What are the most important factors behind successful decathlon athletes?

Michaela Kecskesova

19.5.2022

Introduction

The decathlon is a combined event in athletics consisting of ten track and field events - specifically 100-meter, 400-meter and 1500-meter runs, the 110-meter high hurdles, the javelin and discus throws, shot put, pole vault, high jump, and long jump. These events are held over two consecutive days and the winners are determined by the combined performance in all of them.

Traditionally, the title of "World's Greatest Athlete" has been given to the person who wins the decathlon. But what factors seem to be the determinants of success in a discipline consisting of so many individual events?

This project uses factor analysis to analyse data from men's olympic decathlon to assess what factors could possibly be the most important when determining the overall result.

The results are very intuitive - individual disciplines can be grouped to three factors, with disciplines under each factor requiring different approach to training - targeting strength and explosive power, upper body strength or endurance.

```r
# Necessary libraries
library(DescTools)
library(ellipse)
library(car)
library(psych)
```

Data

Data come from the results of the men's Decathlon at the Athens Summer Olympics, 2004. These include results of each discipline and also standardized scores of individual athletes along with total score, their country and an information about whether the athlete finished or not.

Data also includes some athletes that did not finish all of the events. These athletes were removed from the analysis so that their missing results would not bias the estimates.

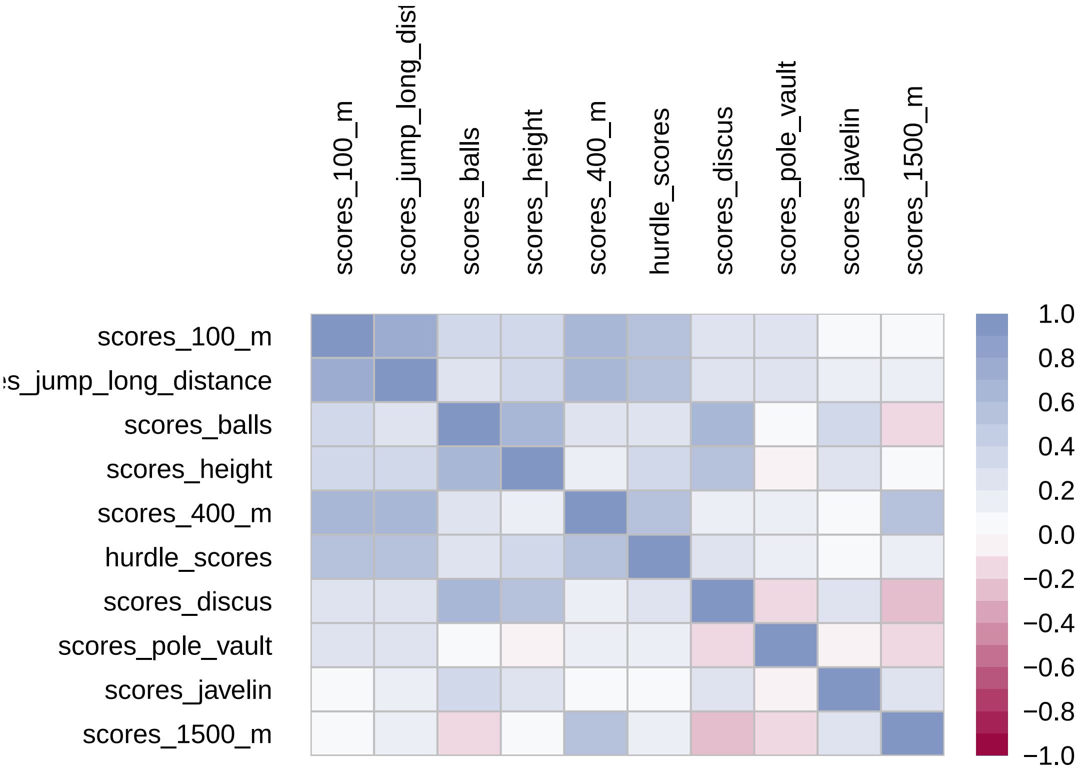
For the analysis, only the standardized scores were used as they are all on the same fixed scale and therefore are directly comparable.

```r
rm(list=ls())
load("Decathlon.RData")
# omit athletes who did not complete all disciplines
data <- na.omit(Decathlon)[,15:24]  # use only score variables 15-24
```

Analysis

```
# correlation matrix
(R <- cor(data,use="pairwise.complete.obs"))
```

From the correlation matrix, we can get a general idea about which of the events are correlated (the corresponding matrix element approaches -1 or 1), however, as the matrix is quite large for 10 variables, correlation plots might be more intuitive. Output from the correlation matrix was suppressed in this document due to its size.



455995/2022−05−19

From the correlation matrix and individual correlation plots (for two more see last section of this document - additional codes and figures) we can get the idea that some of the variables seem to be (mostly positively) correlated. This can be further formally tested by Bartlett's χ^2 test:

```
# is this an appropriate method?
# is correlation matrix siginificantly different from unit matrix?
cortest.bartlett(R, n = 100, diag = TRUE)
```

```
## $chisq
## [1] 468.9534
##
## $p.value
## [1] 6.161419e-72
##
## $df
## [1] 45
```

p-value of the test is extremely small. The test thus rejects the hypothesis that the correlation matrix is equal to unit matrix (in which case the individual variables would not be correlated and therefore performing factor analysis would lose its purpose). Usage of factor analysis is thus justified.

At first, the factors are considered to be equal with principal components. With the help of principal components, we can determine the appropriate number of factors behind our variables:

```
p<-prcomp(x=data, center=T,scale.=T)
a<-p$sdev^2
a
```

```
##  [1] 3.5592119 1.9529141 1.4265849 0.9053430 0.5587522 0.5315694 0.4328044
##  [8] 0.3657415 0.1646338 0.1024449
```

```
# percentage of total variance expl. by the first three components
sum(a[1:3])/10
```

```
## [1] 0.6938711
```

```
# percentage of total variance expl. by the first four components
sum(a[1:4])/10
```

```
## [1] 0.7844054
```

```
summary(p)
```

```
## Importance of components:
##                           PC1    PC2    PC3     PC4     PC5     PC6     PC7
## Standard deviation     1.8866 1.3975 1.1944 0.95150 0.74750 0.72909 0.65788
## Proportion of Variance 0.3559 0.1953 0.1427 0.09053 0.05588 0.05316 0.04328
## Cumulative Proportion  0.3559 0.5512 0.6939 0.78441 0.84028 0.89344 0.93672
##                            PC8     PC9    PC10
## Standard deviation     0.60477 0.40575 0.32007
## Proportion of Variance 0.03657 0.01646 0.01024
## Cumulative Proportion  0.97329 0.98976 1.00000
```

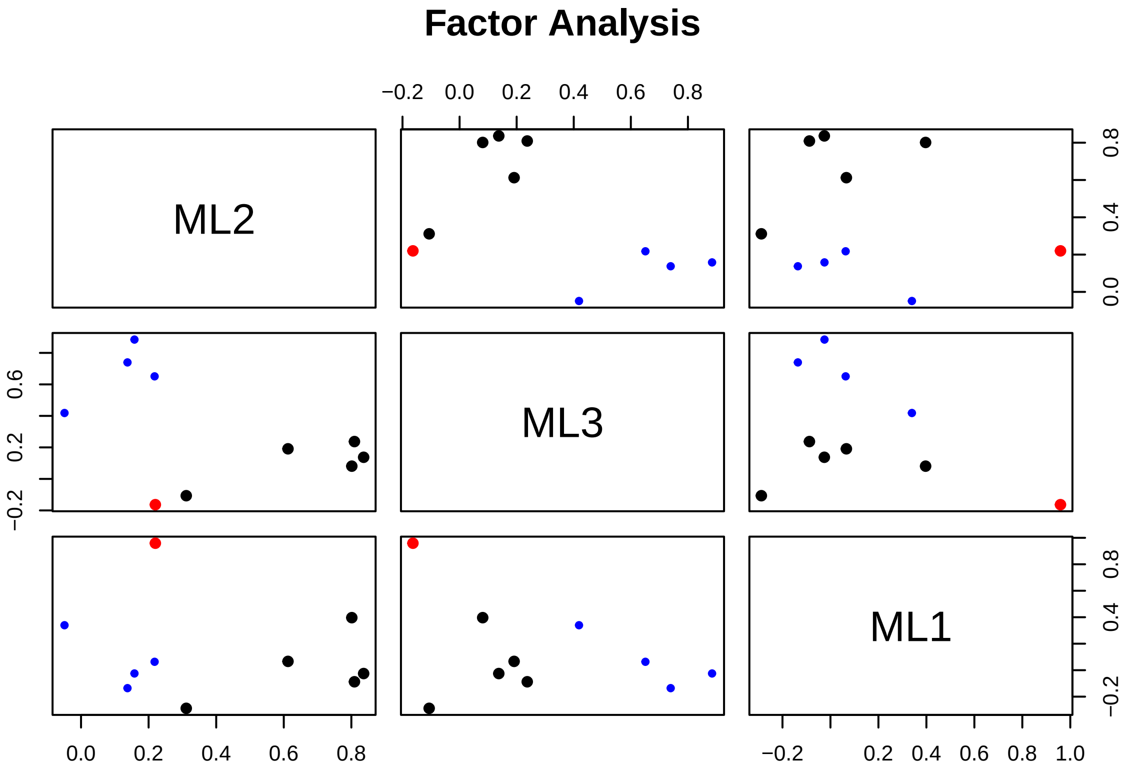
The cumulative proportion tells us, what percentage of variance would be explained by including the specific number of components. The row above it, the proportion of variance tells us, what additional percentage of variance would be explained by adding the one specific component.

Taking into account the calculated principal components, we will decide between using three or four factors (3-4 components explain majority of variance of the original variables).

```
### factors
f1<-fa(r=R,nfactors=3,rotate="varimax",fm="ml",scores="regression",residuals=T)
f2<-fa(r=R,nfactors=4,rotate="varimax",fm="ml",scores="regression",residuals=T)
```

Following plot can be used to indentify the individual factors and divide the variables to "clusters" according to the three factors. The individual observations in the graphs should not be strongly correlated (they should not be lined up forming a linear curve). If there was a significant correlation visible between two of the clusters, we could merge those factors into one, but that is not our case.

```
plot(f1)
```



```
# plot(f2) - more complicated output, not used in results
```

Now we have to interpret the computed factor loadings. If two variables both have large loadings for the same factor, then we know they have something in common.

```
# factor loadings
f1
```

```
## Factor Analysis using method =  ml
## Call: fa(r = R, nfactors = 3, rotate = "varimax", scores = "regression",
##     residuals = T, fm = "ml")
## Standardized loadings (pattern matrix) based upon correlation matrix
##                              ML2   ML3   ML1   h2    u2 com
## scores_100_m                0.81  0.24 -0.09 0.72 0.281 1.2
## scores_jump_long_distance   0.84  0.14 -0.03 0.72 0.281 1.1
## scores_balls                0.16  0.88 -0.02 0.81 0.192 1.1
## scores_height               0.22  0.65  0.06 0.48 0.525 1.2
## scores_400_m                0.80  0.08  0.40 0.81 0.194 1.5
## hurdle_scores               0.61  0.19  0.07 0.42 0.584 1.2
## scores_discus               0.14  0.74 -0.14 0.58 0.416 1.1
## scores_pole_vault           0.31 -0.11 -0.29 0.19 0.809 2.2
## scores_javelin             -0.05  0.42  0.34 0.29 0.707 2.0
## scores_1500_m               0.22 -0.16  0.96 1.00 0.005 1.2
##
##                        ML2  ML3  ML1
## SS loadings           2.61 2.08 1.31
## Proportion Var        0.26 0.21 0.13
## Cumulative Var        0.26 0.47 0.60
## Proportion Explained  0.43 0.35 0.22
## Cumulative Proportion 0.43 0.78 1.00
##
## Mean item complexity =  1.4
## Test of the hypothesis that 3 factors are sufficient.
##
## The degrees of freedom for the null model are  45  and the objective function was  4.95
## The degrees of freedom for the model are 18  and the objective function was  0.65
##
## The root mean square of the residuals (RMSR) is  0.05
## The df corrected root mean square of the residuals is  0.07
##
## Fit based upon off diagonal values = 0.98
## Measures of factor score adequacy
##                                                    ML2  ML3  ML1
## Correlation of (regression) scores with factors   0.94 0.93 0.99
## Multiple R square of scores with factors          0.88 0.86 0.98
## Minimum correlation of possible factor scores     0.77 0.72 0.96
```

```
# f2 - not used
```

According to factor loadings results for three and four factors, it would be better to use three factors, for better interpretability. From the results, we can see that there could be three distinct measures within the 10 decathlon disciplines.

Following the ML2, ML3 an ML1 columns in the output and by using the highest loading per item, we can extract information about which variables could be influenced by which factor.

Scores for 100m and 400m, along with hurdles, pole vault and the long distance jump are mainly correlated with one factor. Another factor is correlated mainly with the throwing disciplines - javelin throw, discus throw and ball throw, along with the height jump. Third factor is mostly correlated with the 1500m long run.

Conclusion

According to our data and analysis results, there could possibly be three important factors behind the success of the top decathlon athletes. First found factor was mostly associated with short-distance running disciplines such as sprints or hurdles. These are disciplines that usually require strength and muscles that are limber for short bursts of speed, as well as explosive force. Explosive training that targets strength and speed to increase power output could therefore be helpful for advancing in these disciplines.

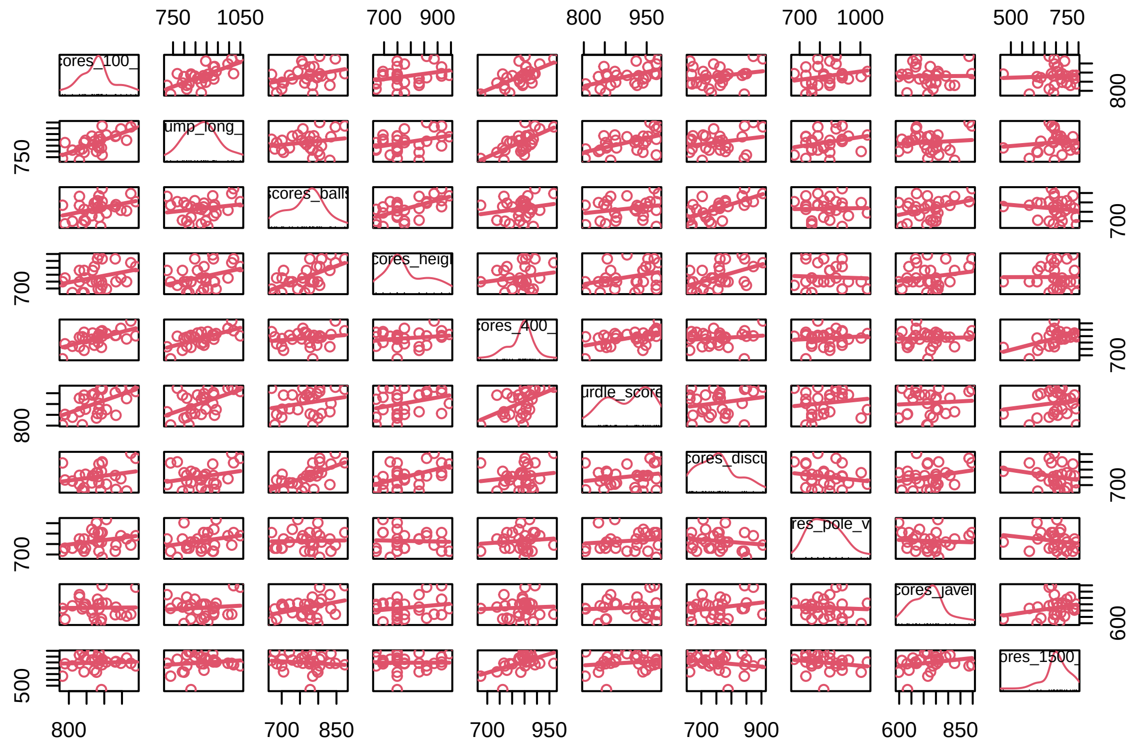
Second factor was associated with the throwing disciplines - those, on the other hand, require upper body strength and core stability (however, explosive upper body force is also important in this case). To improve performance in these events, training focused on upper body muscles could be beneficial.

Last factor was only correlated with one variable, which is, indeed, different than all the others - the 1500m run. This is the only discipline for which stamina and muscle endurance are of big importance as opposed to strength. Endurance-based training could therefore be used to target the 1500m run discipline.

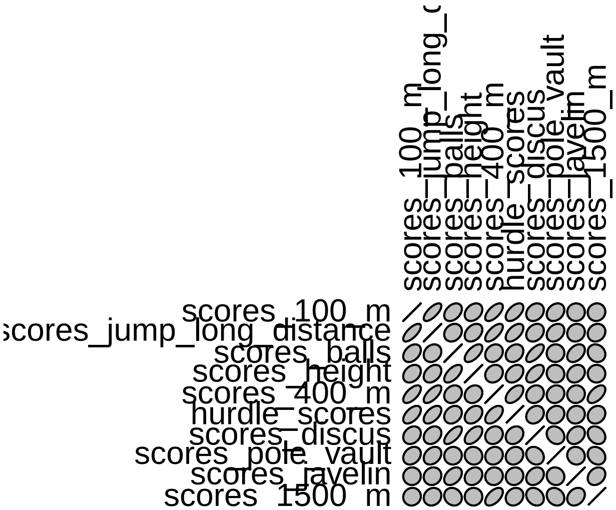
Additional codes and figures

```
# Residuals and factor scores
f1$residual
f1$weights
factor.scores(x=data,f1)$scores
```

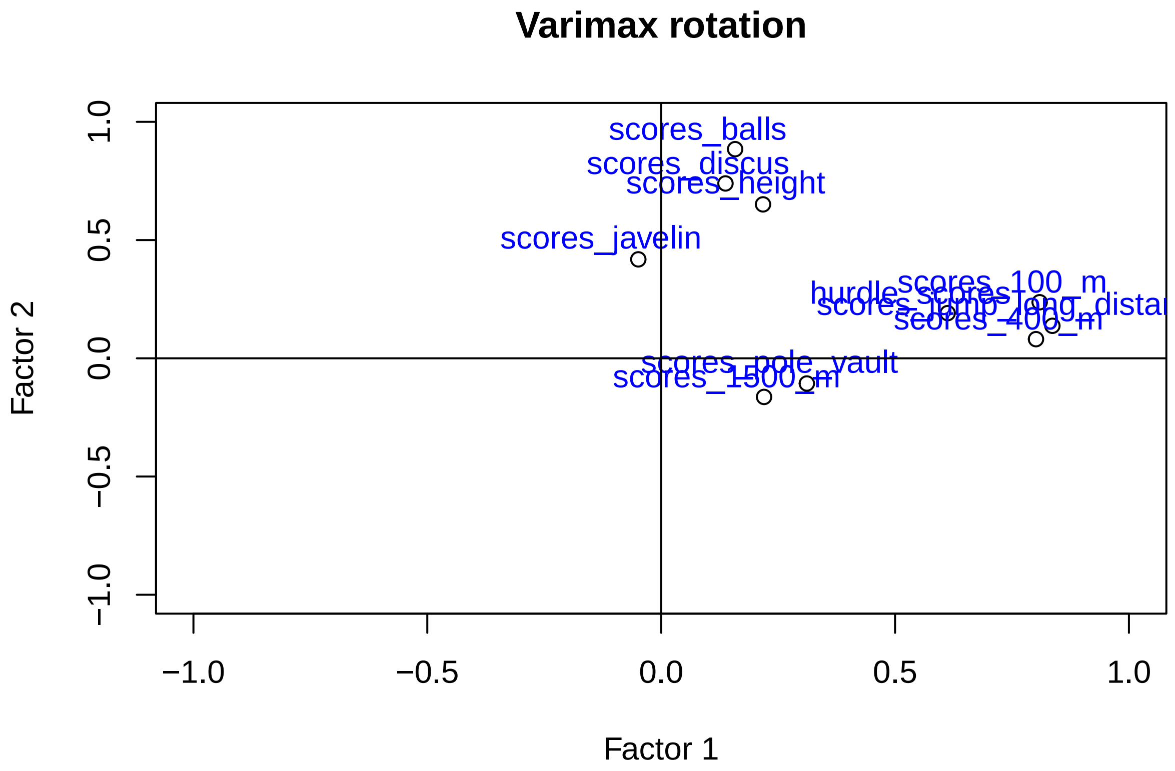
```
# Correlation matrices
scatterplotMatrix(data,smooth=F,diagonal="histogram",col=c(2,1,4))
```



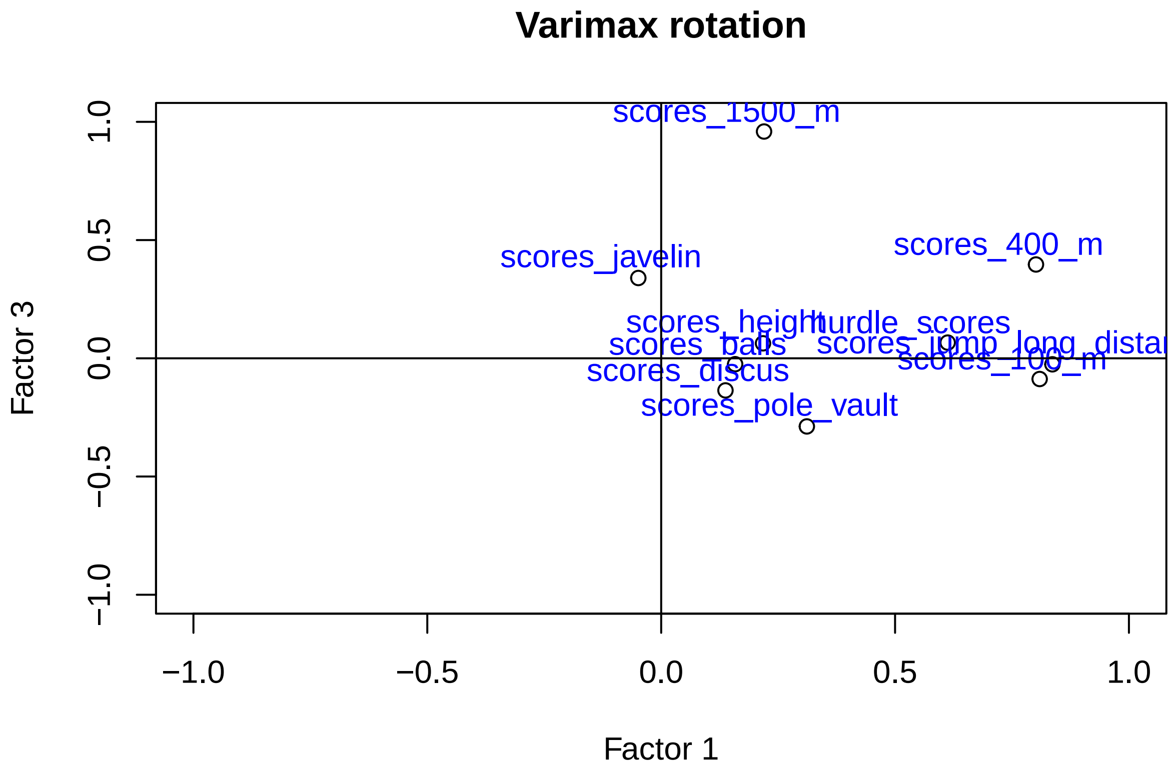
```
plotcorr(R)
```



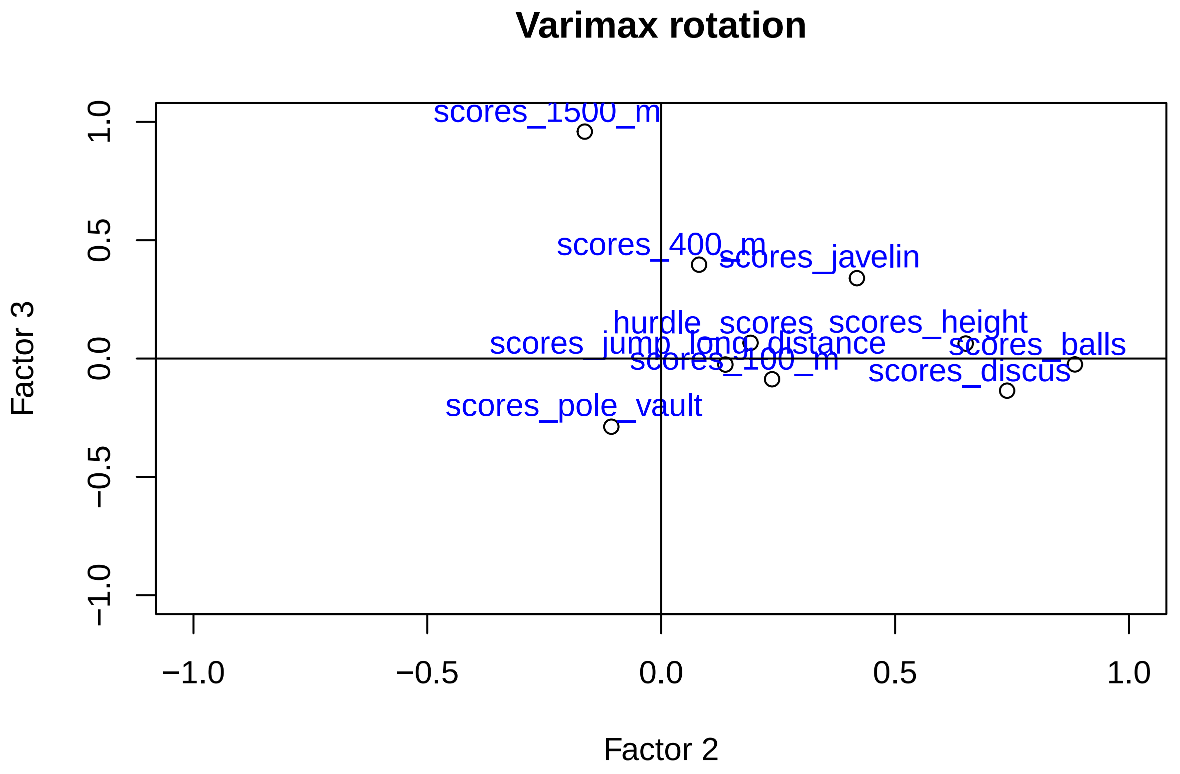
```r
# Scatterplot of factor loading combinations, varimax rotation
fa_varimax <- factanal(data, factors = 3, rotation = "varimax")

plot(fa_varimax$loadings[,1],
     fa_varimax$loadings[,2],
     xlab = "Factor 1",
     ylab = "Factor 2",
     ylim = c(-1,1),
     xlim = c(-1,1),
     main = "Varimax rotation")
text(fa_varimax$loadings[,1]-0.08,
     fa_varimax$loadings[,2]+0.08,
     colnames(data),
     col="blue")
abline(h = 0, v = 0)
```



```
plot(fa_varimax$loadings[,1],
     fa_varimax$loadings[,3],
     xlab = "Factor 1",
     ylab = "Factor 3",
     ylim = c(-1,1),
     xlim = c(-1,1),
     main = "Varimax rotation")
text(fa_varimax$loadings[,1]-0.08,
     fa_varimax$loadings[,3]+0.08,
     colnames(data),
     col="blue")
abline(h = 0, v = 0)
```



```
plot(fa_varimax$loadings[,2],
     fa_varimax$loadings[,3],
     xlab = "Factor 2",
     ylab = "Factor 3",
     ylim = c(-1,1),
     xlim = c(-1,1),
     main = "Varimax rotation")
text(fa_varimax$loadings[,2]-0.08,
     fa_varimax$loadings[,3]+0.08,
     colnames(data),
     col="blue")
abline(h = 0, v = 0)
```


Varimax rotation
Factor 3
Factor 2
1.0
0.5
0.0
−0.5
−1.0
−1.0
−0.5
0.0
0.5
1.0
scores_1500_m
scores_400_m
scores_javelin
hurdle_scores
scores_height
scores_jump_long_distance
scores_balls
scores_100_m
scores_discus
scores_pole_vault