Online Shopper Intention Analysis

Filipp Krasovsky, Sai Thiha

12/12/2020

Technical Overview: Implementation is done using R Version 4.0.3 (2020-10-10) – "Bunny-Wunnies Freak OOut" data pull conducted using a .csv file located in the same directory as the .rmd document.

```
#import dataset
shoppers <- read.csv("online_shoppers_intention.csv")
library('cluster')
library('dendextend')
library('factoextra')
library(ggvis)
library(ggplot2)
library("ggdendro")
library("infotheo")
library("dplyr")
library("rms")
head(shoppers)
```

```
##   Administrative Administrative_Duration Informational Informational_Duration
## 1              0                       0             0                      0
## 2              0                       0             0                      0
## 3              0                       0             0                      0
## 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                0.000000  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
```

Overview of variables:

1.ProductRelated/ProductRelated_Duration - refers to the number of URLs viewed by a user in the ProductRelated category during a session as well as the total amount of time spent in each session browsing this type of URL in seconds. Both of these are Ratio values with a meaningful zero point, that is - zero indicates an absence of any browsing activity in this context.

2. Month - a nominal variable used for our time series analysis component.
3. Informational/Informational_Duration - refers to the same context as #1 but for URLs categorized as Informational.
4. Bounce Rates - refers to the rate at which a website is left without triggering any further correspondence with the Google analytics server. This is a numeric ratio level variable.
5. Exit Rates - refers to the frequency at which a given page was the last in a browsing session. This is a numeric ratio level variable.

6.PageValues - refers to the average value of a web page that the user visited prior to completing a purchase. This is a numeric ratio level variable where zero indicates a non-arbitrary lack of page value.

7. TrafficType - this is a nominal variable which refers to the different types of web traffic that lead to the website. Although they are not mapped directly to corresponding sources, we can assume some examples include redirects, search results, and direct navigation to a URL.
8. VisitorType - a nominal variable for whether a user has been on the website before.

9.Revenue - a nominal binary variable and the dependent variable of interest for this analysis, where True refers to the completion of a purchase.

10. SpecialDay - a ratio variable that determines proximity of a given browsing session to a holiday.

```
#data transformations and discretization

#subsetting
shoppers <- subset(
  shoppers,
  select = c(
    ProductRelated,
    ProductRelated_Duration,
    Informational,
    Informational_Duration,
    Month,
    ExitRates,
    PageValues,
    TrafficType,
    VisitorType,
    SpecialDay,
    Revenue
  )
)

#Dependent Variable Transformtation
shoppers$Revenue <- ifelse(shoppers$Revenue=="True",yes = 1,no = 0)

#set up an alias for shoppers
df <- shoppers
```

Having filtered for the variables of interest needed for our modeling approach, we then move on to visualization and possible reduction. In assessing the duration-url variabes, we can impose our subject matter knowledge on the issue to assert that if the number of URLs browsed for a given category is zero, then the duration for that URL ought to also be zero.

```
#get count of missing values
sapply(df,function(x) sum(is.na(x)))
```

```
##          ProductRelated ProductRelated_Duration           Informational
##                       0                       0                     128
##  Informational_Duration                   Month              ExitRates
##                       0                       0                       0
##              PageValues             TrafficType             VisitorType
##                     135                       0                       0
##              SpecialDay                 Revenue
##                       0                       0
```

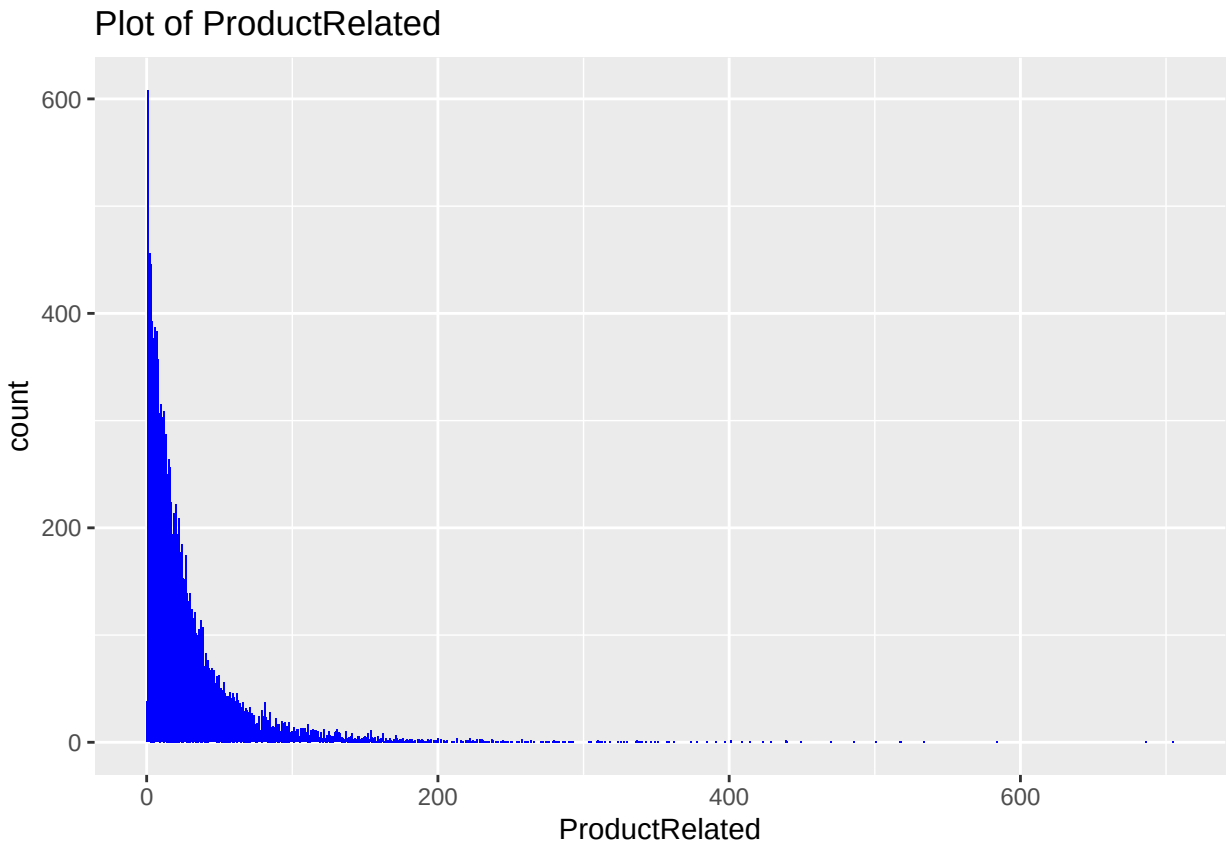
An initial overview shows us a mostly complete dataset with missing values for Informational and PageValues. Since we have no Apriori information on how page values are calculated, we have no choice but to omit these rows.

Similarly, for Informational, we could use a proxy to remove some missing values by asserting that if the browsing duration is zero, the number of URLs must also be zero. This would be incorrect, however, as Google Analytics only begins to record a session after a trigger fires off, so it's conceivable that a user may have some browsing history. In any case, our large sample size justifies removal of both.

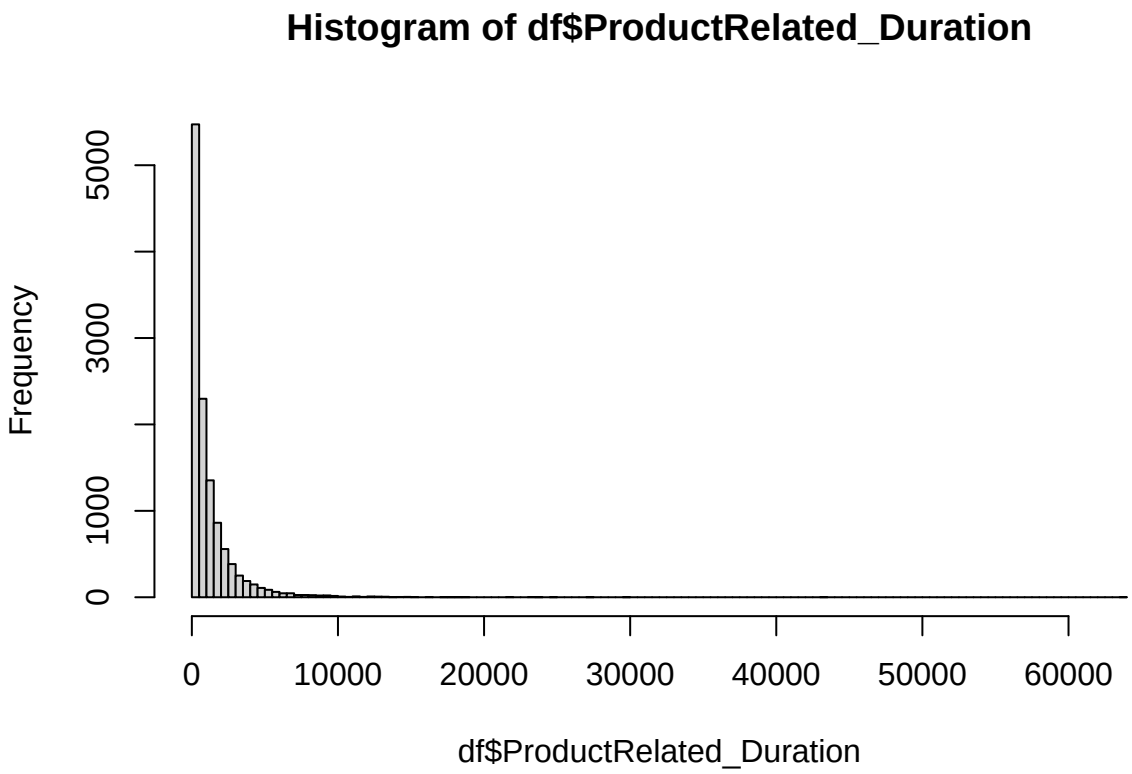
```
#remove missing values for Informational and PageValues
df <- subset(df,!is.na(df$Informational))
df <- subset(df,!is.na(df$PageValues))
```

```
#plot product related, product related duration, and a graph with both.

ggplot(df)+geom_bar(aes(x=ProductRelated),fill="blue")+ggtitle("Plot of ProductRelated")
```

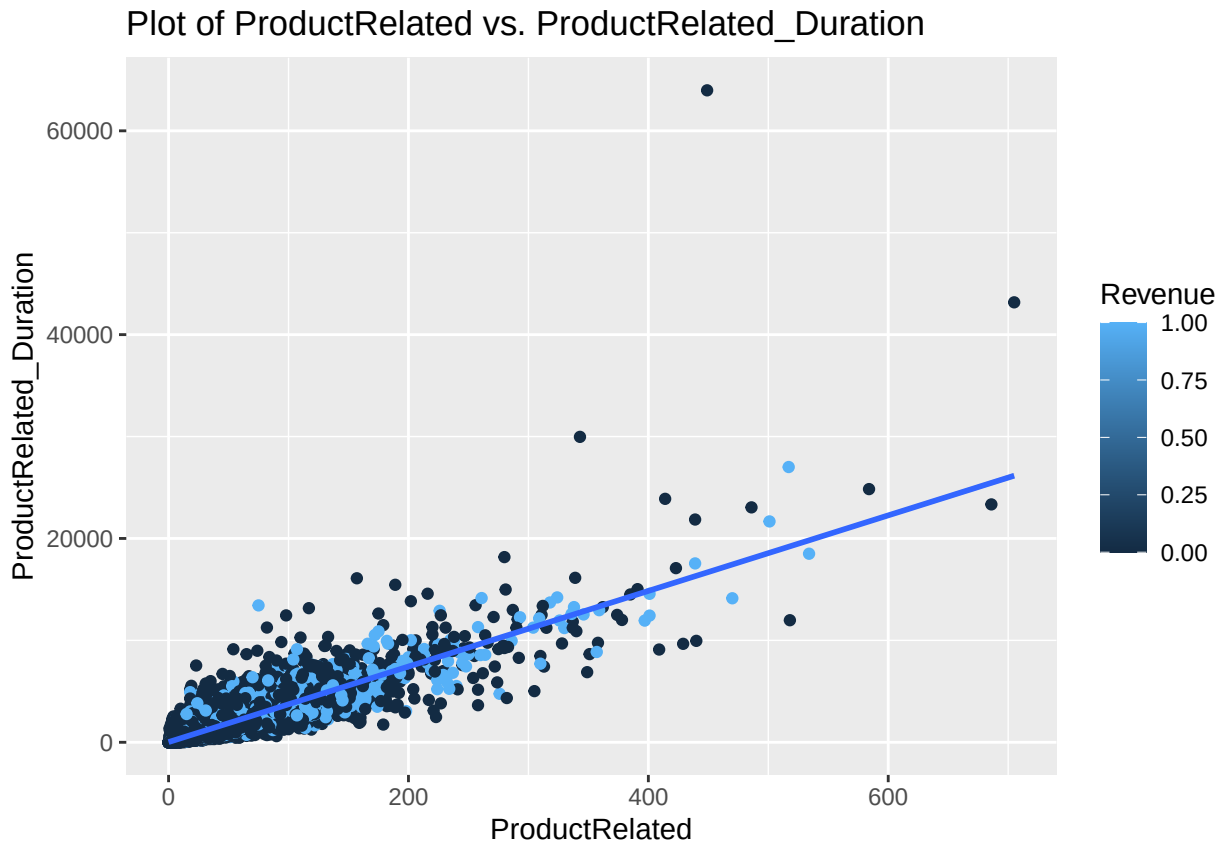



```
hist(df$ProductRelated_Duration,breaks = 100,freq = TRUE)
```




```
ggplot(df, aes(ProductRelated,ProductRelated_Duration,color=Revenue))+geom_point()+geom_smooth(method =
```

```
## `geom_smooth()` using formula 'y ~ x'
```



ProdudctRelated exhibits a strong right skew with a large concentration of values as zeroes - suggesting that many sessions involved no ProductRelated URLs. We can make a similar observation about the browsing duration, although we must note in the final graph that the two exhibit what appears to be a strong correlation with each other. Coloring these pairs with the revenue variable doesn't provide much insight at this point, but we can make assertions about dimension reduction since the two variables move closely with each other:

```
#print summary statistics for both

print(summary(df$ProductRelated))
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     0.0     7.0    18.0    31.7    38.0   705.0
```

```
paste("Variance:",var(df$ProductRelated))
```

```
## [1] "Variance: 1969.20544592326"
```

```
print(summary(df$ProductRelated_Duration))
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     0.0   184.5   599.0  1194.4  1463.8 63973.5
```

```
paste("Variance:",var(df$ProductRelated_Duration))
```

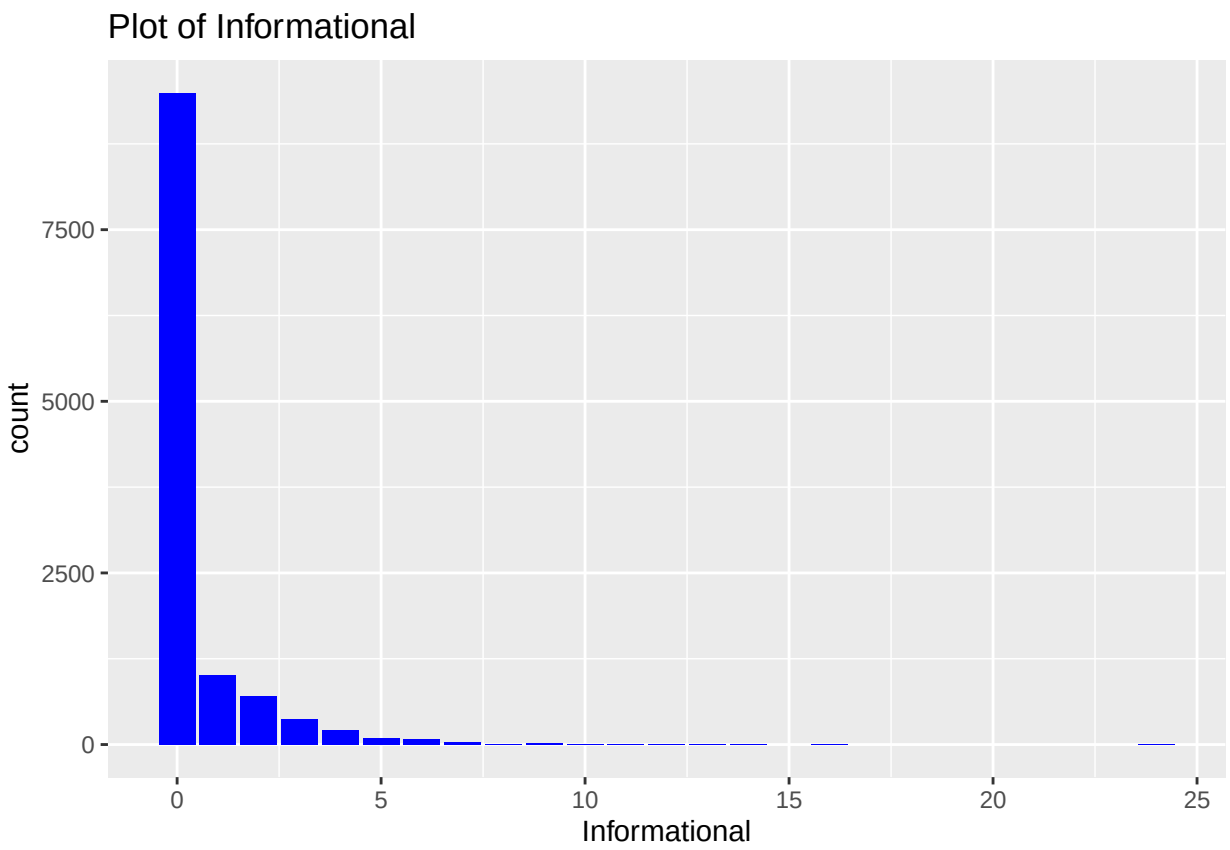
```
## [1] "Variance: 3663462.41825697"
```

```
paste("Correlation:",cor(df$ProductRelated,df$ProductRelated_Duration))
```

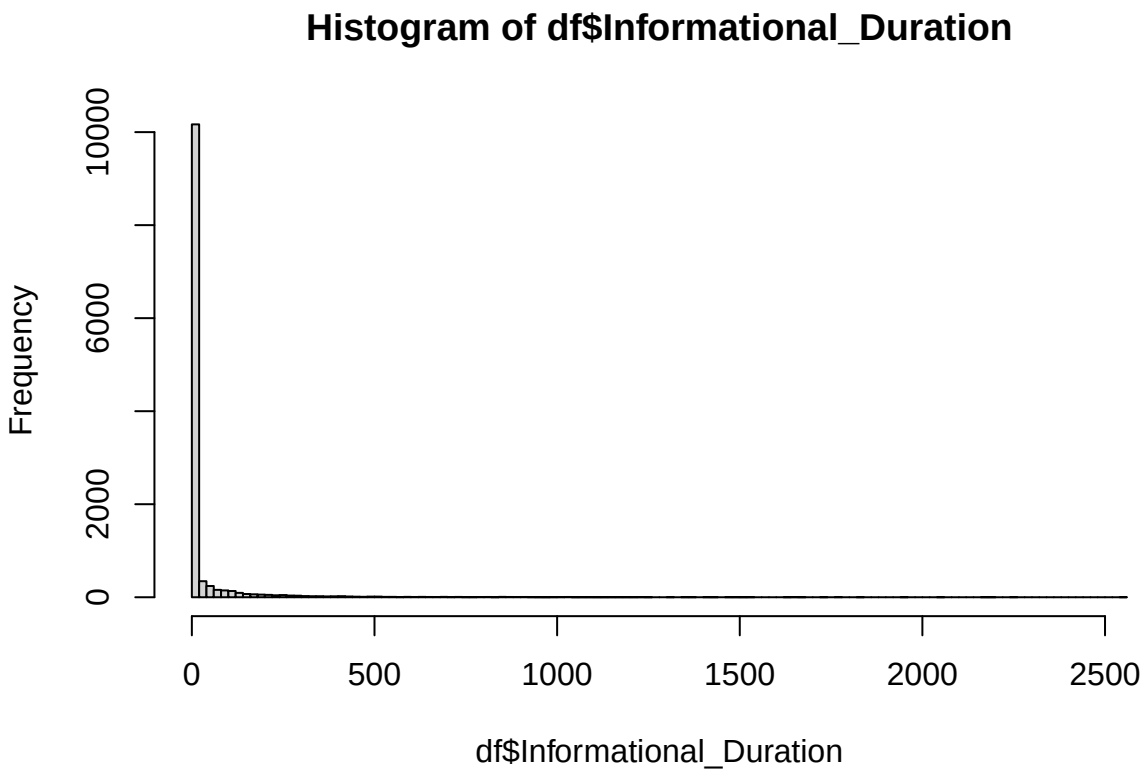
```
## [1] "Correlation: 0.859609224249491"
```

at an 86% correlation, we may consider removing one of the two variables in our modeling.

```
ggplot(df)+geom_bar(aes(x=Informational),fill="blue")+ggtitle("Plot of Informational")
```

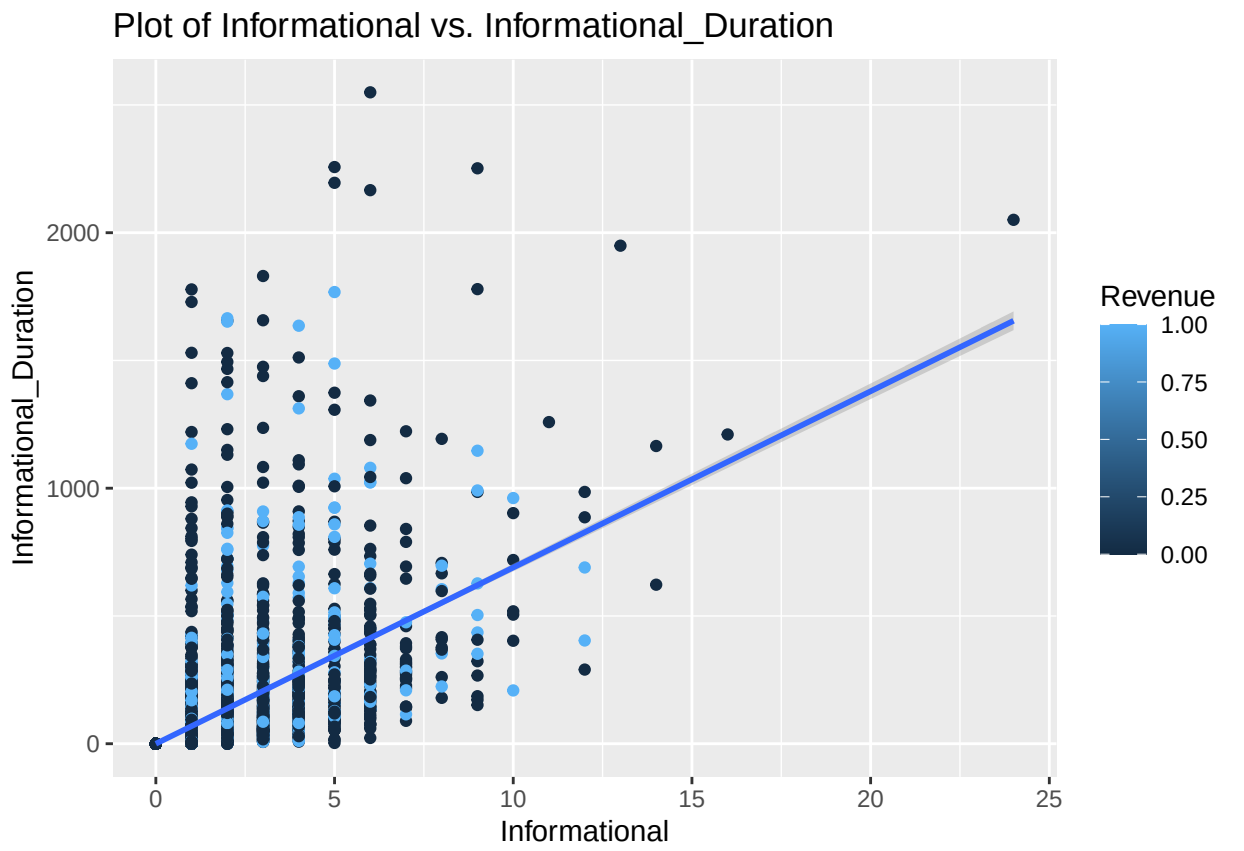



```
hist(df$Informational_Duration,breaks = 100,freq = TRUE)
```




```
ggplot(df, aes(Informational,Informational_Duration,color=Revenue))+geom_point()+geom_smooth(method = "
```

```
## `geom_smooth()` using formula 'y ~ x'
```




```
print(summary(df$Informational))
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0000  0.0000  0.0000  0.5029  0.0000 24.0000
```

```
paste("Variance:",var(df$Informational))
```

```
## [1] "Variance: 1.61312159260586"
```

```
print(summary(df$Informational_Duration))
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00    0.00    0.00   34.56    0.00 2549.38
```

```
paste("Variance:",var(df$Informational_Duration))
```

```
## [1] "Variance: 19983.4861709258"
```

```
paste("Correlation:",cor(df$Informational,df$Informational_Duration))
```

```
## [1] "Correlation: 0.619719845039567"
```

Initial observations suggest that Informational and it's duration are not strongly correlated enough to run into a colinearity problem, but we still have a case to make for removing Informational on the grounds that it's a near variance predictor, which we suspect due to its variance being ~1. The formal rule of thumb requires the following to be true:

1. The ratio of the first most frequent value to the second most frequent value is large
2. The number of unique values relative to the sample size is small ()

We know the two most frequent values are 0 and 1:

```
fr = length(df$Informational[df$Informational==0])/length(df$Informational[df$Informational==1])
paste("frequency ratio:",fr)
```

```
## [1] "frequency ratio: 9.33824975417896"
```

Which is sufficiently high to raise suspicion. After we inspect the ratio of unique values to n:

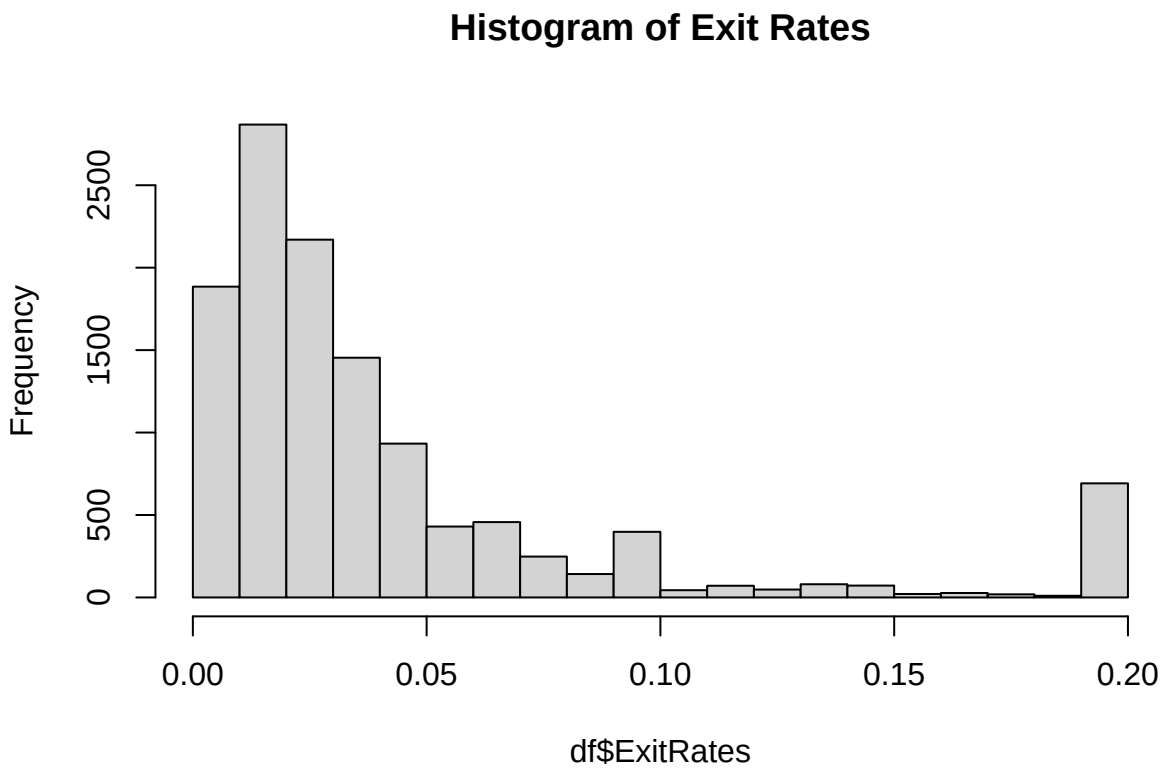
```
ur = length(unique(df$Informational)) / length(df$Informational)
paste("Unique values to sample:",ur)
```

```
## [1] "Unique values to sample: 0.00140856740409313"
```

We can conclude that Informational can be disposed of in a modeling context.

```
#delete Informational column
df$Informational <- NULL
```

```
#Observations on Exit Rates
hist(df$ExitRates,main = "Histogram of Exit Rates")
```




```
print(summary(df$ExitRates))
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.00000 0.01429 0.02501 0.04298 0.05000 0.20000
```

```
paste("std dev:",sd(df$ExitRates))
```

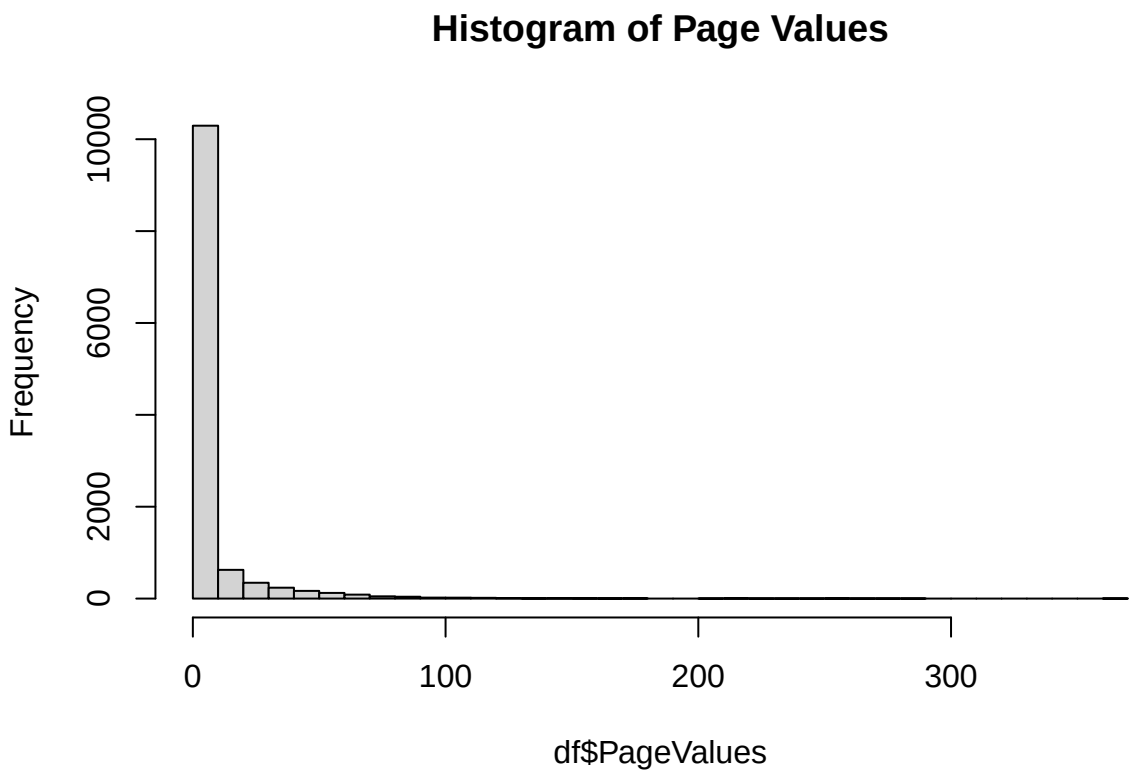
```
## [1] "std dev: 0.0485203862618832"
```

We can observe a right skewed distribution of exit rates where the expected exit rate is 4% and a median exit rate

of 2%. This dimension does not justify discretization - although it contains a somewhat multimodaldistribution, it also exhibits continuous behavior that offsets this dynamic.

This variable appears to be dispersed enough to be an appropriate input for modeling purposes, and we can begin exploring its relationship with others from this point, namely page value.

```
#Observations on Page Values
hist(df$PageValues,main = "Histogram of Page Values",breaks = 30)
```




```
print(summary(df$PageValues))
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   0.000   0.000   5.934   0.000 361.764
```

```
paste("std dev:",sd(df$PageValues))
```

```
## [1] "std dev: 18.6823854705661"
```

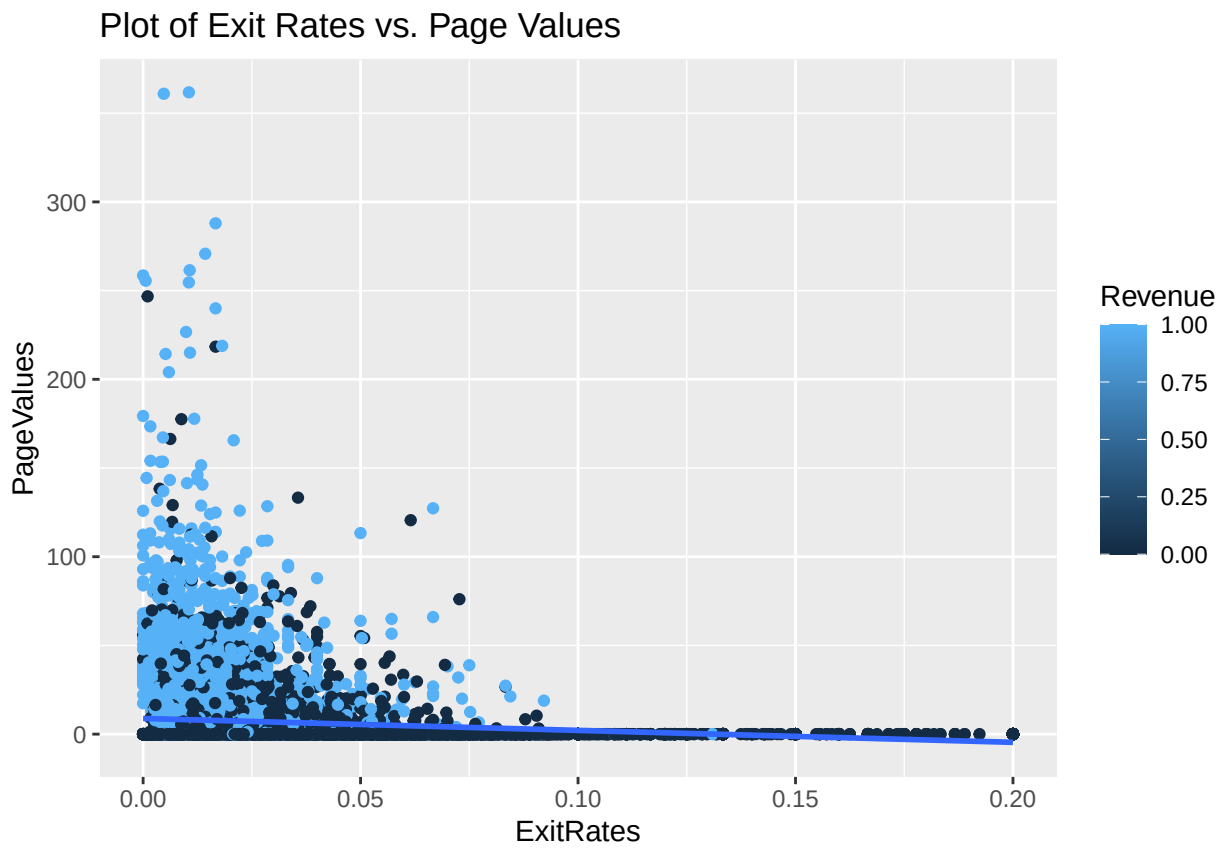
```
paste("unique to n ratio:",length(unique(df$PageValues)) / length(df$PageValues))
```

```
## [1] "unique to n ratio: 0.220233656475267"
```

Page values seems to exhibit a strong right skew and the majority of values are near zero. It's possible that this might be related to URL browsing activity, but first we can inspect it's relationship with exit rates:

```
df <- subset(df,!is.na(df$PageValues))
ggplot(df, aes(ExitRates,PageValues,color=Revenue))+geom_point()+geom_smooth(method = "lm")+ggtitle("Pl
```

```
## `geom_smooth()` using formula 'y ~ x'
```



While we do not see any direct linear correlation between exit rates and page values, we do notice that the majority of nonzero page values occur at a value of exit rates $< 10\%$. Similarly, coloring this distrubtion with Revenue, we observe that almost all sales occur on pages below this threshhold. From here, we can justify creating a discrete component for this variable. We retain the unmodified value for redundancy in case this analysis fails.

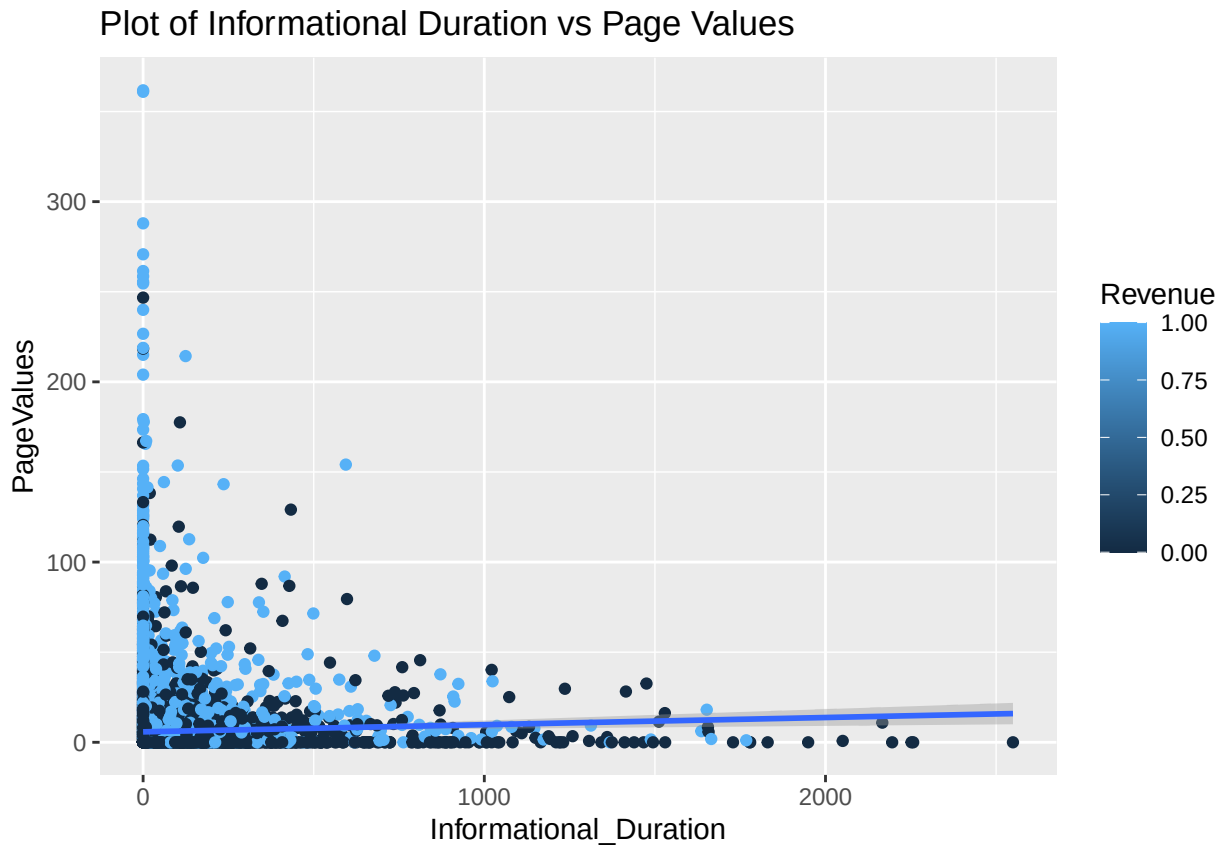
```
#we create four equally distributed bins:
#0-0.05, 0.05-0.1,0.1-0.15,0.15-0.2


df$exit_rate_disc <- infotheo::discretize(df$ExitRates,nbins = 4)
```

We can further explore the relationship between page values and browsing patterns:

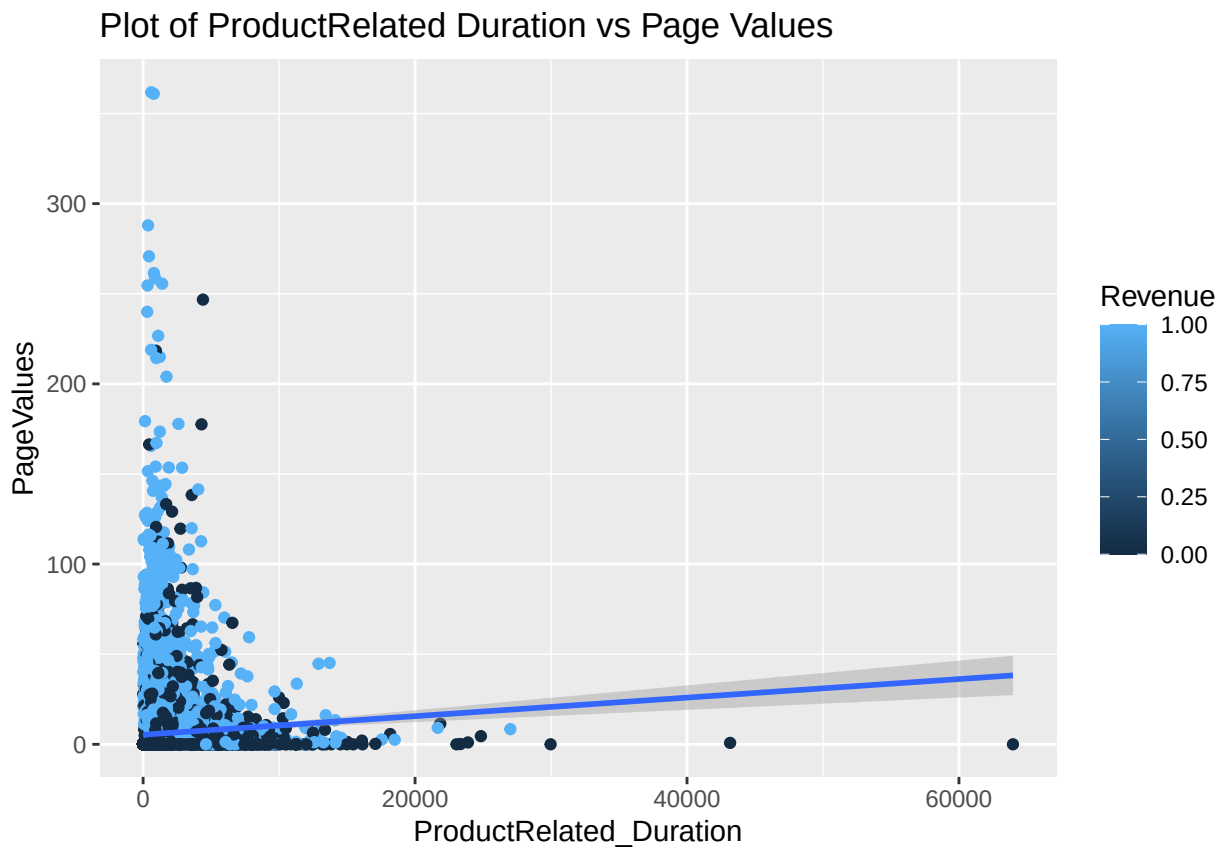
```
ggplot(df, aes(Informational_Duration,PageValues,color=Revenue))+geom_point()+geom_smooth(method = "lm")
```

```
## `geom_smooth()` using formula 'y ~ x'
```




```
ggplot(df, aes(ProductRelated_Duration,PageValues,color=Revenue))+geom_point()+geom_smooth(method = "lm
```

```
## `geom_smooth()` using formula 'y ~ x'
```



Again, there is no strong correlation between duration and page values for either type of URL. Given that ProductRelated URL browsing moves with duration, and Informational URLs have been disposed, there is no need to inspect either for unique trends.

From here, we can unfold a brief exercise in unsupervised learning between visitor types and browsing patterns for insight into what kind of URLs different demographics engage with:

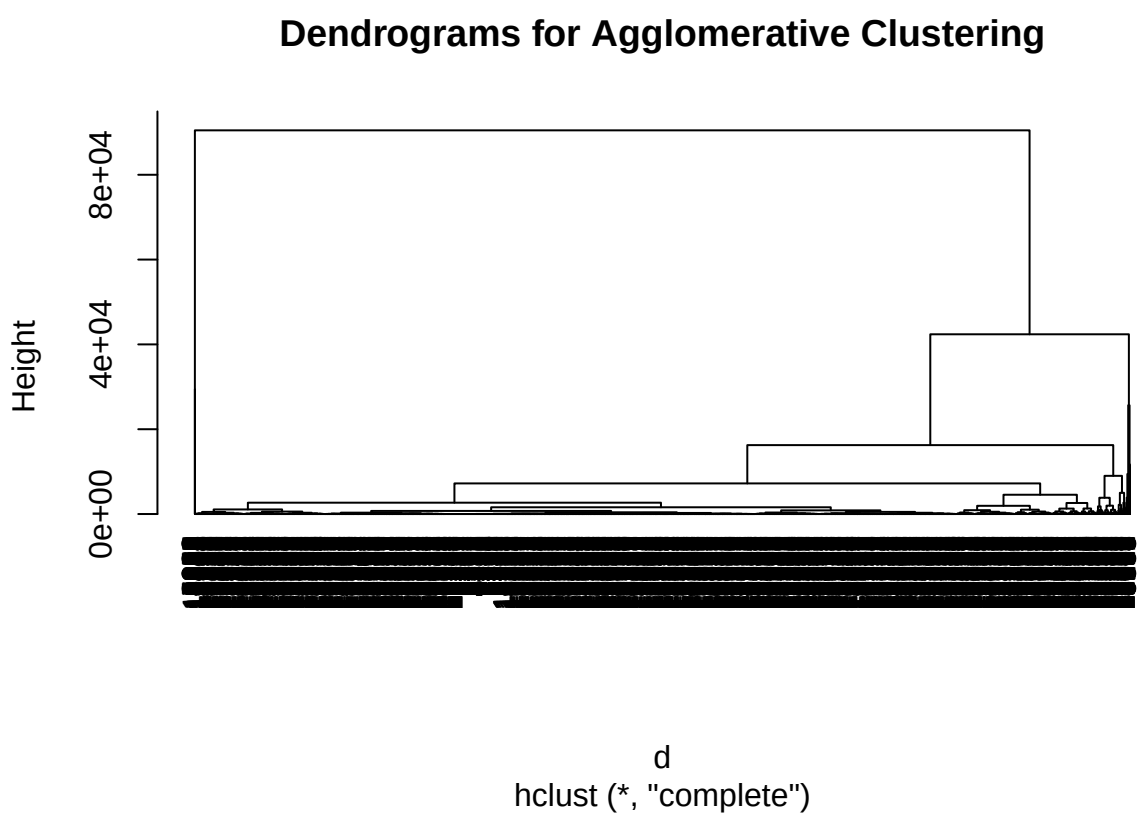
```
dfvisitor <- select(df, ProductRelated_Duration, VisitorType)

d <- dist(dfvisitor, method = "euclidean")
```

```
## Warning in dist(dfvisitor, method = "euclidean"): NAs introduced by coercion
```

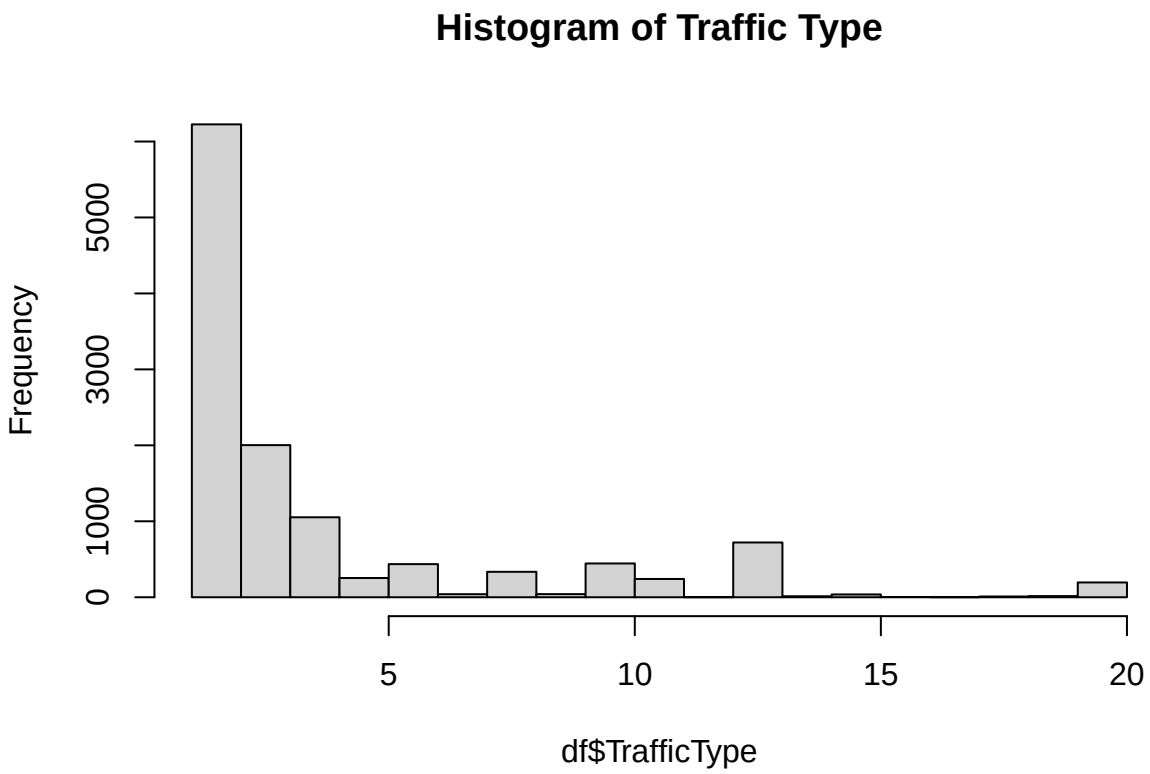
```
AHC <- hclust(d, method = "complete" )

plot(AHC, cex = 0.8, hang = -1, main = "Dendrograms for Agglomerative Clustering")
```



Moving forward, we can inspect the traffic type variable:

```
hist(df$TrafficType,main="Histogram of Traffic Type")
```



While distribution of traffic type is, at it's core, nominal,we notice that the majority of traffic comes from sources 1-5, with the mode obviously being 1. We do not have mapping for the actual source of traffic, but can still apply it in analysis. Specifically, we can see how much of each traffic type has new and returning customers.

Before we proceed, however, can eliminate the "Other" values in visitor type:

```
df <- subset(df,!(df$VisitorType=="Other"))

visitors.unique <- unique(df$VisitorType)
traffic.unique  <- unique(df$TrafficType)

for (traffic in traffic.unique){
  num_new_visitors = df$TrafficType[df$TrafficType==traffic  & df$VisitorType=="New_Visitor"]
  print(paste("traffic type:",traffic,"ratio of new visitors:",length(num_new_visitors)/length(df$Traff
}
```

```
## [1] "traffic type: 1 ratio of new visitors: 0.00300375469336671"
## [1] "traffic type: 2 ratio of new visitors: 0.076846057571965"
## [1] "traffic type: 3 ratio of new visitors: 0.0115978306216103"
## [1] "traffic type: 4 ratio of new visitors: 0.00817688777638715"
## [1] "traffic type: 5 ratio of new visitors: 0.0119315811430955"
## [1] "traffic type: 6 ratio of new visitors: 0.00350438047559449"
## [1] "traffic type: 7 ratio of new visitors: 0.000417188151856487"
## [1] "traffic type: 8 ratio of new visitors: 0.0141843971631206"
## [1] "traffic type: 9 ratio of new visitors: 0.00066750104297038"
## [1] "traffic type: 10 ratio of new visitors: 0.00183562786816854"
## [1] "traffic type: 11 ratio of new visitors: 0.00425531914893617"
```

```
## [1] "traffic type: 12 ratio of new visitors: 0"
## [1] "traffic type: 13 ratio of new visitors: 0.000500625782227785"
## [1] "traffic type: 14 ratio of new visitors: 8.34376303712975e-05"
## [1] "traffic type: 15 ratio of new visitors: 0.000166875260742595"
## [1] "traffic type: 18 ratio of new visitors: 8.34376303712975e-05"
## [1] "traffic type: 16 ratio of new visitors: 8.34376303712975e-05"
## [1] "traffic type: 17 ratio of new visitors: 0"
## [1] "traffic type: 19 ratio of new visitors: 8.34376303712975e-05"
## [1] "traffic type: 20 ratio of new visitors: 0.00141843971631206"
```

There seems to be considerable variation between traffic type and visitor type. provided that visitor type has any sort of connection with revenue, both of these would become crucial variables for predicting the possibility of success (revenue=1):

```
nv.sale   <-length(df$Revenue[df$Revenue==1  & df$VisitorType=="New_Visitor"]) / length(df$Revenue)
nv.noSale <-length(df$Revenue[df$Revenue==0  & df$VisitorType=="New_Visitor"]) / length(df$Revenue)

paste("New visitors % of Revenue = 1 ",nv.sale)
```

```
## [1] "New visitors % of Revenue = 1  0.0346266166040884"
```

```
paste("New visitors % of Revenue = 0 ",nv.noSale)
```

```
## [1] "New visitors % of Revenue = 0  0.104213600333751"
```

Without moving into extensive hypothesis testing, we can observe that browsing sessions that result in sales can be associated with a lower rate of new visitors, or conversely, with a higher rate of returning visitors. Therefore, we can provide some validity to the asertion that visitor type and traffic type both play a crucial role in predicting the probability of a sale.

Data Analysis Component

To begin our analysis, we observe that REVENUE is the dependent variable, with the aforementioned variables in the EDA serving as potential inputs. We will be conducting logistic regression as the response variable is a binary variable that is not numerical and had to be transformed from y = {TRUE,FALSE} to y={0,1}:

```
#declare necessary categorical variables
df$Month        <-as.factor(df$Month)
df$TrafficType  <- as.factor(df$TrafficType)
df$VisitorType  <- as.factor(df$VisitorType)
df$Revenue      <- as.factor(df$Revenue)

#declare training dataset and testing dataset
train <- df[1:10785,]
test  <- df[10786:11985,]

train <- subset(
  train,
  select = c(
    ProductRelated,
    ProductRelated_Duration,
    Informational_Duration,
    ExitRates,
    PageValues,
    VisitorType,
    Revenue,
    Month,
```

```
    TrafficType
  )

)



#model logistic regression
#we change to maxit=100 given the breadth of the variables
model <- glm(Revenue ~ ., family=binomial(link='logit'), data = train,maxit=100)

#print summary
summary(model)
```

```
##
## Call:
## glm(formula = Revenue ~ ., family = binomial(link = "logit"),
##     data = train, maxit = 100)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -6.2466  -0.4475  -0.3198  -0.1594   3.0888
##
## Coefficients:
##                                Estimate Std. Error z value Pr(>|z|)
## (Intercept)                  -1.820e+00  2.009e-01  -9.058  < 2e-16 ***
## ProductRelated                2.016e-03  1.194e-03   1.689 0.091276 .
## ProductRelated_Duration       5.623e-05  2.809e-05   2.002 0.045296 *
## Informational_Duration        1.139e-04  2.051e-04   0.555 0.578746
## ExitRates                    -1.552e+01  1.781e+00  -8.717  < 2e-16 ***
## PageValues                    8.481e-02  2.653e-03  31.972  < 2e-16 ***
## VisitorTypeReturning_Visitor -1.733e-01  9.853e-02  -1.758 0.078688 .
## MonthDec                     -8.628e-01  2.034e-01  -4.241 2.22e-05 ***
## MonthFeb                     -1.802e+00  6.520e-01  -2.764 0.005712 **
## MonthJul                      9.651e-02  2.204e-01   0.438 0.661499
## MonthJune                    -3.876e-01  2.828e-01  -1.371 0.170530
## MonthMar                     -6.039e-01  1.873e-01  -3.224 0.001263 **
## MonthMay                     -5.736e-01  1.727e-01  -3.321 0.000897 ***
## MonthNov                      4.629e-01  1.707e-01   2.711 0.006700 **
## MonthOct                     -1.109e-01  2.071e-01  -0.536 0.592271
## MonthSep                     -3.535e-02  2.150e-01  -0.164 0.869422
## TrafficType2                  1.563e-01  1.025e-01   1.526 0.127024
## TrafficType3                 -3.142e-01  1.326e-01  -2.369 0.017815 *
## TrafficType4                  2.396e-02  1.433e-01   0.167 0.867198
## TrafficType5                  2.293e-01  2.209e-01   1.038 0.299205
## TrafficType6                 -1.035e-01  2.106e-01  -0.492 0.622981
## TrafficType7                  3.876e-01  5.267e-01   0.736 0.461809
## TrafficType8                  5.140e-01  2.044e-01   2.515 0.011906 *
## TrafficType9                 -1.993e-02  6.796e-01  -0.029 0.976600
## TrafficType10                 2.469e-01  1.898e-01   1.301 0.193201
## TrafficType11                 3.391e-01  2.435e-01   1.393 0.163729
## TrafficType12                -1.194e+01  1.455e+03  -0.008 0.993454
## TrafficType13                -6.486e-01  2.136e-01  -3.036 0.002397 **
## TrafficType14                -4.216e-01  1.064e+00  -0.396 0.691915
```

```
## TrafficType15               -1.235e+01  2.326e+02  -0.053 0.957673
## TrafficType16                1.962e+00  1.236e+00   1.588 0.112320
## TrafficType17               -1.185e+01  1.455e+03  -0.008 0.993505
## TrafficType18               -1.251e+01  4.458e+02  -0.028 0.977618
## TrafficType19               -1.172e+00  1.479e+00  -0.792 0.428314
## TrafficType20                4.445e-01  2.832e-01   1.570 0.116458
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 9108.3  on 10784  degrees of freedom
## Residual deviance: 6004.1  on 10750  degrees of freedom
## AIC: 6074.1
##
## Number of Fisher Scoring iterations: 14
```

Initial Observations demonstrate several notable findings:

1. ProductRelated_Duration, ExitRates, and PageValues are all signifcant predictors at alpha = 0.05.
2. TrafficType islargely insignificant.
3. Month is also largely insignificant - however, the months that are (November, December, etc.) are all intuitively within proximity of notable holidays such as thanksgiving and valentine's day, so we can assume that adding in the specialDay variable will provide a useful proxy.

From here, we can proceed to the second round of modeling:

```
#reinit testing and training:

#declare training dataset and testing dataset
#train on 90% of the data
train <- df[1:10785,]
test  <- df[10786:11985,]

train <- subset(
  train,
  select = c(
    ProductRelated,
    ProductRelated_Duration,
    Informational_Duration,
    ExitRates,
    PageValues,
    VisitorType,
    Revenue,
    SpecialDay
  )
)

test <- subset(
  test,
  select = c(
    ProductRelated,
    ProductRelated_Duration,
    Informational_Duration,
    ExitRates,
    PageValues,
```

```
    VisitorType,
    Revenue,
    SpecialDay
  )
)


#model logistic regression
#we change to maxit=100 given the breadth of the variables
model <- glm(Revenue ~ ., family=binomial(link='logit'), data = train,maxit=100)

#print summary
summary(model)
```

```
##
## Call:
## glm(formula = Revenue ~ ., family = binomial(link = "logit"),
##     data = train, maxit = 100)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -6.1816  -0.4594  -0.3676  -0.1995   3.2905
##
## Coefficients:
##                                Estimate Std. Error z value Pr(>|z|)
## (Intercept)                  -1.868e+00  8.860e-02 -21.080  < 2e-16 ***
## ProductRelated                3.903e-03  1.183e-03   3.299 0.000971 ***
## ProductRelated_Duration       4.550e-05  2.871e-05   1.585 0.113011
## Informational_Duration        1.326e-04  2.041e-04   0.650 0.515861
## ExitRates                    -1.612e+01  1.705e+00  -9.457  < 2e-16 ***
## PageValues                    8.327e-02  2.582e-03  32.248  < 2e-16 ***
## VisitorTypeReturning_Visitor -3.206e-01  9.191e-02  -3.488 0.000486 ***
## SpecialDay                   -8.831e-01  2.240e-01  -3.942 8.08e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 9108.3  on 10784  degrees of freedom
## Residual deviance: 6258.0  on 10777  degrees of freedom
## AIC: 6274
##
## Number of Fisher Scoring iterations: 6
```

Despite being more succint and compact than the previous model, we observe that most of the variables, including specialDay , end up being significant. What we can interpret from these findings is as follows:

1. Browsing Duration for either category of URL used is not significant.
2. The number of ProductRelated URLs browsed is positively significant in determining the likelihood of a purchase.
3. All else constant, pages with higher exit rates are associated with a smaller probability of purchase, as well as returning visitors and higher proximity to a holiday.
4. The larger a website's page value is, the higher the association with a purchase is.

We can now observe how our model performs against the test values.

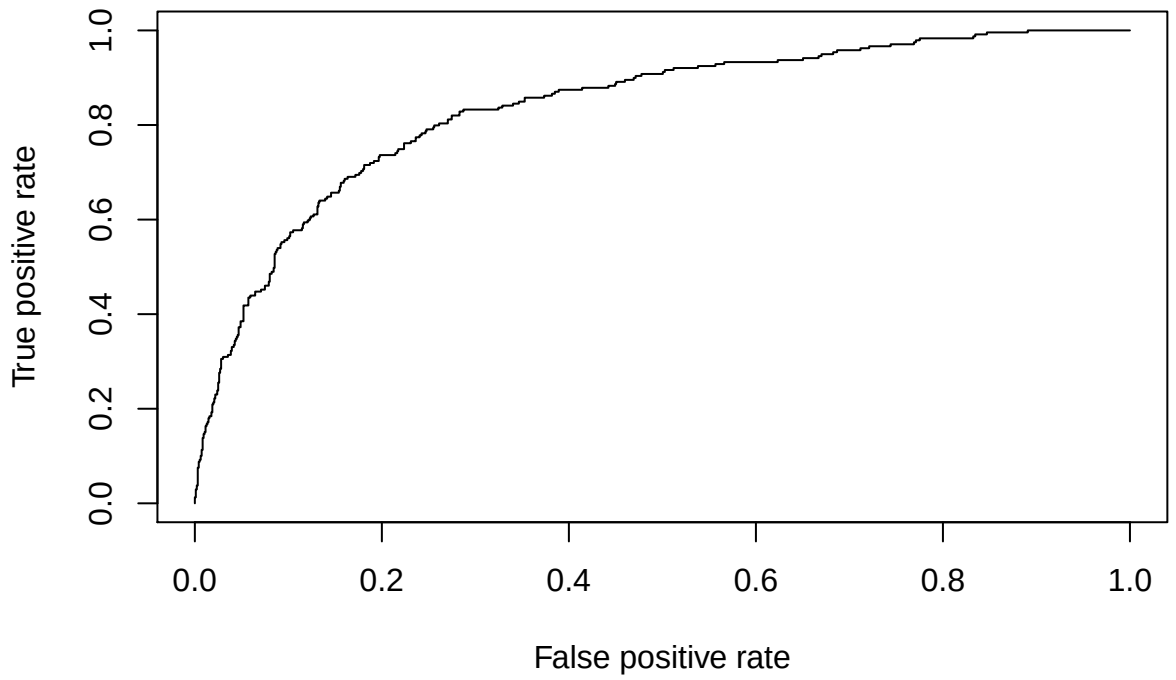
```
fitted.results <- predict(model,newdata=test,type = 'response')
fitted.results <- ifelse(fitted.results >  0.5, 1, 0)
difPred        <- fitted.results != test$Revenue

misClasError  <- mean(fitted.results != test$Revenue)
print(paste("Accuracy",1 - misClasError))
```

```
## [1] "Accuracy 0.8375"
```

The accuracy of our model in predicting the labels of the test instances is about 84%, which suggests that the model performed decently. Finally, we can plot the Receiver Operating Curve and calculate the area under the curve.

```
library(ROCR)
p <- predict(model,newdata=test,type = 'response')
pr <- prediction(p,test$Revenue)
prf <- performance(pr,measure="tpr",x.measure="fpr")
plot(prf)
```




```
auc<-performance(pr,measure="auc")
print(auc@y.values)
```

```
## [[1]]
## [1] 0.8367069
```

Given that our AUC is closer to 1 than to 0.5 at 83.6%, we appear to have a good and balanced classification model.