Problem Set 2: PCA

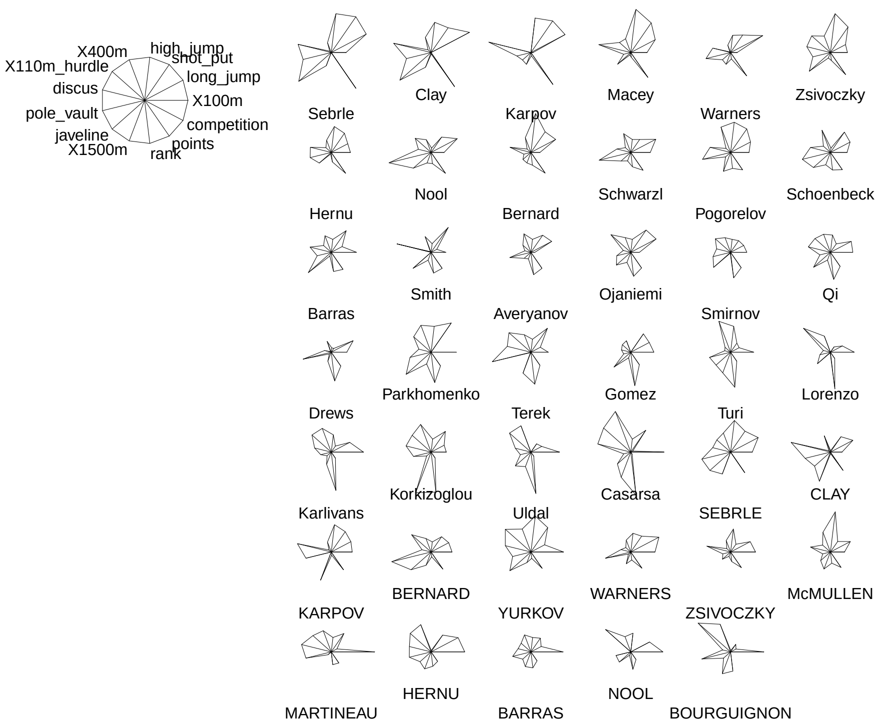
Donggyun Kim
27008257
Lab 102

2/16/2018

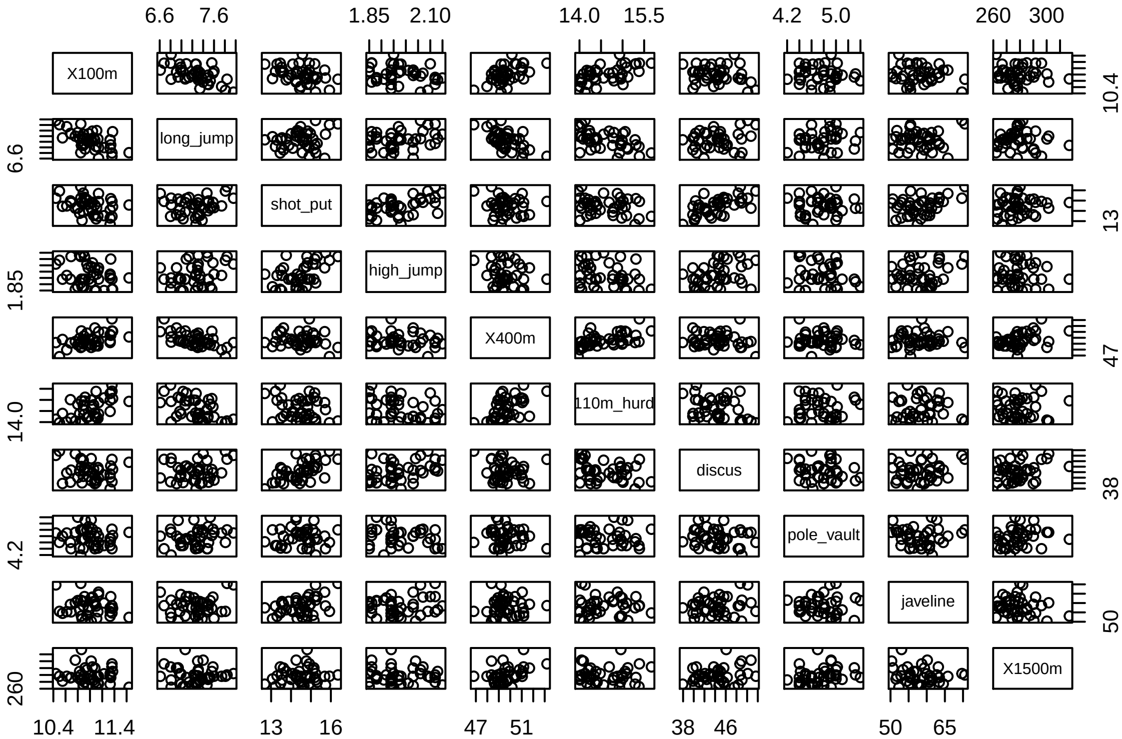
Exploratory Phase

```
dat <- read.csv("data/decathlon.csv", stringsAsFactors = FALSE)
stars(dat[, -1], full = TRUE, scale = TRUE, labels = as.character(dat$athlete),
      key.loc = c(-2, 15), key.labels = colnames(dat)[-1], cex = 0.5)
```

```
## Warning in data.matrix(x): NAs introduced by coercion

## Warning in min(x, na.rm = TRUE): no non-missing arguments to min; returning
## Inf

## Warning in min(x): no non-missing arguments to min; returning Inf

## Warning in max(x): no non-missing arguments to max; returning -Inf
```



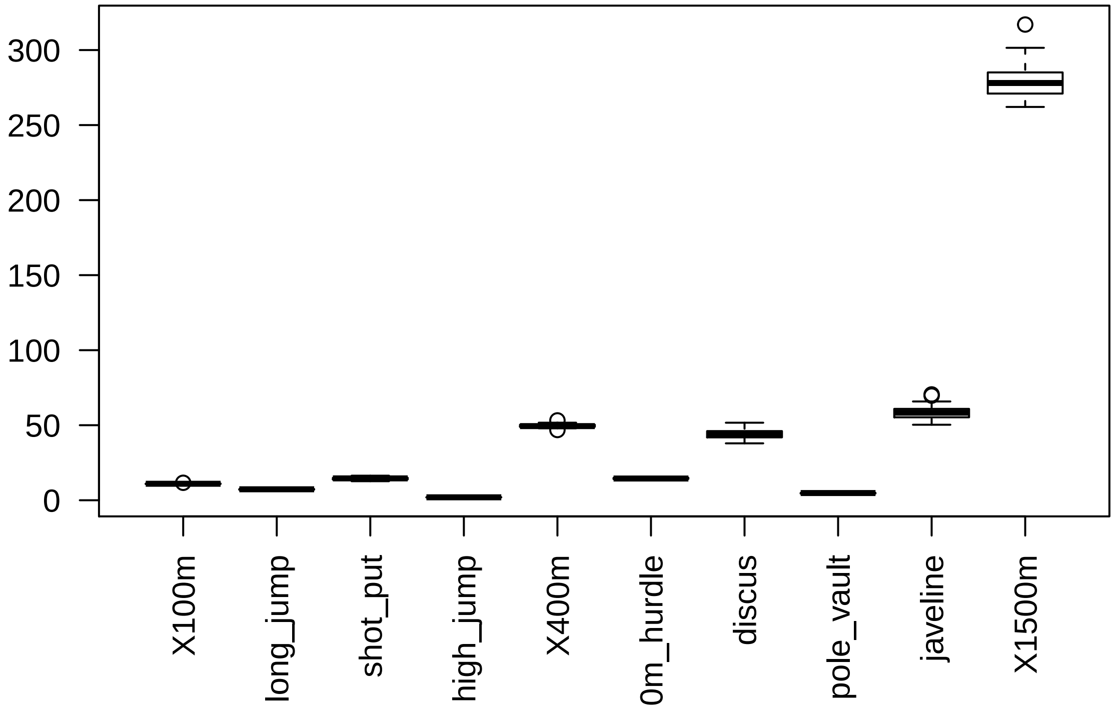

```
pairs(dat[, c(-1, -12:-14)])
```




```r
summary(dat)
```

```
##    athlete              X100m         long_jump       shot_put
##  Length:41          Min.   :10.44   Min.   :6.61   Min.   :12.68
##  Class :character   1st Qu.:10.85   1st Qu.:7.03   1st Qu.:13.88
##  Mode  :character   Median :10.98   Median :7.30   Median :14.57
##                     Mean   :11.00   Mean   :7.26   Mean   :14.48
##                     3rd Qu.:11.14   3rd Qu.:7.48   3rd Qu.:14.97
##                     Max.   :11.64   Max.   :7.96   Max.   :16.36
##    high_jump         X400m        X110m_hurdle       discus
##  Min.   :1.850   Min.   :46.81   Min.   :13.97   Min.   :37.92
##  1st Qu.:1.920   1st Qu.:48.93   1st Qu.:14.21   1st Qu.:41.90
##  Median :1.950   Median :49.40   Median :14.48   Median :44.41
##  Mean   :1.977   Mean   :49.62   Mean   :14.61   Mean   :44.33
##  3rd Qu.:2.040   3rd Qu.:50.30   3rd Qu.:14.98   3rd Qu.:46.07
##  Max.   :2.150   Max.   :53.20   Max.   :15.67   Max.   :51.65
##    pole_vault       javeline         X1500m           rank
##  Min.   :4.200   Min.   :50.31   Min.   :262.1   Min.   : 1.00
##  1st Qu.:4.500   1st Qu.:55.27   1st Qu.:271.0   1st Qu.: 6.00
##  Median :4.800   Median :58.36   Median :278.1   Median :11.00
##  Mean   :4.762   Mean   :58.32   Mean   :279.0   Mean   :12.12
##  3rd Qu.:4.920   3rd Qu.:60.89   3rd Qu.:285.1   3rd Qu.:18.00
##  Max.   :5.400   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
```

```r
boxplot(dat[, c(-1, -12:-14)], las = 2)
```



1) Calculation of primary PCA outputs

```r
dat_act <- dat[1:28, 2:11]  # active individuals and variables

avg <- apply(dat_act, 2, mean)  # column mean of active data
scale <- apply(dat_act, 2, sd)  # column sd of active data

Xc <- sweep(dat_act, 2, avg, "-")  # mean-centered data
Xsd <- sweep(Xc, 2, scale, "/")  # standardized data

SVD <- svd(Xsd)

loadings <- SVD$v
rownames(loadings) <- names(Xsd)
colnames(loadings) <- paste0("v", 1:10)

loadings[, 1:4]  # first four loadings
```

```
##                       v1         v2          v3           v4
## X100m         0.42270533 -0.1806841  0.21199128 -0.075009372
## long_jump    -0.42146649  0.2315408 -0.13017356  0.006144987
## shot_put     -0.33407359 -0.4437320 -0.01889119 -0.140442615
## high_jump    -0.33249211 -0.3362530  0.01083254  0.111008069
## X400m         0.38995573 -0.3524322 -0.19266472 -0.116944533
## X110m_hurdle  0.37654258 -0.1655859  0.03684219 -0.115374735
## discus       -0.28793579 -0.4754243 -0.01497490  0.206205419
## pole_vault   -0.09539301  0.2324861 -0.52373161 -0.643167759
```

```
## javeline     -0.15213083 -0.2415176  0.43702142 -0.689806306
## X1500m        0.11193576 -0.3372567 -0.65852601  0.057300779
```

```
PCs <- as.matrix(Xsd) %*% loadings
rownames(PCs) <- dat$athlete[1:28]
colnames(PCs) <- paste0("PC", 1:10)

PCs[, 1:4]  # first four PCs
```

```
##                     PC1        PC2        PC3         PC4
## Sebrle      -3.64687853 -1.5046838  0.2162631 -1.74472299
## Clay        -3.60330295 -0.8537756 -0.3196647 -1.16403050
## Karpov      -4.20070330 -0.4155663 -0.3533370  1.70482900
## Macey       -1.90491357 -1.3402994  1.2384762  1.09465304
## Warners     -1.89845545  1.6971105 -0.8885198  0.49575117
## Zsivoczky   -0.69529257 -1.2474627  1.0235787 -0.53486534
## Hernu       -0.69023057  0.5068215  0.7320204  0.17198585
## Nool        -0.17778111  1.7420595 -0.9865815 -1.97322479
## Bernard     -1.57350593 -0.1338441  0.1139975  1.63517179
## Schwarzl     0.09249421  1.4510863 -0.7211613 -0.49054597
## Pogorelov   -0.25580109 -0.6051491 -1.7486821  0.22767233
## Schoenbeck   0.12114605  0.2872966 -0.5531648 -1.00087299
## Barras       0.28609389 -0.3817102  1.6529060 -0.61055310
## Smith       -0.47451303 -1.1087005  1.5460578  1.09545064
## Averyanov   -0.21829441  1.7113985 -0.5069687  0.28850069
## Ojaniemi    -0.11595075  0.7797977  0.1865277 -0.10490637
## Smirnov      0.62752609  1.0467549  1.2218190 -0.31568082
## Qi           0.72606940  0.1849499  1.0043829  0.34313787
## Drews        0.41555968  3.0780560 -0.8637160  0.54571271
## Parkhomenko  1.31164623 -1.8259536  0.7181978 -1.66622181
## Terek        0.89200732 -0.2586709 -2.3429373 -0.24618999
## Gomez        0.64388302  1.0754572  1.5068250  0.16939887
## Turi         1.80329186 -0.1925375 -0.8147291 -0.36824998
## Lorenzo      2.57797811  1.5344745  1.6135571  0.08449894
## Karlivans    2.31672132  0.1869460  0.1352429  1.17059258
## Korkizoglou  1.45766664 -1.7903823 -2.9486653  0.88518819
## Uldal        2.88307554 -0.1301175  0.5209506 -0.04088944
## Casarsa      3.30046389 -3.4933557 -0.3826751  0.34841043
```

```
eigenvalues <- SVD$d^2 / (nrow(Xsd) - 1)
eigenvalues
```

```
##  [1] 3.5446573 1.9699560 1.4217248 0.9034912 0.5636320 0.5282270 0.4328613
##  [8] 0.3658102 0.1634956 0.1061447
```

```
sum(eigenvalues)
```

```
## [1] 10
```

2) Choosing the number of dimensions to retain/examine

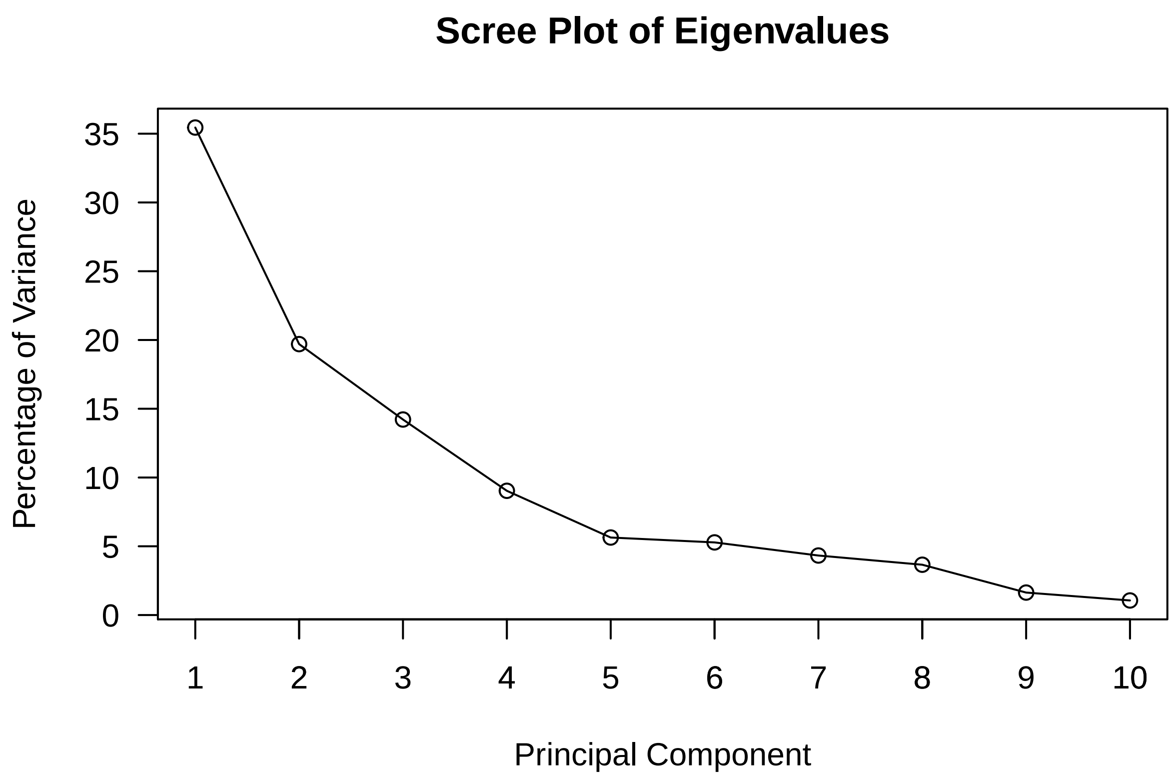
```
eigenvalue <- round(eigenvalues, 4)
percentage <- round(prop.table(eigenvalues) * 100, 4)
cumulative.percentage <- cumsum(percentage)
```

```r
# a summary table of the eigenvalues
cbind(eigenvalue, percentage, cumulative.percentage)
```

```
##       eigenvalue percentage cumulative.percentage
##  [1,]     3.5447    35.4466               35.4466
##  [2,]     1.9700    19.6996               55.1462
##  [3,]     1.4217    14.2172               69.3634
##  [4,]     0.9035     9.0349               78.3983
##  [5,]     0.5636     5.6363               84.0346
##  [6,]     0.5282     5.2823               89.3169
##  [7,]     0.4329     4.3286               93.6455
##  [8,]     0.3658     3.6581               97.3036
##  [9,]     0.1635     1.6350               98.9386
## [10,]     0.1061     1.0614              100.0000
```

If we use first four PCs to interpret the data, they explain about 78% of the variance in the data.

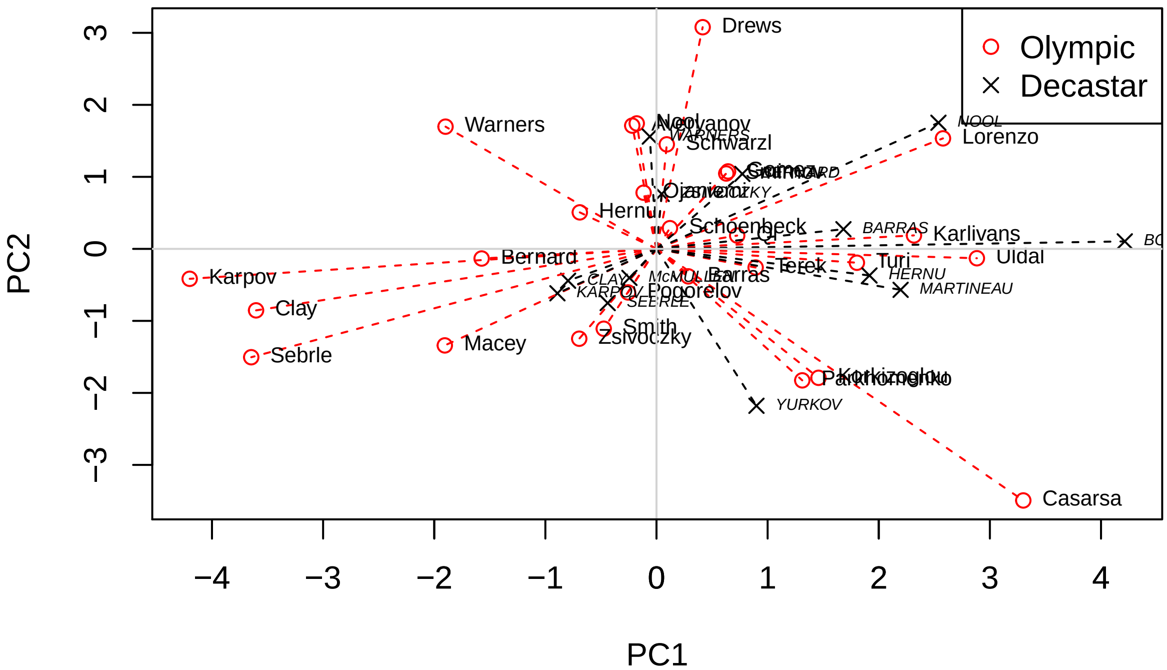
```r
# a scree-plot of the eigenvalues
plot(percentage, pch = 1, las = 1, xlab = "Principal Component",
     ylab = "Percentage of Variance", lty = 1,
     main = "Scree Plot of Eigenvalues")
lines(percentage)
axis(1, at = 1:10)
```



The first principal component(or eigenvalue) explains about 35% of the variance in the data, the second one explains about 20% of the variance in the data, and the third one explains about 13% of the variance in the data. I would like to use first three principal components that explain about 70% of the variance in the data because that might be enough to interpret the data set. If not, add one more component. I will repeat the same process until I will get what I want.

3) Studying the cloud of individuals

```r
dat_sup <- dat[29:41, 2:11]  # supplementary individuals

# standardize supplementary data with mean and sd of active data
Xc_sup <- sweep(dat_sup, 2, avg, "-")  # mean centered data
Xsd_sup <- sweep(Xc_sup, 2, scale, "/")  # standardized data

PCs_sup <- as.matrix(Xsd_sup) %*% loadings
rownames(PCs_sup) <- dat$athlete[29:41]
colnames(PCs_sup) <- paste0("PC", 1:10)

PCs_total <- rbind(PCs, PCs_sup)  # combine active and supplementary PCs


PC1 <- PCs_total[, 1]
PC2 <- PCs_total[, 2]
PC1_act <- PCs[, 1]
PC2_act <- PCs[, 2]
PC1_sup <- PCs_sup[, 1]
PC2_sup <- PCs_sup[, 2]

# a scatter plot of the athletes on the 1st and 2nd PCs
plot(PC1, PC2, col = factor(dat$competition),
     pch = c(4,1)[factor(dat$competition)])
segments(0, 0, PC1, PC2, col = factor(dat$competition), lty = 2)
text(PC1_act, PC2_act, labels = rownames(PCs), pos = 4,
     font = 1, cex = 0.7)
text(PC1_sup, PC2_sup, labels = rownames(PCs_sup), pos = 4,
     font = 3, cex = 0.5)
legend("topright", legend = unique(factor(dat$competition)),
       col = unique(factor(dat$competition)), pch = c(1,4))
abline(v = 0, h = 0, col = "lightgray")
axis(1, at = -4:4)
```



This graph shows how related for each individual to the first and second principal component. Most of the individuals seem to be related to the first principal component. Casarsa(individual) is highly related to the both components. Schoenbeck(individual) is hardly related to the both components.

```r
cos2 <- matrix(0, nrow = nrow(dat_act), ncol = ncol(dat_act))

for (i in 1:nrow(PCs)) {
  for (j in 1:ncol(PCs)) {
    cos2[i, j] <- (PCs[i, j])^2 / sum(Xsd[i, ]^2)
  }
}

rownames(cos2) <- rownames(PCs)
colnames(cos2) <- colnames(PCs)
cos2[, 1:4]
```

```
##                    PC1         PC2         PC3          PC4
## Sebrle     0.669035920 0.113893074 0.002352728 0.1531298319
## Clay       0.684872300 0.038449930 0.005390107 0.0714721363
## Karpov     0.807526493 0.007903026 0.005713353 0.1330069743
## Macey      0.365296925 0.180841922 0.154408343 0.1206280864
## Warners    0.469734621 0.375380751 0.102893033 0.0320316454
## Zsivoczky  0.085913115 0.276553647 0.186194481 0.0508408961
## Hernu      0.164623173 0.088759044 0.185160775 0.0102208730
## Nool       0.003530089 0.338953770 0.108712735 0.4348781923
## Bernard    0.368360586 0.002665231 0.001933423 0.3977985156
## Schwarzl   0.002117182 0.521093330 0.128704546 0.0595509157
## Pogorelov  0.011783936 0.065949318 0.550690190 0.0093348229
## Schoenbeck 0.005350540 0.030091230 0.111554784 0.3652052586
## Barras     0.018536581 0.032997425 0.618740779 0.0844227194
## Smith      0.021825530 0.119150811 0.231696770 0.1163199340
## Averyanov  0.008011492 0.492414129 0.043210615 0.0139933567
## Ojaniemi   0.002813447 0.127249309 0.007280783 0.0023030073
```

```
## Smirnov     0.105413948 0.293308394 0.399620921 0.0266766253
## Qi          0.159601801 0.010355946 0.305407841 0.0356466594
## Drews       0.014763754 0.809996269 0.063778145 0.0254599841
## Parkhomenko 0.158131897 0.306454161 0.047410457 0.2551829622
## Terek       0.071263379 0.005992732 0.491644168 0.0054283815
## Gomez       0.066720146 0.186135544 0.365400119 0.0046181078
## Turi        0.339699139 0.003872514 0.069340801 0.0141660203
## Lorenzo     0.503874434 0.178518505 0.197393373 0.0005413353
## Karlivans   0.577048807 0.003757487 0.001966500 0.1473250138
## Korkizoglou 0.126805942 0.191299889 0.518888596 0.0467621421
## Uldal       0.857528056 0.001746657 0.027998158 0.0001724879
## Casarsa     0.450004772 0.504141868 0.006049613 0.0050147517
```

```r
best <- which.max(cos2[, 1] + cos2[, 2])
worst <- which.min(cos2[, 1] + cos2[, 2])

names(best)
```

```
## [1] "Casarsa"
```

```r
cos2[best, 1:2]
```

```
##       PC1       PC2
## 0.4500048 0.5041419
```

```r
names(worst)
```

```
## [1] "Schoenbeck"
```

```r
cos2[worst, 1:2]
```

```
##        PC1        PC2
## 0.00535054 0.03009123
```

```r
ctr <- matrix(0, nrow = nrow(dat_act), ncol = ncol(dat_act))

for (i in 1:nrow(ctr)) {
  for (j in 1:ncol(ctr)) {
    ctr[i, j] <- 100 / (nrow(ctr) - 1) * (PCs[i, j])^2 / eigenvalues[j]
  }
}

rownames(ctr) <- rownames(PCs)
colnames(ctr) <- colnames(PCs)
ctr[, 1:4]
```

```
##                      PC1         PC2         PC3          PC4
## Sebrle      13.896472718  4.25667207  0.12183879 12.478583027
## Clay        13.566366232  1.37046272  0.26620124  5.554449503
## Karpov      18.437668318  0.32468361  0.32523627 11.914448816
## Macey        3.791512875  3.37740675  3.99572873  4.912078298
## Warners      3.765848145  5.41501879  2.05662407  1.007487819
## Zsivoczky    0.505123020  2.92573389  2.72937503  1.172738593
## Hernu        0.497794814  0.48293619  1.39594113  0.121254458
## Nool         0.033024268  5.70565731  2.53563429 15.961196069
## Bernard      2.587013828  0.03368047  0.03385408 10.960719825
## Schwarzl     0.008939045  3.95882420  1.35483238  0.986442407
## Pogorelov    0.068370188  0.68850078  7.96603923  0.212487210
```

```
## Schoenbeck   0.015334884  0.15518173  0.79713121  4.106485050
## Barras       0.085522257  0.27393487  7.11732807  1.528126098
## Smith        0.235265509  2.31104393  6.22690381  4.919239127
## Averyanov    0.049790581  5.50658088  0.66954990  0.341197634
## Ojaniemi     0.014047826  1.14325620  0.09063735  0.045114489
## Smirnov      0.411458048  2.06001181  3.88896838  0.408515646
## Qi           0.550830837  0.06431140  2.62796365  0.482669233
## Drews        0.180438327 17.81282299  1.94340179  1.220788555
## Parkhomenko  1.797609753  6.26843595  1.34372003 11.380934728
## Terek        0.831378555  0.12579836 14.30019672  0.248458059
## Gomez        0.433187509  2.17453292  5.91488543  0.117634126
## Turi         3.397770397  0.06969640  1.72920767  0.555901396
## Lorenzo      6.944171406  4.42689362  6.78249358  0.029269465
## Karlivans    5.608020249  0.06570709  0.04764855  5.617251120
## Korkizoglou  2.220130040  6.02658464 22.65017943  3.212059105
## Uldal        8.685084058  0.03183105  0.70699096  0.006853852
## Casarsa     11.381826314 22.94379938  0.38148824  0.497616293
```

```r
best <- which.max(ctr[, 1] + ctr[, 2])
worst <- which.min(ctr[, 1] + ctr[, 2])

names(best)
```

```
## [1] "Casarsa"
```

```r
ctr[best, 1:2]
```

```
##      PC1      PC2
## 11.38183 22.94380
```

```r
names(worst)
```

```
## [1] "Schoenbeck"
```

```r
ctr[worst, 1:2]
```

```
##        PC1        PC2
## 0.01533488 0.15518173
```

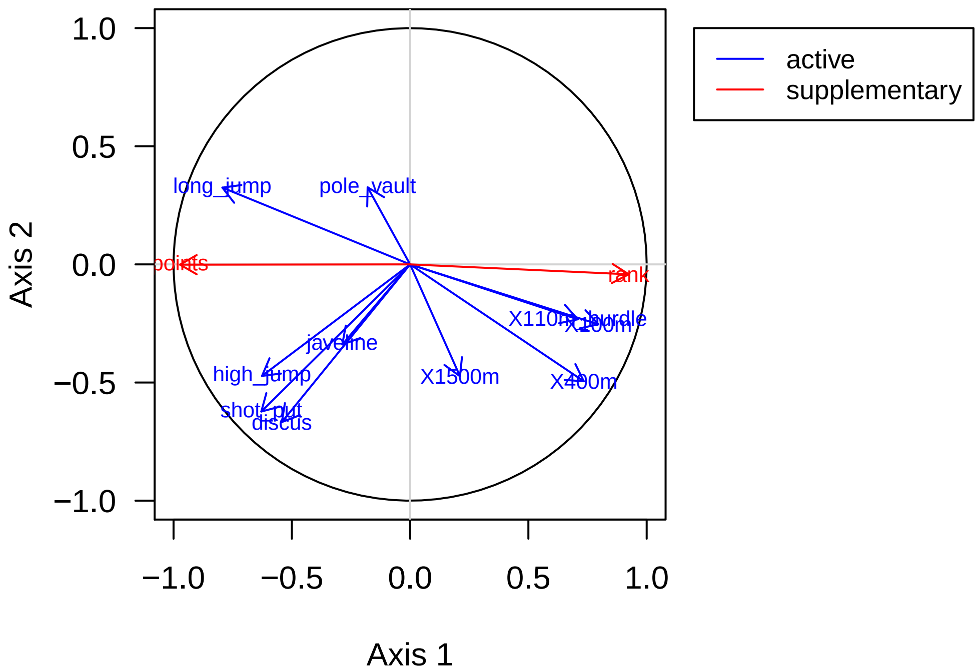
4) Studying the cloud of variables

```r
# calculate the correlation of all quantitative variables with PCs
X <- dat[1:28, c(-1, -14)]
PC_cor <- cor(X, PCs)
PC_cor[, 1:4]
```

```
##                    PC1          PC2         PC3          PC4
## X100m        0.7958383 -0.253599340  0.25277014 -0.071298025
## long_jump   -0.7935059  0.324979385 -0.15521388  0.005840942
## shot_put    -0.6289690 -0.622800538 -0.02252512 -0.133493731
## high_jump   -0.6259915 -0.471948288  0.01291630  0.105515561
## X400m        0.7341798 -0.494656647 -0.22972590 -0.111158299
## X110m_hurdle  0.7089265 -0.232408269  0.04392919 -0.109666171
## discus      -0.5421042 -0.667282351 -0.01785549  0.196002694
## pole_vault  -0.1795989  0.326306097 -0.62447715 -0.611344812
## javeline    -0.2864207 -0.338982261  0.52108731 -0.655675756
```

```
## X1500m        0.2107444 -0.473357014 -0.78520075  0.054465625
## rank          0.9243932 -0.041903953 -0.07680790  0.148552939
## points       -0.9724931 -0.001294792  0.06188580 -0.196710089
```

```r
circle <- function(center = c(0, 0), npoints = 100) {
r = 1
tt = seq(0, 2 * pi, length = npoints)
xx = center[1] + r * cos(tt)
yy = center[2] + r * sin(tt)
data.frame(x = xx, y = yy)
}
corcir <- circle(c(0, 0), npoints = 100)

# circle of correlations plot
par(pty="s")
plot(PC_cor[, 1], PC_cor[, 2], xlim = c(-1, 1), ylim = c(-1, 1),
     xlab = "Axis 1", ylab = "Axis 2", type = "n", las = 1)
lines(corcir)
abline(h = 0, v = 0, col = "lightgray")
text(PC_cor[1:10, 1], PC_cor[1:10, 2], rownames(PC_cor)[1:10],
     col = "blue", cex = 0.7)
text(PC_cor[11:12, 1], PC_cor[11:12, 2], rownames(PC_cor)[11:12],
     col = "red", cex = 0.7)
arrows(x0 = 0, y0 = 0, x1 = PC_cor[1:10, 1], y1 = PC_cor[1:10, 2],
       length = 0.1, col = "blue")
arrows(x0 = 0, y0 = 0, x1 = PC_cor[11:12, 1], y1 = PC_cor[11:12, 2],
       length = 0.1, col = "red")
par(xpd = TRUE)
legend(1.2, 1, legend = c("active", "supplementary"),
       col = c("blue", "red"), cex = 0.8, lwd = 1)
```



This plot shows that almost all of information about supplementary variables is represented by the first axis. Some of active variables seem to be correlated to each other. For example, “high_jump”, “shot_put”, "discus", and "javeline" have negative values in both axes. Only two variables, "long_jump" and "ple_vault", have positive values in the second axis.

5) Conclusions

By Principal Component Analysis, we are able to interpret relationship among variables and resemblance among individuals. We first figure out how many independent variables the data has and decompose the data to find loadings, which is a matrix of eigenvectors and useful to find Principal Components(PCs). We want to maximize PCs so we can preserve information as much as possible. Eigenvalues of correlation data matrix represents the variance of PCs. Proportion of eigenvalues indicates how much information of the original data it contains. We choose first three PCs to interpret the data, which also means we can handle 70% of the original data with only three PCs. For easier interpreting purpose, we make plots in terms of PCs, a correlation matrix between PCs and variables, and plot the correlation matrix in two dimensional space.