```
install.packages("openxlsx")
library(openxlsx)
install.packages("caTools")
library(caTools)
install.packages("cluster")
library(cluster)

raw_data <- read.xlsx("C:\\Users\\fatma\\Desktop\\IE425_HW3\\EastWestAirlines.xlsx", sheet = 3)
raw_data
sum(is.na(raw_data))
str(raw_data)

#Some Preprocessing In The Data
#Convert the categorical attribute to the numerical value.
raw_data$cc1_miles <- ifelse(raw_data$cc1_miles == 1, 2500,
                        ifelse(raw_data$cc1_miles == 2, 7500,
                               ifelse(raw_data$cc1_miles == 3, 17500,
                                      ifelse(raw_data$cc1_miles == 4, 37500,
                                             ifelse(raw_data$cc1_miles == 5, 50000, raw_data$cc1_miles)))))
raw_data$cc2_miles <- ifelse(raw_data$cc2_miles == 1, 2500,
                        ifelse(raw_data$cc2_miles == 2, 7500,
                               ifelse(raw_data$cc2_miles == 3, 17500,
                                      ifelse(raw_data$cc2_miles == 4, 37500,
                                             ifelse(raw_data$cc2_miles == 5, 50000, raw_data$cc2_miles)))))
raw_data$cc3_miles <- ifelse(raw_data$cc3_miles == 1, 2500,
                        ifelse(raw_data$cc3_miles == 2, 7500,
                               ifelse(raw_data$cc3_miles == 3, 17500,
                                      ifelse(raw_data$cc3_miles == 4, 37500,
                                             ifelse(raw_data$cc3_miles == 5, 50000, raw_data$cc3_miles)))))


str(raw_data)
#Scaling
train_scaled=scale(raw_data)

#1. Hierarchical Clustering
hc_complete=hclust(dist(train_scaled), method="complete")
#Plot the tree.
plot(hc_complete,main="Complete Linkage", xlab="", cex=.7)

#Finding the Best NUmber of Clusters With Silhouette Index
silh=c()
for (k in 2:10){
  kume=cutree(hc_complete,k=k)
  X_sil=silhouette(kume, dist(train_scaled))
  silh[k-1]=mean(X_sil[,3])
}
data.frame(k=2:10,silh)
plot(silh, main = "Silhouette Plot", type="l")
#Choose the number with maximum Silhouette value.
best_k <- which.max(silh)+1
cat("The best number of clusters is: " , best_k)
#Cut the tree from the index of the maximum Silhouette value.
clusters <- cutree(hc_complete, k = best_k)
#See the distribution of classes.
table(clusters)


#b. Finding the centroids of the clusters.
clusters <- cutree(hc_complete, k = best_k)
centroids <- aggregate(train_scaled, by = list(clusters), FUN = mean)
centroids


#C. Remove Data Randomly
set.seed(425)
index_to_remove <- sample(1:nrow(raw_data),200)
train_removed <- raw_data[-index_to_remove,]
nrow(train_removed)

#Scaling
train_scaled <- scale(train_removed)
#Hierarchical Clustering
hc_complete=hclust(dist(train_scaled), method="complete")
plot(hc_complete,main="Complete Linkage", xlab="", cex=.7)

#Finding the Best NUmber of Clusters With Silhouette Index
silh=c()
for (k in 2:10){
  kume=cutree(hc_complete,k=k)
  X_sil=silhouette(kume, dist(train_scaled))
  silh[k-1]=mean(X_sil[,3])
}
data.frame(k=2:10,silh)
plot(silh, main = "Silhouette Plot", type="l")
#Find the Best Number of Clusters.
best_k <- which.max(silh)+1
cat("The best number of clusters is: " , best_k)
clusters <- cutree(hc_complete, k = best_k)
table(clusters)
#Find the center of the centroids.
clusters <- cutree(hc_complete, k = best_k)
centroids <- aggregate(data_removed, by = list(clusters), FUN = mean)
centroids


#d. KMeans
km <- kmeans(x=train_scaled, 2, nstart=20)
km
km$cluster
km$tot.withinss
km$size
plot(train_scaled, col=(km$cluster), main="K-Means Clustering Results with K=2", xlab="", ylab="", pch=2, cex=1)
#Find The Centroids of The Classes
centroids <- aggregate(train_scaled, by = list(km$cluster), FUN = mean)
centroids
```