```
#Calling all the Libraries which will be in use in this project

library(readxl)
library(dplyr)
library(klaR)
library(psych)
library(fpc)
library(ggcorrplot)
library(factoextra)
library(cluster)

#Importing the data
data= read_excel("1671447588_marketing_campaign.xlsx")

#Checking and correcting the data type of variable date (dt_Customer)
class(Dt_Customer)
Dt_Customer = as.Date(Dt_Customer)
class(Dt_Customer)

colnames(data)

#Finding the variables with missing values
colSums(is.na(data))

#As we can see only one variable have missing value i.e "Income" that too is 1% of the data, we will consider dropping all the rows with missing values
```

```
data= na.omit(data)

colSums(is.na(data))


#Checking the data

str(data)

dim(data)

attach(data)


#Checking newest and oldest customer's enrolment date in the records

Date_sorted= sort(Dt_Customer)

min(Date_sorted)

max(Date_sorted)




#Creating a Column  "Customer_For" of the number of days the customers started to shop in the store
relative to the last recorded date


data$Customer_For= round(difftime("2022-12-21", Dt_Customer, units = "days")- data$Recency)


#creating a column "Age" to show the Age of customers from "Year_Birth

data$Age= 2022- Year_Birth


#Creating a Column "Spent" indicating the total amount spent by the customer in various categories
over two years
```

```
data$Spent=
MntWines+MntFruits+MntMeatProducts+MntFishProducts+MntSweetProducts+MntGoldProds


#Creating a column "Living_With" out of "Marital_Status" to extract the living situation of couples.

data$Living_With= data$Marital_Status


data$Living_With[data$Living_With== "Married" | data$Living_With== "Together"] = "Partner"

data$Living_With[data$Living_With != "Partner"] = "Alone"

table(data$Living_With)



#Creating a column "Children" to indicate the total number of children in a household

data= data |> mutate(Children= data$Kidhome+data$Teenhome)

colnames(data)


#Creating column "Family_size" indicating total no. of persons in household

No_of_adult= ifelse(data$Living_With == "Partner", 2 , 1)

data$Family_size= data$Children+No_of_adult


#Creating column "Is_Parent" to indicate the parenthood status

data$Is_parent= ifelse(data$Children>0, 1 , 0)

table(data$Is_parent)


#converting Education 5 levels into 2 levels namely "Graduate" and "UnderGraduate"

data$Education[data$Education == "2n Cycle" | data$Education == "Basic"] = "Undergraduate"

data$Education[data$Education != "Undergraduate"] = "Graduate"
```

```
#For clarity, change the name of the few variables
colnames(data)
colnames(data)[10]= "Wines"
colnames(data)[11]= "Fruits"
colnames(data)[12]= "MeatProducts"
colnames(data)[13]= "FishProducts"
colnames(data)[14]= "SweetsProducts"
colnames(data)[15]= "GoldProds"
colnames(data)


#Dropping the redundant columns
data= subset(data, select = -c(Marital_Status, Dt_Customer,Z_CostContact, Z_Revenue, Year_Birth,ID))
colnames(data)

#Creating box plots and histograms for age and income to identify the outliers.
par(mfrow = c(1,2))
hist(data$Age, xlab = "Age", ylab = "Frequency", main = "Distribution of Age")
boxplot(data$Age)

#From the boxplot we can see that above the age of 100 are outliers, lets drop them
data= data |> filter(data$Age<100)

par(mfrow = c(1,2))
```

```
hist(data$Income, xlab = "Income", ylab = "Frequency", main = "Distribution of Income")

boxplot(data$Income)


#lets check the outlier and drop the rows with outliers


quantile(data$Income)

iqr = IQR(data$Income)

Up = quantile(data$Income, .75)+1.5*iqr


data= data  |> filter(data$Income<Up)




#Lets check out the correlation between numeric variables.

data_numeric= select_if(data, is.numeric)

cor(data_numeric)


#lets create heatmap to understand correlatrion better

ggcorrplot(cor(data_numeric))


#Changing the data type for clustering

str(data)


class(data$Customer_For)

data$Customer_For= as.numeric(data$Customer_For)
```

```
#chnaging chrachater variable to numeric for clustering
table(data$Education)
data$Education[data$Education == "Graduate"]= 0
data$Education[data$Education == "Undergraduate"]= 1
table(data$Education)

table(data$Living_With)
data$Living_With[data$Living_With == "Alone"]= 0
data$Living_With[data$Living_With == "Partner"]= 1
table(data$Living_With)


data$Education= as.numeric(data$Education)
data$Living_With= as.numeric(data$Living_With)
table(data$Is_parent)


#Creating dummy variable for factor variables
dummyEducation = as.data.frame(dummy.code(data$Education))
dummyLiving_with = as.data.frame(dummy.code(data$Living_With))
dummyIs_parent = as.data.frame(dummy.code(data$Is_parent))

names(dummyEducation)= c("Graduate", "undergraduate")
names(dummyLiving_with) = c("Alone", "Partner")
names(dummyIs_parent) = c("No", "Yes")
```

```
dummy = data.frame(dummyEducation, dummyIs_parent, dummyLiving_with)


#merging the dummy data frame and orginal data frame
final= data.frame(data, dummy)
colnames(final)


#Scaling the data for clustering
final= scale(final)


fviz_nbclust(final, kmeans, method = "wss")


#Using elbow method from graph we can say we should use 3 clusters for Kmeans
#Performing kmeans
km <- kmeans(final, centers = 3, nstart = 25)
km



#Plotting clusters
gap_stat <- clusGap(final,
                    FUNcluster = kmeans,
                    nstart = 25,
                    K.max = 10,
                    B = 50)
```

```
fviz_cluster(km, data = data)


#cluster profiling

final_data = cbind(data, cluster = km$cluster)

head(final_data)


#Lets look at how clusters are divided in factor variables

table(final_data$Education, final_data$cluster)

table(final_data$Living_With, final_data$cluster)

table(final_data$Is_parent, final_data$cluster)


#Lets see how cluster is divided through visualizations


barplot(table(final_data$Education, final_data$cluster), xlab = "Clusters", ylab = "Customers", main = "Education wise - cluster divided")

barplot(table(final_data$Living_With, final_data$cluster), xlab = "Clusters", ylab = "Customers", main = "Partner wise - cluster divided")

barplot(table(final_data$Is_parent, final_data$cluster), xlab = "Clusters", ylab = "Customers", main = "parent wise - cluster divided")


#Export the data

write.csv(final_data, Final.csv, row.names = FALSE)


#Thank you

print("Thank you")
```