```
#set working directory
setwd('D:/Work/SA Training/Clustering')


#load data
fullData <- read.csv("OxaneData2.csv")


#Data exploration and correction
str(fullData)
summary(fullData)

fullData$Age <- 2017-floor(fullData$Date.of.Birth/10000)

#creating dummy variables
fullData$Fixed <- ifelse(fullData$Fixed.or.Variable=='F',1,0)
fullData$EverInArrears. <- ifelse(fullData$Ever.in.Arrears.=='Y',1,0)
fullData$LifeInsPolicy. <- ifelse(fullData$Life.Insurance.Policy.=='Y',1,0)
fullData$sex_male <- ifelse(fullData$Sex=='M',1,0)

#check method of payment, acquisition channel no variation, can be ignored
#Missing value treatment
fullData$CRIF.Indicator.Repayment.Capacity <- NULL
fullData2 <- na.omit(fullData)
summary(fullData2)

#more dummy variables
fullData[fullData$Accomodation.Type=='A',]
fullData2$Accomodation.Type <- trimws(fullData2$Accomodation.Type)

fullData2$AccoType_A <- ifelse(fullData2$Accomodation.Type=='A',1,0)
fullData2$AccoType_C <- ifelse(fullData2$Accomodation.Type=='C',1,0)
fullData2$AccoType_G <- ifelse(fullData2$Accomodation.Type=='G',1,0)
fullData2$AccoType_P <- ifelse(fullData2$Accomodation.Type=='P',1,0)
fullData2$AccoType_T <- ifelse(fullData2$Accomodation.Type=='T',1,0)
fullData2$AccoType_U <- ifelse(fullData2$Accomodation.Type=='U',1,0)
fullData2$AccoType_X <- ifelse(fullData2$Accomodation.Type=='X',1,0)

#Prepare the data for standardization
LoanId <- fullData2$LoanId
ncol(fullData2)
summary(fullData2)

DummyVar <- fullData2[,c(29:38)]

#Taking only selected variables
newData <- fullData2[,c(2:7,9:10,12:15,22:27)]

#standardization (method1)
newData2 <- scale(newData)

# method2
fn_norm <- function(var){
  newData[,var] <- (newData[,var] - min(newData[,var]))/(max(newData[,var])
                                                        -min(newData[,var]))
  return(newData[,var])
}


#Creating training data and test data
FinData <- cbind(LoanId,newData2,DummyVar)

set.seed(3)
```

```
test =  sample(1:nrow(FinData),floor(nrow(FinData)/10))
train = -test
training_data = FinData[train,]
testing_data = FinData[test,]

#Remove loan id from the training/testing data
LoanIdTrain <- training_data$LoanId
LoanIdTest <- testing_data$LoanId
training_data$LoanId <- NULL
testing_data$LoanId <- NULL

# Estimating the number of clusters (method1) - This should be used when you have powerful systems
install.packages('NbClust')
library(NbClust)
nb <- NbClust(testing_data, distance = "euclidean", min.nc = 2,
              max.nc = 4, method = "complete", index ="all")

# Estimating the number of clusters (method2)
k = 2:10
set.seed(42)
WSS = sapply(k, function(k) {kmeans(training_data, k)$tot.withinss})
plot(k, WSS, type="l", xlab= "Number of k", ylab="Within sum of squares")

kmeans(training_data, 6)$tot.withinss


###Run the kmeans algorithm to generate the clusters
k1<-kmeans(training_data, 3, iter.max = 200, nstart = 20)

###See the clustering results
k1

###Fetch the group means for each variable
k1$centers

###Fetch size/n of obs for the groups
k1$size

###Fetch the cluster for each obs
training_data$Segment <- k1$cluster

#Add LoanId back to the dataset
cbind(LoanIdTrain, training_data)
training_data$LoanId <- LoanIdTrain

#Find out the cluster no. for each loan id test dataset
#Hint: use dist() function to find out the euclidean distance between two vectors. Use cluster
centers (centroids) to find out the nearest cluster.

#example of using dist function
a <- c (23,34,43)
b <- c(21,63,22)
c<-data.frame(rbind(a,b))
dist(c)

k1$centers
```