COMP6115 - Assigmment 2

Ottor Mills

April 30, 2022

ID: 620098373

Task 01

Cleaning and preparing environment

```
rm(list = ls())
options(scipen = 99999)
```

Loading libraries

```
library(classInt)
```

Setting working directory and loading dataset

```
tryCatch({
    setwd(getSrcDirectory()[1])
}, error = function(e) {
    setwd(dirname(rstudioapi::getActiveDocumentContext()$path))
})

dataset <- read.csv("./datasets/SalesData.csv", na.strings = c("", " ", "\"\"", "?",
    "??", "???", "!"), stringsAsFactors = T)
```

Dataset loaded from the datasets folder in the working directory

Data Cleaning

Replacing NA's from Region column

```
dataset$Region <- as.character(dataset$Region)
dataset$Region[is.na(dataset$Region)] <- "Unknown"
dataset$Region <- factor(dataset$Region)
```

NA values were removed from the "Region" column

Replacing NA's from ItemType column

```
dataset$ItemType <- as.character(dataset$ItemType)
dataset$ItemType[is.na(dataset$ItemType)] <- "Unknown"
dataset$ItemType <- factor(dataset$ItemType)
```

NA values were removed from the "Item Type" column

Replacing NA's and irrelevant values from SalesChannel column

```
dataset$Sales.Channel <- as.character(dataset$Sales.Channel)
dataset$Sales.Channel[is.na(dataset$Sales.Channel)] <- "Unknown"
dataset$Sales.Channel[dataset$Sales.Channel != "Online" & dataset$Sales.Channel !=
    "Offline"] <- "Unknown"
dataset$Sales.Channel <- factor(dataset$Sales.Channel)
```

NA values were removed from the "Sales Channel" column

Convert dataset dates to date objects

```
dataset$Order.Date <- as.Date(dataset$Order.Date, format = "%m/%d/%Y")
dataset$Ship.Date <- as.Date(dataset$Ship.Date, format = "%m/%d/%Y")
```

The date columns of the dataset were converted to date objects for easier date related processing

Normalization of dataset unit price column

```
dataset$Unit.Price <- ((dataset$Unit.Price - min(dataset$Unit.Price))/(max(dataset$Unit.Price) -
    min(dataset$Unit.Price))) * (10 - 1) + 1
```

The "Unit Price" column was normalized using the min-max normalization technique

Discretization of "Units Sold" column

```
bins <- classIntervals(dataset$Units.Sold, 3, style = "equal")
unitsSoldType = c()
for (i in 1:length(dataset$Units.Sold)) {
    rating = "L"
    if (dataset$Units.Sold[i] > bins$brks[2] && dataset$Units.Sold[i] <= bins$brks[3]) {
        rating = "M"
    }
    if (dataset$Units.Sold[i] > bins$brks[3]) {
        rating = "H"
    }
    unitsSoldType <- append(unitsSoldType, rating)
}
dataset$Units.Sold.Type <- as.factor(unitsSoldType)
```

Discretization of the "Units Sold" column was carries out using equal width binning and saved into a new column called "Units Sold Type".

Key:

L	Low
M	Medium
H	High

From the density plot we can see that three clusters can be identified

Save Dataset

```
write.csv(dataset, "./output/sales-data_cleaned.csv", row.names = F)
```

Output cleaned dataset to output folder of working directory

Solution

The goal of this study was to identify how attributes of the provided sales data data set are related. Outlined in this section is the approach which was taken to achieve this goal with aiding visual representation of both agglomorative and kmeans clusters.

Loading Cleaned Dataset

```
cleanedDataset <- read.csv("./output/sales-data_cleaned.csv", stringsAsFactors = T)
```

Calculation of shipping time

```
deliveryDays <- c()
for (i in 1:length(cleanedDataset$Order.Date)) {
    shippingDays <- as.Date(cleanedDataset$Ship.Date[i]) - as.Date(cleanedDataset$Order.Date[i])
    deliveryDays <- append(deliveryDays, as.integer(strtoi(shippingDays, base = 0L)))
}
cleanedDataset$Ship.Time <- deliveryDays
```

The skipping time was calculated by finding the difference between the "Ship Date" from the "Order Date" this produced the difference in days which was formatted appropriately.

Removing of ID fields

```
cleanedDataset$Order.ID <- NULL
```

ID fields are not needed for cluster generation and were removed

Numerical conversion of data

```
cleanedDataset <- dplyr::mutate_all(cleanedDataset, function(x) as.numeric(x))
```

All fields were converted into numerical format in preparation for clustering

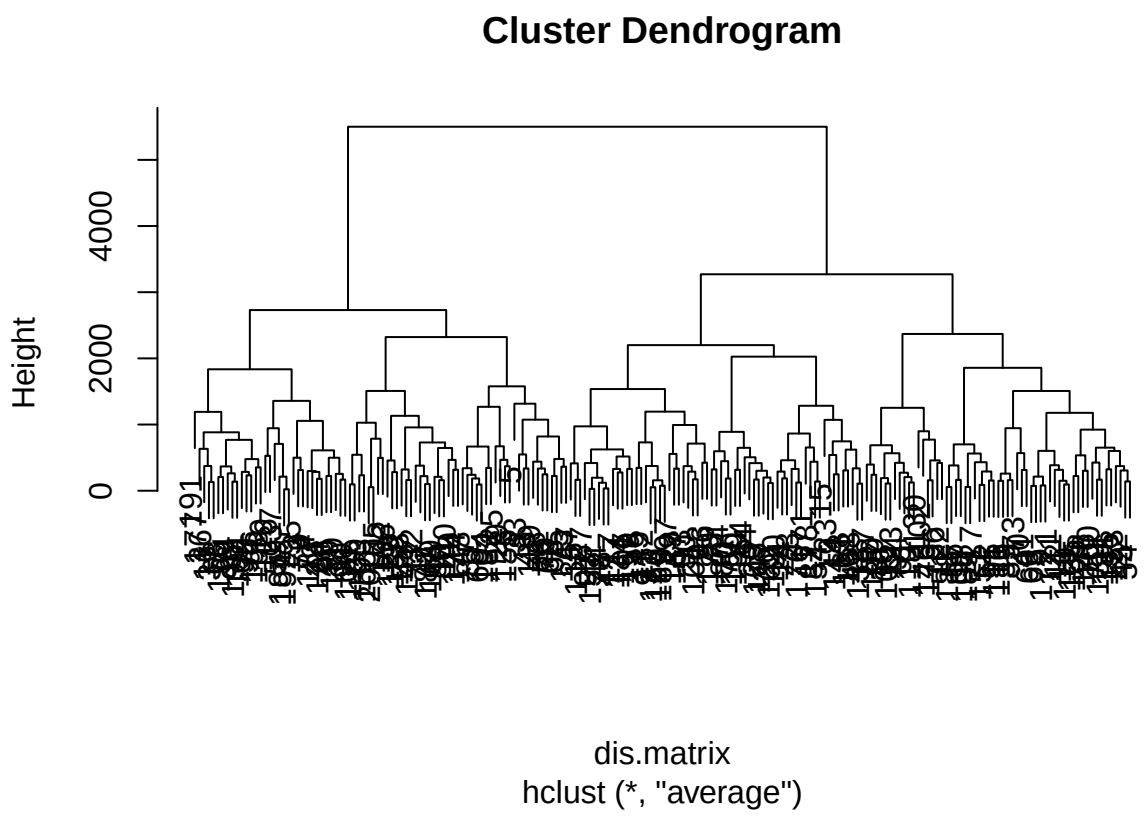
Sample generation

```
sample.salesdata <- cleanedDataset[1:200, ]
```

A subset of the data was selected to be used for cluster generation

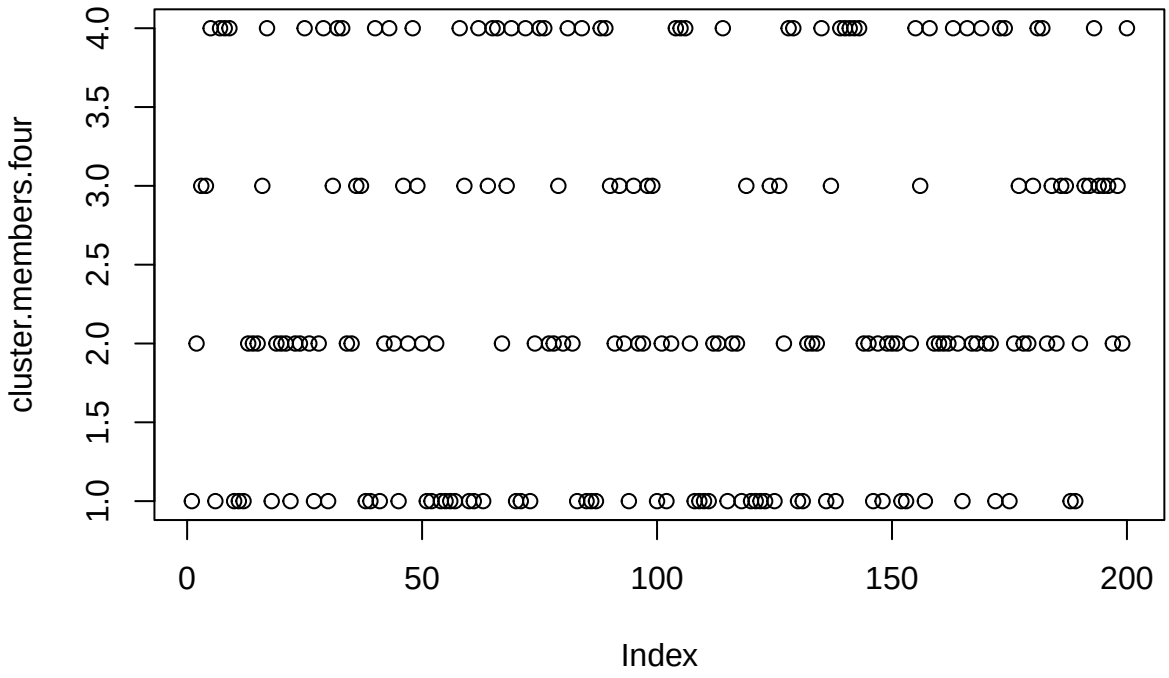
Hierichical Clustering (Agglomorative)

```
dis.matrix <- dist(sample.salesdata)
hclust.01 <- hclust(dis.matrix, method = "average")
plot(hclust.01)
```



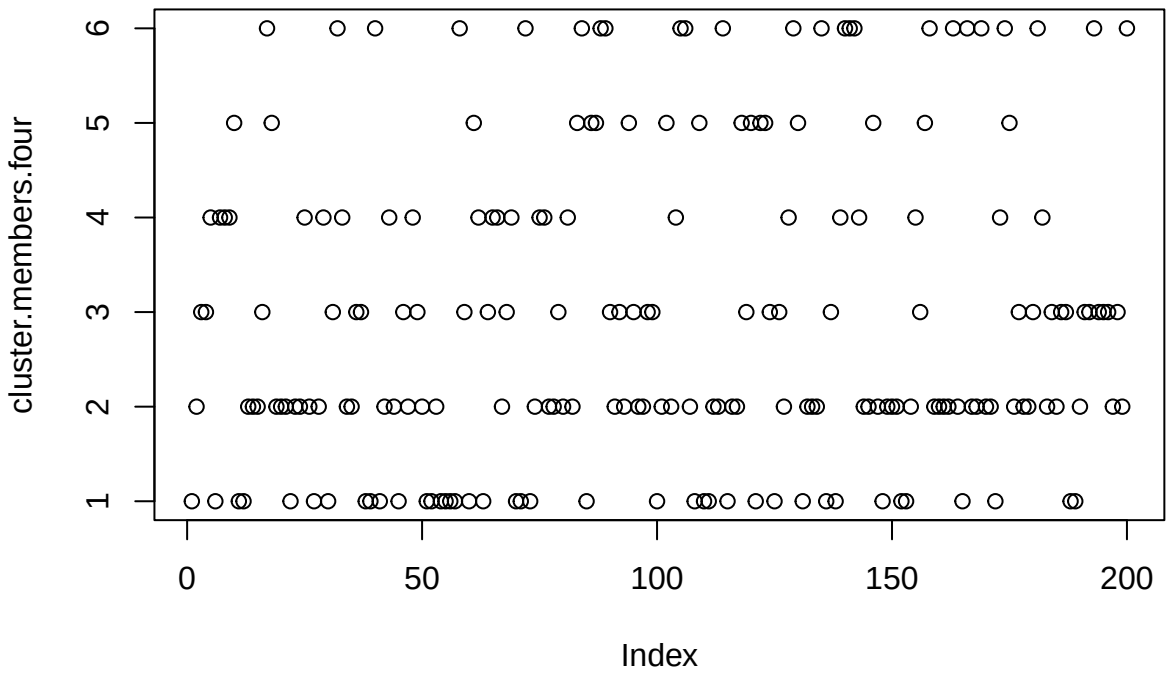
4 cluster :

```
cluster.members.four <- cutree(hclust.01, 4)
plot(cluster.members.four)
```



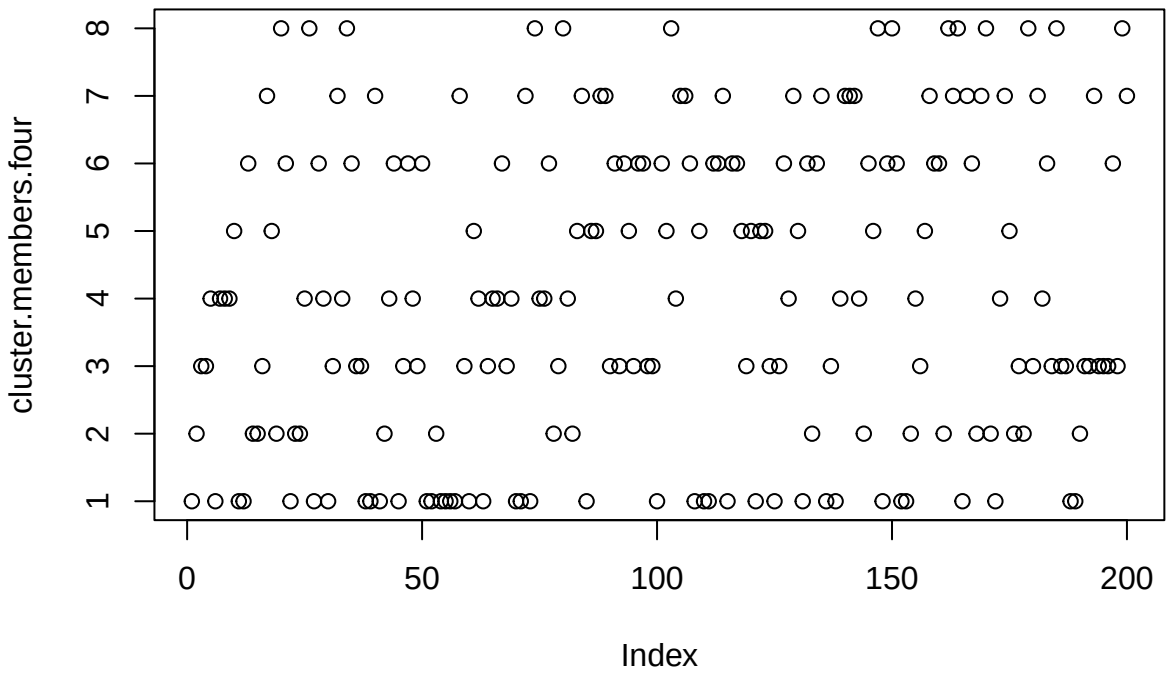
6 cluster :

```
cluster.members.four <- cutree(hclust.01, 6)
plot(cluster.members.four)
```



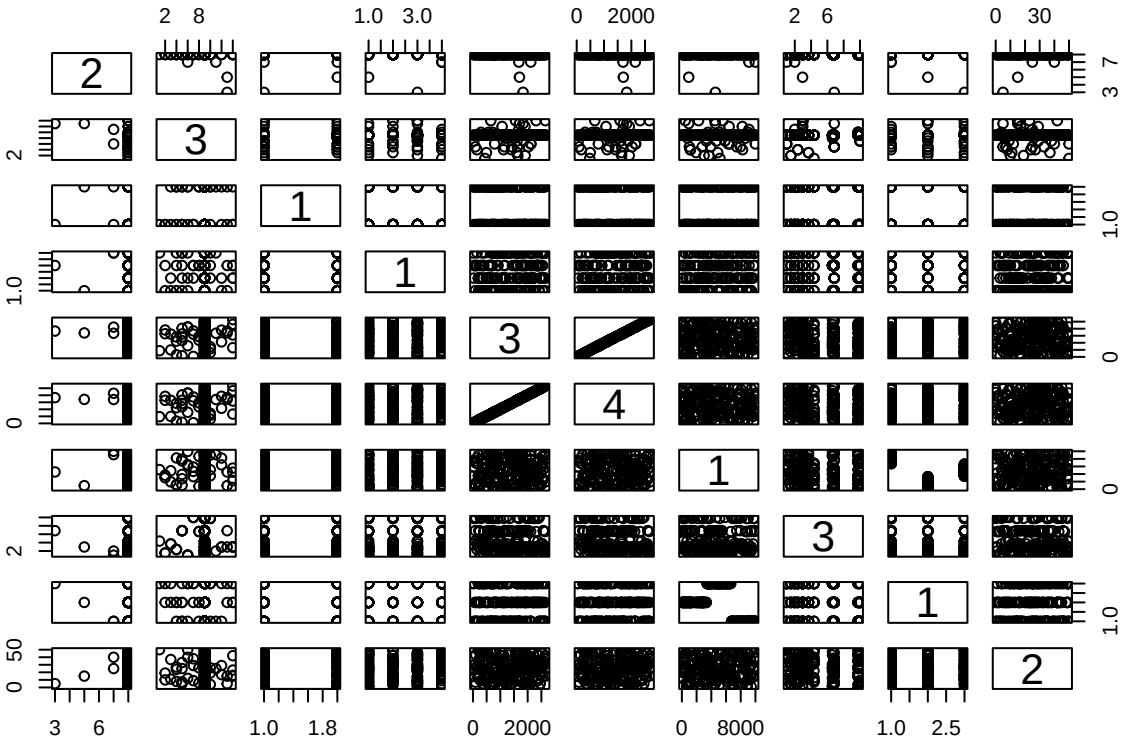
8 cluster :

```
cluster.members.four <- cutree(hclust.01, 8)
plot(cluster.members.four)
```



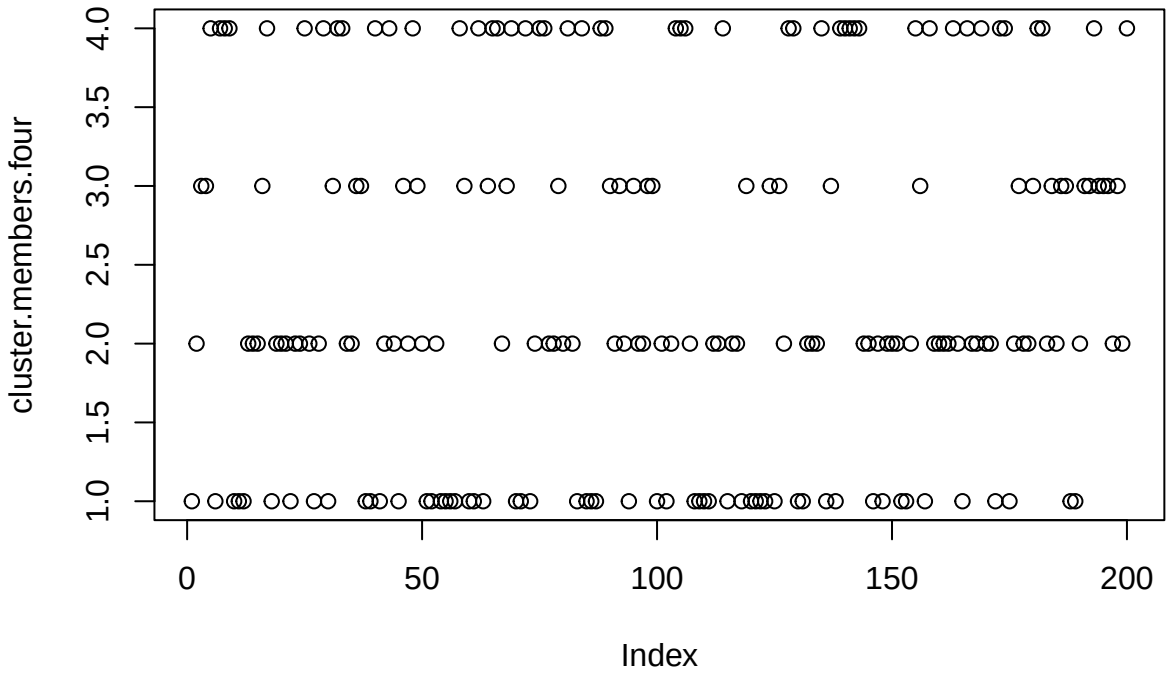
KMeans Clustustering

```
km.results.four <- kmeans(sample.salesdata, 4)
plot(sample.salesdata, km.results.four$cluster)
```



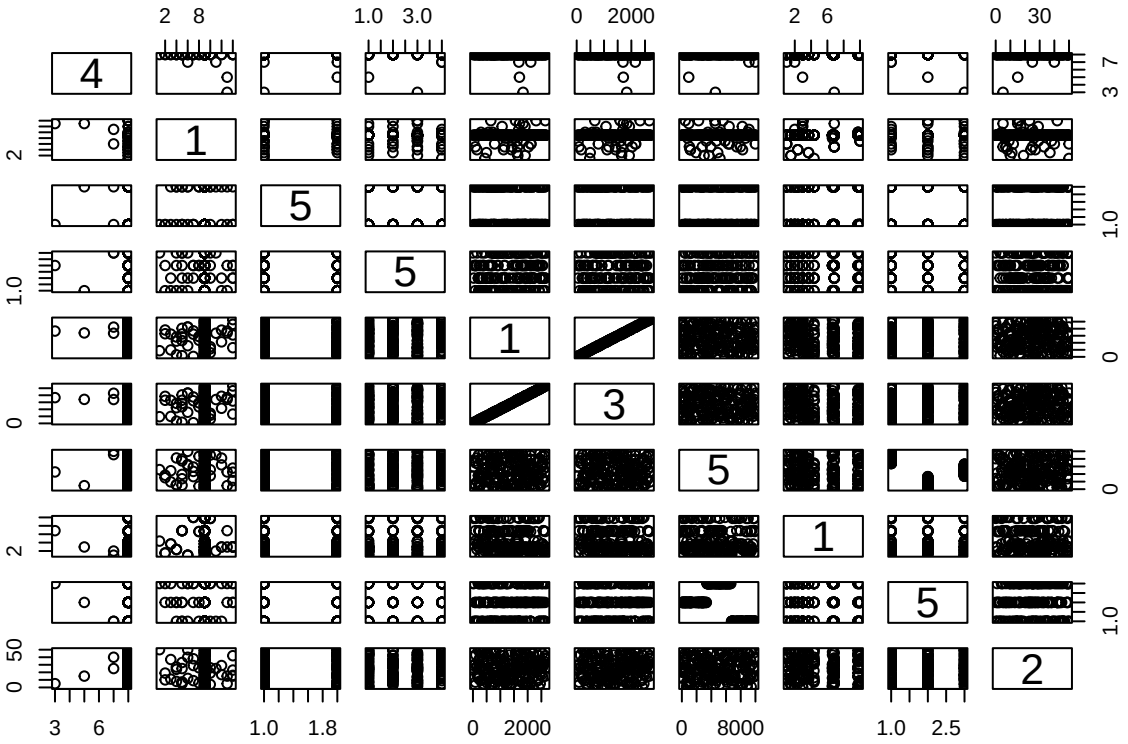
4 cluster :

```
cluster.members.four <- cutree(hclust.01, 4)
plot(cluster.members.four)
```



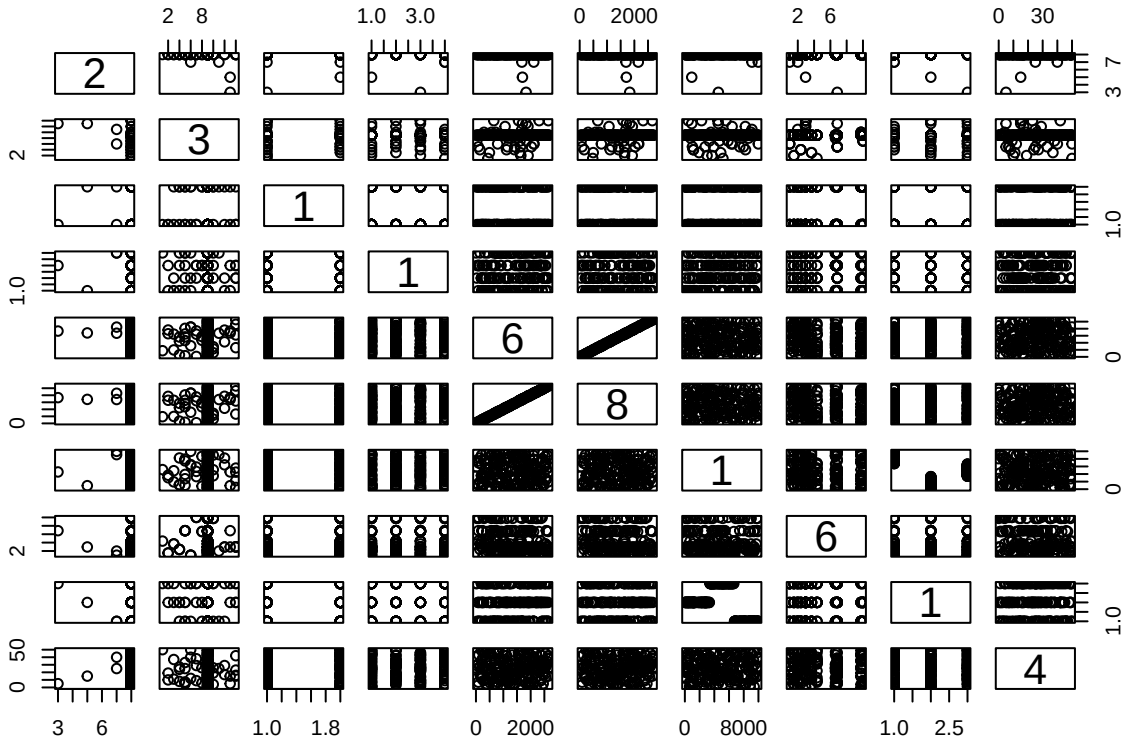
6 cluster :

```
km.results.six <- kmeans(sample.salesdata, 6)
plot(sample.salesdata, km.results.six$cluster)
```



8 cluster :

```
km.results.eight <- kmeans(sample.salesdata, 8)
plot(sample.salesdata, km.results.eight$cluster)
```



Task 02

Cleaning and preparing environment

```
rm(list = ls())
options(scipen = 99999)
```

Loading libraries

```
pacman::p_load(
  "arules",
  # "arulesViz",
  "zeallot",
  "backports",
  "classInt",
  "dplyr",
  "chron"
)
```

Installation of the necessary libraries using pacman

Setting working directory and loading dataset

```
tryCatch({
    setwd(getSrcDirectory()[1])
}, error = function(e) {
    setwd(dirname(rstudioapi::getActiveDocumentContext()$path))
})

dataset <- read.csv("./datasets/OnlineRetail.csv", na.strings = c("", " ", "\"\"",
    "?", "??", "???", "!"), stringsAsFactors = T)
```

Dataset loaded from the datasets folder in the working directory

Data Cleaning

Removing unnecessary countries

```
dataset <- filter(dataset, Country == "Switzerland")
```

Only records originating in "Switzerland" were selected for this study

Removing unnecessary countries

```
dataset <- filter(dataset, nchar(as.character(dataset$Description)) > 5)
```

Removal of meaningless descriptions. Most item descriptions were observed to have long titles.

Correcting negative quantities

```
dataset$Quantity <- abs(dataset$Quantity)
dataset$UnitPrice <- abs(dataset$UnitPrice)
```

Corrections were made by overriding each quality value with their corresponding absolute value

Correcting negative quantities

```
dataset$Quantity <- abs(dataset$Quantity)
dataset$UnitPrice <- abs(dataset$UnitPrice)
```

Outlier Removal

```
q1UnitPrice <- summary(dataset$UnitPrice)[2]
q3UnitPrice <- summary(dataset$UnitPrice)[5]
IQR <- q3UnitPrice - q1UnitPrice

dataset <- dataset[dataset$UnitPrice >= q1UnitPrice - 1.5 * IQR & dataset$UnitPrice <=
    q3UnitPrice + 1.5 * IQR, ]
```

Removal of outliers using the interquartile range (IQR). A point is an outlier if it is above the 75th or below the 25th percentile by a factor of 1.5 times the IQR as shown in the code snippet above.

Discretization of Quantity

```
dataset$Quantity <- cut(dataset$Quantity, c(0, 5, 10, 15, max(dataset$Quantity)),
    right = TRUE, labels = c("L", "M", "H", "VH"))
```

Discretization of "Quantity" was done to generate 4 categorical values, which also addresses outliers

Normalization of UnitPrice

```
dataset$UnitPrice <- ((dataset$UnitPrice - min(dataset$UnitPrice))/(max(dataset$UnitPrice) -
    min(dataset$UnitPrice))) * (10 - 1) + 1
```

Replacing Unwanted Characters

```
dataset$Description <- trimws(dataset$Description)
```

Unwanted characters were removed from the description column. The trimws function removes whitespaces from the ends of each description value

Replacing Whitespaces

```
dataset$Description <- gsub(" ", "_", dataset$Description)
```

Whitespaces were removed from the "Description" column in order to minimize the risk of human related errors due to whitespaces

Removal of free items

```
dataset <- filter(dataset, UnitPrice > 0)
```

Free items were removed from the dataset as they could potentially interrupt the meaningfulness of the study

Removal of free items

```
dataset$InvoiceNo <- gsub("C", "", dataset$InvoiceNo)
```

The "C" characters were removed from the "InvoiceNo" column in order to make the enter column integer values

Cleaning date field (separation of date and time)

```
invoiceTimes <- c()
invoiceDates <- c()
for (i in 1:length(dataset$InvoiceDate)) {
    splittedDate <- strsplit(toString(dataset$InvoiceDate[i]), " ")
    invoiceDate <- splittedDate[[1]][1]
    # Seconds are needed in order to convert string to chron objects
    invoiceTime <- paste(splittedDate[[1]][2], ":00", sep = "")
    invoiceTimes <- append(invoiceTimes, invoiceTime)
    invoiceDates <- append(invoiceDates, invoiceDate)
}
dataset$InvoiceDate <- invoiceDates
```

The dates in the "InvoiceDate" field were split. The time portion of each date was stripped before being used to override the previous date value

Creation of Chron column

```
invoiceDates <- as.Date(dataset$InvoiceDate, format = "%m/%d/%Y")
chronDateValues <- chron(date = as.character(invoiceDates), times = invoiceTimes,
    format = c(dates = "Y-m-d", times = "h:m:s"))
dataset <- tibble::add_column(dataset, ChronDates = chronDateValues, .after = "InvoiceDate")
```

The stripped time portion of each date in the "InvoiceDate" field are utilized to generate chron date values which are saved into a new column called "ChronDates"

Saving of Cleaned Dataset

```
write.csv(dataset, "./output/switzerland-retail-data_cleaned.csv", row.names = F)
```

the cleaned dataset was saved to file

Solution

The goal of this study was to identify ideal support and confidence values that resulted in the maximal lift.

Loading of Cleaned Dataset

```
switzerlandRetailData <- read.transactions("./output/switzerland-retail-data_cleaned.csv",
    format = c("single"), header = TRUE, rm.duplicates = FALSE, cols = c("InvoiceNo",
        "StockCode"), sep = ",")
```

Apriori Mining

```
drules2 <- apriori(switzerlandRetailData, parameter = list(support = 0.1, confidence = 0.8,
    minlen = 2, maxlen = 4))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.8    0.1    1 none FALSE            TRUE       5     0.1      2
##  maxlen target  ext
##       4  rules TRUE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 6
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[889 item(s), 65 transaction(s)] done [0.00s].
## sorting and recoding items ... [19 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 done [0.00s].
## writing ... [7 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
summary(drules2)
```

```
## set of 7 rules
##
## rule length distribution (lhs + rhs):sizes
## 2 3
## 4 3
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   2.000   2.000   2.000   2.429   3.000   3.000
##
## summary of quality measures:
##     support         confidence        coverage           lift
##  Min.   :0.1077   Min.   :0.8000   Min.   :0.1077   Min.   :3.041
##  1st Qu.:0.1231   1st Qu.:0.8535   1st Qu.:0.1308   1st Qu.:3.084
##  Median :0.1231   Median :0.9000   Median :0.1538   Median :3.128
##  Mean   :0.1297   Mean   :0.9023   Mean   :0.1451   Mean   :3.231
```

```
##  3rd Qu.:0.1385   3rd Qu.:0.9545   3rd Qu.:0.1615   3rd Qu.:3.336
##  Max.   :0.1538   Max.   :1.0000   Max.   :0.1692   Max.   :3.611
##      count
##  Min.   : 7.000
##  1st Qu.: 8.000
##  Median : 8.000
##  Mean   : 8.429
##  3rd Qu.: 9.000
##  Max.   :10.000
##
## mining info:
##                   data ntransactions support confidence
##  switzerlandRetailData            65     0.1        0.8
##
##  apriori(data = switzerlandRetailData, parameter = list(support = 0.1, confidence = 0.8, minlen = 2,
```

```
# plot(drules2)
```

Apriori mining carried out on cleansed dataset in order to identify positively related associations. (Lift value > 1). The ideal values for support was found to be 0.1 and the ideal value for lift was found to be 0.8. A diagram could not be provided at this time due to an error installing the "arulesViz" package.