R Notebook

Introduction

a) **Specifying the Question**

The main objective of the study is to identify customer groups and their characteristics thus aiding Kira Plastinina's Sales and Marketing team in formulating their strategies.

b) **Defining the Metric for Success**

- Determining and visualising the descriptive statistics of the variables in the dataset.
- Identifying customer groups through clustering methods.
- Identifying the characteristics of clusters.

c) **Understanding the context**

Sales and Marketing teams aim to maximise a business' profit. Being able to understand a customer's behaviour allows for the planning of more targeted and effective campaigns, as different customer groups may prioritise different products or services.

d) **Recording the Experimental Design**

- Determine the main objectives.
- Load and preview the dataset.
- Understand the data.
- Prepare the dataset - Identify outliers, anomalies, duplicates, missing values, and determine how deal with them, drop unnecessary columns etc.
- Analyse the dataset using univariate, bivariate, and multivariate analysis techniques.
- Challenge the solution.
- Conclusion and recommendations

e) **Data Relevance**

The dataset provided (here) is relevant to the research question. It has relevant information on customer behaviour on the website.

Loading the dataset

```
#loading some required libraries
library(readr)
library(data.table)
library(caret)
```

```
## Loading required package: ggplot2

## Loading required package: lattice

library(psych)

##
## Attaching package: 'psych'

## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha

library(Metrics)

##
## Attaching package: 'Metrics'

## The following objects are masked from 'package:caret':
##
##     precision, recall

library(Amelia)

## Loading required package: Rcpp

## ##
## ## Amelia II: Multiple Imputation
## ## (Version 1.8.0, built: 2021-05-26)
## ## Copyright (C) 2005-2022 James Honaker, Gary King and Matthew Blackwell
## ## Refer to http://gking.harvard.edu/amelia/ for more information
## ##

library(tidyverse)

## ── Attaching packages
## ───────────────────────────────────────
## tidyverse 1.3.2 ──

## ✓ tibble  3.1.7     ✓ dplyr   1.0.9
## ✓ tidyr   1.2.0     ✓ stringr 1.4.0
## ✓ purrr   0.3.4     ✓ forcats 0.5.1
## ── Conflicts ──────────────────────────────────────────
tidyverse_conflicts() ──
## ✕ psych::%+%()       masks ggplot2::%+%()
## ✕ psych::alpha()     masks ggplot2::alpha()
## ✕ dplyr::between()   masks data.table::between()
## ✕ dplyr::filter()    masks stats::filter()
## ✕ dplyr::first()     masks data.table::first()
## ✕ dplyr::lag()       masks stats::lag()
## ✕ dplyr::last()      masks data.table::last()
```

```
## ✖ purrr::lift()      masks caret::lift()
## ✖ purrr::transpose() masks data.table::transpose()

df <- fread("http://bit.ly/EcommerceCustomersDataset")

df <- data.frame(df)
```

Checking the Data

Determining the no. of records in the dataset:

```
dim(df)

## [1] 12330    18

#the dataset has 12330 rows and 18  columns
```

Previewing the top of the dataset:

```
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
```

Previewing the bottom 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
```

Checking datatype of each column:

```
str(df)

## 'data.frame':    12330 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 ...
```

Tidying the Dataset

```
#checking column names
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"

#converting column names to lowercase
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"
```

```r
#checking for missing values
data.frame(colSums(is.na(df)))
```

```
##                         colSums.is.na.df..
## administrative                          14
## administrative_duration                 14
## informational                           14
## informational_duration                  14
## productrelated                          14
## productrelated_duration                 14
## bouncerates                             14
## exitrates                               14
## pagevalues                               0
## specialday                               0
## month                                    0
## operatingsystems                         0
## browser                                  0
## region                                   0
## traffictype                              0
## visitortype                              0
## weekend                                  0
## revenue                                  0
```

There were 14 missing values in administrative, administrative_duration, informational, informational_duration, productrelated, productrelated_duration, bouncerates, and exitrates columns. Given that the dataset has 12330 rows, the missing values will be dropped

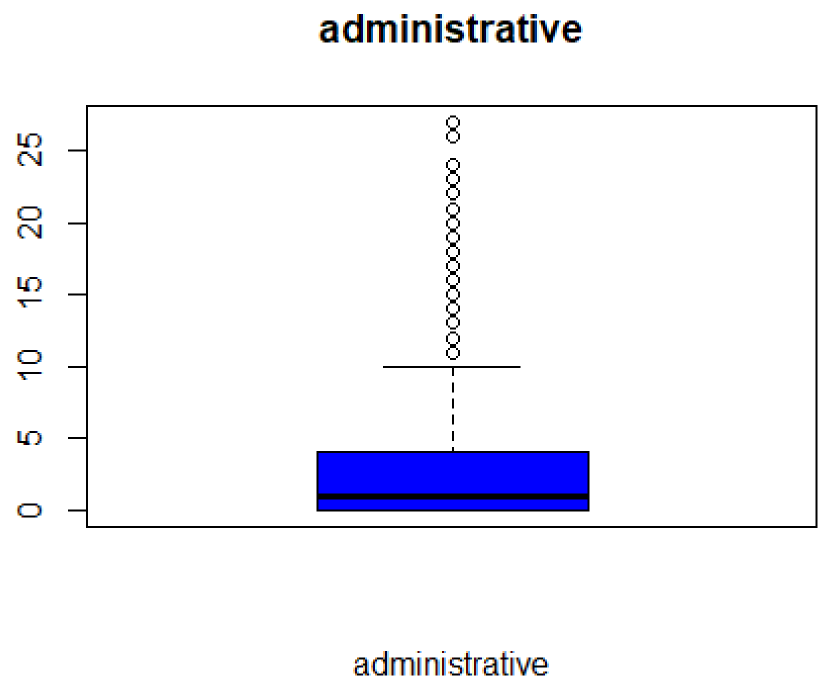
```r
#dropping missing values
df <- na.omit(df)

#the 14 nulls have been dropped
print(data.frame(colSums(is.na(df))))
```

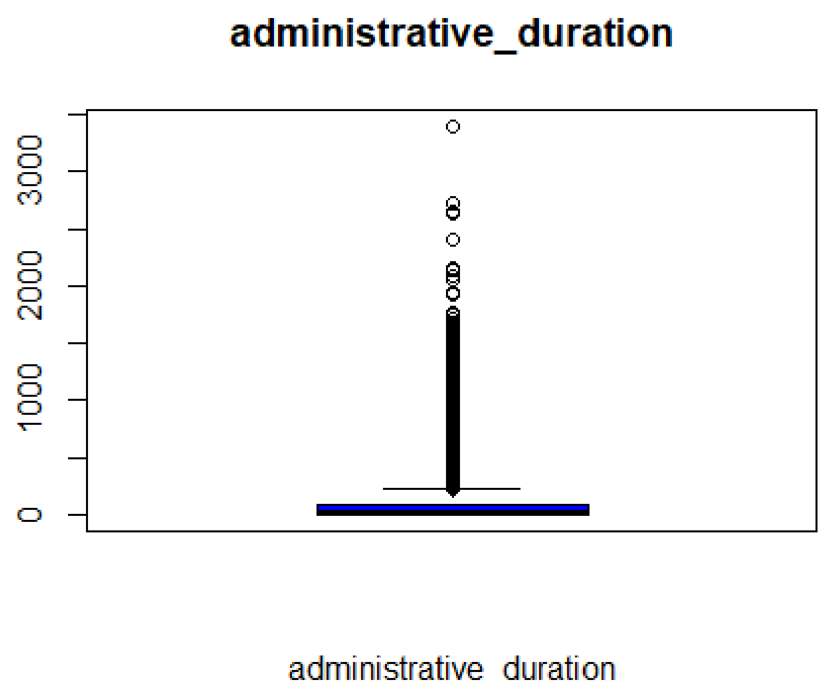
```
##                         colSums.is.na.df..
## administrative                           0
## administrative_duration                  0
## informational                            0
## informational_duration                   0
## productrelated                           0
## productrelated_duration                  0
## bouncerates                              0
## exitrates                                0
## pagevalues                               0
## specialday                               0
## month                                    0
```

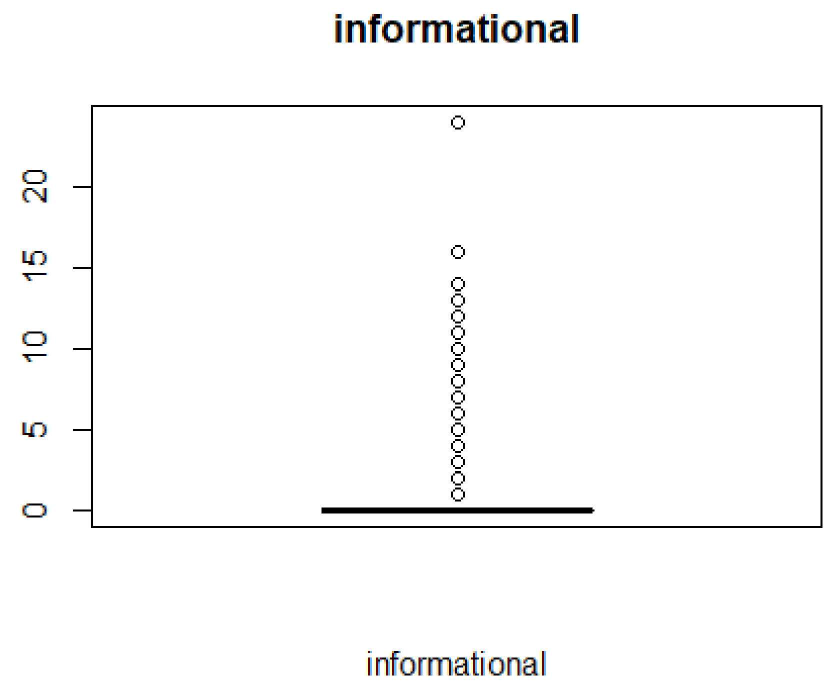
```
## operatingsystems                         0
## browser                                  0
## region                                   0
## traffictype                              0
## visitortype                              0
## weekend                                  0
## revenue                                  0

print(dim(df))

## [1] 12316    18

#checking for duplicates
nrow(df[duplicated(df),])

## [1] 117
```

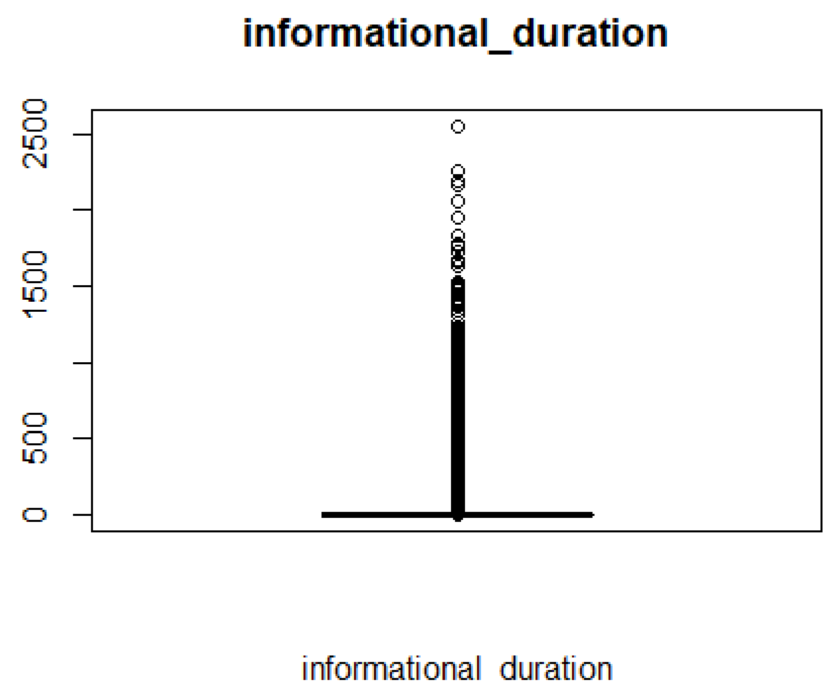
There were 117 duplicates which will not be dropped because it is possible for user behaviour and characteristics on the website to be similar.

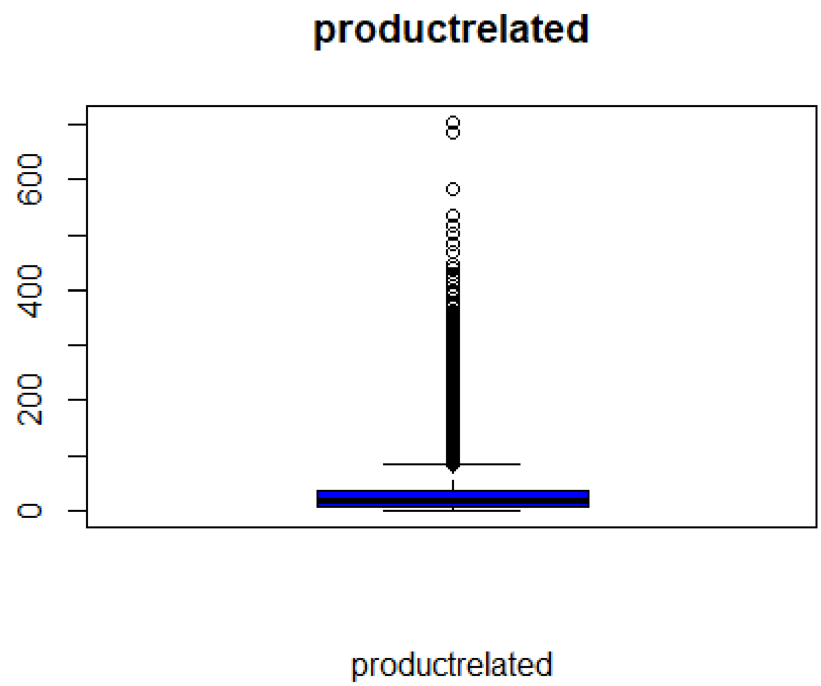
```
#separating continuous and categorical
colnames(df)

##  [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"

contin = c( "administrative","administrative_duration",
"informational","informational_duration",
"productrelated","productrelated_duration",
"bouncerates","exitrates","pagevalues")
cat = c("specialday", "month", "operatingsystems", "browser", "region",
"traffictype", "visitortype", "weekend", "revenue")

#checking for outliers in continuous columns
for (x in contin){
  boxplot(df[x], main=x, xlab=x, col="blue")
}
```

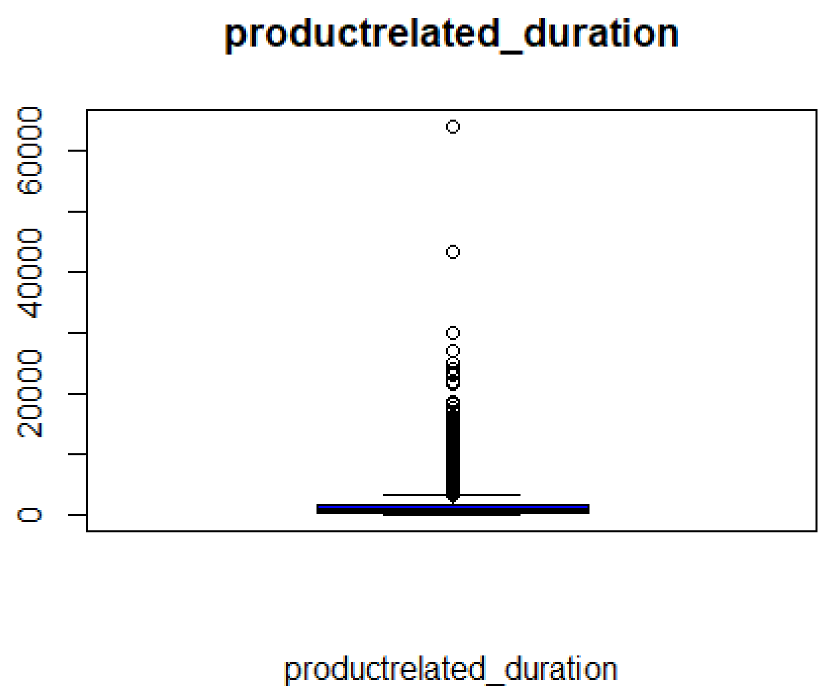

administrative
0
5
10
15
20
25
administrative

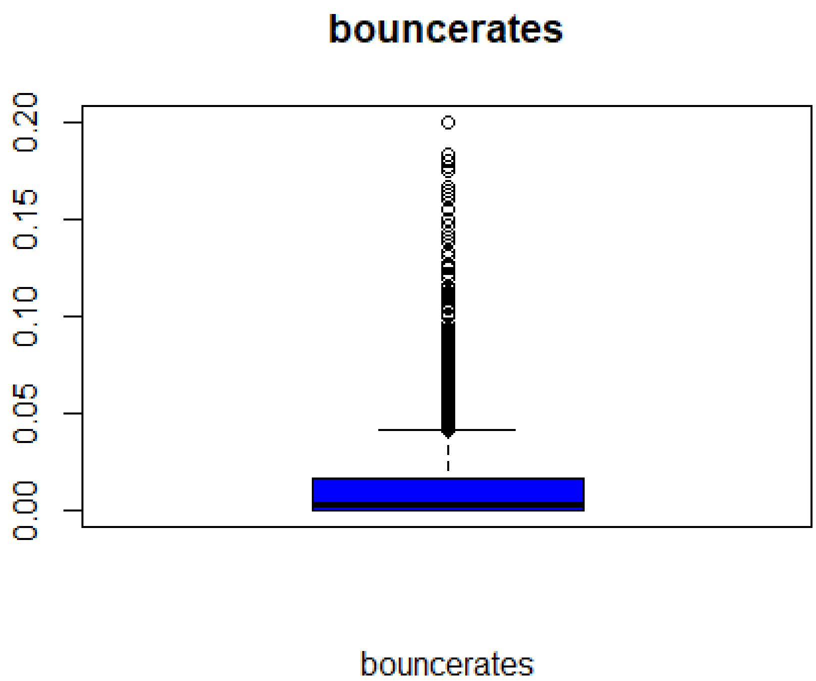

administrative_duration
0
1000
2000
3000
administrative_duration

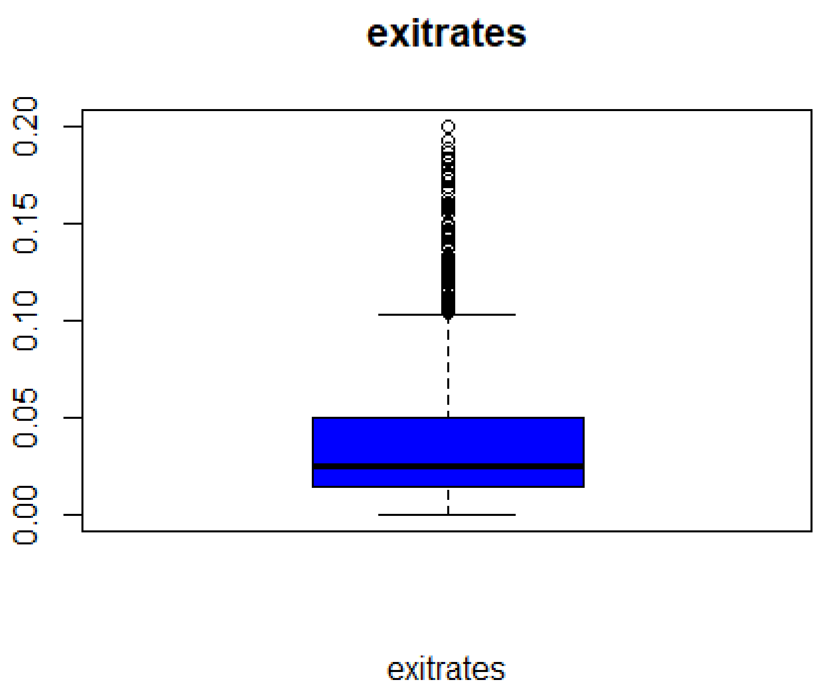

informational
0
5
10
15
20
informational

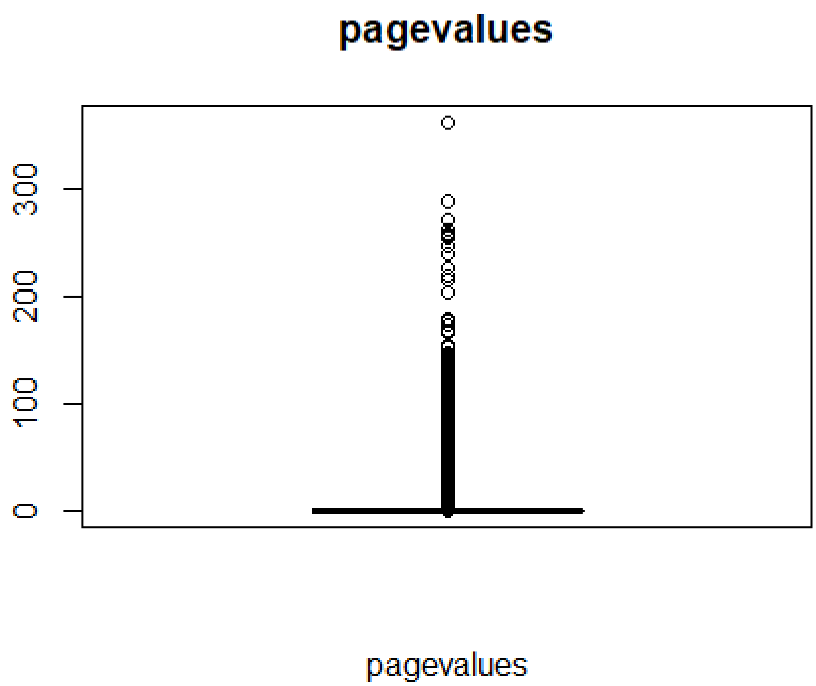

informational_duration
0
500
1500
2500
informational_duration


productrelated
0
200
400
600
productrelated


productrelated_duration
0
20000
40000
60000
productrelated_duration


bouncerates
0.00
0.05
0.10
0.15
0.20
bouncerates


exitrates
0.00
0.05
0.10
0.15
0.20
exitrates



There were outliers in the “administrative”,“administrative_duration”, “informational”,“informational_duration”, “productrelated”, “productrelated_duration”, “bouncerates”, “exitrates” and “pagevalues” columns. They will not be dropped as it is possible for some users to have spent longer than average on the site navigating through the numerous webpages.

```
#checking for anomalies in continuous
#the number of different types of pages visited by the visitor in the session
and total time spent in each of these page categories should not be less than
zero.

for (x in contin){
  print(paste(x, nrow(subset(df, df[x] < 0))))
}

## [1] "administrative 0"
## [1] "administrative_duration 33"
## [1] "informational 0"
## [1] "informational_duration 33"
## [1] "productrelated 0"
## [1] "productrelated_duration 33"
## [1] "bouncerates 0"
## [1] "exitrates 0"
## [1] "pagevalues 0"

dim(df)
```

```
## [1] 12316    18

#dropping observations that have the values above < 0 as those are anomalies

df <- subset(df, df["administrative_duration"] >= 0)

#checking that the 33 observations have been dropped

print(dim(df))

## [1] 12283    18

for (x in contin){
  print(paste(x, nrow(subset(df, df[x] < 0))))
}

## [1] "administrative 0"
## [1] "administrative_duration 0"
## [1] "informational 0"
## [1] "informational_duration 0"
## [1] "productrelated 0"
## [1] "productrelated_duration 0"
## [1] "bouncerates 0"
## [1] "exitrates 0"
## [1] "pagevalues 0"

#checking for number of unique values in categorical columns
for (x in cat){
  print(paste(x, length(unique(df[[x]]))))
}

## [1] "specialday 6"
## [1] "month 10"
## [1] "operatingsystems 8"
## [1] "browser 13"
## [1] "region 9"
## [1] "traffictype 20"
## [1] "visitortype 3"
## [1] "weekend 2"
## [1] "revenue 2"

#checking for anomalies in categorical

for (x in cat){
  print(x)
  print(unique(df[[x]]))

  print("***************************************")
}
```

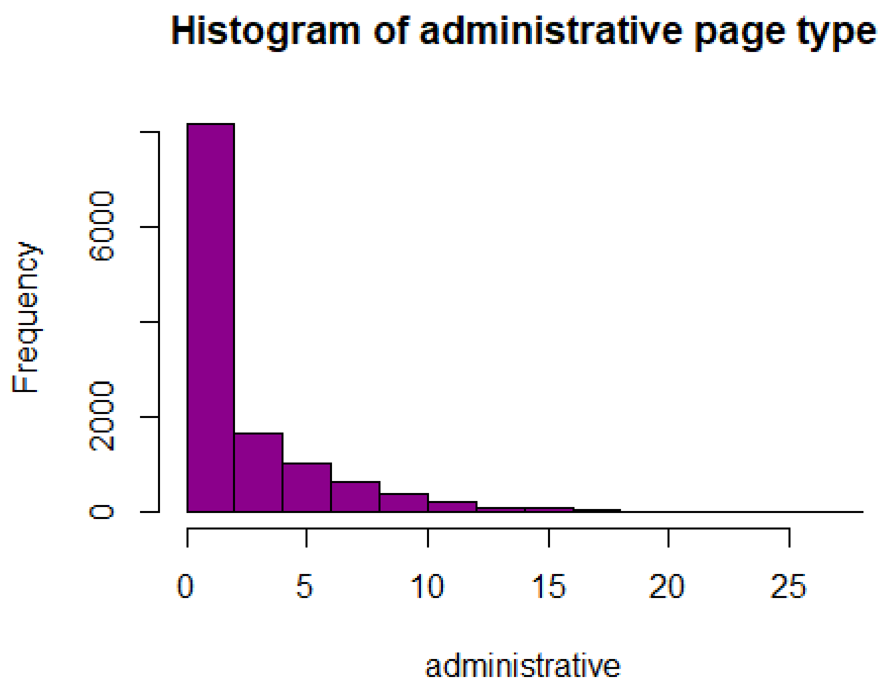
```
## [1] "specialday"
## [1] 0.0 0.8 0.4 1.0 0.2 0.6
## [1] "**************************************"
## [1] "month"
##  [1] "Feb"  "Mar"  "May"  "Oct"  "June" "Jul"  "Aug"  "Nov"  "Sep"  "Dec"
## [1] "**************************************"
## [1] "operatingsystems"
## [1] 1 2 3 4 7 6 8 5
## [1] "**************************************"
## [1] "browser"
##  [1]  1  2  3  4  5  6  7 10  8  9 12 13 11
## [1] "**************************************"
## [1] "region"
## [1] 1 2 3 4 9 5 6 7 8
## [1] "**************************************"
## [1] "traffictype"
##  [1]  1  2  4  3  5  6  7  8  9 10 11 12 13 14 15 18 19 16 17 20
## [1] "**************************************"
## [1] "visitortype"
## [1] "Returning_Visitor" "New_Visitor"       "Other"
## [1] "**************************************"
## [1] "weekend"
## [1] FALSE  TRUE
## [1] "**************************************"
## [1] "revenue"
## [1] FALSE  TRUE
## [1] "**************************************"
```

No anomalous values observed

Univariate Analysis

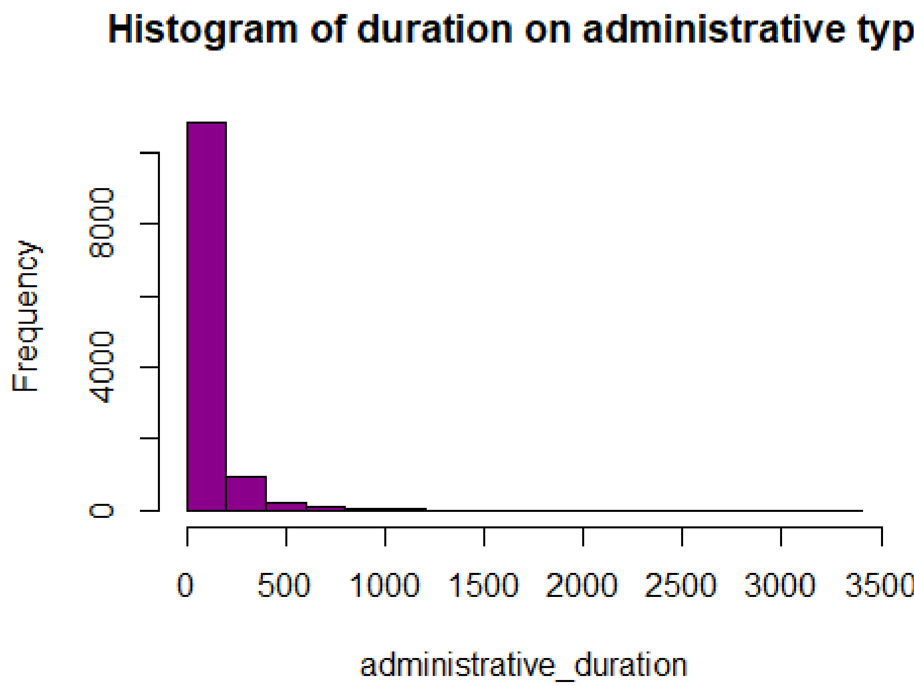
```
#loading ggplot 2 library for visualisation
library(ggplot2)

contin

## [1] "administrative"          "administrative_duration"
## [3] "informational"           "informational_duration"
## [5] "productrelated"          "productrelated_duration"
## [7] "bouncerates"             "exitrates"
## [9] "pagevalues"

#statistical summary of administrative variable
data.frame(describe(df$administrative))

##    vars     n     mean       sd median  trimmed    mad min max range
skew
## X1    1 12283 2.323862 3.325128      1 1.638852 1.4826   0  27    27
1.954851
##    kurtosis         se
## X1 4.674564 0.03000241
```

```r
#plotting administrative histogram
hist(df$administrative, col="darkmagenta",
     main="Histogram of administrative page type",
     xlab="administrative")
```



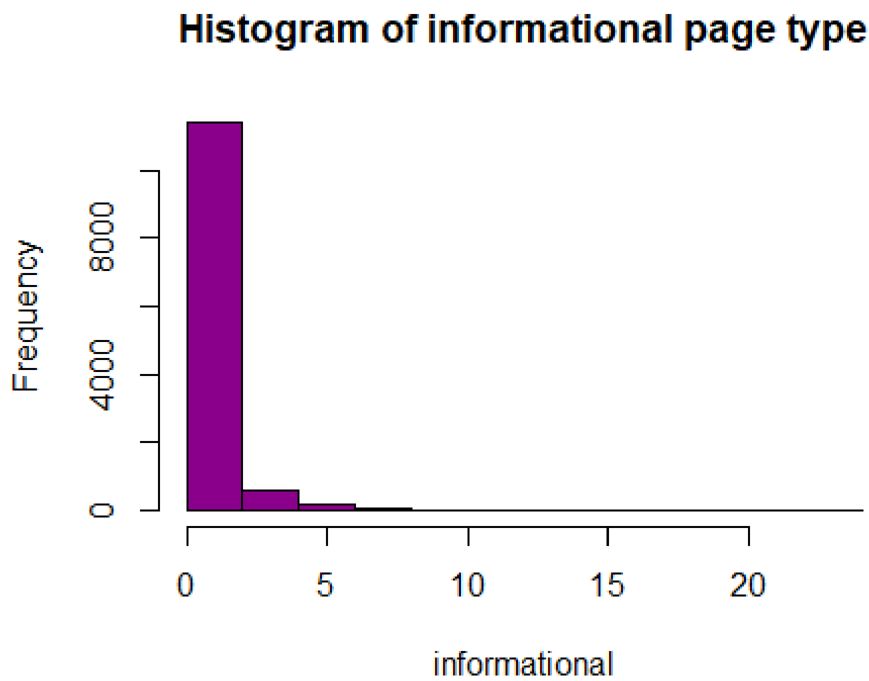
The number of administrative page types visited in a given session mostly ranged from 0 to 2.

```r
#statistical sumary of administrative_duration
describe(df$administrative_duration)

##    vars     n  mean     sd median trimmed   mad min     max   range skew
## X1    1 12283 81.13 177.05      8   42.37 11.86   0 3398.75 3398.75 5.61
##    kurtosis  se
## X1    50.37 1.6

#histogram of administrative_duration
hist(df$administrative_duration, col="darkmagenta",
     main="Histogram of duration on administrative type",
     xlab="administrative_duration")
```



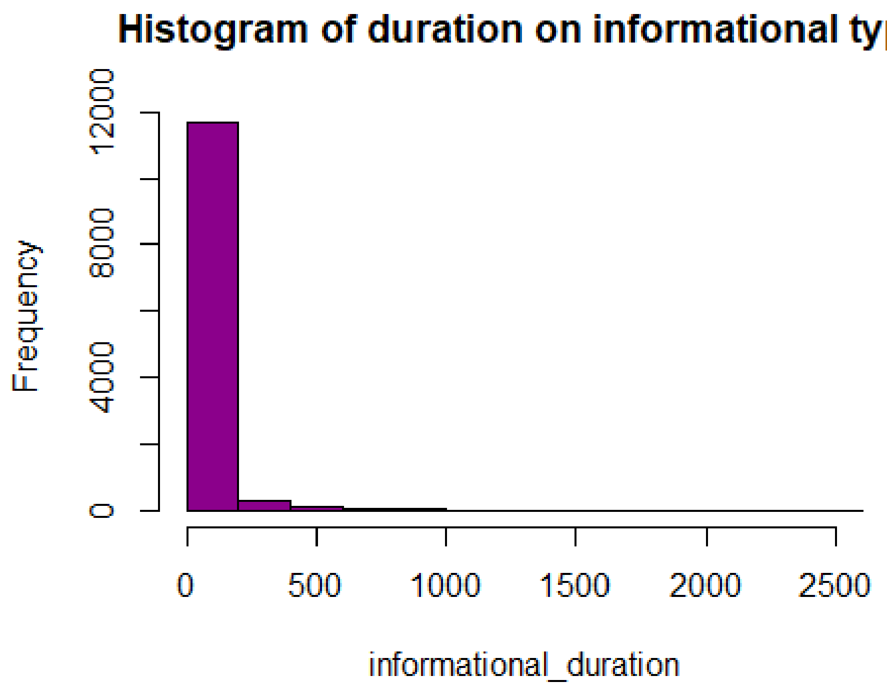
The duration on administrative page types in a given session mostly ranged from 0 to 200.

```
#statistical sumary of informational variable
describe(df$informational)

##    vars     n mean   sd median trimmed mad min max range skew kurtosis
se
## X1    1 12283 0.51 1.27      0    0.18   0   0  24    24 4.03    26.82
0.01

#histogram of informational
hist(df$informational, col="darkmagenta",
     main="Histogram of informational page type",
     xlab="informational")
```



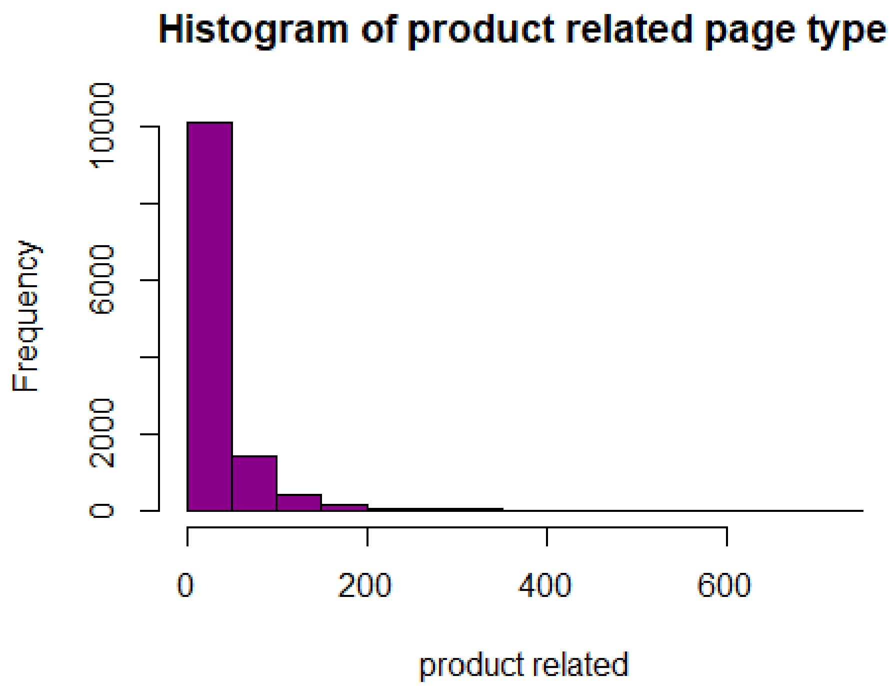
The number of informational page types visited in a given session mostly ranged from 0 to 2.

```
#statistical summary of informational_duration variable
describe(df$informational_duration)

##    vars     n mean  sd median trimmed mad min     max   range skew
kurtosis
## X1    1 12283 34.6 141      0    3.63   0   0 2549.38 2549.38 7.56
75.98
##      se
## X1 1.27

#histogram of informational_duration
hist(df$informational_duration, col="darkmagenta",
     main="Histogram of duration on informational type",
     xlab="informational_duration")
```



The duration on informational page types visited in a given session mostly ranged from 0 to 200.

```
#statistical sumary of productrelated variable
describe(df$productrelated)

##    vars     n  mean    sd median trimmed   mad min max range skew kurtosis
se
## X1    1 12283 31.85 44.52     18    22.86 19.27   0 705   705 4.34    31.14
0.4

#histogram of productrelated
hist(df$productrelated, col="darkmagenta",
     main="Histogram of product related page type",
     xlab="product related")
```

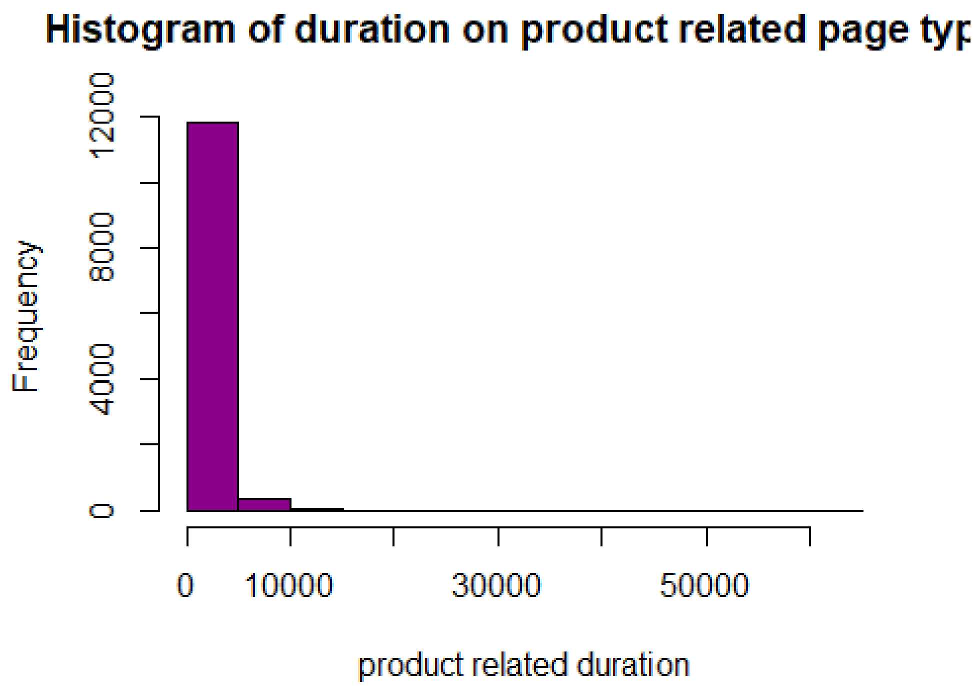


The number of product related page types visited in a given session mostly ranged from 0 to 50.

```
#statistical sumary of productrelated_duration variable
describe(df$productrelated_duration)
```

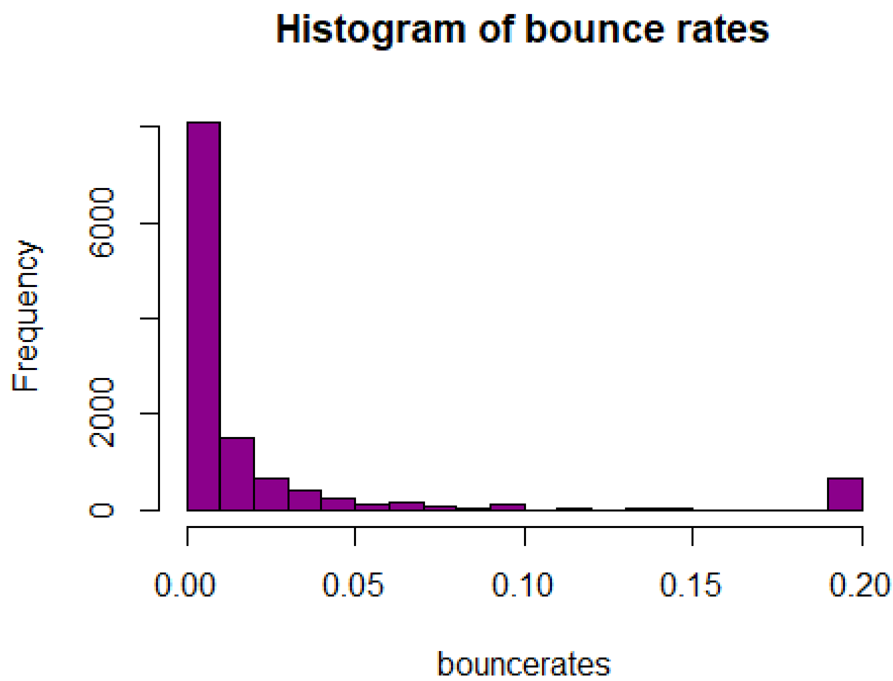
```
##    vars     n    mean      sd median trimmed    mad min      max    range
skew
## X1    1 12283 1199.25 1915.94  602.5  824.43 744.39   0 63973.52 63973.52
7.26
##    kurtosis    se
## X1    136.9 17.29
```

```
#histogram of productrelated_duration
hist(df$productrelated_duration, col="darkmagenta",
     main="Histogram of duration on product related page type",
     xlab="product related duration")
```



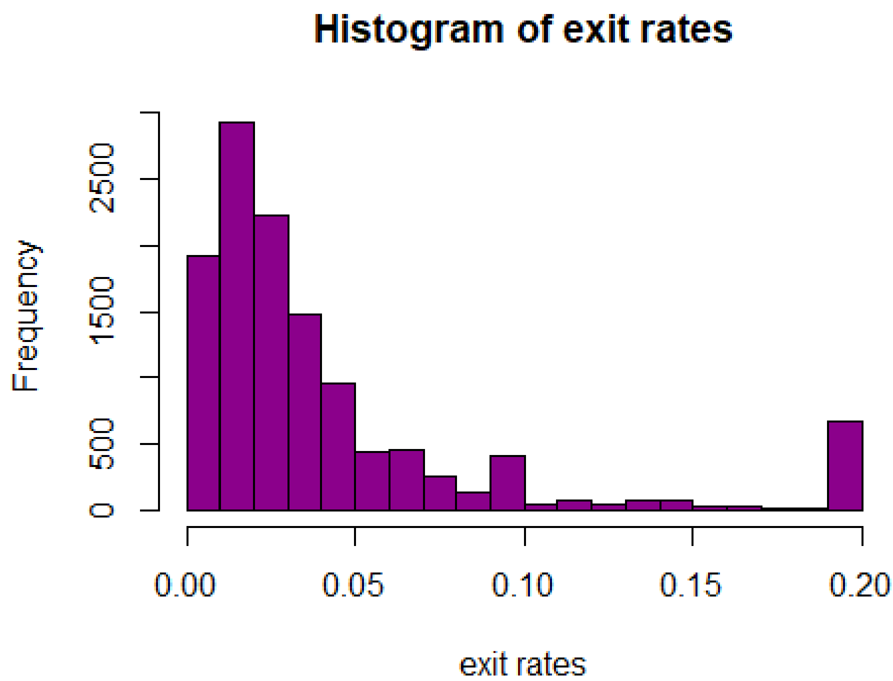
The duration on product-related page types in a given session mostly ranged from 0 to 5000.

```
contin

## [1] "administrative"          "administrative_duration"
## [3] "informational"           "informational_duration"
## [5] "productrelated"          "productrelated_duration"
## [7] "bouncerates"             "exitrates"
## [9] "pagevalues"

#statistical sumary of bouncerates variable
describe(df$bouncerates)

##    vars     n mean   sd median trimmed mad min max range skew kurtosis se
## X1    1 12283 0.02 0.05      0    0.01   0   0 0.2   0.2    3      8.1  0

#histogram of bouncerates
hist(df$bouncerates, col="darkmagenta",
     main="Histogram of bounce rates",
     xlab="bouncerates")
```



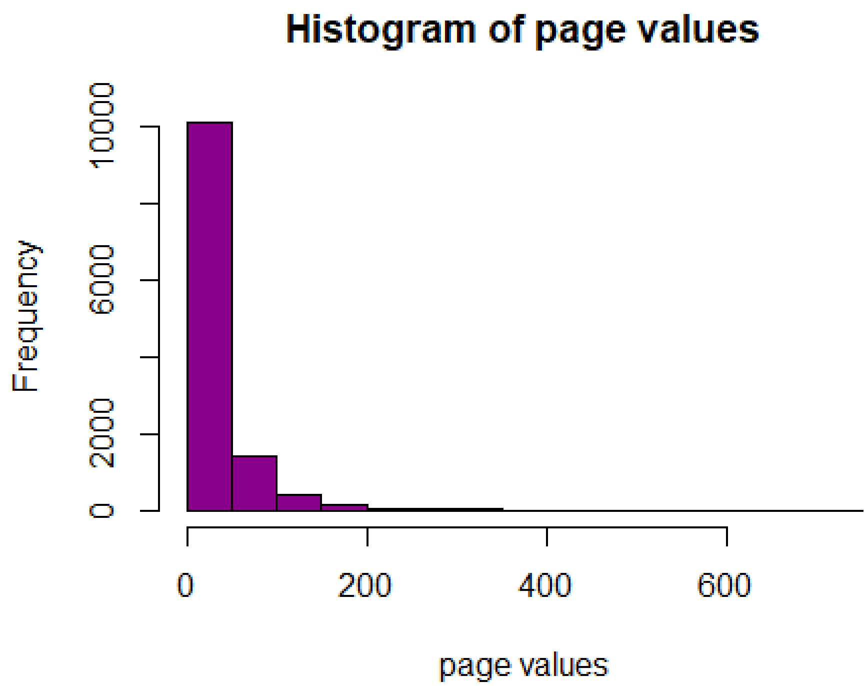
Bounce rates mostly ranged from 0 to 0.01

```
#statistical sumary of exitrates variable
describe(df$exitrates)

##    vars     n mean   sd median trimmed  mad min max range skew kurtosis se
## X1    1 12283 0.04 0.05   0.03    0.03 0.02   0 0.2   0.2 2.17     4.18  0

#histogram of exitrates
hist(df$exitrates, col="darkmagenta",
     main="Histogram of exit rates",
     xlab="exit rates")
```



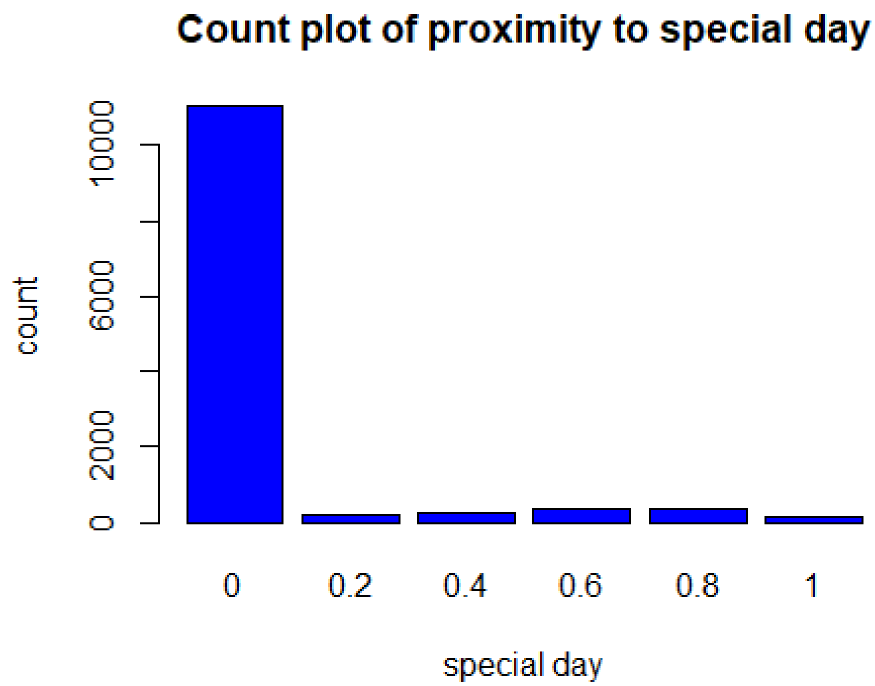
Exit rates mostly ranged from 0.01 to 0.02

```
#statistical sumary of page values variable
describe(df$pagevalues)

##    vars     n mean   sd median trimmed mad min    max  range skew kurtosis
se
## X1    1 12283 5.91 18.6      0    1.31   0   0 361.76 361.76 6.37    65.36
0.17

#histogram of page values
hist(df$productrelated, col="darkmagenta",
     main="Histogram of page values",
     xlab="page values")
```



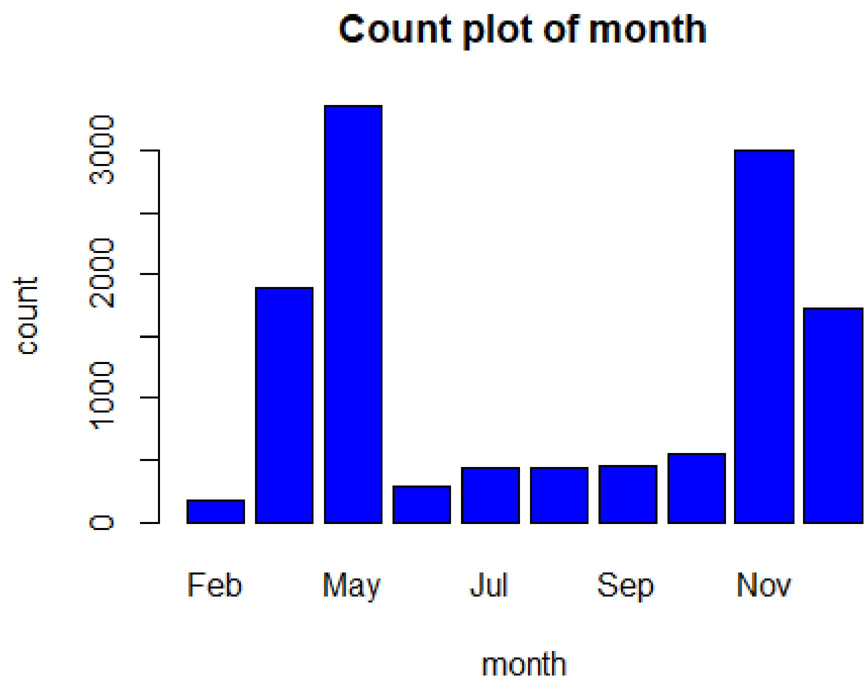
Page values mostly ranged from 0 to 50

```
cat

## [1] "specialday"        "month"             "operatingsystems" "browser"
## [5] "region"            "traffictype"       "visitortype"      "weekend"
## [9] "revenue"

#Count plot of specialday
barplot(table(df$specialday), col="blue", main="Count plot of proximity to
special day",
        xlab = "special day", ylab="count")
```



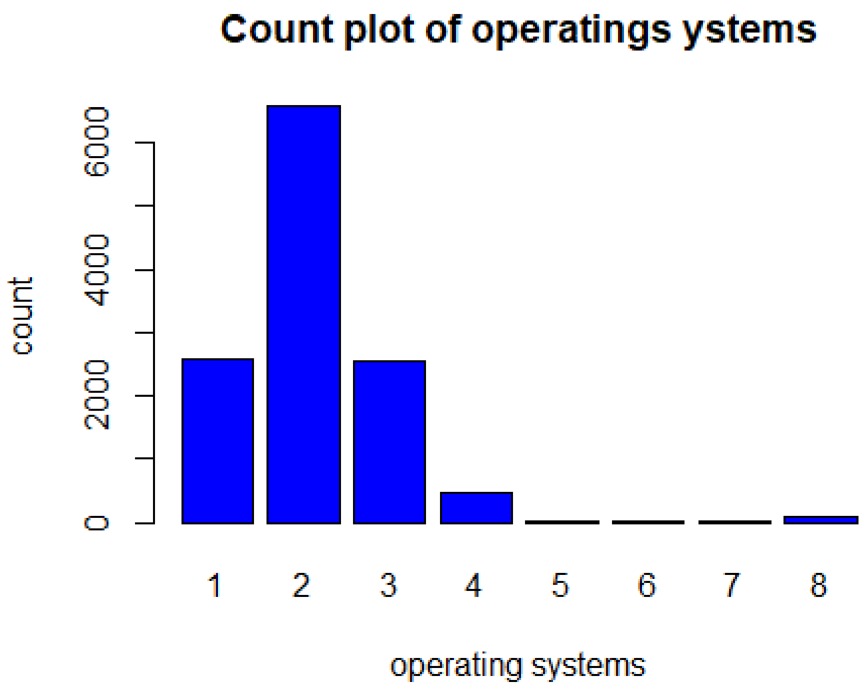
closeness of the site visiting time to a specific special day. Most visits were not close to a special day

```
#count plot of month
df2 <- copy(df)
df2$month <- factor(df$month, levels=c("Feb", "Mar", "May", "June", "Jul",
"Aug", "Sep", "Oct", "Nov", "Dec" ), ordered = TRUE)

barplot(table(df2$month), col="blue", main="Count plot of month",
        xlab = "month", ylab="count")
```



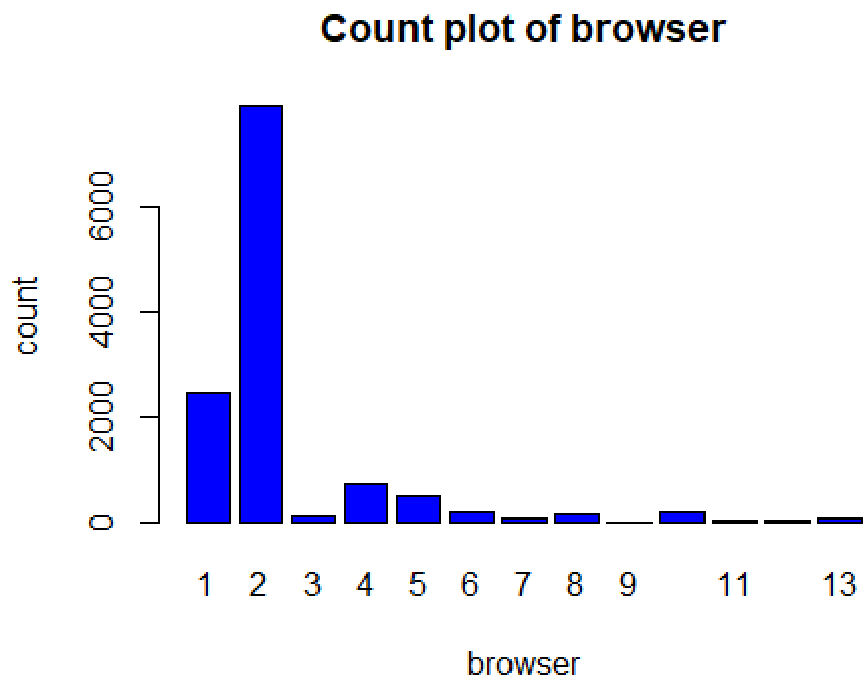
May was the month with the most visits according to the dataset

```
#count plot of  operatingsystems
barplot(table(df$operatingsystems), col="blue", main="Count plot of
operatings ystems",
        xlab = "operating systems", ylab="count")
```



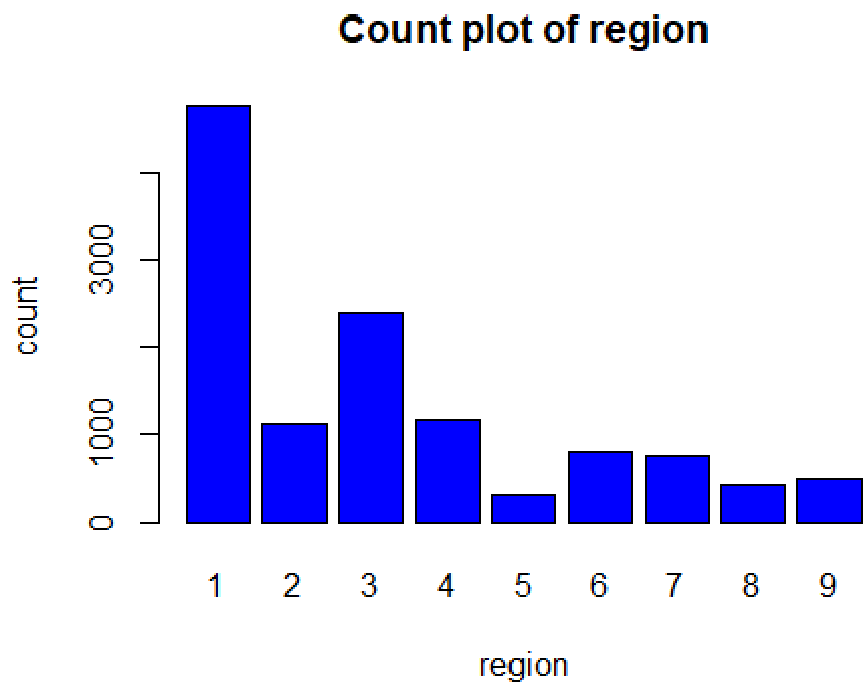
Operating system type 2 was the most common

```
#count plot of browser
barplot(table(df$browser), col="blue", main="Count plot of browser",
        xlab = "browser", ylab="count")
```



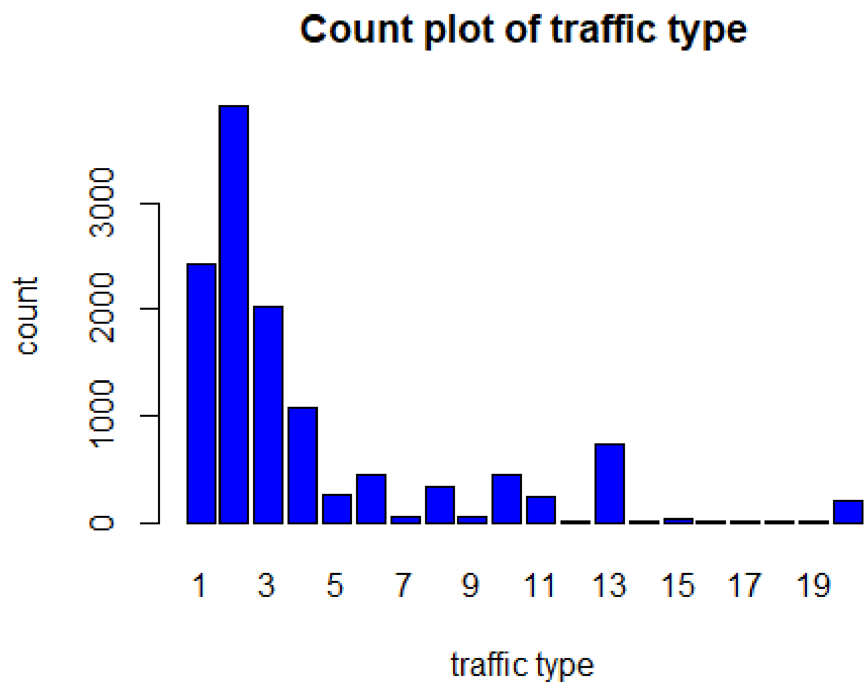
Browser 2 was the most used browser

```
#count plot of region
barplot(table(df$region), col="blue",
        main="Count plot of region",
        xlab = "region", ylab="count")
```



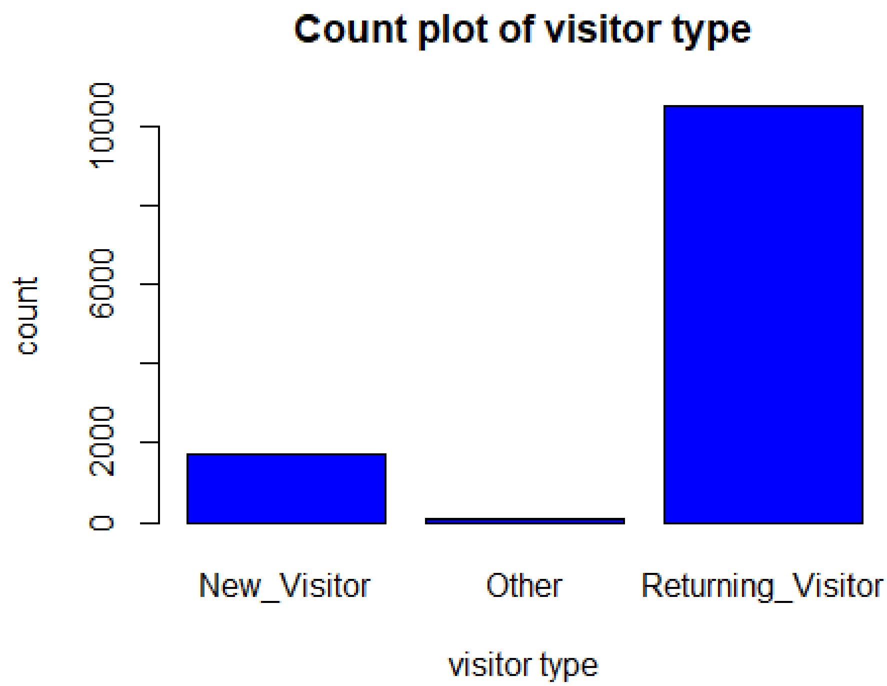
Region 1 was the most represented

```
#count plot of traffictype
barplot(table(df$traffictype), col="blue",
        main="Count plot of traffic type",
        xlab = "traffic type", ylab="count")
```



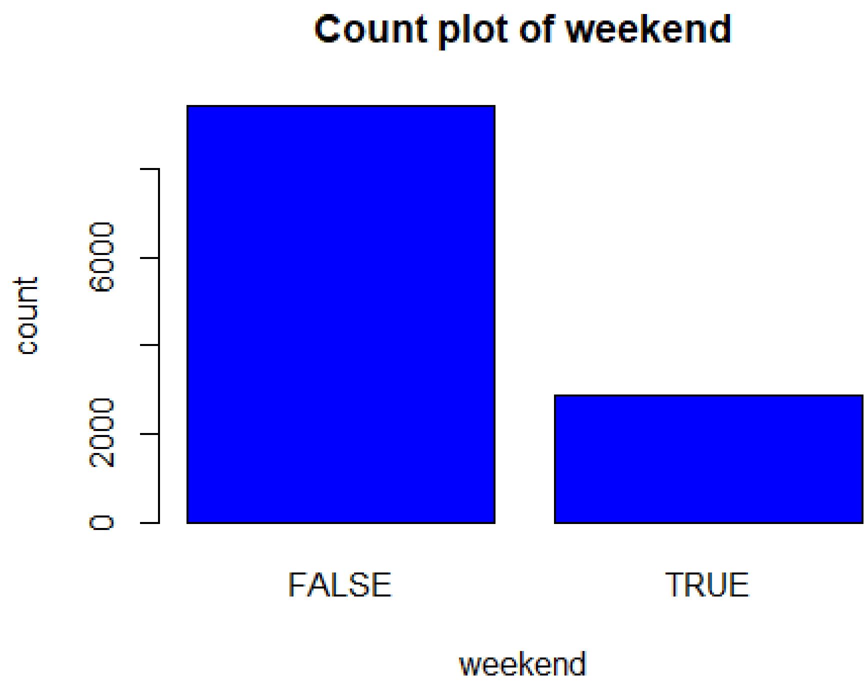
traffic type 2 was the most common

```
#count plot of visitortype
barplot(table(df$visitortype), col="blue",
        main="Count plot of visitor type",
        xlab = "visitor type", ylab="count")
```



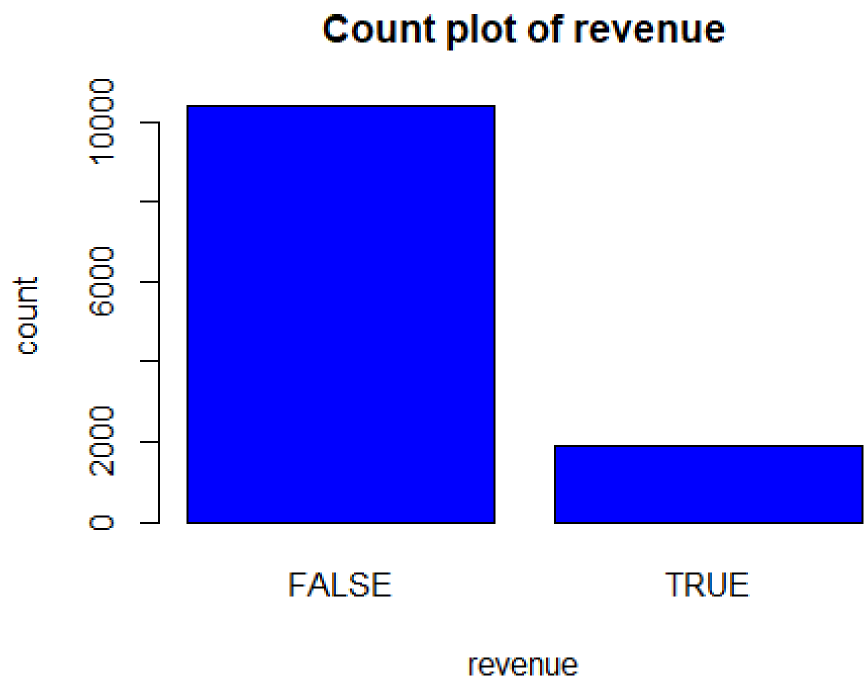
Most visitors were returning visitors

```
#count plot of weekend
barplot(table(df$weekend), col="blue",
        main="Count plot of weekend",
        xlab = "weekend", ylab="count")
```



Most visits were not during the weekend

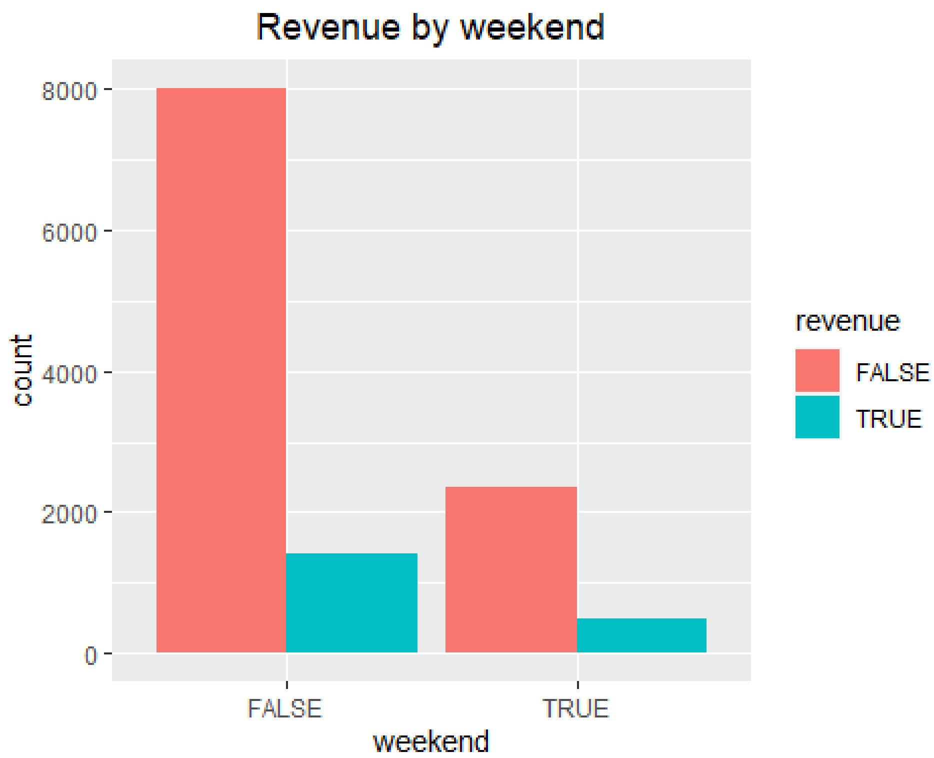
```
#count plot of revenue
barplot(table(df$revenue), col="blue",
        main="Count plot of revenue",
        xlab = "revenue", ylab="count")
```



Most site visits did not result in revenue generation (did not end in a transaction)

Bivariate Analysis

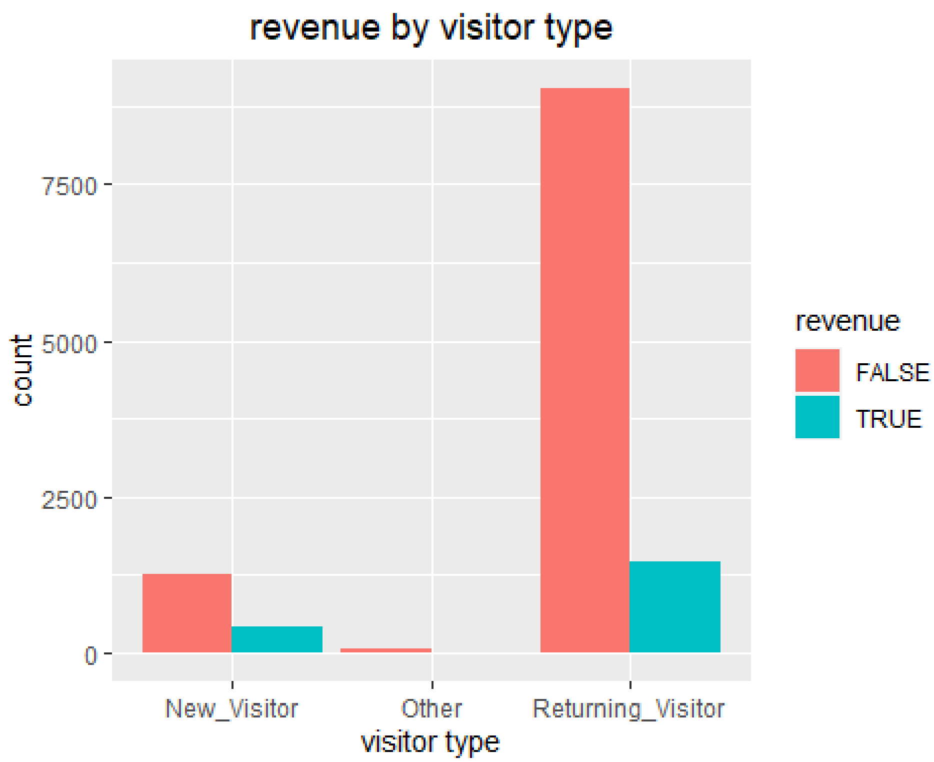
```
#loading library to use functions
library("dplyr")

#plotting revenue by weekend
ggplot() + geom_bar(
    data=df,
    aes(x=factor(weekend), fill = factor(revenue)
    ), position="dodge") + labs(title = "Revenue by weekend",
           y="count", x="weekend", fill="revenue") + theme(plot.title =
element_text(hjust=0.5))
```




```
prop.table(table(df$weekend, df$revenue), 1)

##
##              FALSE      TRUE
##   FALSE 0.8504405 0.1495595
##   TRUE  0.8256464 0.1743536

#rows false true represent weekend
```

The proportion of visits that generated revenue during weekends (0.17) was higher than revenue producing visits during the weekdays (0.14)

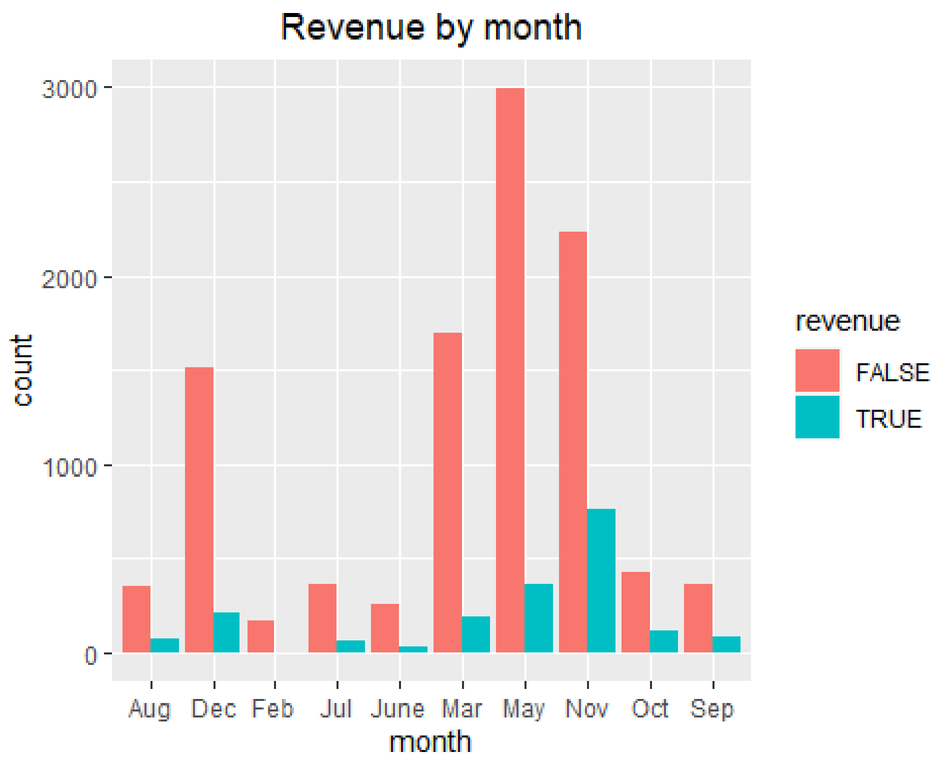
```
table(df$weekend, df$revenue)

##
##          FALSE TRUE
##   FALSE  8012 1409
##   TRUE   2363  499

#revenue by visitortype
ggplot() + geom_bar(
    data=df,
    aes(x=factor(visitortype), fill = factor(revenue)
    ), position="dodge") + labs(title = "revenue by visitor type",
          y="count", x="visitor type", fill="revenue") + theme(plot.title =
element_text(hjust=0.5))
```




```
prop.table(table(df$visitortype, df$revenue), 1)

##
##                          FALSE      TRUE
##   New_Visitor       0.7508855 0.2491145
##   Other             0.8117647 0.1882353
##   Returning_Visitor 0.8600533 0.1399467
```

The proportion of revenue producing visits was highest among new visitors (0.24).

```
#revenue by month
ggplot() + geom_bar(
    data=df,
    aes(x=factor(month), fill = factor(revenue)
    ), position="dodge") + labs(title = "Revenue by month",
          y="count", x="month", fill="revenue") + theme(plot.title =
element_text(hjust=0.5))
```




```
prop.table(table(df$month, df$revenue), 1)
```

```
##
##              FALSE       TRUE
##   Aug  0.82448037 0.17551963
##   Dec  0.87492762 0.12507238
##   Feb  0.98245614 0.01754386
##   Jul  0.84686775 0.15313225
##   June 0.89930556 0.10069444
##   Mar  0.89808917 0.10191083
##   May  0.89127197 0.10872803
##   Nov  0.74624374 0.25375626
##   Oct  0.79052823 0.20947177
##   Sep  0.80803571 0.19196429
```

The month with the highest proportion of revenue generating visits was November (0.25).

Scatterplots of continuous columns

```
#continuous columns
contin
```

```
## [1] "administrative"          "administrative_duration"
## [3] "informational"           "informational_duration"
## [5] "productrelated"          "productrelated_duration"
## [7] "bouncerates"             "exitrates"
## [9] "pagevalues"
```

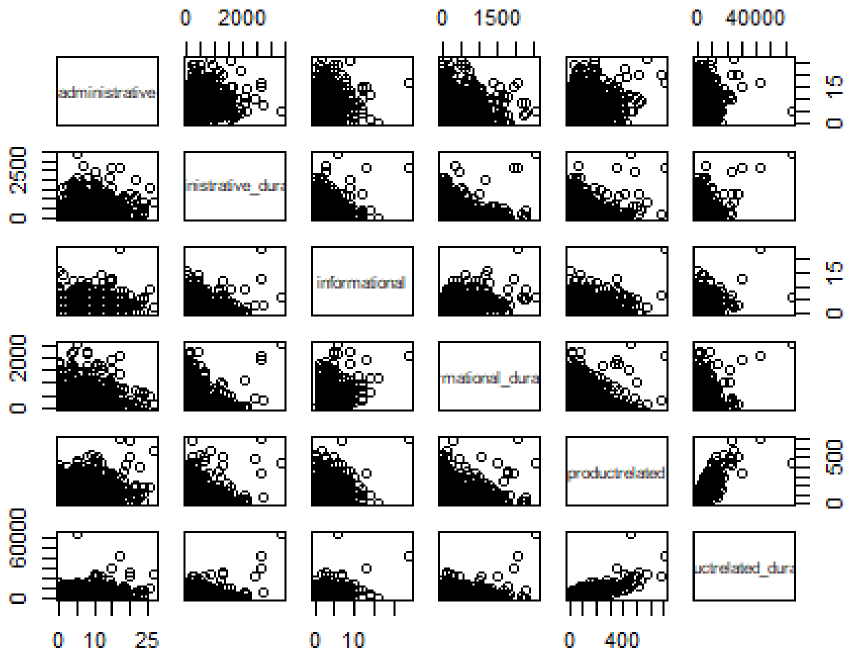
```r
#creating dataframe that containing the continuous variables
scatterp = subset(df, select = c("administrative"
,"administrative_duration", "informational",
"informational_duration",  "productrelated",
"productrelated_duration"))
head(scatterp)
```

```
##   administrative administrative_duration informational
informational_duration
## 1              0                       0             0
0
## 2              0                       0             0
0
## 4              0                       0             0
0
## 5              0                       0             0
0
## 6              0                       0             0
0
## 9              0                       0             0
0
##   productrelated productrelated_duration
## 1              1                0.000000
## 2              2               64.000000
## 4              2                2.666667
## 5             10              627.500000
## 6             19              154.216667
## 9              2               37.000000
```

```r
#loading library for pair plot
library(GGally)
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```r
#plotting scatterplots of continuous variables
plot(scatterp)
```



There are is a positive correlation between administrative (number of page type visited in a session) and administrative duration (duration on said page type). Similarly, between informational and informational duration, and product related and product related duration.

Correlation matrix

```
str(df)

## 'data.frame':    12283 obs. of  18 variables:
##  $ administrative         : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ administrative_duration: num  0 0 0 0 0 0 0 0 0 0 ...
##  $ informational          : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ informational_duration : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ productrelated         : int  1 2 2 10 19 2 3 3 16 7 ...
##  $ productrelated_duration: num  0 64 2.67 627.5 154.22 ...
##  $ bouncerates            : num  0.2 0 0.05 0.02 0.0158 ...
##  $ exitrates              : num  0.2 0.1 0.14 0.05 0.0246 ...
##  $ pagevalues             : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ specialday             : num  0 0 0 0 0 0.8 0.4 0 0.4 0 ...
##  $ month                  : chr  "Feb" "Feb" "Feb" "Feb" ...
##  $ operatingsystems       : int  1 2 3 3 2 2 2 1 1 1 ...
##  $ browser                : int  1 2 2 3 2 2 4 1 1 1 ...
##  $ region                 : int  1 1 2 1 1 2 1 3 4 1 ...
##  $ traffictype            : int  1 2 4 4 3 3 2 3 3 3 ...
##  $ visitortype            : chr  "Returning_Visitor" "Returning_Visitor"
"Returning_Visitor" "Returning_Visitor" ...
```

```
##  $ weekend                : logi  FALSE FALSE FALSE TRUE FALSE FALSE ...
##  $ revenue                : logi  FALSE FALSE FALSE FALSE FALSE FALSE ...
```

```r
#converting categorical to numerical
#removing timestamp column
#dataframe for correlation matrix
enc_df <- copy(df)

enc_df$month <- as.numeric(factor(enc_df$month))
enc_df$weekend <- as.numeric(factor(enc_df$weekend))
enc_df$visitortype <- as.numeric(factor(enc_df$visitortype))
enc_df$revenue <- as.numeric(factor(enc_df$revenue))

#checking that datatype conversion worked
str(enc_df)
```

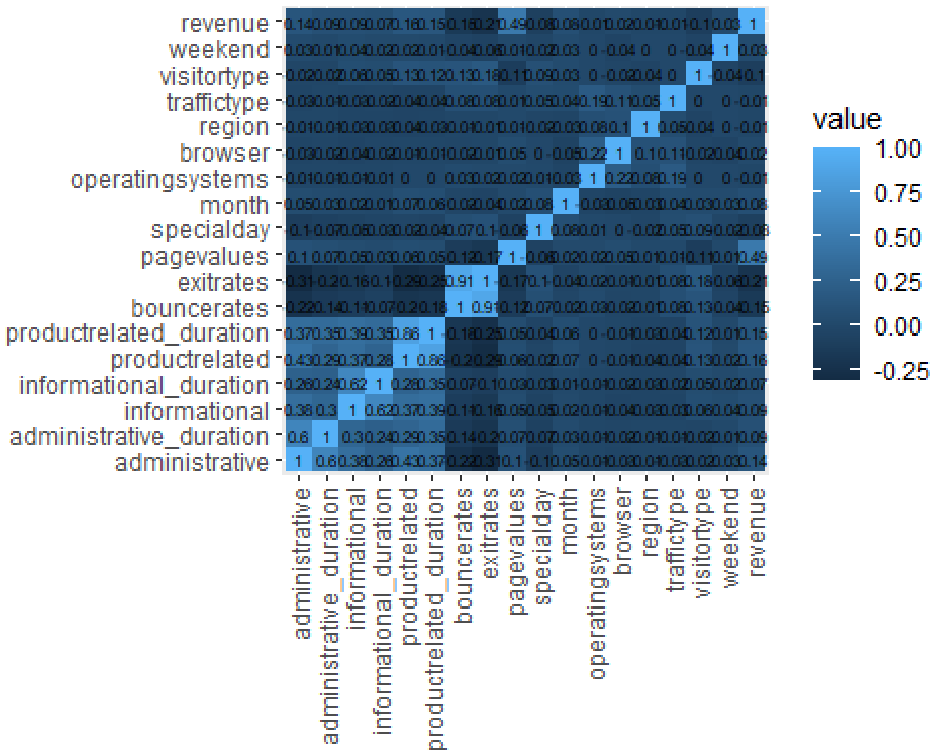
```
## 'data.frame':    12283 obs. of  18 variables:
##  $ administrative         : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ administrative_duration: num  0 0 0 0 0 0 0 0 0 0 ...
##  $ informational          : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ informational_duration : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ productrelated         : int  1 2 2 10 19 2 3 3 16 7 ...
##  $ productrelated_duration: num  0 64 2.67 627.5 154.22 ...
##  $ bouncerates            : num  0.2 0 0.05 0.02 0.0158 ...
##  $ exitrates              : num  0.2 0.1 0.14 0.05 0.0246 ...
##  $ pagevalues             : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ specialday             : num  0 0 0 0 0 0.8 0.4 0 0.4 0 ...
##  $ month                  : num  3 3 3 3 3 3 3 3 3 3 ...
##  $ operatingsystems       : int  1 2 3 3 2 2 2 1 1 1 ...
##  $ browser                : int  1 2 2 3 2 2 4 1 1 1 ...
##  $ region                 : int  1 1 2 1 1 2 1 3 4 1 ...
##  $ traffictype            : int  1 2 4 4 3 3 2 3 3 3 ...
##  $ visitortype            : num  3 3 3 3 3 3 3 3 3 3 ...
##  $ weekend                : num  1 1 1 2 1 1 1 1 1 1 ...
##  $ revenue                : num  1 1 1 1 1 1 1 1 1 1 ...
```

```r
library(reshape2)
```

```
##
## Attaching package: 'reshape2'

## The following object is masked from 'package:tidyr':
##
##     smiths

## The following objects are masked from 'package:data.table':
##
##     dcast, melt
```

```r
#plotting the correlation heatmap
datam = melt(round(cor(enc_df),2))
```

```
ggplot(data=datam, aes(x=Var1, y=Var2, fill=value)) + geom_tile() +
geom_text(aes(Var2, Var1, label=value), color="black",size=2) +
theme(axis.text.x=element_text(angle=90,vjust=0.5,hjust=1), axis.title.x =
element_blank(), axis.title.y = element_blank())
```



According to the correlation heatmap above, revenue seems to be most strongly correlated to page values, exit rates, and product-related, in that order.

Variables with strongest positive correlations: exit rates and bounce rates, product related and product related duration.

Modelling

```
library(caret)
library(factoextra)

## Welcome! Want to learn more? See two factoextra-related books at
https://goo.gl/ve3WBa

# library("psych")
```

1. K-Means clustering

```
#describe
describe(enc_df)

##                           vars     n    mean      sd median trimmed     mad
min
## administrative               1 12283    2.32    3.33   1.00    1.64    1.48
```

```
0
## administrative_duration    2 12283   81.13  177.05   8.00   42.37  11.86
0
## informational              3 12283    0.51    1.27   0.00    0.18   0.00
0
## informational_duration     4 12283   34.60  141.00   0.00    3.63   0.00
0
## productrelated             5 12283   31.85   44.52  18.00   22.86  19.27
0
## productrelated_duration    6 12283 1199.25 1915.94 602.50  824.43 744.39
0
## bouncerates                7 12283    0.02    0.05   0.00    0.01   0.00
0
## exitrates                  8 12283    0.04    0.05   0.03    0.03   0.02
0
## pagevalues                 9 12283    5.91   18.60   0.00    1.31   0.00
0
## specialday                10 12283    0.06    0.20   0.00    0.00   0.00
0
## month                     11 12283    6.17    2.37   7.00    6.36   1.48
1
## operatingsystems          12 12283    2.12    0.91   2.00    2.06   0.00
1
## browser                   13 12283    2.36    1.72   2.00    2.00   0.00
1
## region                    14 12283    3.15    2.40   3.00    2.79   2.97
1
## traffictype               15 12283    4.07    4.03   2.00    3.22   1.48
1
## visitortype               16 12283    2.72    0.69   3.00    2.90   0.00
1
## weekend                   17 12283    1.23    0.42   1.00    1.17   0.00
1
## revenue                   18 12283    1.16    0.36   1.00    1.07   0.00
1
##                              max    range  skew kurtosis    se
## administrative             27.00    27.00  1.95     4.67  0.03
## administrative_duration  3398.75  3398.75  5.61    50.37  1.60
## informational              24.00    24.00  4.03    26.82  0.01
## informational_duration   2549.38  2549.38  7.56    75.98  1.27
## productrelated            705.00   705.00  4.34    31.14  0.40
## productrelated_duration 63973.52 63973.52  7.26   136.90 17.29
## bouncerates                0.20     0.20  3.00     8.10  0.00
## exitrates                  0.20     0.20  2.17     4.18  0.00
## pagevalues               361.76   361.76  6.37    65.36  0.17
## specialday                 1.00     1.00  3.30     9.89  0.00
## month                     10.00     9.00 -0.83    -0.37  0.02
## operatingsystems           8.00     7.00  2.07    10.47  0.01
## browser                   13.00    12.00  3.24    12.76  0.02
## region                     9.00     8.00  0.98    -0.15  0.02
```

```
## traffictype                20.00     19.00  1.96     3.47  0.04
## visitortype                 3.00      2.00 -2.06     2.27  0.01
## weekend                     2.00      1.00  1.26    -0.40  0.00
## revenue                     2.00      1.00  1.90     1.62  0.00
```

```
#scaling the variables
enc_df_sc <- copy(enc_df)
for (col in colnames(enc_df_sc)){
  enc_df_sc[col] <- scale(enc_df_sc[col])
}
summary(enc_df_sc)
```

```
##  administrative.administrative
administrative_duration.administrative_duration
##  Min.   :-0.698879             Min.   :-0.458219
##  1st Qu.:-0.698879             1st Qu.:-0.458219
##  Median :-0.398139             Median :-0.413033
##  Mean   : 0.000000             Mean   : 0.000000
##  3rd Qu.: 0.504082             3rd Qu.: 0.072432
##  Max.   : 7.421108             Max.   :18.738678
##  informational.informational informational_duration.informational_duration
##  Min.   :-0.397231           Min.   :-0.245398
##  1st Qu.:-0.397231           1st Qu.:-0.245398
##  Median :-0.397231           Median :-0.245398
##  Mean   : 0.000000           Mean   : 0.000000
##  3rd Qu.:-0.397231           3rd Qu.:-0.245398
##  Max.   :18.468643           Max.   :17.834955
##  productrelated.productrelated
productrelated_duration.productrelated_duration
##  Min.   :-0.715308             Min.   :-0.62594
##  1st Qu.:-0.558080             1st Qu.:-0.52828
##  Median :-0.311008             Median :-0.31147
##  Mean   : 0.000000             Mean   : 0.00000
##  3rd Qu.: 0.138213             3rd Qu.: 0.14179
##  Max.   :15.119758             Max.   :32.76429
##  bouncerates.bouncerates exitrates.exitrates pagevalues.pagevalues
##  Min.   :-0.455556       Min.   :-0.888394   Min.   :-0.317832
##  1st Qu.:-0.455556       1st Qu.:-0.590549   1st Qu.:-0.317832
##  Median :-0.391031       Median :-0.367165   Median :-0.317832
##  Mean   : 0.000000       Mean   : 0.000000   Mean   : 0.000000
##  3rd Qu.:-0.106045       3rd Qu.: 0.154063   3rd Qu.:-0.317832
##  Max.   : 3.738574       Max.   : 3.281431   Max.   :19.131465
##  specialday.specialday     month.month
operatingsystems.operatingsystems
##  Min.   :-0.309018     Min.   :-2.1781515   Min.   :-1.233186
##  1st Qu.:-0.309018     1st Qu.:-0.0703571   1st Qu.:-0.136356
##  Median :-0.309018     Median : 0.3512018   Median :-0.136356
##  Mean   : 0.000000     Mean   : 0.0000000   Mean   : 0.000000
##  3rd Qu.:-0.309018     3rd Qu.: 0.7727607   3rd Qu.: 0.960474
##  Max.   : 4.713039     Max.   : 1.6158785   Max.   : 6.444625
```

```
##    browser.browser      region.region     traffictype.traffictype
##  Min.   :-0.790209   Min.   :-0.8938929   Min.   :-0.763141  
##  1st Qu.:-0.207887   1st Qu.:-0.8938929   1st Qu.:-0.514720  
##  Median :-0.207887   Median :-0.0612469   Median :-0.514720  
##  Mean   : 0.000000   Mean   : 0.0000000   Mean   : 0.000000  
##  3rd Qu.:-0.207887   3rd Qu.: 0.3550761   3rd Qu.:-0.017879  
##  Max.   : 6.197651   Max.   : 2.4366911   Max.   : 3.956854  
##  visitortype.visitortype   weekend.weekend      revenue.revenue   
##  Min.   :-2.4820823      Min.   :-0.5511485   Min.   :-0.4288224  
##  1st Qu.: 0.4086793      1st Qu.:-0.5511485   1st Qu.:-0.4288224  
##  Median : 0.4086793      Median :-0.5511485   Median :-0.4288224  
##  Mean   : 0.0000000      Mean   : 0.0000000   Mean   : 0.0000000  
##  3rd Qu.: 0.4086793      3rd Qu.:-0.5511485   3rd Qu.:-0.4288224  
##  Max.   : 0.4086793      Max.   : 1.8142453   Max.   : 2.3317779
```

```
 set.seed(123)
grouping <- kmeans(enc_df_sc, 3)
print("Cluster sizes:")
```

```
## [1] "Cluster sizes:"
```

```
grouping$size
```

```
## [1] 1030 9596 1657
```

```
print("Within cluster sum of squares")
```

```
## [1] "Within cluster sum of squares"
```

```
grouping$withinss
```

```
## [1]  10553.72 116122.10  50696.39
```

```
print("Total sum of squares (including between ss)")
```

```
## [1] "Total sum of squares (including between ss)"
```

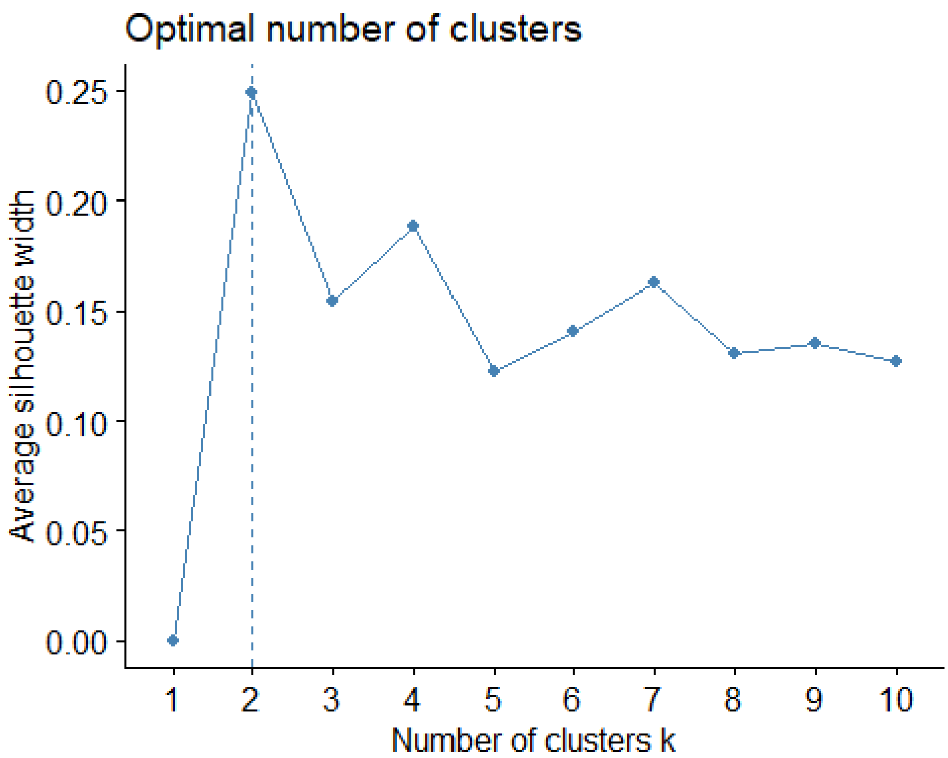
```
grouping$tot.withinss
```

```
## [1] 177372.2
```

```
#
print("***************************************************************************")
# # grouping$cluster
# subset(grouping, select=!cluster)
```

Challenging the solution

```
# Determining Optimal clusters (k) Using Average Silhouette Method
#A good silhouette score is usually near 1 and attempts to minimise within cluster variance while maximising the between cluster variance.

fviz_nbclust(x = enc_df_sc,FUNcluster = kmeans, method = 'silhouette' )
```

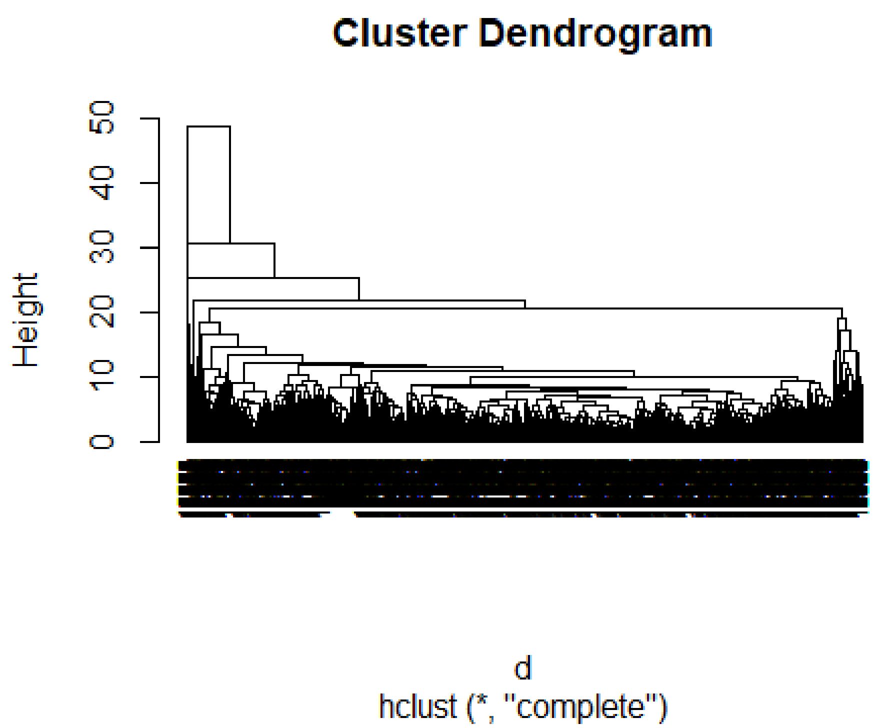


Optimal number of clusters determined to be 2.

```
#grouping with value identified above
set.seed(123)
grouping <- kmeans(enc_df_sc, 2)
print("Cluster sizes:")

## [1] "Cluster sizes:"

grouping$size

## [1] 10178  2105

print("Within cluster sum of squares")

## [1] "Within cluster sum of squares"

grouping$withinss

## [1] 135765.64  61258.13

print("Total sum of squares (including between ss)")

## [1] "Total sum of squares (including between ss)"

grouping$tot.withinss

## [1] 197023.8
```

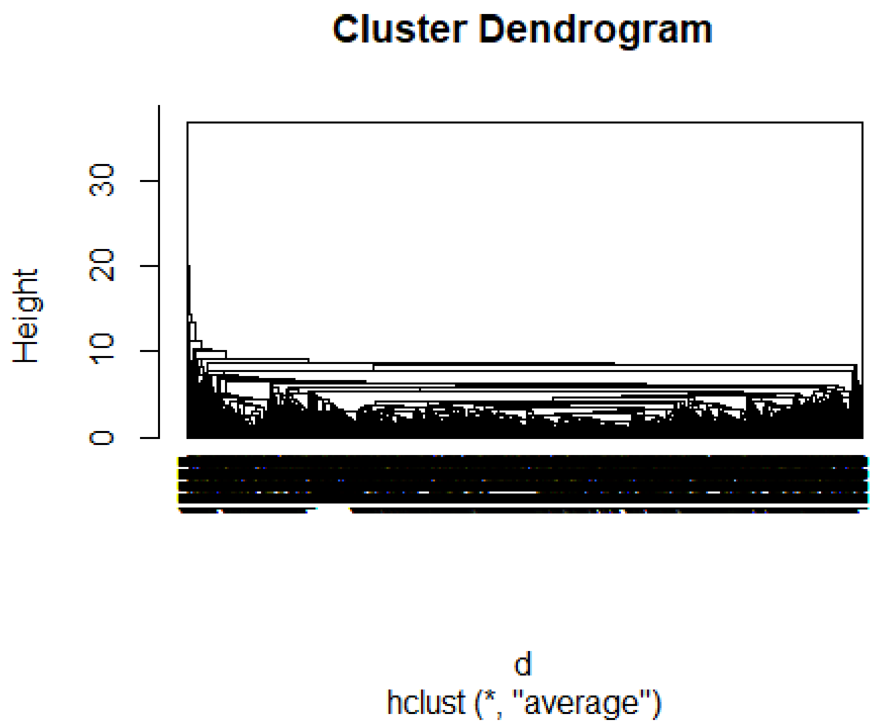
2. Hierarchical clustering

```
# d will be the first argument in the hclust() function distance matrix
# ---
#using scaled df
d <- dist(enc_df_sc, method = "euclidean")

# hierarchical clustering using the complete linkage method
# ---
#
res.hc <- hclust(d, method = "complete" )

plot(res.hc, cex = 0.6, hang = -1)
```

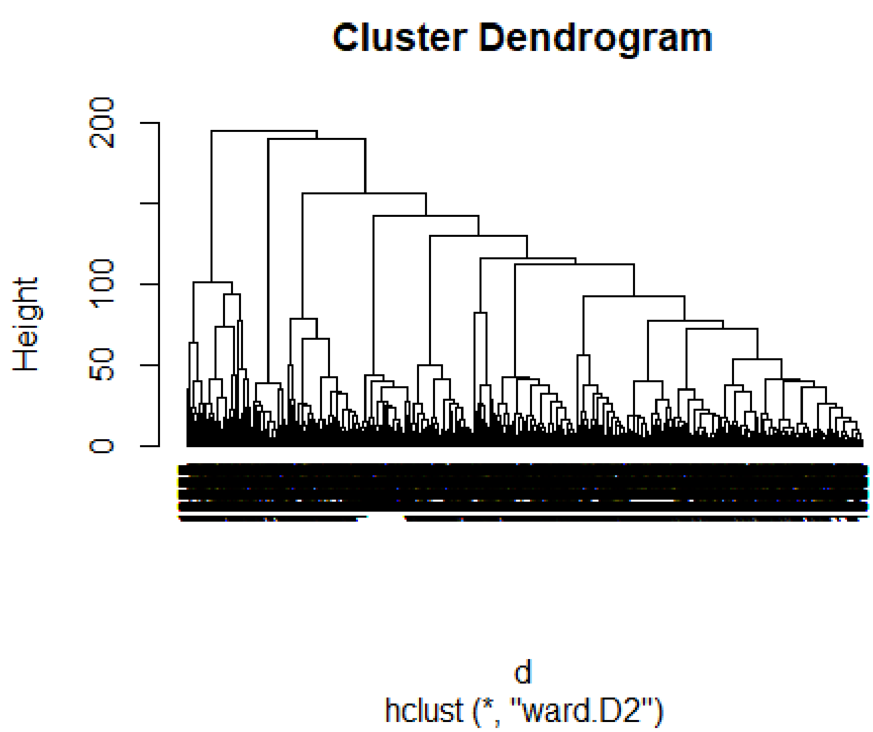


Challenging the approach

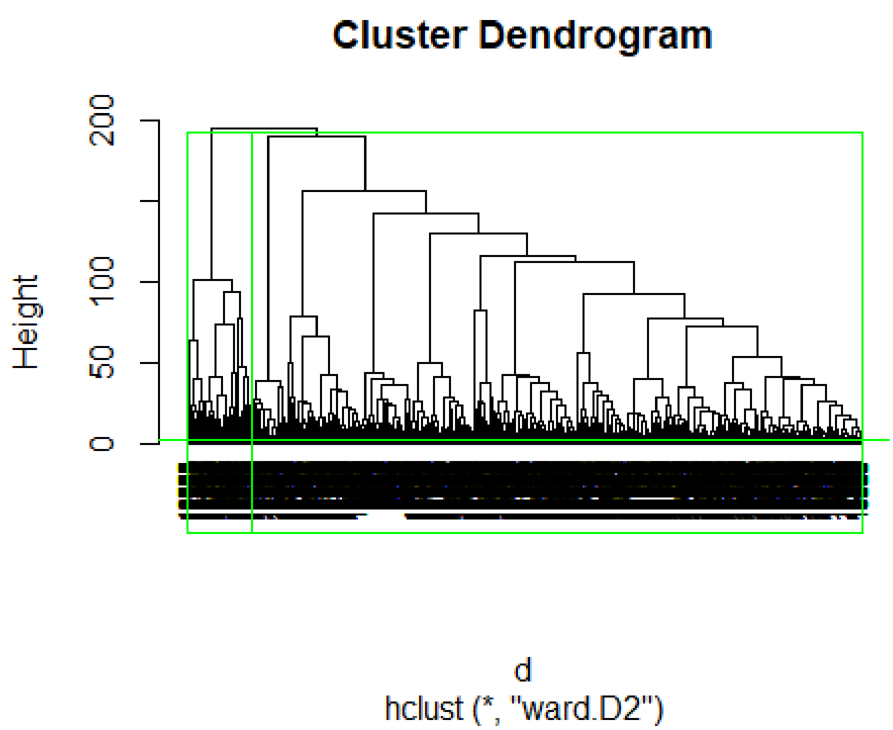
```
res.hc <- hclust(d, method = "average" )
plot(res.hc, cex = 0.6, hang = -1)
```




```
res.hc <- hclust(d, method = "ward.D2" )
plot(res.hc, cex = 0.6, hang = -1)
```



```r
# Choosing no. of clusters to highlight
# Cutting tree by height
# res.hc <- hclust(d, method = "ward.D2" )

# cutting to 2 clusters
two <- cutree(res.hc, k = 2 )

table(two)

## two
##     1     2
## 11109  1174

#dendrogram showing borders of cutting into two clusters. wards method
produces clearest dendrogram

plot(res.hc, cex = 0.6, hang = -1)
abline(h = 1.9, col = "green")
rect.hclust(res.hc, k = 2, border = "green")
```



Group characteristics comparisons - k means clusters (bivariate analysis)

K means identified 2 clusters as optimal number using the average silhouette score. Therefore, further analysis will be carried out on the 2 customer groups that were identified while using kmeans.

```
#summary of the clustering
grouping

## K-means clustering with 2 clusters of sizes 10178, 2105
##
## Cluster means:
##   administrative administrative_duration informational
informational_duration
## 1     -0.2869788              -0.2330162    -0.2615324             -
0.2002143
## 2      1.3875870               1.1266695     1.2645494
0.9680672
##   productrelated productrelated_duration bouncerates  exitrates
pagevalues
## 1     -0.2573956               -0.239908  0.06822025  0.1021581 -
0.07503018
## 2      1.2445474                1.159992 -0.32985544 -0.4939504
0.36278250
##    specialday       month operatingsystems     browser      region
traffictype
## 1  0.03420667 -0.03212765      0.002660684  0.01358711  0.01238067
0.01990431
## 2 -0.16539455  0.15534217     -0.012864820 -0.06569575 -0.05986245 -
0.09624041
##   visitortype      weekend    revenue
## 1 -0.04348151 -0.009185608 -0.1019877
## 2  0.21023982  0.044413832  0.4931263
##
## Clustering vector:
##     1     2     4     5     6     9    10    11    12    13    14    15
16
##     1     1     1     1     1     1     1     1     1     1     1     1
1
##    18    19    20    21    23    24    26    27    28    29    30    31
32
##     1     1     1     1     1     1     1     1     1     1     1     1
1
##    33    34    35    36    37    38    39    40    41    42    43    44
45
##     1     1     1     1     1     1     1     1     1     1     1     1
1
##    46    47    48    49    52    53    54    55    56    57    58    59
60
##     1     1     1     1     1     1     1     1     1     1     1     1
1
##    61    62    63    64    66    67    68    69    70    71    72    73
74
##     1     1     2     1     1     2     1     1     1     1     1     1
1
##    75    76    77    78    79    80    81    82    83    84    85    86
```

(Excluded the pages only showing which cluster each point belongs to)

```
1
## 12307 12308 12309 12310 12311 12312 12313 12314 12315 12316 12317 12318
12319
##     1     2     1     1     1     2     2     2     1     1     1     1
1
## 12320 12321 12322 12323 12324 12325 12326 12327 12328 12329 12330
##     1     1     1     1     1     1     1     1     1     1     1
##
## Within cluster sum of squares by cluster:
## [1] 135765.64  61258.13
##  (between_SS / total_SS =  10.9 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"
"tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

```
print('*******************************************')
```

```
## [1] "*******************************************"
```

```
#creating df with means of continuous columns by cluster
df_clus_means<- aggregate(subset(df, select=contin),
by=list(cluster=grouping$cluster),mean)
df_clus_means
```

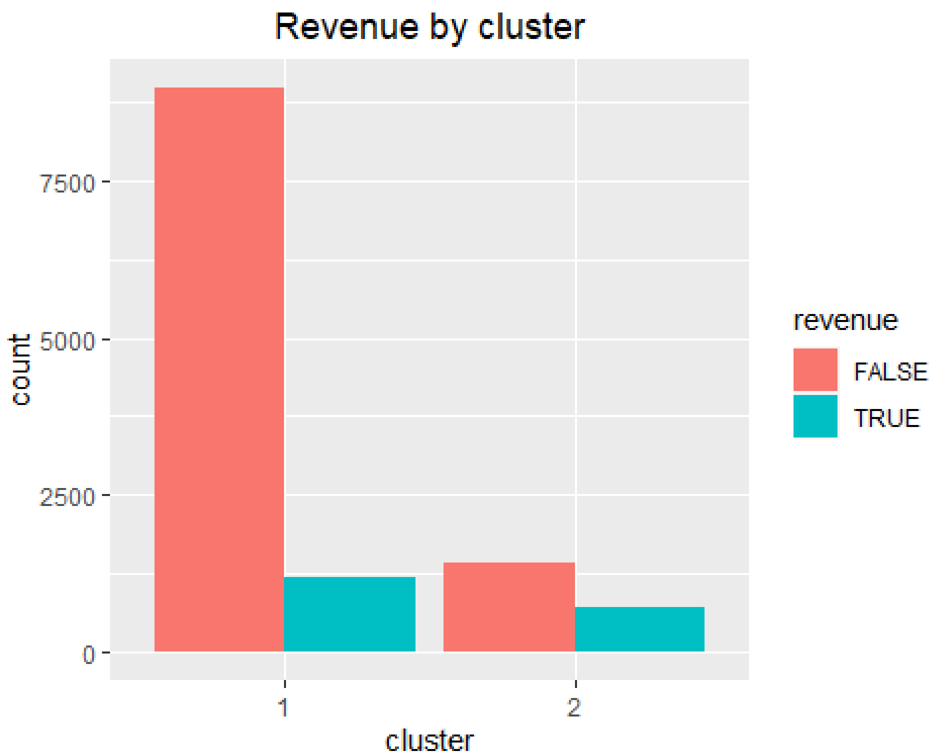
```
##   cluster administrative administrative_duration informational
## 1       1       1.369621                39.87144     0.1726272
## 2       2       6.937767               280.59950     2.1140143
##   informational_duration productrelated productrelated_duration
bouncerates
## 1                6.37106       20.38691                739.6049
0.024976627
## 2              171.10167       87.25558               3421.7229
0.005994113
##    exitrates pagevalues
## 1 0.04751047   4.516205
## 2 0.01891893  12.659674
```

```
#creating dataframe with cluster column and checking that output matches
above
df_clus <- copy(df)
df_clus$cluster <- grouping$cluster
# df_clus
df_clus %>% group_by(cluster) %>%
 summarise(mean_adm=mean(administrative),
mean_col=mean(administrative_duration))
```

```
## # A tibble: 2 × 3
##   cluster mean_adm mean_col
```

```
##      <int>     <dbl>     <dbl>
## 1        1      1.37      39.9
## 2        2      6.94     281.
```

```
#plotting revenue by cluster
ggplot() + geom_bar(
    data=df_clus,
    aes(x=factor(cluster), fill = factor(revenue)
    ), position="dodge") + labs(title = "Revenue by cluster",
        y="count", x="cluster", fill="revenue") + theme(plot.title =
element_text(hjust=0.5))
```



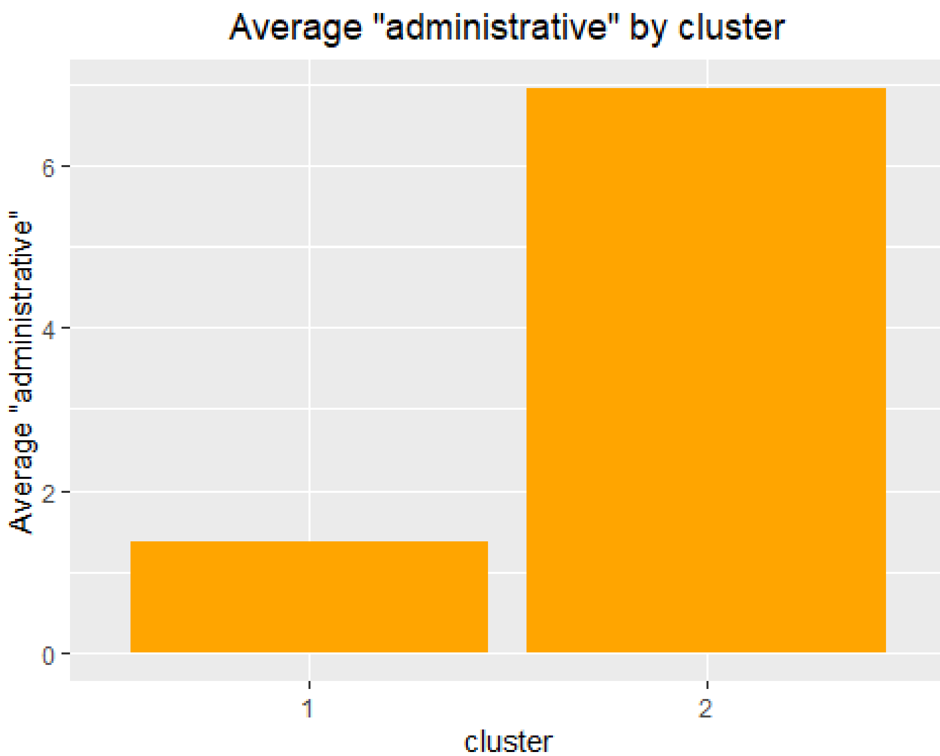

```
prop.table(table(df_clus$cluster, df_clus$revenue), 1)
```

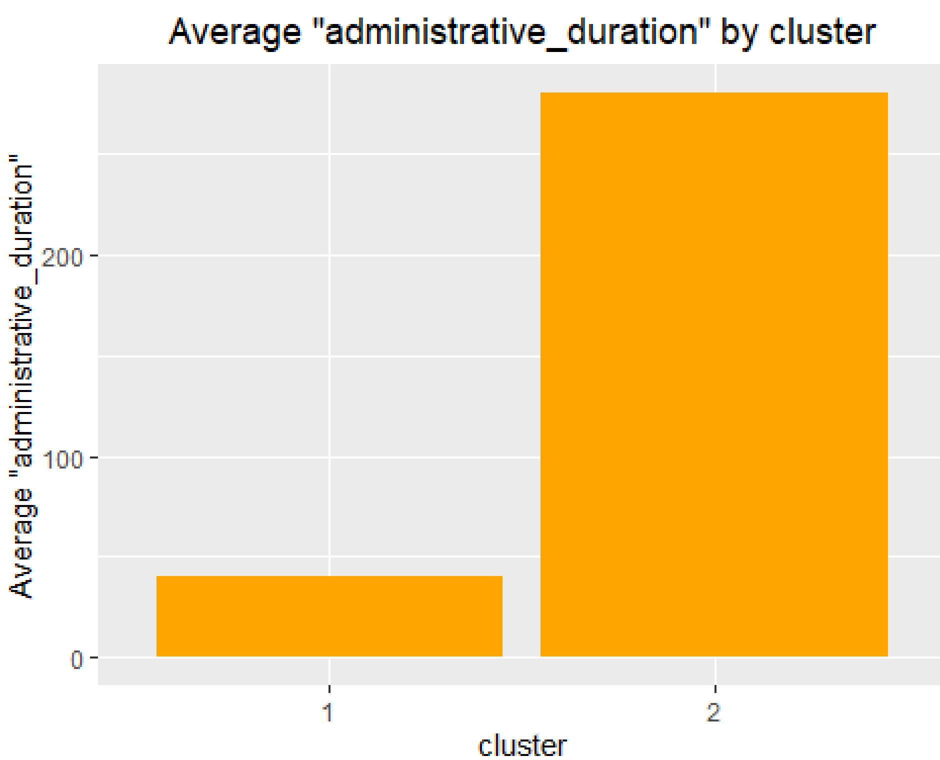
```
##
##          FALSE      TRUE
##    1 0.8816074 0.1183926
##    2 0.6660333 0.3339667
```

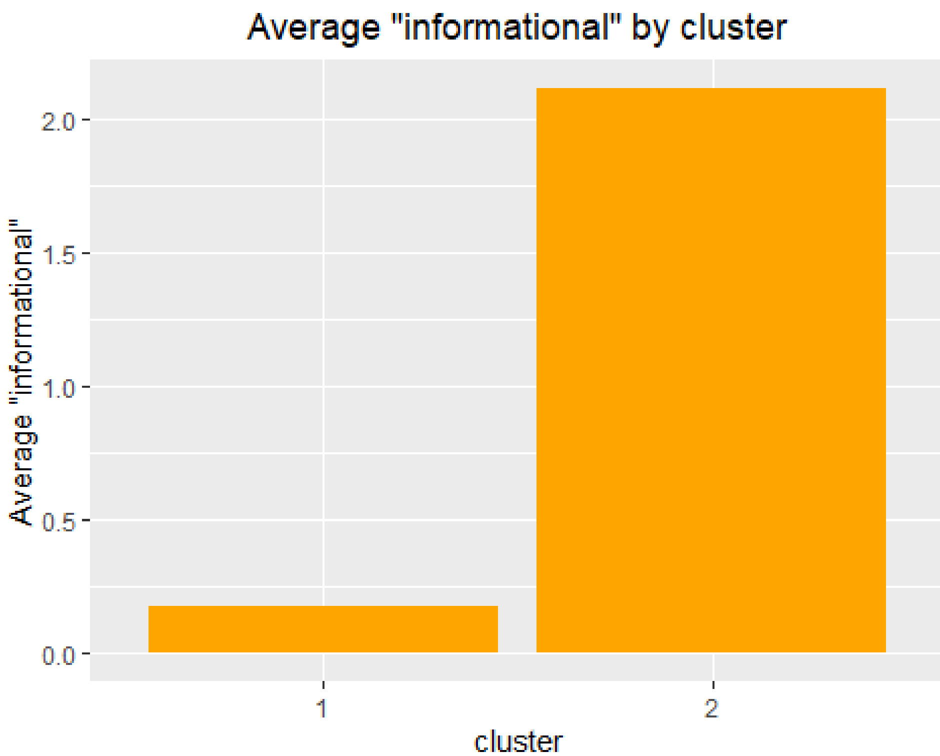
The proportion of customers of cluster 2 who generate revenue (0.33) is higher than the proportion of customers in cluster 1 who generate revenue.

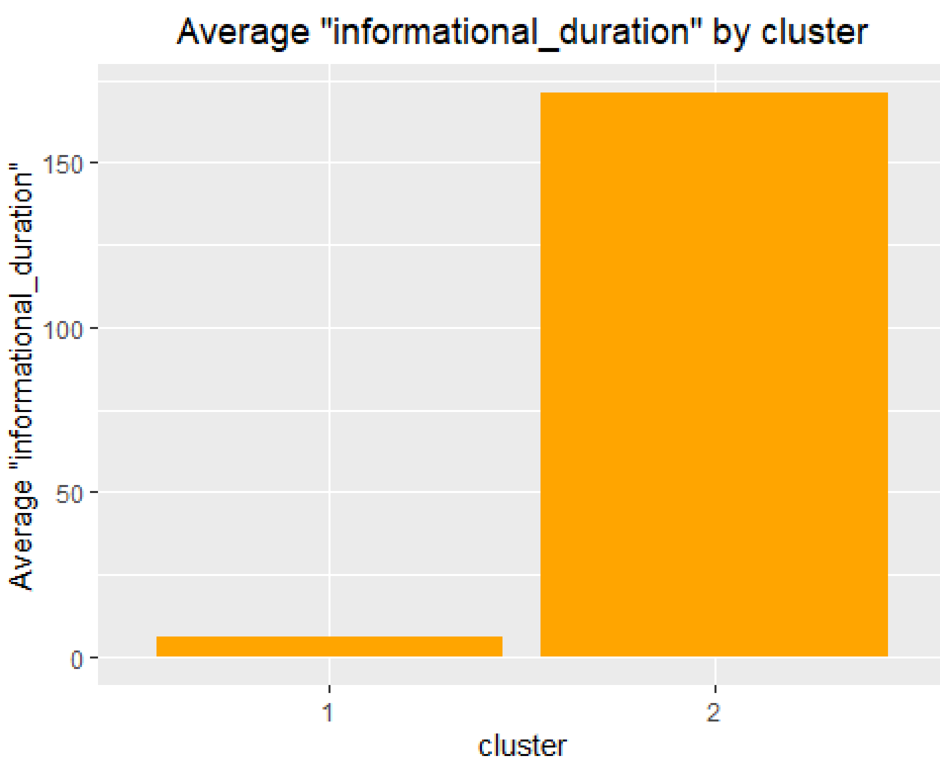
```
library(stringr)

#average values by cluster

for (m in contin){
```

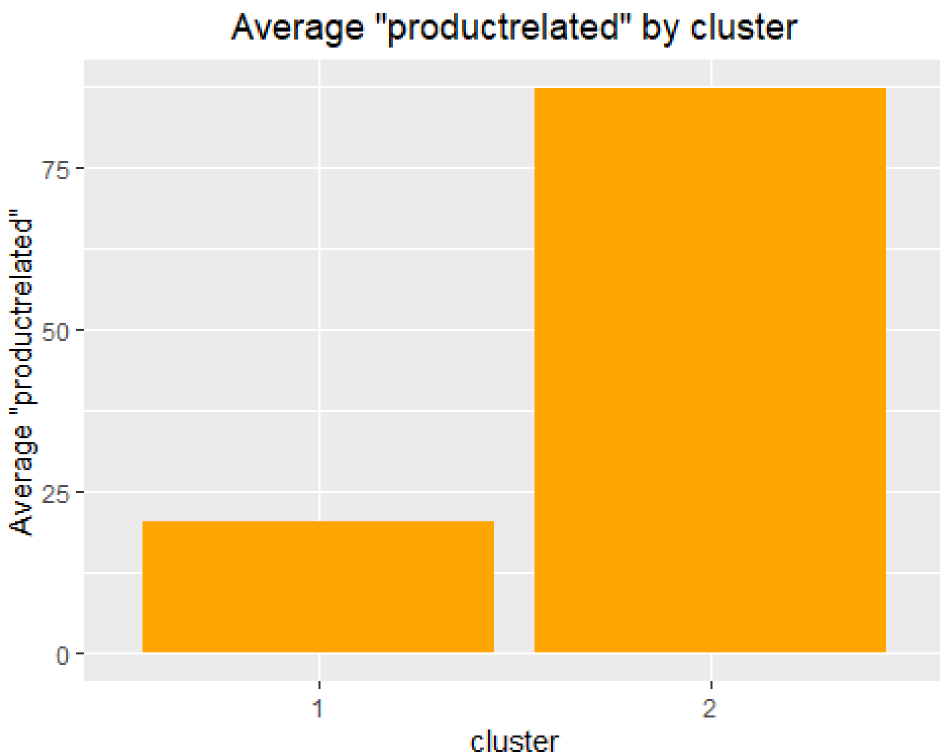
```
  suppressWarnings(print(ggplot() + geom_col(
   data=df_clus_means,
   aes(x=as.factor(cluster), y=df_clus_means[[m]]),
   fill="orange") + labs(title = str_glue('Average "{m}" by cluster'),
   x="cluster", y=str_glue('Average "{m}"')) + theme(plot.title =
  element_text(hjust=0.5))))

}
```

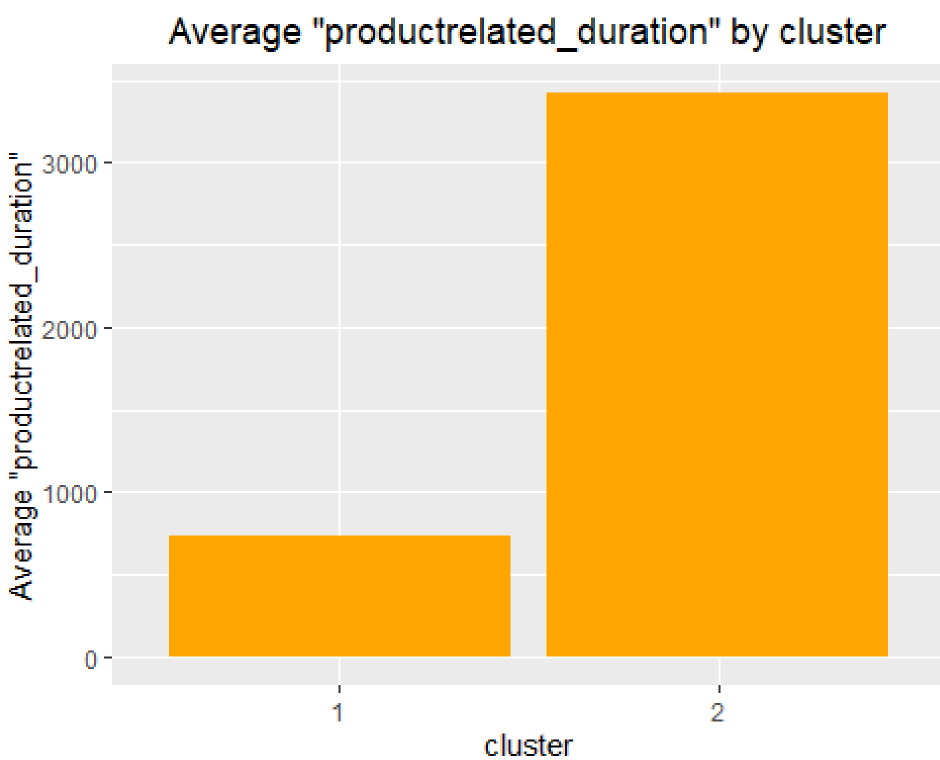

Average "administrative" by cluster
Average "administrative"
0
2
4
6
1
2
cluster

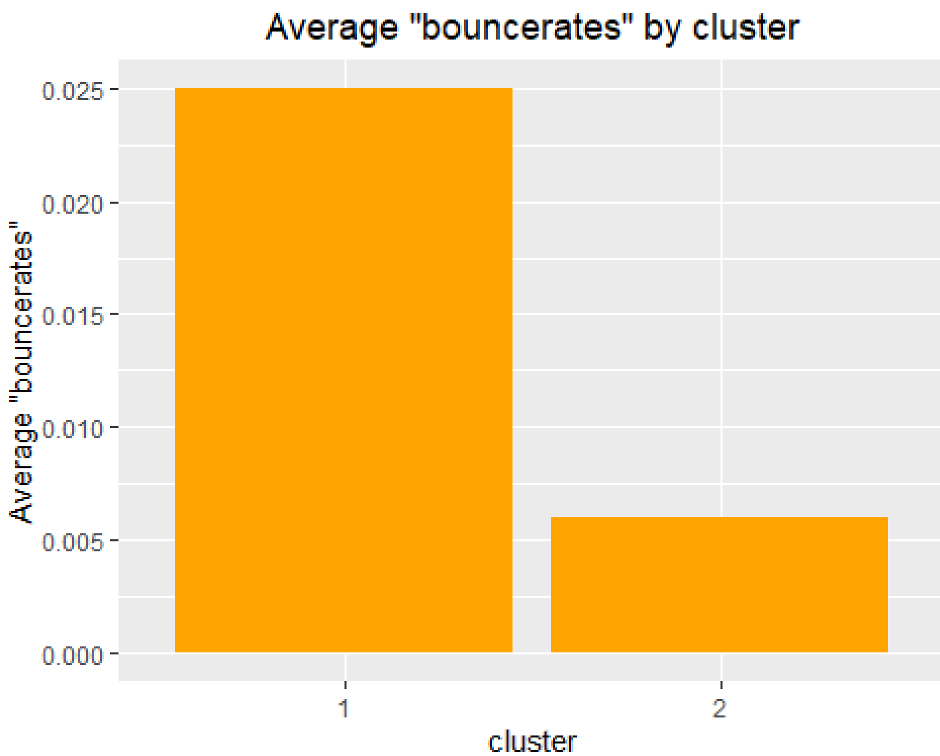

Average "administrative_duration" by cluster
Average "administrative_duration"
0
100
200
1
2
cluster

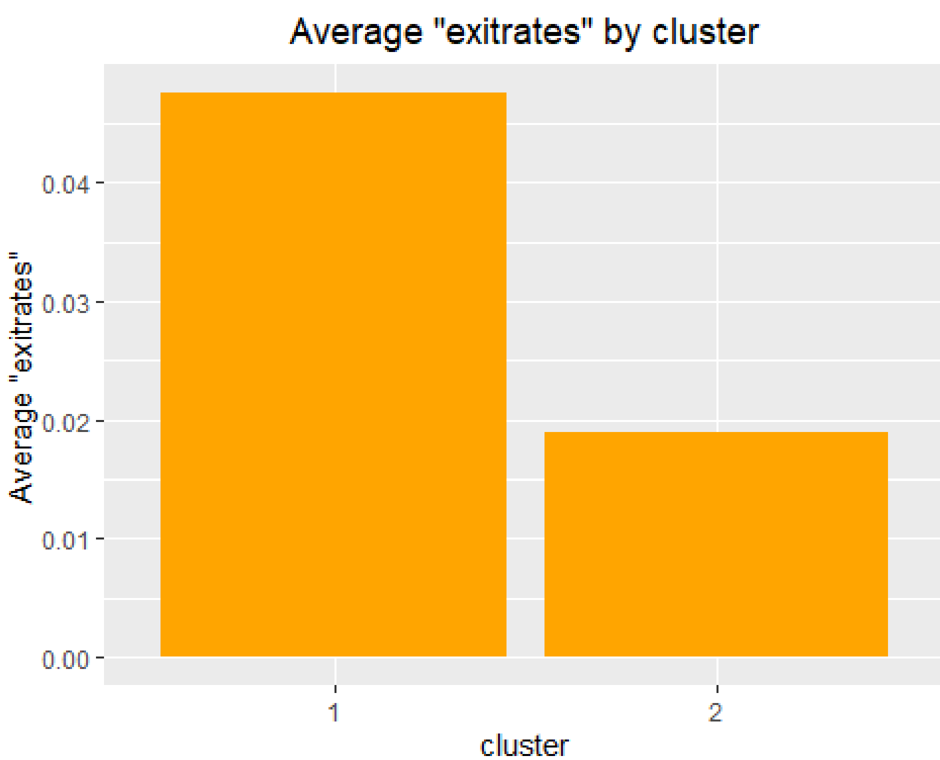

Average "informational" by cluster
Average "informational"
2.0
1.5
1.0
0.5
0.0
1
2
cluster

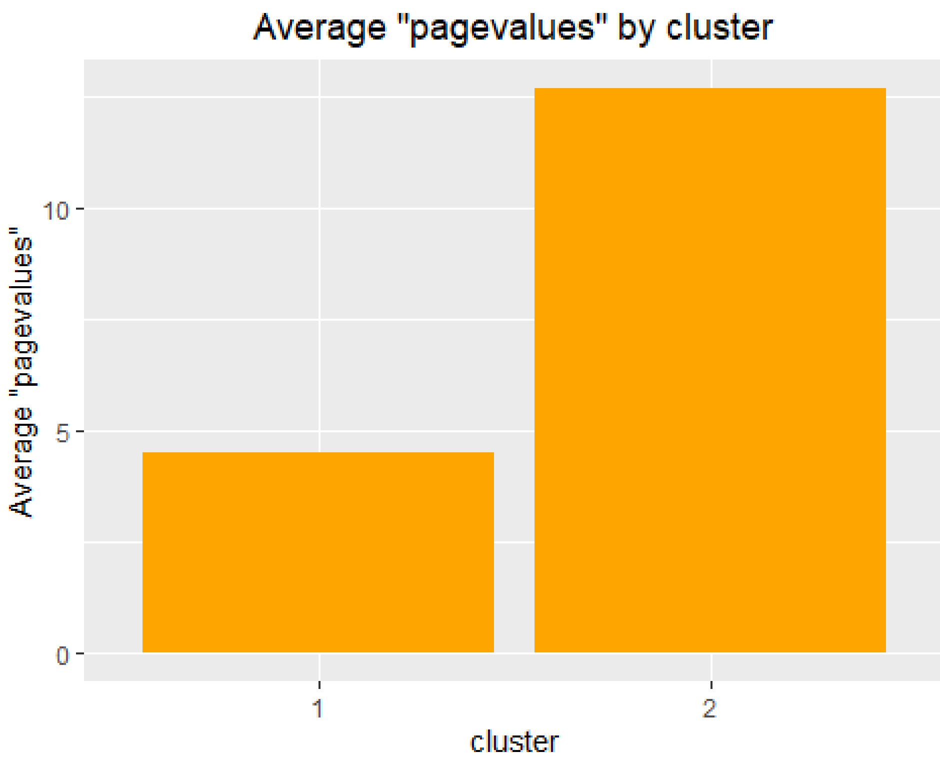

Average "informational_duration" by cluster
Average "informational_duration"
150
100
50
0
1
2
cluster


Average "productrelated" by cluster
Average "productrelated"
0
25
50
75
1
2
cluster


Average "productrelated_duration" by cluster
Average "productrelated_duration"
0
1000
2000
3000
1
2
cluster


Average "bouncerates" by cluster
Average "bouncerates"
0.025
0.020
0.015
0.010
0.005
0.000
1
2
cluster


Average "exitrates" by cluster
Average "exitrates"
0.04
0.03
0.02
0.01
0.00
1
2
cluster



Observations of plots above:

The average number of administrative, informational and product related pages visited in a session, as well as the average durations spent on these different page types, is higher among customers in cluster 2 than in cluster one.

Bouncerates and exit rates are higher among customers in cluster 1.

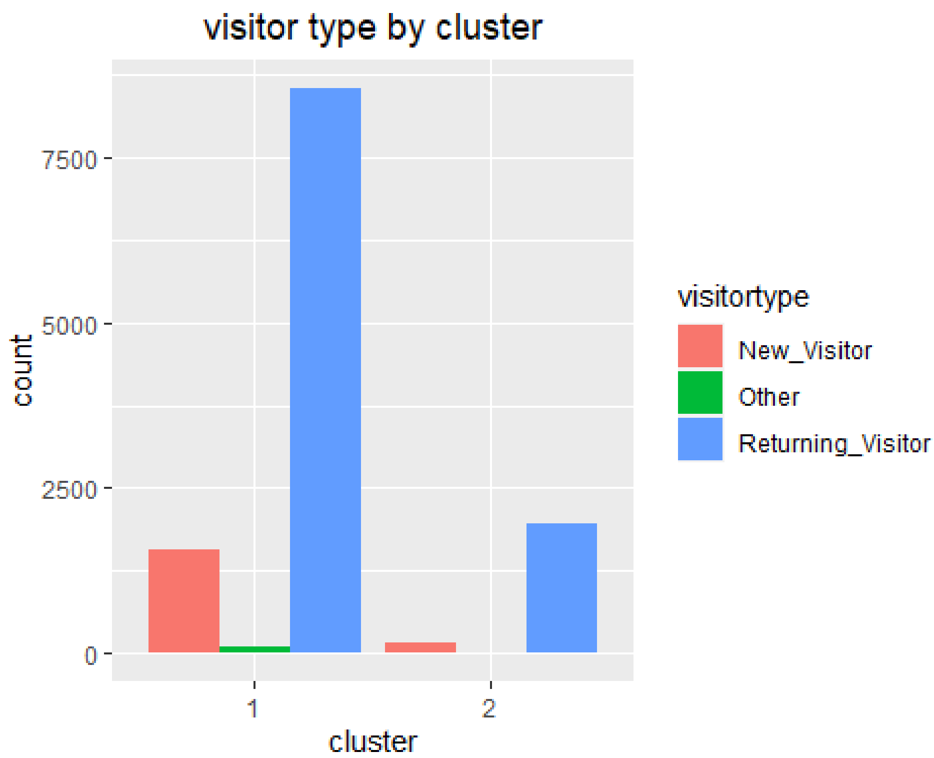
Average page values are higher in cluster 2

```
#plotting weekend by cluster
ggplot() + geom_bar(
    data=df_clus,
    aes(x=factor(cluster), fill = factor(weekend)
    ), position="dodge") + labs(title = "weekend by cluster",
          y="count", x="cluster", fill="weekend") + theme(plot.title =
element_text(hjust=0.5))
```


```
prop.table(table(df_clus$cluster, df_clus$weekend), 1)

##
##          FALSE      TRUE
##   1 0.7708784 0.2291216
##   2 0.7482185 0.2517815

#columns false true represent weekend
```

The proportion of customers visiting the site over the weekend in cluster 2 is higher than the proportion in cluster one who do so.

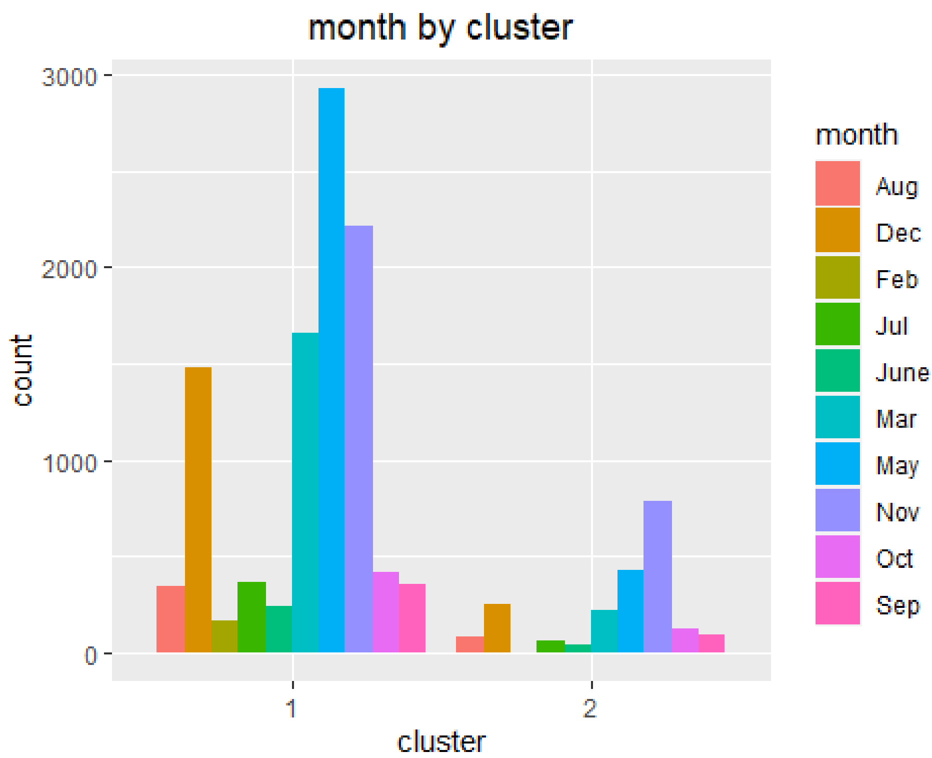
```
#plotting visitortype by cluster
ggplot() + geom_bar(
    data=df_clus,
    aes(x=factor(cluster), fill = factor(visitortype)
    ), position="dodge") + labs(title = "visitor type by cluster",
        y="count", x="cluster", fill="visitortype") + theme(plot.title =
element_text(hjust=0.5))
```




```
prop.table(table(df_clus$cluster, df_clus$visitortype), 1)

##
##     New_Visitor       Other Returning_Visitor
##   1 0.152584005 0.007663588       0.839752407
##   2 0.066983373 0.003325416       0.929691211
```

The proportion of returning visitors among in cluster 2 is higher, while the proportions of new visitor and other is higher in cluster 1.

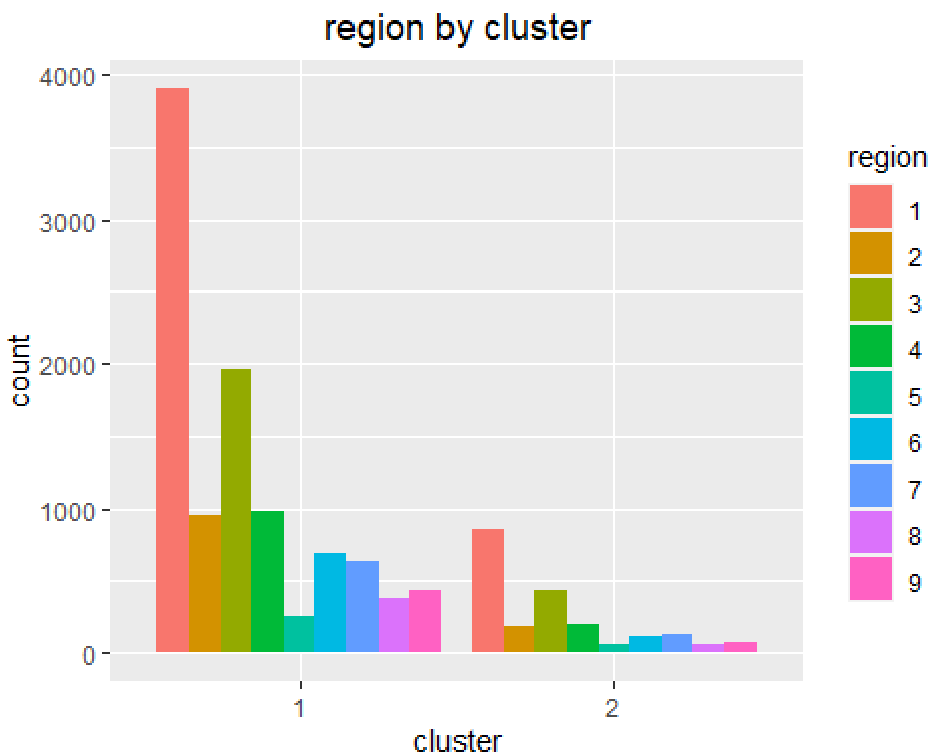
```
#plotting month by cluster
ggplot() + geom_bar(
    data=df_clus,
    aes(x=factor(cluster), fill = factor(month)
    ), position="dodge") + labs(title = "month by cluster",
          y="count", x="cluster", fill="month") + theme(plot.title =
element_text(hjust=0.5))
```




```
prop.table(table(df_clus$cluster, df_clus$month), 1)

##
##             Aug         Dec         Feb         Jul        June
Mar
##   1 0.034289644 0.144920417 0.016407939 0.036058165 0.023973276
0.163096876
##   2 0.039904988 0.119714964 0.001900238 0.030403800 0.020902613
0.106413302
##
##             May         Nov         Oct         Sep
##   1 0.287286304 0.217233248 0.041560228 0.035173904
##   2 0.205700713 0.372446556 0.059857482 0.042755344
```

Most cluster 2 customers visit the site in the month of November, while most in cluster 1 visit in May.

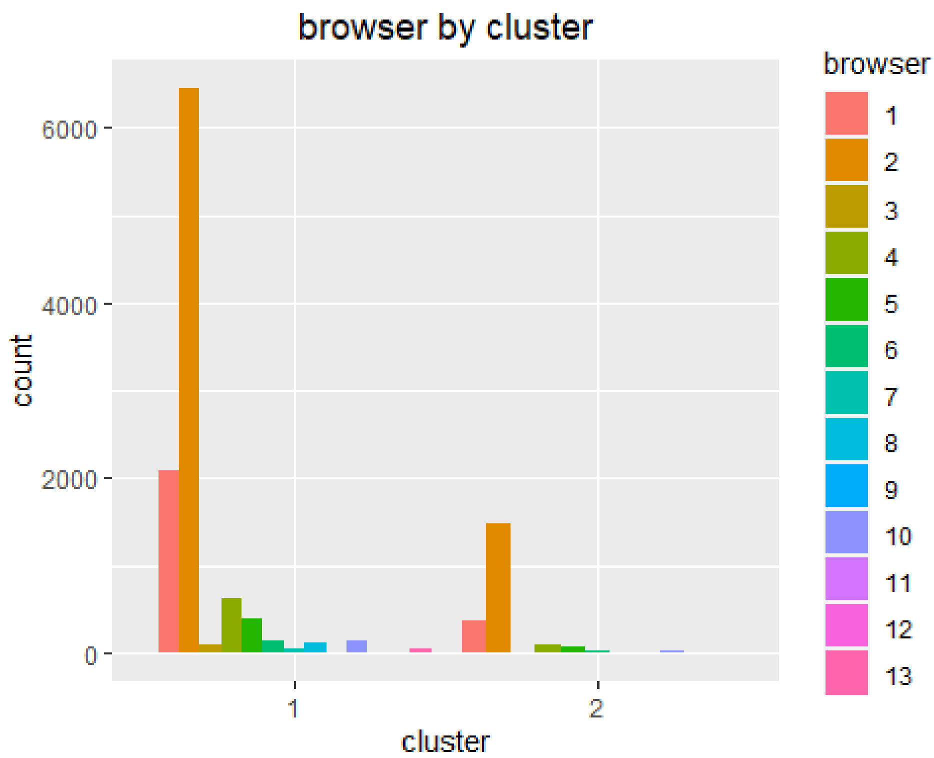
```
#plotting region by cluster
ggplot() + geom_bar(
    data=df_clus,
    aes(x=factor(cluster), fill = factor(region)
    ), position="dodge") + labs(title = "region by cluster",
           y="count", x="cluster", fill="region") + theme(plot.title =
element_text(hjust=0.5))
```




```
prop.table(table(df_clus$cluster, df_clus$region), 1)

##
##              1          2          3          4          5          6
##   1 0.38376891 0.09304382 0.19247396 0.09618786 0.02544704 0.06769503
##   2 0.40807601 0.08693587 0.20665083 0.09311164 0.02660333 0.05463183
##
##              7          8          9
##   1 0.06209471 0.03684417 0.04244449
##   2 0.05985748 0.02802850 0.03610451
```

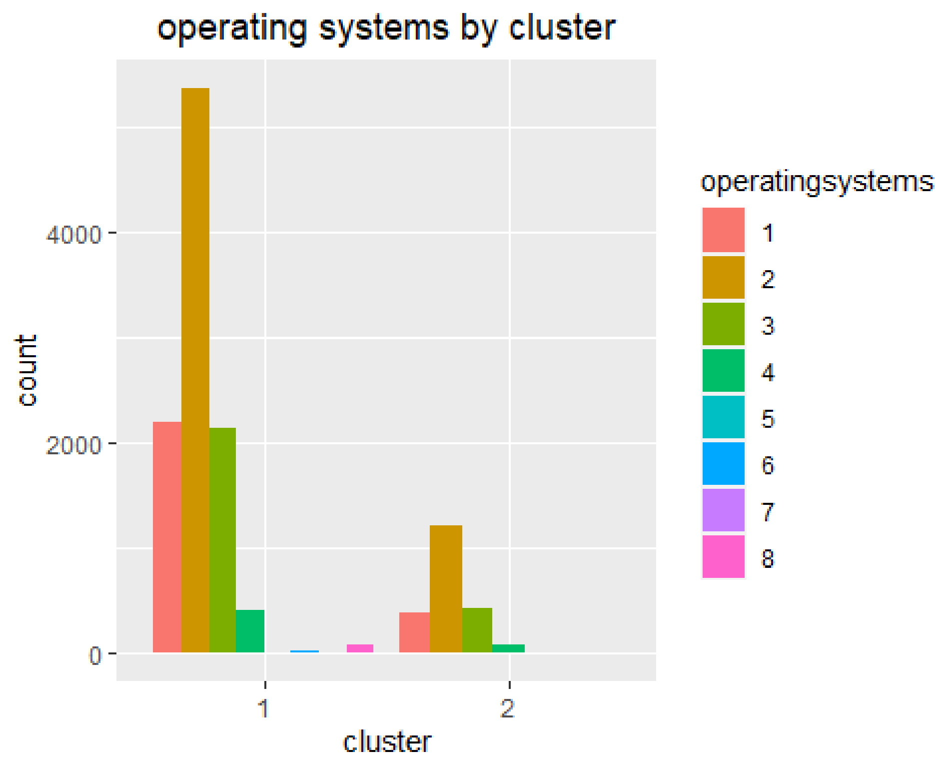
In both clusters, most customers are from region 1

```
#plotting browser by cluster
ggplot() + geom_bar(
    data=df_clus,
    aes(x=factor(cluster), fill = factor(browser)
    ), position="dodge") + labs(title = "browser by cluster",
          y="count", x="cluster", fill="browser") + theme(plot.title =
element_text(hjust=0.5))
```



In both clusters, most customers use browser 2

```
#plotting operatingsystems by cluster
ggplot() + geom_bar(
    data=df_clus,
    aes(x=factor(cluster), fill = factor(operatingsystems)
    ), position="dodge") + labs(title = "operating systems by cluster",
           y="count", x="cluster", fill="operatingsystems") +
theme(plot.title = element_text(hjust=0.5))
```




```
prop.table(table(df_clus$cluster, df_clus$operatingsystems), 1)

##
##                1            2            3            4            5
##   1 0.2148752211 0.5267243073 0.2093731578 0.0391039497 0.0005895068
##   2 0.1833729216 0.5781472684 0.1966745843 0.0370546318 0.0000000000
##
##                6            7            8
##   1 0.0015720181 0.0005895068 0.0071723325
##   2 0.0014251781 0.0004750594 0.0028503563
```

In both clusters, most customers use operating system 2

Comparisons between K Means and Hierarchical

K means clustering

- Advantages: Easy to implement, easily adapts to new examples.
- Disadvantages: The number of clusters has to be predetermined, it is sensitive to scaling, the initial seeds heavily influence the results.

Hierachical clustering:

- Advantages: The number of clusters do not have to be predetermined, ordering of levels in display is informative, easy to implement.

- Disadvantages: Not as suitable for large datasetes due to lower spacial and computational efficiency . This was evident in the duration of time the codes took to run as well as in the structure of the dendrograms.

Conclusion and Recommendations

Conclusion

The objectives of the study were achieved. Following data preparation (where missing values, duplicates, outliers, column creation etc were dealt with accordingly), univariate and bivariate analysis were carried out providing valuable insights on the dataset as a whole.

Some general bivariate analysis insights include: the proportion of visits that generated revenue during weekends was higher than revenue producing visits during the weekdays, the proportion of revenue producing visits was highest among new visitors , the month with the highest proportion of revenue generating visits was November etc.

Modelling:

Two approaches were used in clustering the data: K-means clustering and hierarchical clustering.

Initially k-means was used with an arbitrary value of 3. After comparing the average silhouette score at different levels of k, 2 was determined to be the optimal number of clusters.

For hierarchical clustering, complete linkage method was used initially, and average and wards methods also tested. The dendrogram using ward's method was the best structured. 2 clusters were highlighted on the dendrogram

Customer group characteristics comparisons

Further analysis was carried out on the 2 customer groups that were identified while using kmeans to compare the characteristics of the different groups.

Highlights:

- The proportion of customers of cluster 2 who generate revenue is higher than the proportion of customers in cluster 1 who generate revenue.
- The average number of administrative, informational and product related pages visited in a session, as well as the average durations spent on these different page types, is higher among customers in cluster 2 than in cluster one.
- Bouncerates and exit rates are higher among customers in cluster 1.
- Average page values are higher in cluster 2

- The proportion of customers visiting the site over the weekend in cluster 2 is higher than the proportion in cluster one who do so.
- The proportion of returning visitors among in cluster 2 is higher, while the proportions of new visitor and other is higher in cluster
- Most cluster 2 customers visit the site in the month of November, while most in cluster 1 visit in May.
- In both clusters, most customers are from region 1.
- In both clusters, most customers use operating system 2
- In both clusters, most customers use browser system 2

Recommendations

- Cluster 2 had a higher proprtion of revenue-generating customers compared to cluster 1.
- Cluster 1 had higher bounce rates and exit rates, indicating that more customers in this category are likely to leave without making a transaction. Optional targeted surveys could pop up to customers falling in this category to discover possible causes of dissatisfaction with the site or service. Similarly, since more customers in cluster 2 spent a longer duration on the site and visited more pages, targeted surveys to customers in this categories on what they are satisfied with will help the company know what to keep doing.
- The proportion of returning visitors among cluster 2 is higher. The company should prioritise quality products, services, and presentation from the get go, enabling them to have more returning visitors on the site.
- Although there is more traffic during the week, the proportion of revenue generating visits is higher over the weekend. More ads should be run during the weekends compared to weekdays.

- Future recommendations - Further information such as the gender and age of visitors, specific product categories visited etc should be obtained as they will aid in better understanding customer behaviour and in grouping further.