R Project - Customer Behaviour Analysis

Geoffrey Chege

2022-06-04

# 1. Problem Definition

* Kira Plastinina is a Russian brand that is sold through a defunct chain of retail stores in Russia, Ukraine, Kazakhstan, Belarus, China, Philippines, and Armenia. The brand’s Sales and Marketing team would like to understand their customer’s behavior from data that they have collected over the past year. More specifically, they would like to learn the characteristics of customer groups.

# 2. Steps Taken

* Problem Definition
* Data Sourcing
* Check the Data
* Perform Data Cleaning
* Perform Exploratory Data Analysis:
  + Univariate
  + Bivariate
  + Multivariate
* Implement the Solution
* Challenge the Solution
* Follow up Questions

# 3. Data Sourcing

* The dataset for this Independent project can be found here <http://bit.ly/EcommerceCustomersDataset>
* The dataset consists of 10 numerical and 8 categorical attributes. The ‘Revenue’ attribute can be used as the class label.
* “Administrative”, “Administrative Duration”, “Informational”, “Informational Duration”, “Product Related” and “Product Related Duration” represents the number of different types of pages visited by the visitor in that session and total time spent in each of these page categories. The values of these features are derived from the URL information of the pages visited by the user and updated in real-time when a user takes an action, e.g. moving from one page to another.
* The “Bounce Rate”, “Exit Rate” and “Page Value” features represent the metrics measured by “Google Analytics” for each page in the e-commerce site.
* The value of the “Bounce Rate” feature for a web page refers to the percentage of visitors who enter the site from that page and then leave (“bounce”) without triggering any other requests to the analytics server during that session.
* The value of the “Exit Rate” feature for a specific web page is calculated as for all pageviews to the page, the percentage that was the last in the session.
* The “Page Value” feature represents the average value for a web page that a user visited before completing an e-commerce transaction.
* The “Special Day” feature indicates the closeness of the site visiting time to a specific special day (e.g. Mother’s Day, Valentine’s Day) in which the sessions are more likely to be finalized with the transaction. The value of this attribute is determined by considering the dynamics of e-commerce such as the duration between the order date and delivery date. For example, for Valentina’s day, this value takes a nonzero value between February 2 and February 12, zero before and after this date unless it is close to another special day, and its maximum value of 1 on February 8.
* The dataset also includes the operating system, browser, region, traffic type, visitor type as returning or new visitor, a Boolean value indicating whether the date of the visit is weekend, and month of the year.

# 4. Installing and loading Necessary Packages

# 5. Check the Data

df <- read.csv("C:/Users/user/Downloads/online\_shoppers\_intention.csv") # loading the file  
head(df) # displaying the first 5 elements of the data

## Administrative Administrative\_Duration Informational Informational\_Duration  
## 1 0 0 0 0  
## 2 0 0 0 0  
## 3 0 -1 0 -1  
## 4 0 0 0 0  
## 5 0 0 0 0  
## 6 0 0 0 0  
## ProductRelated ProductRelated\_Duration BounceRates ExitRates PageValues  
## 1 1 0.000000 0.20000000 0.2000000 0  
## 2 2 64.000000 0.00000000 0.1000000 0  
## 3 1 -1.000000 0.20000000 0.2000000 0  
## 4 2 2.666667 0.05000000 0.1400000 0  
## 5 10 627.500000 0.02000000 0.0500000 0  
## 6 19 154.216667 0.01578947 0.0245614 0  
## SpecialDay Month OperatingSystems Browser Region TrafficType  
## 1 0 Feb 1 1 1 1  
## 2 0 Feb 2 2 1 2  
## 3 0 Feb 4 1 9 3  
## 4 0 Feb 3 2 2 4  
## 5 0 Feb 3 3 1 4  
## 6 0 Feb 2 2 1 3  
## VisitorType Weekend Revenue  
## 1 Returning\_Visitor FALSE FALSE  
## 2 Returning\_Visitor FALSE FALSE  
## 3 Returning\_Visitor FALSE FALSE  
## 4 Returning\_Visitor FALSE FALSE  
## 5 Returning\_Visitor TRUE FALSE  
## 6 Returning\_Visitor FALSE FALSE

# 6. Data Cleaning

## 6.1 Missing Values

# checking for missing values  
  
colSums(is.na(df))

## Administrative Administrative\_Duration Informational   
## 14 14 14   
## Informational\_Duration ProductRelated ProductRelated\_Duration   
## 14 14 14   
## BounceRates ExitRates PageValues   
## 14 14 0   
## SpecialDay Month OperatingSystems   
## 0 0 0   
## Browser Region TrafficType   
## 0 0 0   
## VisitorType Weekend Revenue   
## 0 0 0

* There are missing values in 8 of the columns. Each column has 14 missing values.
* I will remove them before I continue my analysis.

# dropping null values  
  
df <- na.omit(df)

* Confirming the changes.

# confirming there are no null values  
  
colSums(is.na(df))

## Administrative Administrative\_Duration Informational   
## 0 0 0   
## Informational\_Duration ProductRelated ProductRelated\_Duration   
## 0 0 0   
## BounceRates ExitRates PageValues   
## 0 0 0   
## SpecialDay Month OperatingSystems   
## 0 0 0   
## Browser Region TrafficType   
## 0 0 0   
## VisitorType Weekend Revenue   
## 0 0 0

## 6.2 Checking for duplicates

duplicates <- df[duplicated(df),] # creating a table and storing the duplicates in it  
head(duplicates) # displaying the table

## Administrative Administrative\_Duration Informational Informational\_Duration  
## 159 0 0 0 0  
## 179 0 0 0 0  
## 419 0 0 0 0  
## 457 0 0 0 0  
## 484 0 0 0 0  
## 513 0 0 0 0  
## ProductRelated ProductRelated\_Duration BounceRates ExitRates PageValues  
## 159 1 0 0.2 0.2 0  
## 179 1 0 0.2 0.2 0  
## 419 1 0 0.2 0.2 0  
## 457 1 0 0.2 0.2 0  
## 484 1 0 0.2 0.2 0  
## 513 1 0 0.2 0.2 0  
## SpecialDay Month OperatingSystems Browser Region TrafficType  
## 159 0 Feb 1 1 1 3  
## 179 0 Feb 3 2 3 3  
## 419 0 Mar 1 1 1 1  
## 457 0 Mar 2 2 4 1  
## 484 0 Mar 3 2 3 1  
## 513 0 Mar 2 2 1 1  
## VisitorType Weekend Revenue  
## 159 Returning\_Visitor FALSE FALSE  
## 179 Returning\_Visitor FALSE FALSE  
## 419 Returning\_Visitor TRUE FALSE  
## 457 Returning\_Visitor FALSE FALSE  
## 484 Returning\_Visitor FALSE FALSE  
## 513 Returning\_Visitor FALSE FALSE

* I will drop the duplicates.

# eliminating duplicates  
df <- df[!duplicated(df), ]

* Confirming that there are no more duplicates.

### Dataset structure  
str(df)

## 'data.frame': 12199 obs. of 18 variables:  
## $ Administrative : int 0 0 0 0 0 0 0 1 0 0 ...  
## $ Administrative\_Duration: num 0 0 -1 0 0 0 -1 -1 0 0 ...  
## $ Informational : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ Informational\_Duration : num 0 0 -1 0 0 0 -1 -1 0 0 ...  
## $ ProductRelated : int 1 2 1 2 10 19 1 1 2 3 ...  
## $ ProductRelated\_Duration: num 0 64 -1 2.67 627.5 ...  
## $ BounceRates : num 0.2 0 0.2 0.05 0.02 ...  
## $ ExitRates : num 0.2 0.1 0.2 0.14 0.05 ...  
## $ PageValues : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ SpecialDay : num 0 0 0 0 0 0 0.4 0 0.8 0.4 ...  
## $ Month : chr "Feb" "Feb" "Feb" "Feb" ...  
## $ OperatingSystems : int 1 2 4 3 3 2 2 1 2 2 ...  
## $ Browser : int 1 2 1 2 3 2 4 2 2 4 ...  
## $ Region : int 1 1 9 2 1 1 3 1 2 1 ...  
## $ TrafficType : int 1 2 3 4 4 3 3 5 3 2 ...  
## $ VisitorType : chr "Returning\_Visitor" "Returning\_Visitor" "Returning\_Visitor" "Returning\_Visitor" ...  
## $ Weekend : logi FALSE FALSE FALSE FALSE TRUE FALSE ...  
## $ Revenue : logi FALSE FALSE FALSE FALSE FALSE FALSE ...  
## - attr(\*, "na.action")= 'omit' Named int [1:14] 1066 1133 1134 1135 1136 1137 1474 1475 1476 1477 ...  
## ..- attr(\*, "names")= chr [1:14] "1066" "1133" "1134" "1135" ...

## 6.3 Changing columns to factors

# changing character and logic columns to factors  
  
df$Month <- factor(df$Month)  
df$VisitorType <- factor(df$VisitorType)  
df$Weekend <- factor(df$Weekend)  
df$Revenue <- factor(df$Revenue)

* Month column is now a factor with 10 levels.
* VisitorType column is now a factor with 3 levels.
* Weekend column is now a factor with 2 levels.
* Revenue column is now a factor with 2 levels.

### Dataset structure  
str(df)

## 'data.frame': 12199 obs. of 18 variables:  
## $ Administrative : int 0 0 0 0 0 0 0 1 0 0 ...  
## $ Administrative\_Duration: num 0 0 -1 0 0 0 -1 -1 0 0 ...  
## $ Informational : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ Informational\_Duration : num 0 0 -1 0 0 0 -1 -1 0 0 ...  
## $ ProductRelated : int 1 2 1 2 10 19 1 1 2 3 ...  
## $ ProductRelated\_Duration: num 0 64 -1 2.67 627.5 ...  
## $ BounceRates : num 0.2 0 0.2 0.05 0.02 ...  
## $ ExitRates : num 0.2 0.1 0.2 0.14 0.05 ...  
## $ PageValues : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ SpecialDay : num 0 0 0 0 0 0 0.4 0 0.8 0.4 ...  
## $ Month : Factor w/ 10 levels "Aug","Dec","Feb",..: 3 3 3 3 3 3 3 3 3 3 ...  
## $ OperatingSystems : int 1 2 4 3 3 2 2 1 2 2 ...  
## $ Browser : int 1 2 1 2 3 2 4 2 2 4 ...  
## $ Region : int 1 1 9 2 1 1 3 1 2 1 ...  
## $ TrafficType : int 1 2 3 4 4 3 3 5 3 2 ...  
## $ VisitorType : Factor w/ 3 levels "New\_Visitor",..: 3 3 3 3 3 3 3 3 3 3 ...  
## $ Weekend : Factor w/ 2 levels "FALSE","TRUE": 1 1 1 1 2 1 1 2 1 1 ...  
## $ Revenue : Factor w/ 2 levels "FALSE","TRUE": 1 1 1 1 1 1 1 1 1 1 ...  
## - attr(\*, "na.action")= 'omit' Named int [1:14] 1066 1133 1134 1135 1136 1137 1474 1475 1476 1477 ...  
## ..- attr(\*, "names")= chr [1:14] "1066" "1133" "1134" "1135" ...

# 7. Exploratory Data Analysis

### A function to determine the mode

mode <- function(v){  
 uniq <- unique(v)  
 uniq[which.max(tabulate(match(v,uniq)))]  
}

### Summary statistics of the columns

summary(df)

## Administrative Administrative\_Duration Informational   
## Min. : 0.00 Min. : -1.00 Min. : 0.0000   
## 1st Qu.: 0.00 1st Qu.: 0.00 1st Qu.: 0.0000   
## Median : 1.00 Median : 9.00 Median : 0.0000   
## Mean : 2.34 Mean : 81.68 Mean : 0.5088   
## 3rd Qu.: 4.00 3rd Qu.: 94.75 3rd Qu.: 0.0000   
## Max. :27.00 Max. :3398.75 Max. :24.0000   
##   
## Informational\_Duration ProductRelated ProductRelated\_Duration  
## Min. : -1.00 Min. : 0.00 Min. : -1.0   
## 1st Qu.: 0.00 1st Qu.: 8.00 1st Qu.: 193.6   
## Median : 0.00 Median : 18.00 Median : 609.5   
## Mean : 34.84 Mean : 32.06 Mean : 1207.5   
## 3rd Qu.: 0.00 3rd Qu.: 38.00 3rd Qu.: 1477.6   
## Max. :2549.38 Max. :705.00 Max. :63973.5   
##   
## BounceRates ExitRates PageValues SpecialDay   
## Min. :0.00000 Min. :0.00000 Min. : 0.000 Min. :0.00000   
## 1st Qu.:0.00000 1st Qu.:0.01422 1st Qu.: 0.000 1st Qu.:0.00000   
## Median :0.00293 Median :0.02500 Median : 0.000 Median :0.00000   
## Mean :0.02045 Mean :0.04150 Mean : 5.952 Mean :0.06197   
## 3rd Qu.:0.01667 3rd Qu.:0.04848 3rd Qu.: 0.000 3rd Qu.:0.00000   
## Max. :0.20000 Max. :0.20000 Max. :361.764 Max. :1.00000   
##   
## Month OperatingSystems Browser Region   
## May :3328 Min. :1.000 Min. : 1.000 Min. :1.000   
## Nov :2983 1st Qu.:2.000 1st Qu.: 2.000 1st Qu.:1.000   
## Mar :1853 Median :2.000 Median : 2.000 Median :3.000   
## Dec :1706 Mean :2.124 Mean : 2.358 Mean :3.153   
## Oct : 549 3rd Qu.:3.000 3rd Qu.: 2.000 3rd Qu.:4.000   
## Sep : 448 Max. :8.000 Max. :13.000 Max. :9.000   
## (Other):1332   
## TrafficType VisitorType Weekend Revenue   
## Min. : 1.000 New\_Visitor : 1693 FALSE:9343 FALSE:10291   
## 1st Qu.: 2.000 Other : 81 TRUE :2856 TRUE : 1908   
## Median : 2.000 Returning\_Visitor:10425   
## Mean : 4.075   
## 3rd Qu.: 4.000   
## Max. :20.000   
##

### Description of Columns

describe(df)

## vars n mean sd median trimmed mad min  
## Administrative 1 12199 2.34 3.33 1.00 1.66 1.48 0  
## Administrative\_Duration 2 12199 81.68 177.53 9.00 42.87 13.34 -1  
## Informational 3 12199 0.51 1.28 0.00 0.18 0.00 0  
## Informational\_Duration 4 12199 34.84 141.46 0.00 3.73 0.00 -1  
## ProductRelated 5 12199 32.06 44.60 18.00 23.06 19.27 0  
## ProductRelated\_Duration 6 12199 1207.51 1919.93 609.54 832.36 745.12 -1  
## BounceRates 7 12199 0.02 0.05 0.00 0.01 0.00 0  
## ExitRates 8 12199 0.04 0.05 0.03 0.03 0.02 0  
## PageValues 9 12199 5.95 18.66 0.00 1.33 0.00 0  
## SpecialDay 10 12199 0.06 0.20 0.00 0.00 0.00 0  
## Month\* 11 12199 6.17 2.37 7.00 6.36 1.48 1  
## OperatingSystems 12 12199 2.12 0.91 2.00 2.06 0.00 1  
## Browser 13 12199 2.36 1.71 2.00 2.00 0.00 1  
## Region 14 12199 3.15 2.40 3.00 2.79 2.97 1  
## TrafficType 15 12199 4.07 4.02 2.00 3.22 1.48 1  
## VisitorType\* 16 12199 2.72 0.69 3.00 2.89 0.00 1  
## Weekend\* 17 12199 1.23 0.42 1.00 1.17 0.00 1  
## Revenue\* 18 12199 1.16 0.36 1.00 1.07 0.00 1  
## max range skew kurtosis se  
## Administrative 27.00 27.00 1.95 4.63 0.03  
## Administrative\_Duration 3398.75 3399.75 5.59 50.09 1.61  
## Informational 24.00 24.00 4.01 26.64 0.01  
## Informational\_Duration 2549.38 2550.38 7.54 75.45 1.28  
## ProductRelated 705.00 705.00 4.33 31.04 0.40  
## ProductRelated\_Duration 63973.52 63974.52 7.25 136.57 17.38  
## BounceRates 0.20 0.20 3.15 9.25 0.00  
## ExitRates 0.20 0.20 2.23 4.62 0.00  
## PageValues 361.76 361.76 6.35 64.93 0.17  
## SpecialDay 1.00 1.00 3.28 9.78 0.00  
## Month\* 10.00 9.00 -0.83 -0.37 0.02  
## OperatingSystems 8.00 7.00 2.03 10.27 0.01  
## Browser 13.00 12.00 3.22 12.53 0.02  
## Region 9.00 8.00 0.98 -0.16 0.02  
## TrafficType 20.00 19.00 1.96 3.47 0.04  
## VisitorType\* 3.00 2.00 -2.05 2.23 0.01  
## Weekend\* 2.00 1.00 1.26 -0.42 0.00  
## Revenue\* 2.00 1.00 1.89 1.58 0.00

## Univariate Analysis

### Administrative Column

* From the summary and description, we can gather the following about the administrative column:
  + Mean: 2.34
  + Median: 1
  + Skewness: 1.95
  + Kurtosis: 4.63
* The mode is:

mode(df$Administrative)

## [1] 0

### Informational Column

* From the summary and description, we can gather the following about the administrative column:
  + Mean: 0.51
  + Median: 0
  + Skewness: 4.01
  + Kurtosis: 26.64
* The mode is:

mode(df$Informational)

## [1] 0

### ProductRelated Column

* From the summary and description, we can gather the following about the administrative column:
  + Mean: 32.06
  + Median: 18
  + Skewness: 4.33
  + Kurtosis: 31.04
* The mode is:

mode(df$ProductRelated)

## [1] 1

### BounceRates

* From the summary and description, we can gather the following about the administrative column:
  + Mean: 0.02
  + Median: 0.00
  + Skewness: 3.15
  + Kurtosis: 9.25
* The mode is:

mode(df$BounceRates)

## [1] 0

### ExitRates

* From the summary and description, we can gather the following about the administrative column:
  + Mean: 0.04
  + Median: 0.03
  + Skewness: 2.23
  + Kurtosis: 4.62
* The mode is:

mode(df$ExitRates)

## [1] 0.2

### PageValues

* From the summary and description, we can gather the following about the administrative column:
  + Mean: 5.95
  + Median: 0
  + Skewness: 6.35
  + Kurtosis: 64.93
* The mode is:

mode(df$PageValues)

## [1] 0

### SpecialDay

* From the summary and description, we can gather the following about the administrative column:
  + Mean: 0.06
  + Median: 0
  + Skewness: 3.28
  + Kurtosis: 9.78
* The mode is:

mode(df$SpecialDay)

## [1] 0

### Month

* From the summary and description, we can gather the following about the administrative column:
  + Mean: 6.17
  + Median: 7
  + Skewness: -0.83
  + Kurtosis: -0.37
* The mode is:

mode(df$Month)

## [1] May  
## Levels: Aug Dec Feb Jul June Mar May Nov Oct Sep

### OperatingSystems

* From the summary and description, we can gather the following about the administrative column:
  + Mean: 2.12
  + Median: 2
  + Skewness: 2.03
  + Kurtosis: 10.27
* The mode is:

mode(df$OperatingSystems)

## [1] 2

### Browser

* From the summary and description, we can gather the following about the administrative column:
  + Mean: 2.36
  + Median: 2
  + Skewness: 3.22
  + Kurtosis: 12.53
* The mode is:

mode(df$Browser)

## [1] 2

### Region

* From the summary and description, we can gather the following about the administrative column:
  + Mean: 3.15
  + Median: 3
  + Skewness: 0.98
  + Kurtosis: -0.16
* The mode is:

mode(df$Region)

## [1] 1

### TrafficType

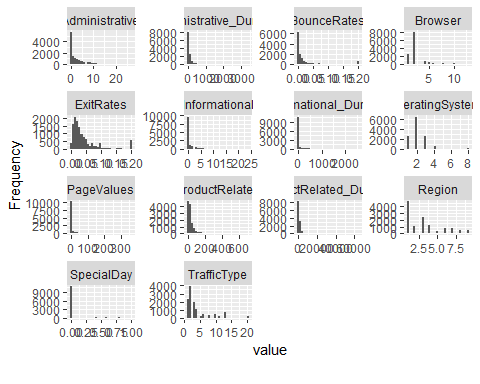
* From the summary and description, we can gather the following about the administrative column:
  + Mean: 4.07
  + Median: 2
  + Skewness: 1.96
  + Kurtosis: 3.47
* The mode is:

mode(df$TrafficType)

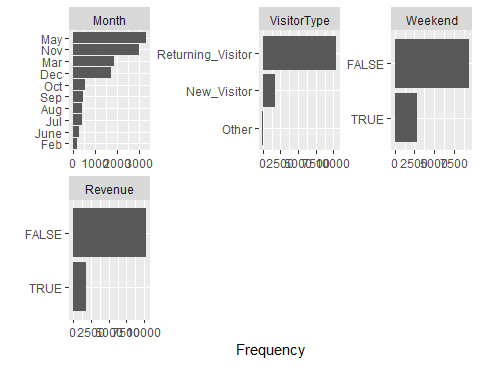
## [1] 2

### Distributions

plot\_histogram(df)



plot\_bar(df)

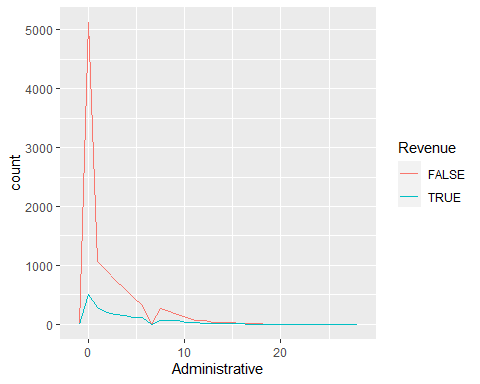


## Bivariate Analysis

* Examining how different variables affect the target variable

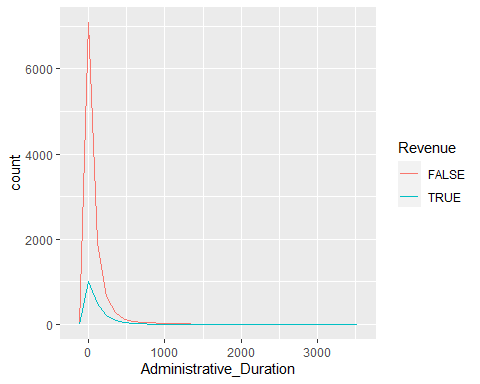
# Administrative sites and Revenue  
ggplot(df, aes(Administrative, color=Revenue)) +  
 geom\_freqpoly()

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



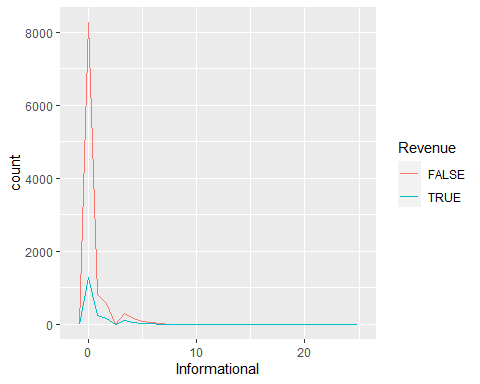
ggplot(df, aes(Administrative\_Duration, color=Revenue)) +  
 geom\_freqpoly()

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



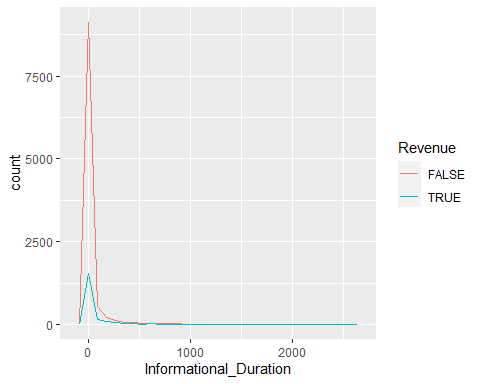
ggplot(df, aes(Informational, color=Revenue)) +  
 geom\_freqpoly()

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



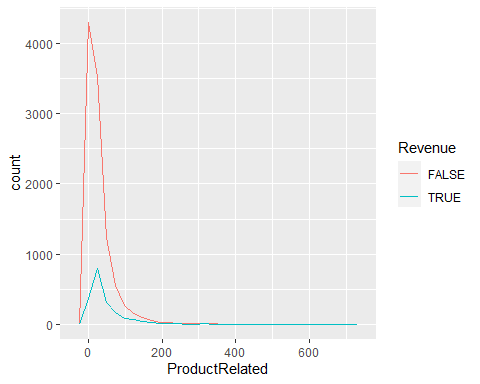
ggplot(df, aes(Informational\_Duration, color=Revenue)) +  
 geom\_freqpoly()

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



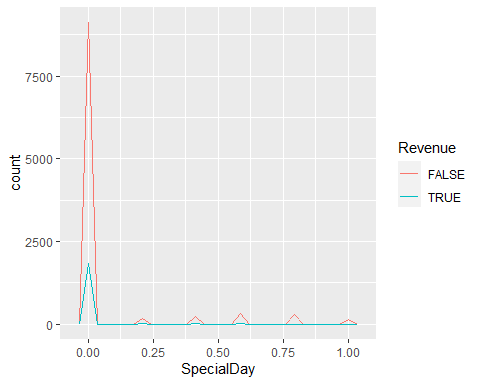
ggplot(df, aes(ProductRelated, color=Revenue)) +  
 geom\_freqpoly()

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



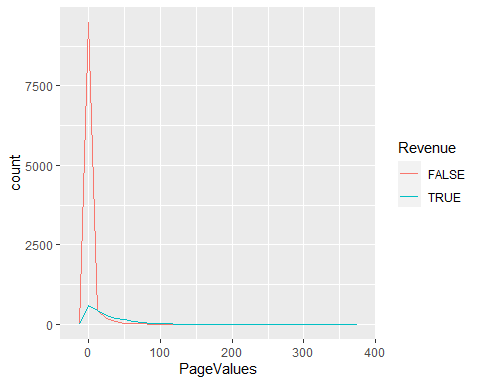
ggplot(df, aes(SpecialDay, color=Revenue)) +  
 geom\_freqpoly()

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

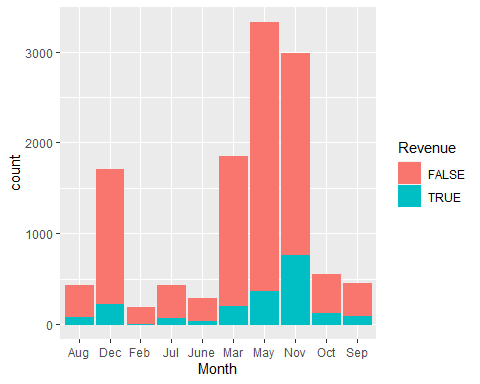


ggplot(df, aes(PageValues, color=Revenue)) +  
 geom\_freqpoly()

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



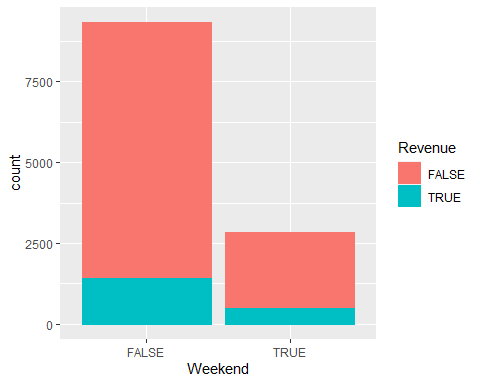
# Months vs GeneratingRevenue  
ggplot(df, aes(Month, color=Revenue, fill=Revenue)) +  
 geom\_bar()



* May, March, and November are the months which generate significantly more revenue for the business.

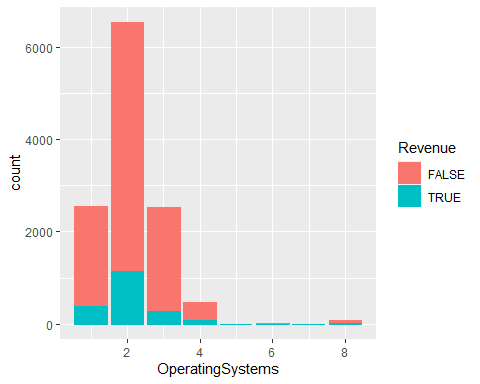
# Day type vs Generating Revenue  
ggplot(df, aes(Weekend, color=Revenue, fill=Revenue)) +  
 geom\_bar(binwidth=1)

## Warning: Ignoring unknown parameters: binwidth



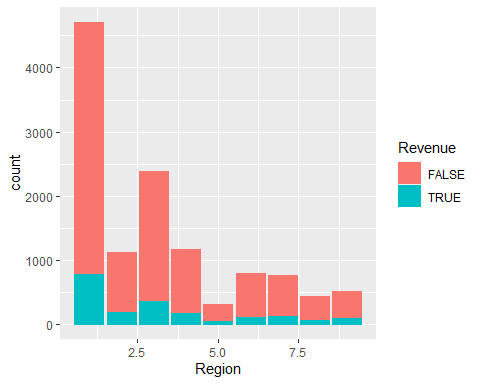
* Weekdays generate more Revenue than weekends.

# Operating systems vs Generating Revenue  
ggplot(df, aes(OperatingSystems, color=Revenue, fill=Revenue)) +  
 geom\_bar()



* Users of type 2 OS generated the most revenue for the site, while 1, and 3 followed.

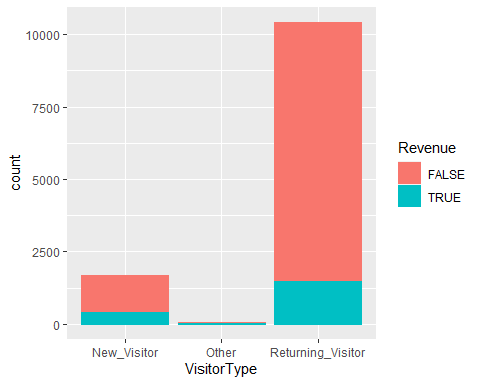
ggplot(df, aes(Region, fill=Revenue, color=Revenue)) +  
 geom\_bar()



* Region 1 produced the most revenue out of all the others with region 5 producing the least.

# Visitor type and revenue  
ggplot(df, aes(VisitorType, color=Revenue, fill=Revenue)) +  
 geom\_bar(binwidth=2)

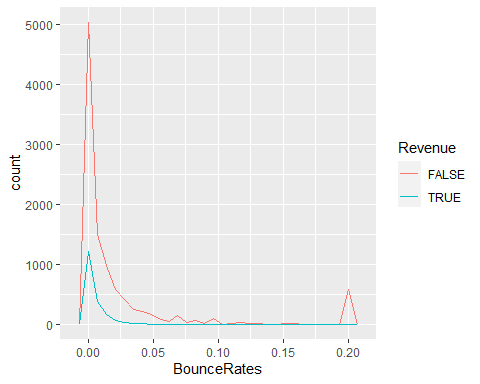
## Warning: Ignoring unknown parameters: binwidth



* Returning visitors generated more revenue than new ones

# Bounce rates vs Revenue  
ggplot(df, aes(BounceRates, color=Revenue)) +  
 geom\_freqpoly()

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



* A lot of sites had a high percentage of visitors just leaving without triggering any requests from our target website.
* All the data profiling statistics will be organized into the report below

create\_report(df)

##   
##   
## processing file: report.rmd

## | | | 0% | |.. | 2%  
## inline R code fragments  
##   
## | |... | 5%  
## label: global\_options (with options)   
## List of 1  
## $ include: logi FALSE  
##   
## | |..... | 7%  
## ordinary text without R code  
##   
## | |....... | 10%  
## label: introduce  
## | |........ | 12%  
## ordinary text without R code  
##   
## | |.......... | 14%  
## label: plot\_intro

## | |............ | 17%  
## ordinary text without R code  
##   
## | |............. | 19%  
## label: data\_structure  
## | |............... | 21%  
## ordinary text without R code  
##   
## | |................. | 24%  
## label: missing\_profile

## | |.................. | 26%  
## ordinary text without R code  
##   
## | |.................... | 29%  
## label: univariate\_distribution\_header  
## | |...................... | 31%  
## ordinary text without R code  
##   
## | |....................... | 33%  
## label: plot\_histogram

## | |......................... | 36%  
## ordinary text without R code  
##   
## | |........................... | 38%  
## label: plot\_density  
## | |............................ | 40%  
## ordinary text without R code  
##   
## | |.............................. | 43%  
## label: plot\_frequency\_bar

## | |................................ | 45%  
## ordinary text without R code  
##   
## | |................................. | 48%  
## label: plot\_response\_bar  
## | |................................... | 50%  
## ordinary text without R code  
##   
## | |..................................... | 52%  
## label: plot\_with\_bar  
## | |...................................... | 55%  
## ordinary text without R code  
##   
## | |........................................ | 57%  
## label: plot\_normal\_qq

## | |.......................................... | 60%  
## ordinary text without R code  
##   
## | |........................................... | 62%  
## label: plot\_response\_qq  
## | |............................................. | 64%  
## ordinary text without R code  
##   
## | |............................................... | 67%  
## label: plot\_by\_qq  
## | |................................................ | 69%  
## ordinary text without R code  
##   
## | |.................................................. | 71%  
## label: correlation\_analysis

## | |.................................................... | 74%  
## ordinary text without R code  
##   
## | |..................................................... | 76%  
## label: principal\_component\_analysis

## | |....................................................... | 79%  
## ordinary text without R code  
##   
## | |......................................................... | 81%  
## label: bivariate\_distribution\_header  
## | |.......................................................... | 83%  
## ordinary text without R code  
##   
## | |............................................................ | 86%  
## label: plot\_response\_boxplot  
## | |.............................................................. | 88%  
## ordinary text without R code  
##   
## | |............................................................... | 90%  
## label: plot\_by\_boxplot  
## | |................................................................. | 93%  
## ordinary text without R code  
##   
## | |................................................................... | 95%  
## label: plot\_response\_scatterplot  
## | |.................................................................... | 98%  
## ordinary text without R code  
##   
## | |......................................................................| 100%  
## label: plot\_by\_scatterplot

## output file: C:/Users/user/Documents/IP\_W13\_Part 2/report.knit.md

## "C:/Program Files/RStudio/bin/quarto/bin/pandoc" +RTS -K512m -RTS "C:/Users/user/Documents/IP\_W13\_Part 2/report.knit.md" --to html4 --from markdown+autolink\_bare\_uris+tex\_math\_single\_backslash --output pandoc1bd063556c75.html --lua-filter "C:\Users\user\Documents\R\win-library\4.1\rmarkdown\rmarkdown\lua\pagebreak.lua" --lua-filter "C:\Users\user\Documents\R\win-library\4.1\rmarkdown\rmarkdown\lua\latex-div.lua" --self-contained --variable bs3=TRUE --standalone --section-divs --table-of-contents --toc-depth 6 --template "C:\Users\user\Documents\R\win-library\4.1\rmarkdown\rmd\h\default.html" --no-highlight --variable highlightjs=1 --variable theme=yeti --mathjax --variable "mathjax-url=https://mathjax.rstudio.com/latest/MathJax.js?config=TeX-AMS-MML\_HTMLorMML" --include-in-header "C:\Users\user\AppData\Local\Temp\RtmpOgPEGU\rmarkdown-str1bd06ee93b45.html"

##   
## Output created: report.html

* The link for the report is here: “<https://github.com/Geoffrey-Chege/Supervised-and-Unsupervised-Learning/blob/main/Customer%20Analysis/report.html>”

# 8. Implementing the Solution

## K-Means Clustering

* Step 1: One hot encoding of the factor variables.

# # One hot encoding of the factor variables.  
  
dmy = dummyVars(" ~ .", data = df)  
  
df2 = data.frame(predict(dmy, newdata = df))

# Checking the data types of each attribute  
sapply(df2, class)

## Administrative Administrative\_Duration   
## "numeric" "numeric"   
## Informational Informational\_Duration   
## "numeric" "numeric"   
## ProductRelated ProductRelated\_Duration   
## "numeric" "numeric"   
## BounceRates ExitRates   
## "numeric" "numeric"   
## PageValues SpecialDay   
## "numeric" "numeric"   
## Month.Aug Month.Dec   
## "numeric" "numeric"   
## Month.Feb Month.Jul   
## "numeric" "numeric"   
## Month.June Month.Mar   
## "numeric" "numeric"   
## Month.May Month.Nov   
## "numeric" "numeric"   
## Month.Oct Month.Sep   
## "numeric" "numeric"   
## OperatingSystems Browser   
## "numeric" "numeric"   
## Region TrafficType   
## "numeric" "numeric"   
## VisitorType.New\_Visitor VisitorType.Other   
## "numeric" "numeric"   
## VisitorType.Returning\_Visitor Weekend.FALSE   
## "numeric" "numeric"   
## Weekend.TRUE Revenue.FALSE   
## "numeric" "numeric"   
## Revenue.TRUE   
## "numeric"

* Step 2: We are instructed to use Revenue as the class label, Hence we will remove it and store it in another variable.

# Step 2  
# We are instructed to use Revenue as the class label,  
# Hence we will remove it and store it in another variable  
  
df2\_copy <- df2[, -c(30:31)]  
df.class<- df[, "Revenue"]  
  
df2\_copy\_copy <- df2[, -c(30,31)]

# Previewing the copy dataset with dummies  
head(df2\_copy)

## Administrative Administrative\_Duration Informational Informational\_Duration  
## 1 0 0 0 0  
## 2 0 0 0 0  
## 3 0 -1 0 -1  
## 4 0 0 0 0  
## 5 0 0 0 0  
## 6 0 0 0 0  
## ProductRelated ProductRelated\_Duration BounceRates ExitRates PageValues  
## 1 1 0.000000 0.20000000 0.2000000 0  
## 2 2 64.000000 0.00000000 0.1000000 0  
## 3 1 -1.000000 0.20000000 0.2000000 0  
## 4 2 2.666667 0.05000000 0.1400000 0  
## 5 10 627.500000 0.02000000 0.0500000 0  
## 6 19 154.216667 0.01578947 0.0245614 0  
## SpecialDay Month.Aug Month.Dec Month.Feb Month.Jul Month.June Month.Mar  
## 1 0 0 0 1 0 0 0  
## 2 0 0 0 1 0 0 0  
## 3 0 0 0 1 0 0 0  
## 4 0 0 0 1 0 0 0  
## 5 0 0 0 1 0 0 0  
## 6 0 0 0 1 0 0 0  
## Month.May Month.Nov Month.Oct Month.Sep OperatingSystems Browser Region  
## 1 0 0 0 0 1 1 1  
## 2 0 0 0 0 2 2 1  
## 3 0 0 0 0 4 1 9  
## 4 0 0 0 0 3 2 2  
## 5 0 0 0 0 3 3 1  
## 6 0 0 0 0 2 2 1  
## TrafficType VisitorType.New\_Visitor VisitorType.Other  
## 1 1 0 0  
## 2 2 0 0  
## 3 3 0 0  
## 4 4 0 0  
## 5 4 0 0  
## 6 3 0 0  
## VisitorType.Returning\_Visitor Weekend.FALSE Weekend.TRUE  
## 1 1 1 0  
## 2 1 1 0  
## 3 1 1 0  
## 4 1 1 0  
## 5 1 0 1  
## 6 1 1 0

* Step 3: Determining whether to Normalize or Scale the data.

### Scaling:

# This is important to ensure that no particular attribute, has more impact on clustering algorithm than others  
  
df2\_scaled <- scale(df2\_copy)

# After scaling the data lets see what we find in the output  
summary(df2\_scaled)

## Administrative Administrative\_Duration Informational   
## Min. :-0.7025 Min. :-0.46574 Min. :-0.3988   
## 1st Qu.:-0.7025 1st Qu.:-0.46011 1st Qu.:-0.3988   
## Median :-0.4023 Median :-0.40941 Median :-0.3988   
## Mean : 0.0000 Mean : 0.00000 Mean : 0.0000   
## 3rd Qu.: 0.4984 3rd Qu.: 0.07361 3rd Qu.:-0.3988   
## Max. : 7.4035 Max. :18.68474 Max. :18.4127   
## Informational\_Duration ProductRelated ProductRelated\_Duration  
## Min. :-0.2533 Min. :-0.7188 Min. :-0.6295   
## 1st Qu.:-0.2463 1st Qu.:-0.5394 1st Qu.:-0.5281   
## Median :-0.2463 Median :-0.3152 Median :-0.3115   
## Mean : 0.0000 Mean : 0.0000 Mean : 0.0000   
## 3rd Qu.:-0.2463 3rd Qu.: 0.1332 3rd Qu.: 0.1407   
## Max. :17.7758 Max. :15.0881 Max. :32.6919   
## BounceRates ExitRates PageValues SpecialDay   
## Min. :-0.45034 Min. :-0.8973 Min. :-0.319 Min. :-0.3103   
## 1st Qu.:-0.45034 1st Qu.:-0.5897 1st Qu.:-0.319 1st Qu.:-0.3103   
## Median :-0.38580 Median :-0.3567 Median :-0.319 Median :-0.3103   
## Mean : 0.00000 Mean : 0.0000 Mean : 0.000 Mean : 0.0000   
## 3rd Qu.:-0.08326 3rd Qu.: 0.1511 3rd Qu.:-0.319 3rd Qu.:-0.3103   
## Max. : 3.95470 Max. : 3.4273 Max. :19.070 Max. : 4.6969   
## Month.Aug Month.Dec Month.Feb Month.Jul   
## Min. :-0.1918 Min. :-0.4032 Min. :-0.1231 Min. :-0.1916   
## 1st Qu.:-0.1918 1st Qu.:-0.4032 1st Qu.:-0.1231 1st Qu.:-0.1916   
## Median :-0.1918 Median :-0.4032 Median :-0.1231 Median :-0.1916   
## Mean : 0.0000 Mean : 0.0000 Mean : 0.0000 Mean : 0.0000   
## 3rd Qu.:-0.1918 3rd Qu.:-0.4032 3rd Qu.:-0.1231 3rd Qu.:-0.1916   
## Max. : 5.2126 Max. : 2.4799 Max. : 8.1254 Max. : 5.2188   
## Month.June Month.Mar Month.May Month.Nov   
## Min. :-0.1547 Min. :-0.4232 Min. :-0.6125 Min. :-0.5689   
## 1st Qu.:-0.1547 1st Qu.:-0.4232 1st Qu.:-0.6125 1st Qu.:-0.5689   
## Median :-0.1547 Median :-0.4232 Median :-0.6125 Median :-0.5689   
## Mean : 0.0000 Mean : 0.0000 Mean : 0.0000 Mean : 0.0000   
## 3rd Qu.:-0.1547 3rd Qu.:-0.4232 3rd Qu.: 1.6326 3rd Qu.:-0.5689   
## Max. : 6.4653 Max. : 2.3628 Max. : 1.6326 Max. : 1.7576   
## Month.Oct Month.Sep OperatingSystems Browser   
## Min. :-0.2171 Min. :-0.1952 Min. :-1.2397 Min. :-0.7940   
## 1st Qu.:-0.2171 1st Qu.:-0.1952 1st Qu.:-0.1371 1st Qu.:-0.2094   
## Median :-0.2171 Median :-0.1952 Median :-0.1371 Median :-0.2094   
## Mean : 0.0000 Mean : 0.0000 Mean : 0.0000 Mean : 0.0000   
## 3rd Qu.:-0.2171 3rd Qu.:-0.1952 3rd Qu.: 0.9654 3rd Qu.:-0.2094   
## Max. : 4.6064 Max. : 5.1213 Max. : 6.4782 Max. : 6.2212   
## Region TrafficType VisitorType.New\_Visitor  
## Min. :-0.89629 Min. :-0.76562 Min. :-0.4014   
## 1st Qu.:-0.89629 1st Qu.:-0.51661 1st Qu.:-0.4014   
## Median :-0.06381 Median :-0.51661 Median :-0.4014   
## Mean : 0.00000 Mean : 0.00000 Mean : 0.0000   
## 3rd Qu.: 0.35244 3rd Qu.:-0.01858 3rd Qu.:-0.4014   
## Max. : 2.43366 Max. : 3.96567 Max. : 2.4910   
## VisitorType.Other VisitorType.Returning\_Visitor Weekend.FALSE   
## Min. :-0.08175 Min. :-2.4241 Min. :-1.8086   
## 1st Qu.:-0.08175 1st Qu.: 0.4125 1st Qu.: 0.5529   
## Median :-0.08175 Median : 0.4125 Median : 0.5529   
## Mean : 0.00000 Mean : 0.0000 Mean : 0.0000   
## 3rd Qu.:-0.08175 3rd Qu.: 0.4125 3rd Qu.: 0.5529   
## Max. :12.23081 Max. : 0.4125 Max. : 0.5529   
## Weekend.TRUE   
## Min. :-0.5529   
## 1st Qu.:-0.5529   
## Median :-0.5529   
## Mean : 0.0000   
## 3rd Qu.:-0.5529   
## Max. : 1.8086

* It is evident that there are some attributes still with large values compared to others.
* Scaling makes the data changes the data to have a mean 0.
* We will normalize the data and see if we get different results.

### Normalizing:

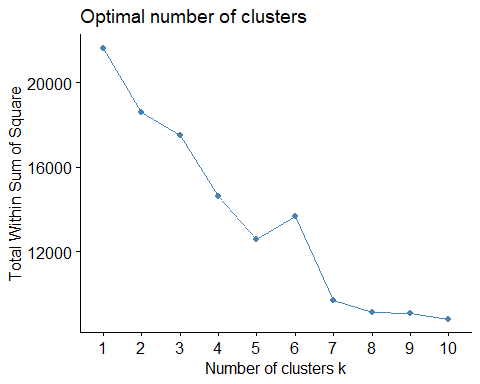
# Normalizing the a copy of the original data  
  
df2\_norm <- as.data.frame(apply(df2\_copy, 2, function(x) (x - min(x))/(max(x)-min(x))))

# summary of the normalized data.  
summary(df2\_norm)

## Administrative Administrative\_Duration Informational   
## Min. :0.00000 Min. :0.0000000 Min. :0.0000   
## 1st Qu.:0.00000 1st Qu.:0.0002941 1st Qu.:0.0000   
## Median :0.03704 Median :0.0029414 Median :0.0000   
## Mean :0.08667 Mean :0.0243201 Mean :0.0212   
## 3rd Qu.:0.14815 3rd Qu.:0.0281638 3rd Qu.:0.0000   
## Max. :1.00000 Max. :1.0000000 Max. :1.0000   
## Informational\_Duration ProductRelated ProductRelated\_Duration  
## Min. :0.0000000 Min. :0.00000 Min. :0.000000   
## 1st Qu.:0.0003921 1st Qu.:0.01135 1st Qu.:0.003042   
## Median :0.0003921 Median :0.02553 Median :0.009543   
## Mean :0.0140518 Mean :0.04547 Mean :0.018891   
## 3rd Qu.:0.0003921 3rd Qu.:0.05390 3rd Qu.:0.023112   
## Max. :1.0000000 Max. :1.00000 Max. :1.000000   
## BounceRates ExitRates PageValues SpecialDay   
## Min. :0.00000 Min. :0.00000 Min. :0.00000 Min. :0.00000   
## 1st Qu.:0.00000 1st Qu.:0.07111 1st Qu.:0.00000 1st Qu.:0.00000   
## Median :0.01465 Median :0.12500 Median :0.00000 Median :0.00000   
## Mean :0.10223 Mean :0.20748 Mean :0.01645 Mean :0.06197   
## 3rd Qu.:0.08333 3rd Qu.:0.24242 3rd Qu.:0.00000 3rd Qu.:0.00000   
## Max. :1.00000 Max. :1.00000 Max. :1.00000 Max. :1.00000   
## Month.Aug Month.Dec Month.Feb Month.Jul   
## Min. :0.00000 Min. :0.0000 Min. :0.00000 Min. :0.00000   
## 1st Qu.:0.00000 1st Qu.:0.0000 1st Qu.:0.00000 1st Qu.:0.00000   
## Median :0.00000 Median :0.0000 Median :0.00000 Median :0.00000   
## Mean :0.03549 Mean :0.1398 Mean :0.01492 Mean :0.03541   
## 3rd Qu.:0.00000 3rd Qu.:0.0000 3rd Qu.:0.00000 3rd Qu.:0.00000   
## Max. :1.00000 Max. :1.0000 Max. :1.00000 Max. :1.00000   
## Month.June Month.Mar Month.May Month.Nov   
## Min. :0.00000 Min. :0.0000 Min. :0.0000 Min. :0.0000   
## 1st Qu.:0.00000 1st Qu.:0.0000 1st Qu.:0.0000 1st Qu.:0.0000   
## Median :0.00000 Median :0.0000 Median :0.0000 Median :0.0000   
## Mean :0.02336 Mean :0.1519 Mean :0.2728 Mean :0.2445   
## 3rd Qu.:0.00000 3rd Qu.:0.0000 3rd Qu.:1.0000 3rd Qu.:0.0000   
## Max. :1.00000 Max. :1.0000 Max. :1.0000 Max. :1.0000   
## Month.Oct Month.Sep OperatingSystems Browser   
## Min. :0.000 Min. :0.00000 Min. :0.0000 Min. :0.00000   
## 1st Qu.:0.000 1st Qu.:0.00000 1st Qu.:0.1429 1st Qu.:0.08333   
## Median :0.000 Median :0.00000 Median :0.1429 Median :0.08333   
## Mean :0.045 Mean :0.03672 Mean :0.1606 Mean :0.11318   
## 3rd Qu.:0.000 3rd Qu.:0.00000 3rd Qu.:0.2857 3rd Qu.:0.08333   
## Max. :1.000 Max. :1.00000 Max. :1.0000 Max. :1.00000   
## Region TrafficType VisitorType.New\_Visitor VisitorType.Other  
## Min. :0.0000 Min. :0.00000 Min. :0.0000 Min. :0.00000   
## 1st Qu.:0.0000 1st Qu.:0.05263 1st Qu.:0.0000 1st Qu.:0.00000   
## Median :0.2500 Median :0.05263 Median :0.0000 Median :0.00000   
## Mean :0.2692 Mean :0.16182 Mean :0.1388 Mean :0.00664   
## 3rd Qu.:0.3750 3rd Qu.:0.15789 3rd Qu.:0.0000 3rd Qu.:0.00000   
## Max. :1.0000 Max. :1.00000 Max. :1.0000 Max. :1.00000   
## VisitorType.Returning\_Visitor Weekend.FALSE Weekend.TRUE   
## Min. :0.0000 Min. :0.0000 Min. :0.0000   
## 1st Qu.:1.0000 1st Qu.:1.0000 1st Qu.:0.0000   
## Median :1.0000 Median :1.0000 Median :0.0000   
## Mean :0.8546 Mean :0.7659 Mean :0.2341   
## 3rd Qu.:1.0000 3rd Qu.:1.0000 3rd Qu.:0.0000   
## Max. :1.0000 Max. :1.0000 Max. :1.0000

* Here, we have a maximum value of 1 and minimum value of 0s and mean of close to zero in all attributes.
* We will use the NORMALIZED dataset for clustering.
* Step 4: Determining optimal k value.

# finding optimum k  
fviz\_nbclust(df2\_norm, kmeans, method="wss")



* 3 is the first elbow, so I will use it as my k value.
* Step 5: Applying K-Means.

# Applying K-Means Clustering algorithm  
# Using 3 centroids as K=3  
  
result <- kmeans(df2\_norm, 3)

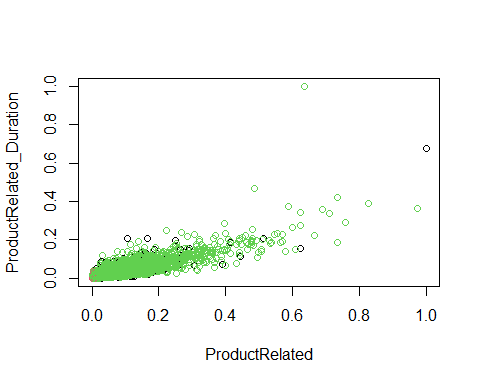
# Previewing the number of records in each cluster  
  
result$size

## [1] 3122 745 8332

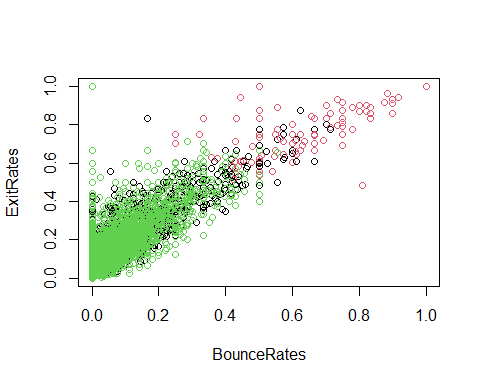
# Viewing the cluster center datapoints by each attribute  
  
result$centers

## Administrative Administrative\_Duration Informational Informational\_Duration  
## 1 0.078297388 0.0223105019 0.0189915652 0.0118682058  
## 2 0.001541138 0.0006645968 0.0005592841 0.0003943097  
## 3 0.097415586 0.0271881865 0.0238738198 0.0160911544  
## ProductRelated ProductRelated\_Duration BounceRates ExitRates PageValues  
## 1 0.040246069 0.0165335824 0.06973702 0.1871326 0.01617796  
## 2 0.003484221 0.0006916973 0.90538300 0.9448789 0.00000000  
## 3 0.051185892 0.0214008125 0.04259713 0.1491759 0.01802882  
## SpecialDay Month.Aug Month.Dec Month.Feb Month.Jul Month.June Month.Mar  
## 1 0.214477899 0.00000000 0.0000000 0.00000000 0.00000000 0.00000000 0.0000000  
## 2 0.069530201 0.02953020 0.1302013 0.05234899 0.04832215 0.05234899 0.1570470  
## 3 0.004152664 0.04932789 0.1931109 0.01716275 0.04752760 0.02952472 0.2083533  
## Month.May Month.Nov Month.Oct Month.Sep OperatingSystems Browser  
## 1 1.0000000 0.0000000 0.00000000 0.00000000 0.1603368 0.1154975  
## 2 0.2765101 0.2228188 0.01744966 0.01342282 0.1718121 0.1082774  
## 3 0.0000000 0.3380941 0.06433029 0.05256841 0.1597284 0.1127480  
## Region TrafficType VisitorType.New\_Visitor VisitorType.Other  
## 1 0.2666960 0.1791699 0.10025625 0.000000000  
## 2 0.2667785 0.2136348 0.03758389 0.016107383  
## 3 0.2702982 0.1506873 0.16226596 0.008281325  
## VisitorType.Returning\_Visitor Weekend.FALSE Weekend.TRUE  
## 1 0.8997438 0.7818706 0.2181294  
## 2 0.9463087 0.8214765 0.1785235  
## 3 0.8294527 0.7549208 0.2450792

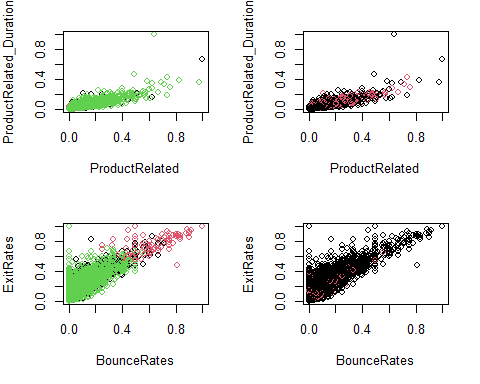
# Plotting two variables to see how their data points  
# have been distributed in the cluster  
# Product Related, vs Product Related Duration  
  
plot(df2\_norm[, 5:6], col = result$cluster)



# Product Related vs Product Related Duration  
  
plot(df2\_norm[, 7:8], col = result$cluster)



# Verifying the results of clustering  
# ---  
#   
par(mfrow = c(2,2), mar = c(5,4,2,2))  
  
# Plotting to see how Product Related vs Product Related Duration data points have been distributed in clusters  
plot(df2\_norm[, 5:6], col = result$cluster)  
  
# Plotting to see how Product Related, vs Product Related Duration data points have been distributed   
# originally as per "class" attribute in dataset  
# ---  
#  
plot(df2\_norm[, 5:6], col = df.class)  
  
# Plotting to see how Product Related vs Product Related Duration data points have been distributed in clusters  
# ---  
#   
plot(df2\_norm[, 7:8], col = result$cluster)  
plot(df2\_norm[, 7:8], col = df.class)



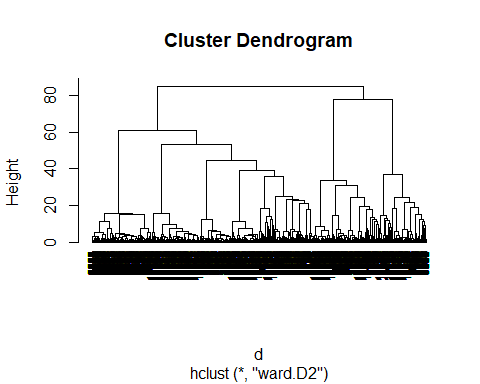
# Result of table shows that Cluster 1 corresponds to False,   
# Cluster 2 corresponds to False and Cluster 3 to False.  
# ---  
#   
table(result$cluster, df.class)

## df.class  
## FALSE TRUE  
## 1 2757 365  
## 2 742 3  
## 3 6792 1540

# 9. Challenging the solution

## Hierachical clustering

# We use R function hclust()  
# For hierarchical clustering  
  
# d will be the first argument in the hclust() dissimilairty matrix  
  
# First we use the dist() to compute the Euclidean distance btwn obs  
d <- dist(df2\_norm, method = "euclidean")  
  
# We then apply hierarchical clustering using the Ward's method  
res.hc <- hclust(d, method = "ward.D2")  
  
# Lastly we plot the obtained dendrogram  
plot(res.hc, cex = 0.6, hang = -1)



## DBSCAN

# Applying DBSCAN algorithm  
# ---  
# I want minimum 4 points with in a distance of eps(0.4)  
#   
db<-dbscan(df2\_norm,eps=0.4,MinPts = 4)

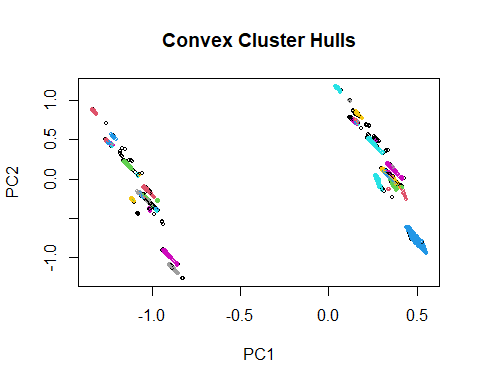
## Warning in dbscan(df2\_norm, eps = 0.4, MinPts = 4): converting argument MinPts  
## (fpc) to minPts (dbscan)!

# Printing out the clustering results  
# ---  
#   
print(db)

## DBSCAN clustering for 12199 objects.  
## Parameters: eps = 0.4, minPts = 4  
## The clustering contains 63 cluster(s) and 422 noise points.  
##   
## 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15   
## 422 26 122 8 5 4 1225 363 138 87 16 2354 217 479 70 126   
## 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31   
## 4 5 4 303 23 79 165 261 60 125 624 87 1856 250 46 70   
## 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47   
## 272 84 59 36 269 24 20 26 5 10 8 38 6 5 8 21   
## 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63   
## 4 4 6 4 6 1007 249 4 40 255 16 63 13 4 4 5   
##   
## Available fields: cluster, eps, minPts

# We also plot our clusters as shown  
# ---  
# The dataset and cluster method of dbscan is used to plot the clusters.  
#   
hullplot(df2\_norm,db$cluster)

## Warning in hullplot(df2\_norm, db$cluster): Not enough colors. Some colors will  
## be reused.



* The DBSCAN and Hierarchical Clustering approaches are difficult to interpret given the nature of the data.
* K-Means is the easiest to understand.

# 10. Conclusion

* Most traffic and revenue was from region 1. During holidays, more regions visit the site and contribute significantly to the total revenue.
* Traffic type 2 brought in the most visitors. Some of the traffic types did not bring in visitors for all the 10 months under analysis.They should be eliminated when considering advertisement or re evaluated to find out the problem.
* Most of the revenue and visits was from return visitors. A good indicator of customer satisfaction.
* Return customers are the main source of revenue.
* Bounce rate is high especial for new customers.