Assignment 5

Lokesh Arora, Ankita Shinde

https://github.com/ankita1598/Walmart

10/8/2020

library(lubridate)

##   
## Attaching package: 'lubridate'

## The following objects are masked from 'package:base':  
##   
## date, intersect, setdiff, union

library("plyr")  
library("ggplot2")  
library(RColorBrewer)  
library("dplyr")

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:plyr':  
##   
## arrange, count, desc, failwith, id, mutate, rename, summarise,  
## summarize

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

library("geosphere")

dataset = read.csv("data.csv", header= T)  
head(dataset)

## Store Dept Date weeklySales isHoliday Type Size Temperature  
## 1 1 1 2010-02-05 24924.50 False A 151315 42.31  
## 2 1 1 2010-02-12 46039.49 True A 151315 38.51  
## 3 1 1 2010-02-19 41595.55 False A 151315 39.93  
## 4 1 1 2010-02-26 19403.54 False A 151315 46.63  
## 5 1 1 2010-03-05 21827.90 False A 151315 46.50  
## 6 1 1 2010-03-12 21043.39 False A 151315 57.79  
## Fuel\_Price MarkDown1 MarkDown2 MarkDown3 MarkDown4 MarkDown5 CPI  
## 1 2.572 NA NA NA NA NA 211.0964  
## 2 2.548 NA NA NA NA NA 211.2422  
## 3 2.514 NA NA NA NA NA 211.2891  
## 4 2.561 NA NA NA NA NA 211.3196  
## 5 2.625 NA NA NA NA NA 211.3501  
## 6 2.667 NA NA NA NA NA 211.3806  
## Unemployment  
## 1 8.106  
## 2 8.106  
## 3 8.106  
## 4 8.106  
## 5 8.106  
## 6 8.106

We can see that there are few null values in the data set for column Markdown 1 - 5. We will also split the data column in 3 as Day, Month and Year.

dataset$Year <- year(ymd(dataset$Date))  
dataset$Month <- month(ymd(dataset$Date))   
dataset$Day <- day(ymd(dataset$Date))  
dataset$Dept = as.factor(dataset$Dept)  
dataset$Store = as.factor(dataset$Store)  
dataset$MarkDown1[is.na(dataset$MarkDown1)] = 0  
dataset$MarkDown2[is.na(dataset$MarkDown2)] = 0  
dataset$MarkDown3[is.na(dataset$MarkDown3)] = 0  
dataset$MarkDown4[is.na(dataset$MarkDown4)] = 0  
dataset$MarkDown5[is.na(dataset$MarkDown5)] = 0  
dataset = fastDummies::dummy\_cols(dataset, select\_columns = "Type")  
dataset$IsHoliday[dataset$isHoliday == "False"] = 0  
dataset$IsHoliday[dataset$isHoliday == "True"] = 1  
head(dataset)

## Store Dept Date weeklySales isHoliday Type Size Temperature  
## 1 1 1 2010-02-05 24924.50 False A 151315 42.31  
## 2 1 1 2010-02-12 46039.49 True A 151315 38.51  
## 3 1 1 2010-02-19 41595.55 False A 151315 39.93  
## 4 1 1 2010-02-26 19403.54 False A 151315 46.63  
## 5 1 1 2010-03-05 21827.90 False A 151315 46.50  
## 6 1 1 2010-03-12 21043.39 False A 151315 57.79  
## Fuel\_Price MarkDown1 MarkDown2 MarkDown3 MarkDown4 MarkDown5 CPI  
## 1 2.572 0 0 0 0 0 211.0964  
## 2 2.548 0 0 0 0 0 211.2422  
## 3 2.514 0 0 0 0 0 211.2891  
## 4 2.561 0 0 0 0 0 211.3196  
## 5 2.625 0 0 0 0 0 211.3501  
## 6 2.667 0 0 0 0 0 211.3806  
## Unemployment Year Month Day Type\_A Type\_B Type\_C IsHoliday  
## 1 8.106 2010 2 5 1 0 0 0  
## 2 8.106 2010 2 12 1 0 0 1  
## 3 8.106 2010 2 19 1 0 0 0  
## 4 8.106 2010 2 26 1 0 0 0  
## 5 8.106 2010 3 5 1 0 0 0  
## 6 8.106 2010 3 12 1 0 0 0

sapply(dataset, function(x) sum(is.infinite(x)))

## Store Dept Date weeklySales isHoliday Type   
## 0 0 0 0 0 0   
## Size Temperature Fuel\_Price MarkDown1 MarkDown2 MarkDown3   
## 0 0 0 0 0 0   
## MarkDown4 MarkDown5 CPI Unemployment Year Month   
## 0 0 0 0 0 0   
## Day Type\_A Type\_B Type\_C IsHoliday   
## 0 0 0 0 0

sapply(dataset, function(x) sum(is.na(x)))

## Store Dept Date weeklySales isHoliday Type   
## 0 0 0 0 0 0   
## Size Temperature Fuel\_Price MarkDown1 MarkDown2 MarkDown3   
## 0 0 0 0 0 0   
## MarkDown4 MarkDown5 CPI Unemployment Year Month   
## 0 0 0 0 0 0   
## Day Type\_A Type\_B Type\_C IsHoliday   
## 0 0 0 0 0

features = c("IsHoliday","Type\_A","Type\_B","Type\_C","Size","Temperature","Fuel\_Price","CPI","Unemployment","Year","Month","Day")  
dataset2 = select(dataset,features) %>% slice(1:1000)

## Note: Using an external vector in selections is ambiguous.  
## i Use `all\_of(features)` instead of `features` to silence this message.  
## i See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.  
## This message is displayed once per session.

dim(dataset2)

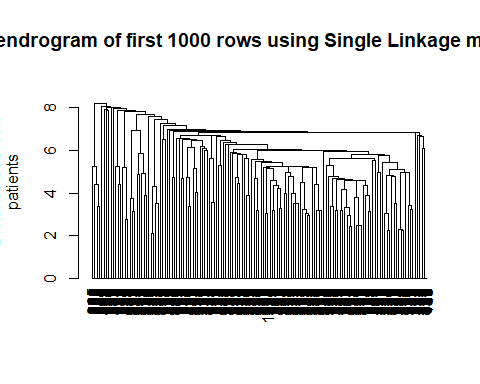
## [1] 1000 12

Distance <- dist(dataset2, method="euclidean")

#Hirerarchical Methods  
#1. Single Linkage Method  
# Invoking hclust command (cluster analysis by single linkage method)  
clus\_sales\_prediction.nn <- hclust(Distance, method = "single")  
clus\_sales\_prediction.nn

##   
## Call:  
## hclust(d = Distance, method = "single")  
##   
## Cluster method : single   
## Distance : euclidean   
## Number of objects: 1000

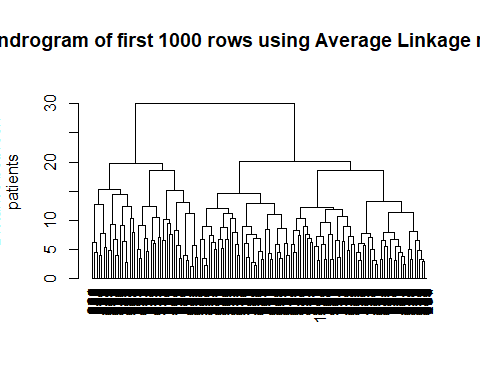
#Plotting of dendrogram using Single Linkage method  
plot(as.dendrogram(clus\_sales\_prediction.nn),ylab="Distance between  
patients",ylim=c(0,9), main="Dendrogram of first 1000 rows using Single Linkage method")



#2. Average Linkage Method  
clus\_sales\_prediction.avl <- hclust(Distance, method = "average")  
clus\_sales\_prediction.avl

##   
## Call:  
## hclust(d = Distance, method = "average")  
##   
## Cluster method : average   
## Distance : euclidean   
## Number of objects: 1000

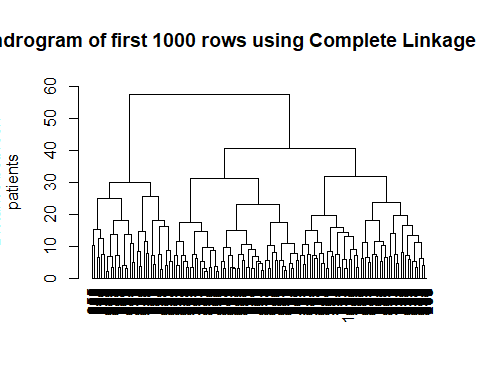
#Plotting of dendrogram using Average Linkage method  
plot(as.dendrogram(clus\_sales\_prediction.avl),ylab="Distance between  
patients",ylim=c(0,33),  
 main="Dendrogram of first 1000 rows using Average Linkage method")



#3. Complete Linkage Method  
clus\_sales\_prediction.fn <- hclust(Distance)  
clus\_sales\_prediction.fn

##   
## Call:  
## hclust(d = Distance)  
##   
## Cluster method : complete   
## Distance : euclidean   
## Number of objects: 1000

plot(as.dendrogram(clus\_sales\_prediction.fn),ylab="Distance between  
patients",ylim=c(0,60),  
 main="Dendrogram of first 1000 rows using Complete Linkage method")



#Non-hirerarchical Method  
#K-Means  
# Centers (k's) are numbers thus, 10 random sets are chosen  
(kmeans2.dataset <- kmeans(dataset2,2,nstart = 10))

## K-means clustering with 2 clusters of sizes 608, 392  
##   
## Cluster means:  
## IsHoliday Type\_A Type\_B Type\_C Size Temperature Fuel\_Price CPI  
## 1 0.03453947 1 0 0 151315 78.00280 3.309079 216.5814  
## 2 0.12500000 1 0 0 151315 53.26589 3.080339 215.0713  
## Unemployment Year Month Day  
## 1 7.530895 2011.044 6.902961 16.57072  
## 2 7.736411 2010.839 5.732143 14.26786  
##   
## Clustering vector:  
## [1] 2 2 2 2 2 2 2 2 2 2 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1  
## [38] 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 1 1 1 1 2 1 1 1 1 1 1 1 1  
## [75] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  
## [112] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 2 2 2 2 2  
## [149] 2 2 2 2 2 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 2 2 2  
## [186] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1  
## [223] 1 1 1 1 1 1 1 1 1 1 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1  
## [260] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 2 2 2 2 2 2 2 2 2 2  
## [297] 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 2 2 2 2 2 2 2 2  
## [334] 2 2 2 2 2 2 2 2 2 2 2 2 1 2 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  
## [371] 1 1 1 1 1 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1  
## [408] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 2 2 2 2 2 2 2 2 2 2 1 2 1 1 1  
## [445] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2  
## [482] 2 2 2 2 2 2 2 1 2 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  
## [519] 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  
## [556] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 2 2 2 2 2 2 2 2 2 2 1 2 1 1 1 1 1 1 1 1  
## [593] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  
## [630] 2 2 1 2 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 2 2 2  
## [667] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  
## [704] 1 1 1 1 1 1 1 1 1 2 1 1 2 2 2 2 2 2 2 2 2 2 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1  
## [741] 1 1 1 1 1 1 1 1 1 1 2 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 1  
## [778] 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 2 2 2 2 2 2 2 2  
## [815] 2 2 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  
## [852] 1 1 1 1 2 1 1 2 2 2 2 2 2 2 2 2 2 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  
## [889] 1 1 1 1 1 2 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 1 1 1 1 2 1  
## [926] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2  
## [963] 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2  
## [1000] 1  
##   
## Within cluster sum of squares by cluster:  
## [1] 92813.91 67925.67  
## (between\_SS / total\_SS = 47.9 %)  
##   
## Available components:  
##   
## [1] "cluster" "centers" "totss" "withinss" "tot.withinss"  
## [6] "betweenss" "size" "iter" "ifault"

# Computing the percentage of variation accounted for. Two clusters  
perc.var.2 <- round(100\*(1 -  
kmeans2.dataset$betweenss/kmeans2.dataset$totss),1)  
names(perc.var.2) <- "Perc. 2 clus"  
perc.var.2

## Perc. 2 clus   
## 52.1