A Handbook of Statistical Analyses Using R

Brian S. Everitt and Torsten Hothorn
CHAPTER 13

Principal Component Analysis: The Olympic Heptathlon

13.1 Introduction

13.2 Principal Component Analysis

13.3 Analysis Using R

To begin it will help to score all the seven events in the same direction, so that ‘large’ values are ‘good’. We will recode the running events to achieve this:

R> data("heptathlon", package = "HSAUR")
R> heptathlon$hurdles <- max(heptathlon$hurdles) -
          heptathlon$hurdles
R> heptathlon$run200m <- max(heptathlon$run200m) -
          heptathlon$run200m
R> heptathlon$run800m <- max(heptathlon$run800m) -
          heptathlon$run800m

Figure 13.1 shows a scatterplot matrix of the results from the 25 competitors on the seven events. We see that most pairs of events are positively correlated to a greater or lesser degree. The exceptions all involve the javelin event – this is the only really ‘technical’ event and it is clear that training to become successful in the other six ‘power’-based events makes this event difficult for the majority of the competitors. We can examine the numerical values of the correlations by applying the cor function

R> round(cor(heptathlon[, -score]), 2)

This correlation matrix demonstrates again the points made earlier.

A principal component analysis of the data can be applied using the prcomp function. The result is a list containing the coefficients defining each component (sometimes referred to as loadings), the principal component scores, etc.

The required code is (omitting the score variable)

R> heptathlon_pca <- prcomp(heptathlon[, -score], scale = TRUE)
R> print(heptathlon_pca)
PRINCIPAL COMPONENT ANALYSIS

```r
R> score <- which(colnames(heptathlon) == "score")
R> plot(heptathlon[, -score])
```

![Scatterplot matrix for the heptathlon data.](image)

Standard deviations (1, ..., p=7):

<p>| | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2.1119364</td>
<td>1.0928497</td>
<td>0.7218131</td>
<td>0.6761421</td>
<td>0.4952441</td>
<td>0.2701029</td>
<td>0.2213617</td>
</tr>
</tbody>
</table>

Rotation (n x k) = (7 x 7):

<table>
<thead>
<tr>
<th></th>
<th>PC1</th>
<th>PC2</th>
<th>PC3</th>
<th>PC4</th>
<th>PC5</th>
<th>PC6</th>
<th>PC7</th>
</tr>
</thead>
<tbody>
<tr>
<td>hurdles</td>
<td>-0.4526710</td>
<td>-0.15792058</td>
<td>-0.04514996</td>
<td>0.02653873</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>highjump</td>
<td>-0.3771992</td>
<td>0.24807386</td>
<td>-0.36777802</td>
<td>0.67998172</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>shot</td>
<td>-0.3630725</td>
<td>-0.28940743</td>
<td>0.67618919</td>
<td>0.12431725</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>run200m</td>
<td>-0.4078950</td>
<td>-0.26038545</td>
<td>0.08359211</td>
<td>-0.36106580</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>longjump</td>
<td>-0.4562318</td>
<td>0.05587394</td>
<td>0.13931653</td>
<td>0.1129249</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>javelin</td>
<td>-0.0754090</td>
<td>-0.84169212</td>
<td>-0.47156621</td>
<td>0.12079924</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>run800m</td>
<td>-0.3749594</td>
<td>0.22448984</td>
<td>-0.39585671</td>
<td>-0.60341330</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>hurdles</th>
<th>highjump</th>
<th>shot</th>
<th>run200m</th>
<th>longjump</th>
<th>javelin</th>
<th>run800m</th>
</tr>
</thead>
<tbody>
<tr>
<td>hurdles</td>
<td>-0.49494792</td>
<td>-0.78334201</td>
<td>0.38024707</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>highjump</td>
<td>0.18779888</td>
<td>0.09939981</td>
<td>-0.43393114</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>shot</td>
<td>0.51165201</td>
<td>-0.05085983</td>
<td>-0.21762491</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>run200m</td>
<td>-0.64993404</td>
<td>0.02495639</td>
<td>-0.45338483</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
ANALYSIS USING R

longjump -0.18429810 0.59020972 0.61206388
javelin 0.13510669 -0.02724076 0.17294667
run800m 0.50432116 0.15555520 -0.09830963

The summary method can be used for further inspection of the details:
R> summary(heptathlon_pca)

Importance of components %:

PC1 PC2 PC3 PC4 PC5
Standard deviation 2.1119 1.0928 0.72181 0.67614 0.49524
Proportion of Variance 0.6372 0.1706 0.07443 0.06531 0.03504
Cumulative Proportion 0.6372 0.8078 0.88223 0.94754 0.98258
PC6 PC7
Standard deviation 0.27010 0.2214
Proportion of Variance 0.01042 0.0070
Cumulative Proportion 0.99300 1.0000

The linear combination for the first principal component is
R> a1 <- heptathlon_pca$rotation[,1]
R> a1

<table>
<thead>
<tr>
<th>hurdles</th>
<th>highjump</th>
<th>shot</th>
<th>run200m</th>
<th>longjump</th>
<th>javelin</th>
<th>run800m</th>
<th>-0.4528710</th>
<th>-0.3771992</th>
<th>-0.3630725</th>
<th>-0.4078950</th>
<th>-0.4562318</th>
<th>-0.0754090</th>
<th>-0.3749594</th>
</tr>
</thead>
</table>

We see that the 200m and long jump competitions receive the highest weight but the javelin result is less important. For computing the first principal component, the data need to be rescaled appropriately. The center and the scaling used by `prcomp` internally can be extracted from the `heptathlon_pca` via

R> center <- heptathlon_pca$center
R> scale <- heptathlon_pca$scale

Now, we can apply the `scale` function to the data and multiply with the loadings matrix in order to compute the first principal component score for each competitor

R> hm <- as.matrix(heptathlon[,,-score])
R> drop(scale(hm, center = center, scale = scale) %*% heptathlon_pca$rotation[,1])

Joyner-Kersee (USA) John (GDR) Behmer (GDR)
-4.121447626 -2.882185935 -2.649633766

Sablovskaite (URS) Choubenkova (URS) Schuls (GDR)
-1.100385629 -0.923173639 -0.530250689

Weingartner (HOL) Dimitrova (BUL)
-0.759819024 -0.556268302 -1.186453832

Schoelder (SWI) Braun (FRG) Roostalainen (FIN)
0.015461226 0.003774223 0.09047709

Yuping (CHN) Hagger (GB) Brown (USA)
-0.137225440 0.171128651 0.519252646

Hull (NTU) Hultenaue (BEL) Rytela (FIN)
1.125481333 1.085697646 1.447055499

or, more conveniently, by extracting the first from all precomputed principal components
The first two components account for 81% of the variance. A barplot of each component’s variance (see Figure 13.2) shows how the first two components dominate. A plot of the data in the space of the first two principal components, with the points labelled by the name of the corresponding competitor can be produced as shown with Figure 13.3. In addition, the first two loadings for the events are given in a second coordinate system, also illustrating the special role of the javelin event. This graphical representation is known as biplot (?)..

The correlation between the score given to each athlete by the standard scoring system used for the heptathlon and the first principal component score can be found from

```
R> cor(heptathlon$score, heptathlon_pca$x[,1])
[1] -0.9910978
```

This implies that the first principal component is in good agreement with the score assigned to the athletes by official Olympic rules; a scatterplot of the official score and the first principal component is given in Figure 13.4.
R> plot(heptathlon_pca)

Figure 13.2  Barplot of the variances explained by the principal components.
Figure 13.3  Biplot of the (scaled) first two principal components.
Figure 13.4 Scatterplot of the score assigned to each athlete in 1988 and the first principal component.