Unsupervised Learning

COMP 2031/8031

Unsupervised learning

- **Unsupervised learning** (UML), no labels are provided, and the learning algorithm focuses solely on detecting structure in unlabelled input data.
- **Clustering**, where the goal is to find homogeneous subgroups within the data; the grouping is based on distance between observations.
- **Dimensionality reduction**, where the goal is to identify patterns in the features of the data.

https://lgatto.github.io/IntroMachineLearningWithR/unsupervised-learning.html#introduction

- **Dimensionality reduction** techniques are widely used and versatile techniques that can be used to:
  - find structure in features
  - pre-processing for other ML algorithms, and
  - aid in visualisation.

Revisiting K-means clustering

k-means clustering algorithms aims at partitioning *n* observations into a fixed number of *k* clusters. The algorithm will find homogeneous clusters.

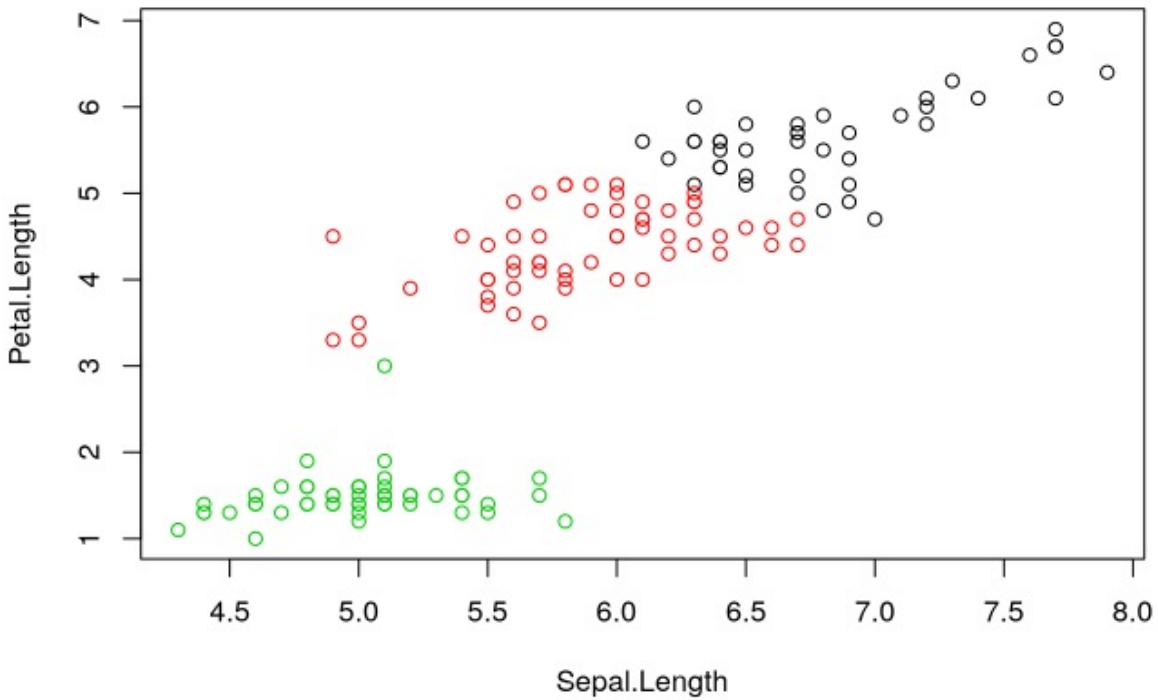
In R, stats::kmeans(x, centers = 3, nstart = 10)

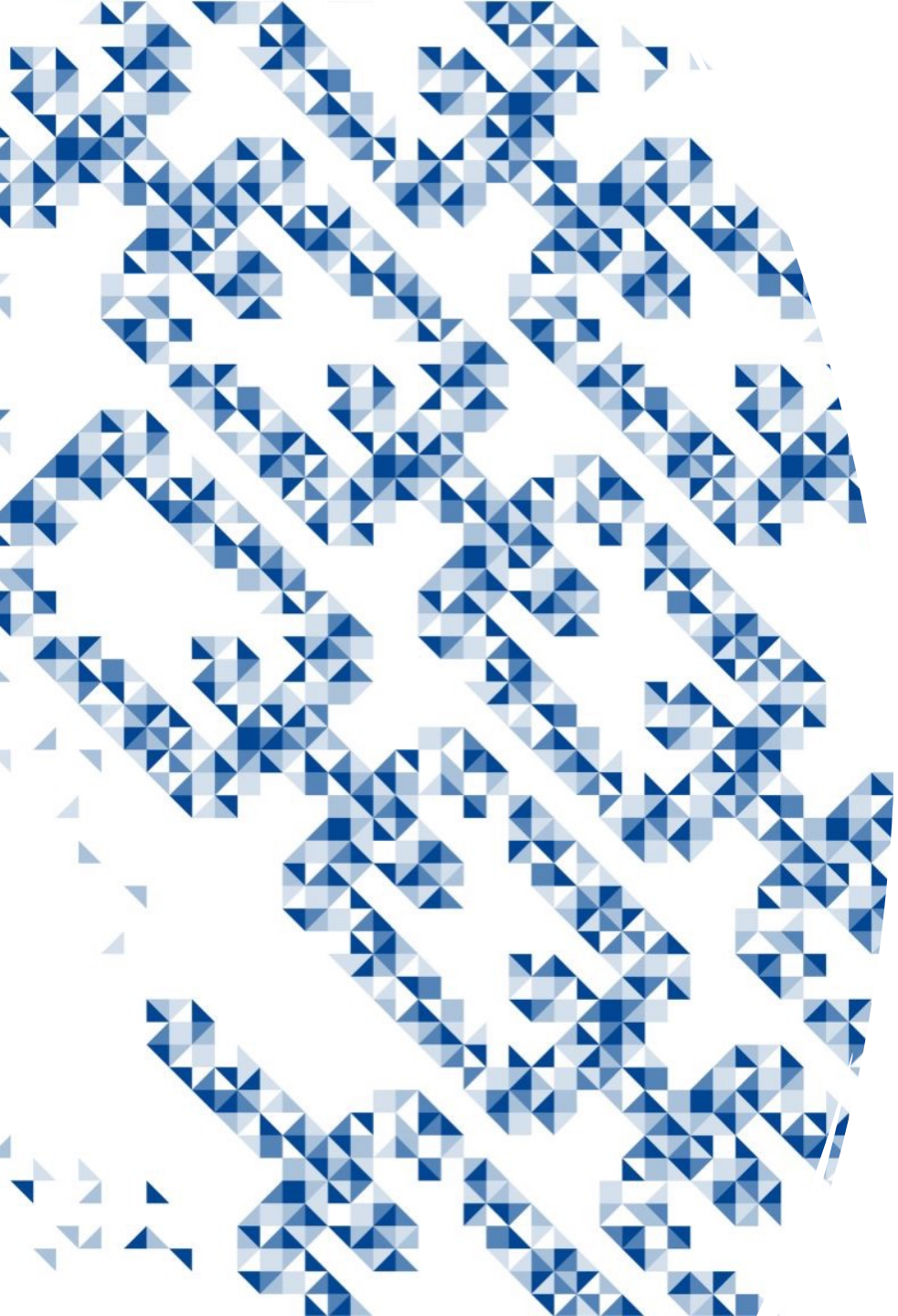
x is a numeric data matrix

centers is the pre-defined number of clusters

the k-means algorithm has a random component and can be repeated nstart times to improve the returned model

Iris data





Iteration

- Calculate the centre of each subgroup as the average position of all observations is that subgroup.
- Each observation is then assigned to the group of its nearest centre.
- It’s also possible to stop the algorithm after a certain number of iterations, or once the centres move less than a certain distance.

A random k-means iteration

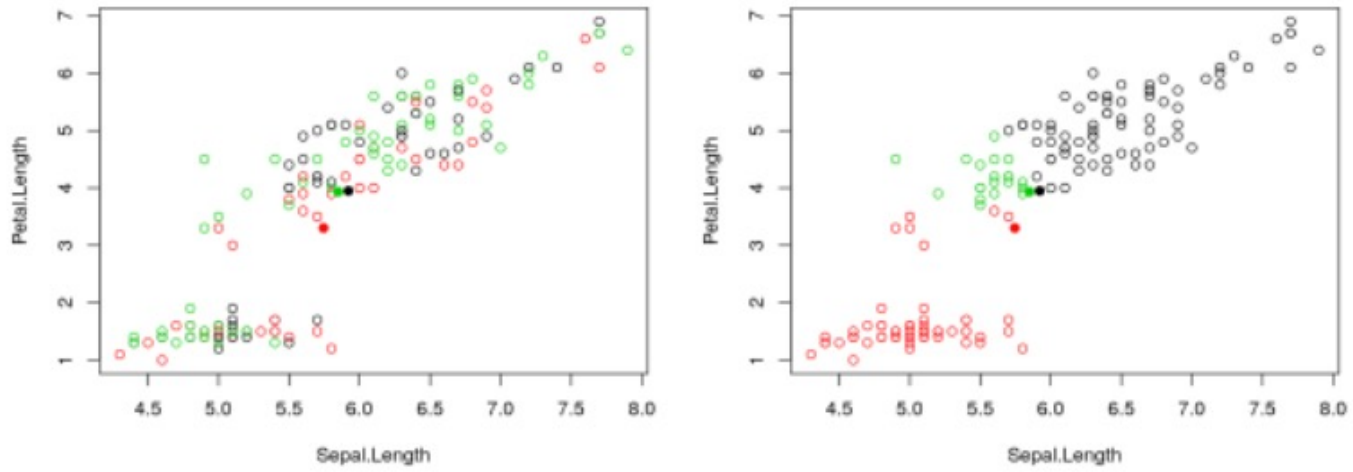
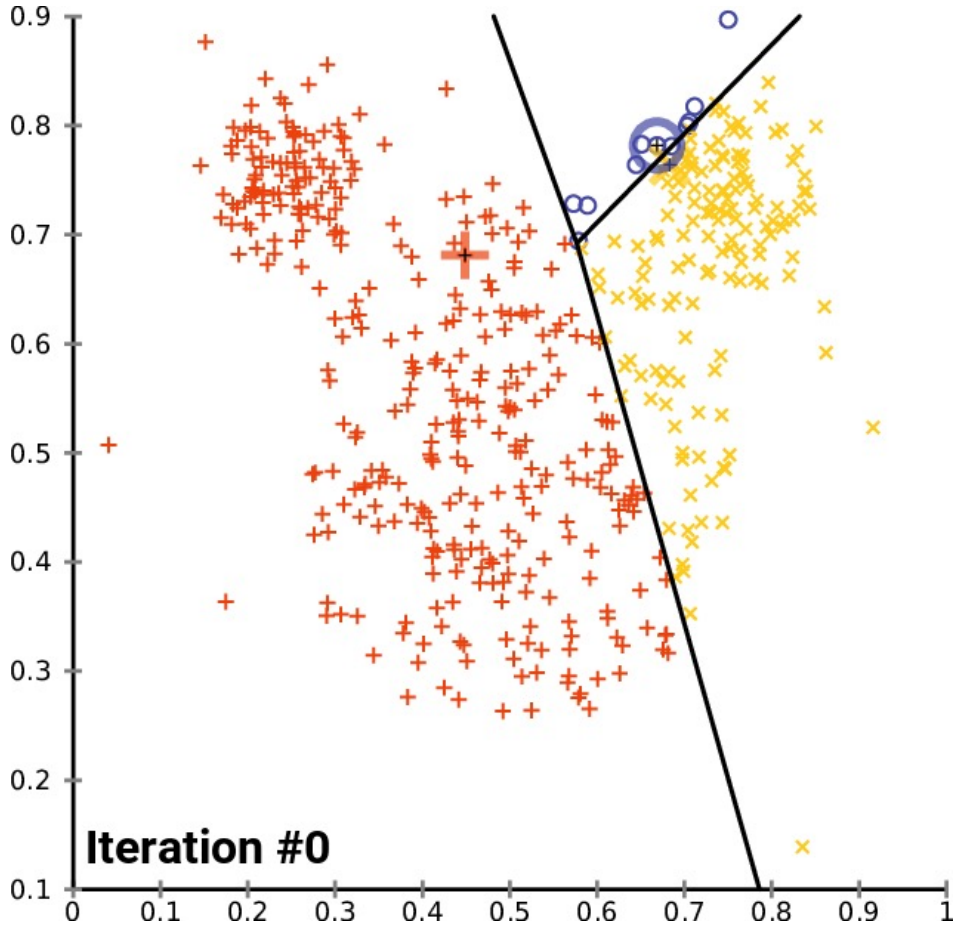


Figure 4.3: k-means iteration: calculate centers (left) and assign new cluster membership (right)

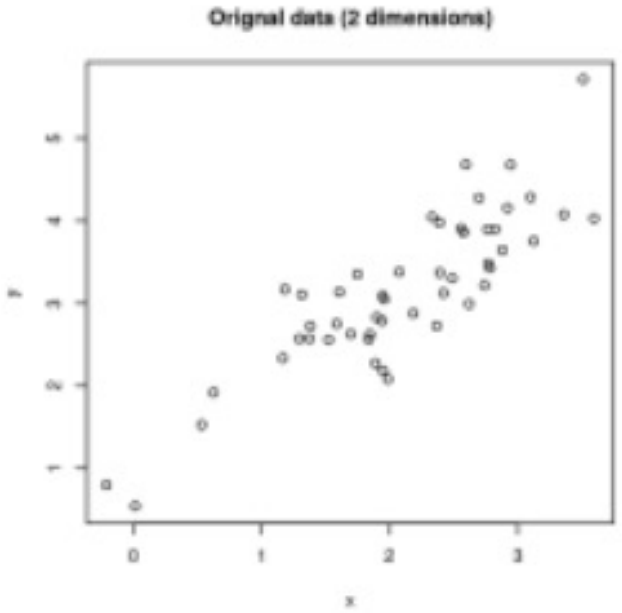
k-means convergence (credit Wikipedia)

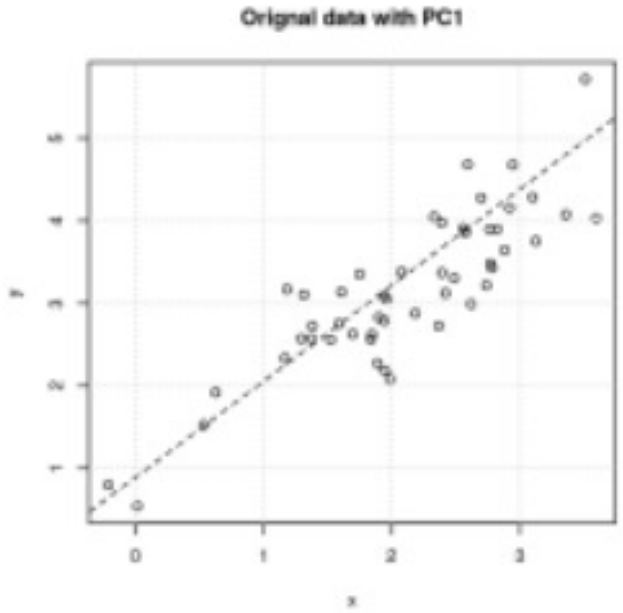


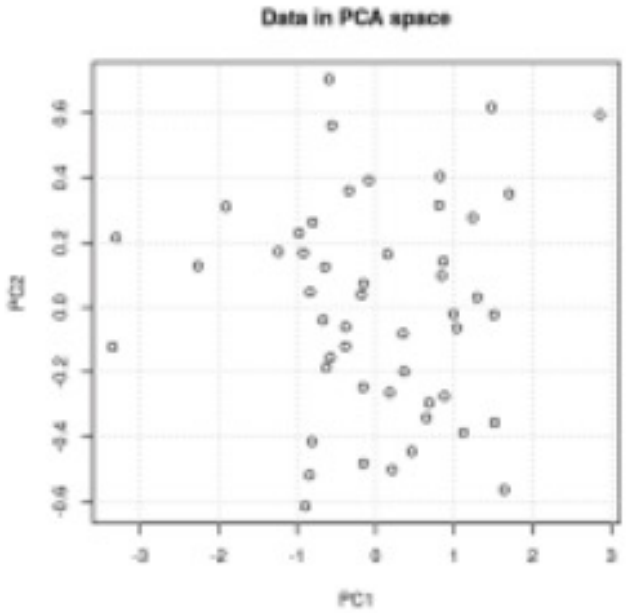
Principal Component Analysis (PCA)

- “(PCA) is a technique that transforms the original n-dimensional data into a new n-dimensional space”

Data in PCA space



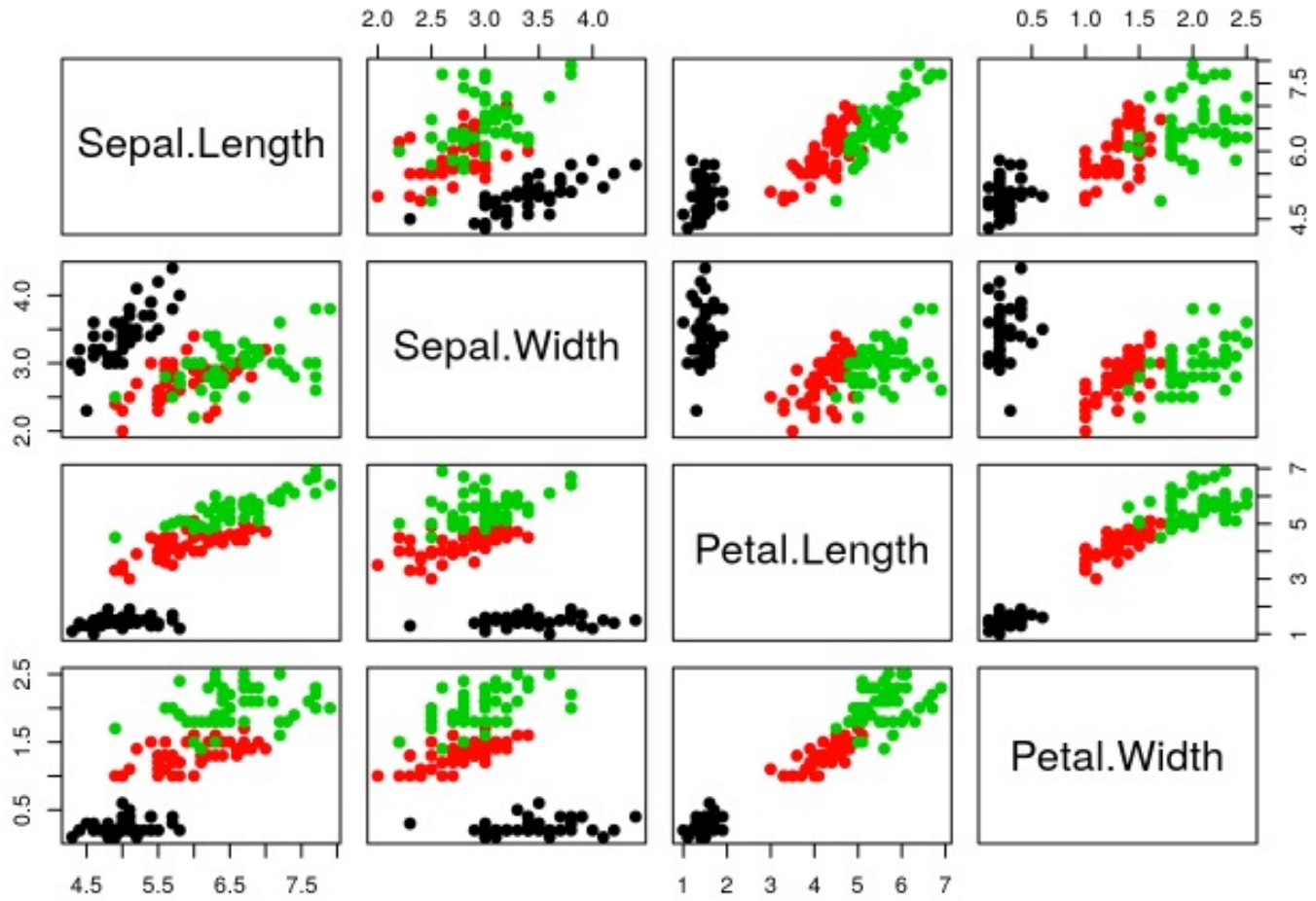




PCA on Iris data

- Hard to visualize more than 3 variables (3D)!
- For instance:

```
pairs(iris[, -5], col = iris[, 5], pch = 19)
```



Let’s reduce the dimensions!

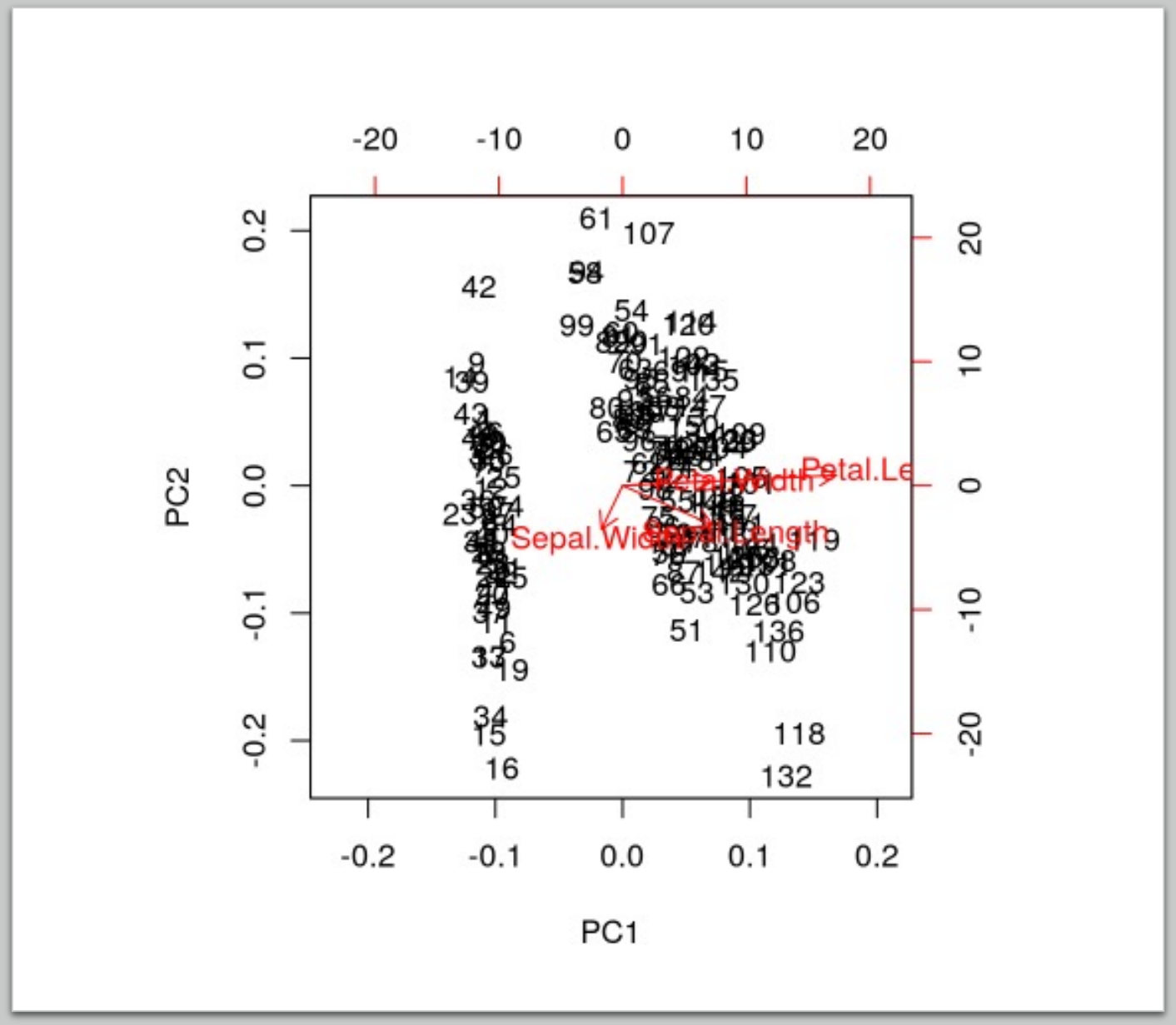
```
irispca <- prcomp(iris[, -5])
summary(irispca)
```

```
## Importance of components:
##                           PC1     PC2    PC3     PC4
## Standard deviation     2.0563 0.49262 0.2797 0.15439
## Proportion of Variance 0.9246 0.05307 0.0171 0.00521
## Cumulative Proportion  0.9246 0.97769 0.9948 1.00000
```

output shows that along PC1 along, we are able to retain over 92% of the total variability in the data

Visualisation

- A **biplot** features all original points re-mapped (rotated) along the first two PCs as well as the original features as vectors along the same PCs. Feature vectors that are in the same direction in PC space are also correlated in the original data space.
- biplot(irispca)



PCA in R

- var <- irispca$sdev^2
- (pve <- var/sum(var))

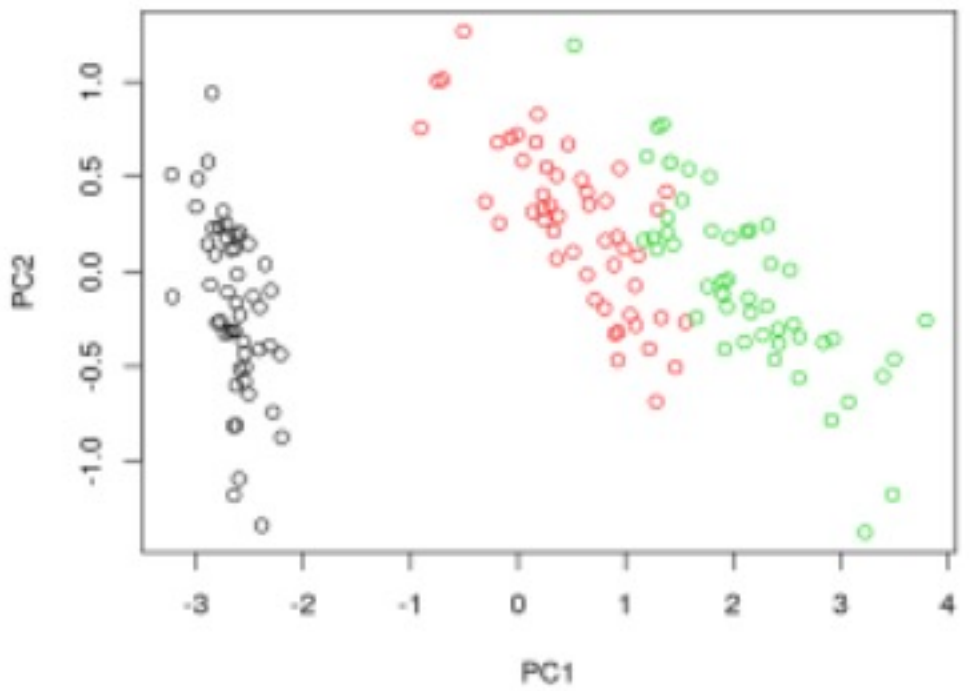
```
## [1] 0.924618723 0.053066483 0.017102610 0.005212184
```

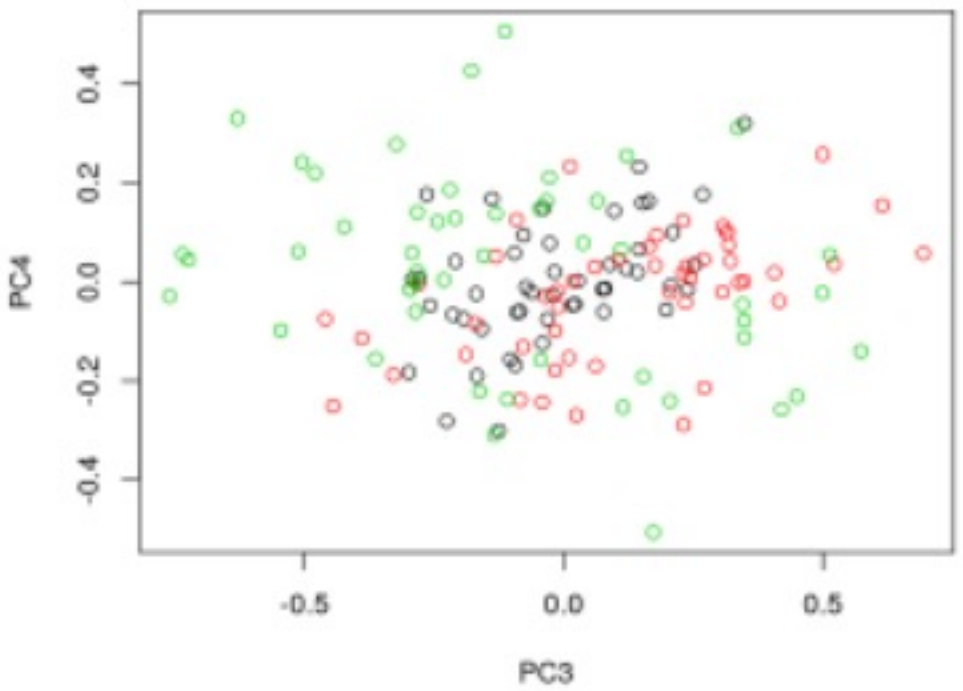
- cumsum(pve)

```
## [1] 0.9246187 0.9776852 0.9947878 1.0000000
```

PCA Plots for Iris data

- Reproduce the PCA plots below, along PC1 and PC2 and PC3 and PC4 respectively.





Missing values and categorical data

- PCA cannot deal with missing values, and observations containing *NA* values will be dropped automatically.

Categorical Variables

it is possible to encode categories as binary **dummy variables**

	x	y
A	1	0
B	0	1
C	0	0