Kira Plastinina Marketing and Advertising in R

Dennis Kiarie

9/3/2021

***1. Define the Question***

***1.1 Research Question***

Our Research seeks to to understand the Kira Plastinina customer’s behavior from the data collected over the past years.

***1.2 Metric of Success***

To learn the characteristics of customer groups.

***1.3 The Context***

Kira Plastinina (Links to an external site.) 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.

***1.4 Experimental Design***

1. Problem Definition
2. Data Sourcing
3. Check the Data
4. Perform Data Cleaning
5. Perform Exploratory Data Analysis (Univariate, Bivariate & Multivariate)
6. Implement the Solution
7. Challenge the Solution
8. Follow up Questions

***1.5 Data Relevance***

The data provided was collected over the past few year hence its appropriate for our analysis. The dataset for this analysis can be found in this link:[ <http://bit.ly/EcommerceCustomersDataset> ]

**Description**

The dataset consists of 10 numerical and 8 categorical attributes.

The ‘Revenue’ attribute can be used as the class label.

**Types of Pages: Administrative, Informational**

**Time spent on pages: Admin Duration and Info Duration**

“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.

**Metrics: Bounce rate, Exit rate and Page Value**

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.

**Type of days: Speical or Ordinary**

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.

**Type of visit, Operating system, Browser and region(location)**

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.

library(tinytex)

***2.Data Preparation***

## Importing libraries  
#---  
#  
library(pacman)  
pacman :: p\_load(pacman,ggbiplot,plyr, dplyr,scales, readr, grid,factoextra, GGally,DataExplorer, ggplot2,moments,corrplot, ggthemes, ggvis, httr, lubridate, plotly, rio , rmarkdown, shiny, stringr,magrittr, tidyr,caret,factoextra)  
theme\_set(theme\_classic())  
options(warn = -1)

## Loading the data from a csv file  
#---  
#  
df <- read.csv("C:/Users/Denoo/Downloads/online\_shoppers\_intention.csv", na.strings = "")  
  
##preview the first 6 rows  
#---  
#  
head(df)

## 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

##preview the last 6 records of the dataset  
#---  
#  
tail(df)

## Administrative Administrative\_Duration Informational  
## 12325 0 0 1  
## 12326 3 145 0  
## 12327 0 0 0  
## 12328 0 0 0  
## 12329 4 75 0  
## 12330 0 0 0  
## Informational\_Duration ProductRelated ProductRelated\_Duration BounceRates  
## 12325 0 16 503.000 0.000000000  
## 12326 0 53 1783.792 0.007142857  
## 12327 0 5 465.750 0.000000000  
## 12328 0 6 184.250 0.083333333  
## 12329 0 15 346.000 0.000000000  
## 12330 0 3 21.250 0.000000000  
## ExitRates PageValues SpecialDay Month OperatingSystems Browser Region  
## 12325 0.03764706 0.00000 0 Nov 2 2 1  
## 12326 0.02903061 12.24172 0 Dec 4 6 1  
## 12327 0.02133333 0.00000 0 Nov 3 2 1  
## 12328 0.08666667 0.00000 0 Nov 3 2 1  
## 12329 0.02105263 0.00000 0 Nov 2 2 3  
## 12330 0.06666667 0.00000 0 Nov 3 2 1  
## TrafficType VisitorType Weekend Revenue  
## 12325 1 Returning\_Visitor FALSE FALSE  
## 12326 1 Returning\_Visitor TRUE FALSE  
## 12327 8 Returning\_Visitor TRUE FALSE  
## 12328 13 Returning\_Visitor TRUE FALSE  
## 12329 11 Returning\_Visitor FALSE FALSE  
## 12330 2 New\_Visitor TRUE FALSE

***3. Checking the data***

## we check for the number of rows and columns  
#---  
#  
cat("Rows:", nrow(df), "\nCols:", ncol(df))

## Rows: 12330   
## Cols: 18

##we check the type  
#---  
#  
class(df)

## [1] "data.frame"

##we check if datatypes are appropriate  
#---  
#  
glimpse(df)

## Rows: 12,330  
## Columns: 18  
## $ Administrative <int> 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 2~  
## $ Administrative\_Duration <dbl> 0, 0, -1, 0, 0, 0, -1, -1, 0, 0, 0, 0, 0, 0, 0~  
## $ Informational <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~  
## $ Informational\_Duration <dbl> 0, 0, -1, 0, 0, 0, -1, -1, 0, 0, 0, 0, 0, 0, 0~  
## $ ProductRelated <int> 1, 2, 1, 2, 10, 19, 1, 1, 2, 3, 3, 16, 7, 6, 2~  
## $ ProductRelated\_Duration <dbl> 0.000000, 64.000000, -1.000000, 2.666667, 627.~  
## $ BounceRates <dbl> 0.200000000, 0.000000000, 0.200000000, 0.05000~  
## $ ExitRates <dbl> 0.200000000, 0.100000000, 0.200000000, 0.14000~  
## $ PageValues <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~  
## $ SpecialDay <dbl> 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.4, 0.0, 0.8, 0~  
## $ Month <chr> "Feb", "Feb", "Feb", "Feb", "Feb", "Feb", "Feb~  
## $ OperatingSystems <int> 1, 2, 4, 3, 3, 2, 2, 1, 2, 2, 1, 1, 1, 2, 3, 1~  
## $ Browser <int> 1, 2, 1, 2, 3, 2, 4, 2, 2, 4, 1, 1, 1, 5, 2, 1~  
## $ Region <int> 1, 1, 9, 2, 1, 1, 3, 1, 2, 1, 3, 4, 1, 1, 3, 9~  
## $ TrafficType <int> 1, 2, 3, 4, 4, 3, 3, 5, 3, 2, 3, 3, 3, 3, 3, 3~  
## $ VisitorType <chr> "Returning\_Visitor", "Returning\_Visitor", "Ret~  
## $ Weekend <lgl> FALSE, FALSE, FALSE, FALSE, TRUE, FALSE, FALSE~  
## $ Revenue <lgl> FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FALS~

##we check for the number of columns  
#---  
#  
length(df)

## [1] 18

#The dataframe has 18 columns.

##we check the column names for easier reference  
#---  
#  
colnames(df)

## [1] "Administrative" "Administrative\_Duration"  
## [3] "Informational" "Informational\_Duration"   
## [5] "ProductRelated" "ProductRelated\_Duration"  
## [7] "BounceRates" "ExitRates"   
## [9] "PageValues" "SpecialDay"   
## [11] "Month" "OperatingSystems"   
## [13] "Browser" "Region"   
## [15] "TrafficType" "VisitorType"   
## [17] "Weekend" "Revenue"

##we check for column data types  
#---  
#  
sapply(df, class)

## Administrative Administrative\_Duration Informational   
## "integer" "numeric" "integer"   
## Informational\_Duration ProductRelated ProductRelated\_Duration   
## "numeric" "integer" "numeric"   
## BounceRates ExitRates PageValues   
## "numeric" "numeric" "numeric"   
## SpecialDay Month OperatingSystems   
## "numeric" "character" "integer"   
## Browser Region TrafficType   
## "integer" "integer" "integer"   
## VisitorType Weekend Revenue   
## "character" "logical" "logical"

## we Check for unique characters  
sapply(df, function(x) length(unique(x)))

## Administrative Administrative\_Duration Informational   
## 28 3337 18   
## Informational\_Duration ProductRelated ProductRelated\_Duration   
## 1260 312 9553   
## BounceRates ExitRates PageValues   
## 1873 4778 2704   
## SpecialDay Month OperatingSystems   
## 6 10 8   
## Browser Region TrafficType   
## 13 9 20   
## VisitorType Weekend Revenue   
## 3 2 2

***4.Tidying the data***

##we change the column names to lowercase for easier manipulation  
#---  
#  
colnames(df) = tolower(colnames(df))  
colnames(df)

## [1] "administrative" "administrative\_duration"  
## [3] "informational" "informational\_duration"   
## [5] "productrelated" "productrelated\_duration"  
## [7] "bouncerates" "exitrates"   
## [9] "pagevalues" "specialday"   
## [11] "month" "operatingsystems"   
## [13] "browser" "region"   
## [15] "traffictype" "visitortype"   
## [17] "weekend" "revenue"

##we replace spaces in column names for easier manipulation  
#---  
  
names(df) = str\_replace\_all(names(df), c(' ' = '\_'))  
names(df)

## [1] "administrative" "administrative\_duration"  
## [3] "informational" "informational\_duration"   
## [5] "productrelated" "productrelated\_duration"  
## [7] "bouncerates" "exitrates"   
## [9] "pagevalues" "specialday"   
## [11] "month" "operatingsystems"   
## [13] "browser" "region"   
## [15] "traffictype" "visitortype"   
## [17] "weekend" "revenue"

##we check for missing values  
#---  
#  
sum(is.na(df))

## [1] 112

#There are 112 missing values

##we Check the sum of missing values per column  
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

## we check the column names containing missing observations  
#---  
#  
list\_na <- colnames(df)[ apply(df, 2, anyNA) ]  
list\_na

## [1] "administrative" "administrative\_duration"  
## [3] "informational" "informational\_duration"   
## [5] "productrelated" "productrelated\_duration"  
## [7] "bouncerates" "exitrates"

##we check for duplicates  
#---  
#  
anyDuplicated(df)

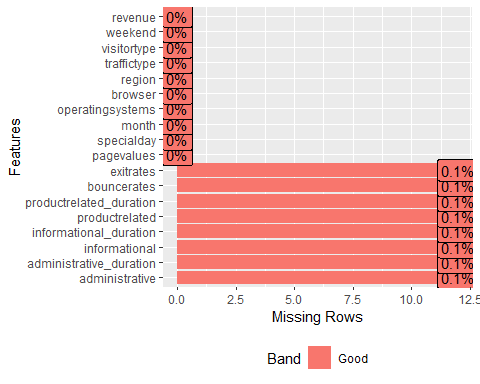
## [1] 159

#There are 159 duplicates

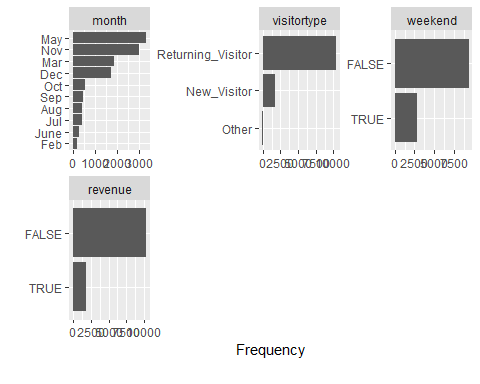
##we remove the duplicates  
df <- df[-which(duplicated(df)), ]  
dim(df)

## [1] 12211 18

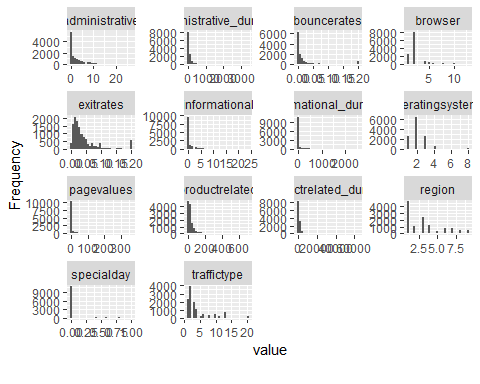
## missing values and the missing data profile  
#---  
#  
plot\_missing(df)



##Categorical frequency for discrete variables  
plot\_bar(df)



##we check for distribution of continuous variables  
#---  
#  
plot\_histogram(df)



plot\_str(df)

***5. Exploratory Data Analysis***

***5.1 Univariate Analysis***

***5.1.1 Administrative***

##we check for unique values in column administrative  
unique(df$administrative)

## [1] 0 1 2 4 12 3 10 6 5 9 8 16 13 11 7 18 14 17 19 15 NA 24 22 21 20  
## [26] 23 27 26

##we check the number of unique features in administrative column  
#---  
#  
factor(unique(df$administrative))

## [1] 0 1 2 4 12 3 10 6 5 9 8 16 13 11 7   
## [16] 18 14 17 19 15 <NA> 24 22 21 20 23 27 26   
## 27 Levels: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 ... 27

#There are 27 unique element in administrative column.

##we check for the number of records  
#---  
#  
length(df$administrative)

## [1] 12211

##we check for the dimension  
#---  
#  
dim(df)

## [1] 12211 18

##we check for sum of missing values  
#---  
#  
sum(is.na(df))

## [1] 96

##we drop missing values  
#---  
#  
df2 <- df[-which(is.na(df)), ]  
  
#preview sum of missing values in our new dataset  
sum(is.na(df2))

## [1] 0

##we check the dimensions of our new dataset  
#---  
#  
dim(df2)

## [1] 12199 18

##preview the columns of our new dataset  
colSums(is.na(df2))

## 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

#There are no missing values

summary(df2$administrative)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.00 0.00 1.00 2.34 4.00 27.00

adm <- df2$administrative

#we check the median  
median(adm)

## [1] 1

## we check the mode  
#---  
#  
Administrative\_x <- adm  
  
#sort(Daily.Internet.Usage\_x)  
names(table(Administrative\_x))[table(Administrative\_x)==max(table(Administrative\_x))]

## [1] "0"

##The variance  
#---  
var(Administrative\_x)

## [1] 11.09457

##The standard deviation  
#---  
#  
sd(Administrative\_x)

## [1] 3.330851

##The range  
#---  
#  
range(Administrative\_x)

## [1] 0 27

##The Interquatile range   
#---  
#  
IQR(Administrative\_x)

## [1] 4

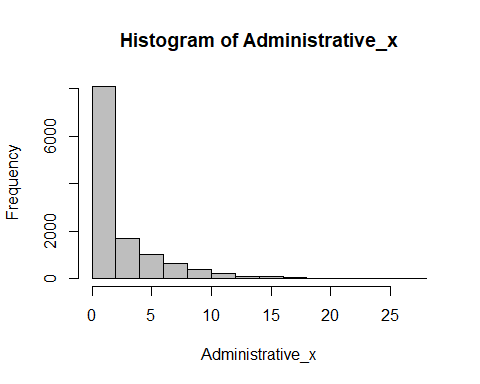
##The skewness of the column  
#---  
#  
skewness(Administrative\_x)

## [1] 1.946248

##The Kurtosis of the column  
#---  
#  
kurtosis(Administrative\_x)

## [1] 7.636106

##we check for the values appearing thrice  
  
hist(Administrative\_x, col=c("grey"))



***Conclusion***

From the variable Administrative\_X we observed the following:

1.The mean is 2.34.

2.The Highest value is 27.0 and minimum is 0.

3.The Variance of the column was 11.09457 with a standard deviation of 3.330851.

4.The admin distribution is right skewed

***5.1.2 Administrative\_Duration***

admin\_d<-(df2$administrative\_duration)

##we check for the unique values  
#---  
#  
length(unique(admin\_d))

## [1] 3336

summary(admin\_d)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## -1.00 0.00 9.00 81.68 94.75 3398.75

##we check the median  
#---  
#  
median(admin\_d)

## [1] 9

##we check the mode  
#---  
#  
names(table(admin\_d))[table(admin\_d)==max(table(admin\_d))]

## [1] "0"

##we check the variance  
var(admin\_d)

## [1] 31516.25

##we check the standard deviation  
#---  
#  
sd(admin\_d)

## [1] 177.5282

##we check the range  
#---  
#  
range(admin\_d)

## [1] -1.00 3398.75

##we check the interquatile range  
#---  
#  
IQR(admin\_d)

## [1] 94.75

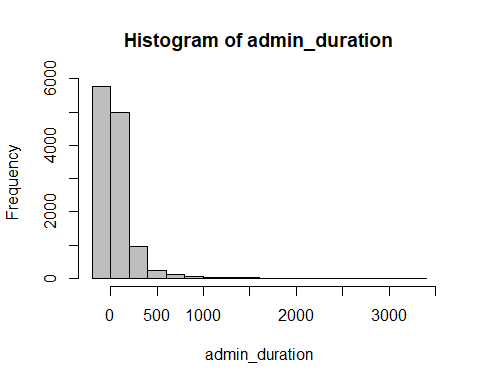
##we check the skewness  
#---  
#  
skewness(admin\_d)

## [1] 5.59021

##we check the kurtosis  
#---  
#  
kurtosis(admin\_d)

## [1] 53.09389

##we check the distribution  
#---  
#  
admin\_duration<-(df2$administrative\_duration)  
hist(admin\_duration, col=c("grey"))



***Conclusion***

From the variable Administrative\_Duration we observed the following:

1.The mean is 81.68.

2.The Highest value is 3398.75 and minimum is -1.0

3.The Variance of the column was 31516.25 with a standard deviation of 177.5282.

4.The Administrative\_Duration distribution is right skewed

***5.1.3 Informational***

info<- (df2$informational)

##we check for unique values  
#---  
#  
length(unique(info))

## [1] 17

#There are 17 unique values

summary(info)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.0000 0.0000 0.0000 0.5088 0.0000 24.0000

##we check the median  
median(info)

## [1] 0

##sort  
names(table(info))[table(info)==max(table(info ))]

## [1] "0"

##we check the variance  
#---  
#  
var(info)

## [1] 1.62771

##we check the standard deviation  
#---  
#  
sd(info)

## [1] 1.275817

##we check the range  
#---  
#  
range(info)

## [1] 0 24

##we check the interquatile range  
#---  
#  
IQR(info)

## [1] 0

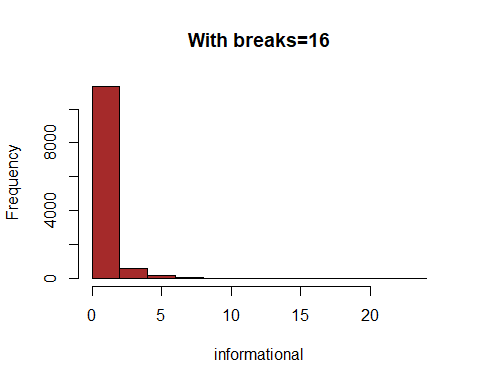
##we check the skewness  
#---  
#  
skewness(info)

## [1] 4.013451

##we check the kurtosis  
#---  
#  
kurtosis(info)

## [1] 29.64254

##we check the distribution  
#---  
#  
informational<- (df2$informational)  
hist(informational,breaks = 16 , main="With breaks=16", col=c("brown"))



***Conclusion***

From the variable Information we observed the following:

1.The mean is 0.5088.

2.The Highest value is 24.0000 and minimum is 0.0000

3.The Variance of the column was 1.62771 with a standard deviation of 1.275817.

4.The Information distribution is right skewed

***5.1.4 Informational\_Duration***

info\_d <- (df2$informational\_duration)

##we check for unique values  
#---  
#  
length(unique(info\_d))

## [1] 1259

summary(info\_d)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## -1.00 0.00 0.00 34.84 0.00 2549.38

##we check the median  
#---  
#  
median(info\_d)

## [1] 0

##sort  
#---  
#  
names(table(info\_d))[table(info\_d)==max(table(info\_d))]

## [1] "0"

##we check the variance  
#---  
#  
var(info\_d)

## [1] 20010.51

##we check the standard deviation  
#---  
#  
sd(info\_d)

## [1] 141.4585

##we check the range  
#---  
#  
range(info\_d)

## [1] -1.000 2549.375

##we check the IQR  
#---  
#  
IQR(info\_d)

## [1] 0

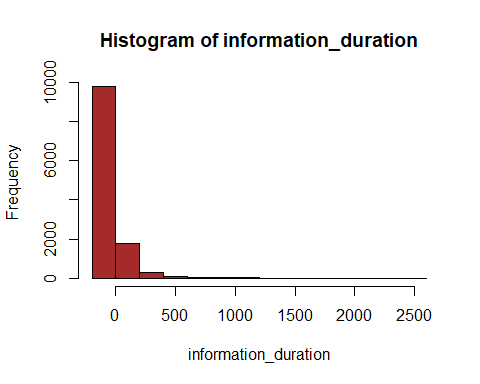
##we check the skewness  
#---  
#  
skewness(info\_d)

## [1] 7.537435

##we check the kurtosis  
#---  
#  
kurtosis(info\_d)

## [1] 78.46409

##Distribution  
#---  
#  
information\_duration <- (df2$informational\_duration)  
hist(information\_duration,col=c("brown"))



***1.5.5 Product Related***

product\_r<-(df2$productrelated)

##we check the unique values   
length(unique(product\_r))

## [1] 311

summary(product\_r)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.00 8.00 18.00 32.06 38.00 705.00

##we check the median  
#---  
#  
median(product\_r)

## [1] 18

#sort  
names(table(product\_r))[table(product\_r)==max(table(product\_r ))]

## [1] "1"

##we check the variance  
#---  
#  
var(product\_r)

## [1] 1989.241

##we check the standard deviation  
#---  
#  
sd(product\_r)

## [1] 44.60091

##we check the range  
#---  
#  
range(product\_r)

## [1] 0 705

##we check the IQR  
#---  
#  
IQR(product\_r)

## [1] 30

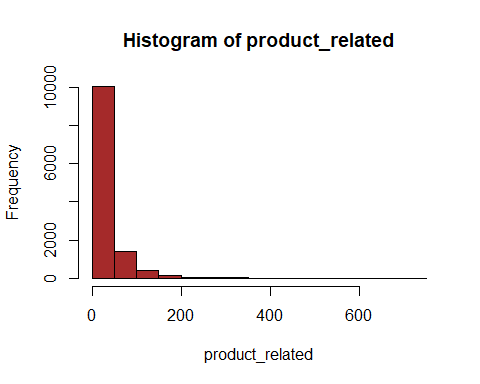
##we check the skewness  
#---  
#  
skewness(product\_r)

## [1] 4.332134

##we check the kurtosis  
#---  
#  
kurtosis(product\_r)

## [1] 34.04903

##we check for the distribution  
#---  
#  
product\_related<-(df2$productrelated)  
hist(product\_related,col=c("brown"))



***1.5.6 Product Related Duration***

product\_rd <- df2$Productrelated\_duration

##we check the unique values  
#---  
#  
length(unique(df2$productrelated\_duration))

## [1] 9552

summary(df2$productrelated\_duration)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## -1.0 193.6 609.5 1207.5 1477.6 63973.5

##we check the median  
#---  
#  
median(df2$productrelated\_duration)

## [1] 609.5417

##we check the variance  
#---  
#  
var(df2$productrelated\_duration)

## [1] 3686121

##we check the standard deviation  
#---  
#  
sd(df2$productrelated\_duration)

## [1] 1919.927

##we check the range  
#---  
#  
range(df2$productrelated\_duration)

## [1] -1.00 63973.52

##we check the interquatile range  
#---  
#  
IQR(df2$productrelated\_duration)

## [1] 1283.981

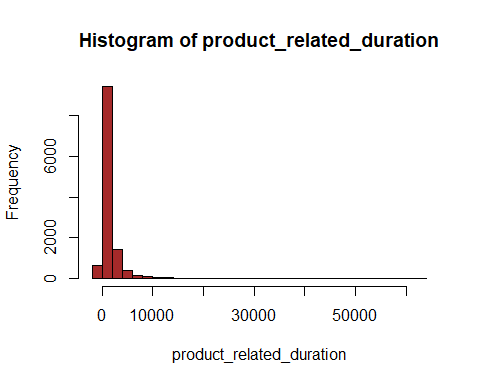
##we check the skewness  
#---  
#  
skewness(df2$productrelated\_duration)

## [1] 7.251403

##we check the kurtosis  
#---  
#  
kurtosis(df2$productrelated\_duration)

## [1] 139.5908

#distribution  
product\_related\_duration<-(df2$productrelated\_duration)  
hist(product\_related\_duration,breaks=30,col=c("brown"))



***1.5.7 Bounce Rates***

##we check the unique values   
#---  
#  
length(unique(df2$BounceRates))

## [1] 0

summary(length(unique(df2$bouncerates)))

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1872 1872 1872 1872 1872 1872

##we check the median  
#---  
#  
median(df2$bouncerates)

## [1] 0.002930403

##sort  
bounce\_rate<-(df2$bouncerates)  
names(table(bounce\_rate))[table(bounce\_rate)==max(table(bounce\_rate))]

## [1] "0"

var(bounce\_rate)

## [1] 0.002061387

sd(bounce\_rate)

## [1] 0.0454025

##we check for the interquatile range  
#---  
#  
IQR(bounce\_rate)

## [1] 0.01666667

##we check for the range  
#---  
#  
range(bounce\_rate)

## [1] 0.0 0.2

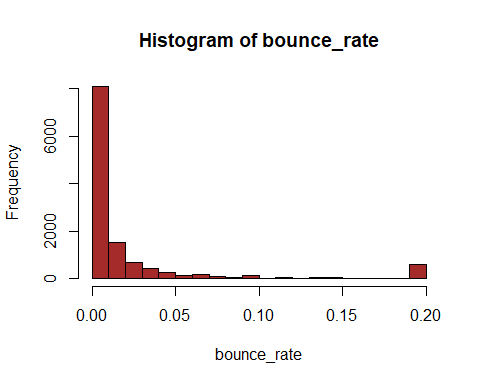
##we check the skewness  
#---  
#  
skewness(bounce\_rate)

## [1] 3.152874

##we check the kurtosis  
#---  
#  
kurtosis(bounce\_rate)

## [1] 12.25506

##distribution  
#---  
#  
hist(bounce\_rate,col=c("brown"))



***1.5.7 Exit Rates***

##we check for the unique values  
#---  
#  
length(unique(df2$exitrates))

## [1] 4777

summary(df2$exitrates)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.00000 0.01422 0.02500 0.04150 0.04848 0.20000

median(df2$exitrates)

## [1] 0.025

names(table(df2$exitrates))[table(df2$exitrates)==max(table(df2$exitrates))]

## [1] "0.2"

##we check the variance  
#---  
#  
var(df2$exitrates)

## [1] 0.0021388

##we check the standard deviation  
#---  
#  
sd(df2$exitrates)

## [1] 0.04624716

##we check the range  
#---  
#  
range(df2$exitrates)

## [1] 0.0 0.2

##we check the interquatile range  
#---  
#  
IQR(df2$exitrates)

## [1] 0.03426227

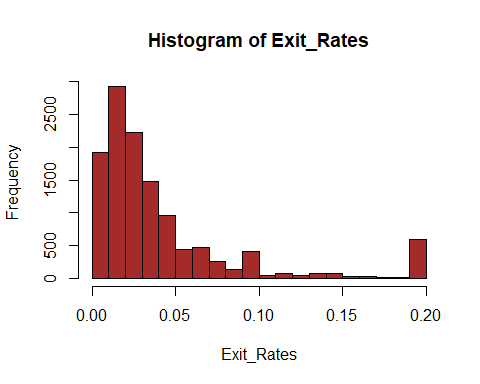
##we check the skewness  
#---  
#  
skewness(df2$exitrates)

## [1] 2.233125

##we check the kurtosis  
#---  
#  
kurtosis(df2$exitrates)

## [1] 7.624252

##distribution  
#---  
#  
Exit\_Rates<-(df2$exitrates)  
hist(Exit\_Rates,col=c("brown"))



***1.5.8 Page Values***

##we check for the unique values  
#---  
#  
length(unique(df2$pagevalues))

## [1] 2704

##we check for the median  
#---  
#  
median(df2$pagevalues)

## [1] 0

##we check for the modal  
#---  
#  
names(table(df2$pagevalues))[table(df2$pagevalues)==max(table(df2$pagevalues))]

## [1] "0"

##we check for the range  
#---  
#  
range(df2$pagevalues)

## [1] 0.0000 361.7637

##we check for the interquatile range  
#---  
#  
IQR(df2$pagevalues)

## [1] 0

##we check the standard deviation  
#---  
#  
sd(df2$pagevalues)

## [1] 18.65779

##we check the variance  
#---  
#  
var(df2$pagevalues)

## [1] 348.1132

##we check the range  
#---  
#  
range(df2$pagevalues)

## [1] 0.0000 361.7637

##we check the interquatile range  
#---  
#  
IQR(df2$pagevalues)

## [1] 0

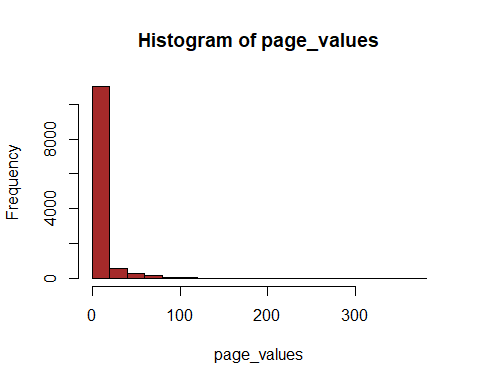
##we check the skewness  
#---  
#  
skewness(df2$pagevalues)

## [1] 6.348663

##we check the kurtosis  
#---  
#  
kurtosis(df2$pagevalues)

## [1] 67.94031

##we check the distribution  
#---  
#  
page\_values<-(df2$pagevalues)  
hist(page\_values,col=c("brown"))



***1.5.9 Special Day***

##we check the unique elements  
#---  
#  
length(unique(df2$specialday))

## [1] 6

summary(df2$specialday)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.00000 0.00000 0.00000 0.06197 0.00000 1.00000

##we check the median  
#---  
#  
median(df2$specialday)

## [1] 0

##we check the mode   
#---  
#  
#sort  
names(table(df2$specialday))[table(df2$specialday)==max(table(df2$specialday))]

## [1] "0"

##we check the variance  
#---  
#  
var(df2$specialday)

## [1] 0.03988432

##we check the standard deviation  
#---  
#  
sd(df2$specialday)

## [1] 0.1997106

##we check the range  
#---  
#  
range(df2$specialday)

## [1] 0 1

##we check the interquatile range  
#---  
#  
IQR(df2$specialday)

## [1] 0

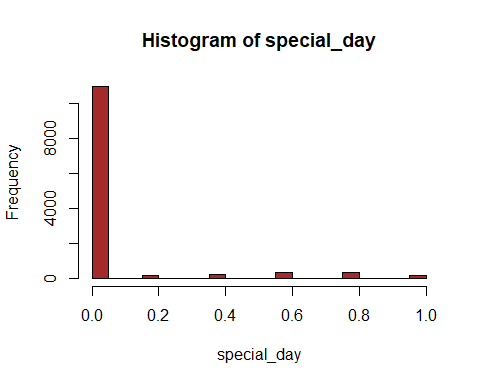
##we check the skewness  
#---  
#  
skewness(df2$specialday)

## [1] 3.284481

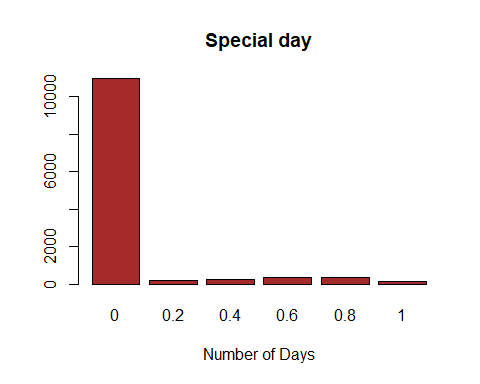
##we check the kurtosis  
#---  
#  
kurtosis(df2$specialday)

## [1] 12.78605

##we check the distribution  
#---  
#  
special\_day<-(df2$specialday)  
hist(special\_day,col=c("brown"))



# Simple Bar Plot  
counts <- table(special\_day)  
barplot(counts, main="Special day",col=c("brown"),  
 xlab="Number of Days")



***1.5.10 Month***

##we check for the unique values  
#---  
#  
length(unique(df2$month))

## [1] 10

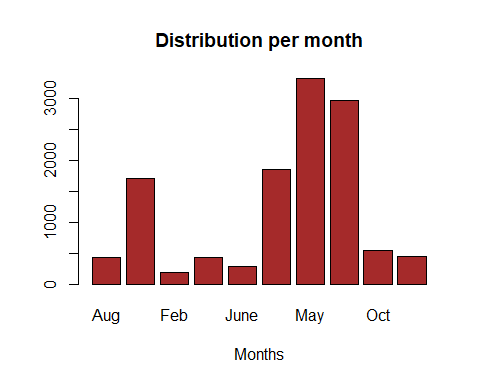
##we check for the summary  
#---  
#  
summary(df2$month)

## Length Class Mode   
## 12199 character character

##we check the mode  
#---  
#  
names(table(df2$month))[table(df2$month)==max(table(df2$month))]

## [1] "May"

## we plot a simple bar Plot  
#---  
#  
month<-(df2$month)  
counts <- table(month)  
barplot(counts, main="Distribution per month",col=c("brown"),xlab="Months")



***1.5.11 Operating System***

##we check the unique elements  
#---  
#  
length(unique(df2$operatingsystems))

## [1] 8

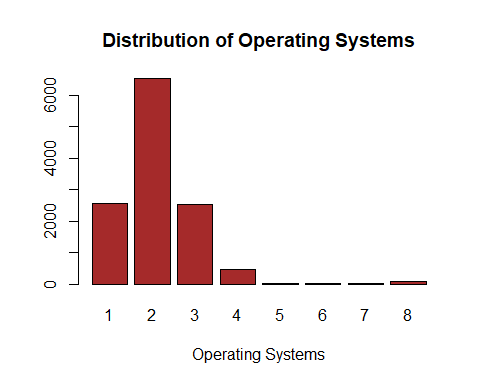
##we check the summary  
#---  
#  
summary(df2$operatingsystems)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 2.000 2.000 2.124 3.000 8.000

##we check the median  
#---  
#  
median(df2$operatingsystems)

## [1] 2

##we check the distribution  
#---  
#  
operating\_systems<-df2$operatingsystems  
counts <- table(operating\_systems)  
barplot(counts, main="Distribution of Operating Systems",col=c("brown"),  
 xlab="Operating Systems")



***1.5.11 Browser***

##we check the unique values  
#---  
#  
length(unique(df2$browser))

## [1] 13

##we check the summary  
#---  
#  
summary(df2$browser)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 2.000 2.000 2.358 2.000 13.000

##we check the median  
#---  
#  
median(df2$browser)

## [1] 2

##we check the mode  
#---  
#  
names(table(df2$browser))[table(df2$browser)==max(table(df2$browser))]

## [1] "2"

##we check the variance  
#---  
#  
var((df2$browser))

## [1] 2.926075

##we check the standard deviation  
#---  
#  
sd((df2$browser))

## [1] 1.710578

##we check the range  
#---  
#  
range((df2$browser))

## [1] 1 13

##we check the interquatile range  
#---  
#  
IQR((df2$browser))

## [1] 0

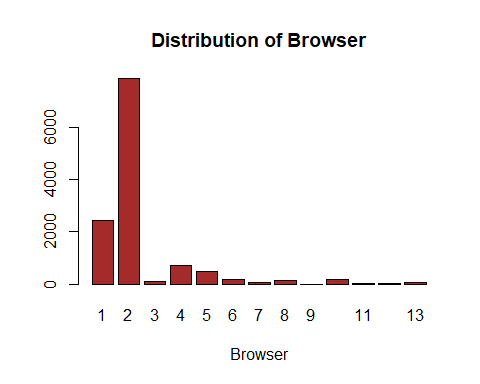
##we check the skewness  
#---  
#  
skewness((df2$browser))

## [1] 3.215653

##we check the kurtosis  
#---  
#  
kurtosis((df2$browser))

## [1] 15.53659

##we check the distribution  
#---  
#  
browser<-(df2$browser)  
counts <- table(browser)  
barplot(counts, main="Distribution of Browser",col=c("brown"),xlab="Browser")



***1.5.12 Region***

##we check for the unique elements  
#---  
#  
length(unique(df2$region))

## [1] 9

##we check for the summary  
#---  
#  
summary(df2$region)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 1.000 3.000 3.153 4.000 9.000

##we check the median  
#---  
#  
median(df2$region)

## [1] 3

##we check the mode  
#---  
#  
names(table(df2$region))[table(df2$region)==max(table(df2$region))]

## [1] "1"

##we check the variance  
#---  
#  
var(df2$region)

## [1] 5.771712

##we check the standard deviation  
#---  
#  
sd(df2$region)

## [1] 2.402439

##we check the range  
#---  
#  
range(df2$region)

## [1] 1 9

##we check the interquatile range  
#---  
#  
IQR(df2$region)

## [1] 3

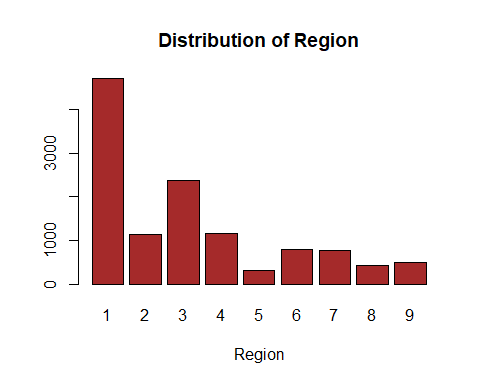
##we check the skewness  
#---  
#  
skewness(df2$region)

## [1] 0.9787304

##we check the kurtosis  
#---  
#  
kurtosis(df2$region)

## [1] 2.840195

##we check the distribution  
#---  
#  
region<-(df2$region)  
counts <- table(region)  
barplot(counts, main="Distribution of Region",col=c("brown"),xlab="Region")



***1.5.13 Traffic Type***

##we check the unique values  
#---  
#  
length(unique(df2$traffictype))

## [1] 20

summary(df2$traffictype)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 2.000 2.000 4.075 4.000 20.000

##we check the median  
#---  
#  
median(df2$traffictype)

## [1] 2

##we check the mode  
#---  
#  
names(table(df2$traffictype))[table(df2$traffictype)==max(table(df2$traffictype))]

## [1] "2"

##we check the variance  
#---  
#  
var(df2$traffictype)

## [1] 16.12675

##we check the standard deviation  
#---  
#  
sd(df2$traffictype)

## [1] 4.015813

##we check the range  
#---  
#  
range(df2$traffictype)

## [1] 1 20

##we check the interquatile range  
#---  
#  
IQR(df2$traffictype)

## [1] 2

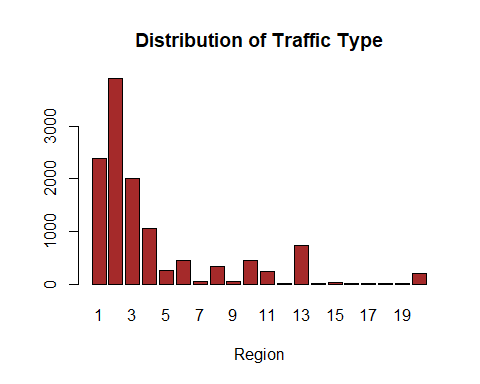
##we check the skewness  
#---  
#  
skewness(df2$traffictype)

## [1] 1.958522

##we check the kurtosis  
#---  
#  
kurtosis(df2$traffictype)

## [1] 6.466127

##we check the distribution  
#---  
#  
traffic\_type <- (df2$traffictype)  
counts <- table(traffic\_type)  
barplot(counts, main="Distribution of Traffic Type",col=c("brown"),xlab="Region")



***1.5.14 Visitor Type***

##we check the unique elements  
#---  
#  
length(unique(df2$visitortype))

## [1] 3

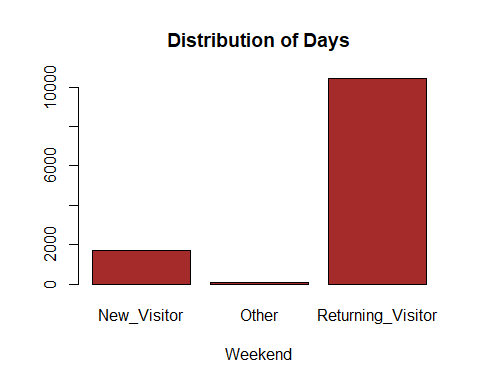
##we check the summary  
#---  
#  
summary(df2$visitortype)

## Length Class Mode   
## 12199 character character

##we check the mode  
#---  
#  
names(table(df2$visitortype))[table(df2$visitortype)==max(table(df2$visitortype ))]

## [1] "Returning\_Visitor"

##we check the distribution  
#---  
#  
visitor\_type<-(df2$visitortype)  
  
counts <- table(visitor\_type)  
barplot(counts, main="Distribution of Days",col=c("brown"),xlab="Weekend")



***1.5.14 Weekend***

##we check the unique elements  
#---  
#  
length(unique(df2$weekend))

## [1] 2

##we check the summary  
#---  
#  
summary(df2$weekend)

## Mode FALSE TRUE   
## logical 9343 2856

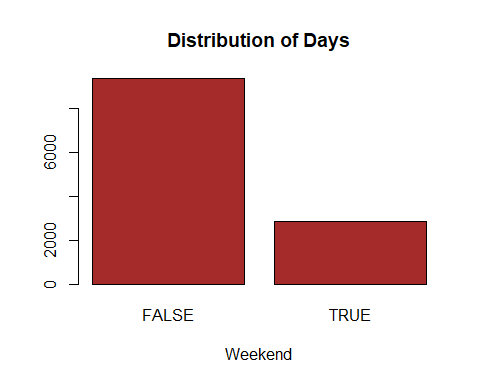
##we check the median  
#---  
#  
median(df2$weekend)

## [1] FALSE

##we check the mode  
#---  
#  
names(table(df2$weekend))[table(df2$weekend)==max(table(df2$weekend))]

## [1] "FALSE"

##we check the distribution  
#---  
#  
weekend<-(df2$weekend)  
counts <- table(weekend)  
barplot(counts, main="Distribution of Days",col=c("brown"),xlab="Weekend")



***1.5.15 Revenue***

##we check the unique elements  
#---  
#  
length(unique(df2$revenue))

## [1] 2

##we check the summary  
#---  
#  
summary(df2$revenue)

## Mode FALSE TRUE   
## logical 10291 1908

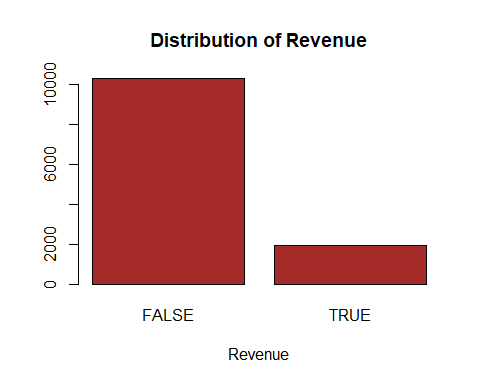
##we check the median  
#---  
#  
median(df2$revenue)

## [1] FALSE

##we check the mode  
#---  
#  
names(table(df2$revenue))[table(df2$revenue)==max(table(df2$revenue ))]

## [1] "FALSE"

revenue<-(df2$revenue)  
counts <- table(revenue)  
barplot(counts, main="Distribution of Revenue",col=c("brown"),xlab="Revenue")

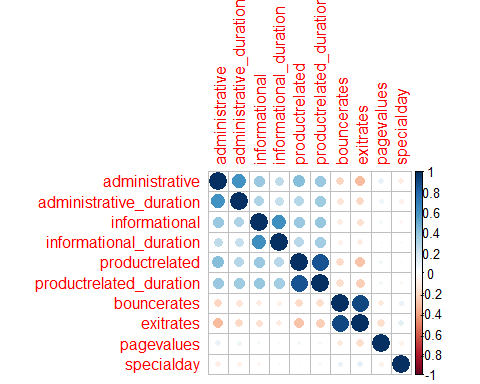


***5.2 Bivariate Analysis***

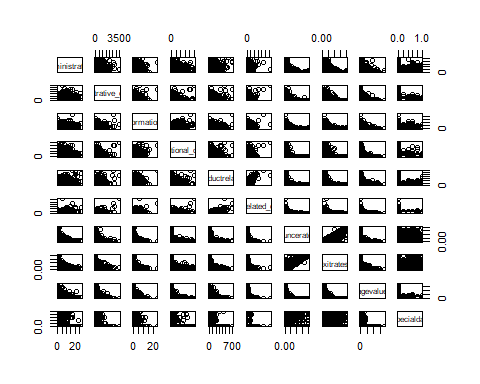
##we check the correlation  
#---  
#  
# calculate correlations  
correlations <- cor(df2[,1:10])  
correlations

## administrative administrative\_duration informational  
## administrative 1.00000000 0.60040965 0.37528761  
## administrative\_duration 0.60040965 1.00000000 0.30143630  
## informational 0.37528761 0.30143630 1.00000000  
## informational\_duration 0.25478602 0.23718986 0.61867795  
## productrelated 0.42819151 0.28678391 0.37260472  
## productrelated\_duration 0.37102722 0.35351379 0.38608372  
## bouncerates -0.21366664 -0.13733340 -0.10950530  
## exitrates -0.31127413 -0.20202445 -0.15956681  
## pagevalues 0.09692097 0.06616837 0.04739015  
## specialday -0.09707210 -0.07473689 -0.04937677  
## informational\_duration productrelated  
## administrative 0.25478602 0.42819151  
## administrative\_duration 0.23718986 0.28678391  
## informational 0.61867795 0.37260472  
## informational\_duration 1.00000000 0.27906195  
## productrelated 0.27906195 1.00000000  
## productrelated\_duration 0.34658069 0.86030819  
## bouncerates -0.07015947 -0.19351577  
## exitrates -0.10293268 -0.28616321  
## pagevalues 0.03006416 0.05411549  
## specialday -0.03129304 -0.02593062  
## productrelated\_duration bouncerates exitrates  
## administrative 0.37102722 -0.21366664 -0.3112741  
## administrative\_duration 0.35351379 -0.13733340 -0.2020245  
## informational 0.38608372 -0.10950530 -0.1595668  
## informational\_duration 0.34658069 -0.07015947 -0.1029327  
## productrelated 0.86030819 -0.19351577 -0.2861632  
## productrelated\_duration 1.00000000 -0.17437550 -0.2453340  
## bouncerates -0.17437550 1.00000000 0.9033582  
## exitrates -0.24533401 0.90335819 1.0000000  
## pagevalues 0.05084062 -0.11599198 -0.1735715  
## specialday -0.03821065 0.08783999 0.1167838  
## pagevalues specialday  
## administrative 0.09692097 -0.09707210  
## administrative\_duration 0.06616837 -0.07473689  
## informational 0.04739015 -0.04937677  
## informational\_duration 0.03006416 -0.03129304  
## productrelated 0.05411549 -0.02593062  
## productrelated\_duration 0.05084062 -0.03821065  
## bouncerates -0.11599198 0.08783999  
## exitrates -0.17357154 0.11678376  
## pagevalues 1.00000000 -0.06453271  
## specialday -0.06453271 1.00000000

# create correlation plot  
corrplot(correlations, method="circle")



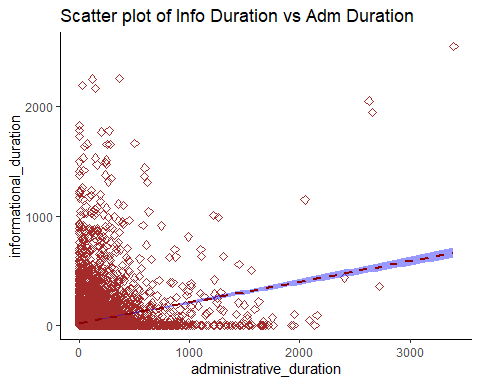
##we plot a pair plot  
#---  
#  
pairs(df2[,1:10])



***5.2.1 Sites Visited Duration***

##Scatter plot for Administrative\_Duration against Informational\_Duration  
#---  
#  
ggplot(df2, aes(x = administrative\_duration, y =informational\_duration)) +  
 geom\_point(size = 2, color= "brown", shape = 23)+  
 geom\_smooth(method=lm, linetype="dashed",color="darkred",fill="blue")+  
 labs(title = "Scatter plot of Info Duration vs Adm Duration")

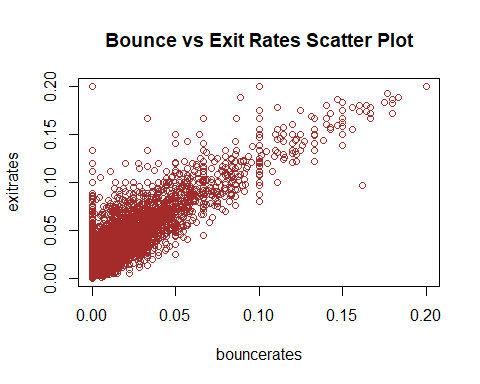
## `geom\_smooth()` using formula 'y ~ x'



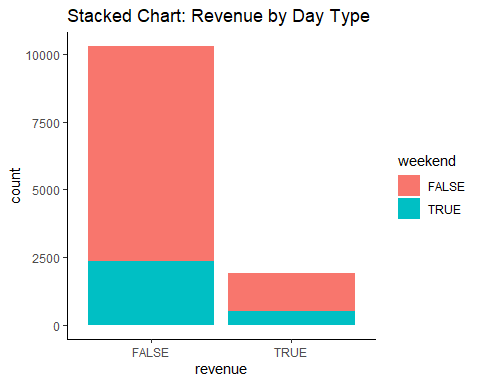
#There is a positive non-linear correlation between the time spent on the Administrative site  
#and the Informational site

***5.2.2 Metrics***

##Scatter plot for Bounce vs Exit Rates Scatter Plot  
#---  
#  
plot(exitrates ~ bouncerates, dat = df2,col = "brown",main = "Bounce vs Exit Rates Scatter Plot")

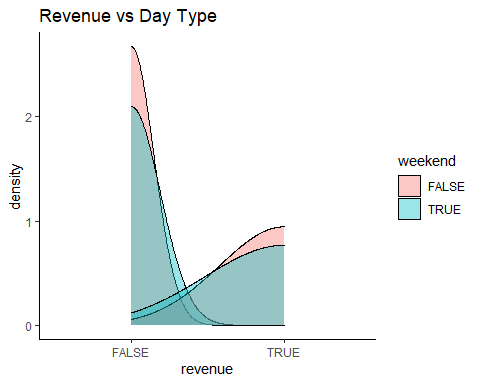


##we plot the stacked bar chart for revenue against daytype  
#---  
#  
df2 %>%ggplot(aes(revenue)) +geom\_bar(aes(fill = weekend))+labs(title = "Stacked Chart: Revenue by Day Type")

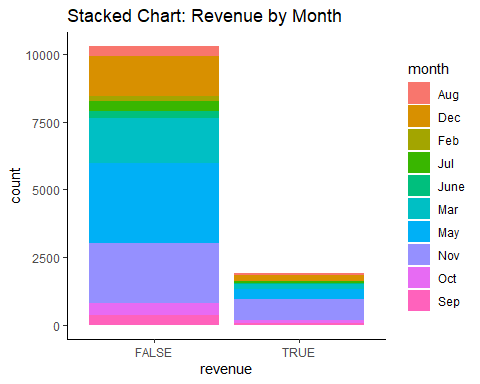


#A lot of revenue is generated over the weekend

ggplot(df2, aes(x = revenue, fill = weekend)) +geom\_density(alpha = 0.4) +  
 labs(title = "Revenue vs Day Type")

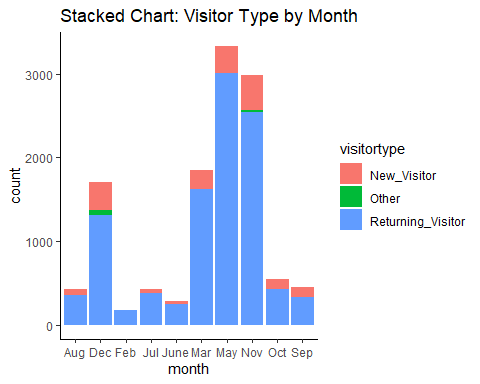


# Stacked bar chart: Revenue vs Month  
df2 %>%ggplot(aes(revenue)) +geom\_bar(aes(fill = month))+labs(title = "Stacked Chart: Revenue by Month")



***5.2.3 Type of visitor***

##stacked chart for visitor against month  
#---  
#  
df2 %>%ggplot(aes(month)) +geom\_bar(aes(fill = visitortype))+  
 labs(title = "Stacked Chart: Visitor Type by Month")



***5.3 Multivariate Analysis***

##we check the glimpse of the data  
#---  
#  
glimpse(df2)

## Rows: 12,199  
## Columns: 18  
## $ administrative <int> 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 2~  
## $ administrative\_duration <dbl> 0, 0, -1, 0, 0, 0, -1, -1, 0, 0, 0, 0, 0, 0, 0~  
## $ informational <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~  
## $ informational\_duration <dbl> 0, 0, -1, 0, 0, 0, -1, -1, 0, 0, 0, 0, 0, 0, 0~  
## $ productrelated <int> 1, 2, 1, 2, 10, 19, 1, 1, 2, 3, 3, 16, 7, 6, 2~  
## $ productrelated\_duration <dbl> 0.000000, 64.000000, -1.000000, 2.666667, 627.~  
## $ bouncerates <dbl> 0.200000000, 0.000000000, 0.200000000, 0.05000~  
## $ exitrates <dbl> 0.200000000, 0.100000000, 0.200000000, 0.14000~  
## $ pagevalues <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~  
## $ specialday <dbl> 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.4, 0.0, 0.8, 0~  
## $ month <chr> "Feb", "Feb", "Feb", "Feb", "Feb", "Feb", "Feb~  
## $ operatingsystems <int> 1, 2, 4, 3, 3, 2, 2, 1, 2, 2, 1, 1, 1, 2, 3, 1~  
## $ browser <int> 1, 2, 1, 2, 3, 2, 4, 2, 2, 4, 1, 1, 1, 5, 2, 1~  
## $ region <int> 1, 1, 9, 2, 1, 1, 3, 1, 2, 1, 3, 4, 1, 1, 3, 9~  
## $ traffictype <int> 1, 2, 3, 4, 4, 3, 3, 5, 3, 2, 3, 3, 3, 3, 3, 3~  
## $ visitortype <chr> "Returning\_Visitor", "Returning\_Visitor", "Ret~  
## $ weekend <lgl> FALSE, FALSE, FALSE, FALSE, TRUE, FALSE, FALSE~  
## $ revenue <lgl> FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FALS~

##we check the resultant data types  
#---  
#  
sapply(df2, class)

## administrative administrative\_duration informational   
## "integer" "numeric" "integer"   
## informational\_duration productrelated productrelated\_duration   
## "numeric" "integer" "numeric"   
## bouncerates exitrates pagevalues   
## "numeric" "numeric" "numeric"   
## specialday month operatingsystems   
## "numeric" "character" "integer"   
## browser region traffictype   
## "integer" "integer" "integer"   
## visitortype weekend revenue   
## "character" "logical" "logical"

***5.3.1 Dummify the data***

## One hot encoding of the factor variables.  
#---  
#  
dmy <- dummyVars(" ~ .", data = df2)  
dummy\_df <- data.frame(predict(dmy, newdata =df2))  
  
#preview  
glimpse(dummy\_df)

## Rows: 12,199  
## Columns: 31  
## $ administrative <dbl> 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0,~  
## $ administrative\_duration <dbl> 0, 0, -1, 0, 0, 0, -1, -1, 0, 0, 0, 0, 0,~  
## $ informational <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~  
## $ informational\_duration <dbl> 0, 0, -1, 0, 0, 0, -1, -1, 0, 0, 0, 0, 0,~  
## $ productrelated <dbl> 1, 2, 1, 2, 10, 19, 1, 1, 2, 3, 3, 16, 7,~  
## $ productrelated\_duration <dbl> 0.000000, 64.000000, -1.000000, 2.666667,~  
## $ bouncerates <dbl> 0.200000000, 0.000000000, 0.200000000, 0.~  
## $ exitrates <dbl> 0.200000000, 0.100000000, 0.200000000, 0.~  
## $ pagevalues <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~  
## $ specialday <dbl> 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.4, 0.0, 0~  
## $ monthAug <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~  
## $ monthDec <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~  
## $ monthFeb <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,~  
## $ monthJul <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~  
## $ monthJune <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~  
## $ monthMar <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~  
## $ monthMay <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~  
## $ monthNov <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~  
## $ monthOct <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~  
## $ monthSep <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~  
## $ operatingsystems <dbl> 1, 2, 4, 3, 3, 2, 2, 1, 2, 2, 1, 1, 1, 2,~  
## $ browser <dbl> 1, 2, 1, 2, 3, 2, 4, 2, 2, 4, 1, 1, 1, 5,~  
## $ region <dbl> 1, 1, 9, 2, 1, 1, 3, 1, 2, 1, 3, 4, 1, 1,~  
## $ traffictype <dbl> 1, 2, 3, 4, 4, 3, 3, 5, 3, 2, 3, 3, 3, 3,~  
## $ visitortypeNew\_Visitor <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~  
## $ visitortypeOther <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~  
## $ visitortypeReturning\_Visitor <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,~  
## $ weekendFALSE <dbl> 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1,~  
## $ weekendTRUE <dbl> 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0,~  
## $ revenueFALSE <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,~  
## $ revenueTRUE <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~

##we select the resultant data type  
#---  
#  
sapply(dummy\_df, class)

## administrative administrative\_duration   
## "numeric" "numeric"   
## informational informational\_duration   
## "numeric" "numeric"   
## productrelated productrelated\_duration   
## "numeric" "numeric"   
## bouncerates exitrates   
## "numeric" "numeric"   
## pagevalues specialday   
## "numeric" "numeric"   
## monthAug monthDec   
## "numeric" "numeric"   
## monthFeb monthJul   
## "numeric" "numeric"   
## monthJune monthMar   
## "numeric" "numeric"   
## monthMay monthNov   
## "numeric" "numeric"   
## monthOct monthSep   
## "numeric" "numeric"   
## operatingsystems browser   
## "numeric" "numeric"   
## region traffictype   
## "numeric" "numeric"   
## visitortypeNew\_Visitor visitortypeOther   
## "numeric" "numeric"   
## visitortypeReturning\_Visitor weekendFALSE   
## "numeric" "numeric"   
## weekendTRUE revenueFALSE   
## "numeric" "numeric"   
## revenueTRUE   
## "numeric"

##separating dependent and independent variable  
#---  
#  
## we remove the revenue column from the data  
dummy\_df <- dummy\_df[, -c(30:31)]  
dim(dummy\_df)

## [1] 12199 29

dummy\_df.class<- df2[, "revenue"]  
  
#There are 29 columns in dummy\_df

***5.3.3 Scaling Vs Normalization***

***5.3.3.1 Scaling***

At this step data is transformed to fit within the range of 0 and 1

dummy\_df\_scaled <- scale(dummy\_df)  
summary(dummy\_df\_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   
## monthAug monthDec monthFeb monthJul   
## 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   
## monthJune monthMar monthMay monthNov   
## 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   
## monthOct monthSep 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 visitortypeNew\_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   
## visitortypeOther visitortypeReturning\_Visitor weekendFALSE   
## 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   
## weekendTRUE   
## Min. :-0.5529   
## 1st Qu.:-0.5529   
## Median :-0.5529   
## Mean : 0.0000   
## 3rd Qu.:-0.5529   
## Max. : 1.8086

***5.3.3.2 Normalization***

Normalization is a technique often applied to change the values of numeric columns in the dataset to a common scale, without distorting differences in the ranges of values.

dummy\_df\_norm <- as.data.frame(apply(dummy\_df, 2, function(x) (x -  
min(x))/(max(x)-min(x))))  
summary(dummy\_df\_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   
## monthAug monthDec monthFeb monthJul   
## 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   
## monthJune monthMar monthMay monthNov   
## 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   
## monthOct monthSep 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 visitortypeNew\_Visitor visitortypeOther   
## 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   
## visitortypeReturning\_Visitor weekendFALSE weekendTRUE   
## 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

***Conclusion***

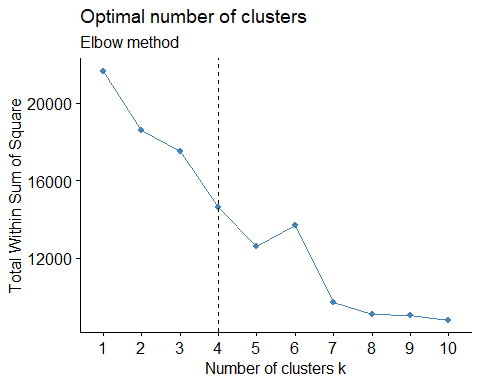
The normalized dataset has a smaller range for the values which are between 0 and 1

***5.4 Implementing the solution***

***5.4.1 Finding the optimal value of clusters***

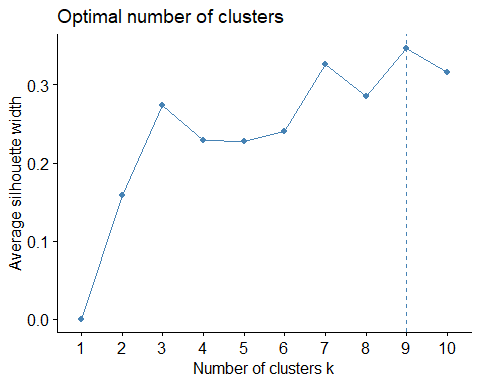
***5.4.1.1 Elbow Method***

##we check the number of clusters using Elbow method  
#---  
#  
fviz\_nbclust(dummy\_df\_norm, kmeans, method = "wss") +geom\_vline(xintercept = 4, linetype = 2)+  
 labs(subtitle = "Elbow method")



***5.4.1.2 Silhouette Method***

##we check the number of clusters through silhouette method  
#---  
#  
fviz\_nbclust(dummy\_df\_norm, kmeans, method = "silhouette")



***5.4.2 K means clustering***

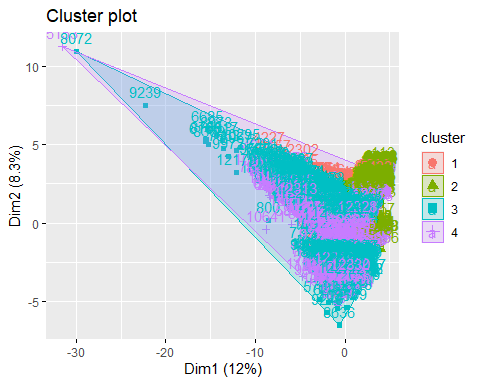
outputk <- kmeans(dummy\_df\_norm, 4)  
  
#preview the number of records in each cluster size  
outputk$size

## [1] 2447 585 6311 2856

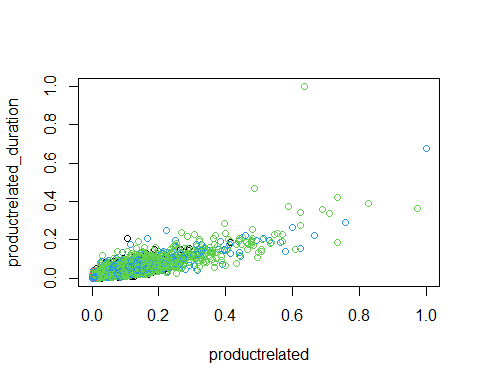
##The cluster center datapoints Per attribute  
#---  
#  
outputk$centers

## administrative administrative\_duration informational informational\_duration  
## 1 0.074921673 0.0214748616 0.0178960632 0.011090342  
## 2 0.001139601 0.0007227781 0.0005698006 0.000396925  
## 3 0.096721186 0.0270371335 0.0229031321 0.015410136  
## 4 0.092034962 0.0255872972 0.0244952148 0.016384511  
## productrelated productrelated\_duration bouncerates exitrates pagevalues  
## 1 0.039009121 0.0160651203 0.07501658 0.1950075 0.01525303  
## 2 0.003290295 0.0006655947 0.91710417 0.9511681 0.00000000  
## 3 0.051178680 0.0215498449 0.04401766 0.1536894 0.01798473  
## 4 0.047043427 0.0191676942 0.08728349 0.1847150 0.01747125  
## specialday monthAug monthDec monthFeb monthJul monthJune monthMar  
## 1 0.215856150 0.00000000 0.0000000 0.000000000 0.00000000 0.00000000 0.0000000  
## 2 0.071111111 0.02735043 0.1145299 0.051282051 0.04102564 0.05470085 0.1623932  
## 3 0.004341626 0.05086357 0.2025036 0.019648233 0.04816986 0.03264142 0.2032958  
## 4 0.055602241 0.03361345 0.1264006 0.009803922 0.03641457 0.01645658 0.1663165  
## monthMay monthNov monthOct monthSep operatingsystems browser  
## 1 1.0000000 0.0000000 0.00000000 0.00000000 0.1623562 0.1179335  
## 2 0.2854701 0.2410256 0.01025641 0.01196581 0.1748474 0.1146724  
## 3 0.0000000 0.3249881 0.06322294 0.05466646 0.1584309 0.1158163  
## 4 0.2500000 0.2769608 0.05042017 0.03361345 0.1610644 0.1029704  
## region traffictype visitortypeNew\_Visitor visitortypeOther  
## 1 0.2632816 0.1816618 0.09317532 0.00000000  
## 2 0.2717949 0.2168241 0.03760684 0.01709402  
## 3 0.2716091 0.1494550 0.15274917 0.01014102  
## 4 0.2682511 0.1608801 0.16771709 0.00245098  
## visitortypeReturning\_Visitor weekendFALSE weekendTRUE  
## 1 0.9068247 1 0  
## 2 0.9452991 1 0  
## 3 0.8371098 1 0  
## 4 0.8298319 0 1

##Visualising the clusters of the whole dataset  
#---  
#  
options(repr.plot.width = 11, repr.plot.height = 6)  
fviz\_cluster(outputk, dummy\_df\_norm)

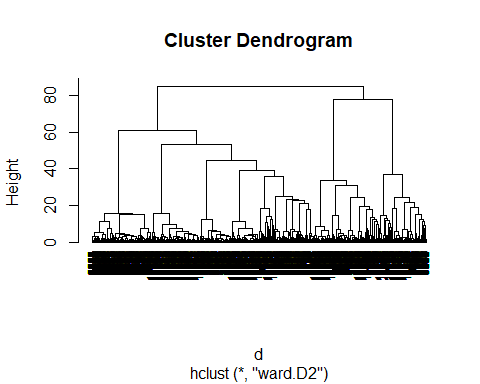


##Visualizing variable datatypes on a scatter plot  
#Product Related, vs Product Related Duration  
#---  
#  
plot(dummy\_df\_norm[, 5:6], col = outputk$cluster)



***5.4.2 K Hierachical clustering***

d <- dist(dummy\_df\_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)



***5.5.1.1 Dummify the variables***

## we apply the PCA  
# We also set two arguments, center and scale,  
# to be TRUE then preview our object with summary  
dummy\_PCA <- prcomp(dummy\_df\_norm,  
 center = TRUE,  
 scale = FALSE)  
summary(dummy\_PCA)

## Importance of components:  
## PC1 PC2 PC3 PC4 PC5 PC6 PC7  
## Standard deviation 0.6027 0.5249 0.4890 0.4369 0.37908 0.31341 0.30033  
## Proportion of Variance 0.2047 0.1553 0.1348 0.1076 0.08101 0.05537 0.05085  
## Cumulative Proportion 0.2047 0.3600 0.4948 0.6024 0.68343 0.73880 0.78965  
## PC8 PC9 PC10 PC11 PC12 PC13 PC14  
## Standard deviation 0.25907 0.21400 0.20283 0.19014 0.18821 0.17371 0.15733  
## Proportion of Variance 0.03784 0.02582 0.02319 0.02038 0.01997 0.01701 0.01395  
## Cumulative Proportion 0.82748 0.85330 0.87649 0.89687 0.91684 0.93385 0.94781  
## PC15 PC16 PC17 PC18 PC19 PC20 PC21  
## Standard deviation 0.15027 0.1298 0.12147 0.11865 0.08500 0.06923 0.06523  
## Proportion of Variance 0.01273 0.0095 0.00832 0.00794 0.00407 0.00270 0.00240  
## Cumulative Proportion 0.96054 0.9700 0.97835 0.98629 0.99036 0.99307 0.99546  
## PC22 PC23 PC24 PC25 PC26 PC27  
## Standard deviation 0.05217 0.04953 0.04018 0.03288 0.01328 3.259e-15  
## Proportion of Variance 0.00153 0.00138 0.00091 0.00061 0.00010 0.000e+00  
## Cumulative Proportion 0.99700 0.99838 0.99929 0.99990 1.00000 1.000e+00  
## PC28 PC29  
## Standard deviation 2.477e-15 1.496e-15  
## Proportion of Variance 0.000e+00 0.000e+00  
## Cumulative Proportion 1.000e+00 1.000e+00

##The principal component and how well they explain the variance  
#---  
#

## we plot the Correlation circle  
#---  
#

***Conclusion***

From the Correlation Circle and PCA we can see that the most important components are the following:

1.Administrative; Site. 2.Administrative\_Duration ; Time spent on the admin site. 3.Informational; Site. 4.Product Related; Site. 5.Product Related Duration; Time spent on the Product related site. 6.Bounce Rates ; Metric 7.Exit Rates ; Metric 8.Page Values ;Metric

***Scree plot*** A scree plot shows the eigenvalues on the y-axis and the number of factors on the x-axis. It always displays a downward curve. The point where the slope of the curve is clearly leveling off (the “elbow) indicates the number of factors that should be generated by the analysis.

**Conclusion**\_

From the plot above, the elbow forms in between the 7th and 8th dimensions. This indicates that the analysis should yield 7 factors. The first 7 principal components explain about 76% of the variance in the data.

***5.6 Recommendation***

Some effort should be geared towards weekends so as to generate more revenue