

# MA5810: Introduction to Data Mining

## Week 4; Collaborate Session 1: Clustering

Martha Cooper, PhD

JCU Masters of Data Science

2019-21-9 (updated: 2020-11-18)

# Housekeeping

- Collaborate 1 = [Wednesdays 6-7pm](#) (Martha)
- Collaborate 2 = [Thursdays 7-8pm](#) (Hongbin)

For my Collaborate Sessions, you can get the [slides & R code](#) for each week here:

<https://github.com/MarthaCooper/MA5810>



# Subject: MA5810 Intro to Data Mining

## MA5810 Learning Outcomes

1. Overview of Data Mining and Examples
2. Unsupervised data mining methods e.g. clustering and outlier detection;
3. Unsupervised and supervised techniques for dimensionality reduction;
4. Supervised data mining methods for pattern classification;
5. Apply these concepts to real data sets using R (Today).

# Today's Goals

- Understand how  $K$ -means clustering works
- Understand how to obtain the optimal value for  $K$
- Understand the pros and cons of  $K$ -means clustering
- Apply  $K$ -means clustering on real datasets using R

# Unsupervised Learning

A set of statistical tools to understand a set of features,  $X_1, \dots, X_p$ , without having an associated response variable,  $Y$ , to predict.

Data visualization & identification of subgroups

## Clustering

A broad class of methods for discovering unknown subgroups (*clusters*) within data

Why Cluster?

- Bioinformatics - identify subtypes of cancer from gene expression data
- Marketing- identify subgroups of shoppers who buy certain products

# Cluster Analysis

Clustering looks to find homogeneous subgroups among the observations

Observations should be:

- similar to observations within the same cluster
- different to observations in different cluster

How do we determine these similarities & differences?

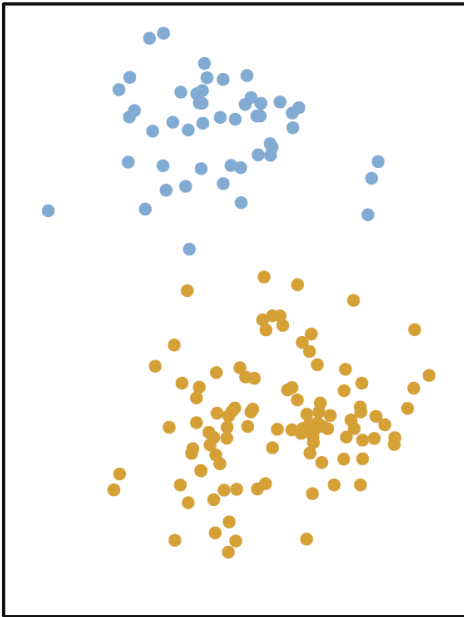
Two approaches:

- $K$ -means clustering
- Hierarchical Clustering

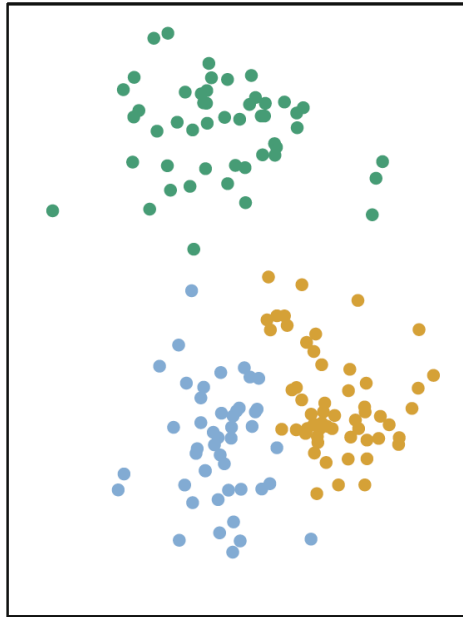
# K-Means Clustering

- Partition a dataset into  $K$  distinct, non-overlapping clusters
- We define the number of clusters,  $K$ , and the  $K$ -Means algorithm will assign each observation to *one* of those clusters

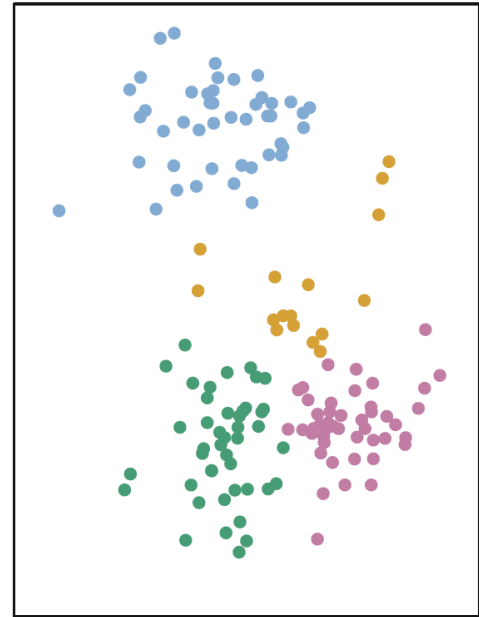
K=2



K=3

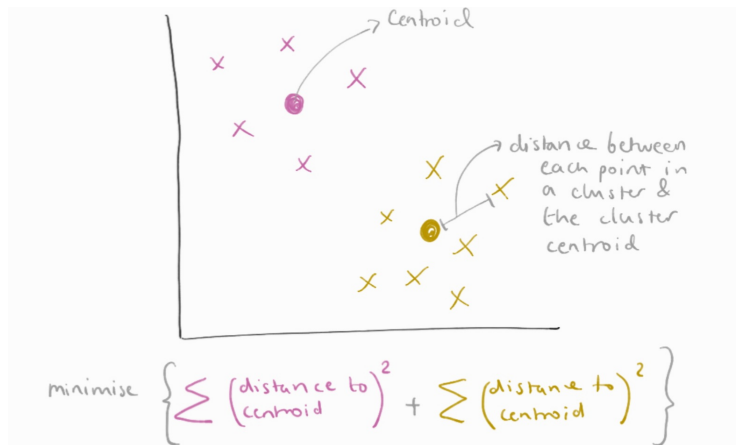


K=4



# How does *K*-means select the best clusters?

- Minimize total within cluster variation



*K* clusters    *n* observations    *p* variables

$$\text{minimise } \left\{ \sum_{k=1}^K \sum_{i=1}^n \sum_{j=1}^p (x_{ij}^{(k)} - c_{kj})^2 \right\}$$

*j*<sup>th</sup> variable for observation *x<sub>i</sub>*    in cluster *k*    centroid of cluster *k* for the *j*<sup>th</sup> variable

The sum of within cluster variation across all clusters



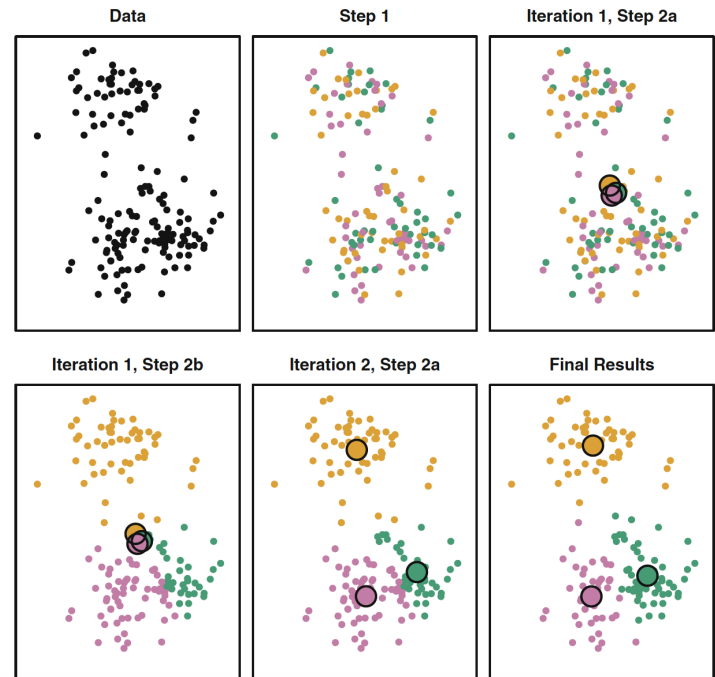
# K-means clustering algorithm

---

**Algorithm 10.1** *K-Means Clustering*

---

1. Randomly assign a number, from 1 to  $K$ , to each of the observations. These serve as initial cluster assignments for the observations.
  2. Iterate until the cluster assignments stop changing:
    - (a) For each of the  $K$  clusters, compute the cluster *centroid*. The  $k$ th cluster centroid is the vector of the  $p$  feature means for the observations in the  $k$ th cluster.
    - (b) Assign each observation to the cluster whose centroid is closest (where *closest* is defined using Euclidean distance).
- 

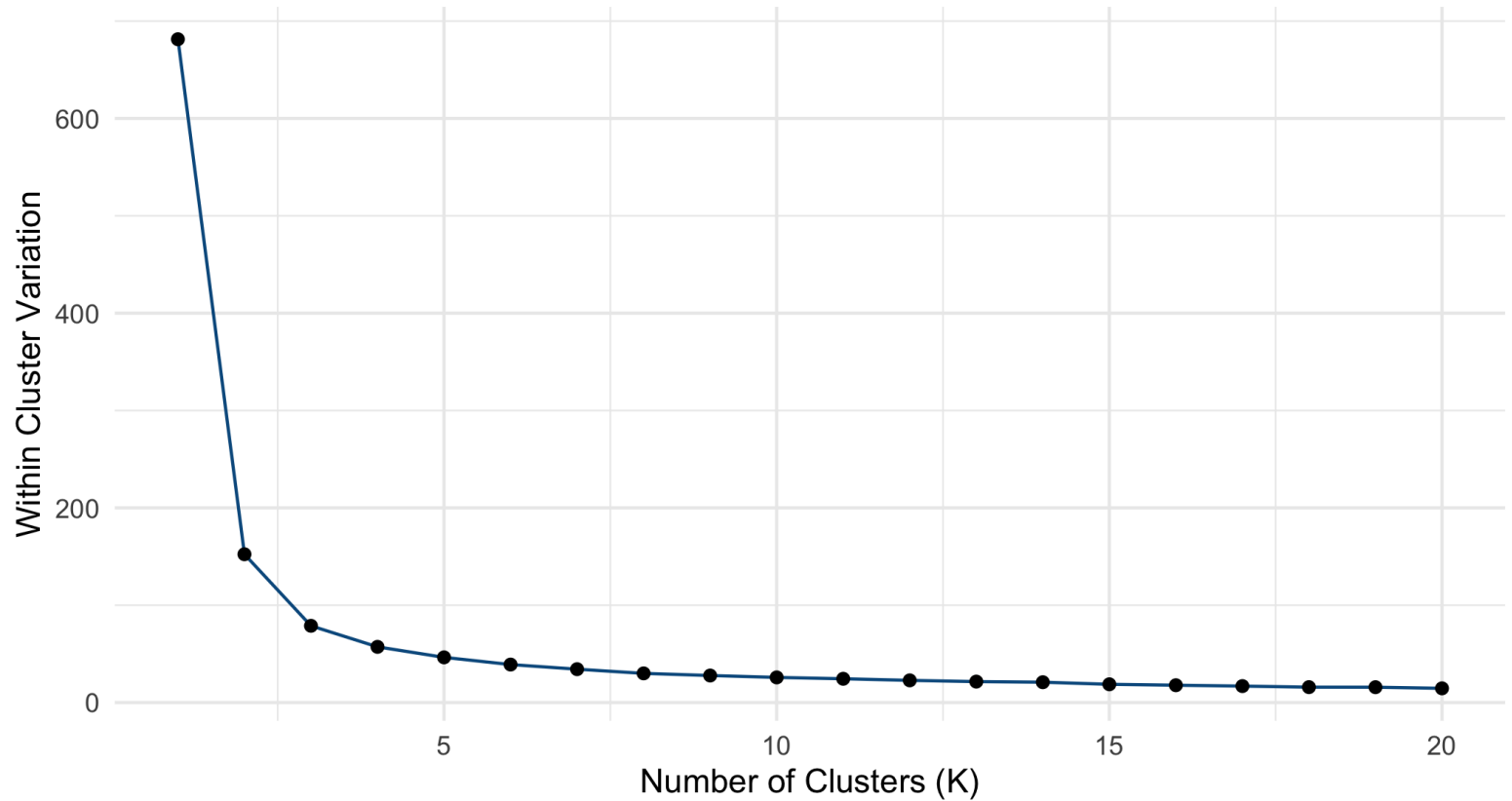


# Repeating *K*-means clustering

- Each solution depends on the *initial, random cluster assignment*
- This is called the **local optimum**
- Because of this, we should:
  1. Repeat *K*-means algorithm
  2. Select the iteration that minimizes *within cluster variation*



# Choosing $K$



# Pros and Cons

## Pros

- Scales to large data sets well
- Doesn't make any assumptions about data distribution
- Generalizes to clusters of different shapes and sizes

## Cons

- Subjective
- Interpretation requires domain knowledge
- Exploratory data analysis
- Difficult to assess results

# K-means clustering in R

```
library(cluster, warn.conflicts = F, quietly = T) #clustering algorithms
library(factoextra, warn.conflicts = F, quietly = T) #data visualization

head(iris) #data

dat <- iris[,1:4]
dat <- na.omit(dat) #what to do if NA values?
dat_scaled <- scale(dat) #why?

set.seed(6) #why?
kmeans_res <- kmeans(dat_scaled, centers = 3, nstart = 25) #centers? nstart?
str(kmeans_res)

fviz_cluster(kmeans_res, data = dat_scaled)
```

# Interpretation using domain knowledge

- What do the clusters represent?

```
fviz_cluster(kmeans_res, #set up plot
              data = dat_scaled,
              geom = "point", # only shows points and not labels
              shape = 19, # define one shape for all clusters (a circle)
              alpha = 0)+ # make circles see-through
  geom_point(aes(colour = as.factor(kmeans_res$cluster),
                 shape = iris$Species))+ #colour by species
  ggtitle("Comparing Clusters and Iris Species") #add a title
```

- Conclusion?

# Choosing $K$ in R

```
total_sum_squares <- function(k){ #perform kmeans & calculate ss
  kmeans(dat_scaled, centers = k, nstart = 25)$tot.withinss
}

all_ks <- seq(1,20,1) #define a sequence of values for k

choose_k <- sapply(seq_along(all_ks), function(i){ #apply to all values
  total_sum_squares(all_ks[i])
})

choose_k_plot <- data.frame(k = all_ks, # dataframe for plotting
                           within_cluster_variation = choose_k)

head(choose_k_plot)
ggplot(choose_k_plot, aes(x = k, # plot
                        y = within_cluster_variation))+
  geom_point()+
  geom_line()+
  xlab("Number of Clusters (K)")+
  ylab("Within Cluster Variation")
```

# Extra reading

- Chapter 10.3 of James *et al.*, [ISLR](#)
- STHDA [Factoextra R Package Guide](#)



# References

- James *et al.*, [ISLR](#)

## Slides

- xaringan, xaringantheme, remark.js, knitr, R Markdown