Principal Components Analysis

PCA was invented by Karl Pearson in 1901. It is a very widely used statistics learning method.

Motivation i.e. why PCA?

For data with many variables/dimensions it is difficult to comprehend or visualize the associations between them.

There may be high levels of multicollinearity.

PCA 're-expresses' the data to account for most of the information in the data through a few linear combinations of the original variables.

Imagine that a data set is made up of three positively correlated variables.

Could one linear combination of the three variables be sufficient to convey most of the information given by the 3 individual variables?

How does PCA work?

Thus the overall aim in PCA is to describe the variation in a set of *correlated* variables $x_1, x_2, \ldots, x_p$ in terms of a new set of *uncorrelated* variables $y_1, y_2, \ldots, y_q$ (where $q << p$) where each y_j is a linear combination of the $x_1, x_2, \ldots, x_p$ variables.

The new coordinate values that represent the data are known as the **principal components** (PCs).

The new 'variables' are derived in decreasing order of importance in the sense that the first new variable y_1 accounts for the most variation in the original data, out of all linear combinations of $x_1, x_2, \ldots, x_p$.

The second principal component y_2 is chosen to account for as much as possible of the remaining variation, subject to the constraint that y_2 is uncorrelated with y_1. And so on...

The general hope is that the first few principal components will account for a substantial amount of the variation in the original variables and can be used as a convenient lower dimensional summary of the data.

How do we get the PCs?

PCA is performed by solving the eigenvalue problem or by iterative algorithms which estimate the principal components.

The data: n cases, p variables.

Arrange in an $\mathbf{X}_{n\times p}$ matrix, ie cases are in rows, variables are in columns

x_{ij} = value of jth variable for case i

Let $\mathbf{x_i}$ denote the ith row of $\mathbf{X}$, represented as a column vector.

Let $\mathbf{x_j}$ denote the jth column of $\mathbf{X}$.

The observations are centered.

The sample covariance matrix for centered data is given by:

$$\mathbf{S} = \frac{1}{n}\mathbf{X}^T\mathbf{X}.$$

Geometrically, the sample covariance matrix determines the shape of an ellipsoid centered at zero:

$$\mathbf{x_i^T S^{-1} x_i} = \mathbf{z^2}$$

where z is the distance from a point $\mathbf{x_i}$ to the zero mean.

Eigen decomposition of the sample covariance matrix finds the directions of the principal axes of the ellipse.

The sample covariance matrix is symmetric and can be decomposed as:

$$\mathbf{S} = \mathbf{V}\Lambda\mathbf{V}^T,$$

where $\mathbf{V}$ is a $p \times p$ dimensional matrix of eigenvectors $\mathbf{v}_j$, and Λ is a diagonal matrix of all the eigenvalues $\lambda_j \geq 0$, $\forall j = 1, ..., p$.

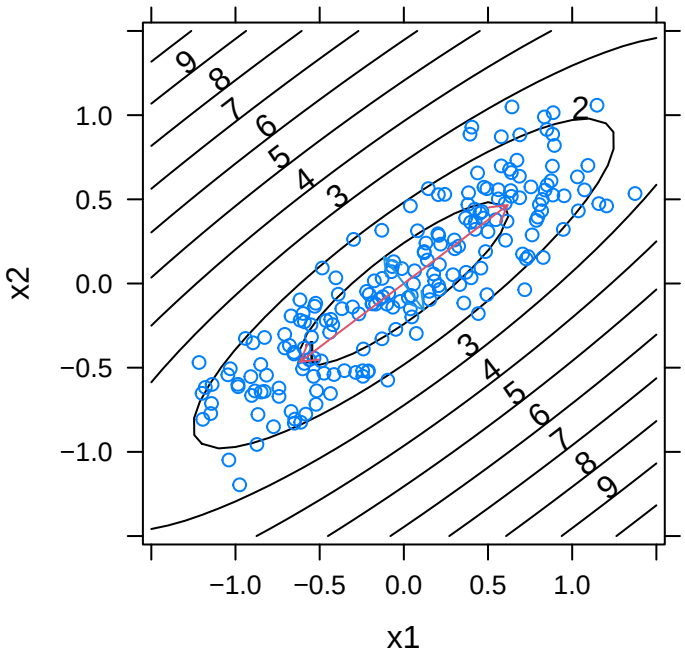
Post-multiplying both sides of the eigen-decomposition by $\mathbf{V}$ obtained is:

$$\mathbf{SV} = \mathbf{V}\Lambda$$

since $\mathbf{V}^T\mathbf{V} = \mathbf{0}$ as the eigenvectors are uncorrelated.

Geometrically, the eigenvectors correspond to the directions of the principal axes of the ellipse.

Each eigenvalue corresponds to the variance of a linear combination of the features using weights given by the associated eigenvector.



PCA implicitly makes assumptions that the features have a linear relationship and elliptical dispersion.

Principal components: definition

The first principal component (PC) is the linear combination

$$\mathbf{y_1} = \mathbf{X}\mathbf{v_1}$$

where $\mathbf{v_1}$ is an eigenvector associated with the largest eigenvalue λ_1.
The second principal component is the linear combination

$$\mathbf{y_2} = \mathbf{X}\mathbf{v_2}$$

where $\mathbf{v_2}$ is the eigenvector associated with the second largest eigenvalue etc. Note eigenvectors obtained from the covariance matrix are orthonormal, i.e both are unit vectors and $\mathbf{v_2}$ is perpendicular to $\mathbf{v_1}$, i.e. $\mathbf{v_1^t v_2} = \mathbf{0}$.

The jth principal component is the linear combination

$$\mathbf{y_j} = \mathbf{X}\mathbf{v_j}$$

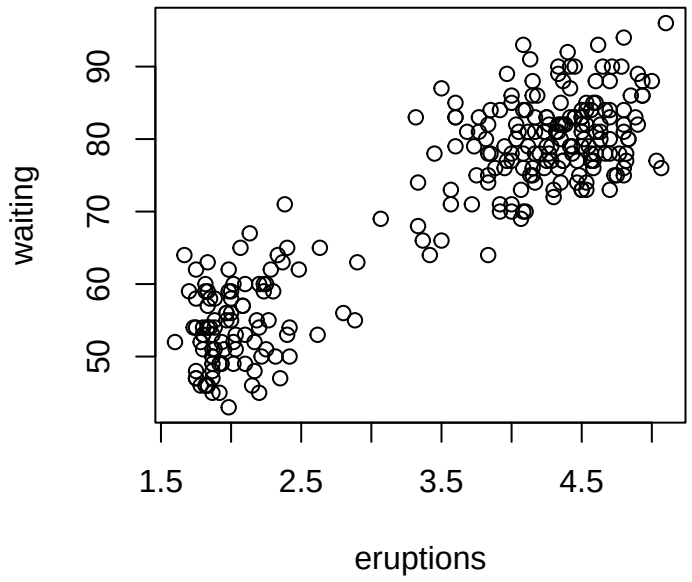
, where $\mathbf{v_j}$ is the eigenvector corresponding to the jth largest eigenvalue and is perpendicular to $\mathbf{v_k}$ for $k = 1, 2, \ldots, j-1$.

- The vectors $\mathbf{v_j}$ are called the **principal component directions** or the **loading vectors**.
- The vectors $\mathbf{y_j}$ are called the **principal component scores**.
- While there are at most min$(n-1, p)$ principal components, we would hope that the variance of most of them is low.
- In this way, principal components is said to be a **dimension reduction technique.**
- When the variables are not measured in comparable units it is customary to scale the variables to have standard deviation of 1, prior to computing the principal components.
- As a consequence of the definitions above, the principal components $\mathbf{y_j}$ and $\mathbf{y_k}$ are uncorrelated, for $k \neq j$.
- There is arbitrariness in the signs of $\mathbf{v_j}$ i.e. could multiply by by -1.
- The variance of $\mathbf{y_j}$ is λ_j.
- If the variables are scaled to have unit standard deviation, then $\mathbf{v_j}$ are the eigen vectors of $\mathbf{R}$.

Old faithful data, $p = 2$ variables.

Waiting time between eruptions and the duration of the eruption for the Old Faithful geyser in Yellowstone National Park, Wyoming, USA. Usually principal components is applied to data with large p. We will use $p = 2$ for our first example.

```
plot(faithful)
```



For the Old faithful data both variables are measured in time, but the standard deviations of the variables are very different. Therefore it is appropriate to calculate PCA based on the scaled variables.

```
p <- prcomp(faithful,scale=TRUE)
p

Standard deviations (1, .., p=2):
[1] 1.38 0.31

Rotation (n x k) = (2 x 2):
            PC1   PC2
eruptions -0.71  0.71
waiting   -0.71 -0.71
```

This output tells us that

- The first PC direction is

$$\mathbf{v_1} = \begin{bmatrix} -.707 \\ -.707 \end{bmatrix}$$

- The first score vector is

$$\mathbf{y_1} = -.\mathbf{707} \times \text{eruptions} - .\mathbf{707} \times \text{waiting}$$

- The standard deviation of $\mathbf{y_1}$ is 1.38.
- The second PC direction is

$$\mathbf{v_2} = \begin{bmatrix} .707 \\ -.707 \end{bmatrix}$$

- The second score vector is

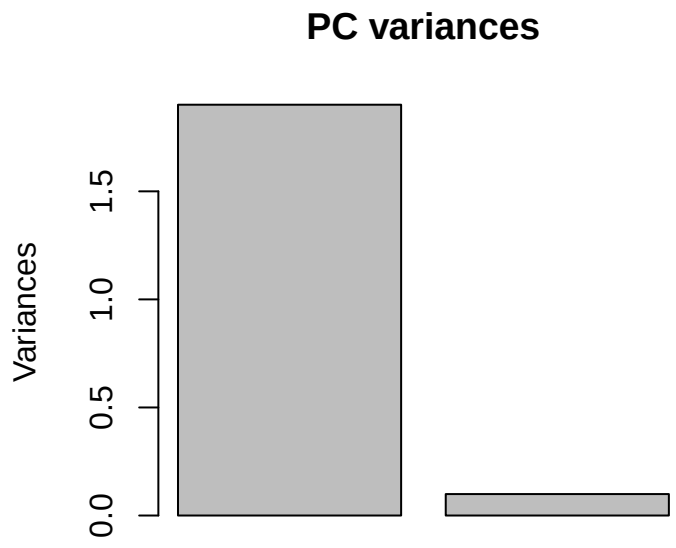
$$\mathbf{y_2} = .\mathbf{707} \times \text{eruptions} - .\mathbf{707} \times \text{waiting}$$

- The standard deviation of $\mathbf{y_2}$ is 0.31

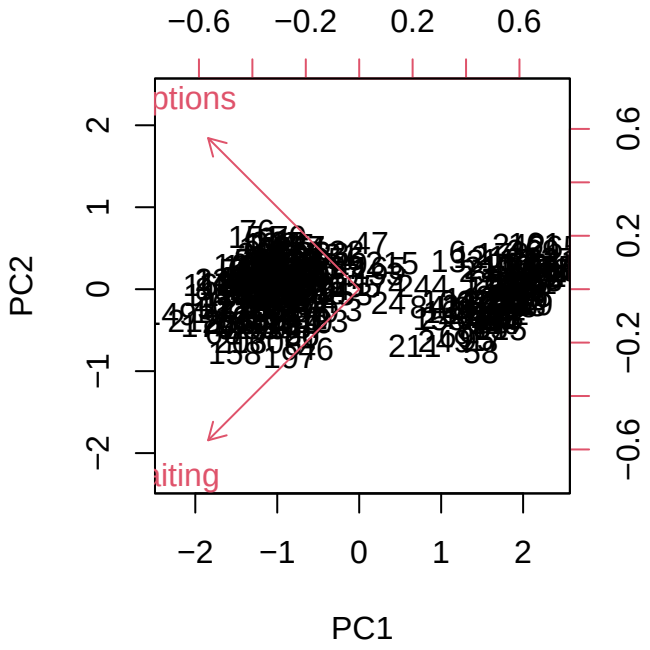
Plotting the results

This plot shows the variances of the two principal components.

```
plot(p, main="PC variances")
```

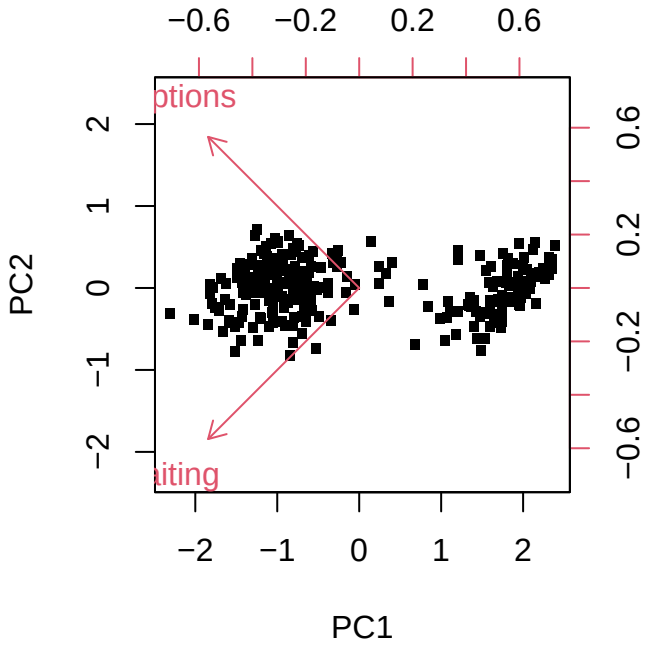


```
biplot(p, scale=0)
```



To get rid of all the numbers use

```
biplot(p, scale=0, xlabs=rep(".", nrow(faithful)), cex=c(3,1))
```



This plot shows $\mathbf{y_1}$ on the x axis and $\mathbf{y_2}$ on the y axis.

The arrows are proportional to the loadings.

The biplot is so-called because it shows both loadings and scores.

Notice the biplot is simply a rotation of the data.

If we did not standardise the variables:

```
p <- prcomp(faithful)
p

Standard deviations (1, .., p=2):
[1] 13.63  0.49

Rotation (n x k) = (2 x 2):
             PC1    PC2
eruptions -0.076  0.997
waiting   -0.997 -0.076
```

Now the first score vector is

$$\mathbf{y_1} = -.\mathbf{07} \times \text{eruptions} - .\mathbf{99} \times \text{waiting}$$

and the second score vector is

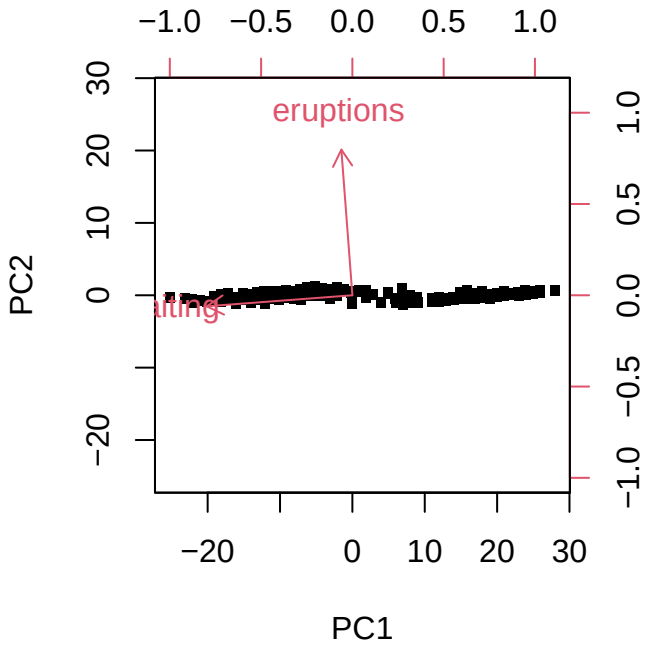
$$\mathbf{y_2} = .\mathbf{997} \times \text{eruptions} - .\mathbf{07} \times \text{waiting}$$

ie the first score vector is - waiting, and the second score vector is eruptions.

This is because the values in the waiting variable have a bigger variance than eruptions.

Using scaled variables for PCA is usually advised unless variables have similar scales.

```
biplot(p, scale=0, xlabs=rep(".", nrow(faithful)), cex=c(3,1))
```



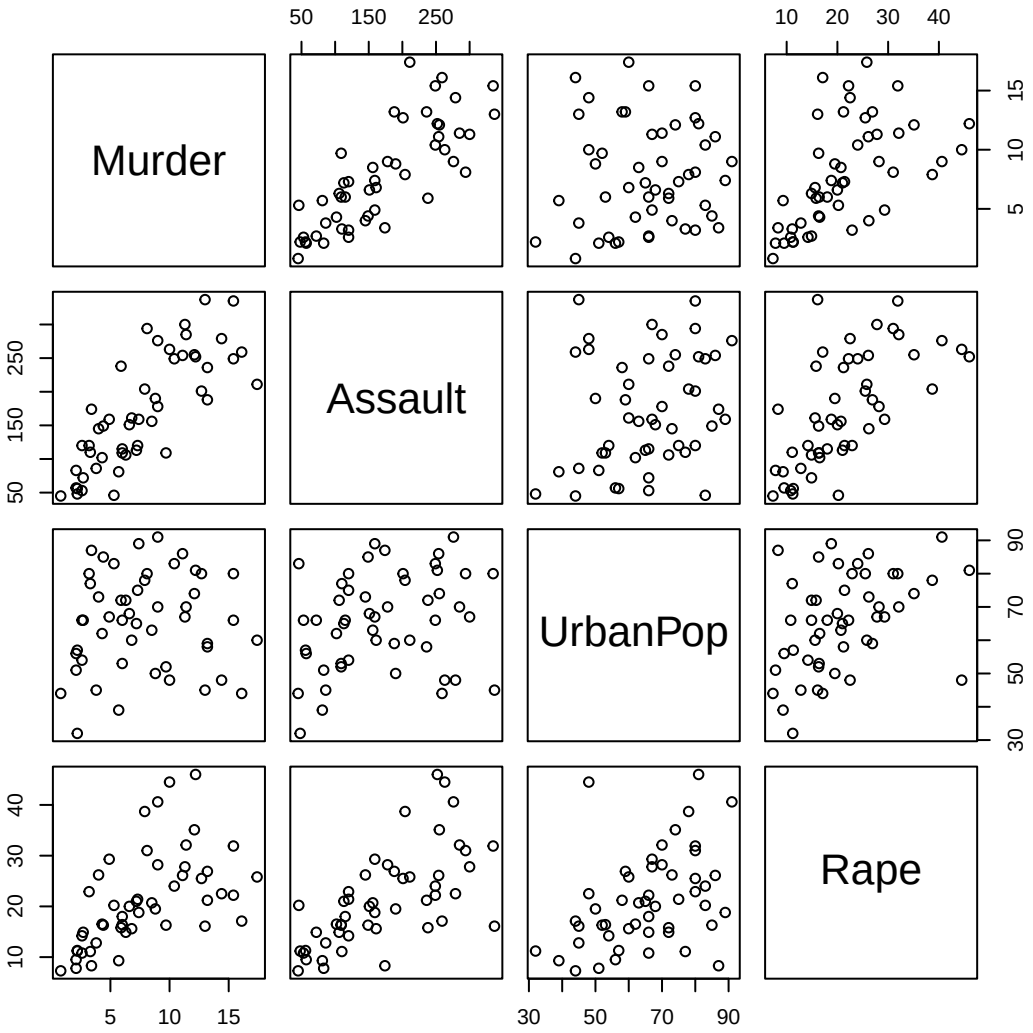
Example: USArrests data

This data set contains statistics, in arrests per 100,000 residents for assault, murder, and rape in each of the 50 US states in 1973. Also given is the percent of the population living in urban areas.

```
head(USArrests)
```

```
           Murder Assault UrbanPop Rape
Alabama      13.2     236       58   21
Alaska       10.0     263       48   44
Arizona       8.1     294       80   31
Arkansas      8.8     190       50   20
California    9.0     276       91   41
Colorado      7.9     204       78   39
```

```
plot(USArrests)
```




```
apply(USArrests,2, var)

  Murder  Assault UrbanPop     Rape
      19     6945      210       88
```

As the variances of the 4 variables is radically different we should do PCA on scaled variables.

```
p <- prcomp(USArrests,scale=TRUE)
p

Standard deviations (1, .., p=4):
[1] 1.57 0.99 0.60 0.42


Rotation (n x k) = (4 x 4):
           PC1    PC2    PC3     PC4
```

```
Murder   -0.54  0.42 -0.34  0.649
Assault  -0.58  0.19 -0.27 -0.743
UrbanPop -0.28 -0.87 -0.38  0.134
Rape     -0.54 -0.17  0.82  0.089
```

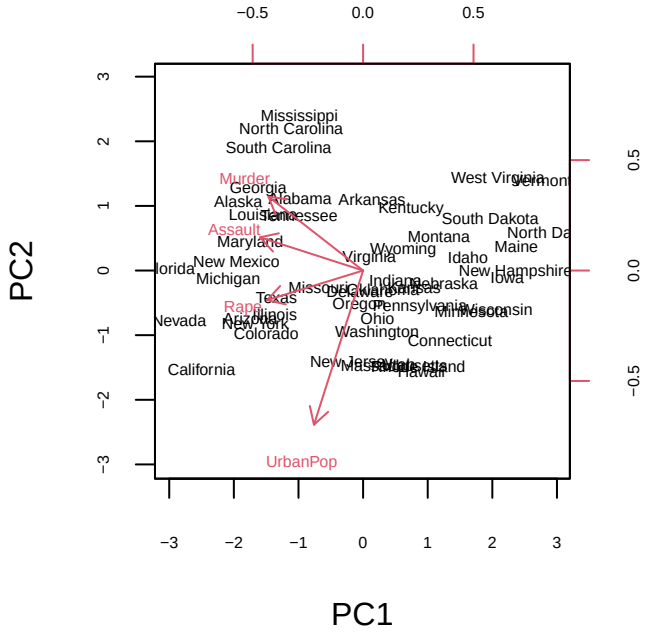
The first component is -(.53 Murder + .58 Assault + .27 UrbanPop + .54 Rape)
where variables are scaled.

This component is mostly a measure of the overall crime rate, (as UrbanPop gets a lower weight than other variables).

High crime states score low and low crime states score high.

The second component gives most weight to UrbanPop, so high population states will have a low score on PC2.

```
biplot(p, scale = 0, cex = c(.5,.5), cex.axis = .5)
```



States on the upper right quadrant are high on PC1 and PC2, so mostly have low crime and low population.

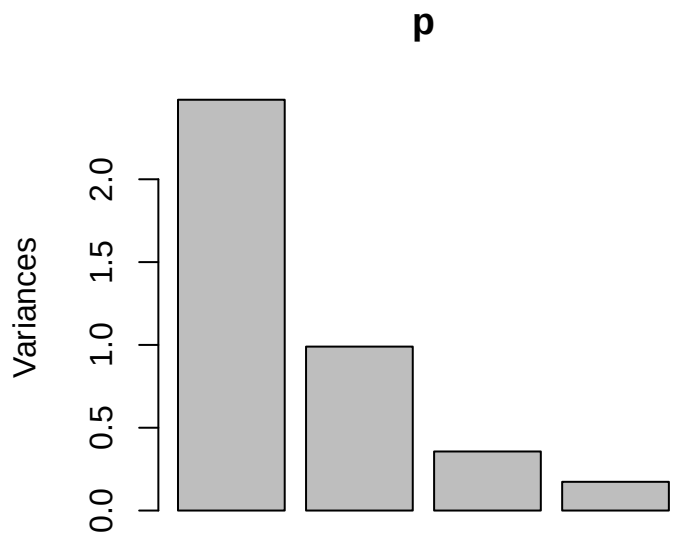
States on the lower right quadrant are high on PC1 and low on PC2, so mostly have low crime and high population.

States in the upper left quadrant have mostly high crime and low population, while those on the lower left quadrant have high crime and population.

States close to the middle (eg Virginia) are about average on crime and UrbanPop.

The fact that the arrows for Murder, Assault and Rape are close to each other indicates the correlation of these 3 variables.

```
plot(p)
```



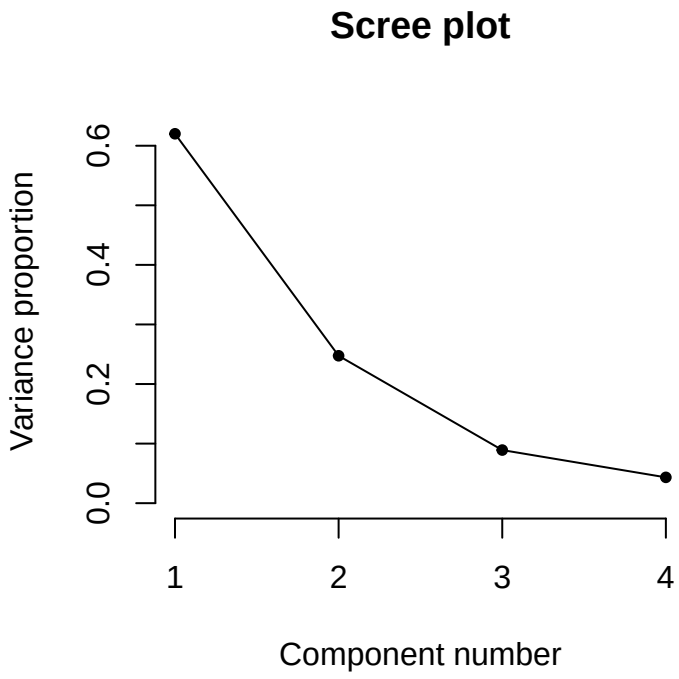
The above plot shows the variances.

Sometimes it helps to calculate the proportion of variance explained by each principal component,

```
summary(p)

Importance of components:
                          PC1    PC2    PC3    PC4
Standard deviation     1.57 0.995 0.5971 0.4164
Proportion of Variance 0.62 0.247 0.0891 0.0434
Cumulative Proportion  0.62 0.868 0.9566 1.0000
```

and to plot this: (the so-called screeplot)

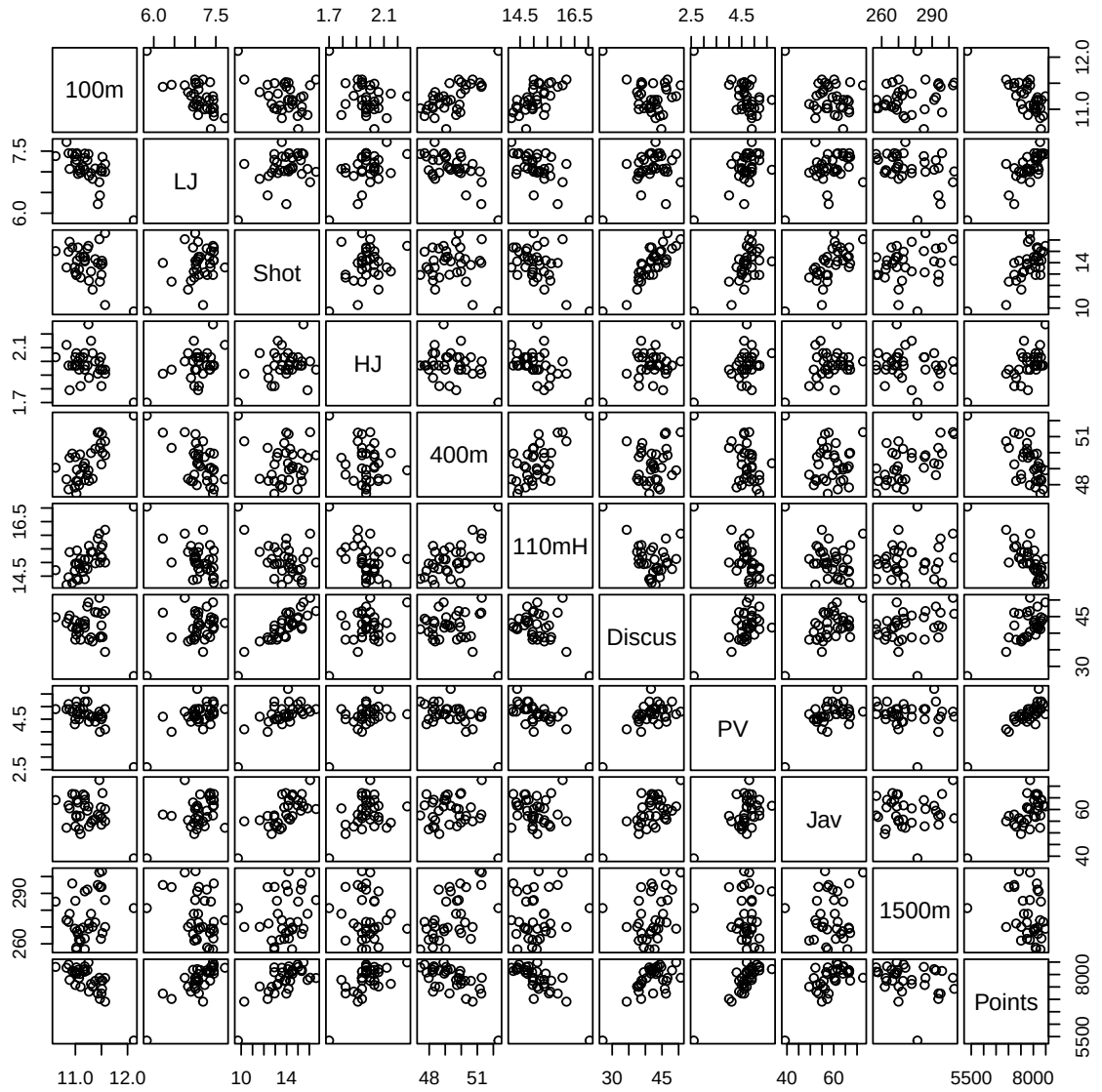
```r
screeplot <- function(p) {
  e <- p$sdev ^ 2
  e <- e / sum(e)
  plot(
    1:length(e),
    e,
    xlab = "Component number",
    pch = 20,
    ylab = "Variance proportion",
    main = "Scree plot",
    axes = F,
    ylim = c(0, max(e)*1.04)
  )
  lines(1:length(e), e)
  axis(1, at = 1:length(e))
  axis(2)
}
screeplot(p)
```



We can see that 86% of the variability in the dataset is accounted for by the first two components.

Example: Olympics – decathlon data

```
olympic <- read.table("https://raw.githubusercontent.com/rafamoral/courses/main/intro
                      header = TRUE, check.names = FALSE)
head(olympic)

  100m  LJ Shot  HJ 400m 110mH Discus  PV Jav 1500m Points
1   11 7.4   15 2.3   49    15     49 4.7  61   269   8488
2   11 7.5   15 2.0   48    14     44 5.1  62   273   8399
3   11 7.4   14 2.0   48    15     44 5.2  64   263   8328
4   11 7.4   15 2.0   49    15     45 4.9  64   285   8306
5   11 7.4   13 2.0   47    14     41 5.2  57   257   8286
6   11 7.7   14 2.1   48    14     43 4.9  52   274   8272

pairs(olympic, gap=.3)
```



This data set gives the performances of 34 men's decathlon at the Olympic Games (1988). There is also the score, which are the final points scores of the competition.

```
p <- prcomp(olympic[,-11], scale=TRUE)
```

```
p$rotation[,1:2]
```

```
           PC1     PC2
100m    -0.359  0.2042
LJ       0.361 -0.1979
Shot     0.324  0.3943
HJ       0.267 -0.0074
400m    -0.294  0.4272
110mH   -0.373  0.1312
```

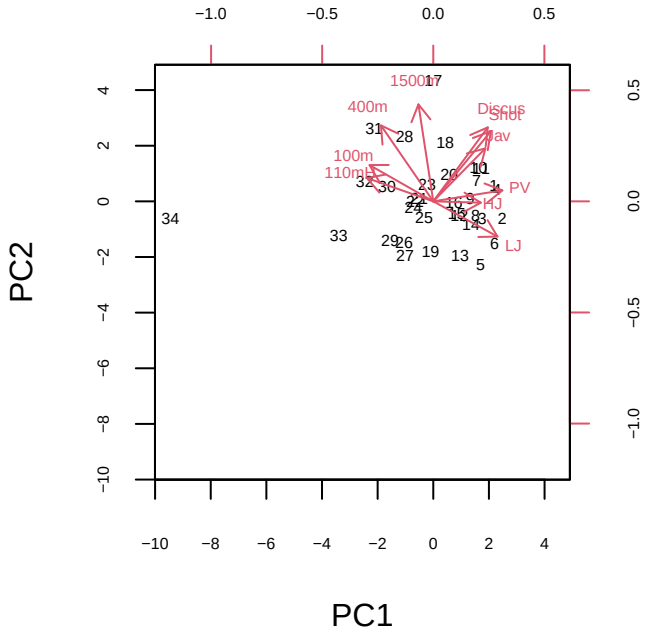
```
Discus  0.307  0.4164
PV      0.389  0.0619
Jav     0.293  0.2984
1500m  -0.084  0.5456

summary(p)

Importance of components:
                          PC1   PC2    PC3    PC4    PC5    PC6    PC7    PC8
Standard deviation     2.241 1.442 0.8576 0.8281 0.6134 0.5496 0.5343 0.4731
Proportion of Variance 0.502 0.208 0.0736 0.0686 0.0376 0.0302 0.0285 0.0224
Cumulative Proportion  0.502 0.710 0.7839 0.8525 0.8901 0.9203 0.9488 0.9712
                          PC9   PC10
Standard deviation     0.4525 0.2881
Proportion of Variance 0.0205 0.0083
Cumulative Proportion  0.9917 1.0000
```

For this data, 50% of the variability is in the first component, 71% of the variability is in the first two components.

The first component has negative weights for the events where the result is measured in time (ie small values are good performances) and positive weights for events where the result is measured in metres (ie large values are good performances). The first component is an overall measure of performance.

The second component has positive weights for all variables except long jump (ignore high jump). The second component contrasts performance on throwing events (shot, discus and javelin) with that on the running events + long jump (speed events).

```
biplot(p, scale=0, cex=c(.5,.5), cex.axis=.5)
```



The most obvious issue with this plot is case 34 is an outlier. He has the worst performance across the board.

This case may also be affecting the results of the PCA, so we omit it.

```
p <- prcomp(olympic[-34,-11], scale=TRUE)
p$sdev

 [1] 1.85 1.61 0.97 0.94 0.75 0.70 0.66 0.55 0.52 0.32

p$rotation[,1:2]

         PC1    PC2
100m   -0.42  0.149
LJ      0.39 -0.152
Shot    0.27  0.484
HJ      0.21  0.028
400m   -0.36  0.352
110mH  -0.43  0.070
Discus  0.18  0.503
PV      0.38  0.150
Jav     0.18  0.372
1500m  -0.17  0.421
```

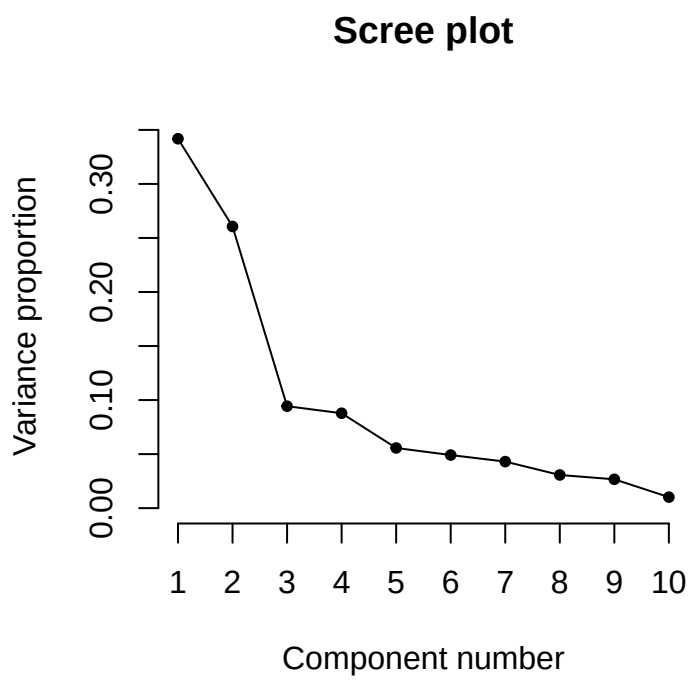
```
e <- p$sdev ^ 2
e <- e / sum(e)
e
```

```
 [1] 0.342 0.261 0.094 0.088 0.056 0.049 0.043 0.031 0.027 0.010
```

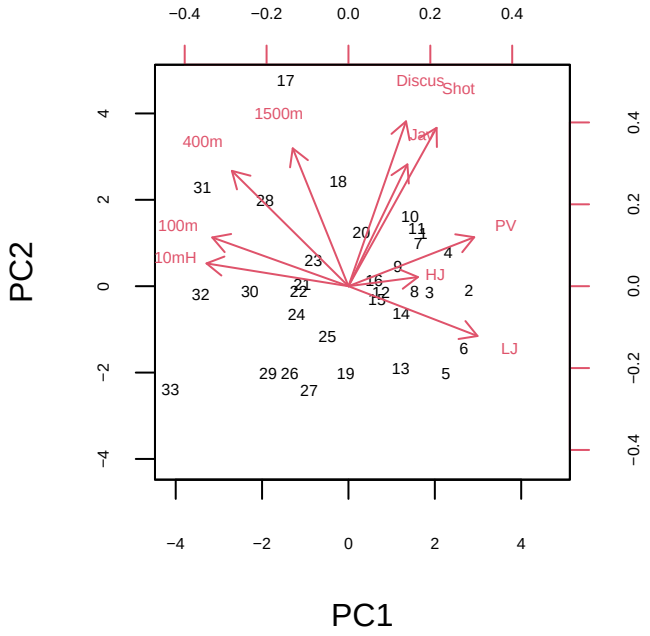
```
cumsum(e)
```

```
 [1] 0.34 0.60 0.70 0.78 0.84 0.89 0.93 0.96 0.99 1.00
```

```
screeplot(p)
```

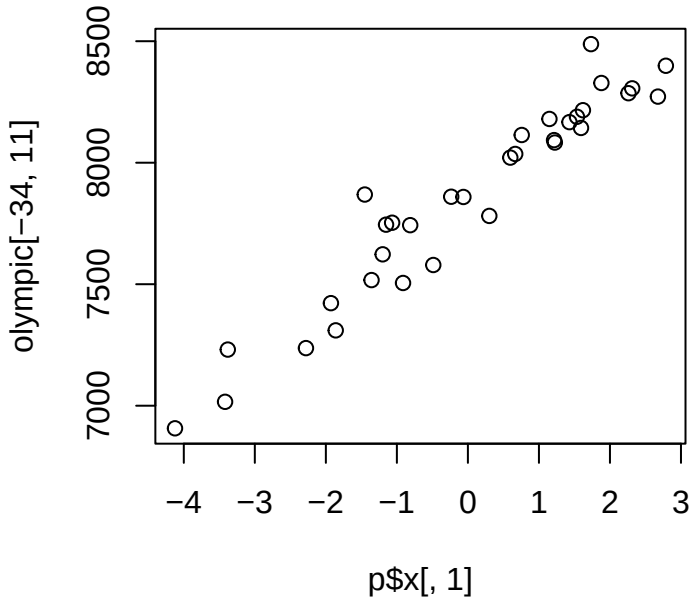



```
biplot(p, scale=0, cex=c(.5,.5), cex.axis=.5)
```



- Less variance is explained by the first PC than before.
  The interpretation of the coefficients is similar.
- Athletes on the first and fourth quadrants are high on PC1. These have a good performance overall. This is confirmed by the point labels on the biplot, small labels are on the right, high labels on the left.
- Athletes with large values on PC2 have a large difference in the contrast of throwing events with speed events.
- This tells that athletes in the upper right of the plot e.g. 10, 11, 1, 7 are high performers, especially at throwing

```
plot(p$x[,1], olympic[-34,11])
```


olympic[−34, 11]
8500
8000
7500
7000
−4
−3
−2
−1
0
1
2
3
p$x[, 1]