cryptography_ad_campaign_data_analysis

Esther_Wairimu_Kamau

1/10/2021

1.Defining the problem The research problem in this case is to find out individuals that are likely to click on a blog advert based on their characteristics which include; Age Daily Time spent on site Area of residence Internet Usage Gender Country of residence

2.Metric of Success The metric success of this project is to identify clients likely to click on the ad after performing intense data analysis(EDA).

3.Data Relevance The data provided by the client is from the performance of a previous blog advert on the same website. The columns are as follows:

- **Daily Time Spent on the site-Integer**
- **Age of the individual browsing-Integer**
- **Area of residence Internet Usage**
- **Gender of the browsing individual**
- **Country of Residence**

4.Understanding the Context

5.Experimental Design

- .Data Loading
- .Data cleaning for missing values and outliers
- .Exploratory Data Analysis
- .Conclusion-Detecting the trend in behaviour.

```
#### Importing our dataset

advertising =read.csv('http://bit.ly/IPAdvertisingData',header = TRUE, sep = ",",fileEncoding = "UTF-8-
```

```
#### exploring the top of our data
head(advertising)
```

6.Data Loading and exploring

```
##   Daily.Time.Spent.on.Site Age Area.Income Daily.Internet.Usage
## 1                    68.95  35    61833.90               256.09
## 2                    80.23  31    68441.85               193.77
## 3                    69.47  26    59785.94               236.50
## 4                    74.15  29    54806.18               245.89
## 5                    68.37  35    73889.99               225.58
## 6                    59.99  23    59761.56               226.74
##                          Ad.Topic.Line           City Male    Country
## 1    Cloned 5thgeneration orchestration    Wrightburgh    0    Tunisia
## 2    Monitored national standardization      West Jodi    1      Nauru
## 3      Organic bottom-line service-desk       Davidton    0 San Marino
## 4 Triple-buffered reciprocal time-frame West Terrifurt    1      Italy
## 5         Robust logistical utilization   South Manuel    0    Iceland
## 6       Sharable client-driven software      Jamieberg    1     Norway
##             Timestamp Clicked.on.Ad
## 1 2016-03-27 00:53:11             0
## 2 2016-04-04 01:39:02             0
## 3 2016-03-13 20:35:42             0
## 4 2016-01-10 02:31:19             0
## 5 2016-06-03 03:36:18             0
## 6 2016-05-19 14:30:17             0
```

```r
#### exploring the bottom of our data
tail(advertising)
```

```
##      Daily.Time.Spent.on.Site Age Area.Income Daily.Internet.Usage
## 995                     43.70  28    63126.96               173.01
## 996                     72.97  30    71384.57               208.58
## 997                     51.30  45    67782.17               134.42
## 998                     51.63  51    42415.72               120.37
## 999                     55.55  19    41920.79               187.95
## 1000                    45.01  26    29875.80               178.35
##                               Ad.Topic.Line          City Male
## 995         Front-line bifurcated ability  Nicholasland    0
## 996         Fundamental modular algorithm     Duffystad    1
## 997       Grass-roots cohesive monitoring   New Darlene    1
## 998          Expanded intangible solution South Jessica    1
## 999  Proactive bandwidth-monitored policy   West Steven    0
## 1000      Virtual 5thgeneration emulation   Ronniemouth    0
##                     Country           Timestamp Clicked.on.Ad
## 995                 Mayotte 2016-04-04 03:57:48             1
## 996                 Lebanon 2016-02-11 21:49:00             1
## 997  Bosnia and Herzegovina 2016-04-22 02:07:01             1
## 998                Mongolia 2016-02-01 17:24:57             1
## 999               Guatemala 2016-03-24 02:35:54             0
## 1000                 Brazil 2016-06-03 21:43:21             1
```

```r
#### Checking the class of the object "advertising"

class(advertising)
```

```
## [1] "data.frame"
```

```r
#### Checking the dimension of our dataset
dim(advertising)
```

```
## [1] 1000   10
```

```r
#### Checking the structure of our data frame
str(advertising)
```

```
## 'data.frame':    1000 obs. of  10 variables:
##  $ Daily.Time.Spent.on.Site: num  69 80.2 69.5 74.2 68.4 ...
##  $ Age                     : int  35 31 26 29 35 23 33 48 30 20 ...
##  $ Area.Income             : num  61834 68442 59786 54806 73890 ...
##  $ Daily.Internet.Usage    : num  256 194 236 246 226 ...
##  $ Ad.Topic.Line           : chr  "Cloned 5thgeneration orchestration" "Monitored national standardi
##  $ City                    : chr  "Wrightburgh" "West Jodi" "Davidton" "West Terrifurt" ...
##  $ Male                    : int  0 1 0 1 0 1 0 1 1 1 ...
##  $ Country                 : chr  "Tunisia" "Nauru" "San Marino" "Italy" ...
##  $ Timestamp               : chr  "2016-03-27 00:53:11" "2016-04-04 01:39:02" "2016-03-13 20:35:42"
##  $ Clicked.on.Ad           : int  0 0 0 0 0 0 0 1 0 0 ...
```

We can observe that we have a mix of datatypes from intergers to strings

```r
#### Getting the names of the columns we will be working with
colnames(advertising)
```

```
##  [1] "Daily.Time.Spent.on.Site" "Age"
##  [3] "Area.Income"              "Daily.Internet.Usage"
##  [5] "Ad.Topic.Line"            "City"
##  [7] "Male"                     "Country"
##  [9] "Timestamp"                "Clicked.on.Ad"
```

we can observe that our column names can all be changed to lowercase

```r
#####  Checking for duplicated values in our data set

anyDuplicated(advertising)
```

7.Data cleaning

```
## [1] 0
```

```r
##### Checking if our dataset has any missing values
sum(is.na(advertising))
```

```
## [1] 0
```

```r
### checking for missing values using case.complete function(just to confirm)
# The function complete.cases() returns a logical vector indicating which cases are complete.
# list rows of data that have missing values

advertising[!complete.cases(advertising),]
```

```
##  [1] Daily.Time.Spent.on.Site Age                      Area.Income
##  [4] Daily.Internet.Usage     Ad.Topic.Line            City
##  [7] Male                     Country                  Timestamp
## [10] Clicked.on.Ad
## <0 rows> (or 0-length row.names)
```

```r
### we rename the column names since they are too long
#we will be Using function rename
advertising=setnames(advertising, tolower(names(advertising[1:10])))

library(reshape)
```

```
##
## Attaching package: 'reshape'

## The following object is masked from 'package:tigerstats':
##
##     tips

## The following object is masked from 'package:Matrix':
##
##     expand

## The following object is masked from 'package:dplyr':
##
##     rename

## The following objects are masked from 'package:tidyr':
##
##     expand, smiths

## The following object is masked from 'package:data.table':
##
##     melt
```

```r
advertising <-  rename(advertising, c(daily.time.spent.on.site="timespent"))
advertising <- rename(advertising, c(ad.topic.line="topic"))
advertising <- rename(advertising, c(daily.internet.usage="usage"))
advertising <- rename(advertising, c(clicked.on.ad ="clicked"))
advertising <- rename(advertising, c(timestamp="timestamp"))
advertising <- rename(advertising, c(area.income="income"))
advertising <- rename(advertising, c(male="gender"))
```

```r
### check if columns have been changed

head(advertising,n=3)
```

```
##   timespent age   income  usage                              topic        city
## 1     68.95  35 61833.90 256.09 Cloned 5thgeneration orchestration Wrightburgh
## 2     80.23  31 68441.85 193.77 Monitored national standardization   West Jodi
## 3     69.47  26 59785.94 236.50   Organic bottom-line service-desk    Davidton
##   gender    country           timestamp clicked
## 1      0    Tunisia 2016-03-27 00:53:11       0
## 2      1      Nauru 2016-04-04 01:39:02       0
## 3      0 San Marino 2016-03-13 20:35:42       0
```

```r
### checking for outliers, we only need the numerical columns
#first we get the numerical columns

nums <- unlist(lapply(advertising, is.numeric))

numcols <- advertising[ ,nums]

head(numcols,n=3)
```

```
##   timespent age   income  usage gender clicked
## 1     68.95  35 61833.90 256.09      0       0
## 2     80.23  31 68441.85 193.77      1       0
## 3     69.47  26 59785.94 236.50      0       0
```

```r
### checking for unique values
uniqueitems <- unique(advertising)

head(uniqueitems,n=3)
```

```
##   timespent age   income  usage                              topic        city
## 1     68.95  35 61833.90 256.09 Cloned 5thgeneration orchestration Wrightburgh
## 2     80.23  31 68441.85 193.77 Monitored national standardization   West Jodi
## 3     69.47  26 59785.94 236.50   Organic bottom-line service-desk    Davidton
##   gender    country           timestamp clicked
## 1      0    Tunisia 2016-03-27 00:53:11       0
## 2      1      Nauru 2016-04-04 01:39:02       0
## 3      0 San Marino 2016-03-13 20:35:42       0
```

```r
####  feature enginering the time/date
#we separate months,year and day each on its own
#library lubridate makes it easier for us to deal with dates
#install packages first then libraries


library(tidyr)
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

```
## The following object is masked from 'package:reshape':
##
##     stamp

## The following objects are masked from 'package:data.table':
##
##     hour, isoweek, mday, minute, month, quarter, second, wday, week,
##     yday, year

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

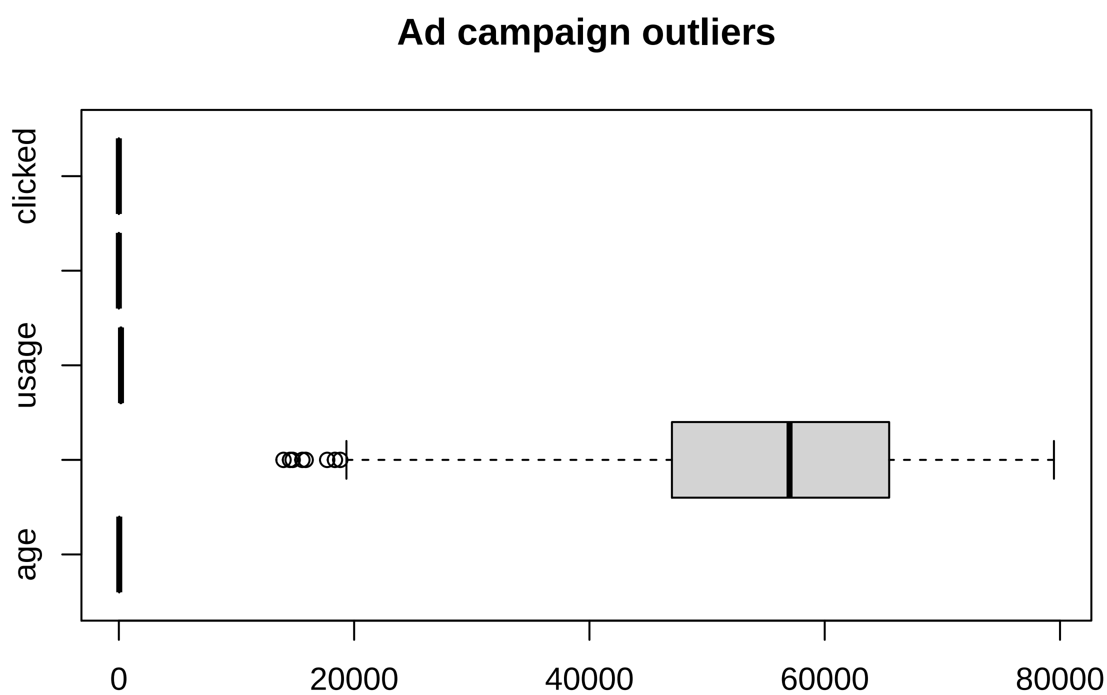
```
advertising <- separate(advertising, timestamp, c("Year", "Month", "Day"))
```

```
## Warning: Expected 3 pieces. Additional pieces discarded in 1000 rows [1, 2, 3,
## 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, ...].
```

```
head(advertising,n=3)
```

```
##   timespent age   income  usage                               topic        city
## 1     68.95  35 61833.90 256.09 Cloned 5thgeneration orchestration Wrightburgh
## 2     80.23  31 68441.85 193.77 Monitored national standardization   West Jodi
## 3     69.47  26 59785.94 236.50   Organic bottom-line service-desk    Davidton
##   gender    country Year Month Day clicked
## 1      0    Tunisia 2016    03  27       0
## 2      1      Nauru 2016    04  04       0
## 3      0 San Marino 2016    03  13       0
```

```
#### Plotting the boxplot to visualize the outliers in the dataset
boxplot(numcols[,-1], horizontal=TRUE, main="Ad campaign outliers")
```



We observe that only income has any outliers,it wont affect the analysis so we countinue with the EDA.

8.Exploratory Data Analysis

Univariate Analysis

```
#For ease in analysis,we convert the data into a tibble REASONS why we use tibble dataframes
#never converts string as factor
#never changes the names of variables
#never create row names
library(tidyverse)

adv<-as_tibble(advertising)

head(adv,n=3)
```

```
## # A tibble: 3 x 12
##   timespent   age income usage topic city  gender country Year  Month Day
##       <dbl> <int>  <dbl> <dbl> <chr> <chr>  <int> <chr>   <chr> <chr> <chr>
## 1      69.0    35 61834.  256. Clon~ Wrig~      0 Tunisia 2016  03    27
## 2      80.2    31 68442.  194. Moni~ West~      1 Nauru   2016  04    04
## 3      69.5    26 59786.  236. Orga~ Davi~      0 San Ma~ 2016  03    13
## # ... with 1 more variable: clicked <int>
```

Extracting Numerical tibble columns

```r
#we define our tibble numerical dataframe

library(dplyr)

numt=adv %>% select_if(is.numeric)

head(numt,n=3)
```

```
## # A tibble: 3 x 6
##   timespent   age income usage gender clicked
##       <dbl> <int>  <dbl> <dbl>  <int>   <int>
## 1      69.0    35 61834.  256.      0       0
## 2      80.2    31 68442.  194.      1       0
## 3      69.5    26 59786.  236.      0       0
```

Extracting categorical tibble columns

```r
Categoryt=adv %>% select_if(is.character)

head(Categoryt,n=3)
```

```
## # A tibble: 3 x 6
##   topic                              city        country    Year  Month Day
##   <chr>                              <chr>       <chr>      <chr> <chr> <chr>
## 1 Cloned 5thgeneration orchestration Wrightburgh Tunisia    2016  03    27
## 2 Monitored national standardization West Jodi   Nauru      2016  04    04
## 3 Organic bottom-line service-desk   Davidton    San Marino 2016  03    13
```

We first find the descriptive statistics of the numerical columns

```r
summary(numt)
```

```
##    timespent          age            income           usage
##  Min.   :32.60   Min.   :19.00   Min.   :13996   Min.   :104.8
##  1st Qu.:51.36   1st Qu.:29.00   1st Qu.:47032   1st Qu.:138.8
##  Median :68.22   Median :35.00   Median :57012   Median :183.1
##  Mean   :65.00   Mean   :36.01   Mean   :55000   Mean   :180.0
##  3rd Qu.:78.55   3rd Qu.:42.00   3rd Qu.:65471   3rd Qu.:218.8
##  Max.   :91.43   Max.   :61.00   Max.   :79485   Max.   :270.0
##      gender         clicked
##  Min.   :0.000   Min.   :0.0
##  1st Qu.:0.000   1st Qu.:0.0
##  Median :0.000   Median :0.5
##  Mean   :0.481   Mean   :0.5
##  3rd Qu.:1.000   3rd Qu.:1.0
##  Max.   :1.000   Max.   :1.0
```

- We observe that mean of the age of individuals in our dataset is 36 with the oldest being 61.
- most individuals have an income of 55000 and with the lowest being 13996.
- the time spent online is mostly 1hr 5mins with the highest being 1hr 31mins.
- the cost of being online on hourly(65mins) rate is 180
- the mean of the page clicks is 0.5 meaning the clicks are equal to 'no clicks'

Plotting Histograms for numerical columns

```
#par(mfrow = c(2, 2))
#hist(numt$timespent)
#hist(numt$age)
#hist(numt$income)
#hist(numt$usage)
#hist(numt$gender)
#hist(numt$clicked)
```

numerical columns mode

```
#The mode is the value that appears most frequently in a data set
#Finding the mode of all numerical columns
#we start with age

v<-adv%>% pull(age)
getmode <- function(v) {
   uniqv <- unique(v)
   uniqv[which.max(tabulate(match(v, uniqv)))]
}
Age.Mode<-getmode(adv$age)
Age.Mode
```

```
## [1] 31
```

The age that appears most is 31years so most individuals who click on the page are in this age group

```
#we start with age

v2<-adv%>% pull(income)
getmode <- function(v) {
   uniqv <- unique(v)
   uniqv[which.max(tabulate(match(v, uniqv)))]
}
income.Mode<-getmode(adv$income)
income.Mode
```

```
## [1] 61833.9
```

We see that most individuals in dataset's income is range of 60000 and above

```r
#we start with age

v3<-adv%>% pull(timespent)
getmode <- function(v) {
   uniqv <- unique(v)
   uniqv[which.max(tabulate(match(v, uniqv)))]
}
time.Mode<-getmode(adv$timespent)
time.Mode
```

```
## [1] 62.26
```

We observe that most time spent that appears most times is 62 which means that our univariate plots were correct

```r
#we start with age

v5<-adv%>% pull(usage)
getmode <- function(v) {
   uniqv <- unique(v)
   uniqv[which.max(tabulate(match(v, uniqv)))]
}
usage.Mode<-getmode(adv$usage)
usage.Mode
```

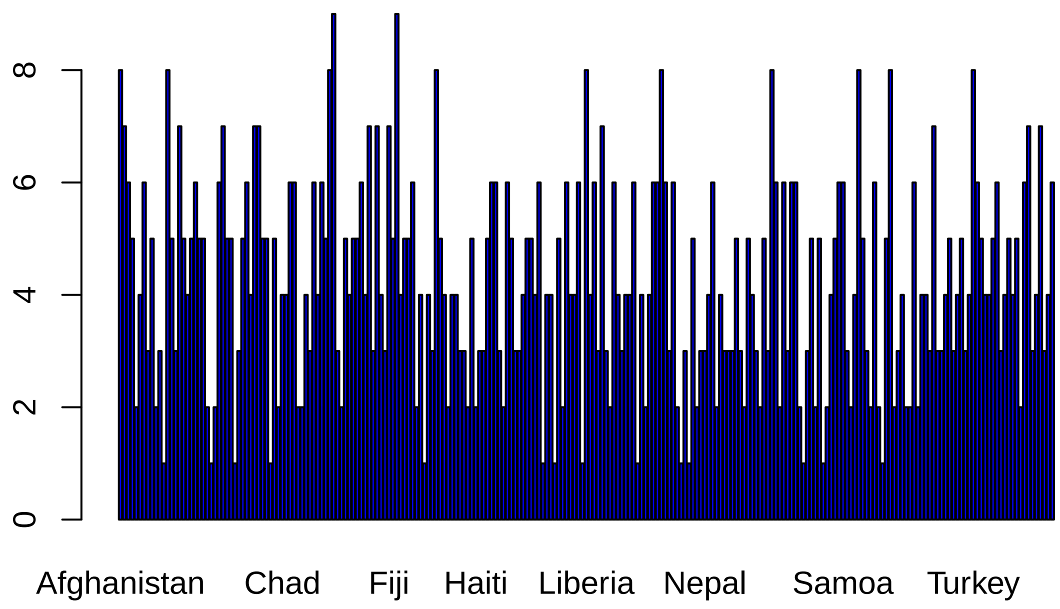
```
## [1] 167.22
```

most guys use 167 on every time they spend online.Which is almost same with the univariate plots.

```r
### Plot frequency plots for categorical columns
#we start with country column

country <- Categoryt$country
Country_frequency<- table(country)
s<-desc(Country_frequency)
head(s,n=2)
```

```
## country
## Afghanistan     Albania
##          -8          -7
```

```r
barplot(Country_frequency,col="Blue")
```

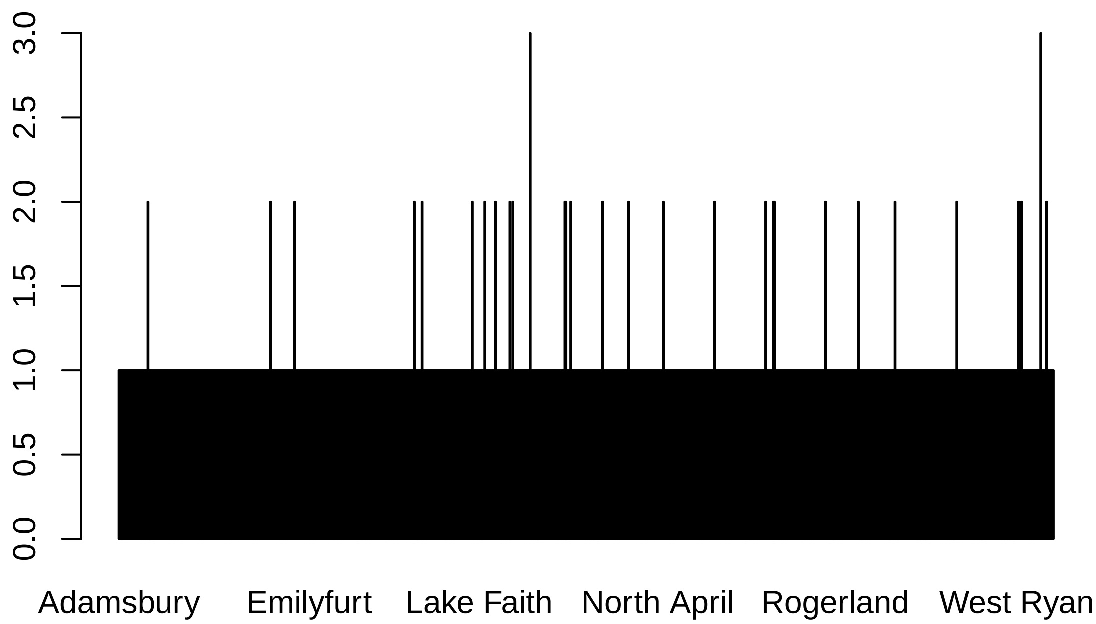


we observe that country that has most customers is Afghanistan followed albania as we can see in first console the second plot confirms it.

```
#secondly we tackle the city column
f2 <- Categoryt$city
f2_frequency<- table(f2)
g<-desc(f2_frequency)
head(g,n=3)
```

```
## f2
## Adamsbury  Adamside Adamsstad
##        -1        -1        -1
```

```
barplot(f2_frequency,col="Red")
```

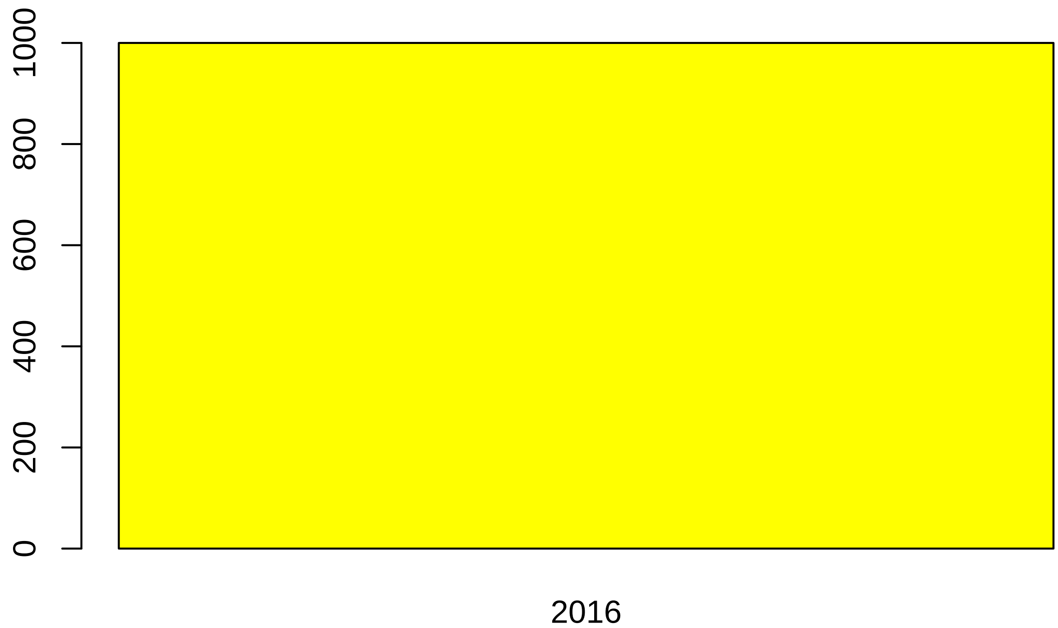


we observe that williamsport city appears thrice more than most city column.It has too many unique values.

```
#
f3 <- Categoryt$Year
f3_frequency<- table(f3)
desc(f3_frequency)
```

```
## f3
##  2016
## -1000
```

```
barplot(f3_frequency,col="Yellow")
```

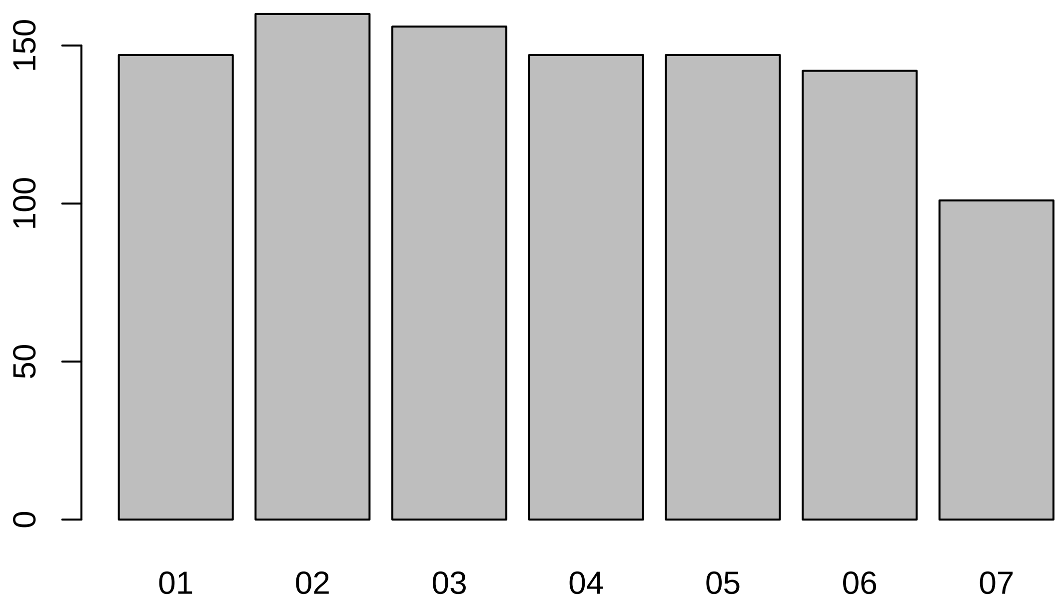


all observations were taken from 2016

```
f4 <- Categoryt$Month
f4_frequency<- table(f4)
desc(f4_frequency)
```

```
## f4
##   01   02   03   04   05   06   07
## -147 -160 -156 -147 -147 -142 -101
```

```
barplot(f4_frequency,col="Grey")
```

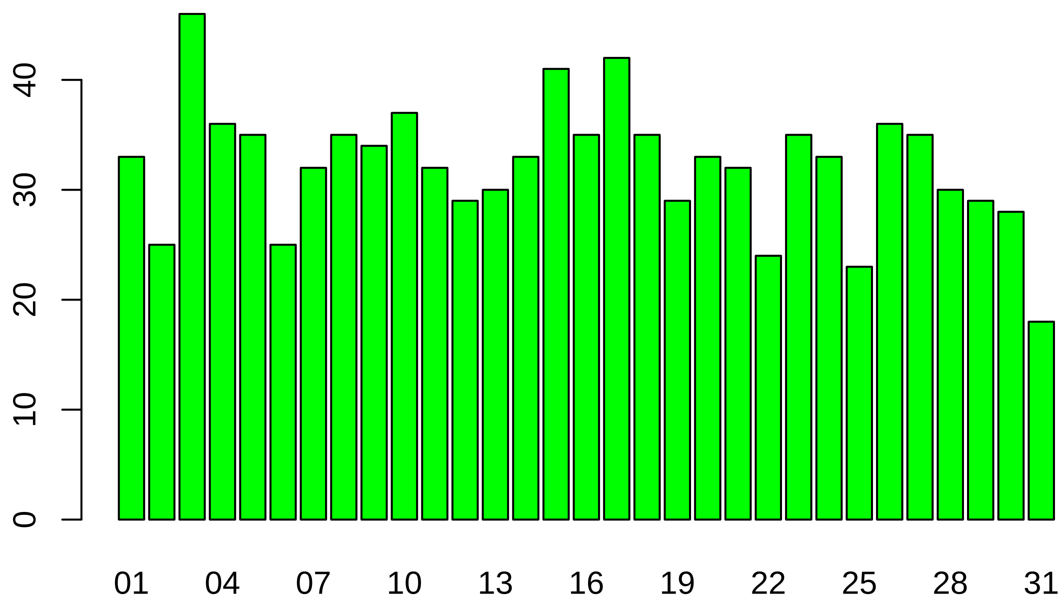


we observe that the month with highest traffic is February followed by march with january,April and may being the same.Also there is consistent traffic month on month.

```
f5 <- Categoryt$Day
f5_frequency<- table(f5)
head.matrix(f5_frequency)
```

```
## f5
## 01 02 03 04 05 06
## 33 25 46 36 35 25
```

```
barplot(f5_frequency,col="green")
```

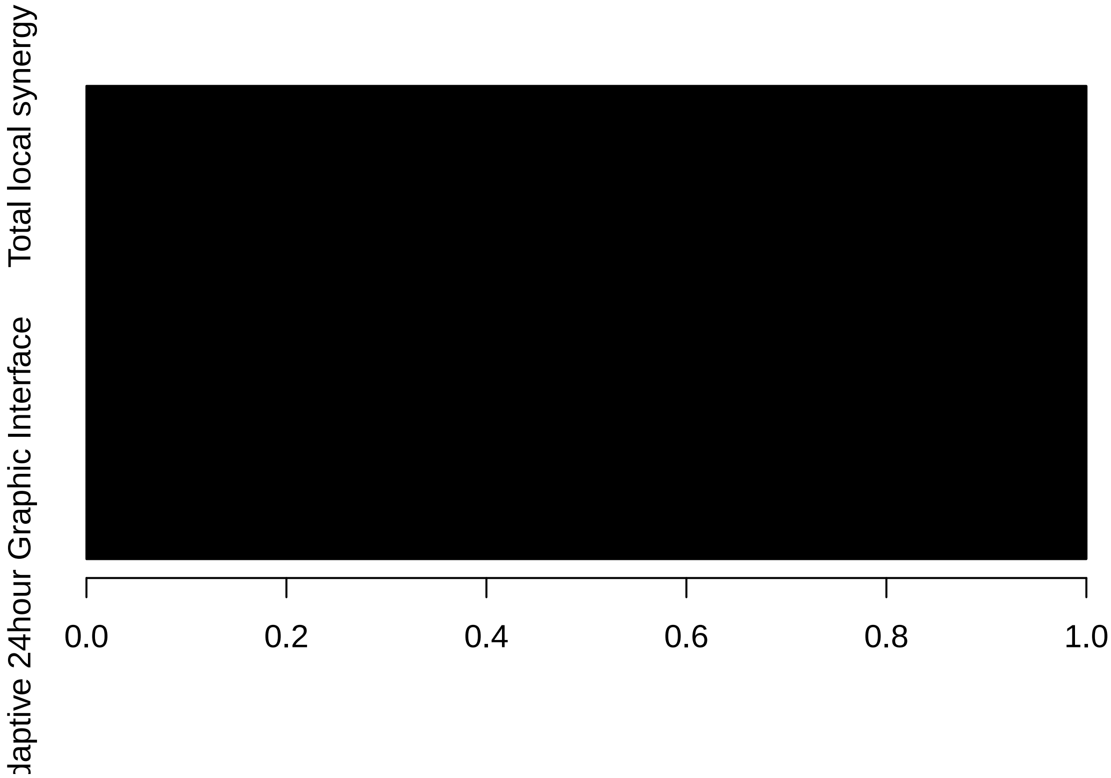


we observe that no specific time of the month is there extra high traffic or extra low traffic is almost same all days.But on 31st we can notice is weirdly low.

```
f4 <- Categoryt$topic
f4_frequency<- table(f4)
head.matrix(f4_frequency)
```

```
## f4
##       Adaptive 24hour Graphic Interface          Adaptive asynchronous attitude
##                                       1                                       1
##   Adaptive context-sensitive application Adaptive contextually-based methodology
##                                       1                                       1
##     Adaptive demand-driven knowledgebase             Adaptive uniform capability
##                                       1                                       1
```

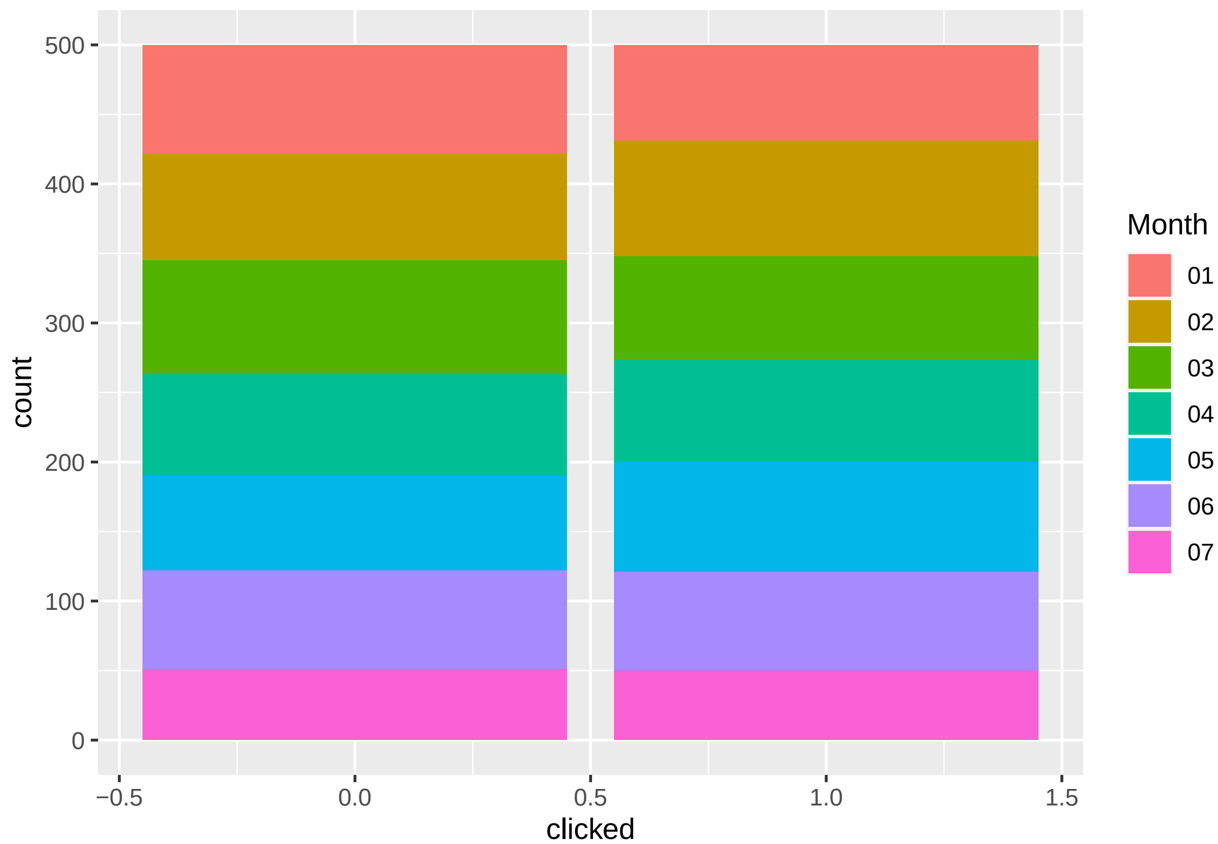
```
barplot(f4_frequency,col="Blue",horiz=TRUE)
```



This means all topics have the same distribution they are too unique and none has counts than the other.

9.Bivariate Analysis

```r
#clicks of individuals in our dataset month on month

ggplot(adv, aes(x = clicked,fill = Month)) + geom_bar(position = "stack")
```

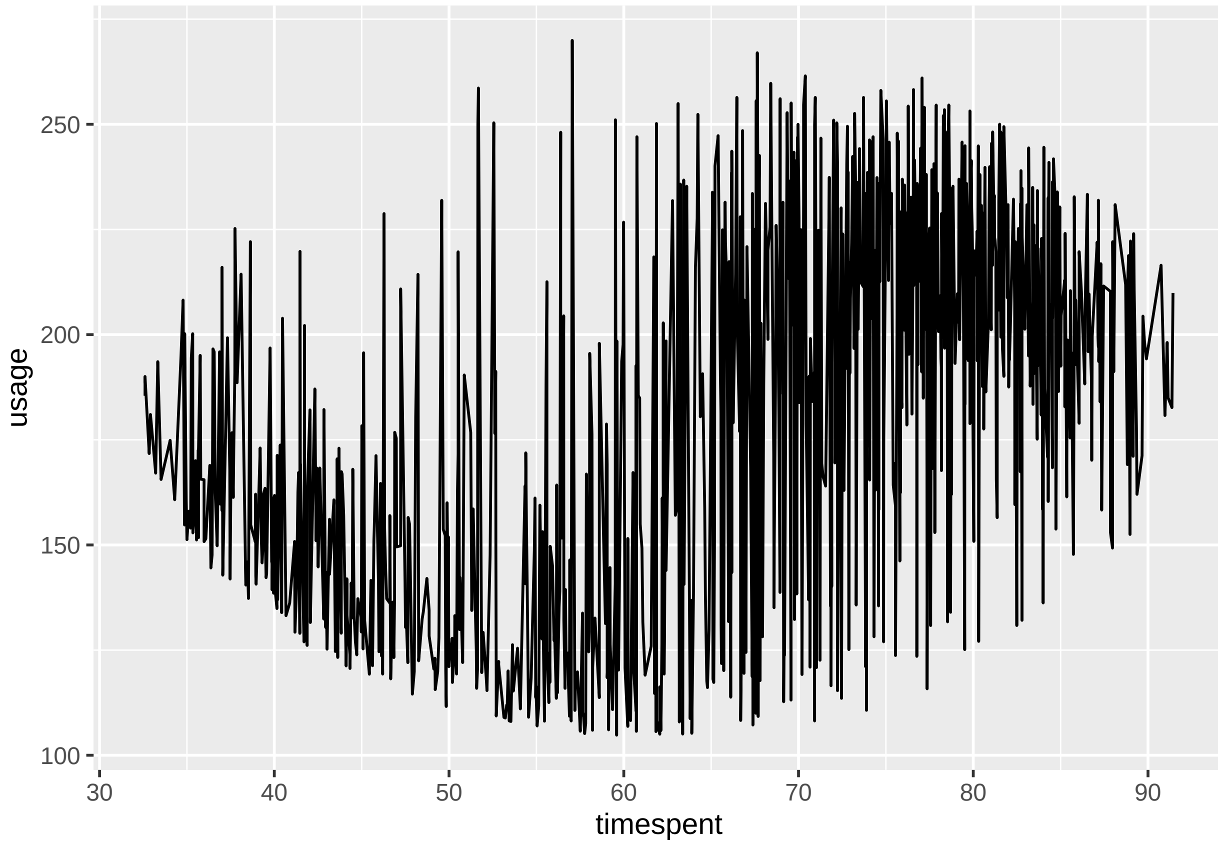


we observe that the distrbution of individuals who clicked and the ones who didn't is the same monthly.

```
#time spent online versus the income of individuals
geom_line()
```

```
## geom_line: na.rm = FALSE, orientation = NA
## stat_identity: na.rm = FALSE
## position_identity
```

```
ggplot(data =adv,aes(x=timespent,y=usage))+
geom_line()
```

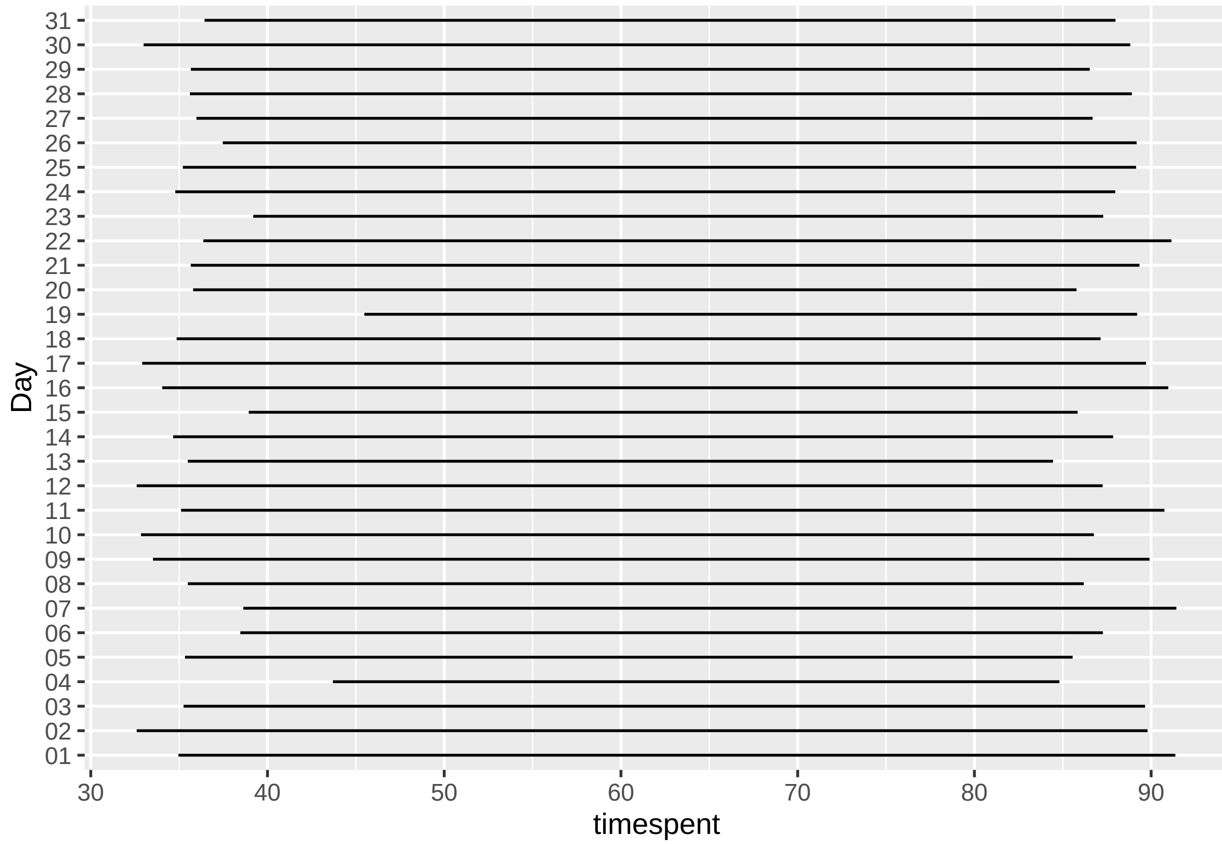


We observe that the more time on spends online the more the usage as we can see above

```
#time spent online versus the income of individuals

geom_line()
```

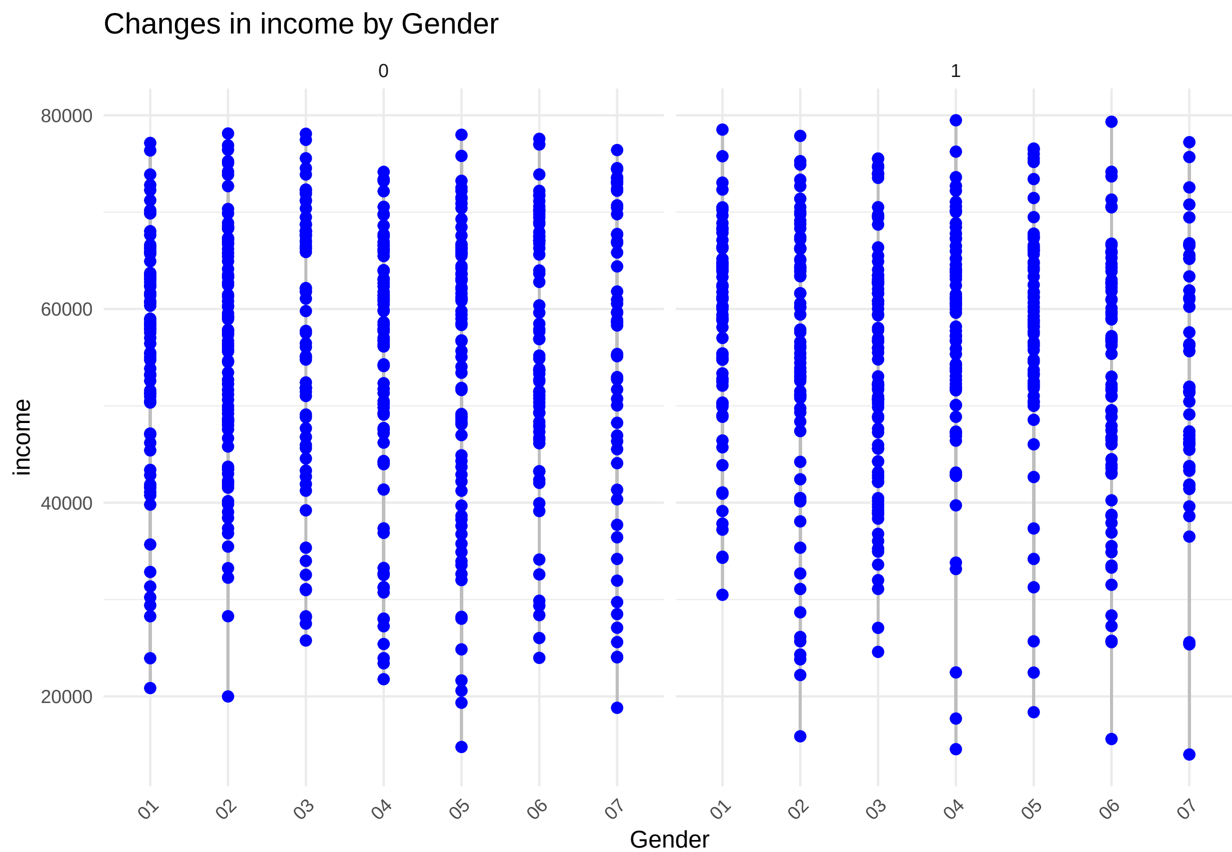
```
## geom_line: na.rm = FALSE, orientation = NA
## stat_identity: na.rm = FALSE
## position_identity
```

```
ggplot(data =adv,aes(x=timespent,y=Day))+geom_line()
```



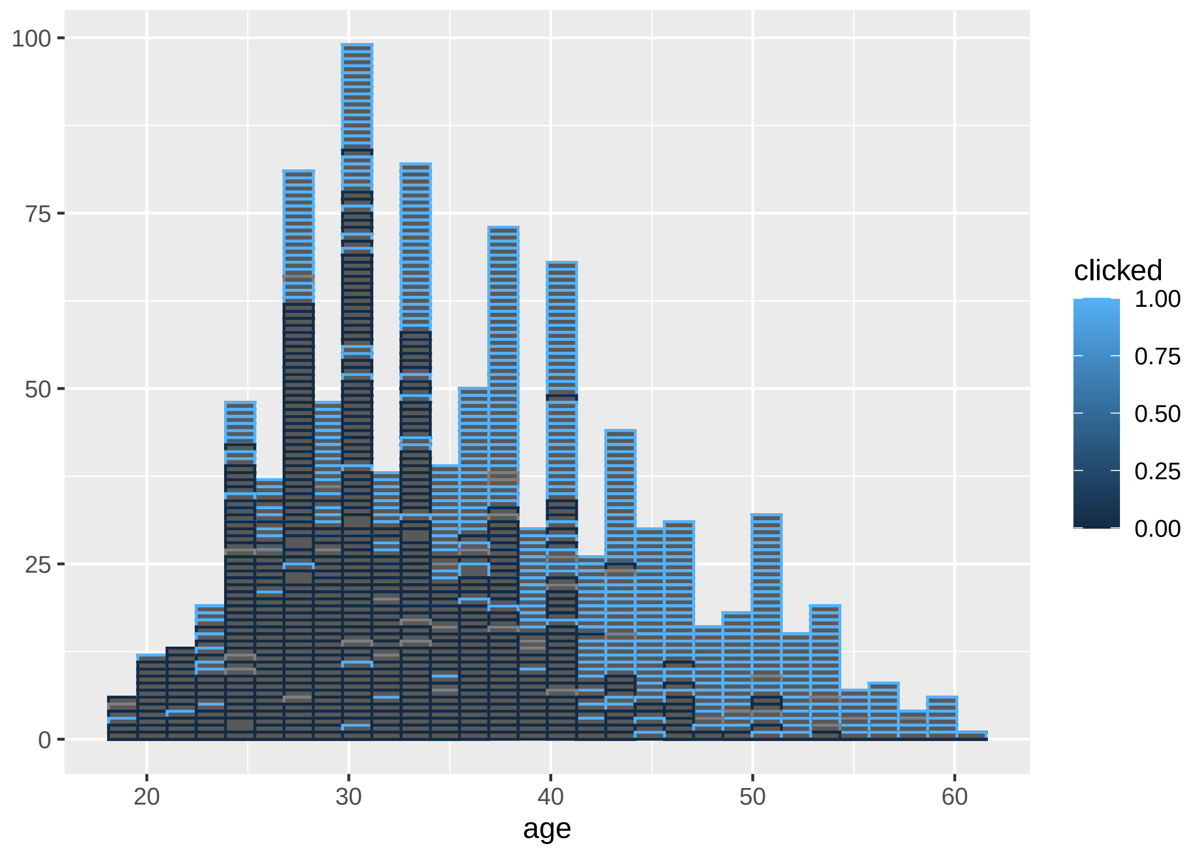
we observe that on a daily basis people spend time online on the page

```
# plot income changes by month, for each Gender
ggplot(adv, aes(x=Month, y = income)) +
geom_line(color="grey") +
geom_point(color="blue") +
facet_wrap(~gender) +
theme_minimal(base_size = 9) +
theme(axis.text.x = element_text(angle = 45,
                                   hjust = 1)) +
  labs(title = "Changes in income by Gender",
       x = "Gender",
       y = "income")
```



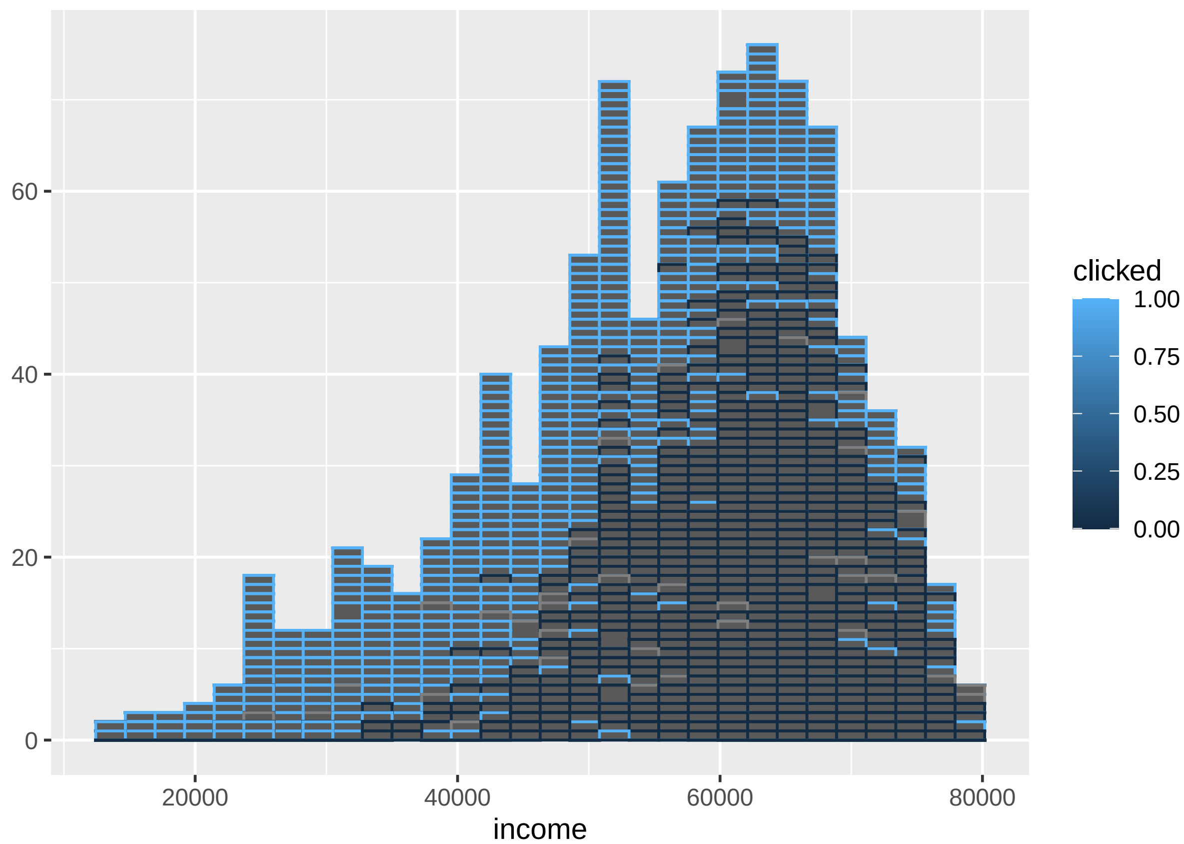
we note that gender o has fewer individuals who earn below 20000 than gender 1 we also note that gender 0 and gender 1 almost have the same salaries over the months * in may and december there is partial disparity when it comes to the incomes gender o has more income earning individuals in those months than gender 1

```
# We check on the timespent versus the age and the click
qplot(x=age,data=adv,group=timespent,colour=clicked,bins=30)
```



we can observe that individuals as age decreases the clicks decrease but time spent in some ages like 30 increase alot.But from 38 to around 40 the time spent decreases but the clicks increase.

```
qplot(x=income,data=adv,group=timespent,colour=clicked,bins=30)
```

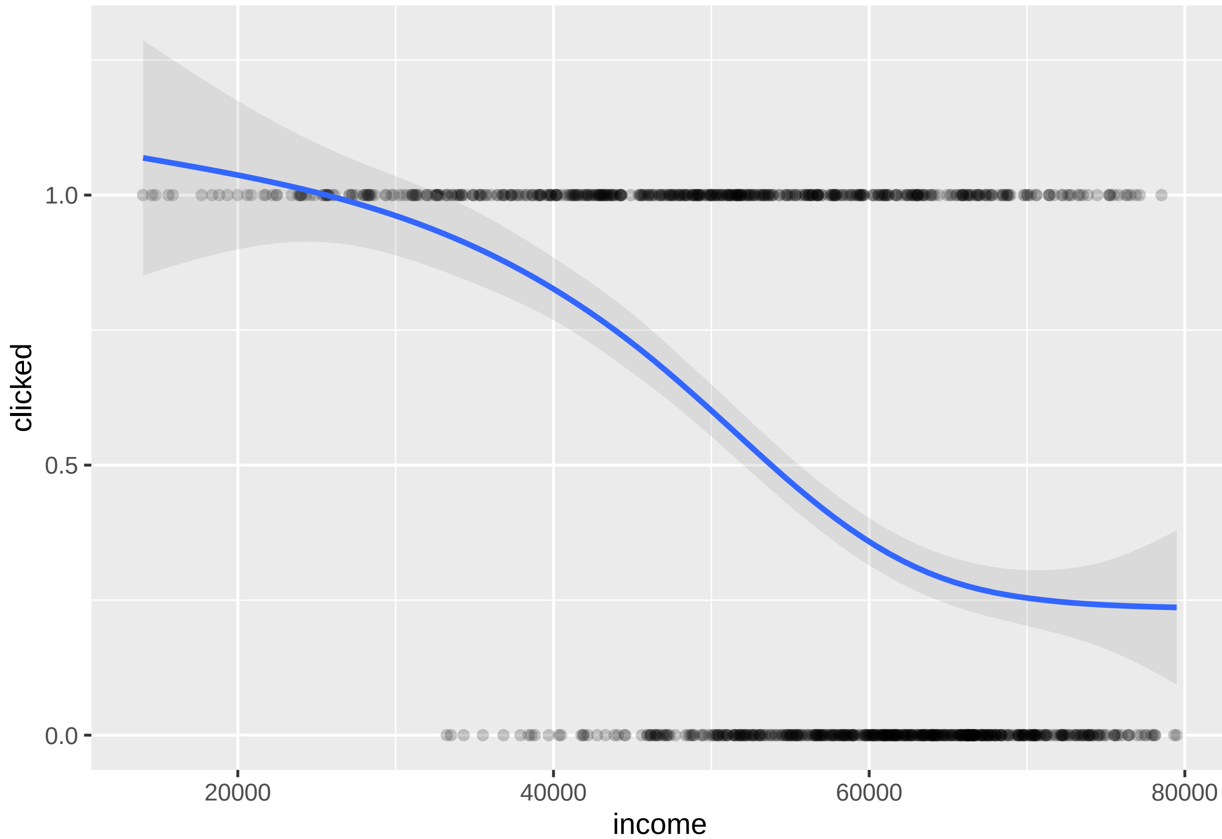


we can observe that the plot is skewed to right meaning that as income increases the more the more the time spent which also increases click

```r
# Plot to show realtionship between clicked and income

qplot(income,
      clicked,
      data = adv,
      geom = c("point", "smooth"),
      alpha = I(1 / 5))
```

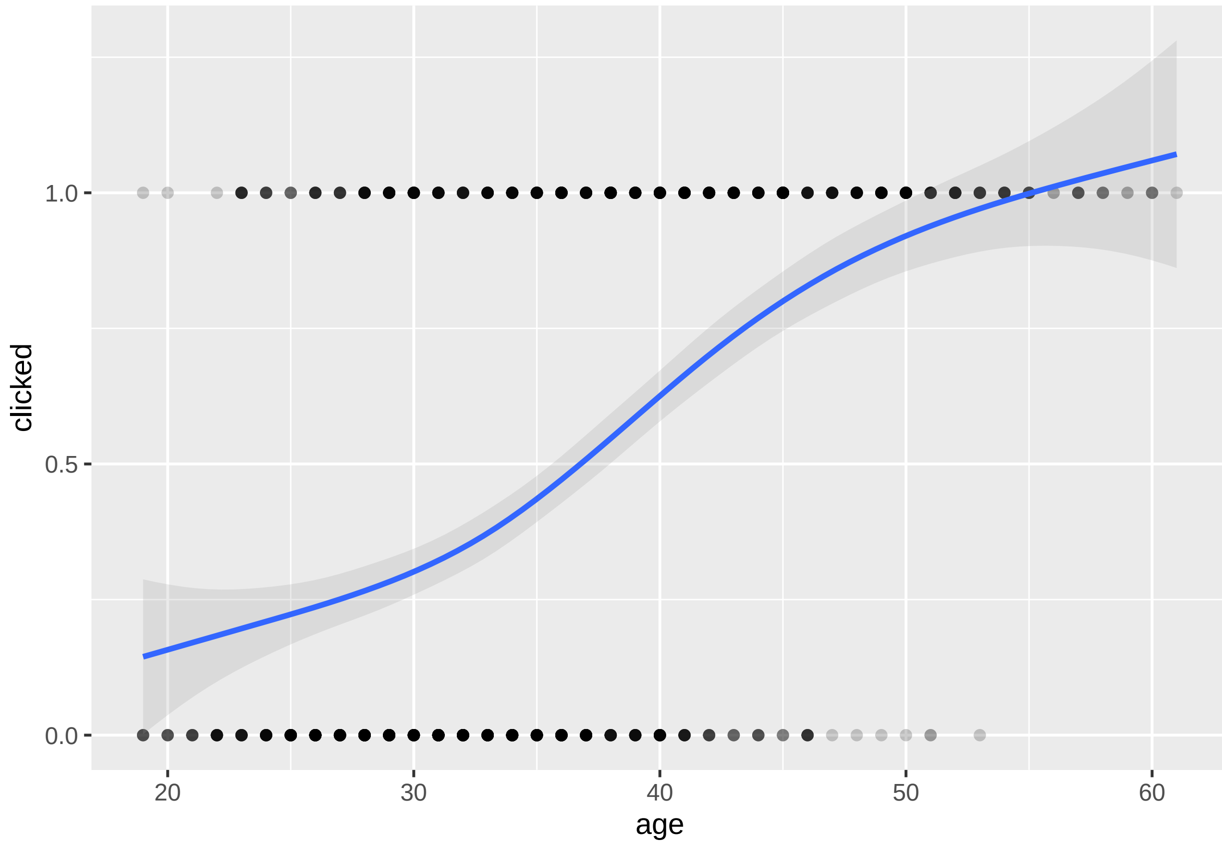
relationships between the target variable(clicked) and features

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



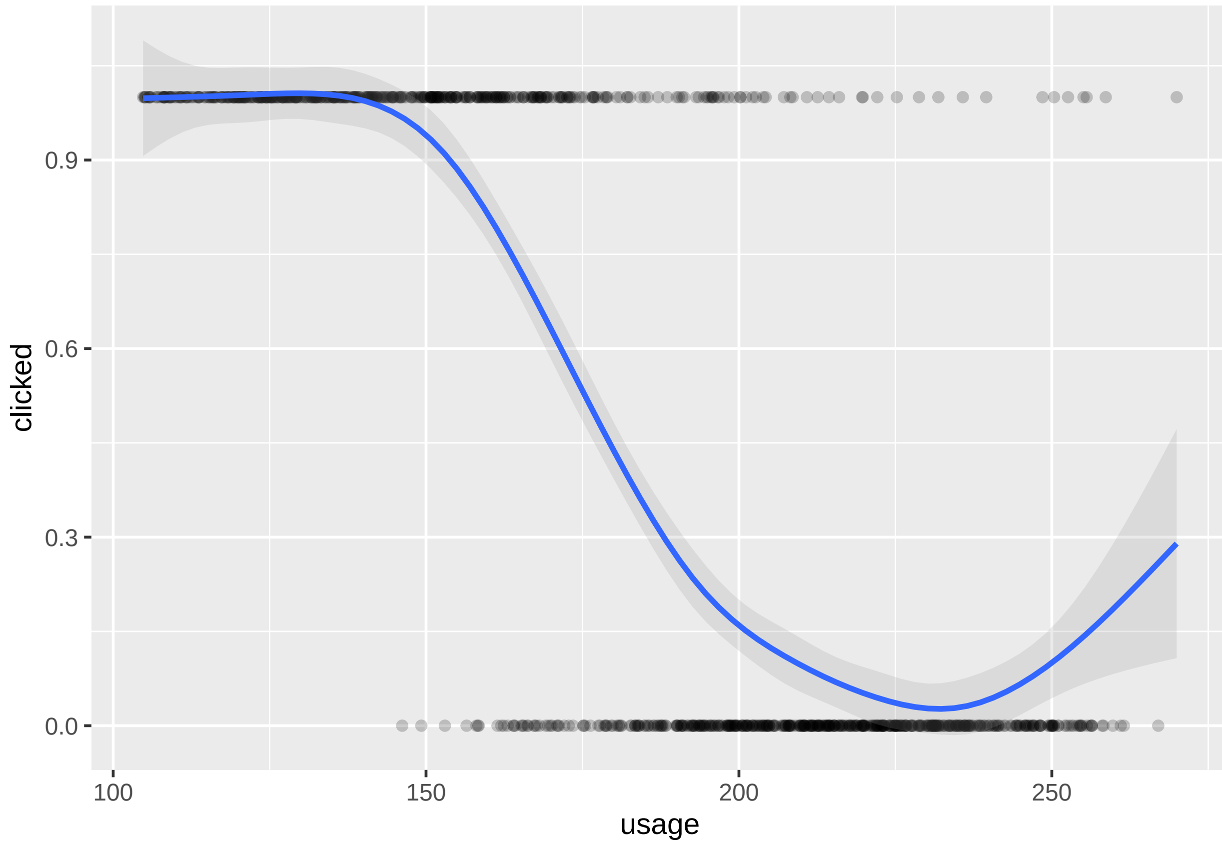

```r
# Plot to show realtionship between clicked and income
qplot(age,
      clicked,
      data = adv,
      geom = c("point", "smooth"),
      alpha = I(1 / 5))
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



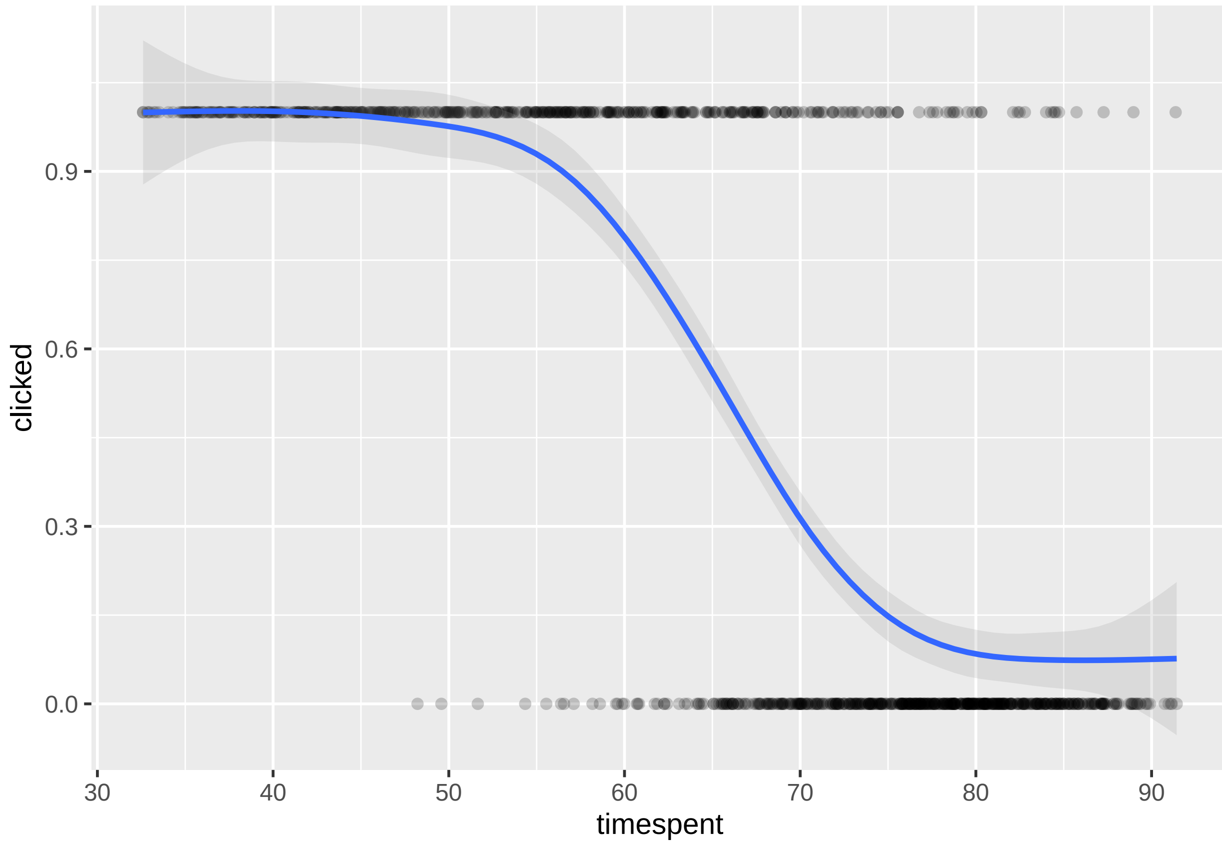

```r
# Plot to show realtionship between clicked and income
qplot(usage,
      clicked,
      data = adv,
      geom = c("point", "smooth"),
      alpha = I(1 / 5))
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



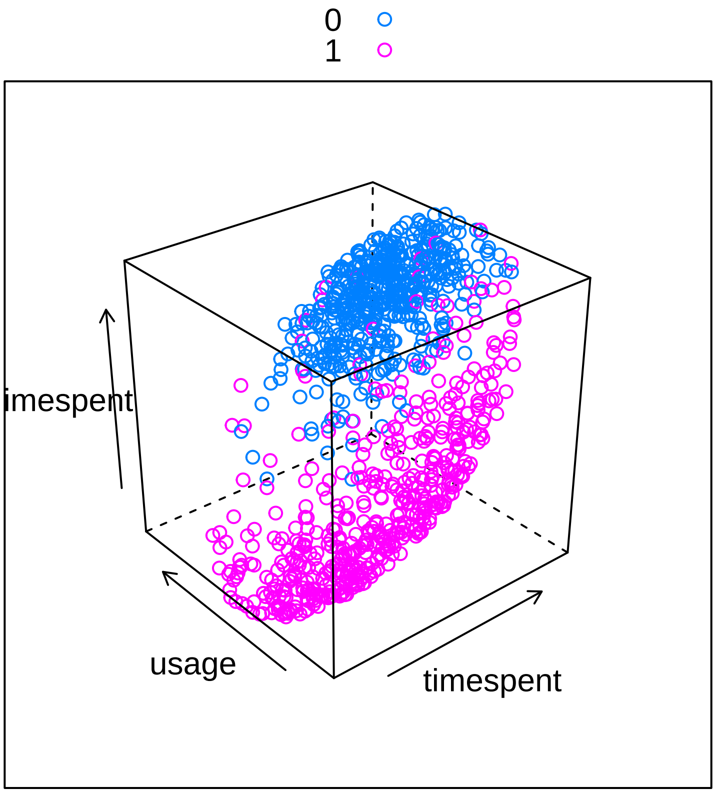

```
# Plot to show realtionship between clicked and income
qplot(timespent,
      clicked,
      data = adv,
      geom = c("point", "smooth"),
      alpha = I(1 / 5))
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



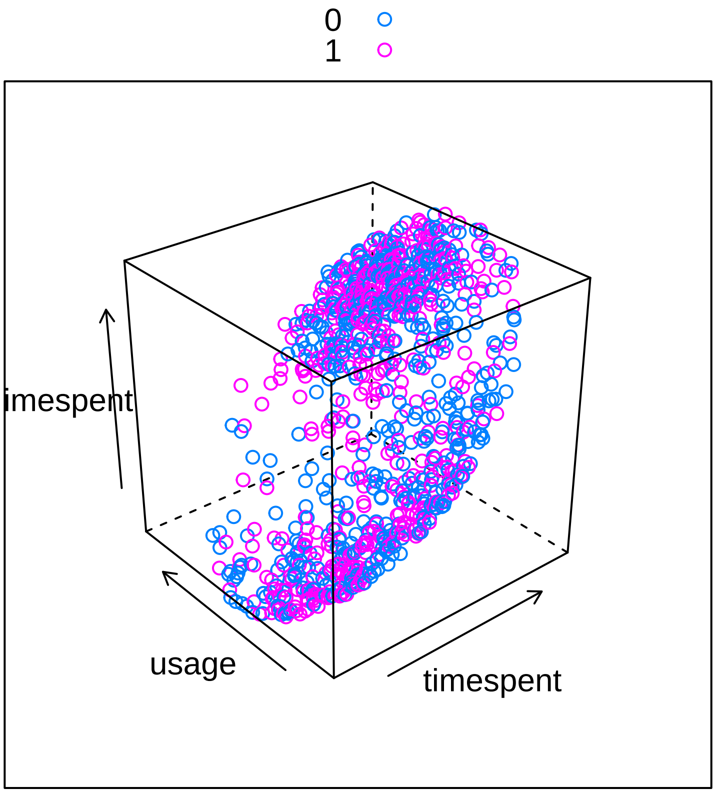
10.Multivariate Analysis

```
#we look at the timespent considering usage groping by clicked or not clicked
# Color by groups; auto.key = TRUE to show legend
cloud(timespent ~ timespent * usage,
       group = clicked, data = adv,
       auto.key = TRUE)
```



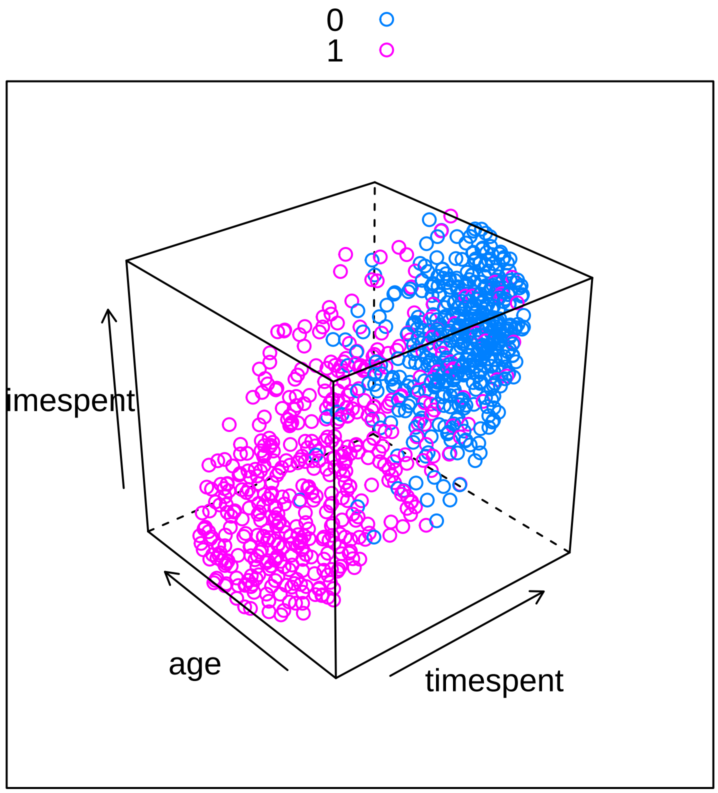
we observe that most clicked spend alot time online and have high usage the purple cluster represents the clicked and blue not clicked.

```
#we look at time spent and usage versus the Gender
cloud(timespent ~ timespent * usage,
       group = gender, data = adv,
       auto.key = TRUE)
```



No Gender spends more time online than the other or has high usage than the other its the same

```
#we llok if Age affescts time spent online and page being clicked
cloud(timespent ~ timespent * age,
       group = clicked, data = adv,
       auto.key = TRUE)
```



We observe that As Age increases and time spent increases so does the click .But age seems to be clustered more in the middle when it comes to click which is purple.

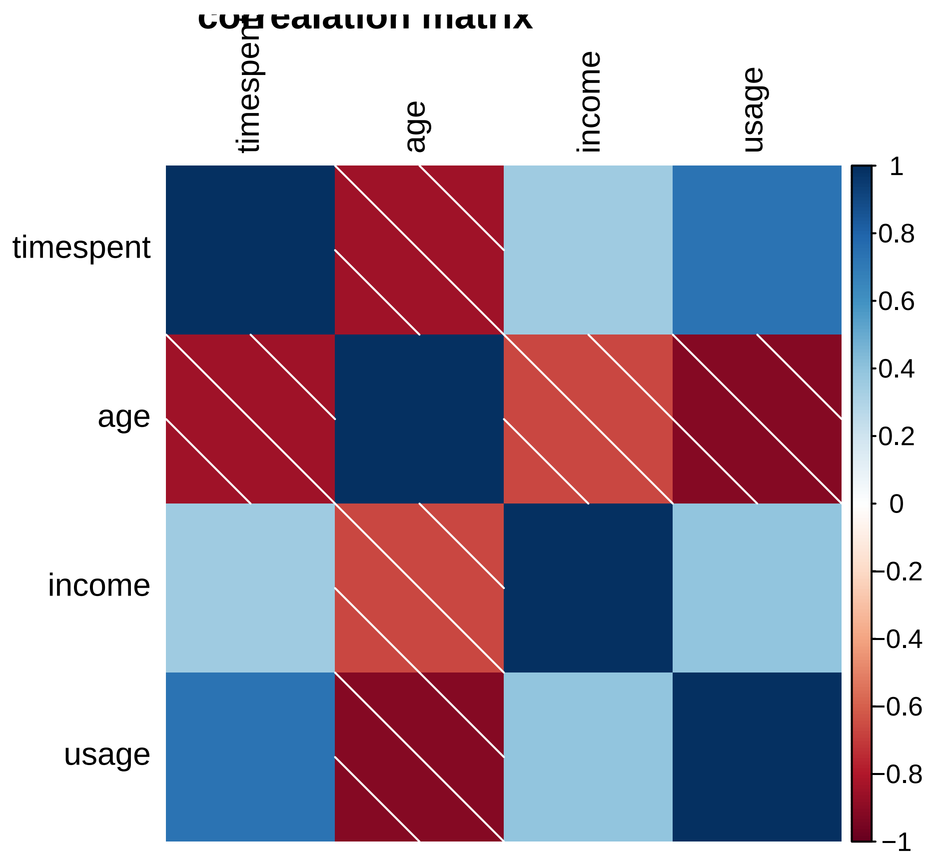
```r
library(corrplot)
```

```
## corrplot 0.84 loaded
```

```r
# Compute a correlation matrix

corr <- round(cor((adv[0:4])),1)
corr
```

```
##           timespent  age income usage
## timespent       1.0 -0.3    0.3   0.5
## age            -0.3  1.0   -0.2  -0.4
## income          0.3 -0.2    1.0   0.3
## usage           0.5 -0.4    0.3   1.0
```

```r
corrplot(cor(corr),        # Correlation matrix
         method = "shade", # Correlation plot method
         type = "full",    # Correlation plot style (also "upper" and "lower")
         diag = TRUE,      # If TRUE (default), adds the diagonal
         tl.col = "black", # Labels color
         bg = "white",     # Background color
         title = "correalation matrix",# Main title
         col = NULL)       # Color palette
```



we observe that The *income* and *Daily time spent on the site* columns have a large positive correlation and so does the *usage* and *timespent.Age* has a very negative correlation with time spent

11.Recommendations

From our indepth Analysis we would advice our client to;

*come up with ad campaigns that lure young people especially the age group (28 to 30) who spent alot of time online.

- since gender is does not affect click she should still decide on her target market invest her resources.

*People who earn alot tend to be the biggest clickers but they dont spend alot of time online.Would recommend to client to come up with service flexible to any income earner since the usage is the same whether Wealthy or not.

12.Feature Importance

- The dataset was appropriate. it contained no missing values and minimal outliers amongst the varaibles
- Both univariate and Bivariate analysis revealed that the dataset is collinear, hence it can be analysed better by use of a classification algorithms
- we will use PCA to determine the most features then we will go ahead and drop the not so important ones

- **And since we cant drop our label the clicked column we will use supervised classification algorithms.In our case we will use Decision Trees**

```r
# We then pass df to the prcomp(). We also set two arguments, center and scale
#we already had secluded the numerical values and changed it into a tibble
# to be TRUE then preview our object with summary
# ---
#
adv.pca <- prcomp(numt, center = TRUE, scale. = TRUE)
summary(adv.pca)
```

```
## Importance of components:
##                           PC1    PC2    PC3    PC4     PC5     PC6
## Standard deviation     1.7046 1.0017 0.9042 0.8224 0.69062 0.34661
## Proportion of Variance 0.4843 0.1672 0.1363 0.1127 0.07949 0.02002
## Cumulative Proportion  0.4843 0.6515 0.7877 0.9005 0.97998 1.00000
```

```r
head(adv.pca,n=3)
```

```
## $sdev
## [1] 1.7045718 1.0016516 0.9042431 0.8224279 0.6906161 0.3466056
##
## $rotation
##                   PC1         PC2          PC3        PC4         PC5
## timespent -0.46661499  0.07147213  0.035360556  0.4277301  0.68257371
## age        0.35113173  0.05024171 -0.638023495  0.6670941 -0.08742641
## income    -0.33512393  0.05140623 -0.765300250 -0.5271134  0.06746389
## usage     -0.48464331 -0.01960296  0.033147486  0.2695202 -0.71832385
## gender    -0.01773162 -0.99460175 -0.069917289  0.0303782  0.06274212
## clicked    0.55809977  0.01039174 -0.001998577 -0.1435965  0.04441454
##                   PC6
## timespent  0.35644346
## age       -0.12020962
## income     0.13024968
## usage      0.41833905
## gender     0.02655397
## clicked    0.81597798
##
## $center
##  timespent        age     income      usage     gender    clicked
##    65.0002    36.0090 55000.0001   180.0001     0.4810     0.5000
```

- As a result we obtain 6 principal components,
- each which explain a percentage of the total variation of the dataset
- PC1 explains 48% of the total variance, which means that nearly half.
- of the information in the dataset (6 variables) can be encapsulated.
- by just that one Principal Component. PC2 explains 17% of the variance and pc3 13%
- pc4 explains 11%,pc5 explains 7% and pc6 explains 2%
- We will consider timespent,age and income columns.

```r
#creating new dataframe with only important features

advf <- subset(adv, select = c(timespent, age, income) )

head(advf,n=3)
```

13.Implement the solution

```
## # A tibble: 3 x 3
##   timespent   age income
##       <dbl> <int>  <dbl>
## 1      69.0    35 61834.
## 2      80.2    31 68442.
## 3      69.5    26 59786.
```

```r
#modelling the decision trees

set.seed(12345)
train <- sample(1:nrow(advf),size = ceiling(0.70*nrow(advf)),replace = FALSE)

#we get our training set
adv_train <- advf[train,]

# test set
adv_test <- advf[-train,]
```

```r
# building the classification tree with rpart
library(rpart)

#tree <- rpart(clicked~,data=adv_train,method = "class")

tree <- rpart(
formula = timespent ~ .,
data = adv_train,
method = "anova"
)
tree
```

```
## n= 700
##
## node), split, n, deviance, yval
##       * denotes terminal node
##
##  1) root 700 177564.800 65.21231
##    2) income< 54357.63 295  84746.030 58.83871
##      4) age>=27.5 246  71496.490 56.93508
##        8) age>=35.5 165  41029.390 55.15030 *
##        9) age< 35.5 81  28870.850 60.57074
##         18) income< 48915.81 46  12966.450 56.28239 *
##         19) income>=48915.81 35  13946.650 66.20686 *
##      5) age< 27.5 49   7882.609 68.39571 *
```

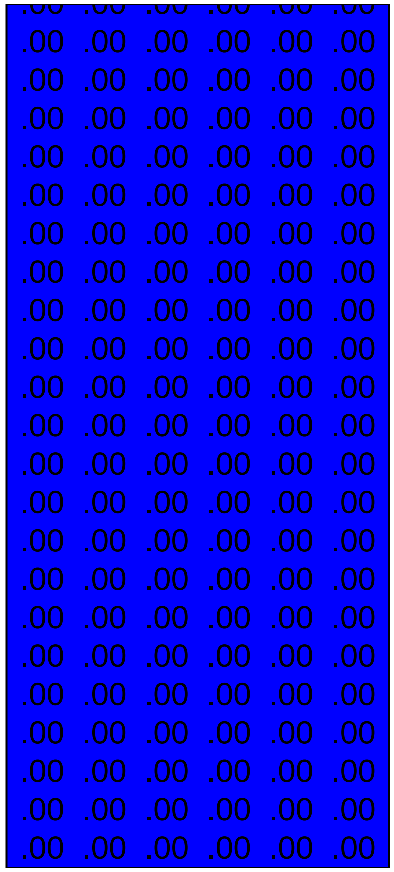
```
##    3) income>=54357.63 405  72106.140 69.85481
##      6) age>=41.5 72  14332.390 58.80931 *
##      7) age< 41.5 333  47090.210 72.24303 *
```

we will try with anova and classification an see which gives accurate

```r
# building the classification tree with rpart
library(rpart)

#tree <- rpart(clicked~,data=adv_train,method = "class")

tree2 <- rpart(
formula = timespent ~ .,
data = adv_train,
method = "class"
)
```

```r
# Visualize the decision tree with rpart.plot

library(rpart.plot)

#rpart.plot(tree, nn=TRUE,colourPalette)
```

```r
# Visualize the decision tree with rpart.plot

library(rpart.plot)

rpart.plot(tree2, nn=TRUE,box.palette="blue")
```




```r
#Testing the model

#pred1 <- predict(object = tree,  newdata = adv_test,   type = "anova")
```

```r
#Testing the model

pred <- predict(object = tree2,
                             newdata = adv_test,
                             type = "class")
```

```r
#Calculating accuracy
library(caret)
```

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

## The following object is masked from 'package:mosaic':
##
##     dotPlot

## The following object is masked from 'package:purrr':
##
##     lift
```

```
adva <- confusionMatrix(data = pred,
                reference = pred)
#head(adva,n=3)
```

#results $overall Accuracy Kappa AccuracyLower AccuracyUpper AccuracyNull 1.000000 NaN 0.987779 1.000000 1.000000 AccuracyPValue McnemarPValue 1.000000 NaN

The results show that all the samples in the test dataset have been correctly classified and we've attained an accuracy of 100% on the test data set with a 95% confidence interval (0.9877, 1).

Class 0 on clicking on ads takes the day.