ST 437/537: Applied Multivariate and Longitudinal Data Analysis

Principal Components Analysis

Arnab Maity
NCSU Department of Statistics
SAS Hall 5240 919-515-1937 amaity[at]ncsu.edu

Introduction

Often multivariate data sets contain too many variables, and this might lead to the curse of dimensionality (Bellman, 1961): using standard graphing techniques, as well as usual analysis methods become problematic. Thus arises the need to reduce the dimensionality and to identify/summarize the crucial variables.

Principal components analysis (PCA) is a dimension reduction technique that is widely used in multivariate statistics. The objective is to condense the information that is present in the original set of variables via linear combinations of the variables while losing as little information as possible.

Linear combination

Given a vector $\boldsymbol{X} = (X_1, \ldots, X_p)^T$, a linear combination of $\boldsymbol{X}$ is defined as $a_1X_1 + \ldots + a_pX_p$. If we define the vector $\boldsymbol{a} = (a_1, \ldots, a_p)^T$, then the linear combination can be written as $\boldsymbol{a}^T\boldsymbol{X}$.

Typically, the number of linear transformations is much smaller than the number of original variables; hence the reduction in the dimensionality of the data. This can be useful in different ways, such as providing better visualization and computational advantages. PCA also *decorrelates* the data, that is, PCA produces linear combinations of the variables that are mutually uncorrelated.

Suppose we have a $p \times 1$ random vector $\boldsymbol{X} = (X_1, \ldots X_p)^T$. The main goal of PCA is to identify *linear combinations* of $\boldsymbol{X}$ of the form

$$Y_i = \boldsymbol{a}_i^T\boldsymbol{X}, \quad i = 1, 2, \ldots, q,$$

that explain most of the variability in $\boldsymbol{X}$.

Total variation

The total variation (TV) of $\boldsymbol{X}$ is defined as the sum of the individual variances,

$$TV = \text{var}(X_1) + \ldots + \text{var}(X_p).$$

In other words, if $\boldsymbol{\Sigma} = cov(\boldsymbol{X})$, then $TV = trace(\Sigma)$.

Typically $q < p$, and the new variables, Y_i, are ordered according to their importance. Specifically, Y_1 is designed to capture the most variability in the original variables (i.e., TV) by any linear combination; Y_2 then captures the most of the remaining variability while being uncorrelated to Y_1, and so on. In the end, we hope that the first few Y_i's will capture most of the variability in $\boldsymbol{X}$.

Loadings

The individual components (the elements of the vector a_i) of each PC are called loadings. The loadings tell us how the original variables are weighted to get the PCs.

A quick example

Consider the **[weekly stock return data] (https://www.stat.ncsu.edu/people/maity/courses/st537-S2019/data/T8-4.DAT)**, where 103 weekly rates of return on five stocks (JPMorgan, Citibank, WellsFargo, Shell, Exxon) are recorded. We define

$$\text{weekly return} = \frac{\text{current closing price} - \text{previous week closing price}}{\text{previous week closing price}}$$

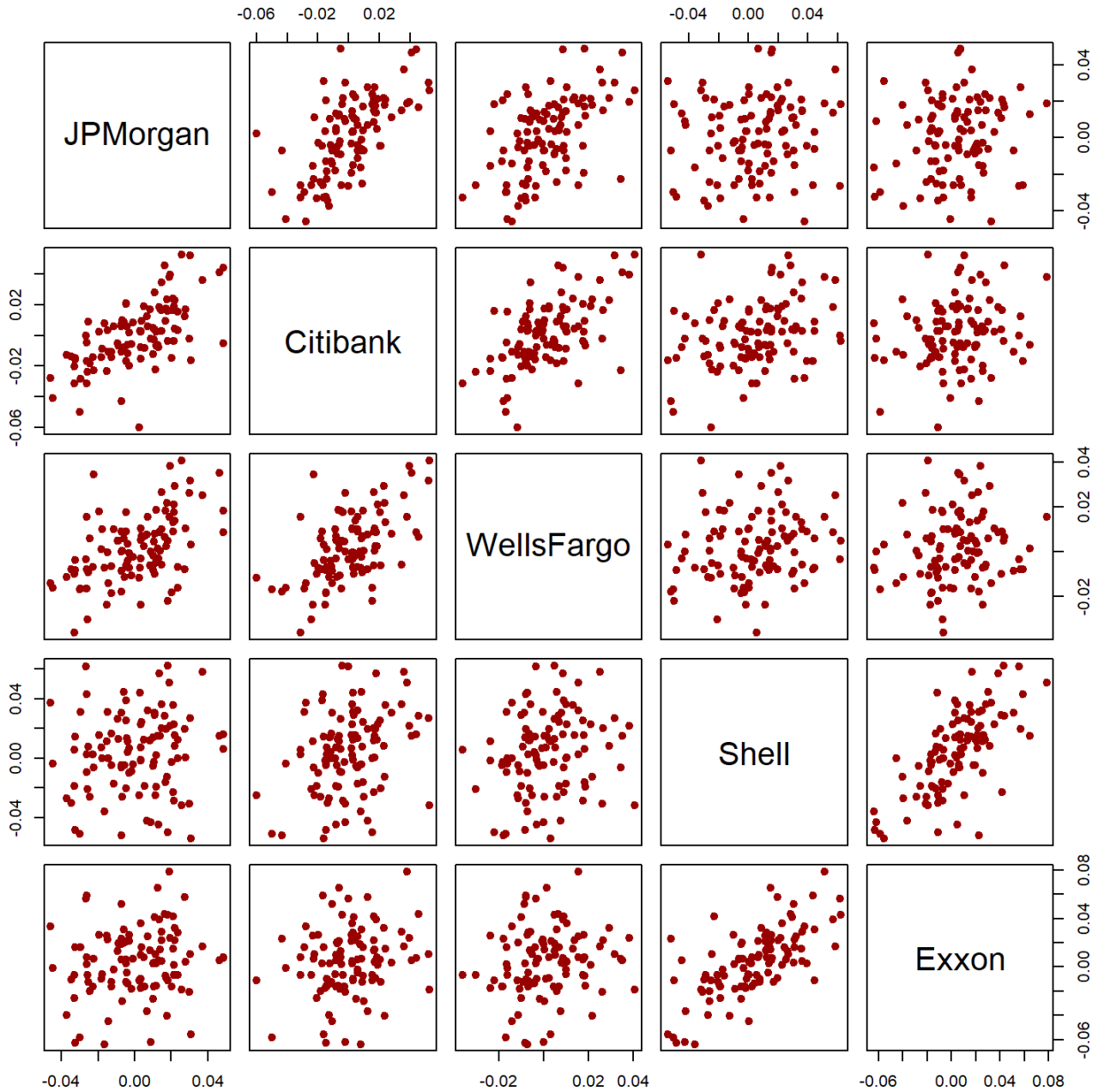
adjusted for stock splits and dividend. Rates of returns across stocks are expected to be correlated. A snapshot of the data is shown below.

```
dat <- read.table("data/T8-4.DAT", header = F)
colnames(dat) <- c("JPMorgan", "Citibank", "WellsFargo", "Shell", "Exxon")
head(dat)
```

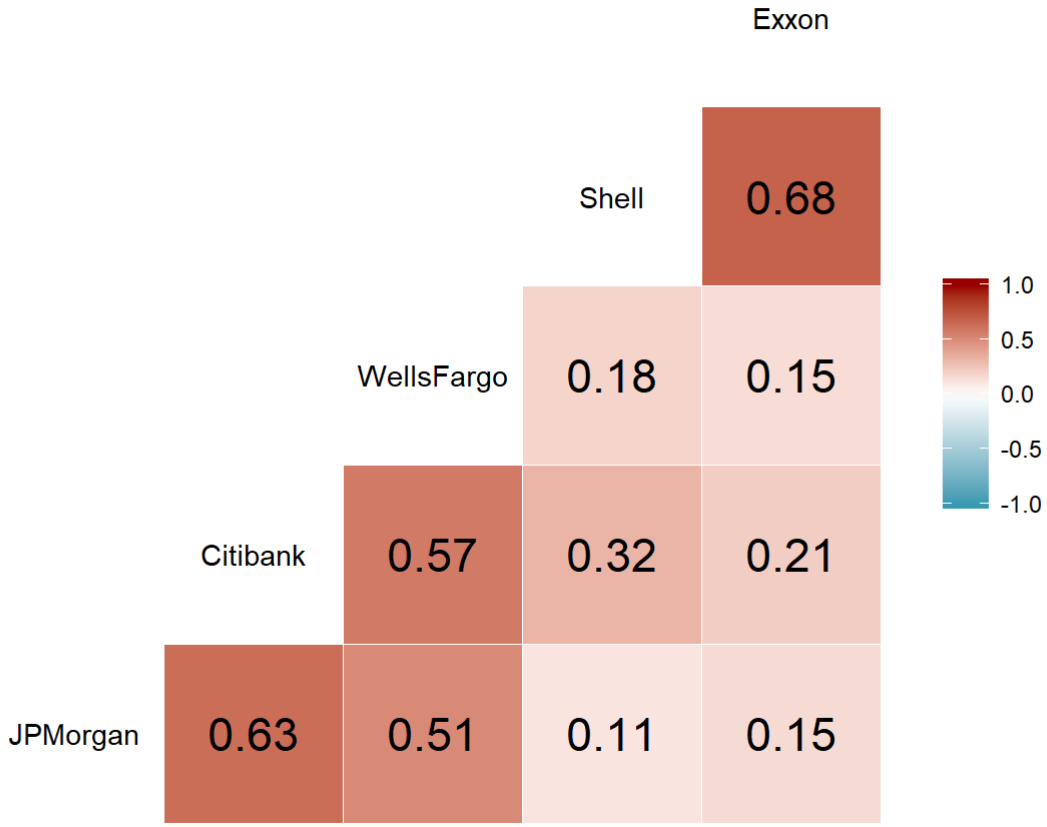
```
##     JPMorgan   Citibank WellsFargo      Shell      Exxon
## 1  0.0130338 -0.0078431 -0.0031889 -0.0447693  0.0052151
## 2  0.0084862  0.0166886 -0.0062100  0.0119560  0.0134890
## 3 -0.0179153 -0.0086393  0.0100360  0.0000000 -0.0061428
## 4  0.0215589 -0.0034858  0.0174353 -0.0285917 -0.0069534
## 5  0.0108225  0.0037167 -0.0101345  0.0291900  0.0409751
## 6  0.0101713 -0.0121978 -0.0083768  0.0137083  0.0029895
```

A pairs-plot of the data and the corraltion matrix are shown below.

```
pairs(dat, pch=19, col="#990000")
```




```
library(GGally)
ggcorr(dat,
       low = "#3B9AB2", mid = "#FFFFFF", high = "#990000",
       label = T, label_color = "black",
       label_size = 6, label_round = 2)
```



Before performing PCA, we should always check whether each variable in the dataset has similar standard deviations (or variances). If not, we need to standardize them.

```
apply(dat, 2, sd)
```

```
##   JPMorgan   Citibank WellsFargo      Shell      Exxon
## 0.02081513 0.02094558 0.01496570 0.02687929 0.02767082
```

It is evident that the variables have very different standard deviations, e.g., sd(Exxon) is almost twice of sd(WellsFargo). Thus we need to standardize each variable.

```
std.dat <- scale(dat, center = T, scale = T)
apply(std.dat, 2, sd)
```

```
##   JPMorgan   Citibank WellsFargo      Shell      Exxon
##          1          1          1          1          1
```

Scaling the variables

When variables are on very different scales or have very different variances, a principal components analysis should be performed on the standardized

variables.

PCA can be performed using the `prcomp()` function in base R.

```
# Perform PCA
data.pca <- prcomp(std.dat)

# Extract the importance of each component
summary(data.pca)
```

```
## Importance of components:
##                           PC1    PC2    PC3     PC4     PC5
## Standard deviation     1.5612 1.1862 0.7075 0.63248 0.50514
## Proportion of Variance 0.4874 0.2814 0.1001 0.08001 0.05103
## Cumulative Proportion  0.4874 0.7689 0.8690 0.94897 1.00000
```

In the output above, the row marked “Standard deviation” gives the standard deviation of each PC, that is, sd(Y_i). Thus, variance of the first PC is $var(Y_1) = 2.4373454$. While each variable has variance 1 (since each variable has been standardized), the fist PC alone has variance 2.4373454.

The second row, marked “Proportion of Variance,” shows the proportion of TV captured by each PC, that is, $var(Y_i)/TV$. Notice that, since we standardized each variable, the variance of each standardized variable is 1, giving us, $TV = 5$. Thus, for the 1st PC, the proportion of variance captured is $var(Y_1)/TV = (1.5612)^2/5 = 0.4874691$.

The third row, marked by “Cumulative Proportion,” explains the proportion of total variation explained cumulatively by first few PCs. For example, the first two PCs explain almost 77% of TV. Such a criterion enables us to choose how many PCs to keep. For example, if we are satisfied with capturing at least 75% of the total variation, we only need to keep two PCs.

The loadings for the first two PCs are shown below.

```
# The loadings of the first two PCs
round( data.pca$rotation[, 1:2], 3 )
```

```
##                PC1    PC2
## JPMorgan    -0.469  0.368
## Citibank    -0.532  0.236
## WellsFargo -0.465  0.315
## Shell       -0.387 -0.585
## Exxon       -0.361 -0.606
```

We can interpret the first PC as a roughly equally weighted sum. This might be a *general market component*. The second PC represents a contrast between the banking stocks and the oil stocks. This might be called an *industry component*.

Computational details

Suppose $\boldsymbol{X}$ is a random vector with $cov(\boldsymbol{X}) = \boldsymbol{\Sigma}$. For now, we assume $\boldsymbol{\Sigma}$ is known. Later we will replace $\boldsymbol{\Sigma}$ by $\boldsymbol{S}$ or $\boldsymbol{R}$ computed from the data. Recall that the "variability" in $\boldsymbol{X}$ is represented by the total variation, $TV = trace(\boldsymbol{\Sigma})$.

Consider the linear combination, the **first principal component**,

$$Y_1 = a_{11}X_1 + \ldots + a_{1p}X_p = \boldsymbol{a}_1^T\boldsymbol{X}$$

for some $\boldsymbol{a}_1 = (a_{11}, \ldots, a_{1p})^T$ that we need to determine. The constants are called *loadings*. We determine the loadings by solving the following problem:

$$\text{maximize } var(Y_1) \text{ with the constraint } \boldsymbol{a}_1^T\boldsymbol{a}_1 = 1.$$

The solution can be obtained by using the Lagrange multiplier method. Specifically, one can show that the optimal choice of $\boldsymbol{a}_1$ is a eigenvector of $\boldsymbol{\Sigma}$ corresponding to the largest eigenvalue, and that $var(Y_1) = \lambda_1$, the largest eigenvalue.

Interpretation of the 1st PC

Here $\boldsymbol{a}_1^T\boldsymbol{X}$ is interpreted as the projection of $\boldsymbol{X}$ onto the direction $\boldsymbol{a}_1$. Thus the 1st PC is the direction such that the projection of the data onto this direction has the largest possible variance; the 1st PC captures most of the total variation.

The loadings $\boldsymbol{a}_2 = (a_{21}, \ldots, a_{2p})^T$ of the **second principal component**

$$Y_2 = a_{21}X_1 + \ldots + a_{2p}X_p = \boldsymbol{a}_2^T\boldsymbol{X}$$

is chosen by solving the following problem:

$$\text{maximize } var(Y_2) \text{ with the constraint } \boldsymbol{a}_2^T\boldsymbol{a}_2 = 1 \text{ and } \boldsymbol{a}_1^T\boldsymbol{a}_2 = 0.$$

Again, using Lagrange multipliers, one can show that the optimal choice of $\boldsymbol{a}_2$ is a eigenvector of $\boldsymbol{\Sigma}$ corresponding to the second largest eigenvalue (orthogonal to the first direction), and that $var(Y_2) = \lambda_2$, the second largest eigenvalue.

Interpretation of the 2nd PC

The 2nd PC is the direction such that it is orthogonal to the 1st PC and the projection of the data onto this direction has the largest possible variance.

Since the lodiang vector of the 2nd PC is orthogonal to that of the 1st PC, it readily follows that the 1st and 2nd PCs are uncorrelated.

We continue this process until we get all the p PCs. It can be shown that

$$TV = var(Y_1) + \ldots + var(Y_p).$$

Specifically, the total variation in $\boldsymbol{X}$ is fully captured by retaining all p PCs.

Optional: Spectral decomposition of $\boldsymbol{\Sigma}$

Since $\boldsymbol{\Sigma}$ is positive definite matrix it follows that it has the *spectral decomposition*

$$\boldsymbol{\Sigma} = \lambda_1\boldsymbol{a}_1\boldsymbol{a}_1^T + \ldots + \lambda_p\boldsymbol{a}_p\boldsymbol{a}_p^T;$$

where $\boldsymbol{a}_j$s as *eigenvectors* and λ_j are the corresponding *eigenvalues*. The vectors $\boldsymbol{a}_j$s can be chosen so that they are orthonormal, that is, $\boldsymbol{a}_i^T\boldsymbol{a}_i = 1$ and $\boldsymbol{a}_i^T\boldsymbol{a}_j = 1$ for $j \neq i$.

The vector $\boldsymbol{a}_i$ is the vector of loadings for the i-th PC Y_i. Also, $var(Y_i) = \lambda_i$, the corresponding eigenvalue.

The proportion of the variance that is explained by the jth PC is

$$\frac{\lambda_j}{\lambda_1 + \ldots + \lambda_p}.$$

The proportion of the variance explained by the first j PCs together is

$$\frac{\lambda_1 + \ldots + \lambda_j}{\lambda_1 + \ldots + \lambda_p}.$$

Sample PCA

In practice, the true covariance matrix $\boldsymbol{\Sigma}$ is unknown, and we only have a random sample $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n$. Thus we can estimate $\boldsymbol{\Sigma}$ by the sample covariance matrix

$$\boldsymbol{S} = \frac{1}{n}\sum_{i=1}^{n}(\boldsymbol{X}_i - \bar{\boldsymbol{X}})(\boldsymbol{X}_i - \bar{\boldsymbol{X}})^T.$$

When the variables are standardized, then we essentially use the sample correlation matrix $\boldsymbol{R}$.

In our stock price example, we suggested to standardize the variables. Thus, using sample covariance and correlation matrices are equivalent.

```
# Recall, std.dat contains standadrdized variables
S <- cov(std.dat)

# The function eigen() can compute the eigenvectors/eigenvalues
eig.out <- eigen(S)
str(eig.out)
```

```
## List of 2
##  $ values : num [1:5] 2.437 1.407 0.501 0.4 0.255
##  $ vectors: num [1:5, 1:5] -0.469 -0.532 -0.465 -0.387 -0.361 ...
##  - attr(*, "class")= chr "eigen"
```

The field “values” gives the eigenvalues (ordered from largest to smallest). These numbers represent the variance of each PC. Let us also compute proportion of variance explained by each PC and the cumulative proportion of variance explained for each PC.

```
# eigenvalues
lam <- eig.out$values

tab <- rbind(lam,
          lam/sum(lam), # proportion of variance explained
          cumsum(lam)/sum(lam)) # cumulative proportion of var explained
rownames(tab) <- c("Variance", "Proportion of variance", "Cumulative proportion")
```

Now we compare output from `prcomp()` with results from directly using `eigen()`.

Eigenvalues:

```
# Results using eigen()
round(tab, 4)
```

```
##                            [,1]   [,2]   [,3]  [,4]   [,5]
## Variance                 2.4373 1.4070 0.5005 0.400 0.2552
## Proportion of variance 0.4875 0.2814 0.1001 0.080 0.0510
## Cumulative proportion  0.4875 0.7689 0.8690 0.949 1.0000
```

Output from `prcomp()`:

```
# Results from prcomp()
summary(data.pca)
```

```
## Importance of components:
##                             PC1    PC2    PC3     PC4     PC5
## Standard deviation     1.5612 1.1862 0.7075 0.63248 0.50514
## Proportion of Variance 0.4874 0.2814 0.1001 0.08001 0.05103
## Cumulative Proportion  0.4874 0.7689 0.8690 0.94897 1.00000
```

Note that the eigenvalues shown in the left panel (in the row “Variance”) should be squares of the values in the row “Standard deviation” in the output of `prcomp()`.

Now we compare the loading vectors obtained directly using `eigen()` to those given produced by `prcomp()`.

Eigenvectors:

```
# Eigenvectors
round(eig.out$vectors, 4)
```

```
##         [,1]    [,2]    [,3]    [,4]    [,5]
## [1,] -0.4691  0.3680  0.6043  0.3630  0.3841
## [2,] -0.5324  0.2365  0.1361 -0.6292 -0.4962
## [3,] -0.4652  0.3152 -0.7718  0.2890  0.0712
## [4,] -0.3873 -0.5850 -0.0934 -0.3813  0.5947
## [5,] -0.3607 -0.6058  0.1088  0.4934 -0.4976
```

Output from `prcomp()`:

```
round(data.pca$rotation, 4)
```

```
##                PC1     PC2     PC3     PC4     PC5
## JPMorgan   -0.4691  0.3680 -0.6043  0.3630  0.3841
## Citibank   -0.5324  0.2365 -0.1361 -0.6292 -0.4962
## WellsFargo -0.4652  0.3152  0.7718  0.2890  0.0712
## Shell      -0.3873 -0.5850  0.0934 -0.3813  0.5947
## Exxon      -0.3607 -0.6058 -0.1088  0.4934 -0.4976
```

It is evident that the eigenvectors are exactly the PC loadings that `prcomp()` produces.

Number of PCs to retain

A common approach to determining the number of components to retain is to keep the first few components that explain a pre-specified large percentage of the total variation of the original variables. We typically use Values between 70% and 95%.

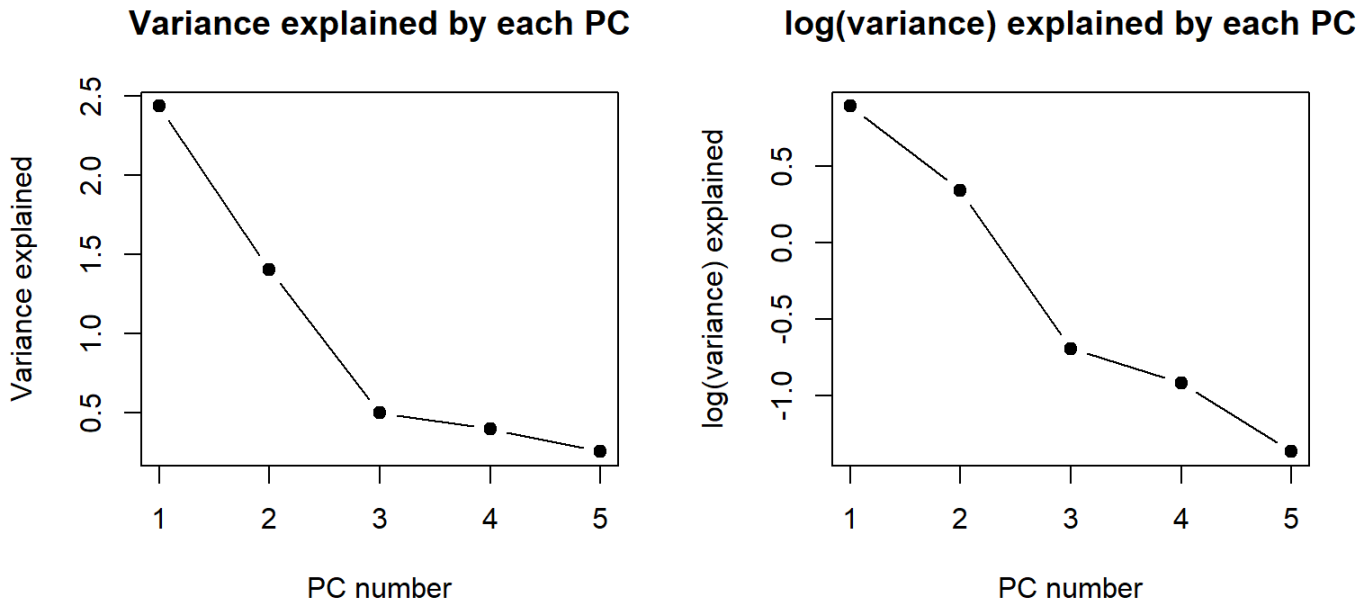
Other possible rules have been suggested by various authors. For example, we could plot the *ordered* eigenvalues, λ_i (variance captured by the PCs) of the i-th component versus i, as proposed by **[Cattell, R. B. (1966). The scree test for the number of factors. Multivariate Behavioural Research, 1, 245 – 276] (https://www.tandfonline.com/doi/abs/10.1207/s15327906mbr0102_10)**. This plot is called the **scree plot**. Later [Farmer, S. A. (1971), An investigation into the results of principal components analysis of data derived from random numbers," Statistician, 20, 63 – 72] suggested plotting $log(\lambda_i)$ instead of λ_i.

In our previous stock price example, scree plots are shown below.

```
# Eigenvalues (variance of PCs)
lambda <- data.pca$sdev^2

# scree plots
par(mfrow = c(1,2))
plot(lambda, type="b", pch = 19, main = "Variance explained by each PC",
     xlab = "PC number", ylab = "Variance explained")

plot(log(lambda), type="b", pch = 19, main = "log(variance) explained by each PC",
     xlab = "PC number", ylab = "log(variance) explained")
```



We look for a bend in the plot where the curve changes from being steep to being almost a straight line. In the plot shown in the left panel above, we see the plot becomes less steep after the 3rd PC. Thus we might wish to retain three PCs. From the PCA output shown before, the first three PCs cumulatively explain almost 87% variability.

Another rule of thumb is to choose those PCs whose variance is less than the average variance TV/p. When the variables are standardized, that is, each of them has variance 1, we have $TV = p$, and average variance is $p/p = 1$. Thus, when the variables are standardized, the PCs with $\lambda_i < 1$ will be rejected. [Jolliffe, I. (1972), Discarding variables in a principal component analysis. I: Artificial data, Journal of the Royal Statistical Society, Series C, 21, 160 – 173] proposed a modified rule to reject PCs with $\lambda_i < 0.7$.

It should be noted that the choice of PCs should not be based on just the percent of variation explained. **One should also look at their subject matter interpretation**. If we obtain a PC which we can not interpret, the usability of such a component may become limited.

Principal components scores

The principal components scores are calculated for each PC for each subject in the dataset. Specifically, suppose we retail k PCs. For the i-th subject with observed data vector $\boldsymbol{x}_i$ (original data or standardized data), the principal components scores for the first k PCs are defined as

$$s_{i1} = \boldsymbol{a}_1^T \boldsymbol{x}_i, \quad \ldots, \quad s_{ik} = \boldsymbol{a}_k^T \boldsymbol{x}_i.$$

Often the variables are centered before computing the scores, so that the scores have mean zero. Specifically,

$$s_{i1} = \boldsymbol{a}_1^T(\boldsymbol{x}_i - \bar{\boldsymbol{x}}), \quad \ldots, \quad s_{ik} = \boldsymbol{a}_k^T(\boldsymbol{x}_i - \bar{\boldsymbol{x}}).$$

This centering does not change the variance of the PCs.

Plots of PC scores can reveal suspect observations and possible outliers. Let us consider the dataset **[Table 4.3 in Johnson and Wichern (2007). Applied Multivariate Analysis.] (data/T4-3.DAT)** where four measures of stiffness $x_1, \ldots, x_4$ are measured of each of the $n = 30$ boards. We used this dataset in the lecture of assessment in multivariate normality.

```
# Reading the data set
dat <- read.table("data/T4-3.DAT", header = F)
colnames(dat) <- c("x1", "x2", "x3", "x4", "d2")

n <- nrow(dat)
p <- ncol(dat) - 1

# snapshot
head(dat)
```

```
##     x1   x2   x3   x4   d2
## 1 1889 1651 1561 1778 0.60
## 2 2403 2048 2087 2197 5.48
## 3 2119 1700 1815 2222 7.62
## 4 1645 1627 1110 1533 5.21
## 5 1976 1916 1614 1883 1.40
## 6 1712 1712 1439 1546 2.22
```

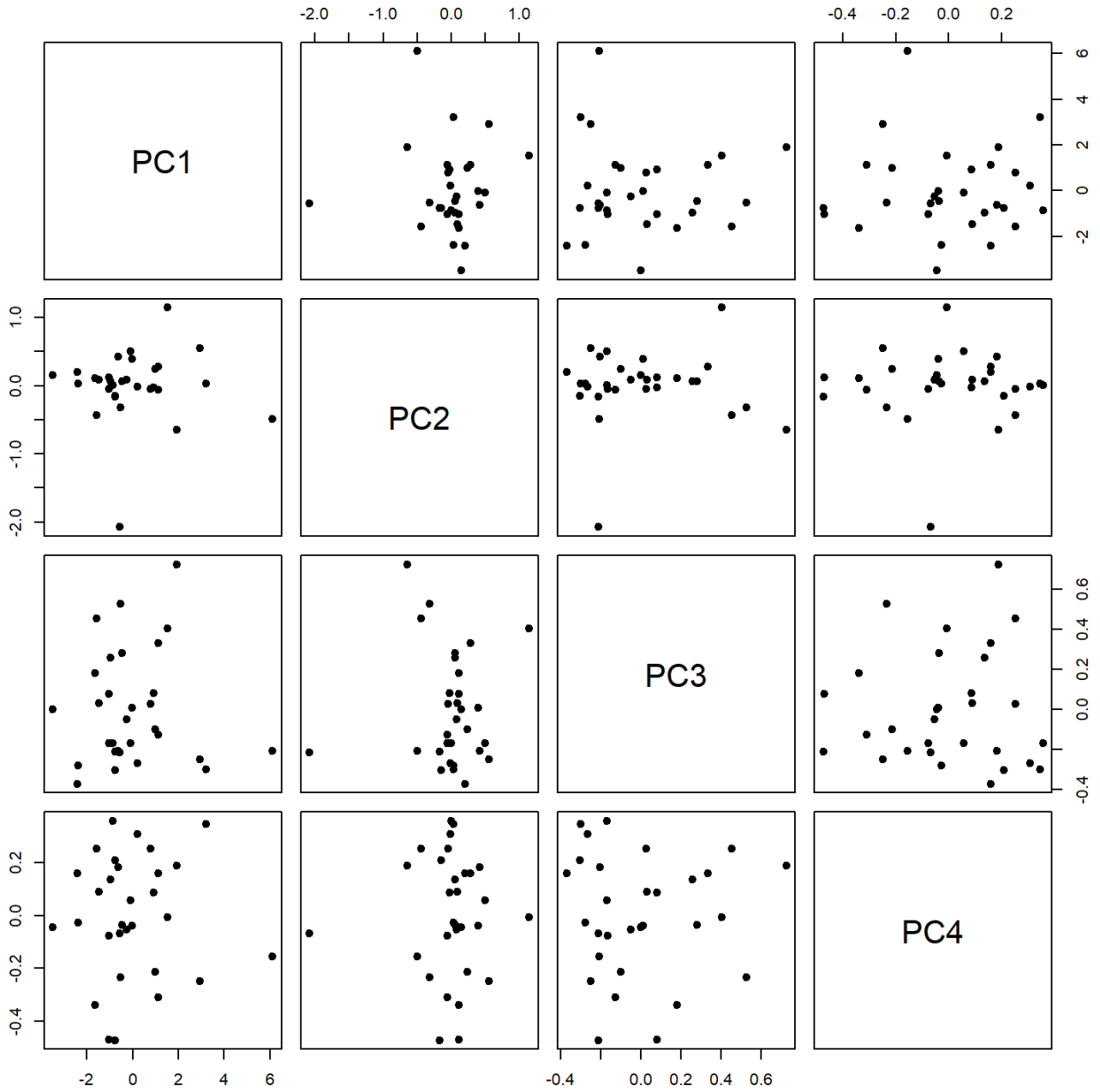
Let us perform PCA of this dataset and compute the PC scores.

```
std.data <- scale(dat[, 1:4], scale = T)
data.pca <- prcomp(std.data)
summary(data.pca)
```

```
## Importance of components:
##                           PC1     PC2     PC3     PC4
## Standard deviation     1.897 0.51735 0.28119 0.23047
## Proportion of Variance 0.900 0.06691 0.01977 0.01328
## Cumulative Proportion  0.900 0.96695 0.98672 1.00000
```

The first two PCs explain almost 97% variability. Let us construct a pairs-plot of the PC scores.

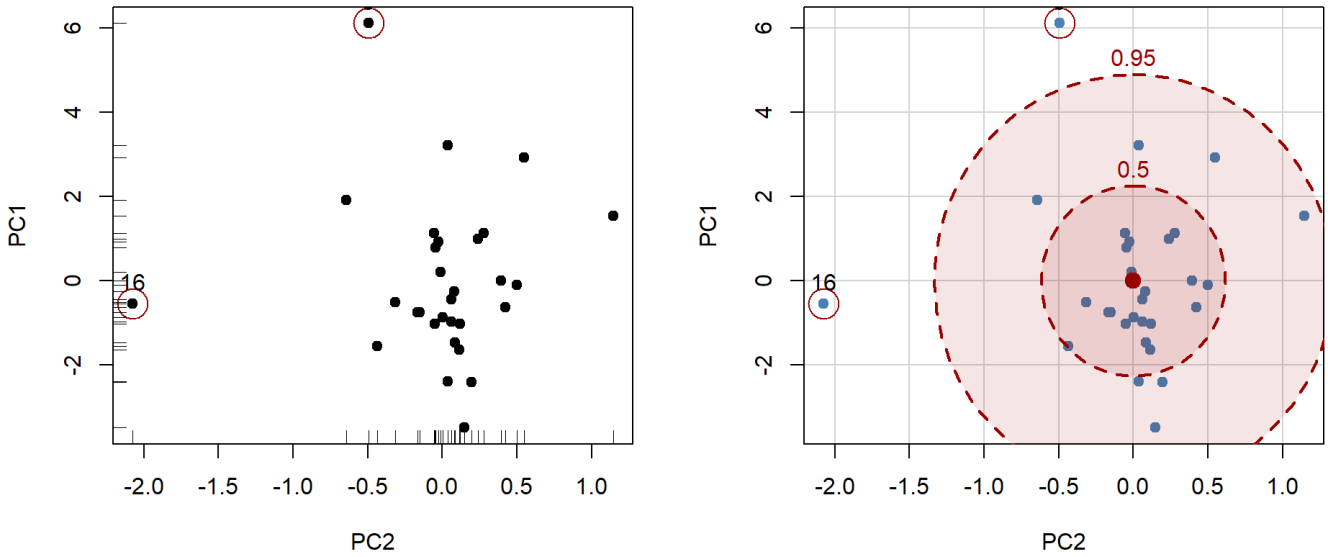
```
# pairs plot of pc scores
pairs(data.pca$x, pch=19)
```



It seems there might be two outliers. Let use see a scatterplot of PC1 versus PC2. Recall we flagged observations 9 and 16 as potential outliers.

```
par(mfrow= c(1,2))
# scatterplot
plot(data.pca$x[, 2:1], pch=19)
points(data.pca$x[c(9, 16), 2:1], cex = 3, col="#990000")
text(data.pca$x[c(9, 16), 2:1], labels = c(9, 16), pos = 3)
rug(data.pca$x[,2], side = 1)
rug(data.pca$x[,1], side = 2)

# Data ellipse
library(car)
dataEllipse(data.pca$x[, 2:1],
            pch=19, col = c("steelblue", "#990000"), lty=2,
            ellipse.label=c(0.5, 0.95), levels = c(0.5, 0.95),
            fill=TRUE, fill.alpha=0.1)
points(data.pca$x[c(9, 16), 2:1], cex = 3, col="#990000")
text(data.pca$x[c(9, 16), 2:1], labels = c(9, 16), pos = 3)
```



The two outliers are clearly separated in the scatterplot above. Looking at the PC loadings we see that the first PC is essentially a average of the four variables, while the second PC represents the difference between X_2 and (X_3, X_4).

```
data.pca$rotation[, 1:2]
```

```
##           PC1        PC2
## x1 0.5137718 -0.2060665
## x2 0.4841620 -0.7316902
## x3 0.4999301  0.4657684
## x4 0.5016927  0.4530186
```

Overall, Johnson and Wichern (2007) suggest to make scatterplots of the fisrt few PC scores and also of the *last few* PCs. These plots help identify suspect observations.

Other considerations

Linear dependence among variables

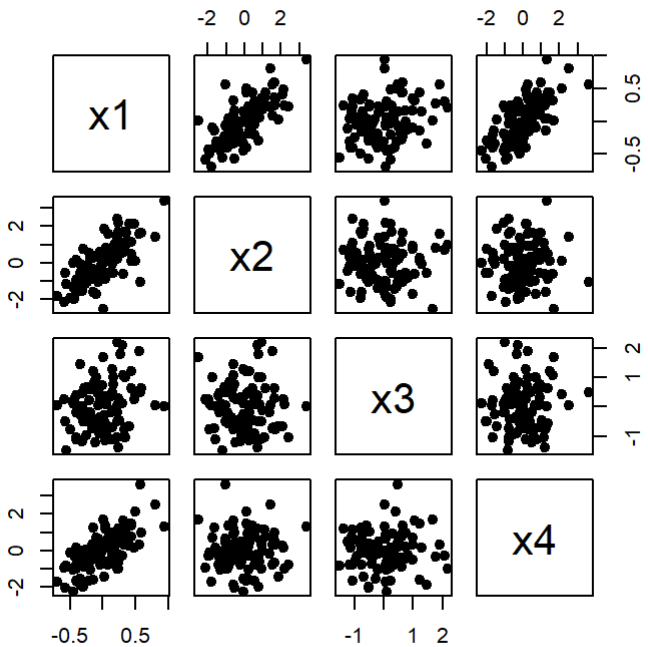
Even though we only retain the first few PCs with highest eigenvalues, we should not completely ignore the components with small eigenvalues. A near zero eigenvalue might indicate a linear relationship in the data. In such a case, one or more of the variables are redundant and should be deleted.

Consider the following example, where we have four variables $X_1, \ldots, X_4$, such that $X_1 = (0.2)X_2 + (0.1)X_3 + (0.2)X_4$.

```
set.seed(1001)
n <- 100
x2 <- rnorm(n)
x3 <- rnorm(n)
x4 <- rnorm(n)
x1 <- 0.2*x2 + 0.1*x3 + 0.2*x4
X <- cbind(x1, x2, x3, x4)
```

A quick look at pairs plot or the correlation matrix does not really reveal the *perfect* relation between X_1 and (X_2, X_3, X_4).

```
pairs(X, pch=19)
```




```
cor(X)
```

```
##           x1          x2          x3         x4
## x1 1.0000000  0.72059184  0.22453311 0.65736531
## x2 0.7205918  1.00000000 -0.05796976 0.01702305
## x3 0.2245331 -0.05796976  1.00000000 0.01894994
## x4 0.6573653  0.01702305  0.01894994 1.00000000
```

A PCA on the data reveals that a possible linear dependency in the data, as the last eigenvalue is essentially zero.

```
# scale. = T tells prcomp to standardize the data
summary( prcomp(X, scale. = T) )
```

```
## Importance of components:
##                           PC1    PC2    PC3       PC4
## Standard deviation     1.4152 1.0257 0.9721 2.001e-16
## Proportion of Variance 0.5007 0.2630 0.2362 0.000e+00
## Cumulative Proportion  0.5007 0.7638 1.0000 1.000e+00
```

Thus, we should not entirely ignore the near-zero eigenvalues as they might point out linear dependencies that might become problematic in subsequent analysis.

PCA is not invariant to scaling

PCA result might change if one changes unit of measurement (e.g., pound to Kg) for variables. Also, if the variables have very different variances, then the top PCs will be dominated by the variables with largest variances.

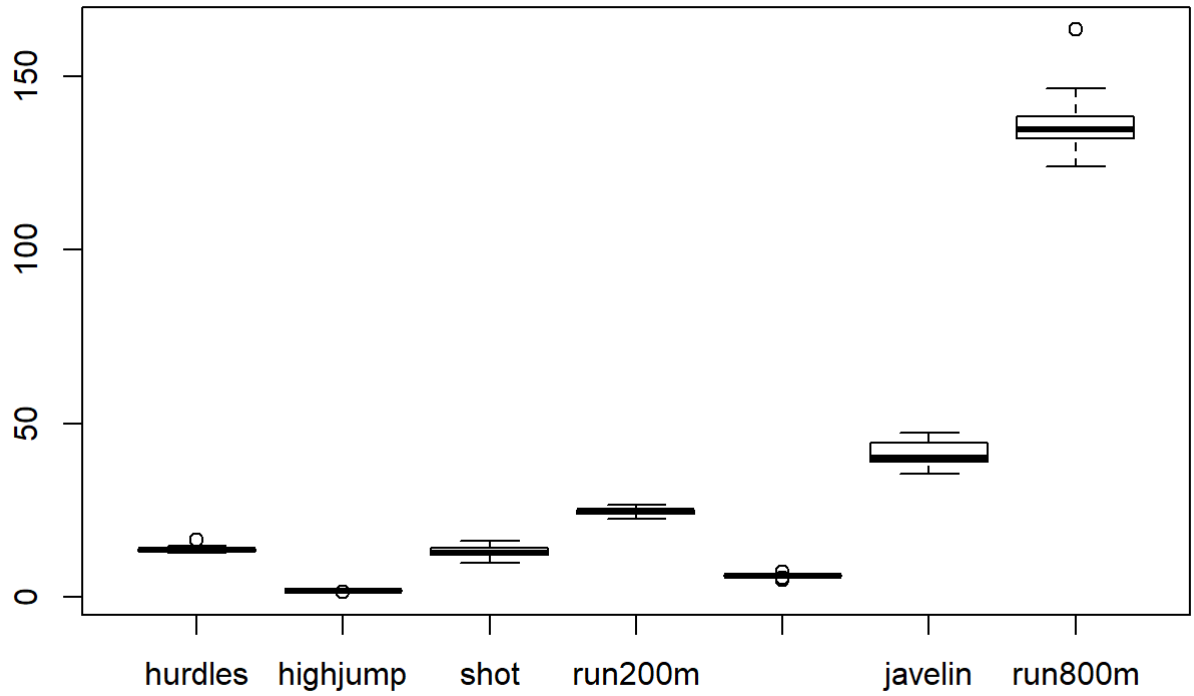
Consider the `heptahlon` dataset in the `HSAUR3` package. The dataset contains results of 1988 the olympic heptathlon competition held in Seoul. The competition contained the seven events: 100m hurdles, shot, high jump, 200m, long jump, javelin and 800m. The last column of the dataset shows the total score of the athelits.

```
head(heptathlon)
```

```
##                     hurdles highjump  shot run200m longjump javelin
## Joyner-Kersee (USA)   12.69     1.86 15.80   22.56     7.27   45.66
## John (GDR)            12.85     1.80 16.23   23.65     6.71   42.56
## Behmer (GDR)          13.20     1.83 14.20   23.10     6.68   44.54
## Sablovskaite (URS)    13.61     1.80 15.23   23.92     6.25   42.78
## Choubenkova (URS)     13.51     1.74 14.76   23.93     6.32   47.46
## Schulz (GDR)          13.75     1.83 13.50   24.65     6.33   42.82
##                     run800m score
## Joyner-Kersee (USA)  128.51  7291
## John (GDR)           126.12  6897
## Behmer (GDR)         124.20  6858
## Sablovskaite (URS)   132.24  6540
## Choubenkova (URS)    127.90  6540
## Schulz (GDR)         125.79  6411
```

Let us consider the first seven columns, and visualize their variances.

```
boxplot(heptathlon[, 1:7])
```



Clearly, the variance of `javelin` and `run800` far exceeds that of the remaining variables.

```
apply(heptathlon[, 1:7], 2, var)
```

```
##    hurdles   highjump       shot    run200m   longjump    javelin
##  0.5426500  0.0060750  2.2257190  0.9400410  0.2248773 12.5716773
##    run800m
## 68.7421417
```

A PCA of the nonstandardized variables produces PCs that are dominated by `javelin` and `run800`.

```
pcout <- prcomp(heptathlon[, 1:7])
summary( pcout )
```

```
## Importance of components:
##                           PC1    PC2     PC3     PC4     PC5     PC6
## Standard deviation     8.3646 3.5910 1.38570 0.58571 0.32382 0.14712
## Proportion of Variance 0.8207 0.1513 0.02252 0.00402 0.00123 0.00025
## Cumulative Proportion  0.8207 0.9720 0.99448 0.99850 0.99973 0.99999
##                            PC7
## Standard deviation     0.03325
## Proportion of Variance 0.00001
## Cumulative Proportion  1.00000
```

```
round( pcout$rotation[, 1:2], 3 )
```

```
##             PC1    PC2
## hurdles   0.070 -0.009
## highjump -0.006  0.001
## shot     -0.078  0.136
## run200m   0.073 -0.101
## longjump -0.040  0.015
## javelin   0.007  0.985
## run800m   0.991  0.013
```

This completely overshadows any possible patterns due to other variables, and may give us misleading interpretation.

On the other hand, the standardized dataset provides new insights.

```
Z <- scale(heptathlon[, 1:7])
pcstd <- prcomp(Z)
summary( pcstd )
```

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

```
round( pcstd$rotation[, 1:2], 3 )
```

```
##             PC1    PC2
## hurdles   0.453 -0.158
## highjump -0.377  0.248
## shot     -0.363 -0.289
## run200m   0.408  0.260
## longjump -0.456  0.056
## javelin  -0.075 -0.842
## run800m   0.375 -0.224
```

While the 2nd PC weights `javelin` highly, the 1st PC is essentilly an overall performance metric except `javelin`. To summarize, when variables have widely different variances, the data should be standardized before performing PCA.

The Romano-British pottery data

The dataset consists of 45 observations on the 9 chemicals on specimens of Romano-British pottery.

```
library(HSAUR3)
```

```
##
## Attaching package: 'HSAUR3'
```

```
## The following objects are masked from 'package:HSAUR2':
##
##     CHFLS, HSAURtable
```

```
dim(pottery)
```

```
## [1] 45 10
```

```
head(pottery)
```

```
##   Al2O3 Fe2O3  MgO  CaO Na2O  K2O TiO2   MnO   BaO kiln
## 1  18.8  9.52 2.00 0.79 0.40 3.20 1.01 0.077 0.015    1
## 2  16.9  7.33 1.65 0.84 0.40 3.05 0.99 0.067 0.018    1
## 3  18.2  7.64 1.82 0.77 0.40 3.07 0.98 0.087 0.014    1
## 4  16.9  7.29 1.56 0.76 0.40 3.05 1.00 0.063 0.019    1
## 5  17.8  7.24 1.83 0.92 0.43 3.12 0.93 0.061 0.019    1
## 6  18.8  7.45 2.06 0.87 0.25 3.26 0.98 0.072 0.017    1
```

The variable `kiln` shows the region where the specimen was made.

```
table(pottery$kiln)
```

```
##
##  1  2  3  4  5
## 21 12  2  5  5
```

There are three regions: (1=Gloucester), (2=Llanedeyrn, 3=Caldicot), and (4=Islands Thorns, 5=Ashley Rails), see **[http://people.tamu.edu/~dcarlson/quant/data/index.html] (http://people.tamu.edu/~dcarlson/quant/data/index.html)**

Les us create a new `region` variable to reflect these three regions.

```
region <- rep(NA, nrow(pottery))
region[pottery$kiln == 1] = 1
region[pottery$kiln == 2 | pottery$kiln == 3] = 2
region[pottery$kiln == 4 | pottery$kiln == 5] = 3

dat <- cbind(pottery[, 1:9], region)
head(dat)
```

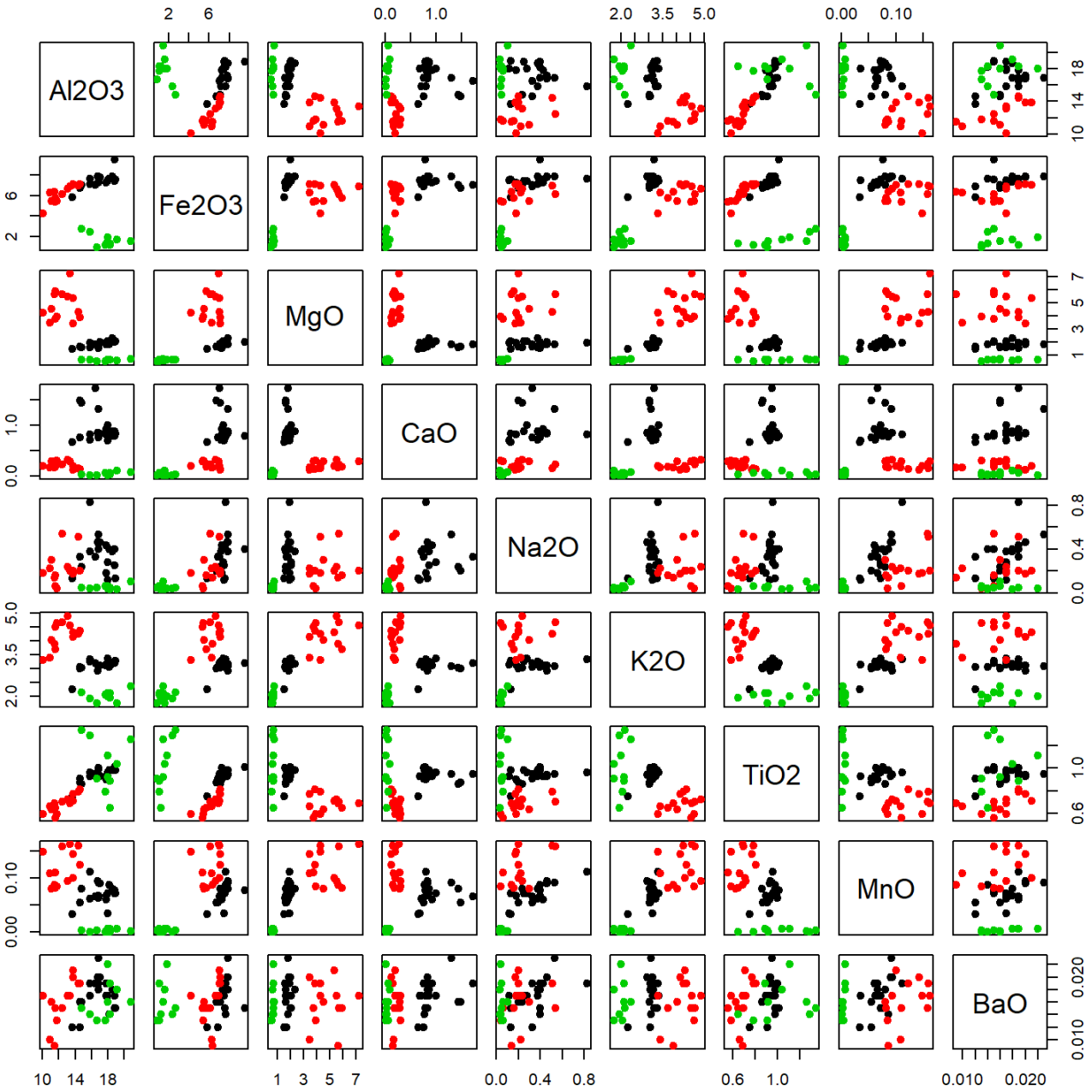
```
##   Al2O3 Fe2O3  MgO  CaO Na2O  K2O TiO2   MnO   BaO region
## 1  18.8  9.52 2.00 0.79 0.40 3.20 1.01 0.077 0.015      1
## 2  16.9  7.33 1.65 0.84 0.40 3.05 0.99 0.067 0.018      1
## 3  18.2  7.64 1.82 0.77 0.40 3.07 0.98 0.087 0.014      1
## 4  16.9  7.29 1.56 0.76 0.40 3.05 1.00 0.063 0.019      1
## 5  17.8  7.24 1.83 0.92 0.43 3.12 0.93 0.061 0.019      1
## 6  18.8  7.45 2.06 0.87 0.25 3.26 0.98 0.072 0.017      1
```

We want to ask the following questions:

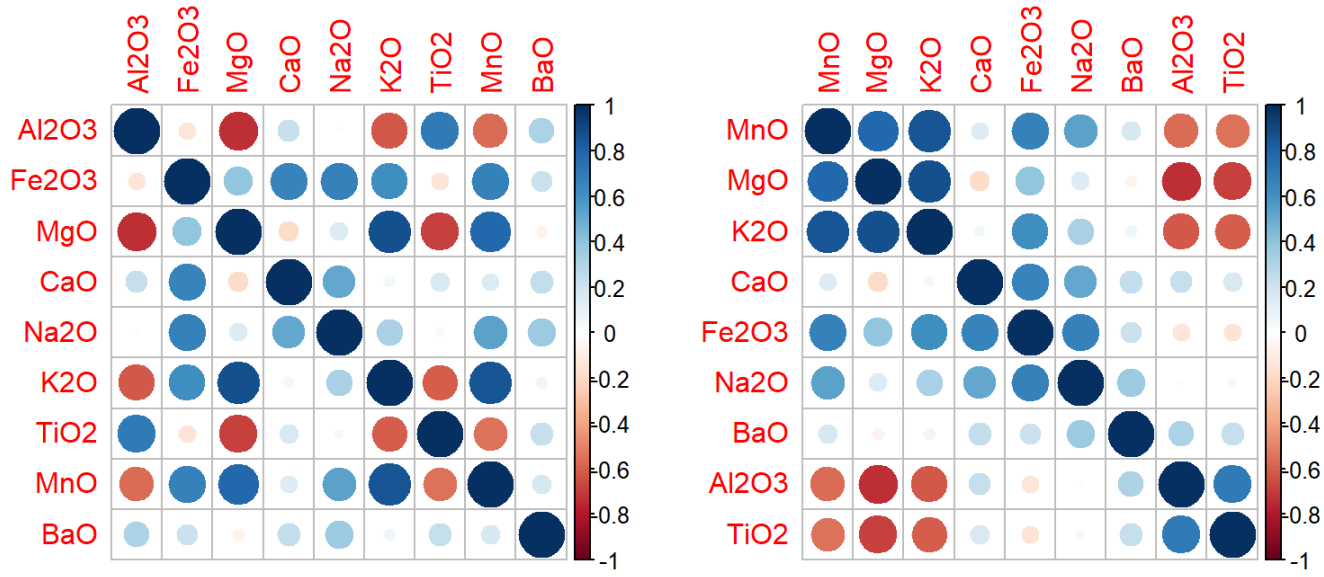
- Do we need to examine all the chemicals or just a few summaries are sufficient to capture variation in the data?
- Can we separate the specimens into differnt regions just by using the variables or their summaries?

Let us visualize the data and the correlation matrix among the chemicals.

```
plot(dat[, 1:9], pch = 19, col = region)
```




```
library(corrplot)
par(mfrow = c(1,2))
corrplot( cor(dat[, 1:9]))
corrplot( cor(dat[, 1:9]), order = "hclust" )
```



Looking at the standard deviation of the nine chemicals, it is evident that we need to standardize the data.

```
apply(dat[, 1:9], 2, sd)
```

```
##       Al2O3       Fe2O3         MgO         CaO        Na2O         K2O
## 2.703013657 2.405811419 1.742110731 0.454278427 0.178244243 0.852725577
##        TiO2         MnO         BaO
## 0.179810506 0.046801137 0.002981932
```

```
# standardize the data
std.dat <- scale(dat[, 1:9], scale = T)
```

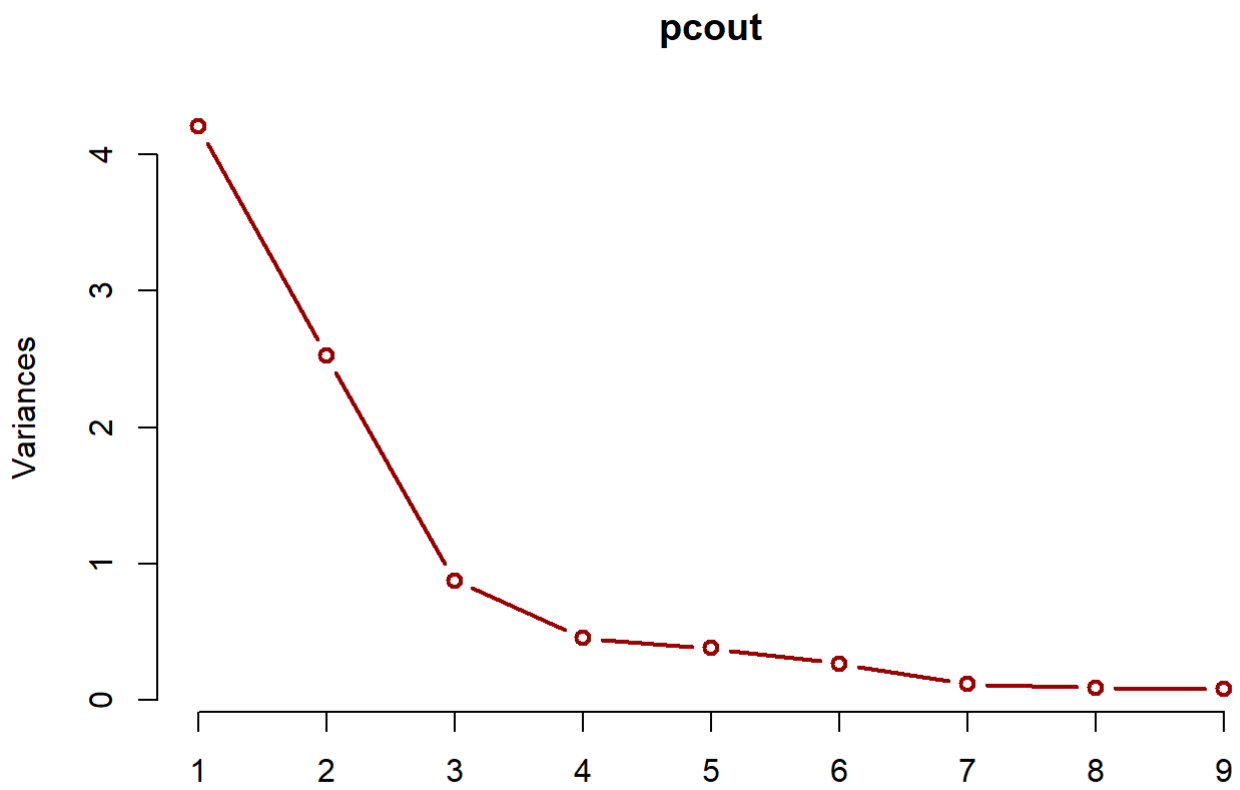
Now let us perform PCA on the standardized data.

```
pcout <- prcomp( std.dat )
summary(pcout)
```

```
## Importance of components:
##                           PC1     PC2     PC3     PC4     PC5     PC6
## Standard deviation     2.0503 1.5885 0.93699 0.67538 0.61647 0.51840
## Proportion of Variance 0.4671 0.2804 0.09755 0.05068 0.04223 0.02986
## Cumulative Proportion  0.4671 0.7475 0.84501 0.89570 0.93792 0.96778
##                           PC7     PC8    PC9
## Standard deviation     0.34325 0.30190 0.2846
## Proportion of Variance 0.01309 0.01013 0.0090
## Cumulative Proportion  0.98087 0.99100 1.0000
```

It seems that that first two or three components are desirable. A screeplot also confirms this.

```
screeplot(pcout, type = "lines", lwd=2, col="#990000")
```

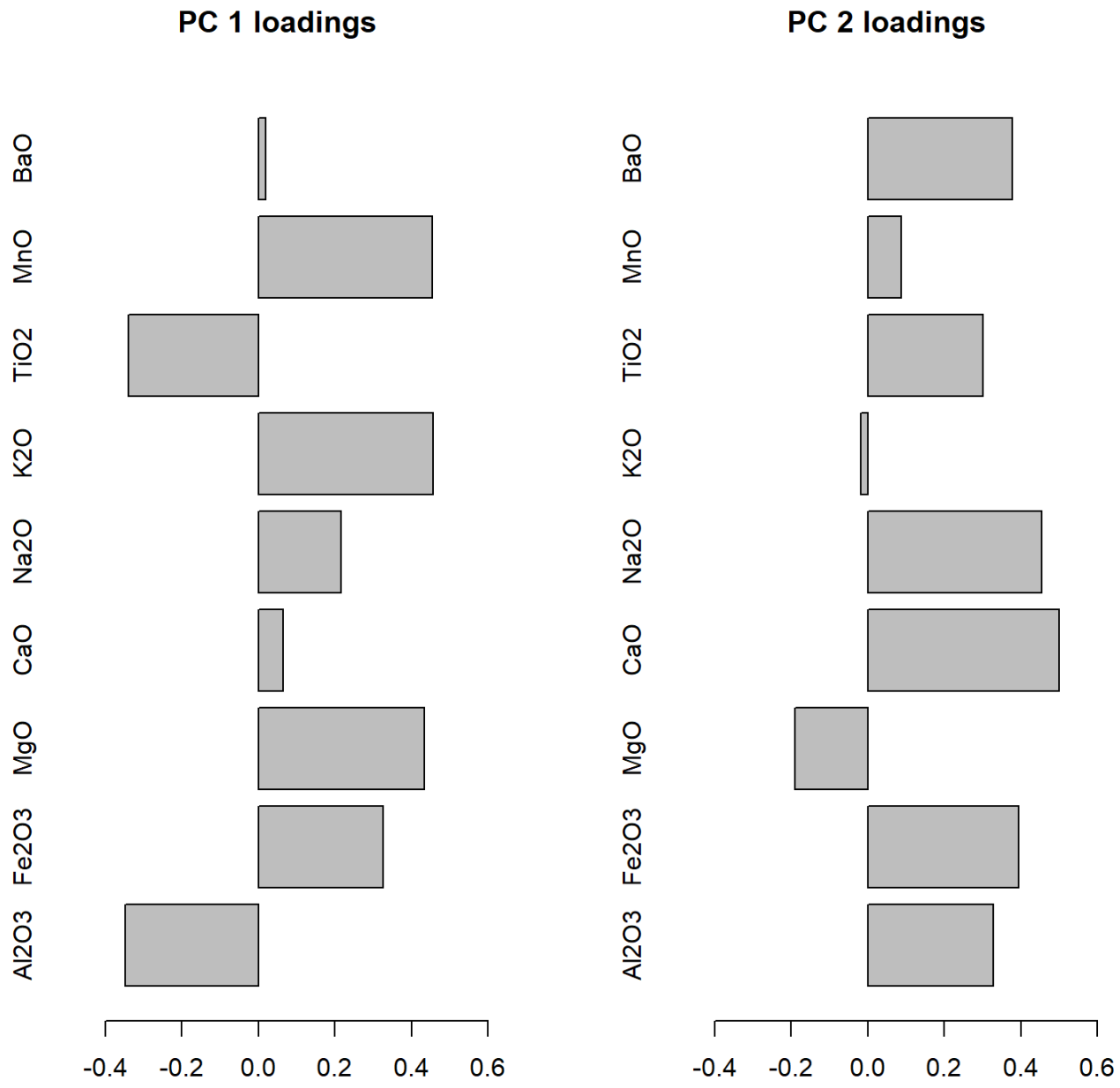


Let us inspect the loadings of the first two PCs.

```
round( pcout$rotation[, 1:2], 2 )
```

```
##          PC1   PC2
## Al2O3 -0.35  0.33
## Fe2O3  0.33  0.40
## MgO    0.43 -0.19
## CaO    0.06  0.50
## Na2O   0.22  0.46
## K2O    0.46 -0.02
## TiO2  -0.34  0.30
## MnO    0.46  0.09
## BaO    0.02  0.38
```

```
par(mfrow = c(1,2))
barplot(pcout$rotation[, 1], main = "PC 1 loadings", xlim = c(-0.5, 0.6), horiz = T)
barplot(pcout$rotation[, 2], main = "PC 2 loadings", xlim = c(-0.5, 0.6), horiz = T)
```

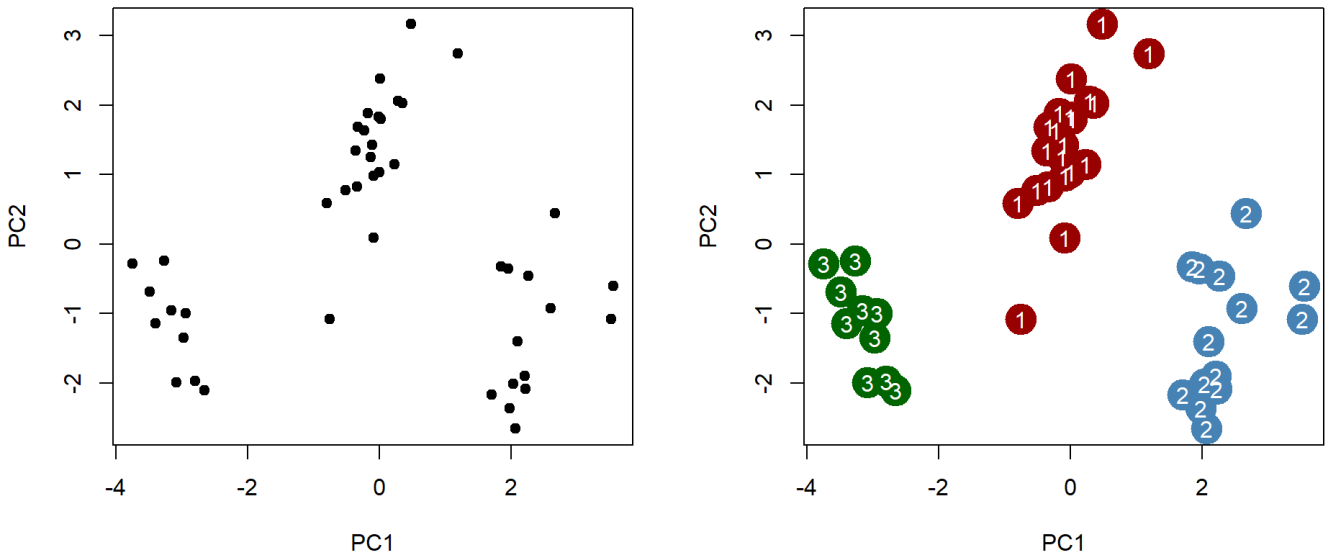


Recall that there are three regions where the specimens were made. We did not use the region information at all while performing the PCA. Let us plot PC1 score versus PC2 score and see whether we can discover any groups among the specimens.

```
PC1 <- pcout$x[, 1]
PC2 <- pcout$x[, 2]

par(mfrow = c(1,2))
# Plot the PC scores
plot(PC1, PC2, pch=19)

# Plot the PC scores with marked regions
colvec = c("#990000", "steelblue", "darkgreen")
plot(PC1, PC2, pch=19, col = colvec[region], cex=3)
text(PC1, PC2, labels = region, col = "white")
```

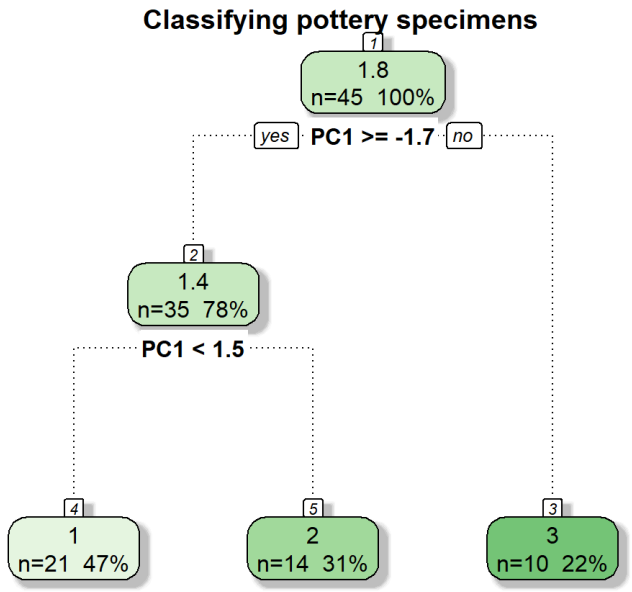


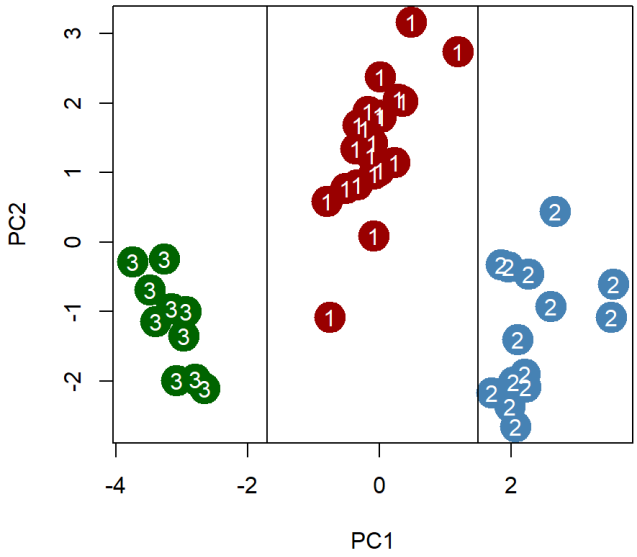
The three groups have been perfectly separated by plotting the first two PCs (we only need PC1 actually). A classification rule can be displayed in the tree form just using the first two PCs.

```
library(rpart)                  # decision tree algorithm
library(rattle)                   # tree plot

# create decision tree
tree <- rpart(region ~ PC1 + PC2)

par(mfrow = c(1,2))
# display tree
fancyRpartPlot(tree, sub="", main = "Classifying pottery specimens")
# Plot the PC scores with marked regions
colvec = c("#990000", "steelblue", "darkgreen")
plot(PC1, PC2, pch=19, col = colvec[region], cex=3)
text(PC1, PC2, labels = region, col = "white")
abline(v = -1.7)
abline(v = 1.5)
```





The rule a specimen belong to region 1 if $PC1 \in [-1.7, 1.5)$, region 2 if $PC1 \geq 1.5$, and region 3 if $PC1 < -1.7$. We will learn about such trees in a future lecture.

Main page: **ST 437/537: Applied Multivariate and Longitudinal Data Analysis (https://maityst537.wordpress.ncsu.edu/)**