# load library

library(ggplot2)

library(gridExtra)

library(readr)

library(ISLR)

library(fpc)

library(dplyr)

library(faraway)

library(cluster)

library(factoextra)

library(tidyverse)

library(NbClust)

library(mvoutlier)

# import data

data <- read\_csv("customers.csv")

data <- data.frame(data)

data\_0 <- data colSums(is.na(data)) ## check existence of no value

sum(duplicated(data)) ## check duplicate observations

# detect possible outliers

outlier = mvoutlier.CoDa(data)$outliers

sum(outlier) ## 45 outliers detected

# check distribution of features

data %>%psych::describe()

hist(data$Fresh)

hist(data$Milk)

hist(data$Grocery)

hist(data$Frozen)

hist(data$Detergents\_Paper)

hist(data$Delicassen)

# transform dataset

data = log(data)

# explore dataset

dim(data)

summary(data)

apply(data,2,var) ## variances of variables are not vastly different from each other apply(data,2,mean) ## mean of variables are not vastly different from each other

### use K-mean clustering with Euclidean distance

## unscaled data

# visualize data with different k

set.seed(4) # choose the best number of clusters

fviz\_nbclust(data, kmeans, method = "wss") + labs(subtitle = "Elbow method") ## using total within sum of squares

fviz\_nbclust(data, kmeans, method = "silhouette") + labs(subtitle = "Silhouette method") ## using average silhouette

set.seed(4)

## using prediction strength

ps\_data <- prediction.strength(data,Gmin=2,Gmax=10, M=100,clustermethod=kmeansCBI)

ps\_data

# k=2 is the best value

# check stability of clusters

boot\_kmean\_2 <-clusterboot(data=data,bscompare=T, multipleboot=F,bootmethod="boot", B=100, clustermethod=kmeansCBI, count=T, showplots=T, krange=2, seed = 4)

boot\_kmean\_2

set.seed(4)

k\_means\_2=kmeans(data,2) ## clustering data with k=2

# visualize data for k=2

fviz\_cluster(k\_means\_2, geom = "point", data = data) + ggtitle("k = 2")

clusplot(data, k\_means\_2$cluster, main='2D representation of the Cluster solution', color=TRUE, shade=TRUE, labels=2, lines=0)

# analyze original data

data\_index <- k\_means\_2$cluster %>% as.factor() ## get labels of observations

data\_kmean <- data\_0 %>%mutate(total\_spend=rowSums(data\_0))%>% mutate(cluster=k\_means\_2$cluster)

table(data\_index)

group\_summary = data\_kmean %>% group\_by(cluster)%>% summarise\_all(funs(round(mean(.))))

group\_summary

## scaled data

data.sc <- scale(data) %>% data.frame()

# choose the best number of clusters

set.seed(4)

fviz\_nbclust(data.sc, kmeans, method = "wss") + labs(subtitle = "Elbow method") ## using total within sum of squares

fviz\_nbclust(data.sc, kmeans, method = "silhouette") + labs(subtitle = "Silhouette method") ## using average silhouette

set.seed(4)

# using prediction strength

ps\_data\_sc <- prediction.strength(data.sc,Gmin=2,Gmax=10, M=100,clustermethod=kmeansCBI) ps\_data\_sc

# k=2 is the best value

## check stability of clusters

boot\_kmean\_2\_sc <-clusterboot(data=data.sc,bscompare=T, multipleboot=F,bootmethod="boot", B=100, clustermethod=kmeansCBI, count=T, showplots=T, krange=2, seed = 4)

set.seed(4)

k\_means\_2\_sc <- kmeans(data.sc,2) ## clustering with k=2

kmean\_index\_sc <- k\_means\_2\_sc$cluster %>% as.factor() ## get labels of observation

# visualize data for k=2

fviz\_cluster(k\_means\_2\_sc, geom = "point", data = data) + ggtitle("k = 2")

clusplot(data.sc, k\_means\_2\_sc$cluster, main='2D representation of the Cluster solution', color=TRUE, shade=TRUE, labels=2, lines=0)

# check the matching of clusters between scaled and unscaled data

data.sc\_index <- k\_means\_2\_sc$cluster %>% as.factor()

matching <- mean(data\_index==data.sc\_index)

matching

which(data\_index!=data.sc\_index) ## unmatched observations

# analyze original data

data\_sc\_kmean <- data\_0 %>%mutate(total\_spend=rowSums(data\_0))%>% mutate(cluster=k\_means\_2\_sc$cluster)

table(kmean\_index\_sc)

group\_summary\_sc = data\_sc\_kmean %>% group\_by(cluster)%>% summarise\_all(funs(round(mean(.))))

group\_summary\_sc

# using the principle component for k-means

data.pc <-prcomp(data.sc, scale. = T)

biplot(data.pc,scale=0)

pr.var=data.pc$sdev^2

pve=pr.var/sum(pr.var)

pve

cor(data.sc) ## correlation matrix