the contingency table. Most of the pots classified in cluster 1 by hierarchical clustering are also in cluster 2 by K-means (22 out of 38), and cluster 3 in hierarchical clustering aligns perfectly with cluster 3 in K-means (10 out of 10). However, there is a small differences in cluster 2 of hierarchical clustering, where one pot overlaps with cluster 1 in K-means.

```
table(clusters_hierarchical, kmeans_res$cluster)
```

```
##
## clusters_hierarchical  1  2  3
##                     1 15 22  0
##                     2  1  0  0
##                     3  0  0 10
```

6. Relation between clustering and ‘kiln’

The relationship between both clustering solutions and the ‘kiln’ variable reveals some alignment, but notable differences exist. Hierarchical clustering groups Caldicot and Thorns together primarily, while K-means separates Caldicot into its own cluster. While there is some consistency, minor differences suggest potential variability in cluster assignments, raising concerns about the reproducibility of the clustering solutions, particularly when different methods or parameters are employed.

```r
# Relationship with 'kiln'
table(pottery$Cluster_Hierarchical, pottery$kiln)
```

```
##
##     Ashley Rails Caldicot Gloucester Islands Thorns Llanedeyrn
##   1            0        2         22              0         13
##   2            0        0          0              0          1
##   3            5        0          0              5          0
```

```r
table(pottery$Cluster_KMeans, pottery$kiln)
```

```
##
##     Ashley Rails Caldicot Gloucester Islands Thorns Llanedeyrn
##   1            0        2          0              0         14
##   2            0        0         22              0          0
##   3            5        0          0              5          0
```

Question 2 : Principal Components Analysis

##1. Introduction

Principal Component Analysis (PCA) is a technique used to reduce the dimensionality of a dataset while retaining as much variances as possible. This is particularly useful when dealing with datasets that have many variables.

2. Data Exploration

The Decathlon Olympics dataset comprises the performance scores of 28 athletes across ten different events. To gain insights into the data, we conducted an initial exploration:

The dataset was loaded and examined using the head() and summary() functions. Numeric columns were isolated to visualize the pairwise relationships between variables using a pair plot.

```r
library(FactoMineR)
```

```
## Warning: le package 'FactoMineR' a été compilé avec la version R 4.3.3
```

```r
library(factoextra)

data(decathlon)

head(decathlon)
```

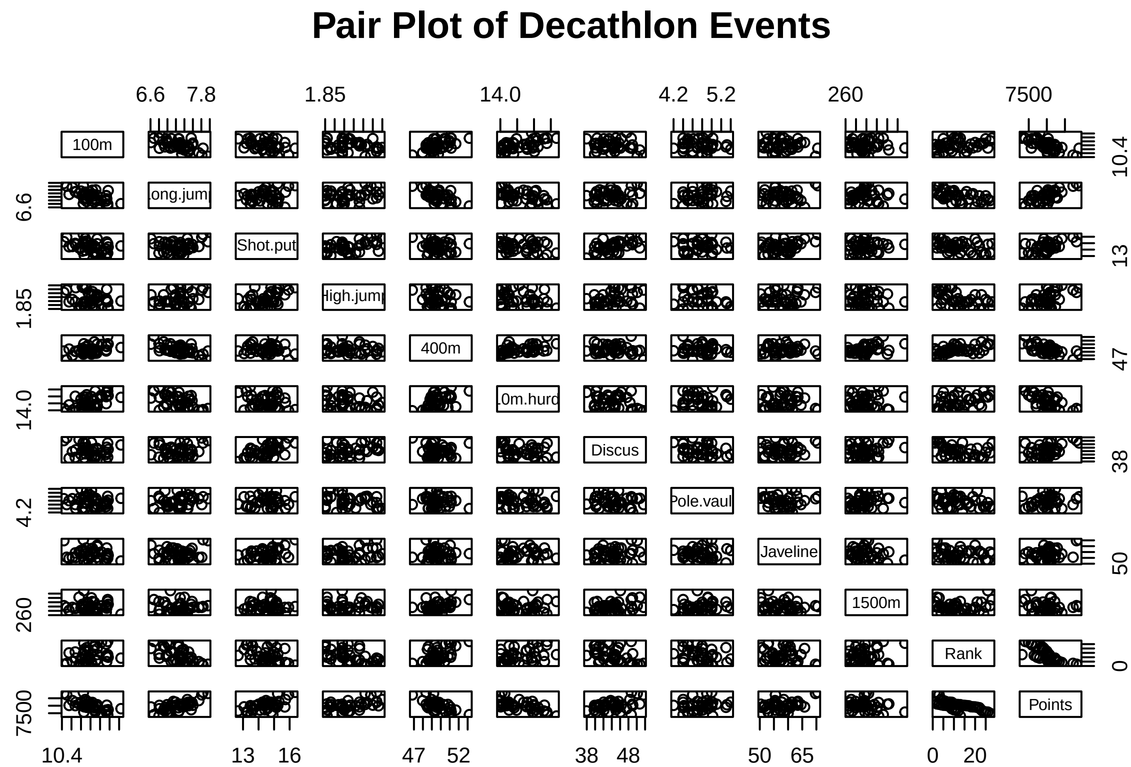
```
##          100m Long.jump Shot.put High.jump  400m 110m.hurdle Discus Pole.vault
## SEBRLE  11.04      7.58    14.83      2.07 49.81       14.69  43.75       5.02
## CLAY    10.76      7.40    14.26      1.86 49.37       14.05  50.72       4.92
## KARPOV  11.02      7.30    14.77      2.04 48.37       14.09  48.95       4.92
## BERNARD 11.02      7.23    14.25      1.92 48.93       14.99  40.87       5.32
## YURKOV  11.34      7.09    15.19      2.10 50.42       15.31  46.26       4.72
## WARNERS 11.11      7.60    14.31      1.98 48.68       14.23  41.10       4.92
```

```
##         Javeline 1500m Rank Points Competition
## SEBRLE     63.19 291.7    1   8217    Decastar
## CLAY       60.15 301.5    2   8122    Decastar
## KARPOV     50.31 300.2    3   8099    Decastar
## BERNARD    62.77 280.1    4   8067    Decastar
## YURKOV     63.44 276.4    5   8036    Decastar
## WARNERS    51.77 278.1    6   8030    Decastar
```

```r
summary(decathlon)
```

```
##       100m         Long.jump       Shot.put       High.jump         400m
##  Min.   :10.44   Min.   :6.61   Min.   :12.68   Min.   :1.850   Min.   :46.81
##  1st Qu.:10.85   1st Qu.:7.03   1st Qu.:13.88   1st Qu.:1.920   1st Qu.:48.93
##  Median :10.98   Median :7.30   Median :14.57   Median :1.950   Median :49.40
##  Mean   :11.00   Mean   :7.26   Mean   :14.48   Mean   :1.977   Mean   :49.62
##  3rd Qu.:11.14   3rd Qu.:7.48   3rd Qu.:14.97   3rd Qu.:2.040   3rd Qu.:50.30
##  Max.   :11.64   Max.   :7.96   Max.   :16.36   Max.   :2.150   Max.   :53.20
##   110m.hurdle        Discus        Pole.vault       Javeline
##  Min.   :13.97   Min.   :37.92   Min.   :4.200   Min.   :50.31
##  1st Qu.:14.21   1st Qu.:41.90   1st Qu.:4.500   1st Qu.:55.27
##  Median :14.48   Median :44.41   Median :4.800   Median :58.36
##  Mean   :14.61   Mean   :44.33   Mean   :4.762   Mean   :58.32
##  3rd Qu.:14.98   3rd Qu.:46.07   3rd Qu.:4.920   3rd Qu.:60.89
##  Max.   :15.67   Max.   :51.65   Max.   :5.400   Max.   :70.52
##      1500m            Rank           Points        Competition
##  Min.   :262.1   Min.   : 1.00   Min.   :7313   Decastar:13
##  1st Qu.:271.0   1st Qu.: 6.00   1st Qu.:7802   OlympicG:28
##  Median :278.1   Median :11.00   Median :8021
##  Mean   :279.0   Mean   :12.12   Mean   :8005
##  3rd Qu.:285.1   3rd Qu.:18.00   3rd Qu.:8122
##  Max.   :317.0   Max.   :28.00   Max.   :8893
```

```r
numeric_cols <- decathlon[, sapply(decathlon, is.numeric)]

pair_plot <- pairs(numeric_cols, main = "Pair Plot of Decathlon Events")
```




```
pair_plot
```

```
## NULL
```

3. Methodology

The dataset was preprocessed by extracting only the columns corresponding to the ten events.

The data was then standardized to ensure that each variable contributes equally to the analysis.

PCA was performed on the standardized dataset using the PCA() function from the FactoMineR package.

Eigenvalues and contributions of variables to principal components were visualized using scree plots and variable contributions plots, respectively.

A biplot was generated to visualize the relationship between individuals and variables in the principal component space.

```r
decathlon_data <- decathlon[, -11]

decathlon_data <- as.data.frame(sapply(decathlon_data, as.numeric))

if (anyNA(decathlon_data)) {
  decathlon_data <- apply(decathlon_data, 2, function(x) {
    ifelse(is.na(x), mean(x, na.rm = TRUE), x)
  })
```

```
}

decathlon_scaled <- scale(decathlon_data)

head_scaled_data <- head(decathlon_scaled)
```

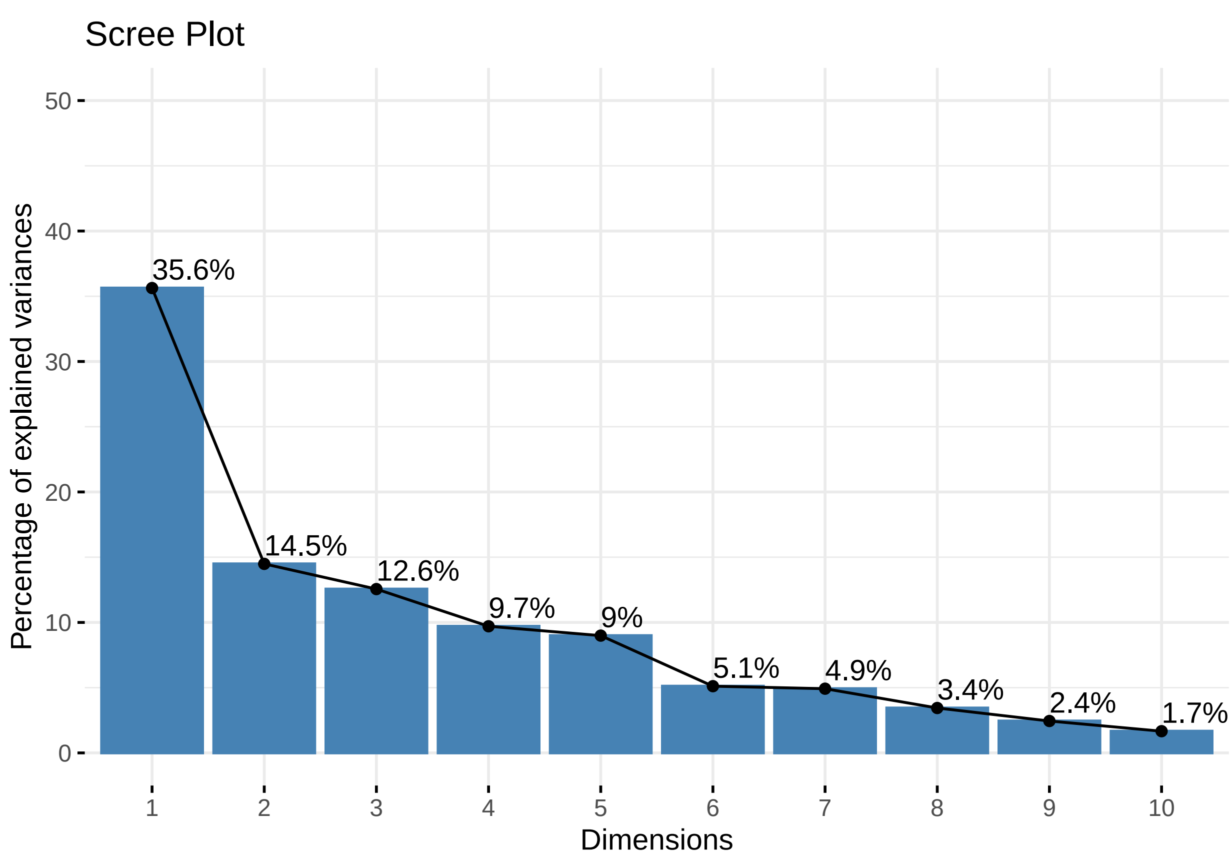
4. Results

Scree Plot: The scree plot revealed the variance explained by each principal component, allowing us to determine the number of significant components with dimension 1 in first with 35.6%.

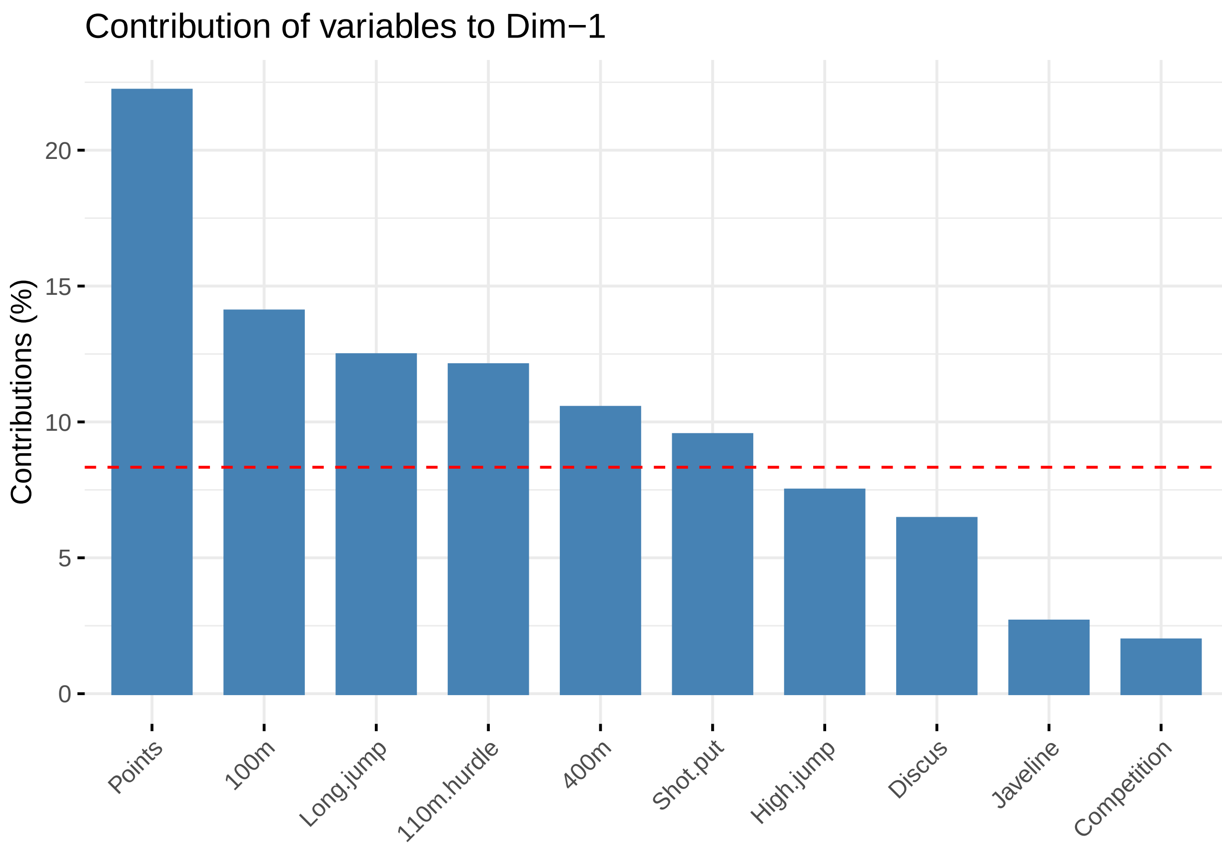
Variable Contributions: The contributions of variables to each principal component were visualized, providing insights into which events contribute most to the variability in the dataset. For the frist dimension it is Points by far then 100M, Long Jump, 110 M hurde, 400M and Short Put pretty close. For the 2nd Dimension it is Shot put, Discuss 400 M pretty simiar in terms of contrivution then 1500M.

Biplot: The biplot displayed the relationship between athletes and events in the principal component space, facilitating the interpretation of athlete performance across different events.

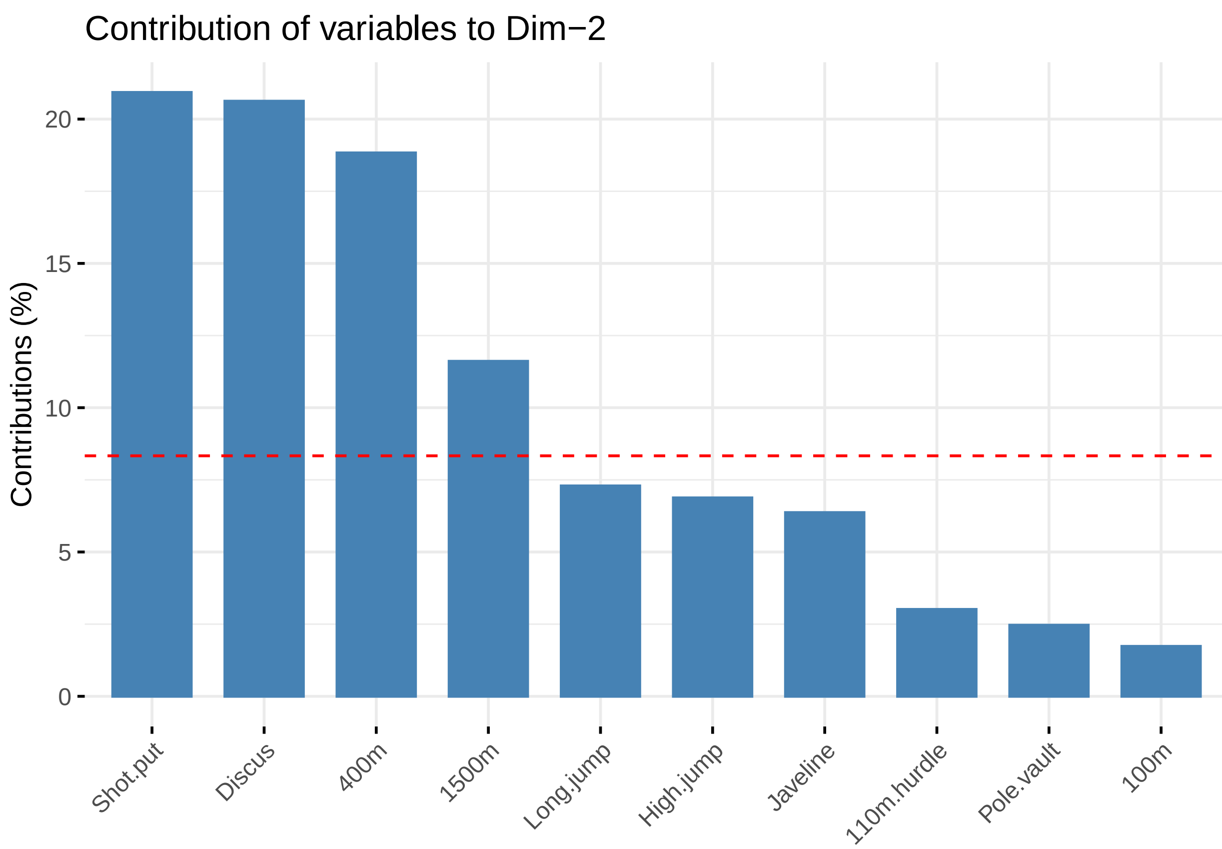
```
pca_res <- PCA(decathlon_scaled, graph = FALSE)

scree_plot <- fviz_eig(pca_res, addlabels = TRUE, ylim = c(0, 50), main = "Scree Plot")

scree_plot
```



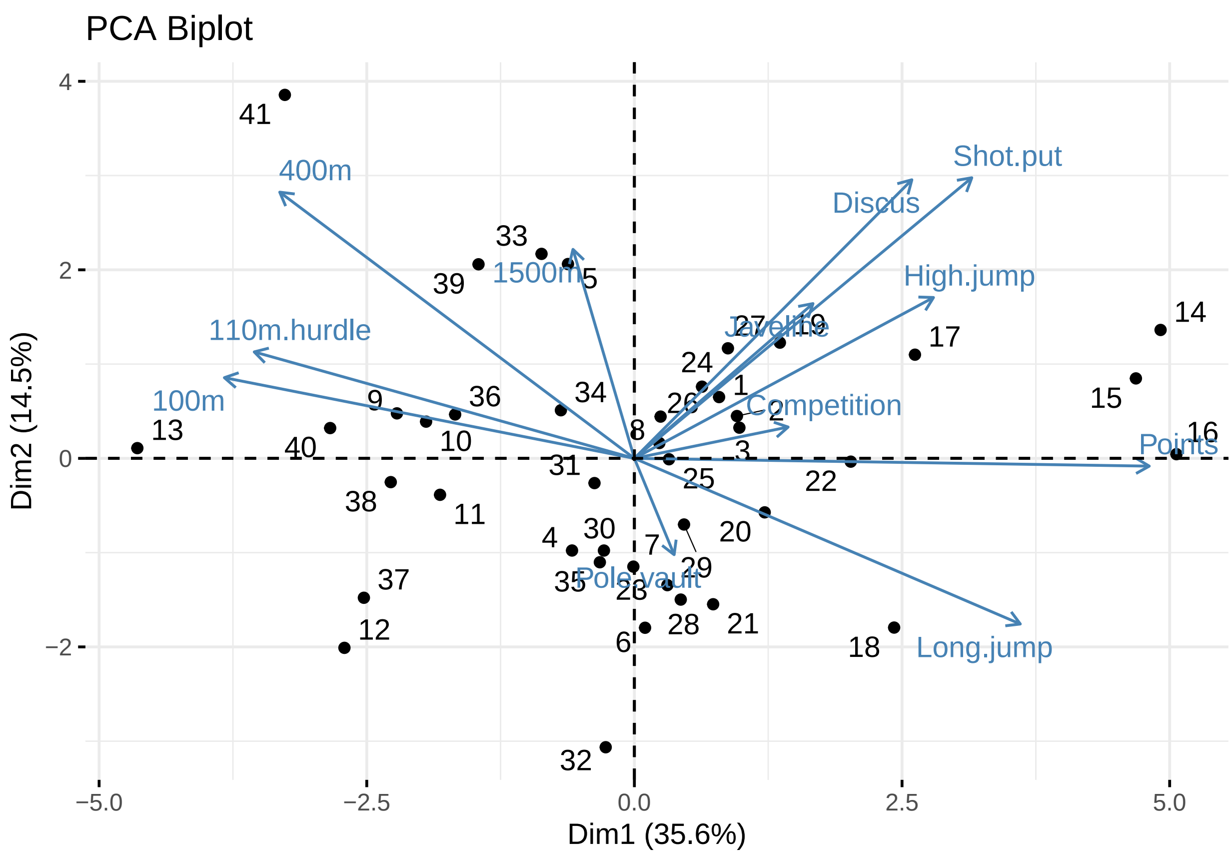
```r
var <- get_pca_var(pca_res)

contrib_pc1 <- fviz_contrib(pca_res, choice = "var", axes = 1, top = 10)

contrib_pc1
```




```r
contrib_pc2 <-fviz_contrib(pca_res, choice = "var", axes = 2, top = 10)

contrib_pc2
```



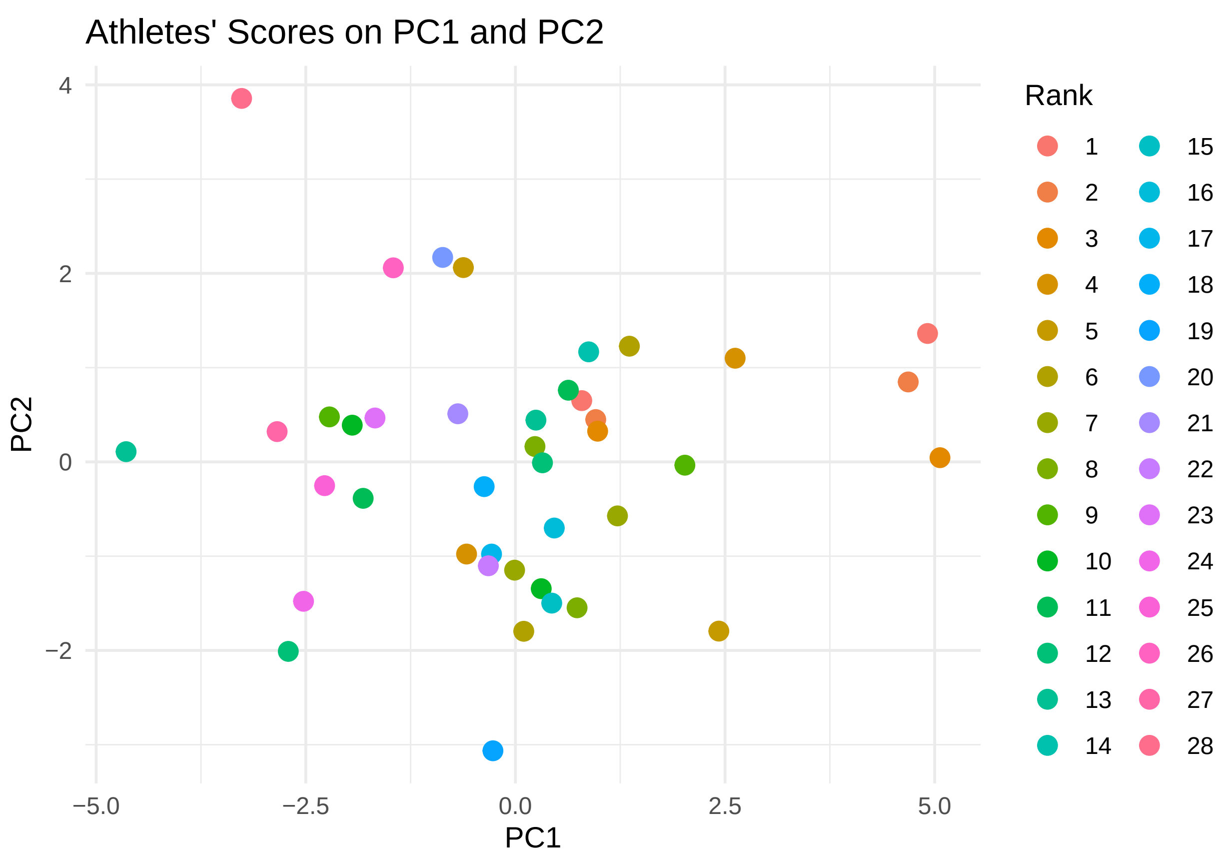

```
biplot<- fviz_pca_biplot(pca_res, repel = TRUE, title = "PCA Biplot")

biplot
```




```r
decathlon$Rank <- factor(decathlon$Rank)


decathlon$PC1 <- pca_res$ind$coord[, 1]
decathlon$PC2 <- pca_res$ind$coord[, 2]

head_with_pca_scores <- head(decathlon)

head_with_pca_scores
```

```
##          100m Long.jump Shot.put High.jump  400m 110m.hurdle Discus Pole.vault
## SEBRLE  11.04      7.58    14.83      2.07 49.81       14.69  43.75       5.02
## CLAY    10.76      7.40    14.26      1.86 49.37       14.05  50.72       4.92
## KARPOV  11.02      7.30    14.77      2.04 48.37       14.09  48.95       4.92
## BERNARD 11.02      7.23    14.25      1.92 48.93       14.99  40.87       5.32
## YURKOV  11.34      7.09    15.19      2.10 50.42       15.31  46.26       4.72
## WARNERS 11.11      7.60    14.31      1.98 48.68       14.23  41.10       4.92
##         Javeline 1500m Rank Points Competition         PC1        PC2
## SEBRLE     63.19 291.7    1   8217    Decastar  0.79121358  0.6501899
## CLAY       60.15 301.5    2   8122    Decastar  0.95795136  0.4491266
## KARPOV     50.31 300.2    3   8099    Decastar  0.98078350  0.3256933
## BERNARD    62.77 280.1    4   8067    Decastar -0.58244724 -0.9773933
## YURKOV     63.44 276.4    5   8036    Decastar -0.62046141  2.0614749
## WARNERS    51.77 278.1    6   8030    Decastar  0.09966088 -1.7968331
```

```r
athletes_plot <- ggplot(decathlon, aes(x = PC1, y = PC2, color = Rank)) +
  geom_point(size = 3) +
  theme_minimal() +
  labs(title = "Athletes' Scores on PC1 and PC2",
       x = "PC1",
       y = "PC2",
       color = "Rank")

athletes_plot
```



5. Conclusion

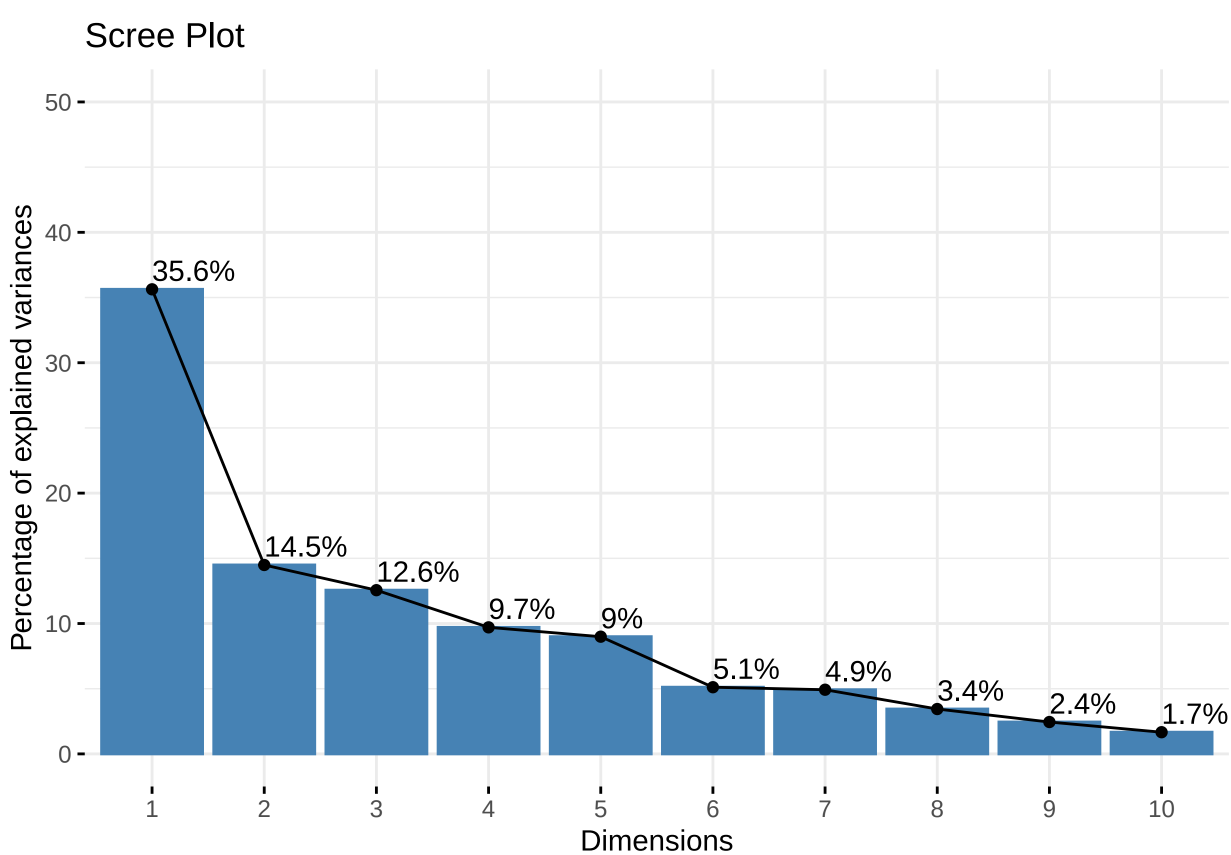
PCA provided valuable insights into the structure of the Decathlon Olympics dataset:

It revealed the underlying patterns in athlete performance across events. It identified the key events driving the variability in the dataset. It allowed for the visualization of athlete performance in a reduced-dimensional space. PCA proved to be a useful tool for analyzing and understanding the complex relationships within the Decathlon Olympics dataset, providing a foundation for further analysis and interpretation.

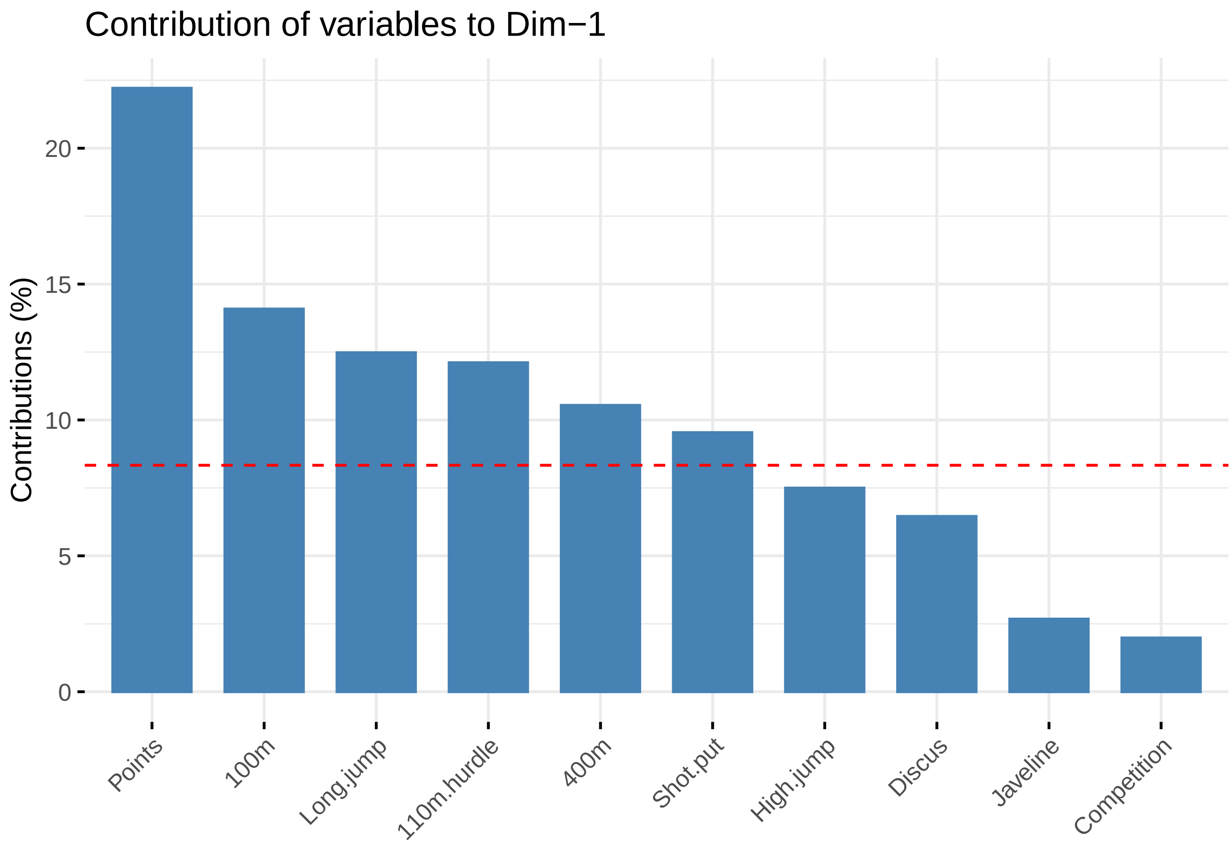
```r
points_plot <- ggplot(decathlon, aes(x = PC1, y = Points)) +
  geom_point() +
  theme_minimal() +
  labs(title = "Relationship between PC1 and Total Points",
       x = "PC1",
       y = "Total Points")
```

```r
output_list <- list( pair_plot, head_scaled_data, scree_plot, contrib_pc1, contrib_pc2, biplot, head_wi

output_list
```

```
## [[1]]
## NULL
##
## [[2]]
##              100m  Long.jump   Shot.put   High.jump       400m 110m.hurdle
## [1,]  0.15949639  1.0113727  0.4280870  1.04744448  0.1678949   0.1783559
## [2,] -0.90504930  0.4424756 -0.2633016 -1.31341860 -0.2135691  -1.1781827
## [3,]  0.08345742  0.1264216  0.3553093  0.71017832 -1.0805328  -1.0933990
## [4,]  0.08345742 -0.0948162 -0.2754312 -0.63888629 -0.5950331   0.8142333
## [5,]  1.30008106 -0.5372918  0.8647535  1.38471063  0.6967428   1.4925026
## [6,]  0.42563282  1.0745835 -0.2026535  0.03564602 -0.8117741  -0.7966562
##          Discus Pole.vault   Javeline       1500m     Points Competition
## [1,] -0.1704074  0.9264789  1.0096532  1.08582657 0.61811720   -1.449591
## [2,]  1.8930385  0.5667665  0.3798390  1.92535303 0.34065189   -1.449591
## [3,]  1.3690358  0.5667665 -1.6587702  1.81398728 0.27347608   -1.449591
## [4,] -1.0230221  2.0056163  0.9226394  0.09210136 0.18001408   -1.449591
## [5,]  0.5726700 -0.1526585  1.0614472 -0.22486271 0.08947277   -1.449591
## [6,] -0.9549313  0.5667665 -1.3562936 -0.07923057 0.07194864   -1.449591
##
## [[3]]
```

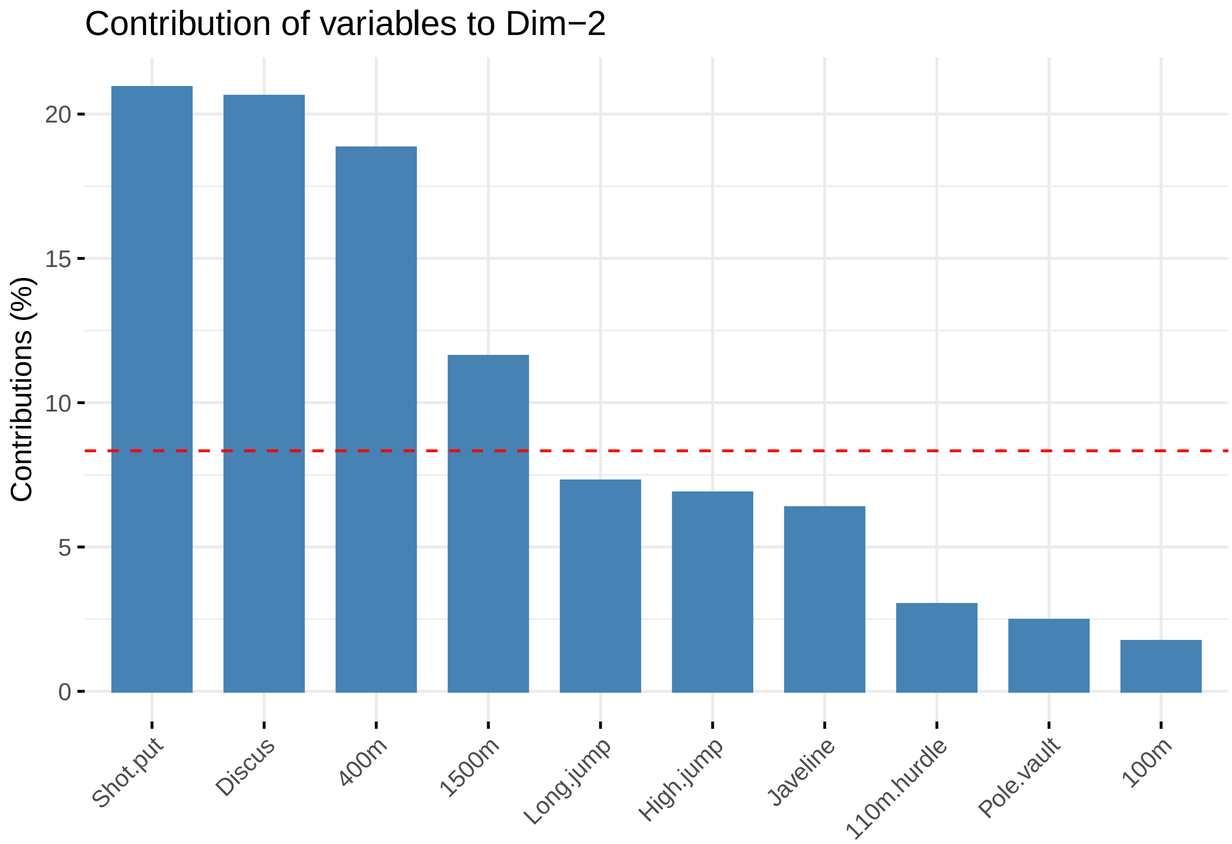


```
##
## [[4]]
```

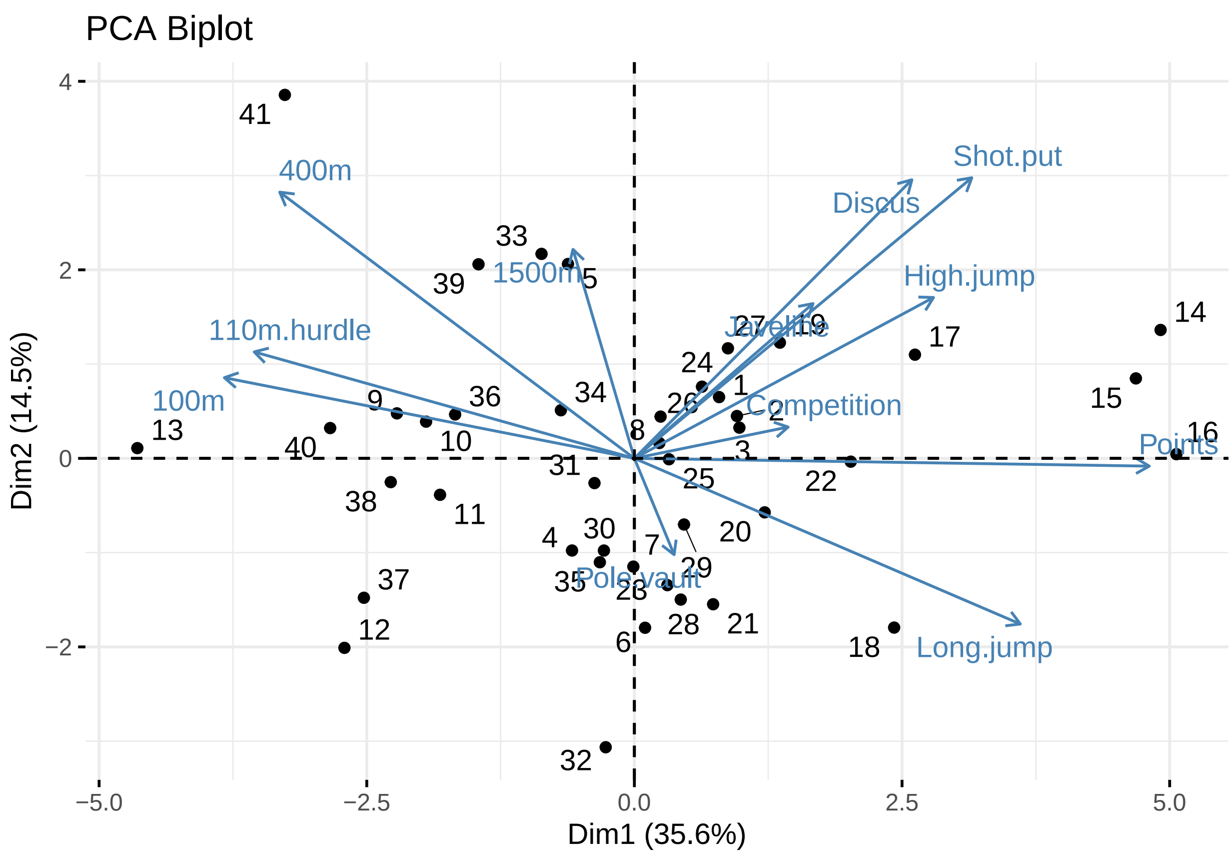


Contribution of variables to Dim−1
Contributions (%)
0
5
10
15
20
Points
100m
Long.jump
110m.hurdle
400m
Shot.put
High.jump
Discus
Javeline
Competition

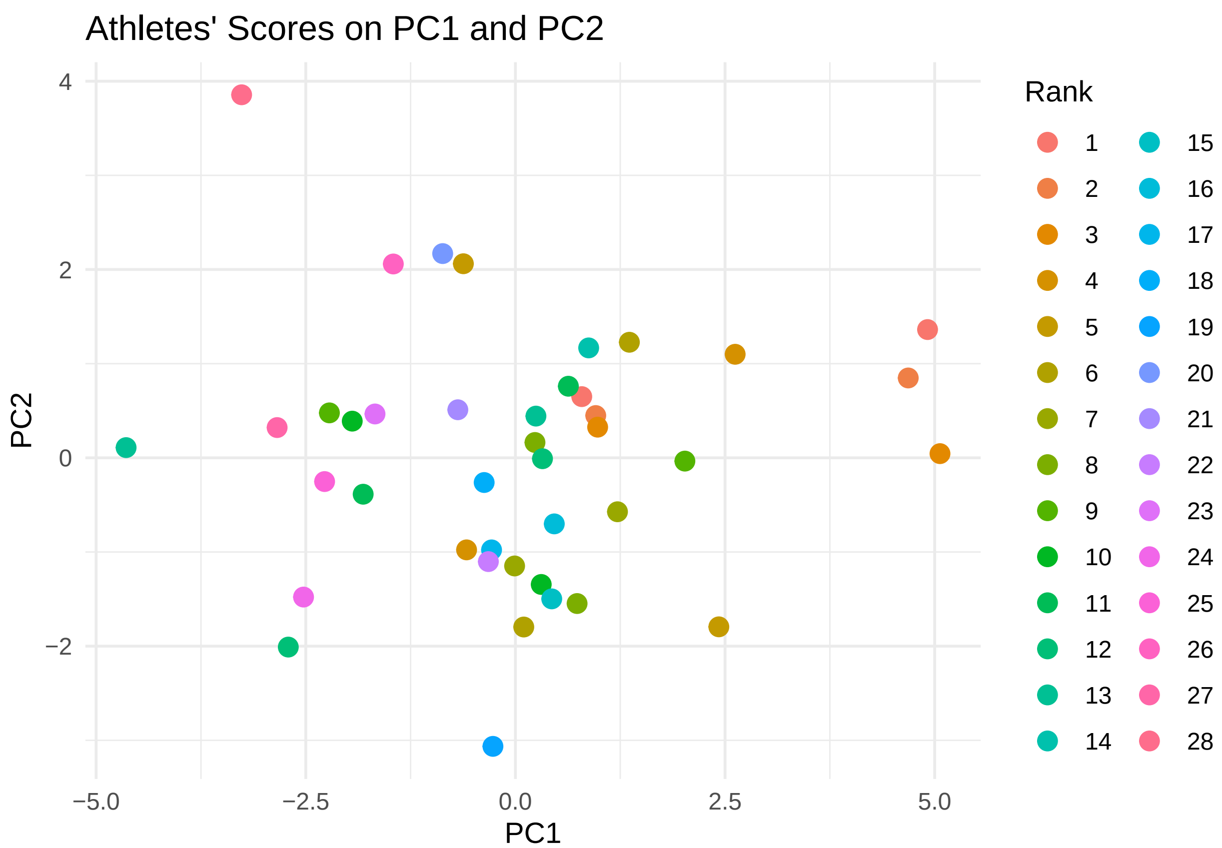

```
##
## [[5]]
```


Contribution of variables to Dim−2
Contributions (%)
20
15
10
5
0
Shot.put
Discus
400m
1500m
Long.jump
High.jump
Javeline
110m.hurdle
Pole.vault
100m

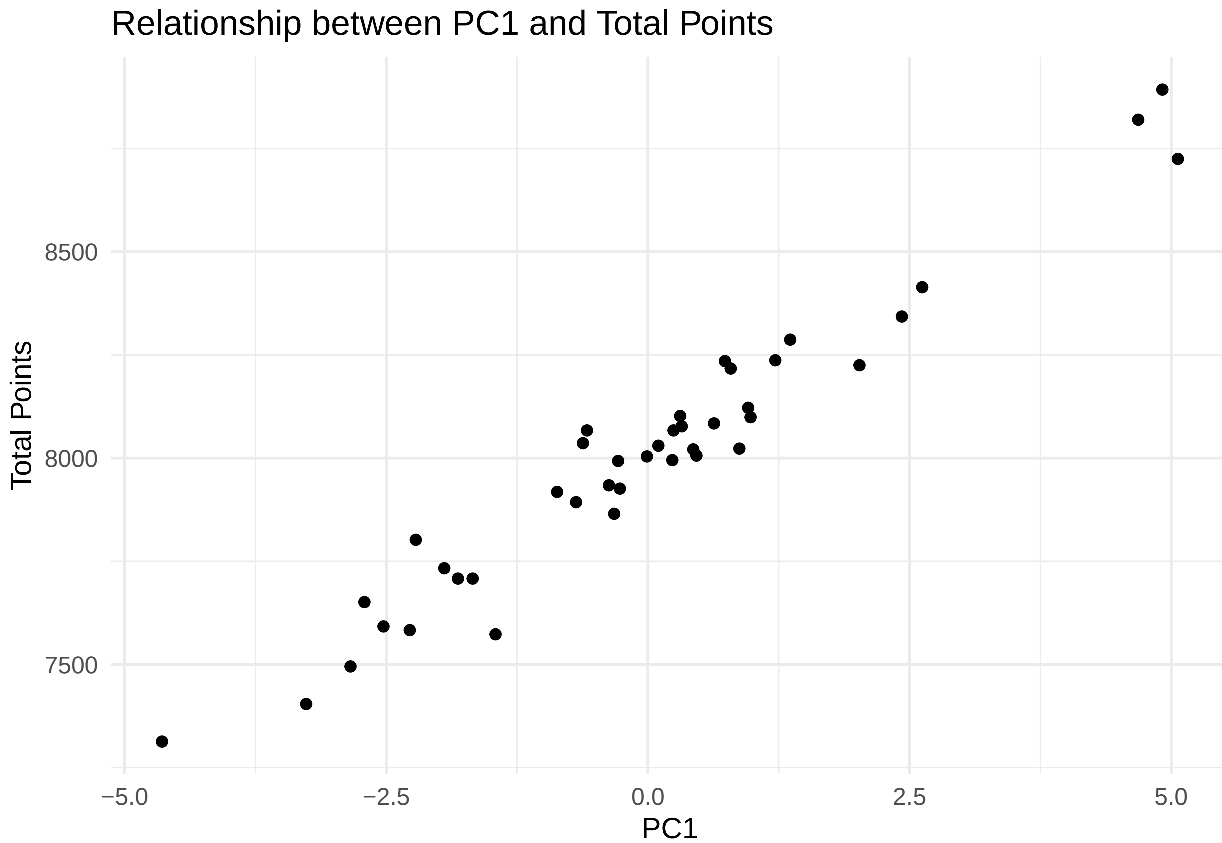
```
##
## [[6]]
```




```
##
## [[7]]
##          100m Long.jump Shot.put High.jump  400m 110m.hurdle Discus Pole.vault
## SEBRLE  11.04      7.58    14.83      2.07 49.81       14.69  43.75       5.02
## CLAY    10.76      7.40    14.26      1.86 49.37       14.05  50.72       4.92
## KARPOV  11.02      7.30    14.77      2.04 48.37       14.09  48.95       4.92
## BERNARD 11.02      7.23    14.25      1.92 48.93       14.99  40.87       5.32
## YURKOV  11.34      7.09    15.19      2.10 50.42       15.31  46.26       4.72
## WARNERS 11.11      7.60    14.31      1.98 48.68       14.23  41.10       4.92
##         Javeline 1500m Rank Points Competition         PC1        PC2
## SEBRLE     63.19 291.7    1   8217    Decastar  0.79121358  0.6501899
## CLAY       60.15 301.5    2   8122    Decastar  0.95795136  0.4491266
## KARPOV     50.31 300.2    3   8099    Decastar  0.98078350  0.3256933
## BERNARD    62.77 280.1    4   8067    Decastar -0.58244724 -0.9773933
## YURKOV     63.44 276.4    5   8036    Decastar -0.62046141  2.0614749
## WARNERS    51.77 278.1    6   8030    Decastar  0.09966088 -1.7968331
##
## [[8]]
```


Athletes' Scores on PC1 and PC2
PC2
PC1
4
2
0
−2
−5.0
−2.5
0.0
2.5
5.0
Rank
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28

```
##
## [[9]]
```



6. Summary

In summary, the application of PCA to the Decathlon Olympics dataset yielded valuable insights into athlete performance, event contributions, and underlying patterns. By reducing the dimensionality of the data, PCA enabled a clearer understanding of the relationships between athletes and events, ultimately enhancing our ability to interpret and analyze the dataset.