PRINCIPAL COMPONENT ANALYSIS

P. Anipa

Loading and familiarizing with the data.

The data used in this covariance based Principal Component Analysis contains results of 48 decathletes from 1973.

```
data = read.table("decathlon.txt", header = TRUE, sep = "\t", row.names = 1)
head(data)
```

```
##          Points R100m Long_jump Shot_put High_jump R400m Hurdles Discus_throw
## Skowrone   8206   853       931      725       857   838     903          772
## Hedmark    8188   853       853      814       769   833     914          855
## Le_Roy     8140   879       951      799       779   838     881          819
## Zeilbaue   8136   826       931      793       865   875     891          729
## Zigert     8134   879       840      924       857   788     892          866
## Bennett    8121   905       859      647       779   938     859          651
##          Pole_vault Javelin R1500m Height Weight
## Skowrone        981     818    528    184     81
## Hedmark         884     975    438    195     90
## Le_Roy         1028     758    408    191     90
## Zeilbaue        909     774    543    192     84
## Zigert          920     671    497    198    105
## Bennett        1028     794    661    173     68
```

```
View(data)
dim(data)
```

```
## [1] 48 13
```

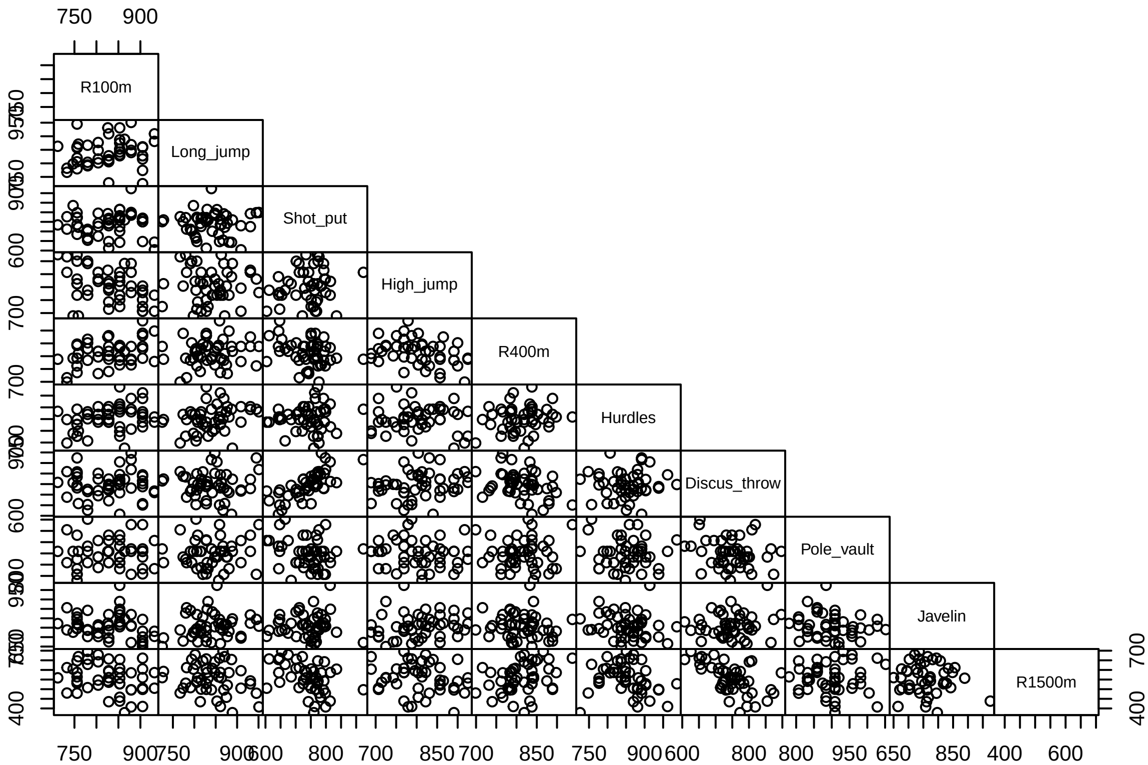
We want to conduct the analyses without the variables points, height and weight so we remove these variables from the data.

```
colnames(data)
```

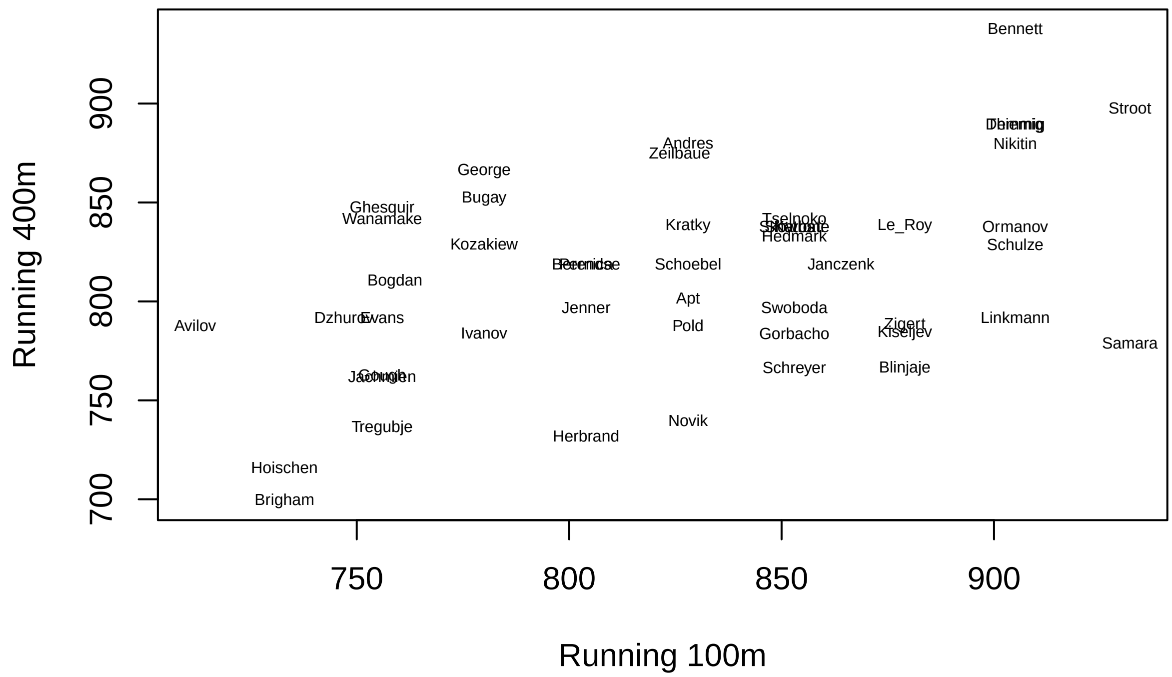
```
##  [1] "Points"       "R100m"        "Long_jump"    "Shot_put"     "High_jump"
##  [6] "R400m"        "Hurdles"      "Discus_throw" "Pole_vault"   "Javelin"
## [11] "R1500m"       "Height"       "Weight"
```

```
data_rem = data[ , -c(1, 12, 13)]
View(data_rem)

#Visualizing the data with a pairwise scatterplot.
pairs(data_rem, gap = 0, upper.panel = NULL)
```




```
#Visualizing just one of the scatterplots where the data points are replaced with the names of the deca
plot(data_rem$R100m, data_rem$R400m, xlab = "Running 100m", ylab = "Running 400m", type="n")
text(data_rem$R100m, data_rem$R400m, labels = rownames(data_rem), cex = 0.5)
```



Performing a covariance matrix based Principal Component Analysis(PCA) using the 'princomp' function.

We wish to answer the question, "How much of the variation of the original data is explained by the k principal components?"

```r
#Set argument 'cor' to FALSE to show that we perform the PCA with the covariance matrix.
data_pca = princomp(data_rem, cor = FALSE )

#Function princomp returns an object of class princomp, that is essentially a list of objects.
names(data_pca)
```

```
## [1] "sdev"     "loadings" "center"   "scale"    "n.obs"    "scores"   "call"
```

The proportion of the total variation explained by the first k principal can be seen straight away with the summary function. See the Cumulative Proportion row in the summary.

```r
summary(data_pca)
```

```
## Importance of components:
##                            Comp.1     Comp.2     Comp.3     Comp.4      Comp.5
## Standard deviation     102.9950759 83.8361146 63.9902194 63.4804991 58.06221588
## Proportion of Variance   0.2900506  0.1921778  0.1119613  0.1101848  0.09217818
## Cumulative Proportion    0.2900506  0.4822284  0.5941898  0.7043745  0.79655273
##                            Comp.6      Comp.7      Comp.8      Comp.9
## Standard deviation     47.3544471 43.07927681 39.76028470 30.35519704
## Proportion of Variance  0.0613144  0.05074318  0.04322549  0.02519457
```

```
## Cumulative Proportion   0.8578671  0.90861031  0.95183579  0.97703037
##                            Comp.10
## Standard deviation     28.98388394
## Proportion of Variance  0.02296963
## Cumulative Proportion   1.00000000
```

We can also calculate proportions of variation explained by the first k principal components manually.

```
vars = data_pca$sdev^2
var_prop = vars / sum(vars)
var_prop_cum = cumsum(var_prop)

#Proportion of variance
var_prop
```

```
##     Comp.1     Comp.2     Comp.3     Comp.4     Comp.5     Comp.6     Comp.7
## 0.29005064 0.19217779 0.11196134 0.11018477 0.09217818 0.06131440 0.05074318
##     Comp.8     Comp.9    Comp.10
## 0.04322549 0.02519457 0.02296963
```

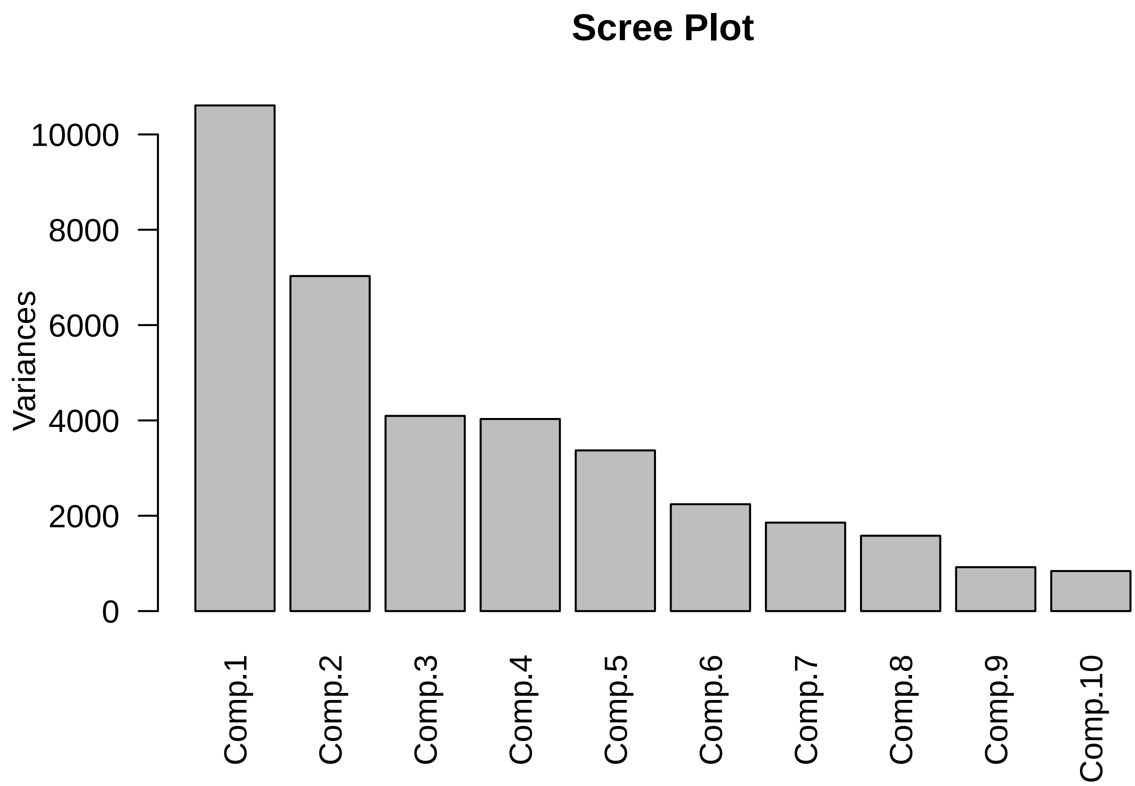
```
#Cumulative proportion
var_prop_cum
```

```
##    Comp.1    Comp.2    Comp.3    Comp.4    Comp.5    Comp.6    Comp.7    Comp.8
## 0.2900506 0.4822284 0.5941898 0.7043745 0.7965527 0.8578671 0.9086103 0.9518358
##    Comp.9   Comp.10
## 0.9770304 1.0000000
```

The 'scree plot' of the principal components.

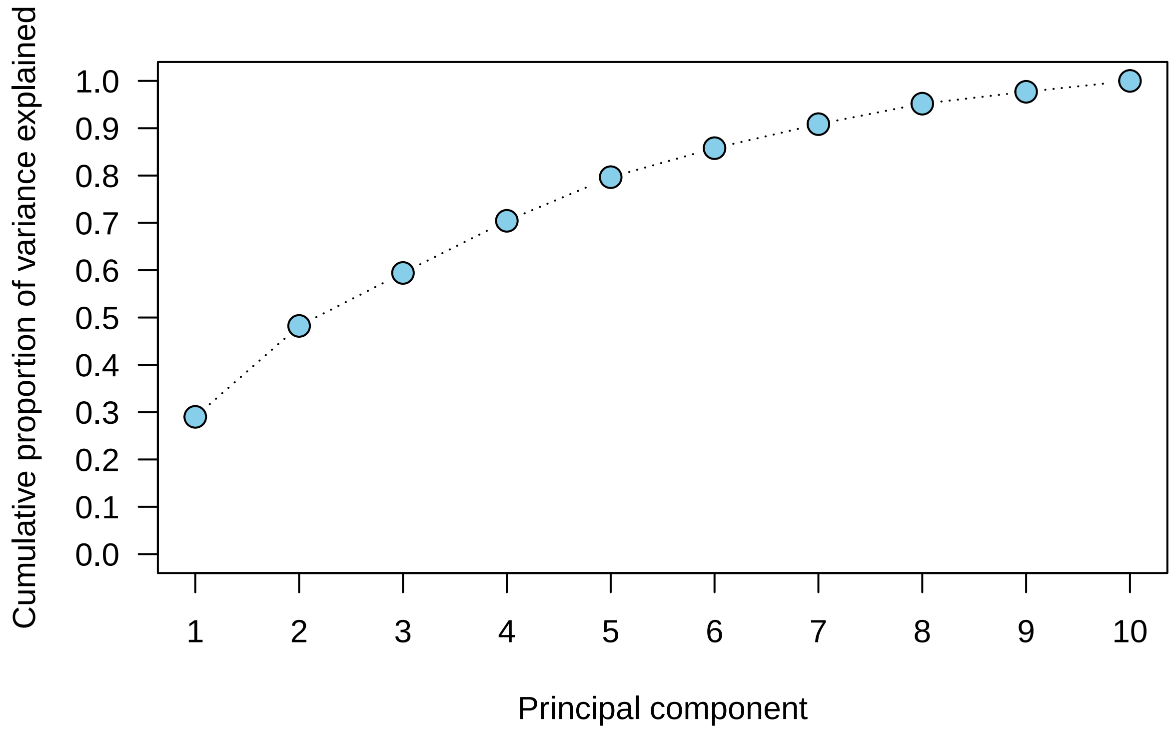
Scree plot can be used as a tool for choosing sufficient number of components.

```
plot(data_pca, las = 2, main = "Scree Plot")
```



The following plot can be also useful for choosing how many principal components to use.

```
plot(var_prop_cum, type = "b", pch = 21, lty = 3, bg = "skyblue", cex = 1.5,
     ylim = c(0, 1), xlab = "Principal component",
     ylab = "Cumulative proportion of variance explained",
     xaxt = "n", yaxt = "n")
axis(1, at = 1:10)
axis(2, at = 0:10 / 10, las = 2)
```



There are many more or less heuristic methods for choosing number of principal components, for example, • include enough components to explain x% of total variation, where x% can chosen to be, e.g. 90%, • Kaiser criterion: include those principal components whose eigenvalues are larger than average, • elbow method,etc

Biplot of scores and loadings.

We choose the first four principal components and try to interpret them. Together the first four components explain approximately 70% of the variation in the original data.

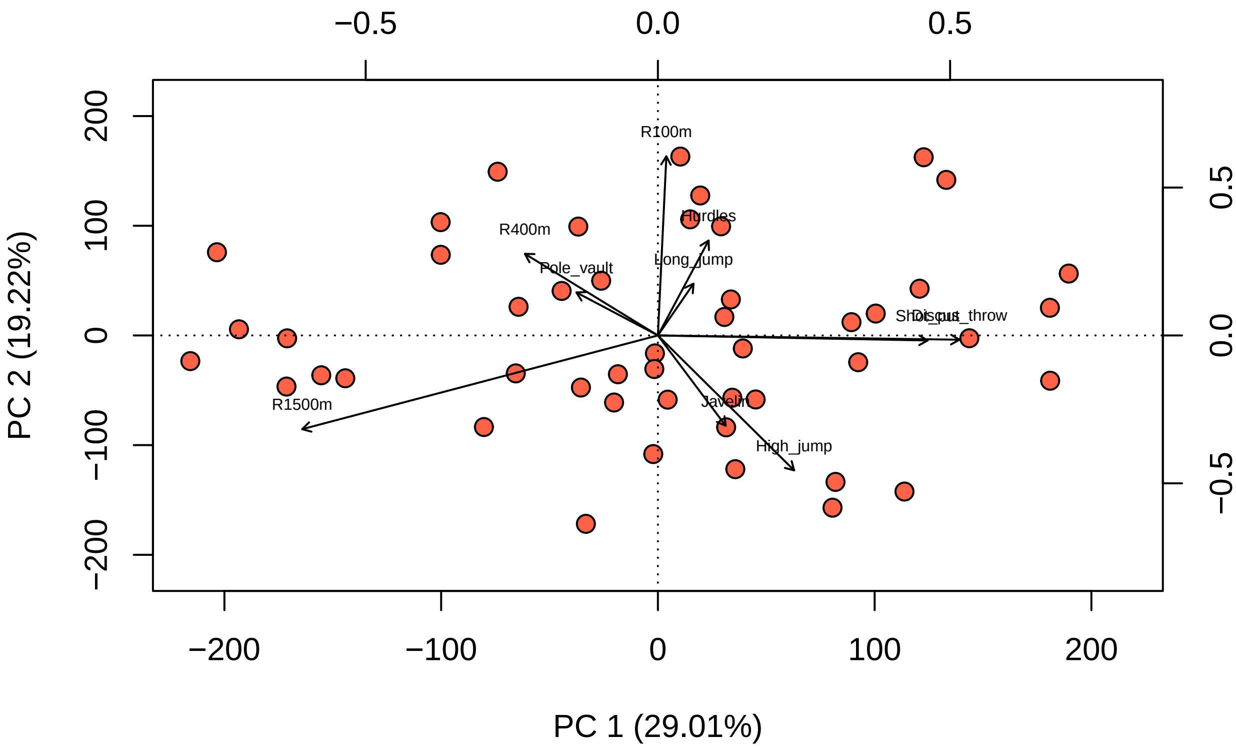
We plot the first two principal components and the corresponding loadings. There are two coordinate systems, one for the principal components and other for the loadings. Loadings and biplots are used to interpret principal components. Mainly, we want to find the variables that contribute to the principal components

```
score = data_pca$scores
load = data_pca$loadings

pc12 = score[, 1:2]
load12 = load[, 1:2]
pc_axis = c(-max(abs(pc12)), max(abs(pc12)))
ld_axis = c(-0.8, 0.8)

plot(pc12, xlim = pc_axis, ylim = pc_axis, pch = 21, bg = "tomato", cex = 1.25,
     xlab = paste0("PC 1 (", round(100 * var_prop[1], 2), "%)"),
     ylab = paste0("PC 2 (", round(100 * var_prop[2], 2), "%)"))
par(new = TRUE)
plot(load12, axes = FALSE, type = "n", xlab = "", ylab = "", xlim = ld_axis,
     ylim = ld_axis)
```

```r
axis(3)
axis(4)
arrows(0, 0, load12[, 1], load12[, 2], length = 0.05)
text(load12[, 1], load12[, 2], rownames(load12), pos = 3, cex = 0.5)
abline(h = 0, lty = 3)
abline(v = 0, lty = 3)
```




```r
#The 'biplot' function gives similar results
#biplot(data_pca)
```

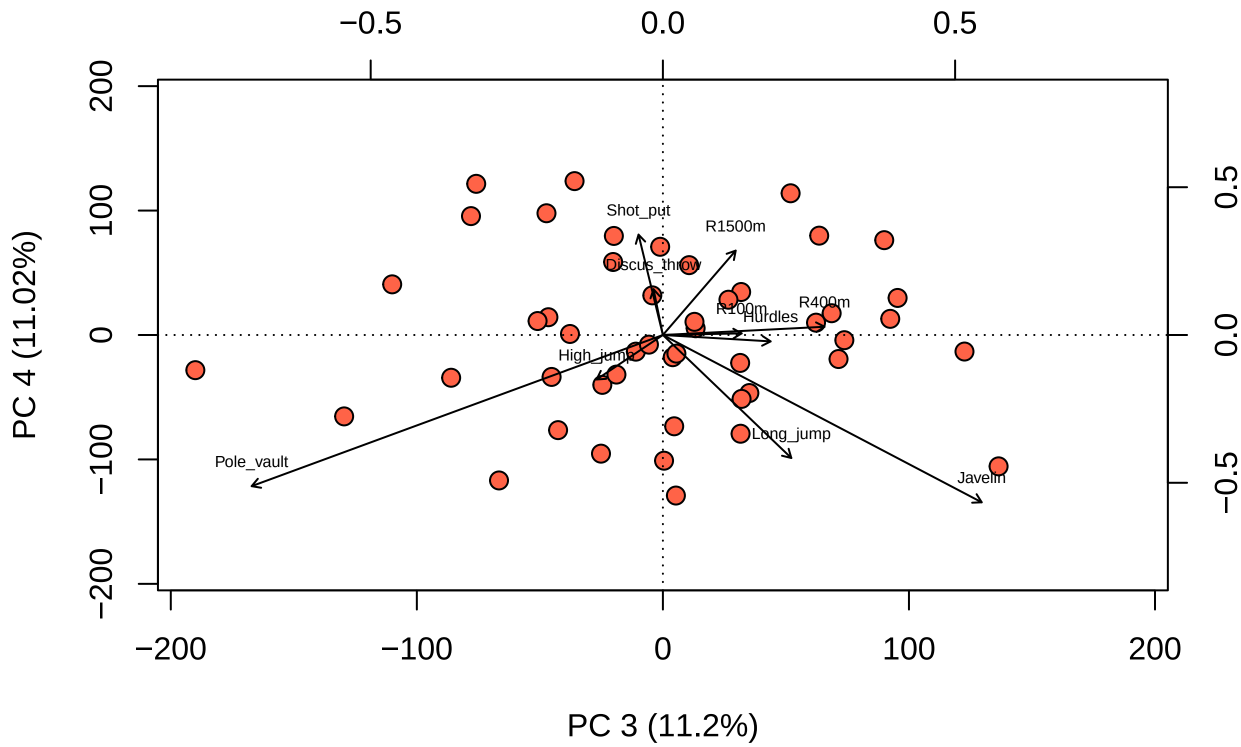
From the above plot, variables Discus_throw and Shot_put have the most significant positive contributions to the first principal component(PC 1). On the other hand R1500m has significant negative contribution to the first component. Therefore the first principal components tells that the decathletes who are good at running long distances are very different compared to the decathletes who are good at discus throw and shot put. Consequently we could interpret first principal component as strength/bulkiness.

For the second component, the variables such as R100m, Hurdles and R400m have significant positive contributions to the second principal component. Whereas the variables High_jump, R1500m and Javelin have significant negative contributions to the second component. Thus the second component can be interpreted as speed.

We plot the second two principal components and the corresponding loadings.

```r
pc34 = score[, 3:4]
load34 = load[, 3:4]
pc_axis = c(-max(abs(pc34)), max(abs(pc34)))
ld_axis = c(-0.8, 0.8)
```

```r
plot(pc34, xlim = pc_axis, ylim = pc_axis, pch = 21, bg = "tomato", cex = 1.25,
     xlab = paste0("PC 3 (", round(100 * var_prop[3], 2), "%)"),
     ylab = paste0("PC 4 (", round(100 * var_prop[4], 2), "%)"))
par(new = TRUE)
plot(load34, axes = FALSE, type = "n", xlab = "", ylab = "", xlim = ld_axis,
     ylim = ld_axis)
axis(3)
axis(4)
arrows(0, 0, load34[, 1], load34[, 2], length = 0.05)
text(load34[, 1], load34[, 2], rownames(load34), pos = 3, cex = 0.5)
abline(h = 0, lty = 3)
abline(v = 0, lty = 3)
```



The sample mean and covariance matrix from the score matrix.

Mean vector corresponding to principal components is a zero vector.

```r
round(colMeans(score), 2)
```

```
##   Comp.1   Comp.2   Comp.3   Comp.4   Comp.5   Comp.6   Comp.7   Comp.8   Comp.9 Comp.10
##        0        0        0        0        0        0        0        0        0       0
```

Principal components are uncorrelated, thus the sample covariance matrix is a diagonal matrix.

```r
round(cov(score), 2)
```

```
##             Comp.1   Comp.2   Comp.3   Comp.4   Comp.5   Comp.6   Comp.7   Comp.8 Comp.9
```

```
## Comp.1  10833.69    0.00    0.00    0.00    0.00    0.00    0.00    0.00   0.00
## Comp.2       0.00 7178.04    0.00    0.00    0.00    0.00    0.00    0.00   0.00
## Comp.3       0.00    0.00 4181.87    0.00    0.00    0.00    0.00    0.00   0.00
## Comp.4       0.00    0.00    0.00 4115.51    0.00    0.00    0.00    0.00   0.00
## Comp.5       0.00    0.00    0.00    0.00 3442.95    0.00    0.00    0.00   0.00
## Comp.6       0.00    0.00    0.00    0.00    0.00 2290.16    0.00    0.00   0.00
## Comp.7       0.00    0.00    0.00    0.00    0.00    0.00 1895.31    0.00   0.00
## Comp.8       0.00    0.00    0.00    0.00    0.00    0.00    0.00 1614.52   0.00
## Comp.9       0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00 941.04
## Comp.10      0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00   0.00
##         Comp.10
## Comp.1     0.00
## Comp.2     0.00
## Comp.3     0.00
## Comp.4     0.00
## Comp.5     0.00
## Comp.6     0.00
## Comp.7     0.00
## Comp.8     0.00
## Comp.9     0.00
## Comp.10  857.94
```