CSDA 1010 - Lab2 - Group 1

Importing libraries

```
library(plyr); library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

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

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

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

```
library(Hmisc)
```

```
## Loading required package: lattice
```

```
## Loading required package: survival
```

```
## Loading required package: Formula
```

```
## Loading required package: ggplot2
```

```
##
## Attaching package: 'Hmisc'
```

```
## The following objects are masked from 'package:dplyr':
##
##     src, summarize
```

```
## The following objects are masked from 'package:plyr':
##
##     is.discrete, summarize
```

```
## The following objects are masked from 'package:base':
##
##     format.pval, units
```

```
library(corrplot)
```

```
## corrplot 0.84 loaded
```

```
library(caret)
```

```
##
## Attaching package: 'caret'
```

```
## The following object is masked from 'package:survival':
##
##     cluster
```

```
library(pROC)
```

```
## Type 'citation("pROC")' for a citation.
```

```
##
## Attaching package: 'pROC'
```

```
## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```
library(ROCR)
library(Metrics)
```

```
##
## Attaching package: 'Metrics'
```

```
## The following object is masked from 'package:pROC':
##
##     auc
```

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

```
library(caTools)
library(rpart.plot)
```

```
## Loading required package: rpart
```

```
library(boot)
```

```
##
## Attaching package: 'boot'
```

```
## The following object is masked from 'package:survival':
##
##     aml
```

```
## The following object is masked from 'package:lattice':
##
##     melanoma
```

Importing dataset

```
data <- read.csv('online_shoppers_intention.csv', header = TRUE)
```

Data understanding (Section 2 of the report)

Dataset structure

```
str(data)
```

```
## 'data.frame':    12330 obs. of  18 variables:
##  $ Administrative         : int  0 0 0 0 0 0 0 1 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 1 2 10 19 1 0 2 3 ...
##  $ ProductRelated_Duration: num  0 64 0 2.67 627.5 ...
##  $ BounceRates            : num  0.2 0 0.2 0.05 0.02 ...
##  $ ExitRates              : num  0.2 0.1 0.2 0.14 0.05 ...
##  $ PageValues             : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ SpecialDay             : num  0 0 0 0 0 0 0.4 0 0.8 0.4 ...
##  $ Month                  : Factor w/ 10 levels "Aug","Dec","Feb",..: 3 3 3 3 3 3 3 3 3 3 ...
##  $ OperatingSystems       : int  1 2 4 3 3 2 2 1 2 2 ...
##  $ Browser                : int  1 2 1 2 3 2 4 2 2 4 ...
##  $ Region                 : int  1 1 9 2 1 1 3 1 2 1 ...
##  $ TrafficType            : int  1 2 3 4 4 3 3 5 3 2 ...
##  $ VisitorType            : Factor w/ 3 levels "New_Visitor",..: 3 3 3 3 3 3 3 3 3 3 ...
##  $ Weekend                : logi  FALSE FALSE FALSE FALSE TRUE FALSE ...
##  $ Revenue                : logi  FALSE FALSE FALSE FALSE FALSE FALSE ...
```

Dataset summary

examining basic descriptive statistics with “summary” function

```
summary(data)
```

```
##  Administrative   Administrative_Duration Informational
##  Min.   : 0.000   Min.   :   0.00         Min.   : 0.0000
##  1st Qu.: 0.000   1st Qu.:   0.00         1st Qu.: 0.0000
##  Median : 1.000   Median :   7.50         Median : 0.0000
##  Mean   : 2.315   Mean   :  80.82         Mean   : 0.5036
##  3rd Qu.: 4.000   3rd Qu.:  93.26         3rd Qu.: 0.0000
##  Max.   :27.000   Max.   :3398.75         Max.   :24.0000
##
##  Informational_Duration ProductRelated   ProductRelated_Duration
##  Min.   :   0.00        Min.   :  0.00   Min.   :    0.0
##  1st Qu.:   0.00        1st Qu.:  7.00   1st Qu.:  184.1
##  Median :   0.00        Median : 18.00   Median :  598.9
##  Mean   :  34.47        Mean   : 31.73   Mean   : 1194.8
##  3rd Qu.:   0.00        3rd Qu.: 38.00   3rd Qu.: 1464.2
##  Max.   :2549.38        Max.   :705.00   Max.   :63973.5
##
##   BounceRates         ExitRates         PageValues        SpecialDay
##  Min.   :0.000000   Min.   :0.00000   Min.   :  0.000   Min.   :0.00000
##  1st Qu.:0.000000   1st Qu.:0.01429   1st Qu.:  0.000   1st Qu.:0.00000
##  Median :0.003112   Median :0.02516   Median :  0.000   Median :0.00000
##  Mean   :0.022191   Mean   :0.04307   Mean   :  5.889   Mean   :0.06143
##  3rd Qu.:0.016813   3rd Qu.:0.05000   3rd Qu.:  0.000   3rd Qu.:0.00000
##  Max.   :0.200000   Max.   :0.20000   Max.   :361.764   Max.   :1.00000
##
##      Month      OperatingSystems    Browser           Region
##  May    :3364   Min.   :1.000    Min.   : 1.000   Min.   :1.000
##  Nov    :2998   1st Qu.:2.000    1st Qu.: 2.000   1st Qu.:1.000
##  Mar    :1907   Median :2.000    Median : 2.000   Median :3.000
##  Dec    :1727   Mean   :2.124    Mean   : 2.357   Mean   :3.147
##  Oct    : 549   3rd Qu.:3.000    3rd Qu.: 2.000   3rd Qu.:4.000
##  Sep    : 448   Max.   :8.000    Max.   :13.000   Max.   :9.000
##  (Other):1337
##   TrafficType               VisitorType     Weekend         Revenue
##  Min.   : 1.00   New_Visitor      : 1694   Mode :logical   Mode :logical
##  1st Qu.: 2.00   Other            :   85   FALSE:9462      FALSE:10422
##  Median : 2.00   Returning_Visitor:10551   TRUE :2868      TRUE :1908
##  Mean   : 4.07
##  3rd Qu.: 4.00
##  Max.   :20.00
##
```

Checking for missing values

```
colSums(is.na(data))
```

```
##          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
```

```
colSums(data=="")
```

```
##          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
```

dataset does not contain missing values

Feature visualizations (Only select visualizations were used/discussed in the report)

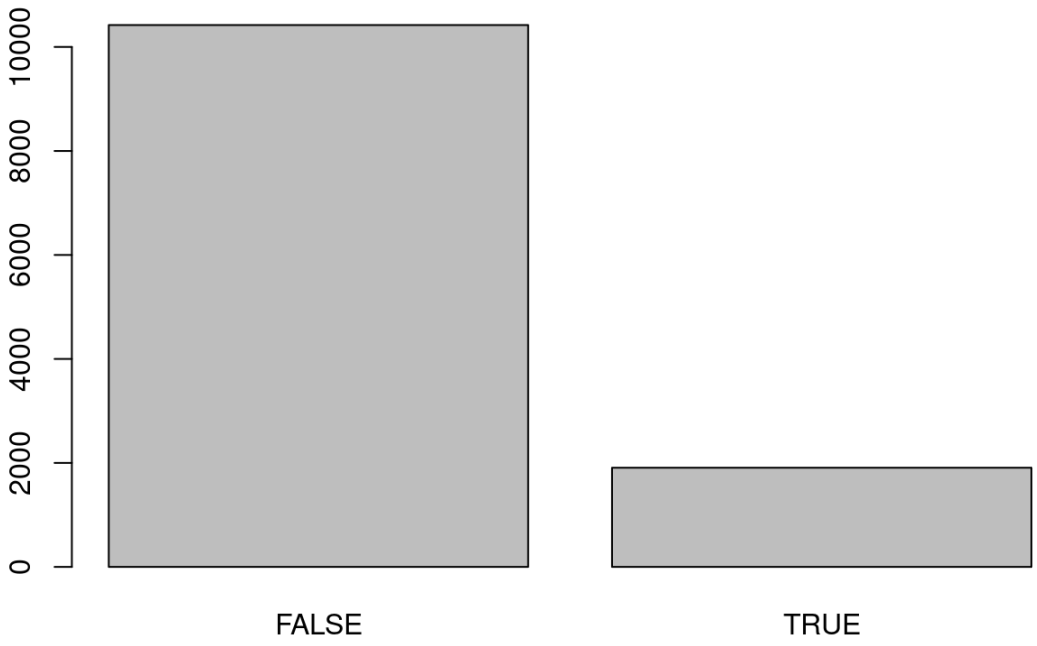
Revenue

extracting and plotting the frequency of target feature “Revenue”

```
freq_Revenue=table(data$Revenue)
head(freq_Revenue)
```

```
##
## FALSE  TRUE
## 10422  1908
```

```
barplot(freq_Revenue)
```

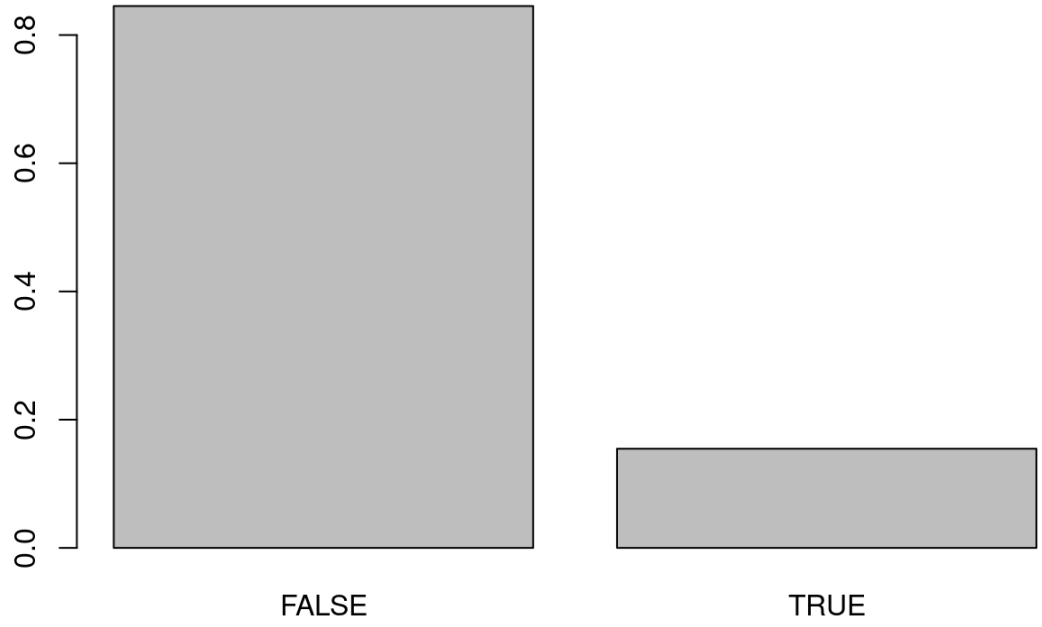


frequencies are converted to proportions, for better understanding and visualization

```
prop.table(freq_Revenue)
```

```
##
##     FALSE      TRUE
## 0.8452555 0.1547445
```

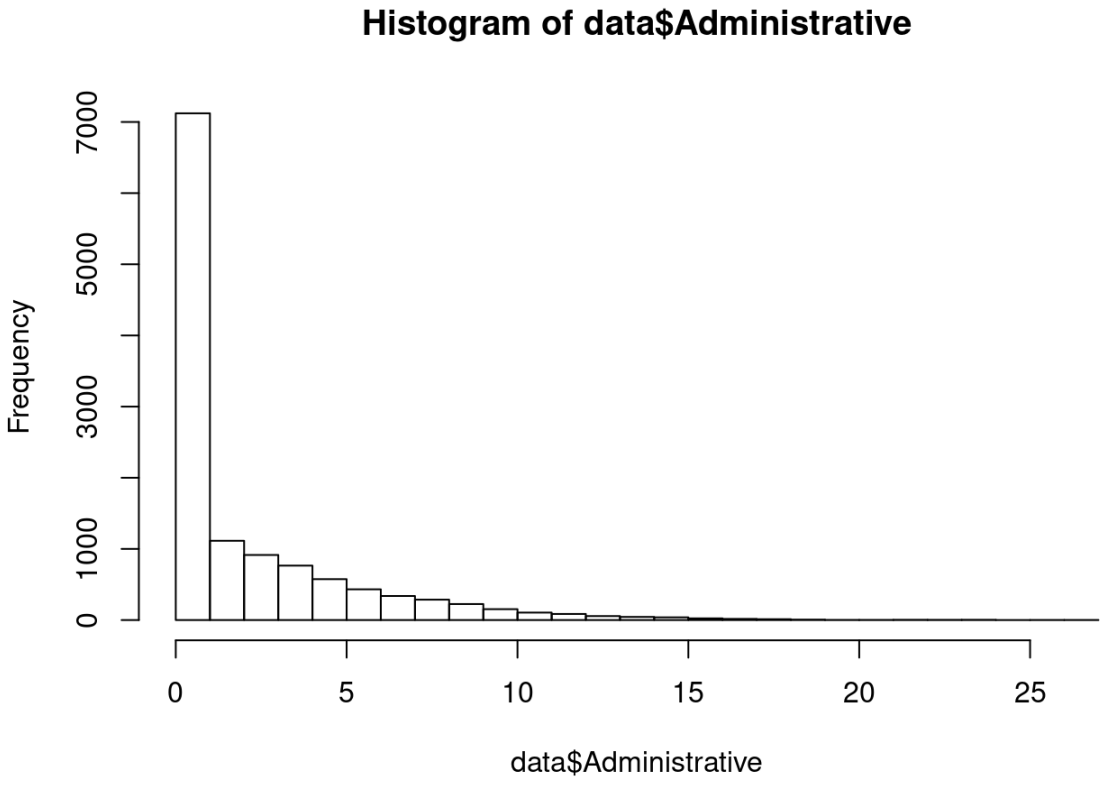
```
barplot(prop.table(freq_Revenue))
```



Administrative

plotting the histogram of "Administrative" feature; analogous plots will be used for other features in continuation, where appropriate

```
hist(data$Administrative, breaks=seq(0,27,1), labels=FALSE)
```

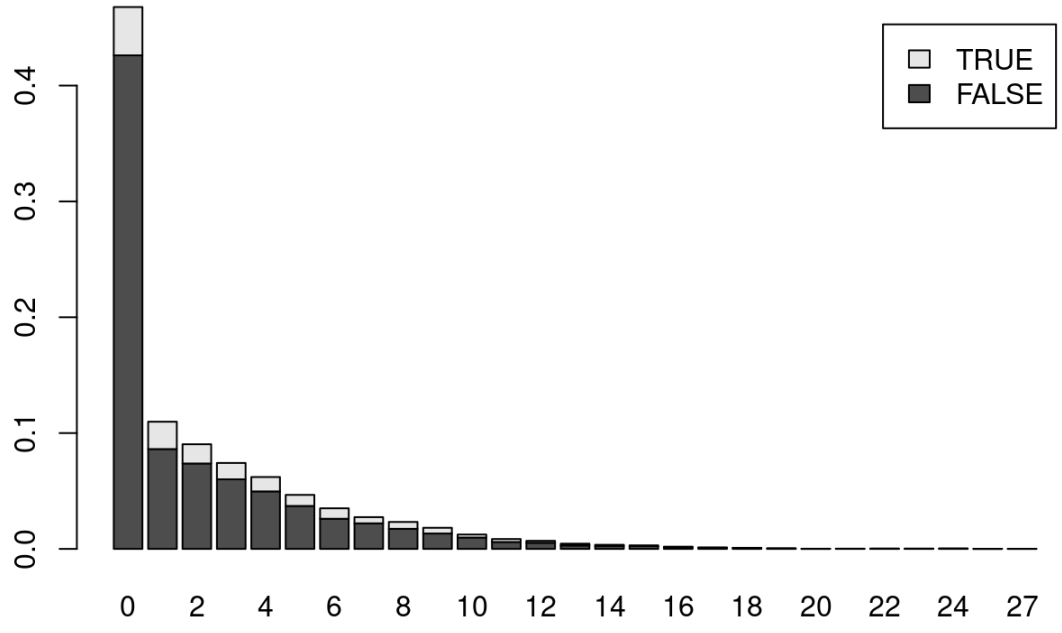


visualizing proportions of "Adminstrative" feature against the target "Revenue" feature; analogous plots will be used for other features in continuation

```
freq_Administrative=xtabs(~data$Revenue+data$Administrative)
prop.table(freq_Administrative)
```

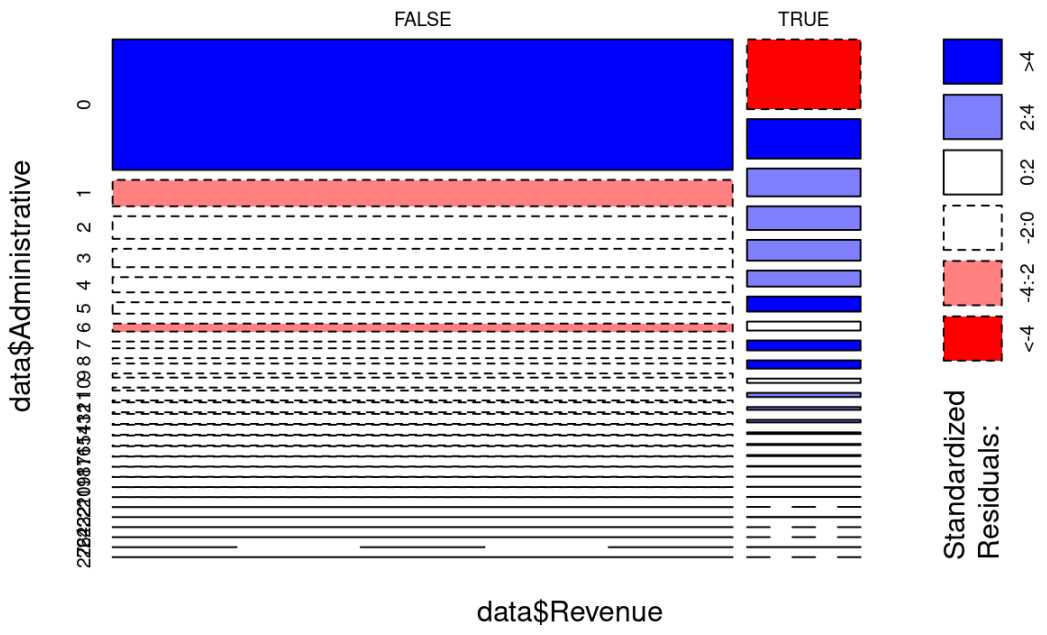
```
##             data$Administrative
## data$Revenue           0           1           2           3           4
##        FALSE 0.426115166 0.086212490 0.073722628 0.060097324 0.049635036
##        TRUE  0.041686942 0.023600973 0.016626115 0.014111922 0.012408759
##             data$Administrative
## data$Revenue           5           6           7           8           9
##        FALSE 0.037064071 0.026034063 0.022060016 0.017356042 0.013300892
##        TRUE  0.009570154 0.009002433 0.005352798 0.005920519 0.004947283
##             data$Administrative
## data$Revenue          10          11          12          13          14
##        FALSE 0.009813463 0.006001622 0.005271695 0.003000811 0.002595296
##        TRUE  0.002595296 0.002514193 0.001703163 0.001540957 0.000973236
##             data$Administrative
## data$Revenue          15          16          17          18          19
##        FALSE 0.002433090 0.001297648 0.000973236 0.000811030 0.000405515
##        TRUE  0.000648824 0.000648824 0.000324412 0.000162206 0.000081103
##             data$Administrative
## data$Revenue          20          21          22          23          24
##        FALSE 0.000081103 0.000162206 0.000162206 0.000243309 0.000324412
##        TRUE  0.000081103 0.000000000 0.000162206 0.000000000 0.000000000
##             data$Administrative
## data$Revenue          26          27
##        FALSE 0.000000000 0.000081103
##        TRUE  0.000081103 0.000000000
```

```
barplot(prop.table(freq_Administrative),legend=rownames(freq_Administrative))
```



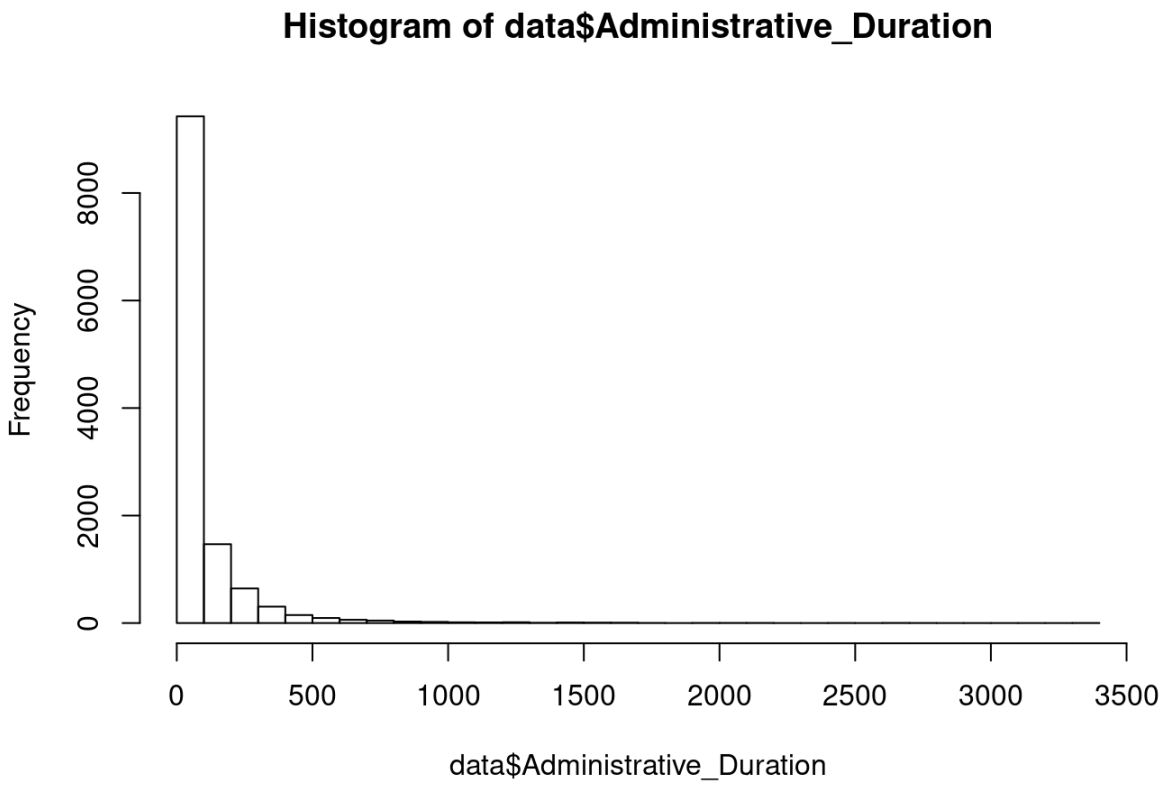
plotting the mosaic plot of frequencies of “Administrative” vs “Revenue” feature, with shaded residual levels for cell/level combinations of the two features; analogous plots will be used for other features in continuation

```
mosaicplot(freq_Administrative, border = "black",
           shade = TRUE)
```



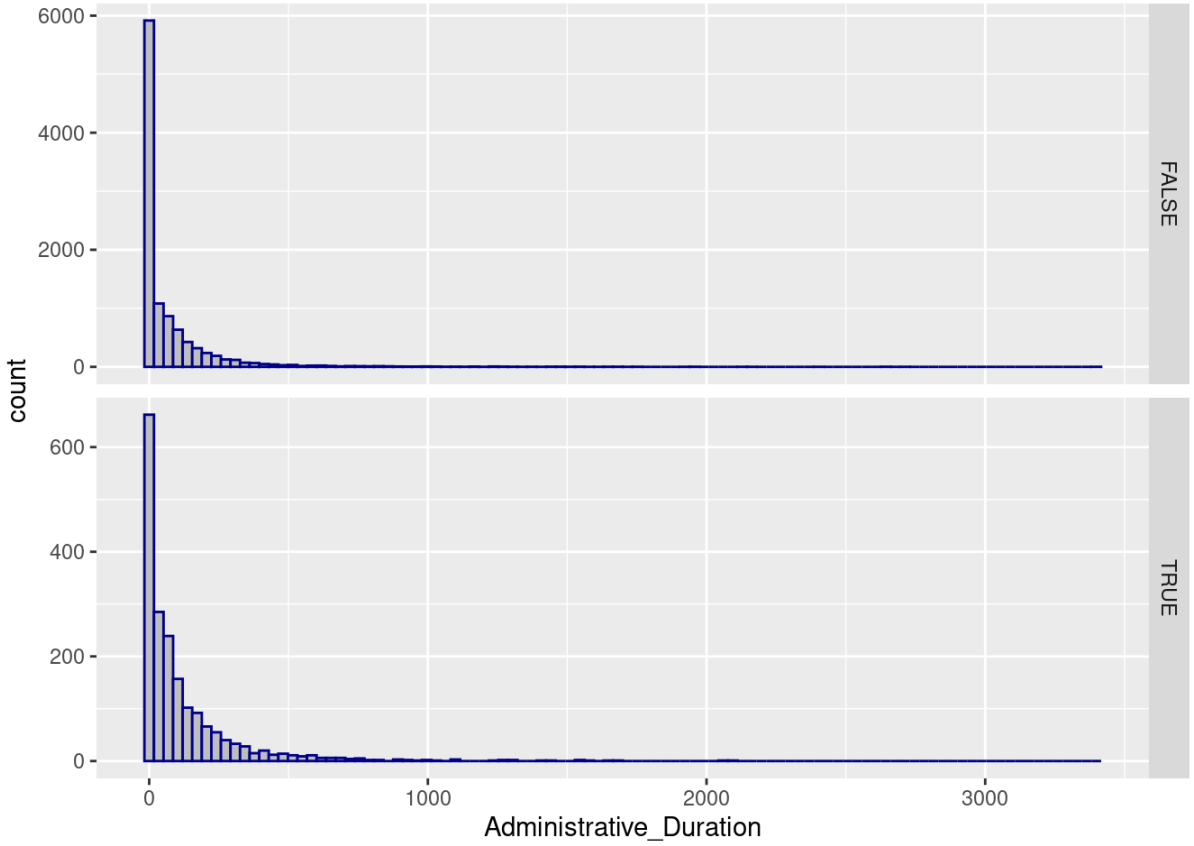
Administrative_Duration

```
hist(data$Administrative_Duration, breaks=seq(0,3400,100))
```



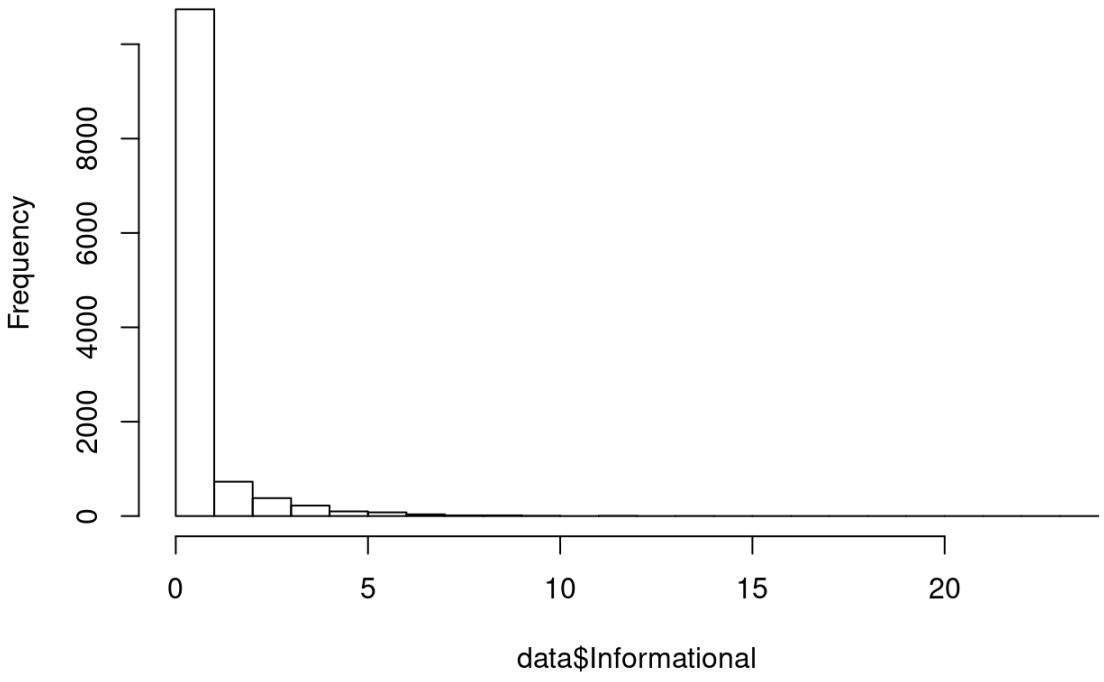
breaking down the histogram of "Administrative_Duration" based on the "Revenue" feature; analogous plots will be used in continuation, as appropriate

```
data %>%
  ggplot() +
  aes(x = Administrative_Duration) +
  geom_histogram(bins = 100, color='darkblue', fill='gray') +
  facet_grid(Revenue ~ .,
             scales = "free_y")
```



Informational

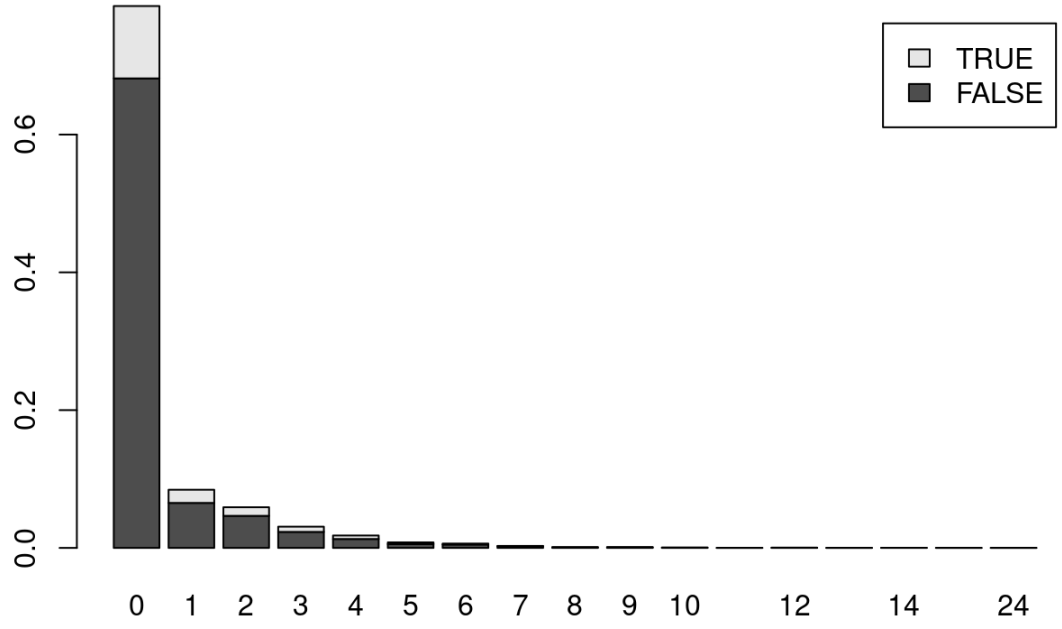
```
hist(data$Informational, breaks=seq(0,24,1), labels=FALSE)
```



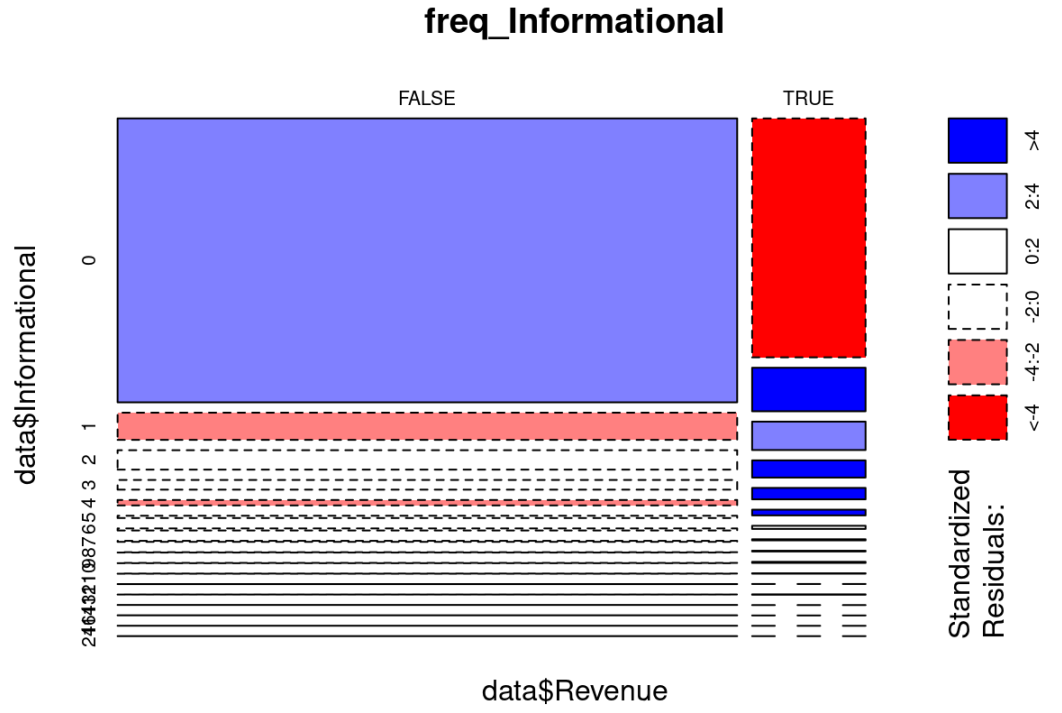

```
freq_Informational=xtabs(~data$Revenue+data$Informational)
prop.table(freq_Informational)
```

```
##             data$Informational
## data$Revenue           0           1           2           3           4
##        FALSE 0.681589619 0.065287916 0.046553122 0.023276561 0.012976480
##        TRUE  0.105028386 0.019140308 0.012489862 0.007542579 0.005028386
##             data$Informational
## data$Revenue           5           6           7           8           9
##        FALSE 0.005515004 0.004947283 0.002433090 0.000811030 0.000729927
##        TRUE  0.002514193 0.001378751 0.000486618 0.000324412 0.000486618
##             data$Informational
## data$Revenue          10          11          12          13          14
##        FALSE 0.000405515 0.000081103 0.000243309 0.000081103 0.000162206
##        TRUE  0.000162206 0.000000000 0.000162206 0.000000000 0.000000000
##             data$Informational
## data$Revenue          16          24
##        FALSE 0.000081103 0.000081103
##        TRUE  0.000000000 0.000000000
```

```
barplot(prop.table(freq_Informational),legend=rownames(freq_Informational))
```

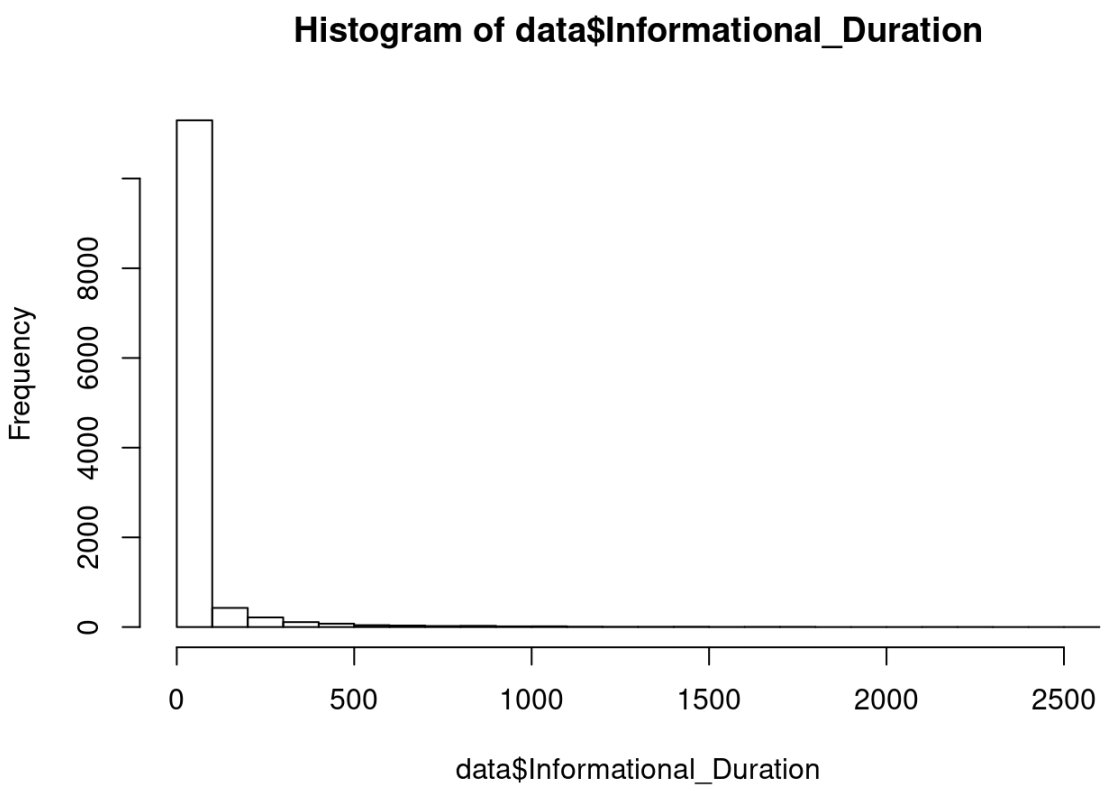



```
mosaicplot(freq_Informational, border = "black",
           shade = TRUE)
```

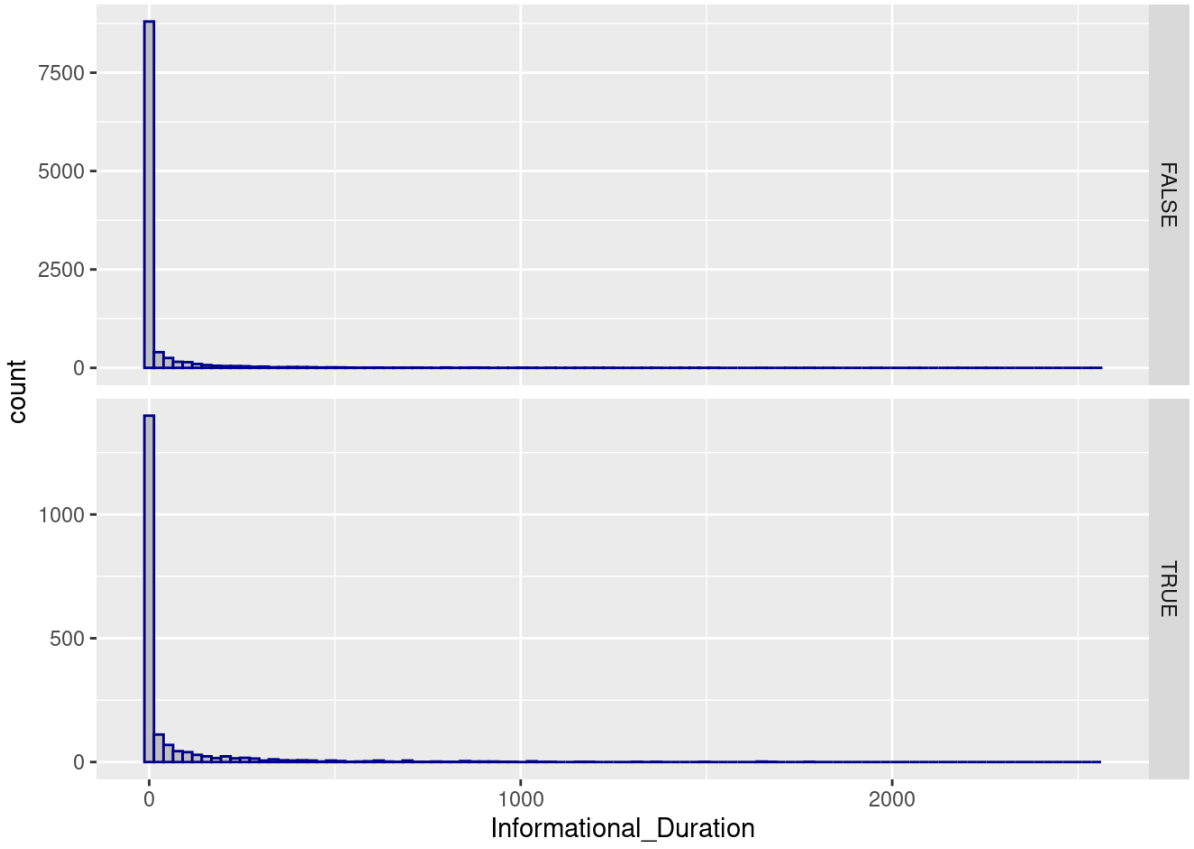


Informational_Duration

```
hist(data$Informational_Duration, breaks=seq(0,2600,100))
```

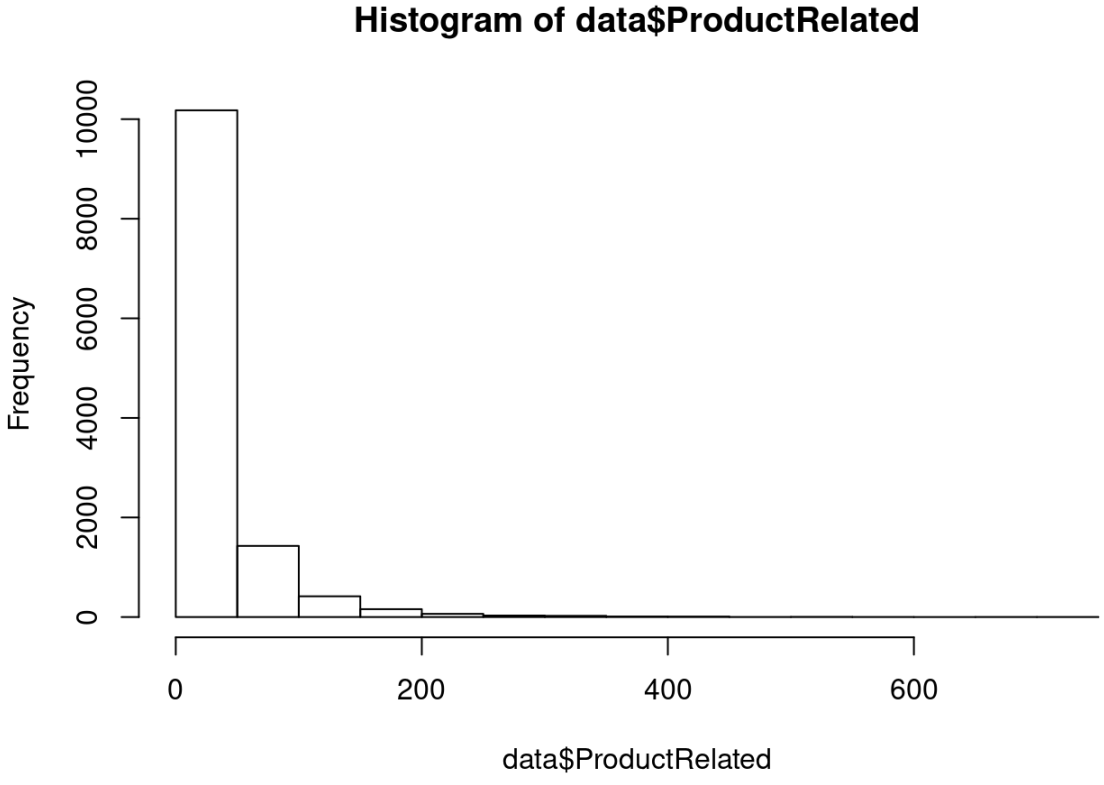



```
data %>%
  ggplot() +
  aes(x = Informational_Duration) +
  geom_histogram(bins = 100, color='darkblue', fill='gray') +
  facet_grid(Revenue ~ .,
             scales = "free_y")
```

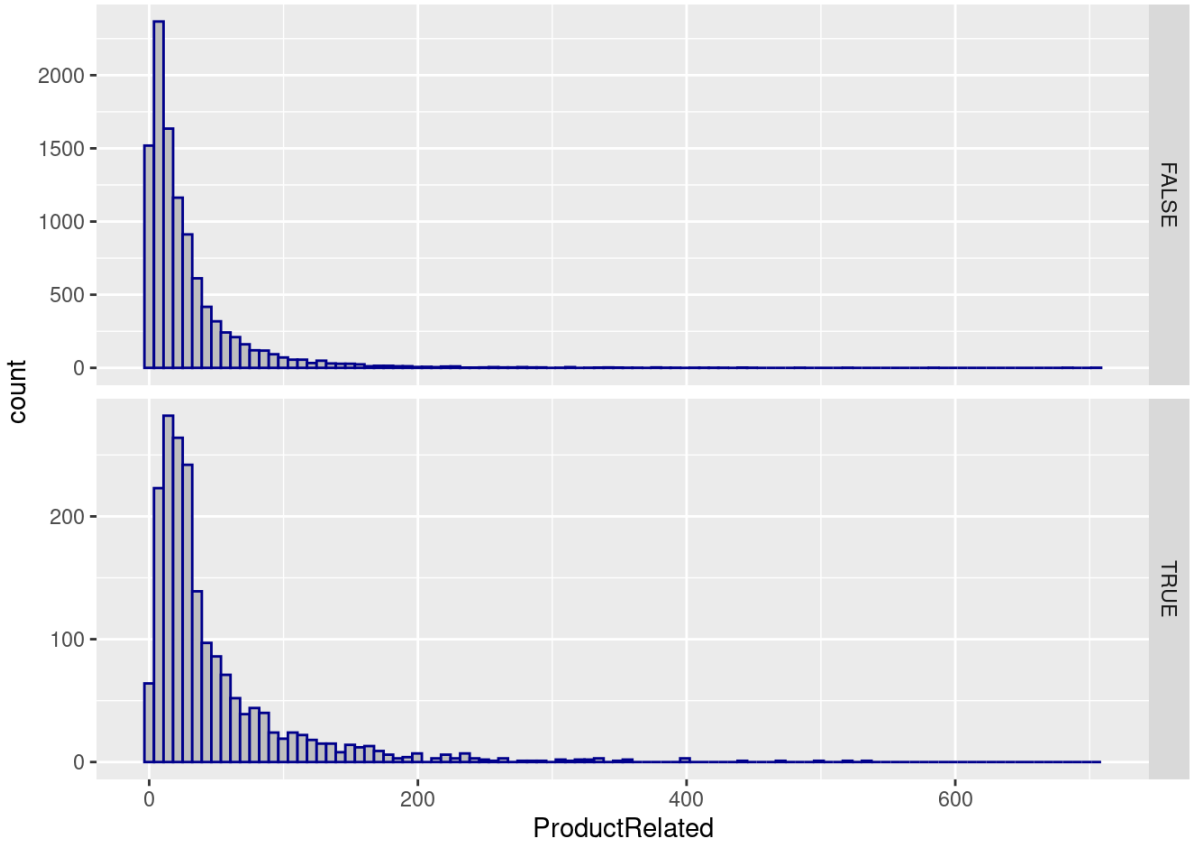


ProductRelated

```
hist(data$ProductRelated,breaks=seq(0,750,50))
```

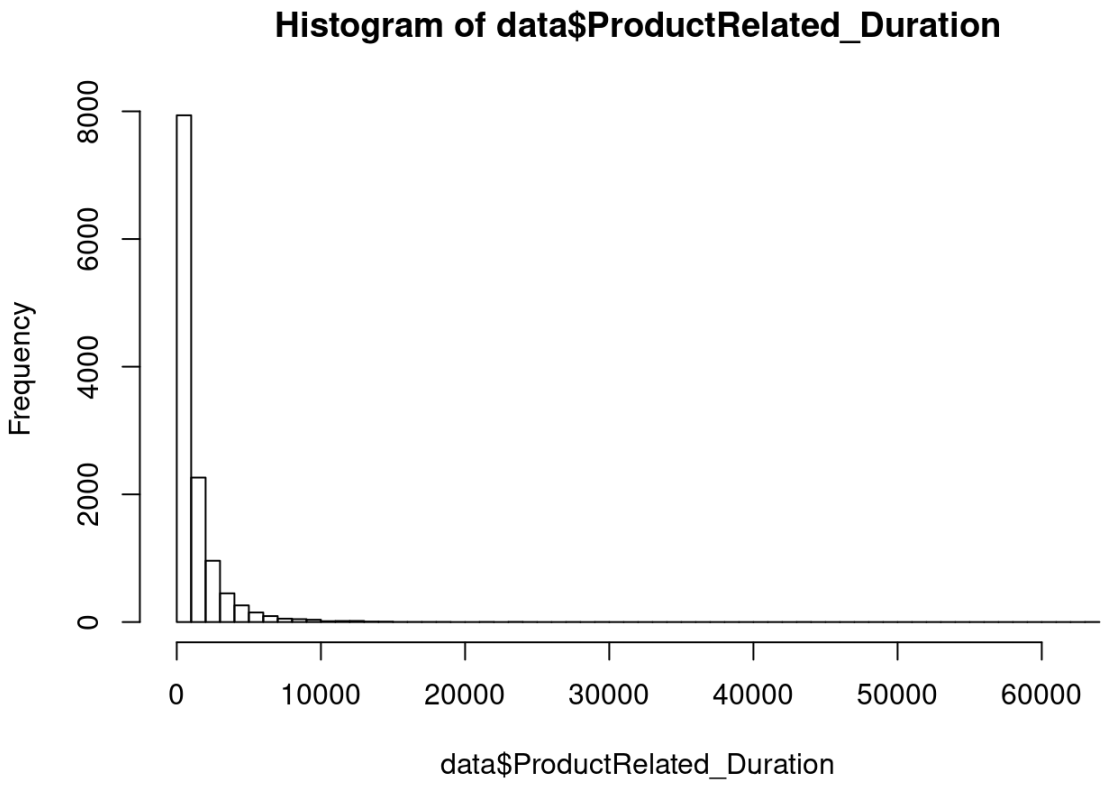


```
data %>%
  ggplot() +
  aes(x = ProductRelated) +
  geom_histogram(bins = 100, color='darkblue', fill='gray') +
  facet_grid(Revenue ~ .,
             scales = "free_y")
```



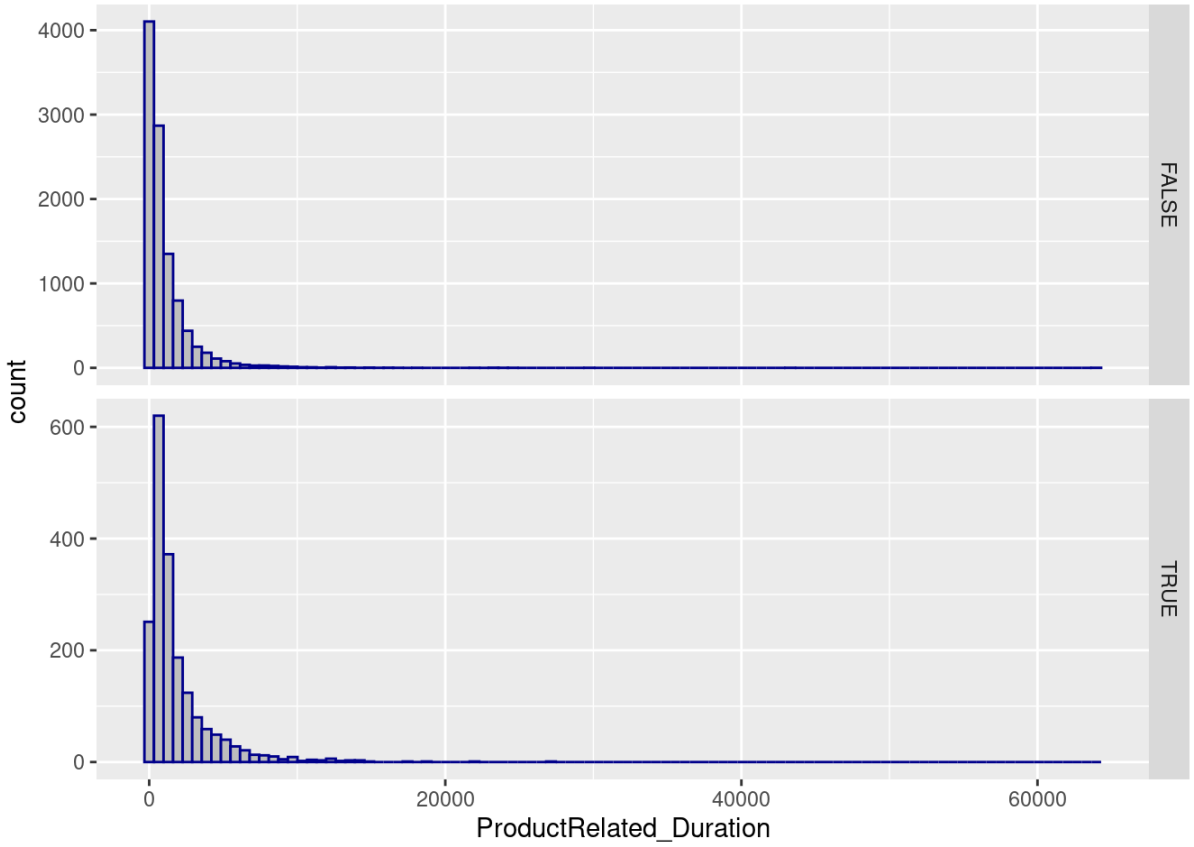
ProductRelated_Duration

```
hist(data$ProductRelated_Duration, breaks=seq(0,64000,1000))
```



breaking down the histogram of "ProductRelated_Duration" based on the "Revenue" feature to check how whether there is a difference of time spent on product pages in sessions that ended up with transaction

```
data %>%
  ggplot() +
  aes(x = ProductRelated_Duration) +
  geom_histogram(bins = 100, color='darkblue', fill='gray') +
  facet_grid(Revenue ~ .,
             scales = "free_y")
```



calculating the range of each bin on the x-axis"

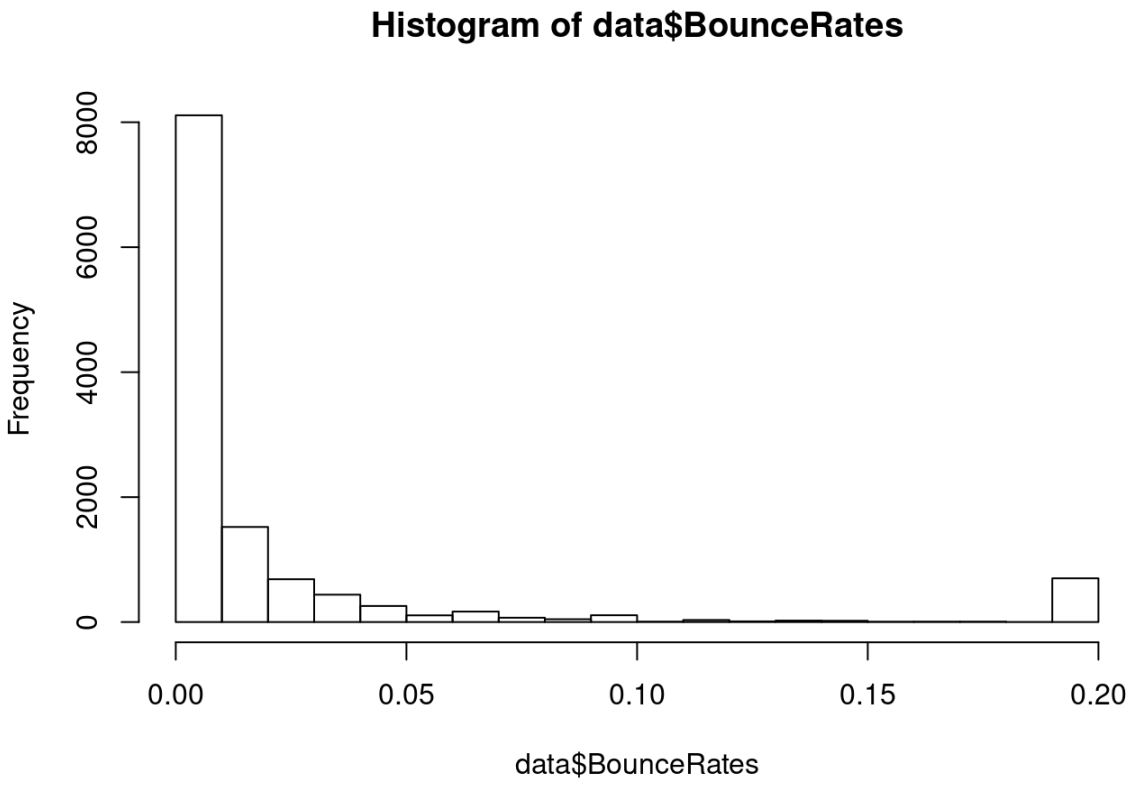
```
(data %>%
  summarise(max_ProductRelated = max(data$ProductRelated_Duration, na.rm = TRUE)))/100
```

max_ProductRelated <dbl>
639.7352

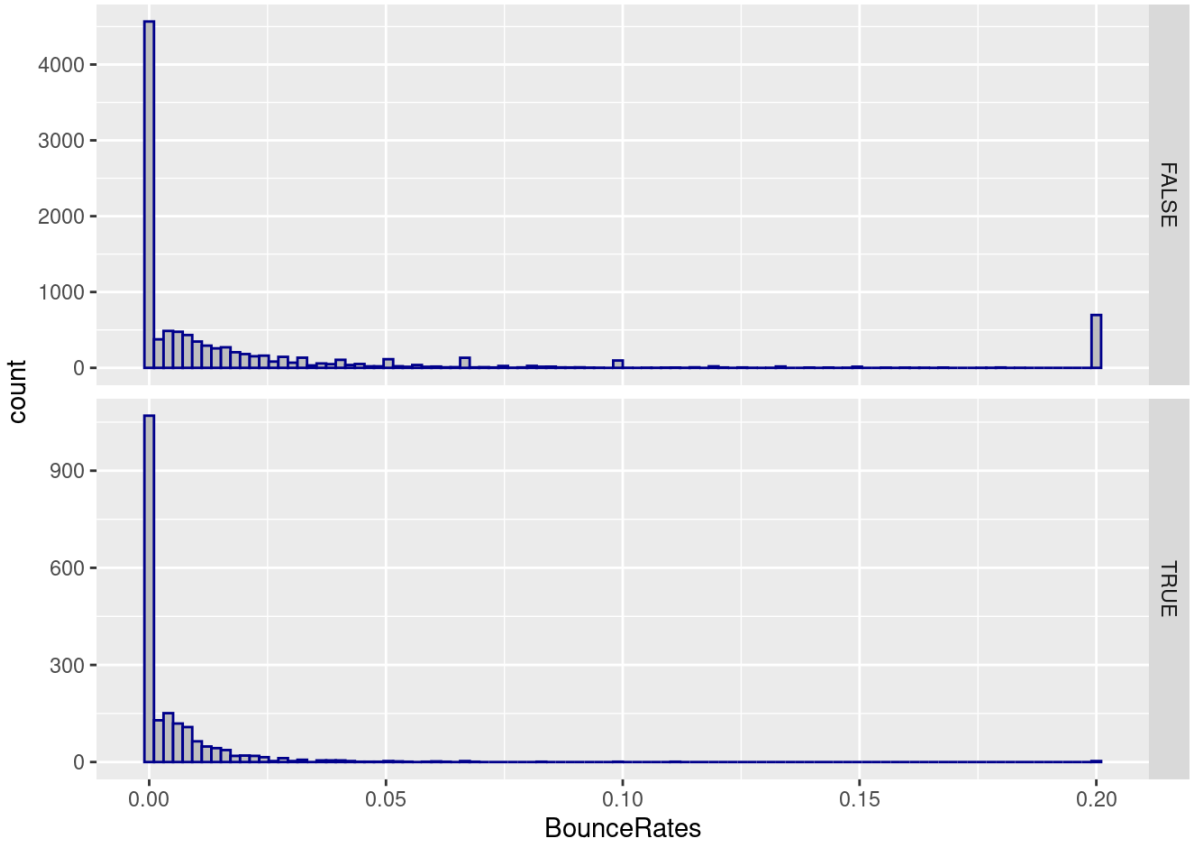
1 row

Bounce rates

```
hist(data$BounceRates)
```

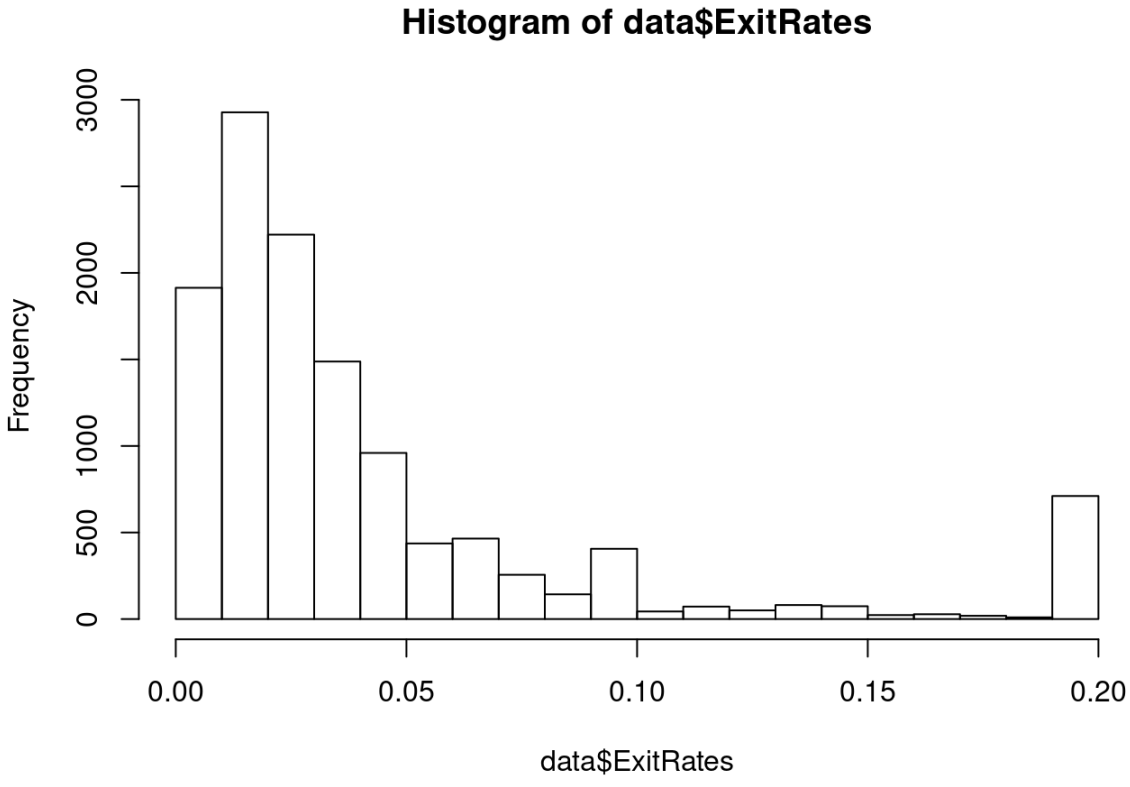



```
data %>%
  ggplot() +
  aes(x = BounceRates) +
  geom_histogram(bins = 100, color='darkblue', fill='gray') +
  facet_grid(Revenue ~ .,
             scales = "free_y")
```

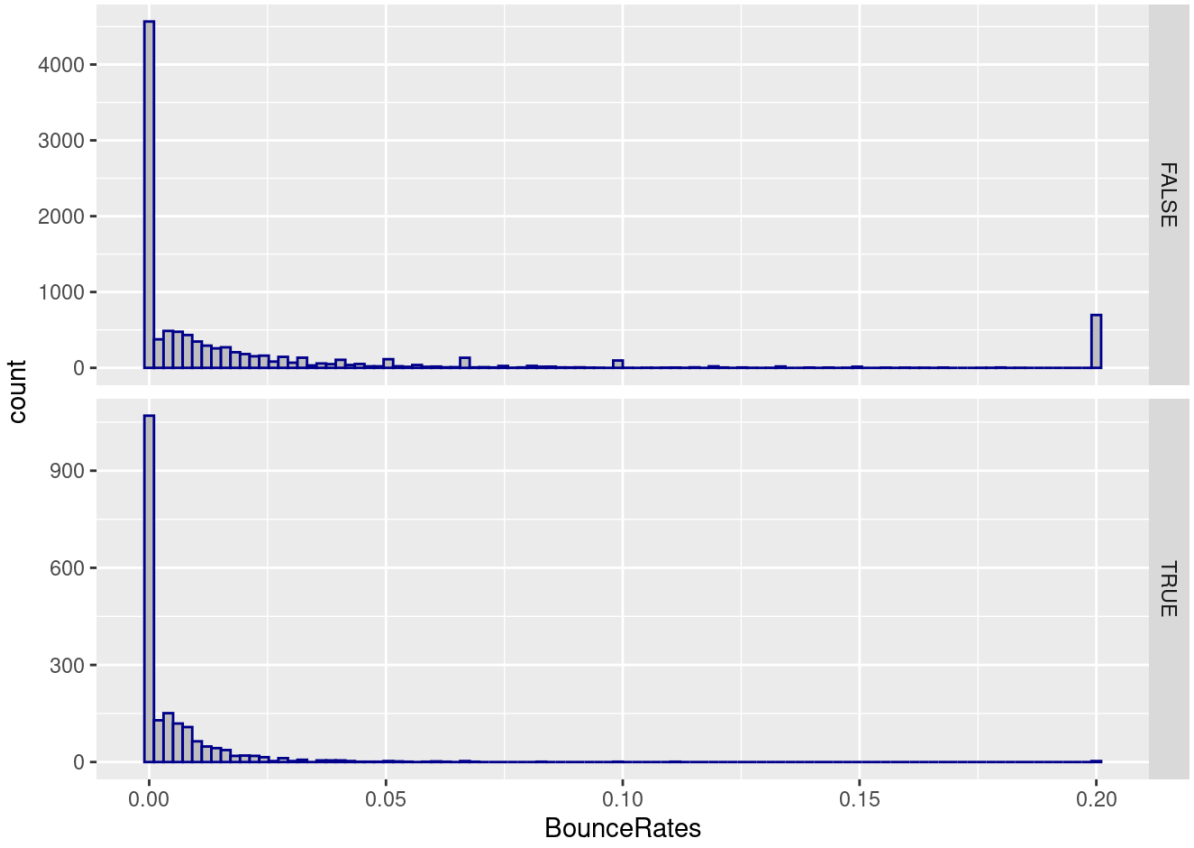


Exit rates

```
hist(data$ExitRates)
```

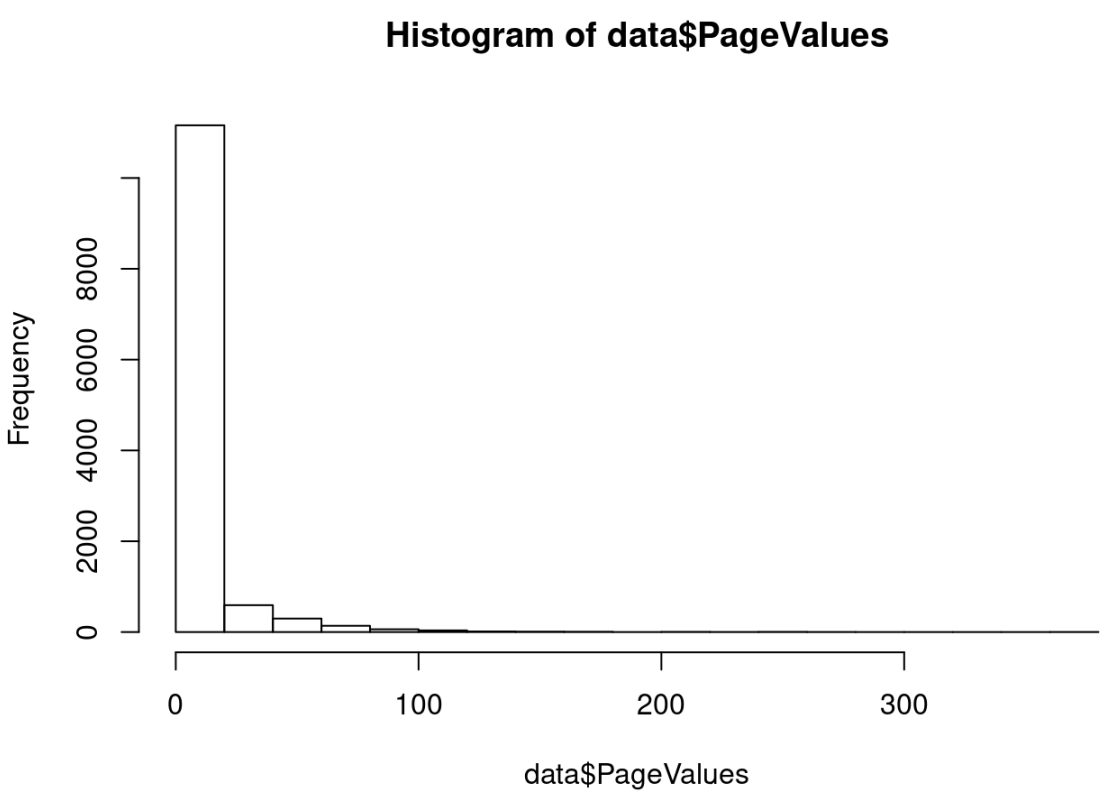


```
data %>%
  ggplot() +
  aes(x = BounceRates) +
  geom_histogram(bins = 100, color='darkblue', fill='gray') +
  facet_grid(Revenue ~ .,
             scales = "free_y")
```

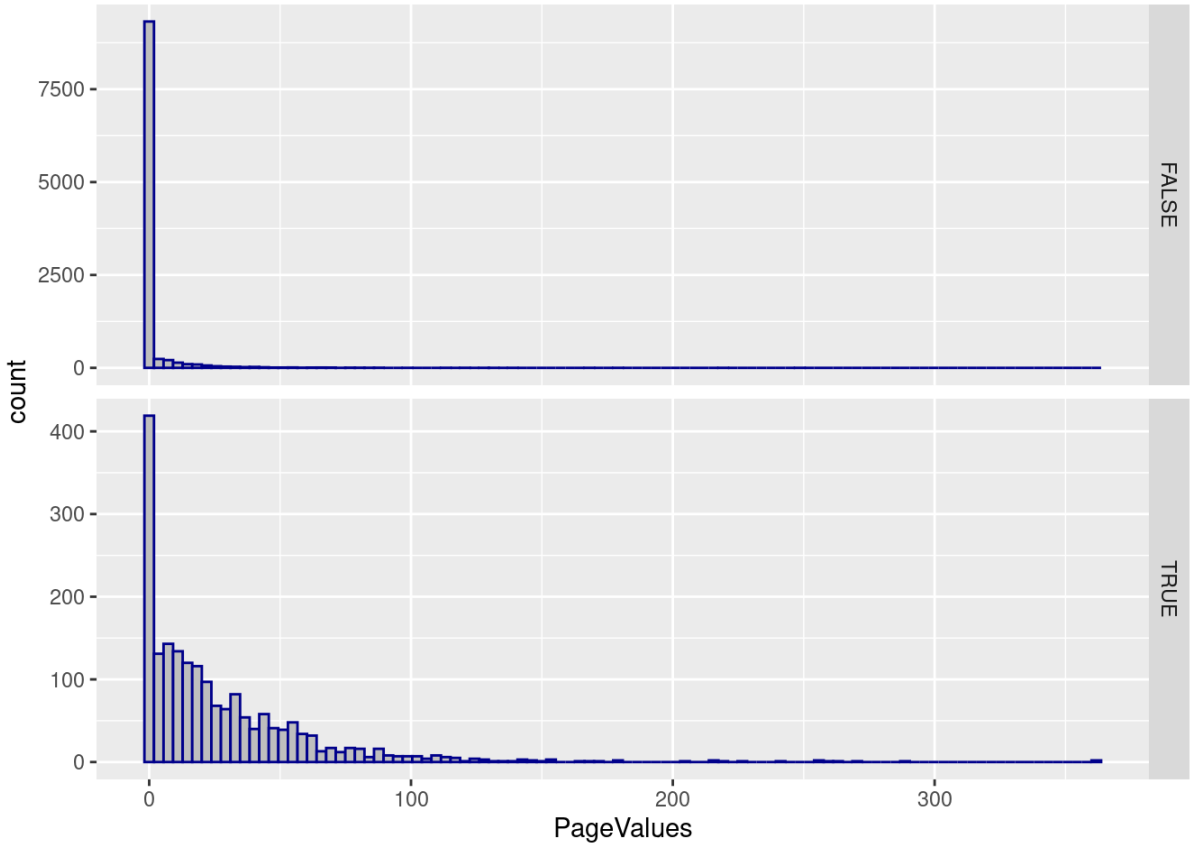


Page values

```
hist(data$PageValues)
```




```
data %>%
  ggplot() +
  aes(x = PageValues) +
  geom_histogram(bins = 100, color='darkblue', fill='gray') +
  facet_grid(Revenue ~ .,
             scales = "free_y")
```

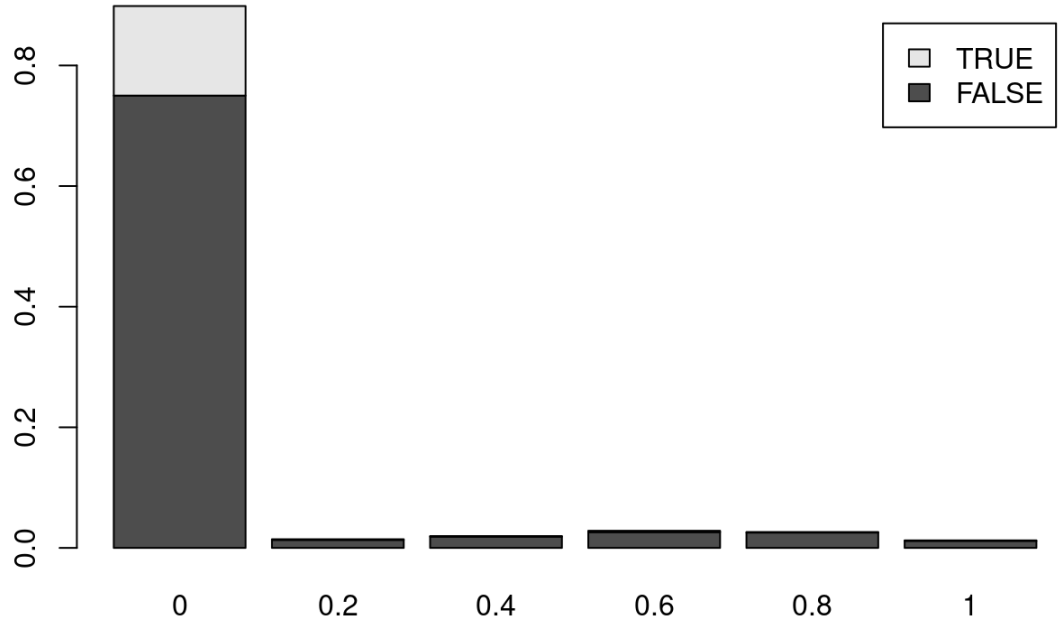


Special day

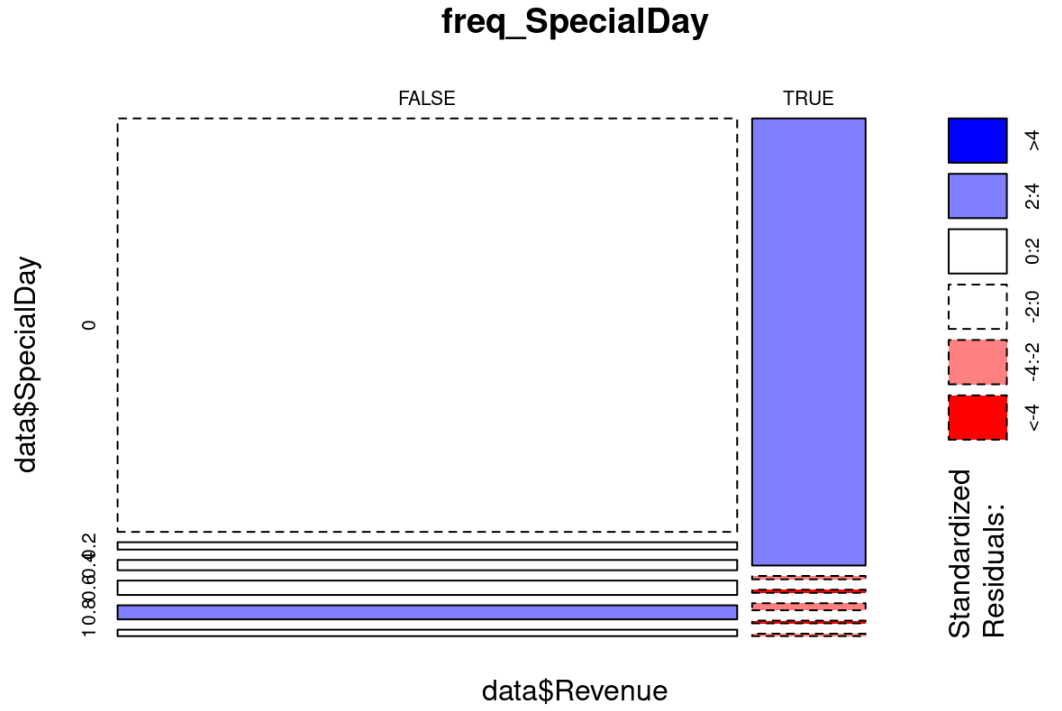
```
freq_SpecialDay=xtabs(~data$Revenue+data$SpecialDay)
prop.table(freq_SpecialDay)
```

```
##             data$SpecialDay
## data$Revenue           0         0.2         0.4         0.6         0.8
##        FALSE 0.750040552 0.013300892 0.018653690 0.026115166 0.025466342
##        TRUE  0.148499594 0.001135442 0.001054339 0.002351987 0.000892133
##             data$SpecialDay
## data$Revenue           1
##        FALSE 0.011678832
##        TRUE  0.000811030
```

```
barplot(prop.table(freq_SpecialDay),legend=rownames(freq_SpecialDay))
```




```
mosaicplot(freq_SpecialDay, border = "black",
           shade = TRUE)
```

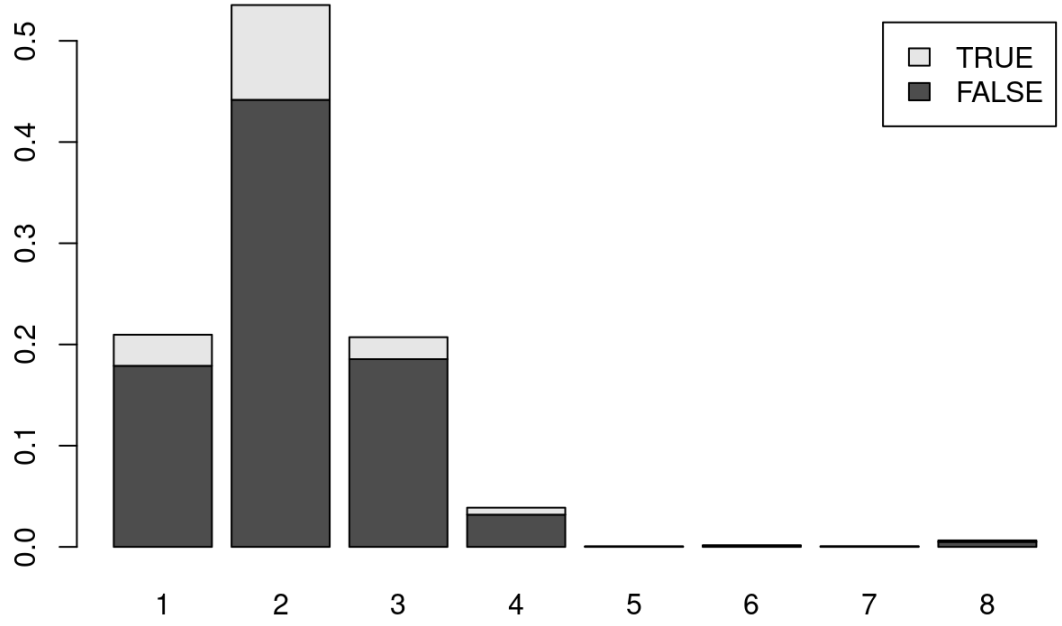


Operating Systems

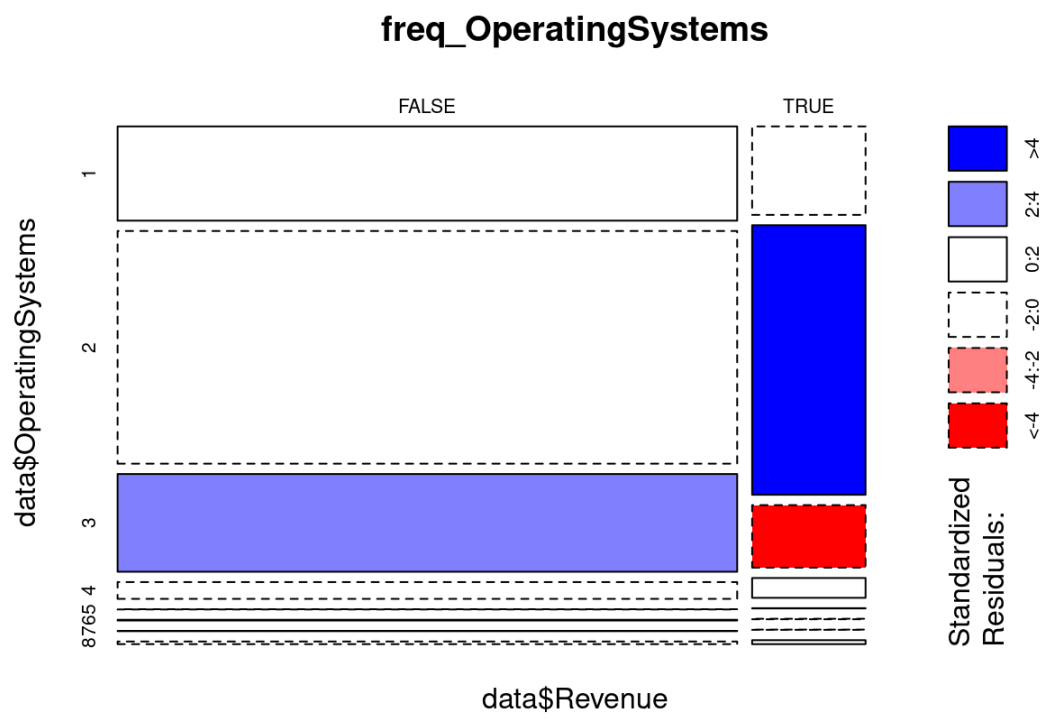
```
freq_OperatingSystems=xtabs(~data$Revenue+data$OperatingSystems)
prop.table(freq_OperatingSystems)
```

```
##             data$OperatingSystems
## data$Revenue           1           2           3           4           5
##        FALSE 0.178913220 0.441686942 0.185482563 0.031873479 0.000405515
##        TRUE  0.030738037 0.093673966 0.021735604 0.006893755 0.000081103
##             data$OperatingSystems
## data$Revenue           6           7           8
##        FALSE 0.001378751 0.000486618 0.005028386
##        TRUE  0.000162206 0.000081103 0.001378751
```

```
barplot(prop.table(freq_OperatingSystems),legend=rownames(freq_OperatingSystems))
```




```
mosaicplot(freq_OperatingSystems, border = "black",
           shade = TRUE)
```

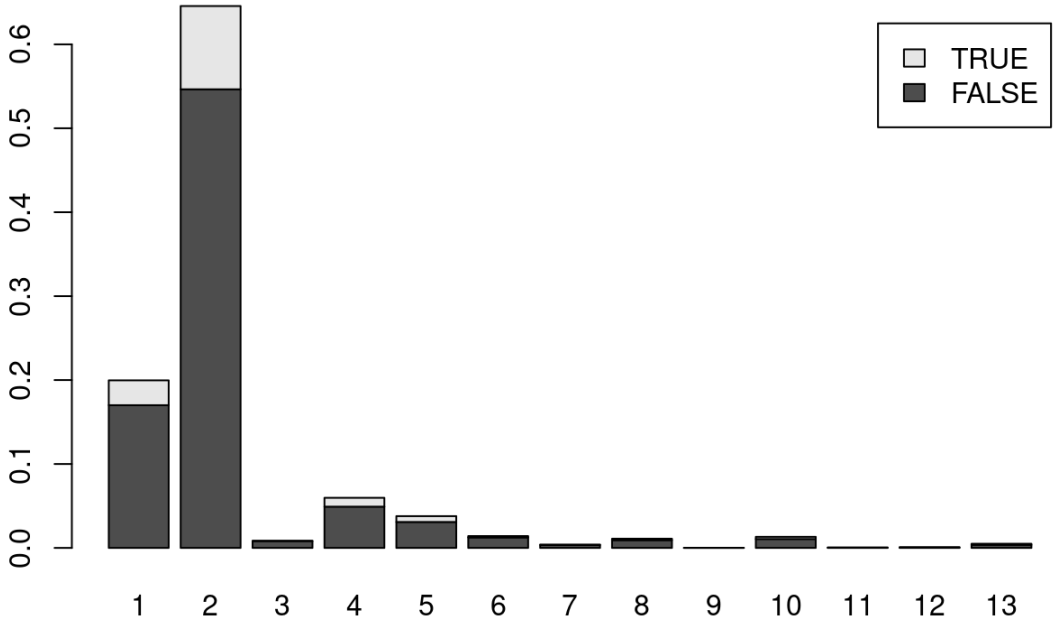


Browser

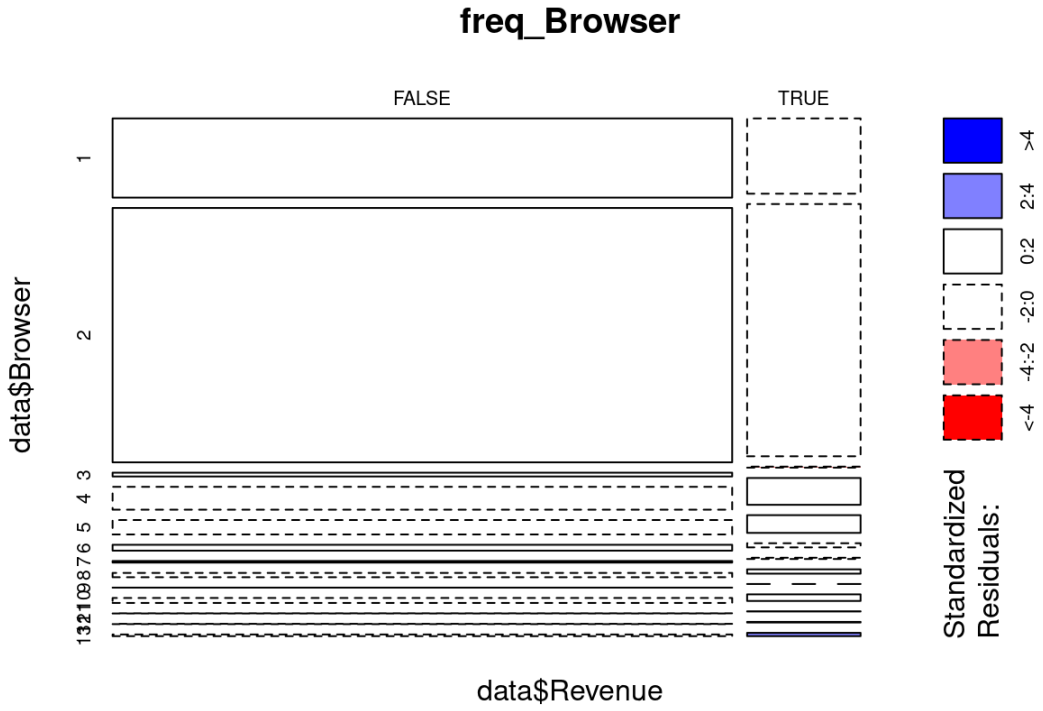
```
freq_Browser=xtabs(~data$Revenue+data$Browser)
prop.table(freq_Browser)
```

```
##             data$Browser
## data$Revenue           1           2           3           4           5
##        FALSE 0.170072993 0.546472019 0.008110300 0.049148418 0.030900243
##        TRUE  0.029602595 0.099188970 0.000405515 0.010543390 0.006974858
##             data$Browser
## data$Revenue           6           7           8           9          10
##        FALSE 0.012489862 0.003487429 0.009245742 0.000081103 0.010624493
##        TRUE  0.001622060 0.000486618 0.001703163 0.000000000 0.002595296
##             data$Browser
## data$Revenue          11          12          13
##        FALSE 0.000405515 0.000567721 0.003649635
##        TRUE  0.000081103 0.000243309 0.001297648
```

```
barplot(prop.table(freq_Browser),legend=rownames(freq_Browser))
```




```
mosaicplot(freq_Browser, border = "black",
           shade = TRUE)
```

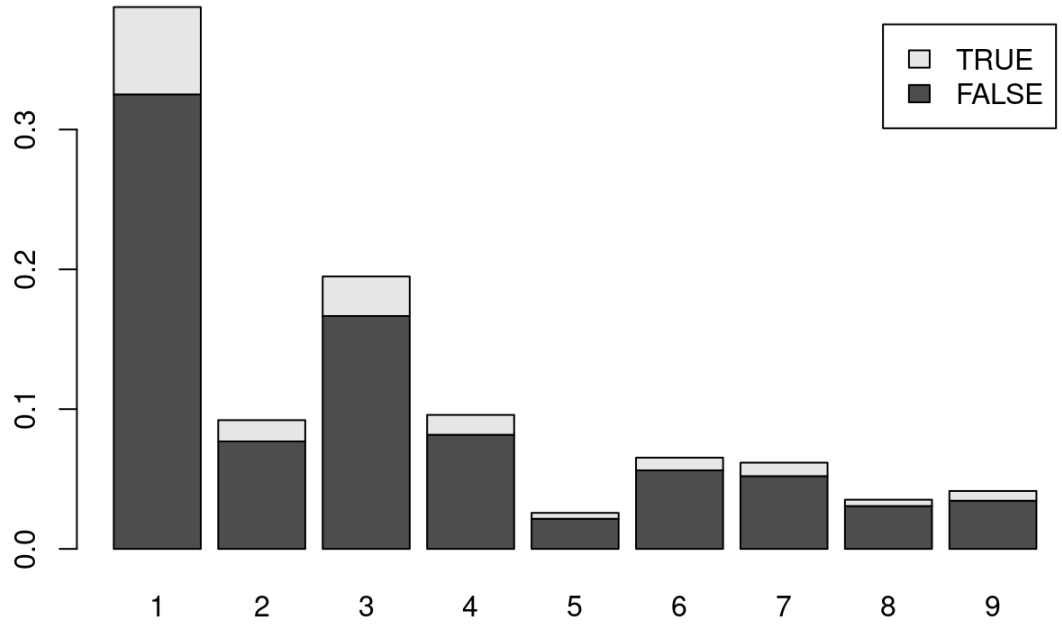


Region

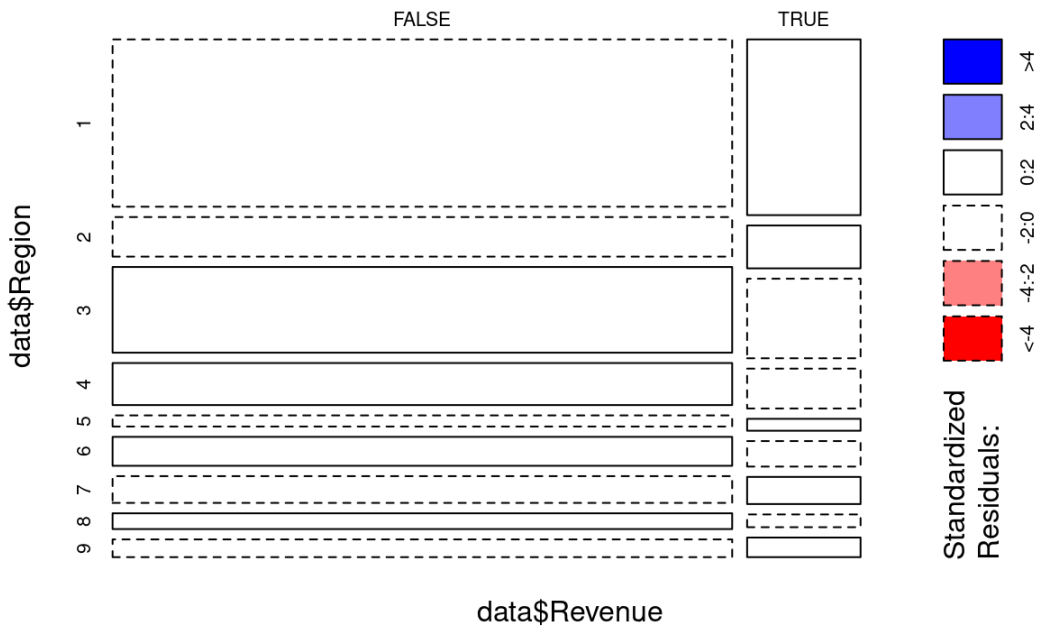
```
freq_Region=xtabs(~data$Revenue+data$Region)
prop.table(freq_Region)
```

```
##             data$Region
## data$Revenue           1           2           3           4           5
##        FALSE 0.325141930 0.076885645 0.166585564 0.081670722 0.021573398
##        TRUE  0.062530414 0.015247364 0.028304947 0.014193025 0.004217356
##             data$Region
## data$Revenue           6           7           8           9
##        FALSE 0.056204380 0.052068127 0.030656934 0.034468775
##        TRUE  0.009083536 0.009651257 0.004541768 0.006974858
```

```
barplot(prop.table(freq_Region),legend=rownames(freq_Region))
```




```
mosaicplot(freq_Region, border = "black",
           shade = TRUE)
```

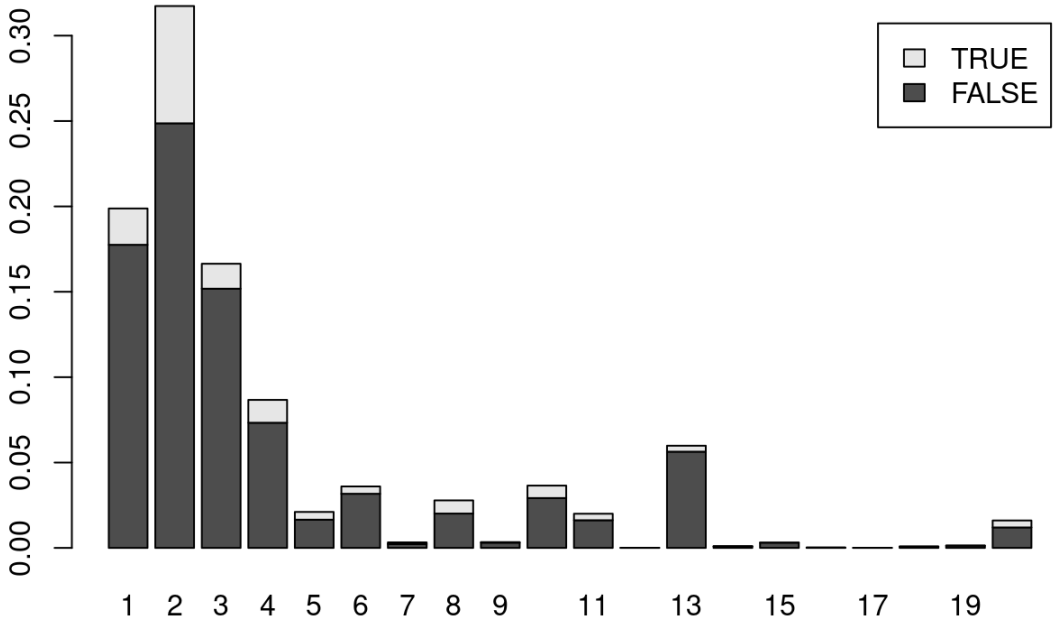


Traffic Type

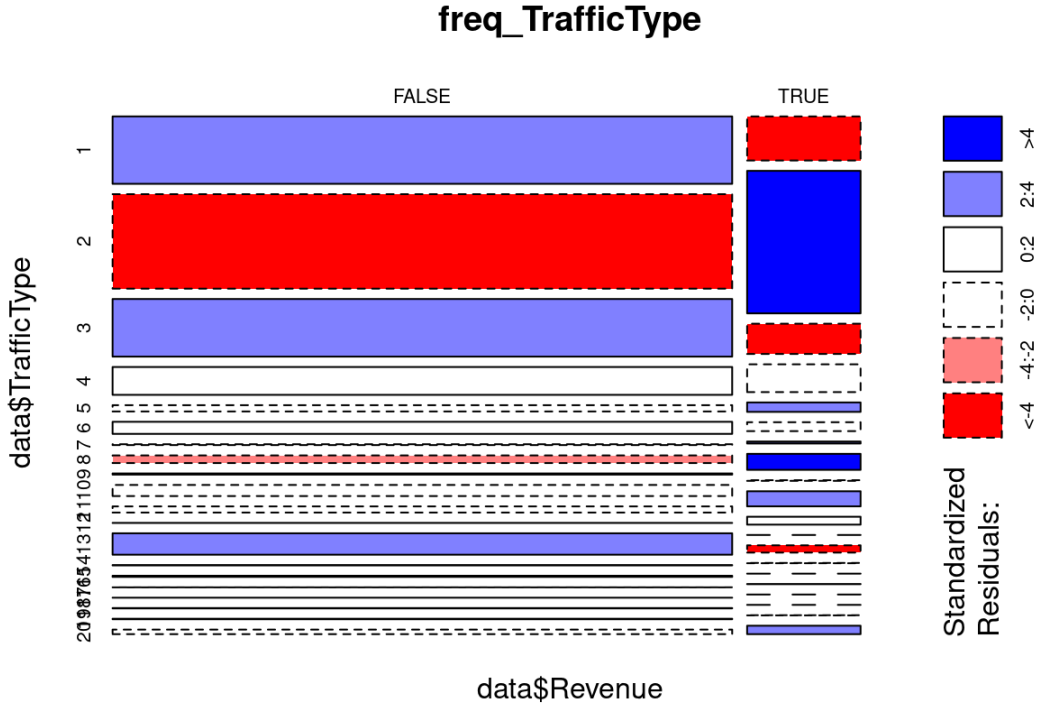
```
freq_TrafficType=xtabs(~data$Revenue+data$TrafficType)
prop.table(freq_TrafficType)
```

```
##             data$TrafficType
## data$Revenue           1           2           3           4           5
##        FALSE 0.177534469 0.248661800 0.151824818 0.073317113 0.016545012
##        TRUE  0.021248986 0.068694242 0.014598540 0.013381995 0.004541768
##             data$TrafficType
## data$Revenue           6           7           8           9          10
##        FALSE 0.031711273 0.002270884 0.020113544 0.003081914 0.029197080
##        TRUE  0.004298459 0.000973236 0.007704785 0.000324412 0.007299270
##             data$TrafficType
## data$Revenue          11          12          13          14          15
##        FALSE 0.016220600 0.000081103 0.056366586 0.000892133 0.003081914
##        TRUE  0.003811841 0.000000000 0.003487429 0.000162206 0.000000000
##             data$TrafficType
## data$Revenue          16          17          18          19          20
##        FALSE 0.000162206 0.000081103 0.000811030 0.001297648 0.012003244
##        TRUE  0.000081103 0.000000000 0.000000000 0.000081103 0.004055150
```

```
barplot(prop.table(freq_TrafficType),legend=rownames(freq_TrafficType))
```




```
mosaicplot(freq_TrafficType, border = "black",
           shade = TRUE)
```

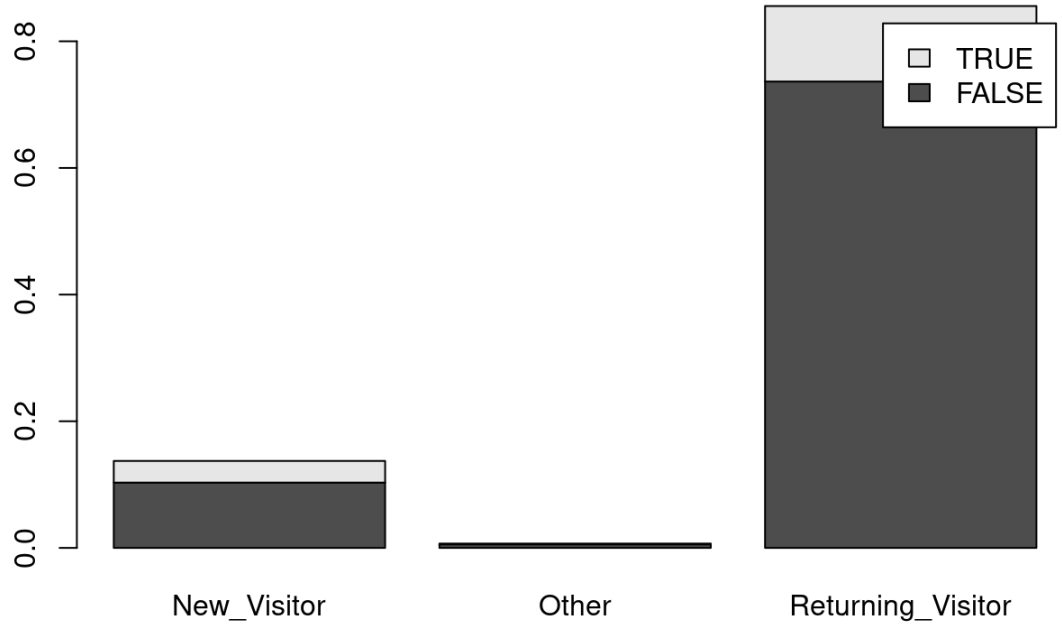


Visitor Type

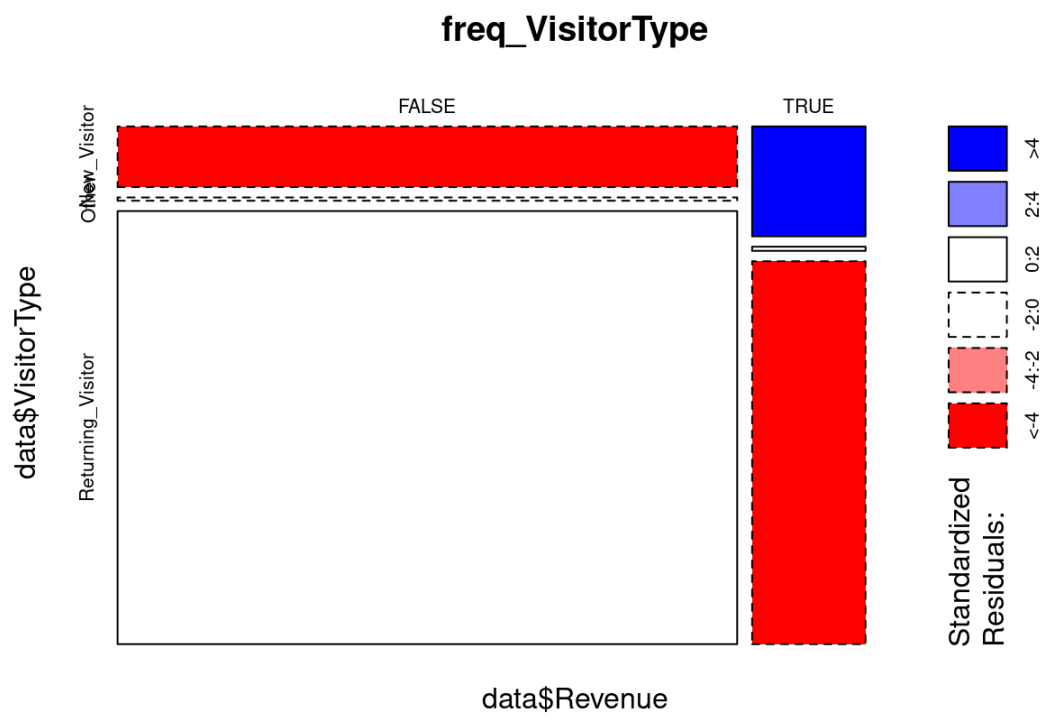
```
freq_VisitorType=xtabs(~data$Revenue+data$VisitorType)
prop.table(freq_VisitorType)
```

```
##             data$VisitorType
## data$Revenue New_Visitor       Other Returning_Visitor
##        FALSE 0.103163017 0.005596107       0.736496350
##        TRUE  0.034225466 0.001297648       0.119221411
```

```
barplot(prop.table(freq_VisitorType),legend=rownames(freq_VisitorType))
```




```
mosaicplot(freq_VisitorType, border = "black",
           shade = TRUE)
```

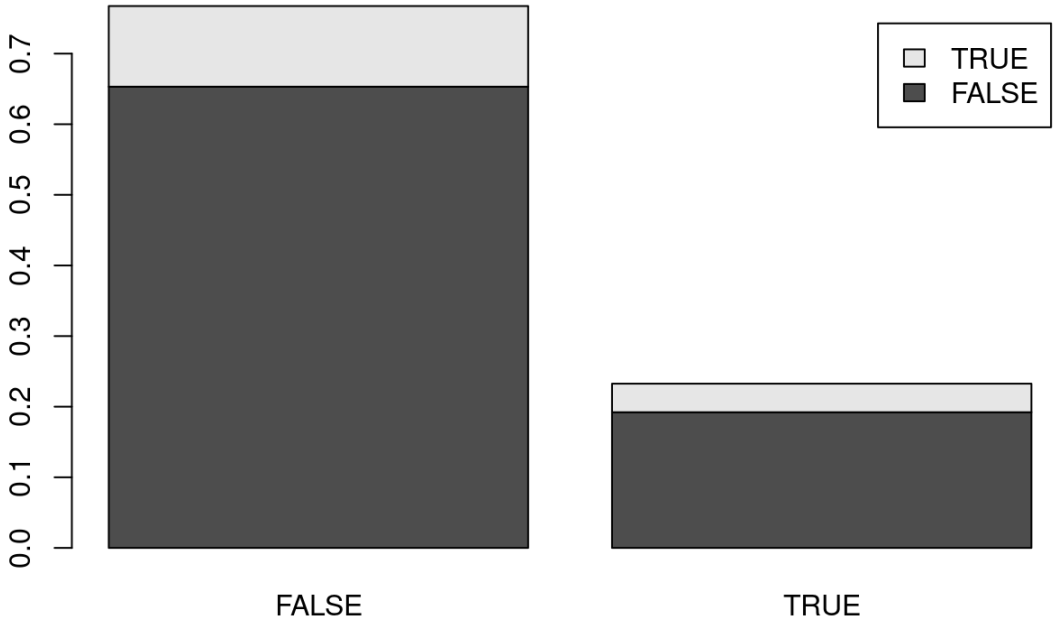


Weekend

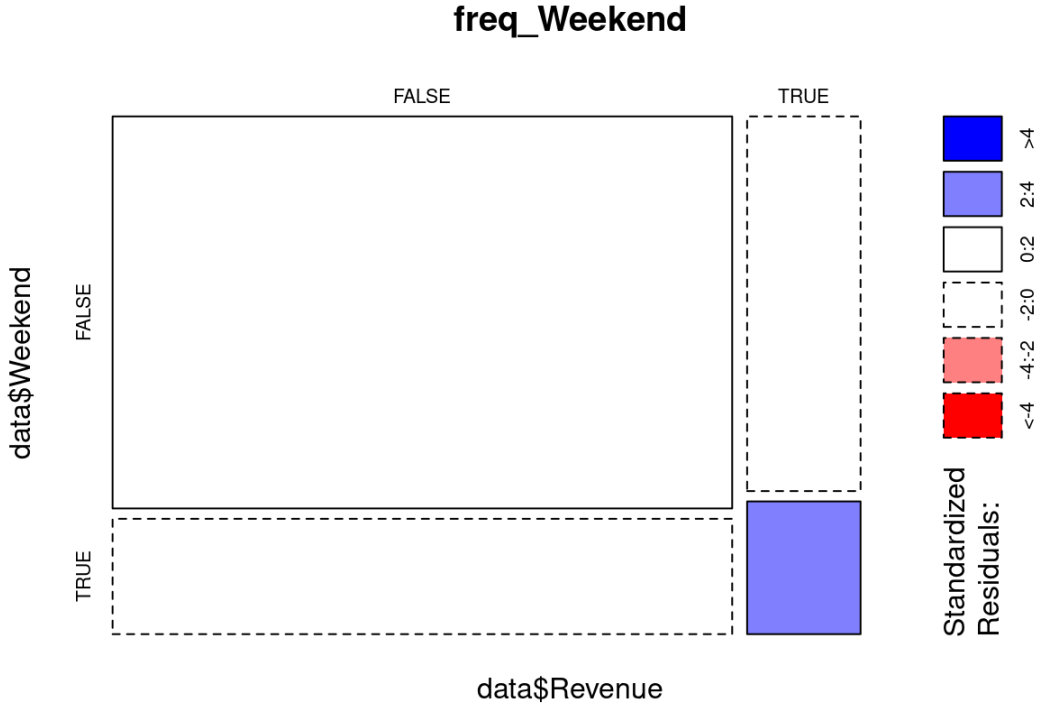
```
freq_Weekend=xtabs(~data$Revenue+data$Weekend)
prop.table(freq_Weekend)
```

```
##             data$Weekend
## data$Revenue     FALSE      TRUE
##        FALSE 0.6531225 0.1921330
##        TRUE  0.1142741 0.0404704
```

```
barplot(prop.table(freq_Weekend),legend=rownames(freq_Weekend))
```




```
mosaicplot(freq_Weekend, border = "black",
           shade = TRUE)
```

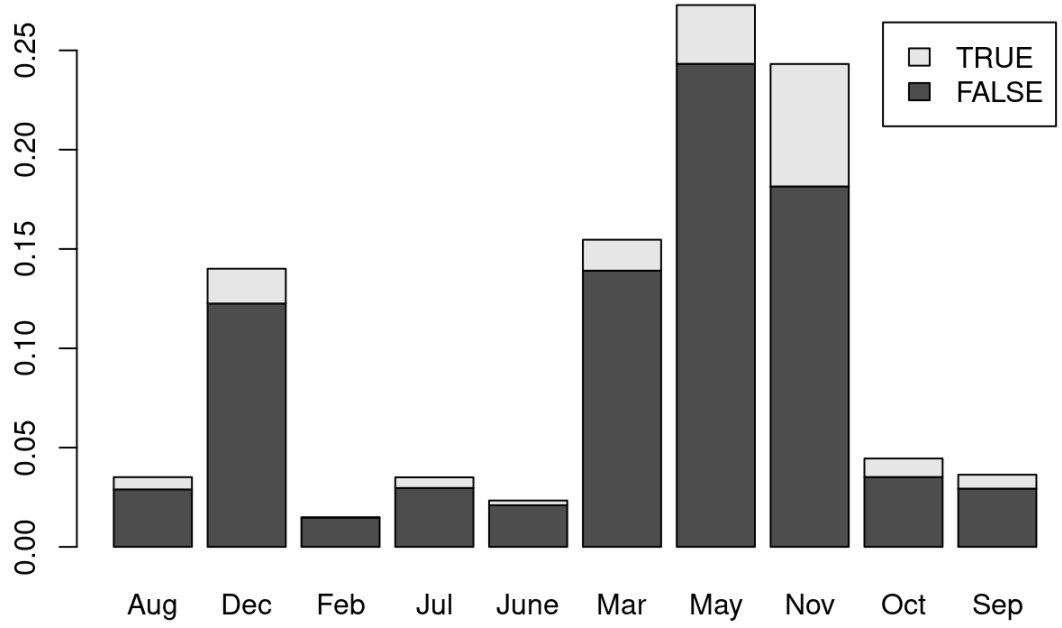


Month

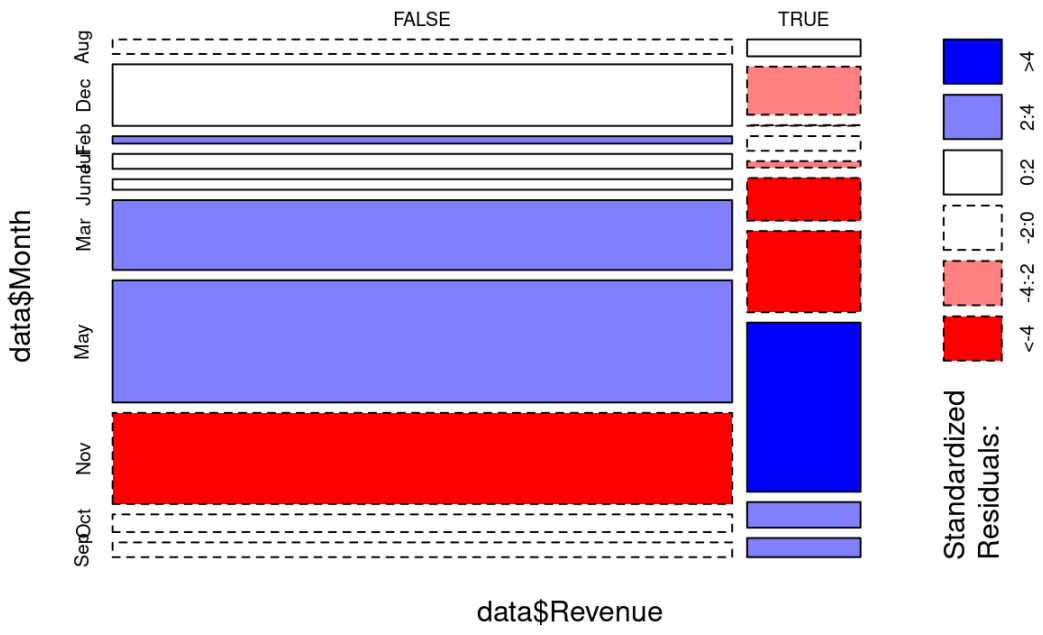
```
freq_Month=xtabs(~data$Revenue+data$Month)
prop.table(freq_Month)
```

```
##             data$Month
## data$Revenue         Aug         Dec         Feb         Jul        June
##        FALSE 0.028953771 0.122546634 0.014679643 0.029683698 0.021005677
##        TRUE  0.006163828 0.017518248 0.000243309 0.005352798 0.002351987
##             data$Month
## data$Revenue         Mar         May         Nov         Oct         Sep
##        FALSE 0.139091646 0.243227899 0.181508516 0.035198702 0.029359286
##        TRUE  0.015571776 0.029602595 0.061638281 0.009326845 0.006974858
```

```
barplot(prop.table(freq_Month),legend=rownames(freq_Month))
```




```
mosaicplot(freq_Month, border = "black",
           shade = TRUE)
```



FEATURE ENGINEERING & MODELLING (Used in Sections 3 and 4 of the report)

Transforming the dataset for Decision tree

```
dataTree = data
```

Changing categorical variables to ordered factors

```
dataTree$OperatingSystems <- factor(dataTree$OperatingSystems, order = TRUE, levels = c(6,3,7,1,5,2,4,8))
dataTree$Browser <- factor(dataTree$Browser, order = TRUE, levels = c(9,3,6,7,1,2,8,11,4,5,10,13,12))
dataTree$Region <- factor(dataTree$Region, order = TRUE, levels = c(8,6,3,4,7,1,5,2,9))
dataTree$TrafficType <- factor(dataTree$TrafficType, order = TRUE, levels = c(12,15,17,18,13,19,3,9,1,6,4,1
4,11,10,5,2,20,8,7,16))
dataTree$Month <- factor(dataTree$Month, order = TRUE, levels =c('Feb', 'Mar', 'May', 'June','Jul', 'Aug',
'Sep','Oct', 'Nov','Dec'))
dataTree$VisitorType <- factor(dataTree$VisitorType, order = TRUE, levels = c('Returning_Visitor', 'Other',
'New_Visitor'))
```

```
str(dataTree)
```

```
## 'data.frame':    12330 obs. of  18 variables:
##  $ Administrative         : int  0 0 0 0 0 0 0 1 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 1 2 10 19 1 0 2 3 ...
##  $ ProductRelated_Duration: num  0 64 0 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                  : Ord.factor w/ 10 levels "Feb"<"Mar"<"May"<..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ OperatingSystems       : Ord.factor w/ 8 levels "6"<"3"<"7"<"1"<..: 4 6 7 2 2 6 6 4 6 6 ...
##  $ Browser                : Ord.factor w/ 13 levels "9"<"3"<"6"<"7"<..: 5 6 5 6 2 6 9 6 6 9 ...
##  $ Region                 : Ord.factor w/ 9 levels "8"<"6"<"3"<"4"<..: 6 6 9 8 6 6 3 6 8 6 ...
##  $ TrafficType            : Ord.factor w/ 20 levels "12"<"15"<"17"<..: 9 16 7 11 11 7 7 15 7 16 ...
##  $ VisitorType            : Ord.factor w/ 3 levels "Returning_Visitor"<..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ Weekend                : logi  FALSE FALSE FALSE FALSE TRUE FALSE ...
##  $ Revenue                : logi  FALSE FALSE FALSE FALSE FALSE FALSE ...
```

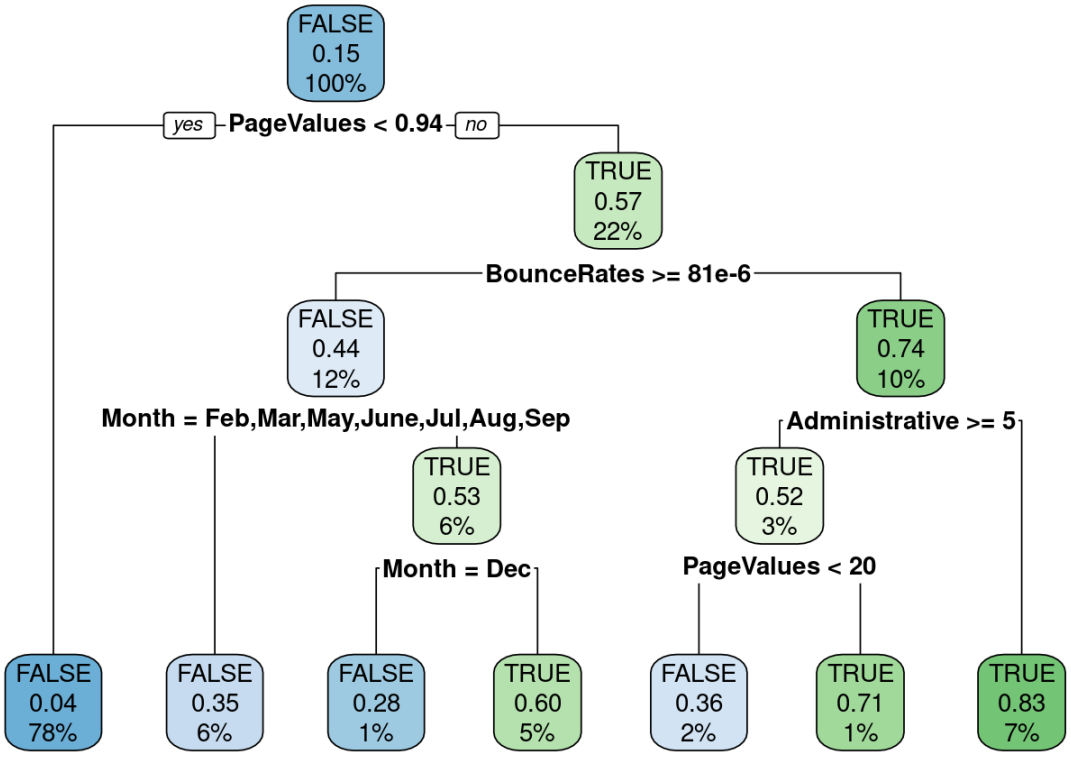
DECISION TREE MODEL

Creating train and test datasets with the ratio of 0.8

```
spl = sample.split(dataTree$Revenue, SplitRatio = 0.8)
train_tree = subset(dataTree, spl==TRUE)
test_tree = subset(dataTree, spl==FALSE)
```

Building decision tree using “rpart”

```
dtree <- rpart(Revenue~., data = train_tree, method = 'class')
rpart.plot(dtree, extra = 'auto')
```

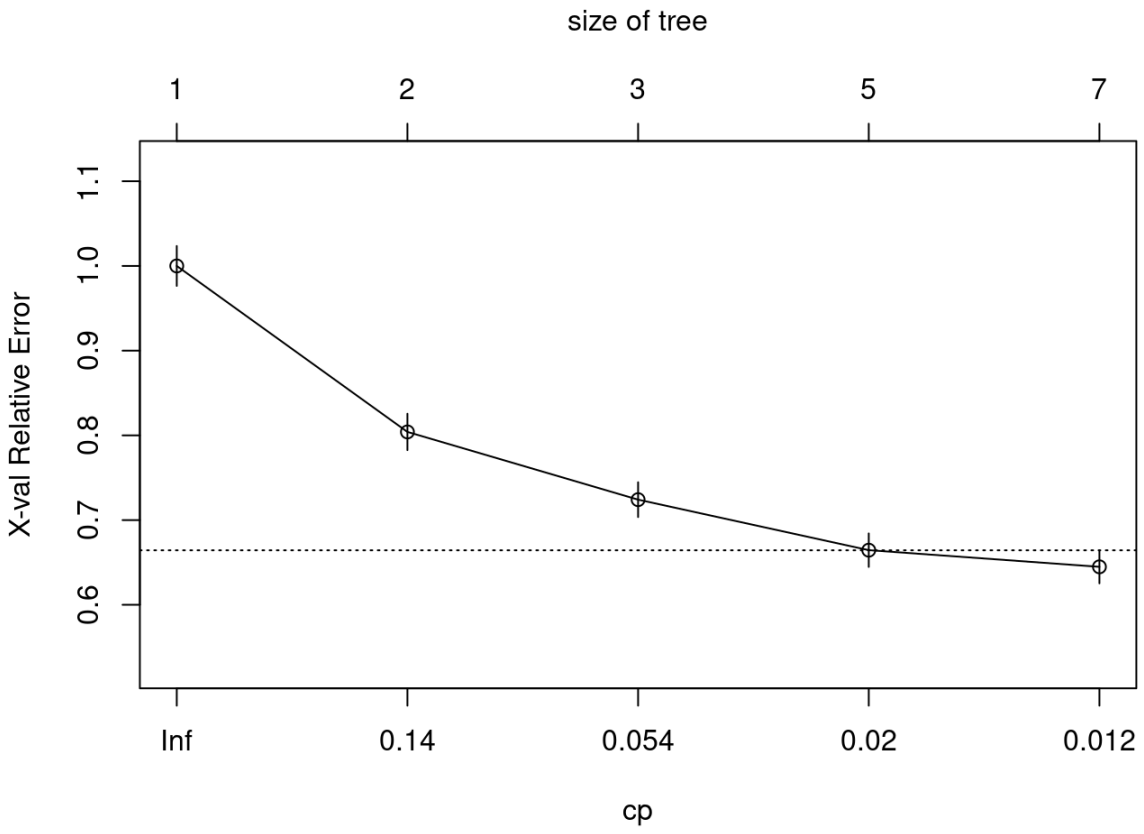


Validate decision tree using complexity parameter (cp) and cross-validation error (xerror)

```
printcp(dtree)
```

```
##
## Classification tree:
## rpart(formula = Revenue ~ ., data = train_tree, method = "class")
##
## Variables actually used in tree construction:
## [1] Administrative BounceRates    Month          PageValues
##
## Root node error: 1526/9864 = 0.1547
##
## n= 9864
##
##         CP nsplit rel error  xerror     xstd
## 1 0.196592      0   1.00000 1.00000 0.023536
## 2 0.101573      1   0.80341 0.80406 0.021479
## 3 0.029161      2   0.70183 0.72412 0.020527
## 4 0.014089      4   0.64351 0.66448 0.019766
## 5 0.010000      6   0.61533 0.64482 0.019504
```

```
plotcp(dtree)
```



Confirm tree has been correctly pruned by manually selecting CP with minimum xerror

```
ptree<- prune(dtree, cp= dtree$cptable[which.min(dtree$cptable[,'xerror']),'CP'])
rpart.plot(ptree)
```

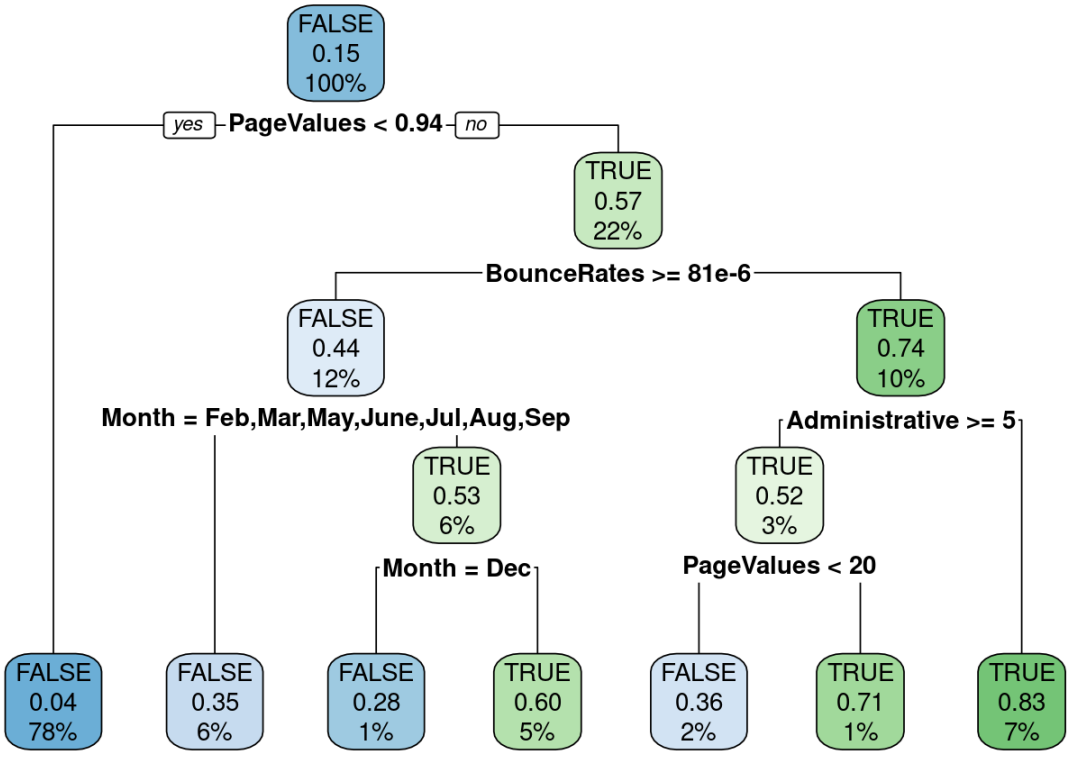


Figure above shows the tree had been pruned by rpart function

Create prediction variable for test data

```
predTree <-predict(ptree, test_tree, type = 'class')
```

Confusion Matrix

```
confmat_Tree <- table(test_tree$Revenue, predTree)
confmat_Tree
```

```
##        predTree
##         FALSE TRUE
##   FALSE  1988   96
##   TRUE    165  217
```

Precision

```
precision_Tree <- confmat_Tree[2,2]/(confmat_Tree[2,2] + confmat_Tree[1,2])
precision_Tree
```

```
## [1] 0.6932907
```

Recall

```
recall_Tree <- confmat_Tree[2,2]/ (confmat_Tree[2,2] + confmat_Tree[2,1])
recall_Tree
```

```
## [1] 0.5680628
```

Accuracy

```
accuracy_Tree <- (confmat_Tree[2,2]+confmat_Tree[1,1])/ nrow(test_tree)
accuracy_Tree
```

```
## [1] 0.8941606
```

Balanced Accuracy

```
balancedacc_Tree <- (confmat_Tree[1,1]/(confmat_Tree[1,1] + confmat_Tree[1,2]) + confmat_Tree[2,2]/ (confma
t_Tree[2,1] + confmat_Tree[2,2]))/2
balancedacc_Tree
```

```
## [1] 0.7609988
```

Transforming the dataset for Regressions

Converting logical variables to binary dummy variables

```
dataReg <- data %>%
  mutate(Revenue_binary = ifelse(Revenue == "FALSE",0,1)) %>%
  mutate(Weekend_binary = ifelse(Weekend == "FALSE",0,1)) %>%
  select (-c(Revenue, Weekend))
```

Converting factor to numeric variable

```
dataReg$Month <- as.numeric(dataReg$Month)
dataReg$VisitorType <- as.numeric(dataReg$VisitorType)
```

```
str(dataReg)
```

```
## 'data.frame':    12330 obs. of  18 variables:
##  $ Administrative         : int  0 0 0 0 0 0 0 1 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 1 2 10 19 1 0 2 3 ...
##  $ ProductRelated_Duration: num  0 64 0 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                  : num  3 3 3 3 3 3 3 3 3 3 ...
##  $ OperatingSystems       : int  1 2 4 3 3 2 2 1 2 2 ...
##  $ Browser                : int  1 2 1 2 3 2 4 2 2 4 ...
##  $ Region                 : int  1 1 9 2 1 1 3 1 2 1 ...
##  $ TrafficType            : int  1 2 3 4 4 3 3 5 3 2 ...
##  $ VisitorType            : num  3 3 3 3 3 3 3 3 3 3 ...
##  $ Revenue_binary         : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Weekend_binary         : num  0 0 0 0 1 0 0 1 0 0 ...
```

Creating correlation matrix

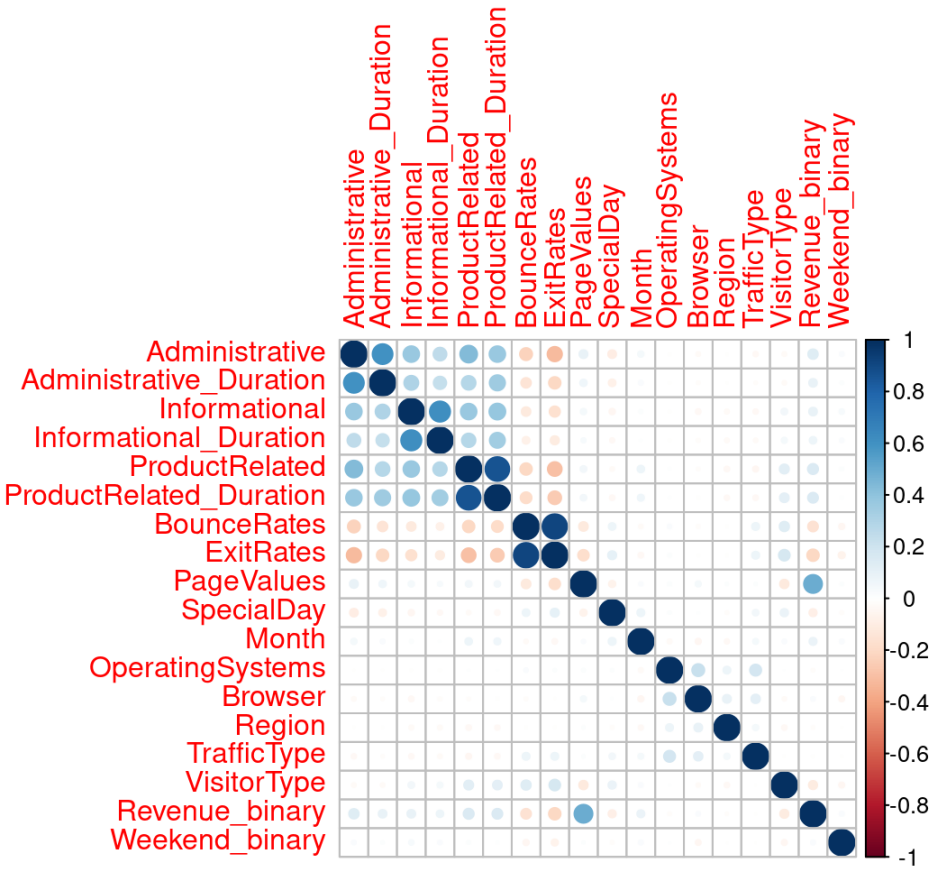
```
cor_result=rcorr(as.matrix(dataReg))
R2 <-  cor_result$r
```

```
cor_result$r
```

```
##                          Administrative Administrative_Duration Informational
## Administrative             1.000000000             0.601583342   0.376850429
## Administrative_Duration    0.601583342             1.000000000   0.302709709
## Informational              0.376850429             0.302709709   1.000000000
## Informational_Duration     0.255848140             0.238030789   0.618954862
## ProductRelated             0.431119340             0.289086621   0.374164291
## ProductRelated_Duration    0.373939013             0.355421954   0.387505306
## BounceRates               -0.223562630            -0.144170410  -0.116113616
## ExitRates                 -0.316482998            -0.205797757  -0.163666061
## PageValues                 0.098989585             0.067608481   0.048631692
## SpecialDay                -0.094777598            -0.073303725  -0.048219254
## Month                      0.048560251             0.029061426   0.019742688
## OperatingSystems          -0.006347063            -0.007343418  -0.009526668
## Browser                   -0.025034572            -0.015391527  -0.038234678
## Region                    -0.005486805            -0.005560563  -0.029168638
## TrafficType               -0.033560713            -0.014376431  -0.034490754
## VisitorType               -0.025819710            -0.023939717   0.055827573
## Revenue_binary             0.138917094             0.093586719   0.095200343
## Weekend_binary             0.026416750             0.014990142   0.035784725
##                          Informational_Duration ProductRelated
## Administrative                      0.255848140    0.431119340
## Administrative_Duration             0.238030789    0.289086621
## Informational                       0.618954862    0.374164291
## Informational_Duration              1.000000000    0.280046268
## ProductRelated                      0.280046268    1.000000000
## ProductRelated_Duration             0.347363577    0.860926836
## BounceRates                        -0.074066610   -0.204577633
## ExitRates                          -0.105275683   -0.292526283
## PageValues                          0.030860874    0.056281794
## SpecialDay                         -0.030576549   -0.023958175
## Month                               0.005987214    0.070298510
## OperatingSystems                   -0.009578676    0.004289621
## Browser                            -0.019284981   -0.013145721
## Region                             -0.027144112   -0.038121842
## TrafficType                        -0.024674908   -0.043064304
## VisitorType                         0.044676760    0.126655811
## Revenue_binary                      0.070344502    0.158537984
## Weekend_binary                      0.024078486    0.016091964
##                          ProductRelated_Duration  BounceRates    ExitRates
## Administrative                       0.373939013 -0.223562630 -0.316482998
## Administrative_Duration              0.355421954 -0.144170410 -0.205797757
## Informational                        0.387505306 -0.116113616 -0.163666061
## Informational_Duration               0.347363577 -0.074066610 -0.105275683
## ProductRelated                       0.860926836 -0.204577633 -0.292526283
## ProductRelated_Duration              1.000000000 -0.184541115 -0.251984097
## BounceRates                         -0.184541115  1.000000000  0.913004396
## ExitRates                           -0.251984097  0.913004396  1.000000000
## PageValues                           0.052823063 -0.119386026 -0.174498310
## SpecialDay                          -0.036379845  0.072702253  0.102241802
## Month                                0.061185682 -0.023762666 -0.039049283
## OperatingSystems                     0.002975790  0.023823182  0.014566735
## Browser                             -0.007380440 -0.015772209 -0.004442355
## Region                              -0.033090520 -0.006485347 -0.008907006
## TrafficType                         -0.036377170  0.078285541  0.078616331
## VisitorType                          0.119329172  0.135536393  0.179143931
## Revenue_binary                       0.152372611 -0.150672912 -0.207071082
## Weekend_binary                       0.007310614 -0.046513997 -0.062587048
##                           PageValues   SpecialDay        Month OperatingSystems
## Administrative           0.09898959 -0.094777598  0.048560251    -0.0063470633
```

```
## Administrative_Duration  0.06760848 -0.073303725  0.029061426    -0.0073434175
## Informational            0.04863169 -0.048219254  0.019742688    -0.0095266679
## Informational_Duration   0.03086087 -0.030576549  0.005987214    -0.0095786764
## ProductRelated           0.05628179 -0.023958175  0.070298510     0.0042896206
## ProductRelated_Duration  0.05282306 -0.036379845  0.061185682     0.0029757898
## BounceRates             -0.11938603  0.072702253 -0.023762666     0.0238231825
## ExitRates               -0.17449831  0.102241802 -0.039049283     0.0145667353
## PageValues               1.00000000 -0.063541272  0.021780268     0.0185079466
## SpecialDay              -0.06354127  1.000000000  0.079341098     0.0126522347
## Month                    0.02178027  0.079341098  1.000000000    -0.0295799600
## OperatingSystems         0.01850795  0.012652235 -0.029579960     1.0000000000
## Browser                  0.04559192  0.003498747 -0.045913324     0.2230128882
## Region                   0.01131530 -0.016097975 -0.032530328     0.0767754856
## TrafficType              0.01253169  0.052301443  0.041839131     0.1891536121
## VisitorType             -0.11122783  0.085556612  0.026481310     0.0015042220
## Revenue_binary           0.49256930 -0.082304598  0.080150468    -0.0146675596
## Weekend_binary           0.01200164 -0.016767155  0.029131513     0.0002842506
##                             Browser        Region  TrafficType  VisitorType
## Administrative          -0.025034572 -0.0054868053 -0.033560713 -0.025819710
## Administrative_Duration -0.015391527 -0.0055605628 -0.014376431 -0.023939717
## Informational           -0.038234678 -0.0291686379 -0.034490754  0.055827573
## Informational_Duration  -0.019284981 -0.0271441124 -0.024674908  0.044676760
## ProductRelated          -0.013145721 -0.0381218417 -0.043064304  0.126655811
## ProductRelated_Duration -0.007380440 -0.0330905198 -0.036377170  0.119329172
## BounceRates             -0.015772209 -0.0064853474  0.078285541  0.135536393
## ExitRates               -0.004442355 -0.0089070060  0.078616331  0.179143931
## PageValues               0.045591919  0.0113152995  0.012531693 -0.111227826
## SpecialDay               0.003498747 -0.0160979746  0.052301443  0.085556612
## Month                   -0.045913324 -0.0325303281  0.041839131  0.026481310
## OperatingSystems         0.223012888  0.0767754856  0.189153612  0.001504222
## Browser                  1.000000000  0.0973928492  0.111938224 -0.021866988
## Region                   0.097392849  1.0000000000  0.047520231 -0.036190794
## TrafficType              0.111938224  0.0475202313  1.000000000 -0.002839178
## VisitorType             -0.021866988 -0.0361907939 -0.002839178  1.000000000
## Revenue_binary           0.023984289 -0.0115950678 -0.005112971 -0.104725722
## Weekend_binary          -0.040260864 -0.0006906703 -0.002221229 -0.043679249
##                         Revenue_binary Weekend_binary
## Administrative             0.138917094   0.0264167503
## Administrative_Duration    0.093586719   0.0149901419
## Informational              0.095200343   0.0357847251
## Informational_Duration     0.070344502   0.0240784862
## ProductRelated             0.158537984   0.0160919642
## ProductRelated_Duration    0.152372611   0.0073106138
## BounceRates               -0.150672912  -0.0465139965
## ExitRates                 -0.207071082  -0.0625870480
## PageValues                 0.492569295   0.0120016392
## SpecialDay                -0.082304598  -0.0167671553
## Month                      0.080150468   0.0291315131
## OperatingSystems          -0.014667560   0.0002842506
## Browser                    0.023984289  -0.0402608638
## Region                    -0.011595068  -0.0006906703
## TrafficType               -0.005112971  -0.0022212292
## VisitorType               -0.104725722  -0.0436792493
## Revenue_binary             1.000000000   0.0292953680
## Weekend_binary             0.029295368   1.0000000000
```

```
corrplot(R2, method="circle")
```



Creating train and test sets for regression models using ratio of 0.7

```
spl2 = sample.split(dataReg$Revenue, SplitRatio = 0.7)
trainReg = subset(dataReg, spl2==TRUE)
testReg = subset(dataReg, spl2==FALSE)
```

LINEAR REGRESSION MODEL

First linear regression including all of the datasets variables

```
lm1 <-lm(ProductRelated_Duration ~., data = dataReg)
summary(lm1)
```

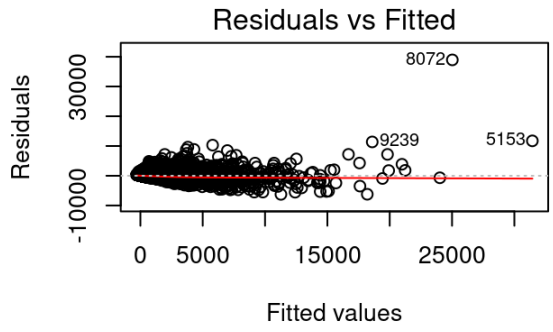
```
##
## Call:
## lm(formula = ProductRelated_Duration ~ ., data = dataReg)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -6390   -278    -78    158  39083
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)              -1.445e+02  4.841e+01  -2.986 0.002832 **
## Administrative           -5.869e+01  3.438e+00 -17.073  < 2e-16 ***
## Administrative_Duration   1.642e+00  5.912e-02  27.780  < 2e-16 ***
## Informational             1.131e+01  8.828e+00   1.282 0.199992
## Informational_Duration    1.360e+00  7.526e-02  18.070  < 2e-16 ***
## ProductRelated            3.574e+01  2.237e-01 159.744  < 2e-16 ***
## BounceRates              -2.199e+03  4.322e+02  -5.087 3.69e-07 ***
## ExitRates                 2.156e+03  4.559e+02   4.728 2.29e-06 ***
## PageValues               -3.136e-01  5.177e-01  -0.606 0.544687
## SpecialDay               -1.273e+02  4.239e+01  -3.004 0.002674 **
## Month                     2.684e+00  3.541e+00   0.758 0.448438
## OperatingSystems          1.514e+00  9.488e+00   0.160 0.873182
## Browser                   4.790e+00  4.999e+00   0.958 0.338018
## Region                    1.514e+00  3.480e+00   0.435 0.663481
## TrafficType               4.395e-01  2.115e+00   0.208 0.835420
## VisitorType               2.858e+01  1.259e+01   2.270 0.023238 *
## Revenue_binary            9.105e+01  2.695e+01   3.379 0.000731 ***
## Weekend_binary           -3.461e+01  1.970e+01  -1.757 0.078924 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 919.8 on 12312 degrees of freedom
## Multiple R-squared:  0.7693, Adjusted R-squared:  0.769
## F-statistic:  2415 on 17 and 12312 DF,  p-value: < 2.2e-16
```

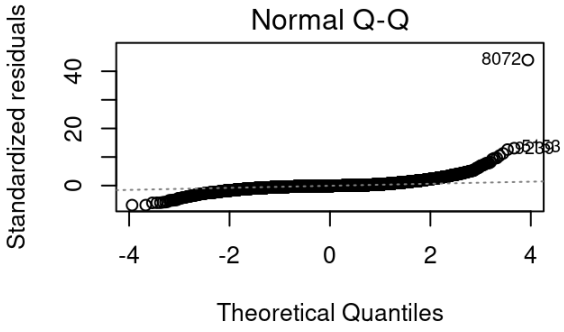
Creating a second linear regression that removed the statisticly insignificant results from the original regression and made the months a binary variable for each month

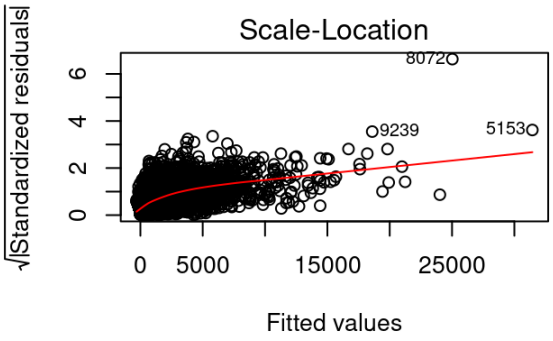
```
lm2 <- lm(ProductRelated_Duration ~ Administrative + Administrative_Duration + Informational_Duration + Pro
ductRelated + BounceRates + ExitRates + SpecialDay + factor(Month) + VisitorType + Weekend_binary + Revenue
_binary, data = dataReg)
summary(lm2)
```

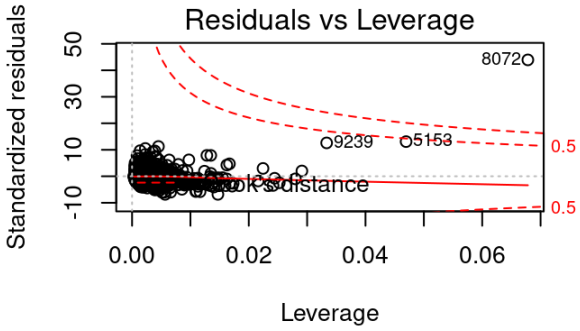
```
##
## Call:
## lm(formula = ProductRelated_Duration ~ Administrative + Administrative_Duration +
##     Informational_Duration + ProductRelated + BounceRates + ExitRates +
##     SpecialDay + factor(Month) + VisitorType + Weekend_binary +
##     Revenue_binary, data = dataReg)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -6257   -284    -79    162  38948
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)              -2.691e+02  5.661e+01  -4.754 2.02e-06 ***
## Administrative           -5.646e+01  3.408e+00 -16.565  < 2e-16 ***
## Administrative_Duration   1.641e+00  5.897e-02  27.832  < 2e-16 ***
## Informational_Duration    1.404e+00  6.238e-02  22.512  < 2e-16 ***
## ProductRelated            3.581e+01  2.241e-01 159.757  < 2e-16 ***
## BounceRates              -2.193e+03  4.302e+02  -5.097 3.51e-07 ***
## ExitRates                 2.164e+03  4.534e+02   4.773 1.84e-06 ***
## SpecialDay               -1.322e+02  4.790e+01  -2.760 0.005796 **
## factor(Month)2            1.988e+02  4.944e+01   4.021 5.82e-05 ***
## factor(Month)3            2.328e+02  8.198e+01   2.839 0.004528 **
## factor(Month)4            6.625e-01  6.248e+01   0.011 0.991540
## factor(Month)5            6.342e+01  6.995e+01   0.907 0.364593
## factor(Month)6            1.926e+02  4.909e+01   3.924 8.75e-05 ***
## factor(Month)7            1.618e+02  4.807e+01   3.367 0.000763 ***
## factor(Month)8            1.872e+02  4.732e+01   3.956 7.67e-05 ***
## factor(Month)9            1.607e+01  5.908e+01   0.272 0.785630
## factor(Month)10           1.772e+02  6.192e+01   2.862 0.004212 **
## VisitorType               2.809e+01  1.262e+01   2.225 0.026080 *
## Weekend_binary           -3.505e+01  1.966e+01  -1.783 0.074658 .
## Revenue_binary            8.545e+01  2.392e+01   3.573 0.000354 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 918.1 on 12310 degrees of freedom
## Multiple R-squared:  0.7702, Adjusted R-squared:  0.7698
## F-statistic:  2171 on 19 and 12310 DF,  p-value: < 2.2e-16
```

```
par(mfrow=c(2,2))
plot (lm2)
```









Creating a train model

```
trainlm2 <-lm(ProductRelated_Duration ~ Administrative + Administrative_Duration + Informational_Duration +
ProductRelated + BounceRates + ExitRates + SpecialDay + factor(Month) + VisitorType + Weekend_binary + Reve
nue_binary, data = trainReg)
summary(trainlm2)
```

```
##
## Call:
## lm(formula = ProductRelated_Duration ~ Administrative + Administrative_Duration +
##     Informational_Duration + ProductRelated + BounceRates + ExitRates +
##     SpecialDay + factor(Month) + VisitorType + Weekend_binary +
##     Revenue_binary, data = trainReg)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -6429   -281    -71    173  37386
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)              -2.854e+02  6.901e+01  -4.136 3.57e-05 ***
## Administrative           -7.174e+01  4.124e+00 -17.395  < 2e-16 ***
## Administrative_Duration   1.835e+00  6.943e-02  26.426  < 2e-16 ***
## Informational_Duration    1.717e+00  7.734e-02  22.200  < 2e-16 ***
## ProductRelated            3.624e+01  2.719e-01 133.263  < 2e-16 ***
## BounceRates              -2.142e+03  5.208e+02  -4.114 3.93e-05 ***
## ExitRates                 2.095e+03  5.512e+02   3.802 0.000145 ***
## SpecialDay               -1.470e+02  5.805e+01  -2.533 0.011340 *
## factor(Month)2            2.249e+02  6.085e+01   3.697 0.000220 ***
## factor(Month)3            2.876e+02  9.957e+01   2.888 0.003881 **
## factor(Month)4            6.776e+01  7.702e+01   0.880 0.379038
## factor(Month)5            5.400e+01  8.759e+01   0.616 0.537584
## factor(Month)6            2.273e+02  6.033e+01   3.768 0.000166 ***
## factor(Month)7            2.053e+02  5.917e+01   3.470 0.000523 ***
## factor(Month)8            2.394e+02  5.828e+01   4.109 4.01e-05 ***
## factor(Month)9            1.098e+02  7.181e+01   1.529 0.126289
## factor(Month)10           2.651e+02  7.535e+01   3.519 0.000436 ***
## VisitorType               2.206e+01  1.524e+01   1.447 0.147905
## Weekend_binary           -3.329e+01  2.384e+01  -1.396 0.162660
## Revenue_binary            5.046e+01  2.905e+01   1.737 0.082405 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 932.8 on 8611 degrees of freedom
## Multiple R-squared:  0.7724, Adjusted R-squared:  0.7719
## F-statistic:  1538 on 19 and 8611 DF,  p-value: < 2.2e-16
```

Testing the model

```
testlm2 <-predict(trainlm2, testReg)
lm2rmse <- rmse(actual = testReg$ProductRelated_Duration,predicted = testlm2)
print(lm2rmse)
```

```
## [1] 890.5697
```

Testing the new model against the first model using the same train and test data

```
trainlm1 <- lm(ProductRelated_Duration ~., data = trainReg)

testlm1 <-predict(trainlm1, testReg)
lm1rmse <- rmse(actual = testReg$ProductRelated_Duration,predicted = testlm1)
print(lm1rmse)
```

```
## [1] 892.8322
```

LOGISTIC REGRESSION MODEL

Building the model on training data

```
glm = glm(Revenue_binary ~ .,data=trainReg, family=binomial)
summary(glm)
```

```
##
## Call:
## glm(formula = Revenue_binary ~ ., family = binomial, data = trainReg)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -5.9173  -0.4874  -0.3591  -0.1750   3.3359
##
## Coefficients:
##                            Estimate Std. Error z value Pr(>|z|)
## (Intercept)              -2.332e+00  2.112e-01 -11.040  < 2e-16 ***
## Administrative            1.467e-03  1.286e-02   0.114  0.90917
## Administrative_Duration -5.594e-05  2.210e-04  -0.253  0.80016
## Informational             4.682e-02  3.168e-02   1.478  0.13941
## Informational_Duration    8.590e-06  2.718e-04   0.032  0.97479
## ProductRelated            4.299e-03  1.309e-03   3.285  0.00102 **
## ProductRelated_Duration   2.317e-05  2.992e-05   0.774  0.43872
## BounceRates              -1.731e+00  3.660e+00  -0.473  0.63630
## ExitRates                -1.615e+01  2.829e+00  -5.710 1.13e-08 ***
## PageValues                7.685e-02  2.749e-03  27.955  < 2e-16 ***
## SpecialDay               -1.048e+00  2.597e-01  -4.037 5.42e-05 ***
## Month                     1.266e-01  1.648e-02   7.681 1.58e-14 ***
## OperatingSystems         -9.471e-02  4.526e-02  -2.093  0.03638 *
## Browser                   2.640e-02  2.194e-02   1.203  0.22891
## Region                   -1.594e-02  1.548e-02  -1.030  0.30308
## TrafficType               1.537e-03  9.712e-03   0.158  0.87424
## VisitorType              -1.343e-01  5.059e-02  -2.655  0.00792 **
## Weekend_binary            1.165e-01  8.384e-02   1.390  0.16467
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 7438.7  on 8630  degrees of freedom
## Residual deviance: 5188.7  on 8613  degrees of freedom
## AIC: 5224.7
##
## Number of Fisher Scoring iterations: 7
```

Prediction on the training set

```
predictglm = predict(glm, type="response")
summary(predictglm)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## 0.001013 0.046595 0.087863 0.154791 0.152877 1.000000
```

Using tapply function to compute the average prediction for each of the outcomes

```
tapply(predictglm, trainReg$Revenue_binary, mean)
```

```
##         0         1
## 0.1032209 0.4363799
```

Selecting a threshold value

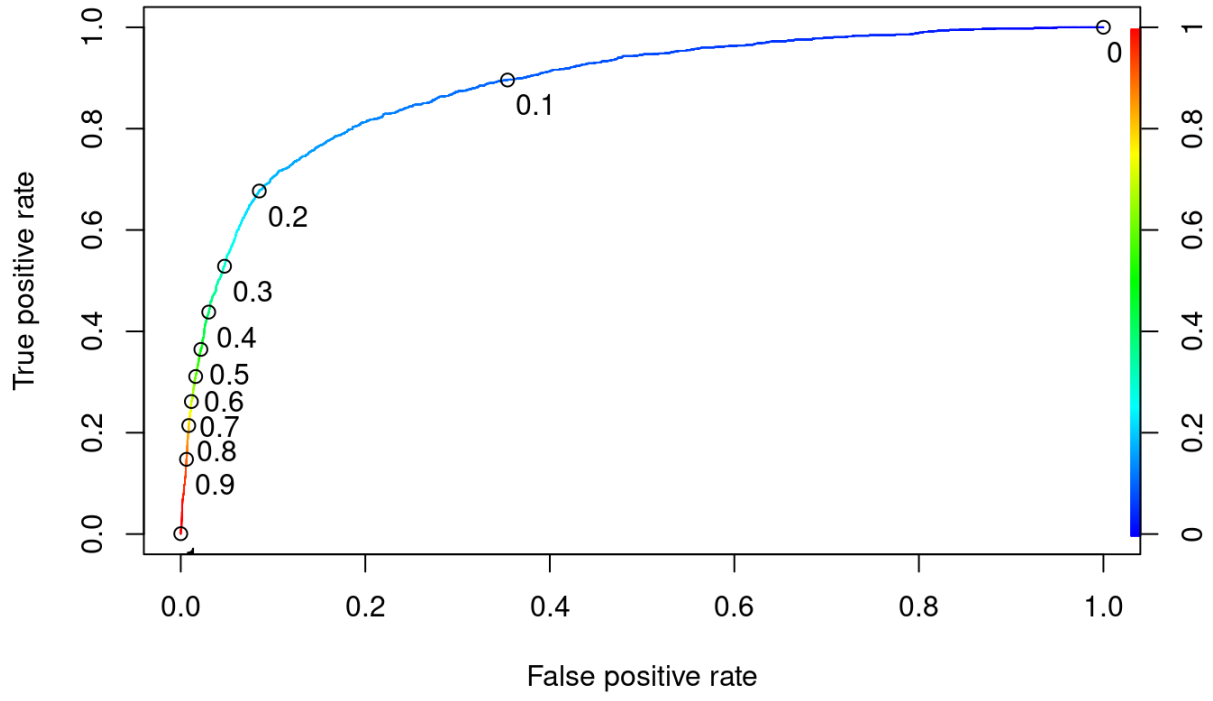
```
ROCRpred = prediction(predictglm, trainReg$Revenue_binary)
```

Calculating performance function

```
ROCRperf = performance(ROCRpred, "tpr", "fpr")
```

Plotting ROC curve with threshold labels

```
ROCRperf = performance(ROCRpred, "tpr", "fpr")
plot(ROCRperf, colorize=TRUE)
plot(ROCRperf, colorize=TRUE, print.cutoffs.at=seq(0,1,by=0.1), text.adj=c(-0.2,1.7))
```



Prediction on test set using the selected threshold

```
predictTest = predict(glm, type = "response", newdata = testReg)
```

```
table(testReg$Revenue_binary,predictTest >= 0.3)
```

```
##
##     FALSE TRUE
##   0  2997  130
##   1   258  314
```

```
confmat_Glm <- table(testReg$Revenue_binary,predictTest >= 0.3)
confmat_Glm
```

```
##
##     FALSE TRUE
##   0  2997  130
##   1   258  314
```

Precision

```
precision_Glm <- confmat_Glm[2,2]/(confmat_Glm[2,2] + confmat_Glm[1,2])
precision_Glm
```

```
## [1] 0.7072072
```

Recall

```
recall_Glm <- confmat_Glm[2,2]/ (confmat_Glm[2,2] + confmat_Glm[2,1])
recall_Glm
```

```
## [1] 0.548951
```

Accuracy

```
accuracy_Glm <- (confmat_Glm[2,2]+confmat_Glm[1,1])/nrow(testReg)
accuracy_Glm
```

```
## [1] 0.8951068
```

Balanced Accuracy

```
balancedacc_Glm <- (confmat_Glm[1,1]/(confmat_Glm[1,1] + confmat_Glm[1,2]) + confmat_Glm[2,2]/ (confmat_Glm
[2,1] + confmat_Glm[2,2]))/2
balancedacc_Glm
```

```
## [1] 0.7536888
```

Root mean square error

```
testReg = testReg %>%
  mutate(predictions_quad = predict(glm, testReg))

sqrt(testReg %>%
  summarise(RMSE_glm = mean((Revenue_binary-predictions_quad)^2)))
```

RMSE_glm <dbl>
2.90655

1 row