Report of Data Science Internship

At

Exposys Data Labs, Bengaluru

Submitted by

Pavitra Desai

ACKNOWLEDGMENT

Working in Exposys Data Labs for Customer Segmentation Project has been a great learning. I want to thank Y Vishnuvardhan, Founder and CEO of Exposys Data Labs, for providing me with this Virtual Internship opportunity. I would also like to thank my mentor Mr. Aravind, R&D Engineer, for providing me constant guidance throughout this project.

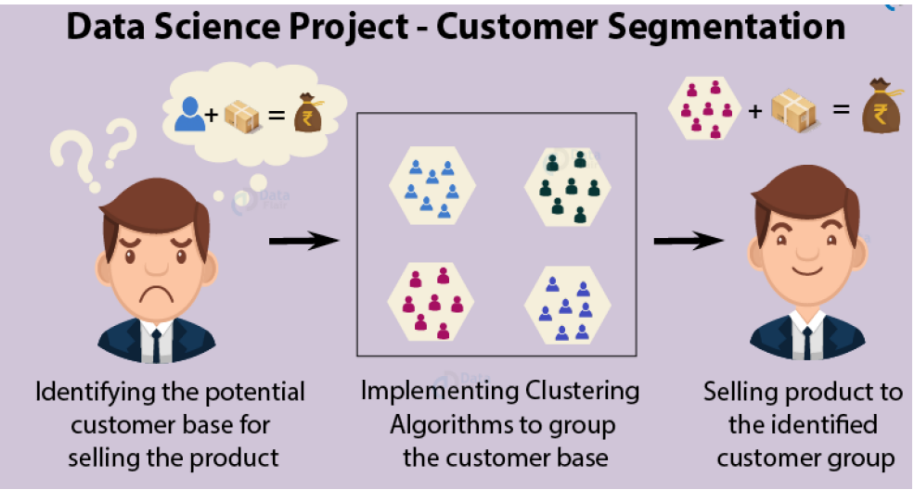
In these times of COVID-19, I am grateful to the company for providing me with work from home internship, which was in the form of a mini-project. It was a great start to my professional development. As my first internship, it was beneficial to boost my confidence level and interest in this particular field of Data Science. I strive to work hard and continue to improve towards my career objectives with all the skills and knowledge acquired.

ABSTRACT

We look around us and find that most companies are now data-driven. They make strategic decisions based on data analysis, enabling them to examine and organize their data for better service. There has always been a lot of competition in the market as to who can provide the best customer experience, attract new customers based on their needs, and satisfy their demands, enhancing their profit and growth. However, this is not very easy and calls for various data mining techniques and algorithms.

Machine learning can help them target potential customers. The algorithms deep dive into the data pool to extract hidden treasures and patterns that can bring wonderous profits to many organizations and better decision making. Customer segmentation is one such beautiful concept. Customer segmentation finds its use in many sectors. For example, in Netflix, it can be used as a recommendation system to find a group of similar users and use it for filtering, categorizing, or recommending movies. Banks or insurance companies use it for fraud detection or to evaluate certain insurance risks to segmented customers.

We will be using Customer Segmentation in the retail industry, a Mall, to segment customers into various groups and target potential. The industry can then work towards attractive ideas to sell products and services inclined towards these specific customers.



EXPOSYS DATA LABS

CHAPTER 1

INTRODUCTION

1.1. Customer segmentation

Customer segmentation is partitioning a customer database into group of people with similar characteristics. It is an application of unsupervised learning. It is a business strategy that allows targeting a specific group of customers and effectively allocate marketing resources. For such large datasets, we need an analytical approach like clustering to make customer segments.

There are four major ways of segmentation, i.e., geographical, economic, demographic, and behavioural patterns.

In this project, we divide a Mall customer's dataset based on gender, age, income, spending habits, etc. We also visualize gender and age distributions and analyse their annual incomes and spending scores to target the potential user base. The method used is K-means clustering in R Studio IDE.

Language: R

1.2. Supervised and Unsupervised Learning

Supervised learning is training a model with labelled data. There are two types regression and classification. Regression is the process of predicting a continuous value as opposed to predicting an absolute value in classification. In classification, the class is predefined and predict categorial classed labels. Classification approaches include decision trees, logistic regression to predict the default value of the new entry.

In unsupervised learning, the model discovers information on its own. There is no prior information on the data or the outcomes of the analysis. Dimension reduction, density estimation, market basket analysis, and clustering are the most widely used unsupervised machine learning techniques. Generally, clustering is used for exploratory data analysis, summarisation, dimension reduction, outlier detection, and other such data mining tasks.

In comparison to supervised learning, unsupervised learning has fewer models and fewer evaluation methods that can be used to ensure that the outcome of the model is accurate. As

such, unsupervised learning creates a less controllable environment as the machine is creating outcomes for us.

1.3. Clustering

Clustering can group data unsupervised solely based on similarities to each other. It will partition customers into mutually exclusive groups aka clusters. Having the result would help understand and predict customer preferences and differences, thus making the company deliver personalised experiences for each group of customers.

Types of Clustering:

1. Partition-based clustering is a group of clustering algorithms that produces sphere-like clusters, such as; K-Means, K-Medians or Fuzzy c-Means. These algorithms are relatively efficient and are used for medium to large-sized databases.
2. Hierarchical clustering algorithms produce trees of clusters, such as agglomerative and divisive algorithms. This group of algorithms are very intuitive and are generally suitable for use with small-size datasets.
3. Density-based clustering algorithms produce arbitrary-shaped clusters. They are outstanding when dealing with spatial clusters or noise in the data set, for example, the DB scan algorithm.

CHAPTER 2
TRAINING HIGHLIGHTS

2.1.Implementation

Customer Segmentation can be done by using primary functions like kmeans (), ggplot (), etc in R. I've used R Studio IDE for this project.

2.2.Code

The first start will be installing necessary packages and libraries.

Importing the necessary libraries

```
library(ggplot2)
library(cluster)
library(dplyr)
```

Downloading Data : the Mall_Customer.csv file from https://drive.google.com/file/d/19BOhwz52NUY3dg8XErVYglctpr5sjTy4/view and saving it to current working directory.

```
dt<-read.csv("Mall_Customers.csv")

#renaming column name for easy handling and viewing some data
data<-rename(dt,ID=CustomerID,Annualincome=Annual.Income..k..,Spendingscore = Spending.Score..1.100.)
dim(data)
```

```
## [1] 200   5
```

The table has 200 rows and 5 columns. Its first six rows are:

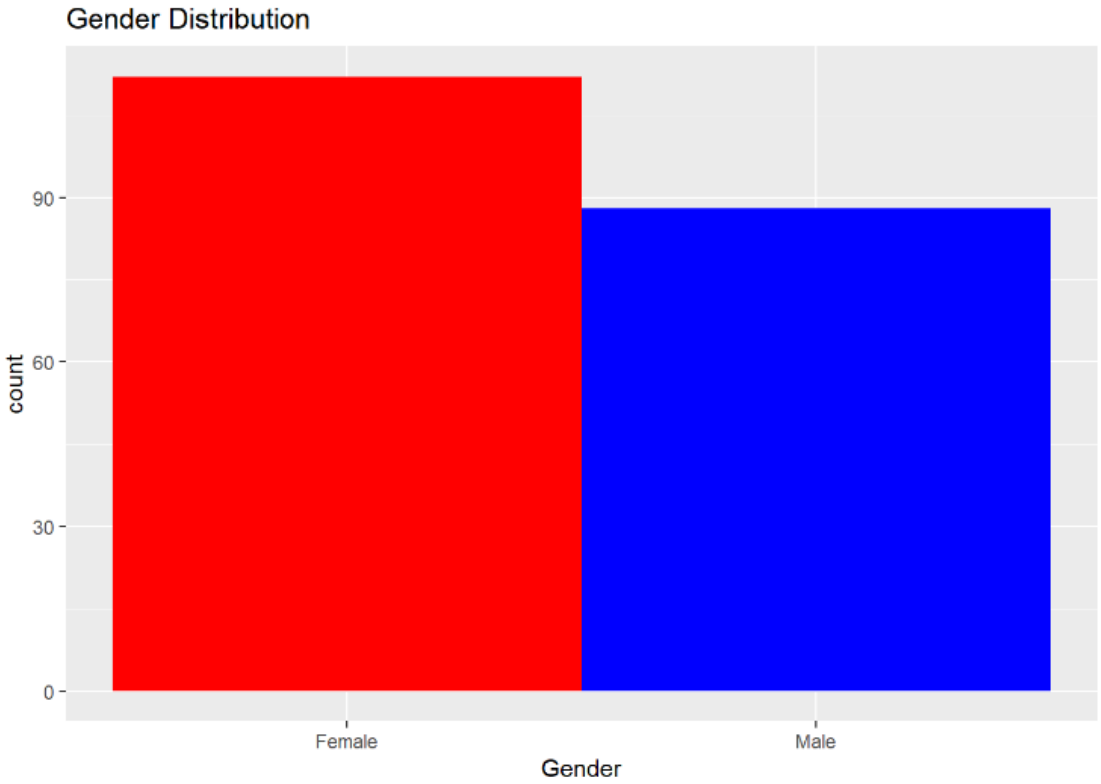
```
head(data)
```

```
##   ID Gender Age Annualincome Spendingscore
## 1  1   Male  19           15            39
## 2  2   Male  21           15            81
## 3  3 Female  20           16             6
## 4  4 Female  23           16            77
## 5  5 Female  31           17            40
## 6  6 Female  22           17            76
```

2.3. Visualizing the data set:

Visualizing Gender Distribution

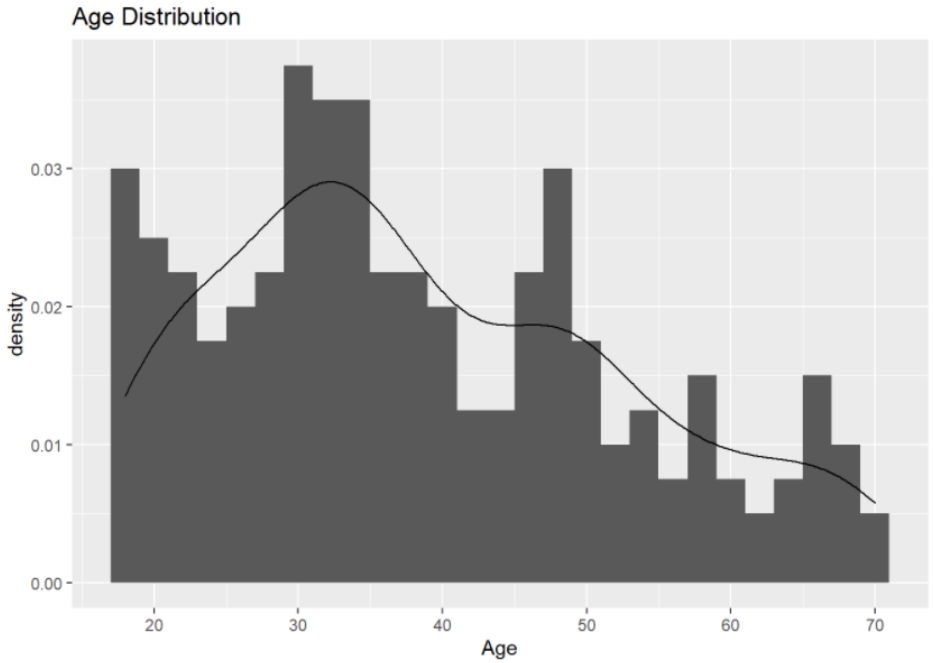
```
ggplot(data,aes(x=Gender))+
  geom_bar(stat="count",width = 1,fill=c("red","blue"))+
  labs(title = "Gender Distribution")
```



From the above graph it is observed that female customers are more than male .

Visualizing Age Distribution

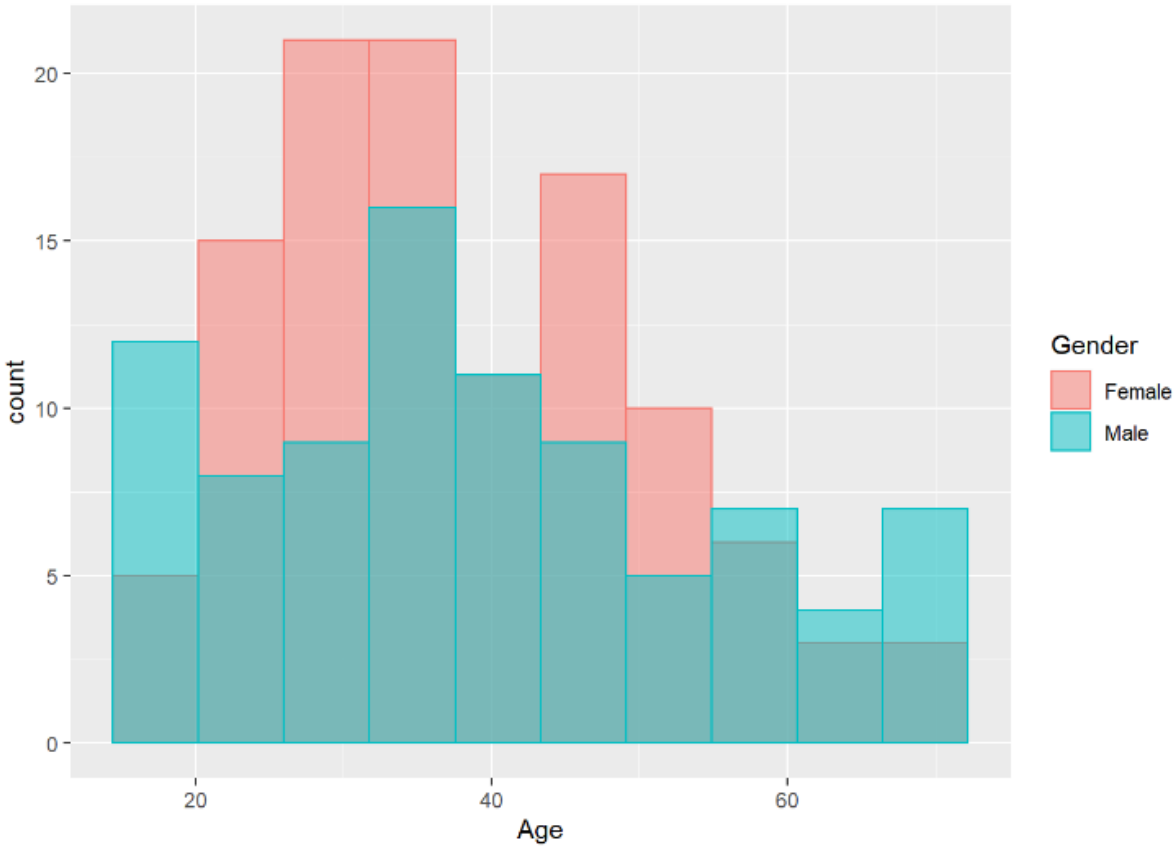
```
ggplot(data,aes(x=Age))+
  geom_histogram(binwidth = 2,aes(y= ..density..)) +
  geom_density(alpha = 0.5) +
  labs(title="Age Distribution")
```



-

Visualizing Age Distribution by Genders

```
ggplot(data,aes(x=Age,fill=Gender,color=Gender))+
  geom_histogram(bins=10,position = "identity",alpha=0.5)
```



2.4.K-means Clustering

K-means clustering is a way of partitioning a group of observations into a fixed number of clusters.

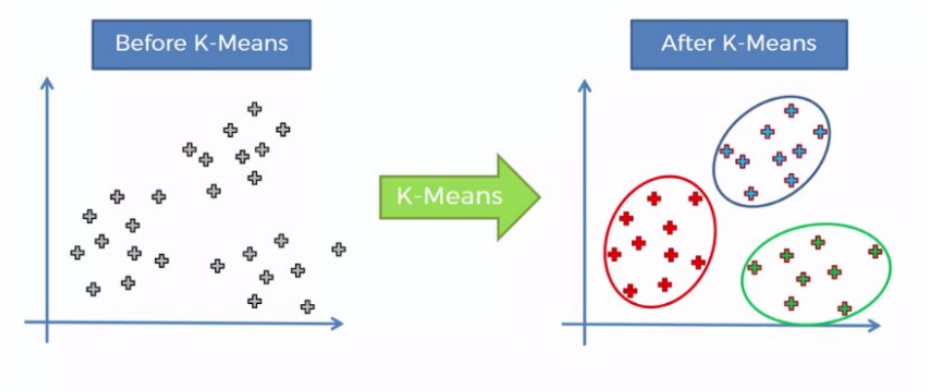


Photo by Google Images

K-Means Algorithm Steps:

1. Determine the number of clusters K

2. Select random K centroids for each cluster

3. Assign all data points to the nearest centroid depending on the minimum Euclidean distance.

4. Calculate new centroid for each cluster as the mean of all its data points

5. Reassign each data point to the new closest centroid

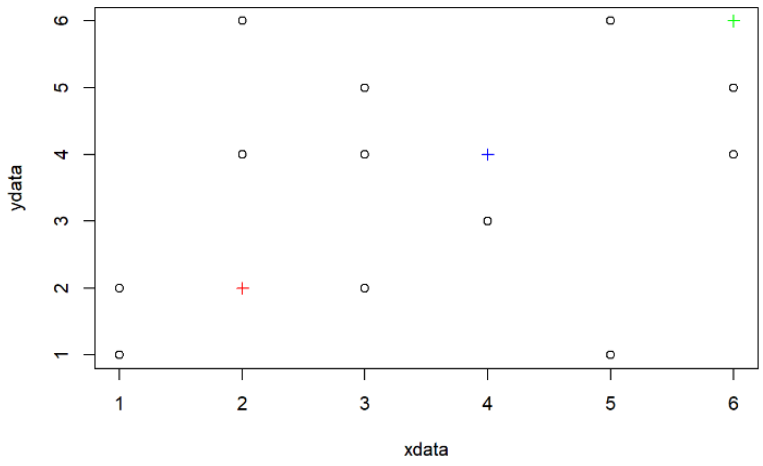
So, we iterate back and forth till all centroid are fixed at a constant location and no more changes in data assignment. The K-means clustering algorithm will then produce a final estimate of where the cluster centroids are and which centroid each observation is assigned.

As a demonstration, I have taken 12 data points and tried to divide them into 3 clusters. Below is the code and its output.

Plotting 12 random data points and 3 centroids

```
xdata<-c(1,3,5,3,4,6,5,1,2,3,2,6)
ydata<-c(1,2,6,4,3,5,1,2,6,5,4,4)
data<-rbind(xdata,ydata)
plot(xdata,ydata)

xc<-c(2,4,6)
yc<-xc
datac<-rbind(xc,yc)
points(xc,yc,col=c("red","blue","green"),pch=3)
```

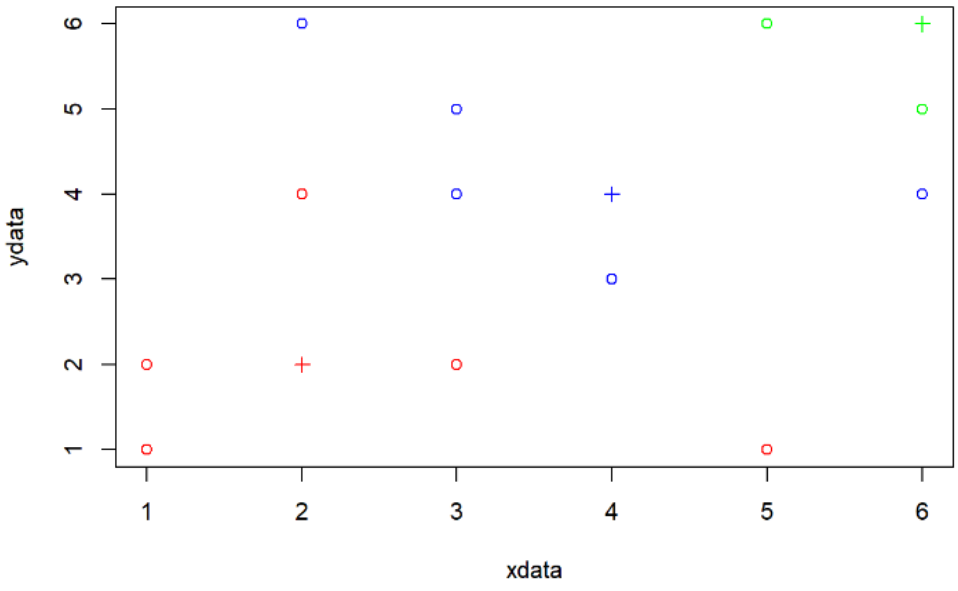


finding minimum distance of all points to centroids

```
mdist<-function(xdata,ydata,xc,yc){
  distTmp <- matrix(NA,nrow=3,ncol=12)
  distTmp[1,] <- (xdata-xc[1])^2 + (ydata-yc[1])^2
  distTmp[2,] <- (xdata-xc[2])^2 + (ydata-yc[2])^2
  distTmp[3,] <- (xdata-xc[3])^2 + (ydata-yc[3])^2
  return(distTmp)
}
distTmp<-mdist(xdata,ydata,xc,yc)
```

assigning data points to nearest centroids

```
clust<-apply(distTmp, 2,which.min)
colc<-c("red","blue","green")
plot(xdata,ydata,col=colc[clust])
points(xc,yc,col=c("red","blue","green"),pch=3)
```



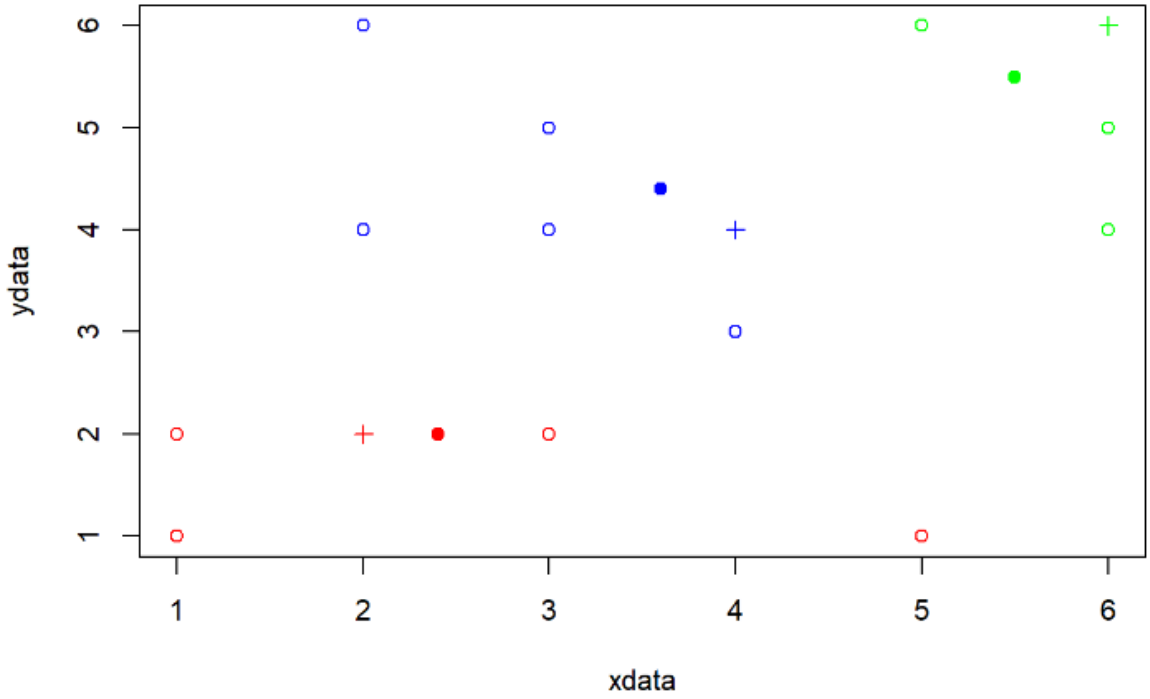
All datapoints have been asigned to their respective colored centroids.

recalculating centroids as mean of respective data points

```
newxc<-tapply(xdata,clust,mean)
newyc<-tapply(ydata,clust,mean)
plot(xdata,ydata,col=colc[clust])
points(xc,yc,col=c("red","blue","green"),pch=3)
points(newxc,newyc,col=colc,pch=19)

## reassigning data points

distTmp2<-mdist(xdata,ydata,newxc,newyc)
newclust<-apply(distTmp2,2,which.min)
points(xdata,ydata,col=colc[newclust])
```



As you can see the filled dots are now the new centroids and according to those two points have changed their clusters. All of this can be done by using kmean() function in R.

First, we need to find optimal value of k. below are the methods to do so.

2.4.1. Elbow Method

The elbow method is based on the observation that increasing the number of clusters can help to reduce the sum of the within-cluster variance of each cluster. In this method WSS (within-cluster sum of squared errors) vs. K is plot, then an optimal value to k is chosen where WSS first start to diminish. It can be formulated as below

$$\text{minimize}\left(\sum_{k}^{k=1} W(Ck)\right)$$

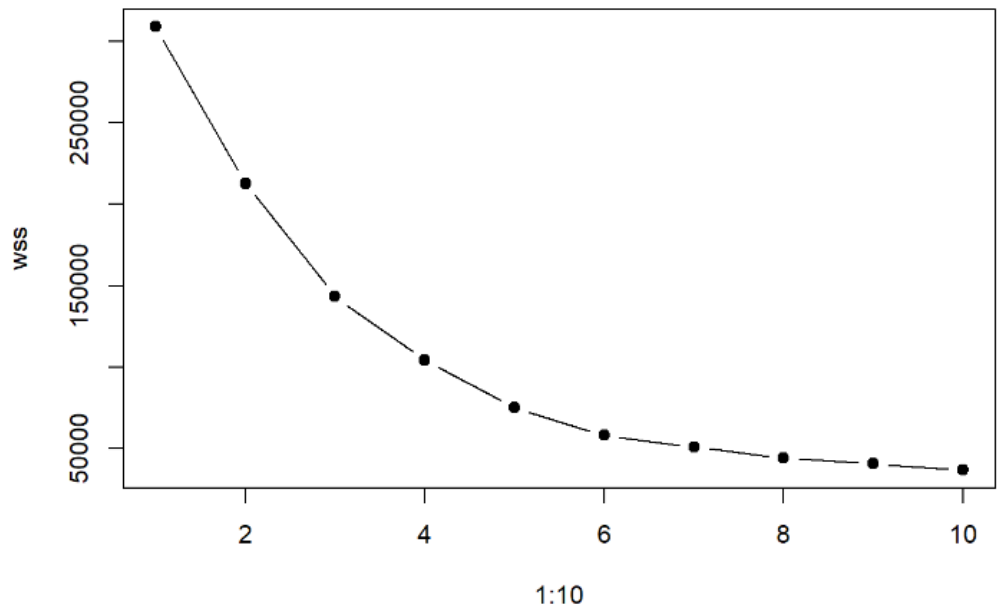
where Ck is the kth cluster and W(Ck) is the within-cluster variation.

Thus, we can use the following algorithm to define the optimal clusters:

1. Compute clustering algorithm (e.g., k-means clustering) for different values of *k*. (1:10)
2. For each *k*, calculate the total within-cluster sum of square (wss)
3. Plot the curve of wss according to the number of clusters *k*.
4. The location of a bend(elbow) in the plot is considered as an optimal value.

```
#elbow method

set.seed(100)
wss <- sapply(1:10,function(k){kmeans(data[,3:5], k, nstart=50,iter.max = 15 )$tot.withinss})
plot(1:10, wss,
     type="b", pch = 19
)
```



Here value of k is not quite clear therefore we should go for another method.

2.4.2. Gap Statistic Method

The gap statistic compares the total intraclustural variation for different values of *k* with their expected values under null reference distribution of the data (i.e. a distribution with no obvious clustering).

$$\text{Gapn(k)} = \text{En}^{*}\log(\text{Wk}) - \log(\text{Wk})$$

In short, the algorithm involves the following steps:

1. Cluster the observed data, varying the number of clusters from k=1…, kmax and compute the corresponding Wk.
2. Generate B reference data sets and cluster each of them with varying number of clusters. Compute the estimated gap statistics presented in eq. above.
3. Let w=(1/B) $\sum_b$log (W^{*} kb), compute the standard deviation

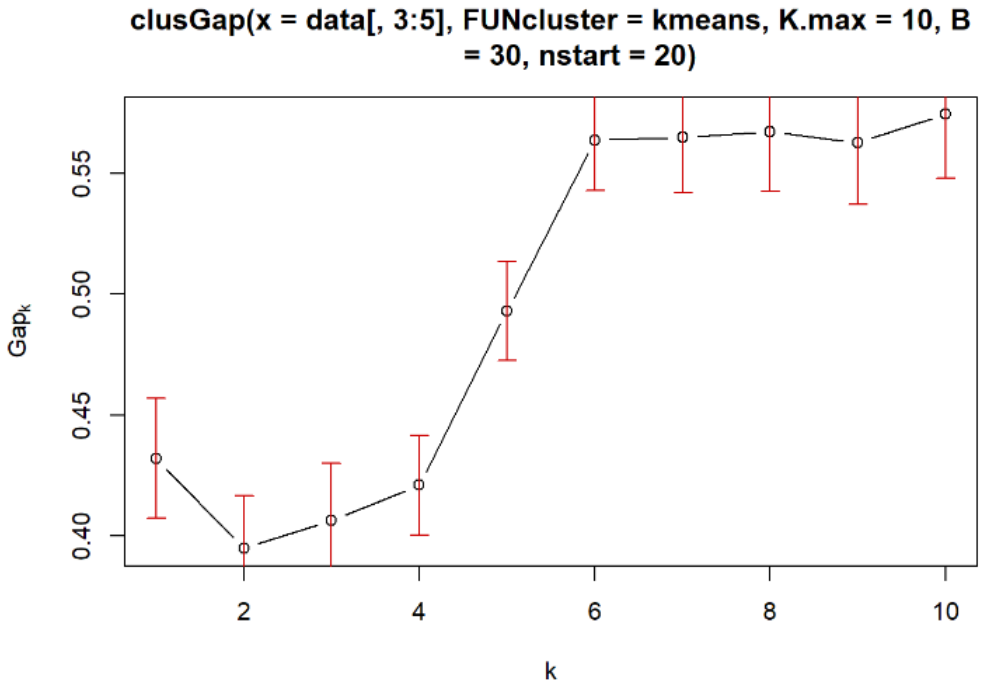
   $$\text{Sd(k)}= \sqrt{(\tfrac{1}{b})\sum_b(\log(W^{*}_{kb}) - w)^2}$$ and define S_k=Sd_k× $\sqrt{1 + 1/B}$

4. Choose the number of clusters as the smallest k such that Gap(k)≥Gap(k+1) −sk+1.

   To compute the gap statistic method, we can use the clusGap function which provides the gap statistic and standard error for an output.

Calculating optimal value of k using gap statistic method

```
gap<-clusGap(data[,3:5],FUNcluster = kmeans,nstart=20,K.max = 10,B=30)
plot(gap)
```

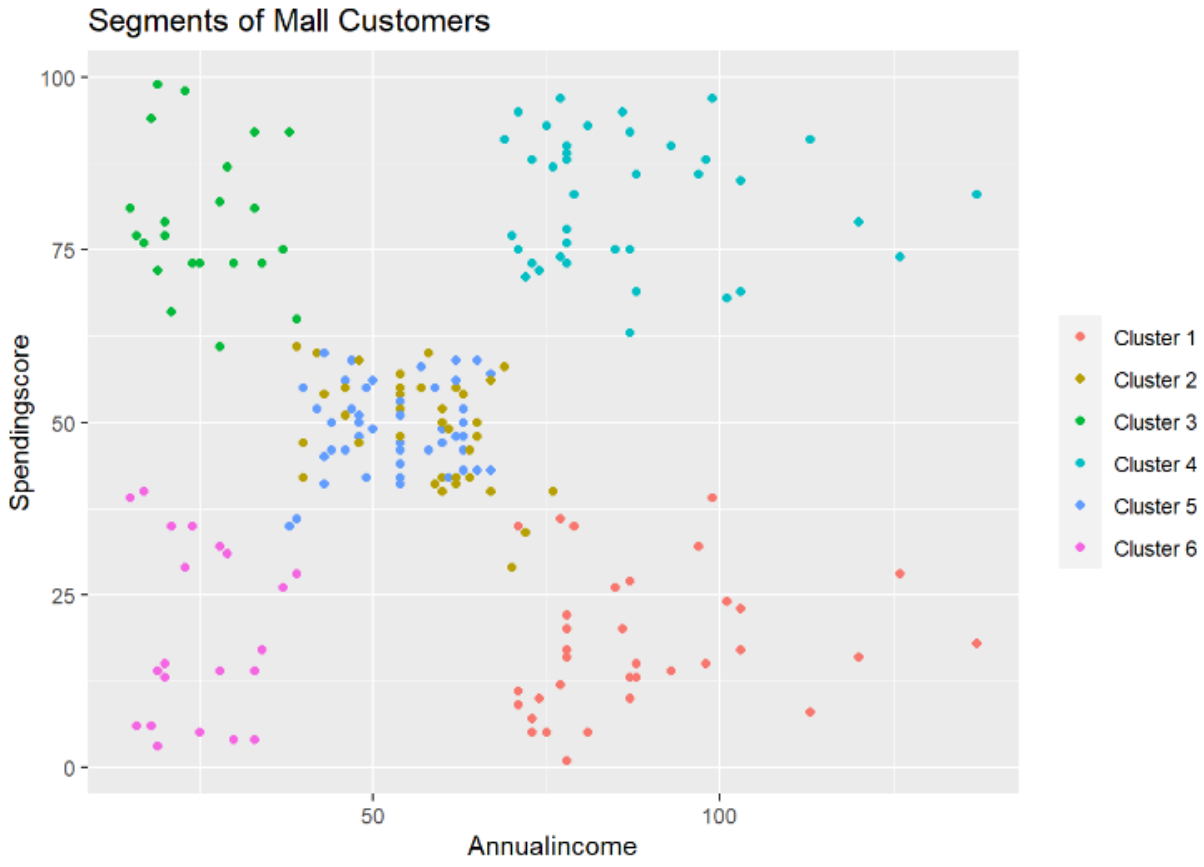


Verified k as 6

Analyzing annual incomes and spending scores using K-means

```
k6<-kmeans(data[,3:5],6,iter.max = 100,nstart = 50,algorithm = "Lloyd")
```

```
set.seed(1)
ggplot(data, aes(x = Annualincome , y = Spendingscore)) +
  geom_point(stat = "identity", aes(color = as.factor(k6$cluster))) +
  scale_color_discrete(name = " ",
                       breaks=c("1", "2", "3", "4", "5","6"),
                       labels=c("Cluster 1", "Cluster 2", "Cluster 3",
                                "Cluster 4", "Cluster 5","Cluster 6")) +
  ggtitle("Segments of Mall Customers")
```



CHAPTER 3
SUMMARY AND CONCLUSION

3.1.Result

The goal of K means is to group data points into distinct non-overlapping subgroups.

Cluster 4: high spending scores and high-income; alert them with new arrivals as they are potential customer for increase in revenue.

```
##      ID Gender Age Annualincome Spendingscore
## 124 124   Male  39           69            91
## 126 126 Female  31           70            77
## 128 128   Male  40           71            95
## 130 130   Male  38           71            75
## 132 132   Male  39           71            75
## 134 134 Female  31           72            71
```

Cluster 1: high income and low spending score; ask them for feedback and advertise them with new products that might attract them, they have the potential to convert into cluster 4.

```
##      ID Gender Age Annualincome Spendingscore
## 127 127   Male  43           71            35
## 129 129   Male  59           71            11
## 131 131   Male  47           71             9
## 135 135   Male  20           73             5
## 137 137 Female  44           73             7
## 139 139   Male  19           74            10
```

Cluster 3: low income and high spending scores; can help them by providing new deals and sales offers so that despite low income they still remain loyal.

```
##    ID Gender Age Annualincome Spendingscore
## 2   2   Male  21           15            81
## 4   4 Female  23           16            77
## 6   6 Female  22           17            76
## 8   8 Female  23           18            94
## 10 10 Female  30           19            72
## 12 12 Female  35           19            99
```

Cluster 6: low income and low spending score; it won't be beneficial to both the parties to target these customers.

```
##    ID Gender Age Annualincome Spendingscore
## 1   1   Male  19           15            39
## 3   3 Female  20           16             6
## 5   5 Female  31           17            40
## 7   7 Female  35           18             6
## 9   9   Male  64           19             3
## 11 11   Male  67           19            14
```

Rest are average and the company can use them according to market conditions.

3.2. Advantages

- Determine appropriate product pricing.
- Develop customized marketing campaigns.
- Design an optimal distribution strategy.
- Choose specific product features for deployment.
- Prioritize new product development efforts.

References

1. https://uc-r.github.io/kmeans_clustering
2. https://www.coursera.org/learn/machine-learning-with-python/lecture/Nlxjw/intro-to-clustering
3. https://www.coursera.org/learn/machine-learning-with-python/lecture/jgzpX/supervised-vs-unsupervised
4. https://medium.datadriveninvestor.com/k-means-clustering-4a700d4a4720