FINALPROJECT

Muhammed Khan

```
setwd("C:/Users/AliDesktop/Desktop/Bit Briefcase/Big Data/Kaggle/CTR")
#---------------------- Data preparation

# Click through rate is an important metric for evaluating performance
# in online advertising and marketing. The data is from the world's leading
# company specialized in cross device advertising.240  hours of click data
# is available.
# Column headers defined at https://www.kaggle.com/c/avazu-ctr-prediction/data

# Problem statement

# Predicting whether a mobile ad will be clicked. Historical data
# is provided. Metrics related to the app, site, device, time of activity
# is available. Using the data given, have to predict the variables which
# assist in the conversion of click.


#----First determine number of lines in CSV-----
# testcon <- file("C:/Users/AliDesktop/Desktop/Bit Briefcase/Big Data/Kaggle/CTR/train.csv",open="r")
# readsizeof <- 20000
# nooflines <- 0
# ( while((linesread <- length(readLines(testcon,readsizeof))) > 0 )
#   nooflines <- nooflines+linesread )
# close(testcon)
# nooflines
# > nooflines
# total # of 40,428,968 rows in train.csv file size is 6163 MB
# ~=6560 rows per MB


# Since it is the hourly click data, the dataset size is large. We must import  it
# by parts and taking a 10% sample out of it.


ctr <- read.csv("train.csv", nrow = 1000000) # importing by parts
ctr_sample <- sample(nrow(ctr), nrow(ctr)*0.1)
ctr_1 <- ctr[ctr_sample,] # creating 10% sample
nrow(ctr_1)
```

```
## [1] 100000
```

```
rm(ctr)

ctr2 <-  read.csv("train.csv", header = TRUE,nrow = 1000000, skip = 1000000,
col.names = c("id","click", "hour","C1", "banner_pos", "site_id"  , "site_domain",
"site_category", "app_id" , "app_domain" ,"app_category", "device_id", "device_ip",
"device_model","device_type","device_conn_type", "C14" , "C15", "C16", "C17", "C18",
"C19", "C20" , "C21"))
```

```
ctr_sample <- sample(nrow(ctr2), nrow(ctr2)*0.1)
ctr_2 <- ctr2[ctr_sample,]
nrow(ctr_2)
```

```
## [1] 100000
```

```
rm(ctr2)

ctr3 <-  read.csv("train.csv", header = TRUE,nrow = 1000000, skip = 2000000,
col.names = c("id","click", "hour","C1", "banner_pos", "site_id"  , "site_domain",
 "site_category", "app_id" , "app_domain" ,"app_category", "device_id", "device_ip",
 "device_model","device_type","device_conn_type", "C14" , "C15", "C16", "C17", "C18",
 "C19", "C20" , "C21"))
ctr_sample <- sample(nrow(ctr3), nrow(ctr3)*0.1)
ctr_3 <- ctr3[ctr_sample,]
nrow(ctr_3)
```

```
## [1] 100000
```

```
rm(ctr3)

ctr4 <-  read.csv("train.csv", header = TRUE,nrow = 1000000, skip = 3000000,
col.names = c("id","click", "hour","C1", "banner_pos", "site_id"  , "site_domain",
"site_category", "app_id" , "app_domain" ,"app_category", "device_id", "device_ip",
"device_model","device_type","device_conn_type", "C14" , "C15", "C16", "C17", "C18",
 "C19", "C20" , "C21"))
ctr_sample <- sample(nrow(ctr4), nrow(ctr4)*0.1)
ctr_4 <- ctr4[ctr_sample,]
nrow(ctr_4)
```

```
## [1] 100000
```

```
rm(ctr4)

ctr5 <-  read.csv("train.csv", header = TRUE,nrow = 1000000, skip = 4000000,
col.names = c("id","click", "hour","C1", "banner_pos", "site_id"  , "site_domain",
"site_category", "app_id" , "app_domain" ,"app_category", "device_id", "device_ip",
"device_model","device_type","device_conn_type", "C14" , "C15", "C16", "C17", "C18",
"C19", "C20" , "C21"))
ctr_sample <- sample(nrow(ctr5), nrow(ctr5)*0.1)
ctr_5 <- ctr5[ctr_sample,]
nrow(ctr_5)
```

```
## [1] 100000
```

```
rm(ctr5)

ctr6 <-  read.csv("train.csv", header = TRUE,nrow = 1000000, skip = 5000000,
 col.names = c("id","click", "hour","C1", "banner_pos", "site_id"  , "site_domain",
 "site_category", "app_id" , "app_domain" ,"app_category", "device_id", "device_ip",
```

```
 "device_model","device_type","device_conn_type", "C14" , "C15", "C16", "C17", "C18",
 "C19", "C20" , "C21"))
ctr_sample <- sample(nrow(ctr6), nrow(ctr6)*0.1)
ctr_6 <- ctr6[ctr_sample,]
nrow(ctr_6)
```

```
## [1] 100000
```

```
rm(ctr6)

ctr7 <-  read.csv("train.csv", header = TRUE,nrow = 1000000, skip = 6000000,
col.names = c("id","click", "hour","C1", "banner_pos", "site_id"  , "site_domain",
 "site_category", "app_id" , "app_domain" ,"app_category", "device_id", "device_ip",
 "device_model","device_type","device_conn_type", "C14" , "C15", "C16", "C17", "C18",
 "C19", "C20" , "C21")  )
ctr_sample <- sample(nrow(ctr7), nrow(ctr7)*0.1)
ctr_7 <- ctr7[ctr_sample,]
nrow(ctr_7)
```

```
## [1] 100000
```

```
rm(ctr7)


# Merging all the parts together
ctr_set1 <- rbind(ctr_1, ctr_2, ctr_3, ctr_4, ctr_5, ctr_6, ctr_7)


# Checking the data size

#object_size attempts to take into account the size of the environments associated with an object.
# This is particularly important for closures and formulas, since otherwise you may not realise that
# you've accidentally captured a large object.

#Additionally, the env argument allows you to specify another environment at which to stop.
#This defaults to the environment from which object_size is called to prevent double-counting of
#objects created elsewhere.

library(pryr)
object_size(ctr_set1)
```

```
## 249 MB
```

```
# Verifying the data

nrow(ctr_set1)
```

```
## [1] 700000
```

```r
# Column names of the dataset
colnames(ctr_set1)
```

```
##  [1] "id"               "click"            "hour"
##  [4] "C1"               "banner_pos"       "site_id"
##  [7] "site_domain"      "site_category"    "app_id"
## [10] "app_domain"       "app_category"     "device_id"
## [13] "device_ip"        "device_model"     "device_type"
## [16] "device_conn_type" "C14"              "C15"
## [19] "C16"              "C17"              "C18"
## [22] "C19"              "C20"              "C21"
```

```r
# Sample dataset
head(ctr_set1)
```

```
##                  id click     hour   C1 banner_pos  site_id site_domain
## 166244 1.569981e+19     0 14102101 1005          1 856e6d3f    58a89a43
## 447599 8.706423e+18     0 14102102 1005          0 1fbe01fe    f3845767
## 432875 7.527377e+17     0 14102102 1005          0 1fbe01fe    f3845767
## 721121 1.399846e+19     0 14102104 1005          0 b99a2c43    cc962a1f
## 242648 8.330406e+18     1 14102101 1005          0 1fbe01fe    f3845767
## 317160 1.486551e+19     0 14102102 1005          0 85f751fd    c4e18dd6
##        site_category   app_id app_domain app_category device_id device_ip
## 166244      f028772b ecad2386   7801e8d9     07d7df22  a99f214a  27838c15
## 447599      28905ebd ecad2386   7801e8d9     07d7df22  a99f214a  cb514114
## 432875      28905ebd ecad2386   7801e8d9     07d7df22  a99f214a  a34de27f
## 721121      f028772b ecad2386   7801e8d9     07d7df22  a99f214a  a89031c5
## 242648      28905ebd ecad2386   7801e8d9     07d7df22  a99f214a  d90eb941
## 317160      50e219e0 54c5d545   2347f47a     0f2161f8  345ec3f6  2c44fdd5
##        device_model device_type device_conn_type   C14 C15 C16  C17 C18
## 166244     9efa421a           1                0 19771 320  50 2227   0
## 447599     1f0bc64f           1                0 15704 320  50 1722   0
## 432875     3bd9e8e7           1                0 15699 320  50 1722   0
## 721121     5096d134           1                0 19251 320  50 2201   3
## 242648     5096d134           1                0 15708 320  50 1722   0
## 317160     8a2cc27a           1                0 21647 320  50 2487   1
##        C19    C20 C21
## 166244 687 100075  48
## 447599  35     -1  79
## 432875  35     -1  79
## 721121  35 100119  43
## 242648  35 100083  79
## 317160 547     -1  51
```

```r
#Event rate: it is the no of 1's in the click = 16.6%
table(ctr_set1$click)
```

```
##
##      0      1
## 583541 116459
```

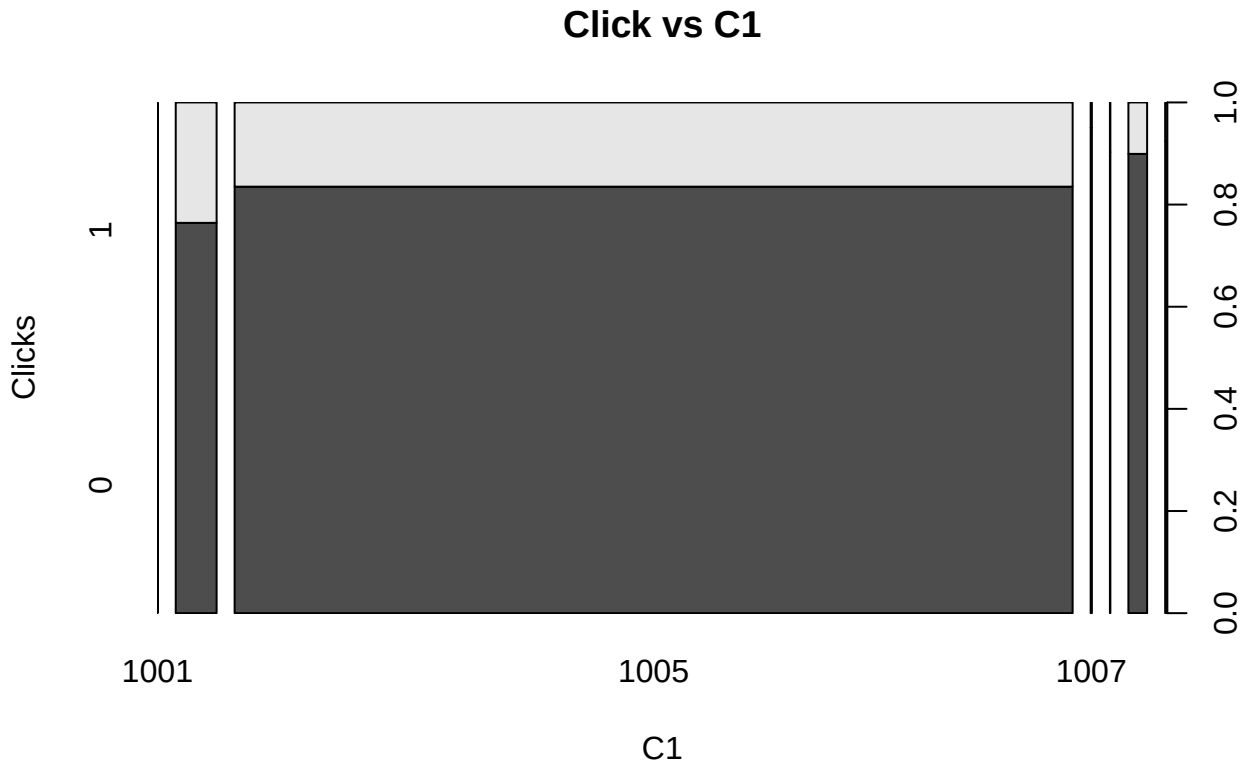
```
# Summary of the dataset - this displays the class of each column, min, median, mean, max
# values in case of continuous variables and distribution in case of categorical variables

summary(ctr_set1)
```

```
##        id                 click             hour                C1
##  Min.   :2.053e+12   Min.   :0.0000   Min.   :14102100   Min.   :1001
##  1st Qu.:4.561e+18   1st Qu.:0.0000   1st Qu.:14102108   1st Qu.:1005
##  Median :9.352e+18   Median :0.0000   Median :14102118   Median :1005
##  Mean   :9.269e+18   Mean   :0.1664   Mean   :14102150   Mean   :1005
##  3rd Qu.:1.391e+19   3rd Qu.:0.0000   3rd Qu.:14102206   3rd Qu.:1005
##  Max.   :1.845e+19   Max.   :1.0000   Max.   :14102211   Max.   :1012
##
##    banner_pos         site_id         site_domain      site_category
##  Min.   :0.0000   85f751fd:271288   c4e18dd6:280985   50e219e0:296591
##  1st Qu.:0.0000   1fbe01fe:108301   f3845767:108301   f028772b:203989
##  Median :0.0000   e151e245: 29652   7e091613: 35769   28905ebd:131001
##  Mean   :0.2464   d9750ee7: 19862   98572c79: 20838   3e814130: 49700
##  3rd Qu.:0.0000   856e6d3f: 14911   7687a86e: 19166   f66779e6:  7779
##  Max.   :7.0000   5b08c53b: 14009   58a89a43: 14911   335d28a8:  2959
##                   (Other) :241977   (Other) :220030   (Other) :  7981
##       app_id          app_domain       app_category       device_id
##  ecad2386:428712   7801e8d9:455537   07d7df22:436365   a99f214a:571556
##  e2fcccd2: 18191   2347f47a:110332   0f2161f8:173530   c357dbff:  1511
##  a5184c22: 17322   5c5a694b: 18192   cef3e649: 40866   936e92fb:   319
##  7358e05e: 14326   b8d325c3: 17461   f95efa07: 22362   afeffc18:   135
##  66f5e02e: 12621   d9b5648e: 17121   8ded1f7a: 18215   cef4c8cc:    90
##  febd1138: 11430   b9528b13: 14870   d1327cf5:  3054   d2e4c0ab:    85
##  (Other) :197398   (Other) : 66487   (Other) :  5608   (Other) :126304
##     device_ip        device_model     device_type   device_conn_type
##  6b9769f2:  3416   8a4875bd: 39604   Min.   :0.00   Min.   :0.0000
##  431b3174:  2216   1f0bc64f: 25404   1st Qu.:1.00   1st Qu.:0.0000
##  e9d5636f:  2081   d787e91b: 23996   Median :1.00   Median :0.0000
##  c6563308:  1211   7abbbd5c: 12829   Mean   :1.02   Mean   :0.2479
##  488a9a3e:  1184   76dc4769: 12357   3rd Qu.:1.00   3rd Qu.:0.0000
##  ceffea69:  1183   4ea23a13: 12353   Max.   :5.00   Max.   :5.0000
##  (Other) :688709   (Other) :573457
##       C14              C15               C16               C17
##  Min.   :  375   Min.   : 120.0   Min.   :  20.00   Min.   : 112
##  1st Qu.:17566   1st Qu.: 320.0   1st Qu.:  50.00   1st Qu.:1899
##  Median :20362   Median : 320.0   Median :  50.00   Median :2333
##  Mean   :18763   Mean   : 318.8   Mean   :  58.68   Mean   :2115
##  3rd Qu.:21746   3rd Qu.: 320.0   3rd Qu.:  50.00   3rd Qu.:2502
##  Max.   :21924   Max.   :1024.0   Max.   :1024.00   Max.   :2528
##
##       C18              C19               C20               C21
##  Min.   :0.000   Min.   :  33.0   Min.   :    -1   Min.   : 13.00
##  1st Qu.:0.000   1st Qu.:  35.0   1st Qu.:    -1   1st Qu.: 33.00
##  Median :2.000   Median :  39.0   Median :    -1   Median : 68.00
##  Mean   :1.425   Mean   : 176.1   Mean   : 48843   Mean   : 89.77
##  3rd Qu.:3.000   3rd Qu.: 167.0   3rd Qu.:100084   3rd Qu.:157.00
##  Max.   :3.000   Max.   :1835.0   Max.   :100248   Max.   :221.00
##
```

```r
# Variable click is the dependent variable here. Few columns needs to be treated before
# starting off with the analysis. Treatment includes changing it to suitable datatype and
# removing the null values.


# Changing the  datatypes. This is an important step for long term projects out of the scope of this #
# a decision tree classifier around CTR down the road.

ctr_set1$id <- as.numeric(ctr_set1$id)
ctr_set1$hour <- as.factor(ctr_set1$hour)
ctr_set1$click <- as.factor(ctr_set1$click)
ctr_set1$C1 <- as.factor(ctr_set1$C1)
ctr_set1$C14 <- as.factor(ctr_set1$C14)
ctr_set1$C15 <- as.factor(ctr_set1$C15)
ctr_set1$C16 <- as.factor(ctr_set1$C16)
ctr_set1$C17 <- as.factor(ctr_set1$C17)
ctr_set1$C18 <- as.factor(ctr_set1$C18)
ctr_set1$C19 <- as.factor(ctr_set1$C19)
ctr_set1$C20 <- as.factor(ctr_set1$C20)
ctr_set1$C21 <- as.factor(ctr_set1$C21)
ctr_set1$banner_pos <- as.factor(ctr_set1$banner_pos)
ctr_set1$device_type <- as.factor(ctr_set1$device_type)
ctr_set1$device_conn_type <- as.factor(ctr_set1$device_conn_type)

# Replacing the null values with zero

ctr_set1[is.na(ctr_set1)] <- 0

# Bivariate profiling
attach(ctr_set1)

# Event is click.

# Click vs C1

table(C1, click)
```

```
##       click
## C1          0      1
##   1001    188      5
##   1002  24245   7481
##   1005 543256 107298
##   1007    885     46
##   1008    325     35
##   1010  13070   1465
##   1012   1572    129
```

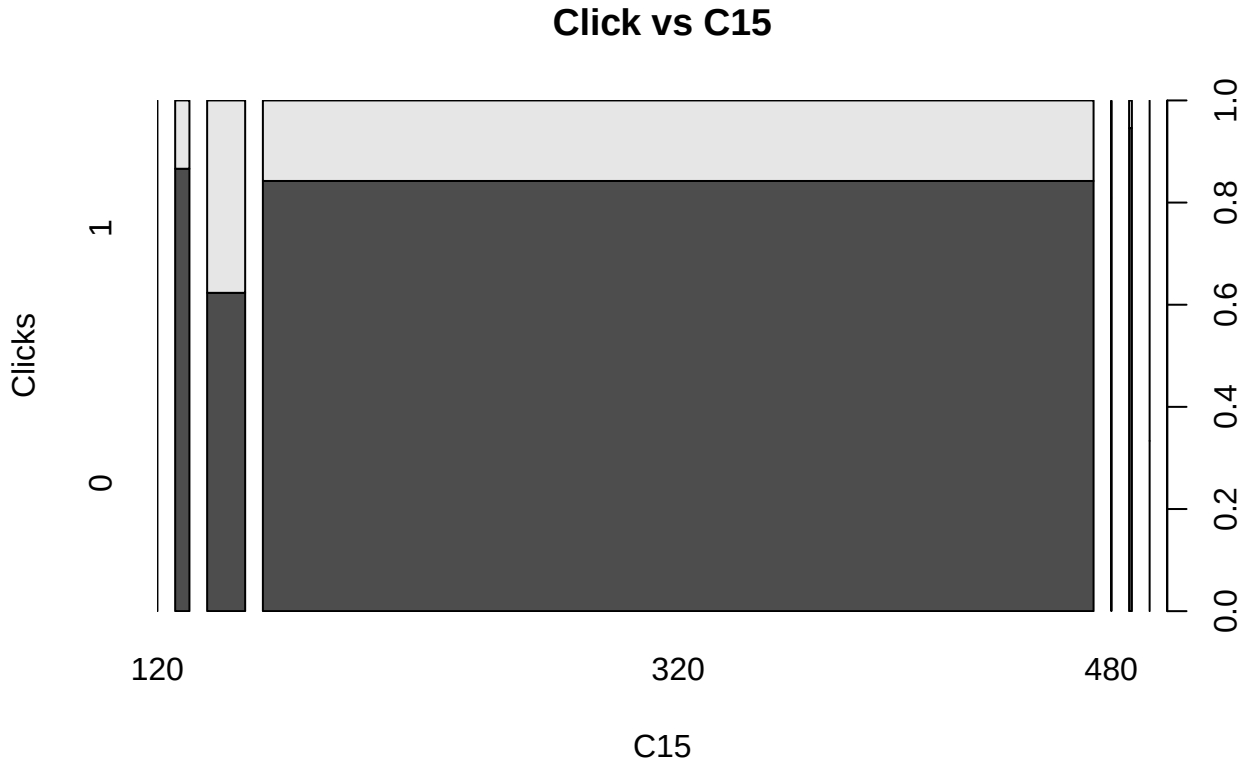
```r
plot(C1, click, xlab = "C1", ylab= "Clicks", main = "Click vs C1")
```




```
# From the plot, it is observed that the category 1005 has maximum population
# and 1002 has a better event rate comparing to the other categories.

# Click vs C15

table(C15, click)
```

```
##       click
## C15         0      1
##   120       2      0
##   216    9770   1509
##   300   18721  11316
##   320  552894 103489
##   480      55     13
##   728    2091    120
##   768       3      6
##   1024      5      6
```

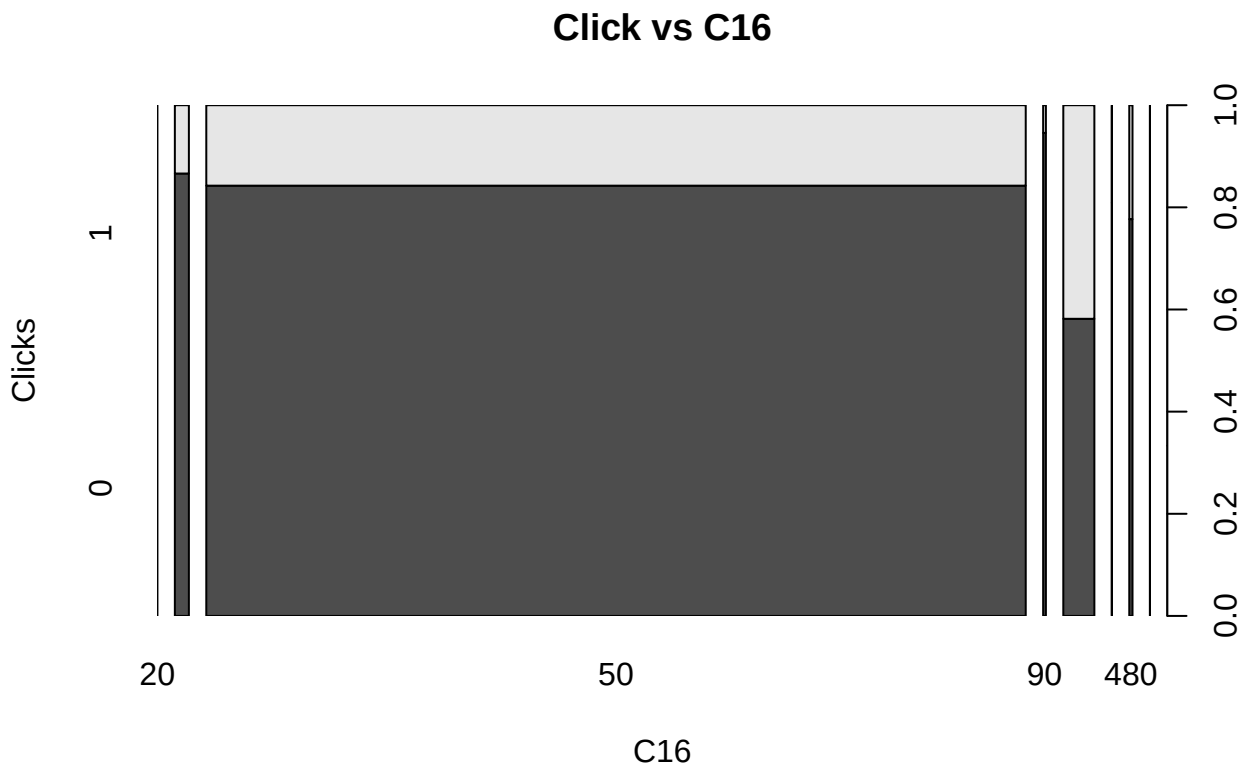
```
plot(C15, click, xlab = "C15", ylab= "Clicks", main = "Click vs C15")
```




```
# Category C15=320 captures maximum touches to the respective page and category 300
# has maximum event rate.


# Click vs C16

table(C16, click)
```

```
##       click
## C16         0      1
##   20        2      0
##   36     9770   1509
##   50   555074 103783
##   90     2091    120
##   250   14525  10443
##   320      55     13
##   480    2016    579
##   768       5      6
##   1024      3      6
```

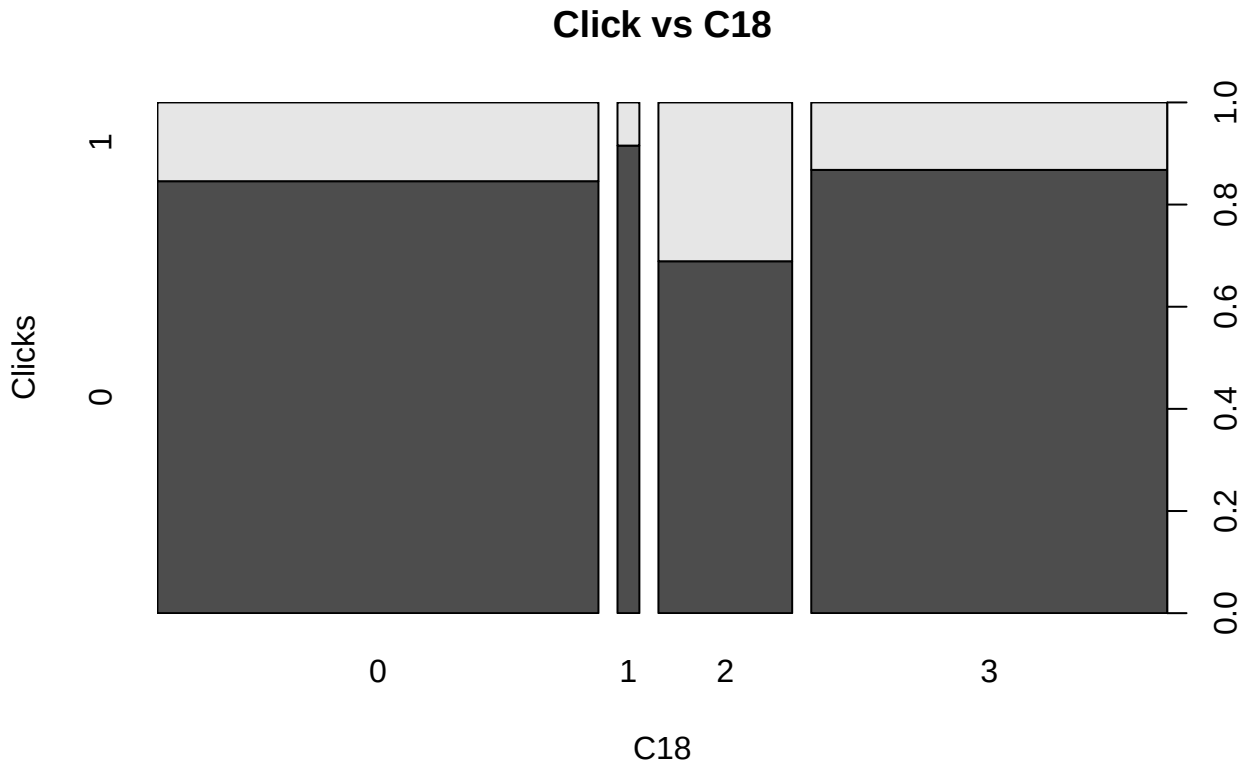
```r
plot(C16, click, xlab = "C16", ylab= "Clicks", main = "Click vs C16")
```




```r
# Only the categories 36, 50, 90, 250 and 480 has considerable traffic and from the
# plot it is inferred the category 250 has higher event rate.

# Click vs C18

table(C18, click)
```

```
##    click
## C18      0      1
##   0 274043  49968
##   1  14718   1359
##   2  67719  30564
##   3 227061  34568
```

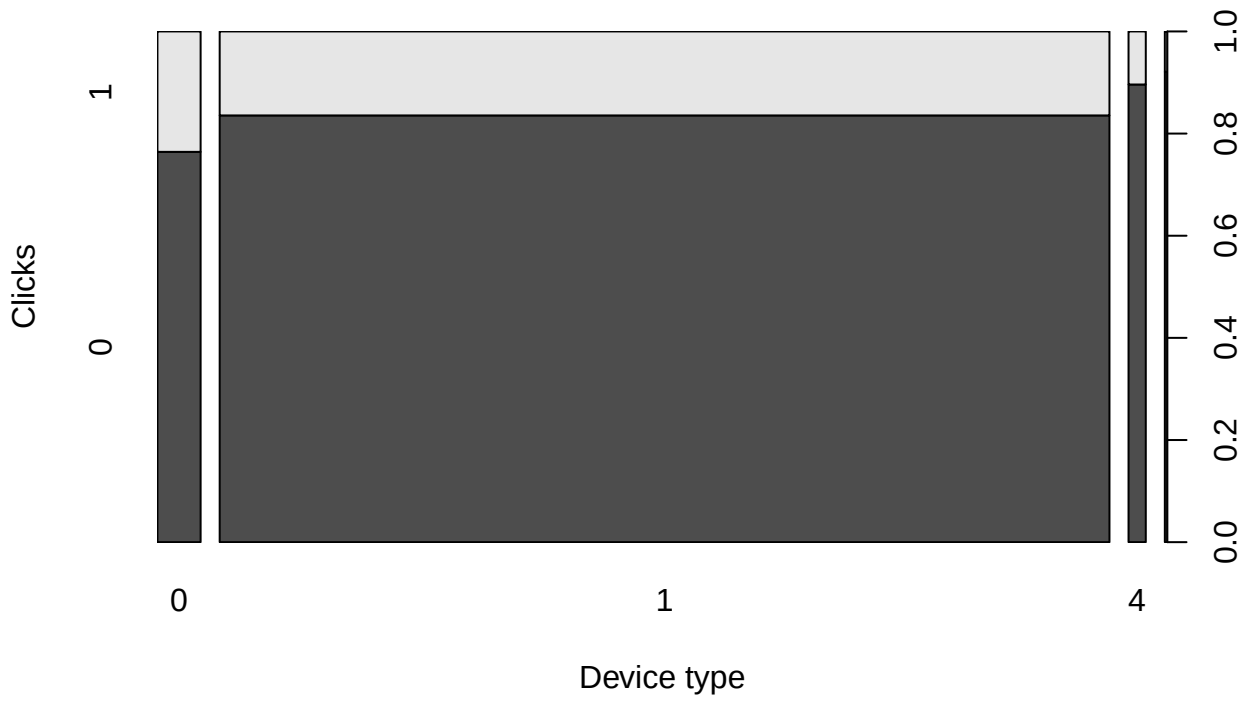
```r
plot(C18, click, xlab = "C18", ylab= "Clicks", main = "Click vs C18")
```




```
# Category C18 =2 has a conversion rate of 46% which is higher than all the other
# categories.


# Click vs Device type

table(device_type, click)
```

```
##             click
## device_type      0      1
##           0  24245   7481
##           1 546226 107513
##           4  11364   1318
##           5   1706    147
```

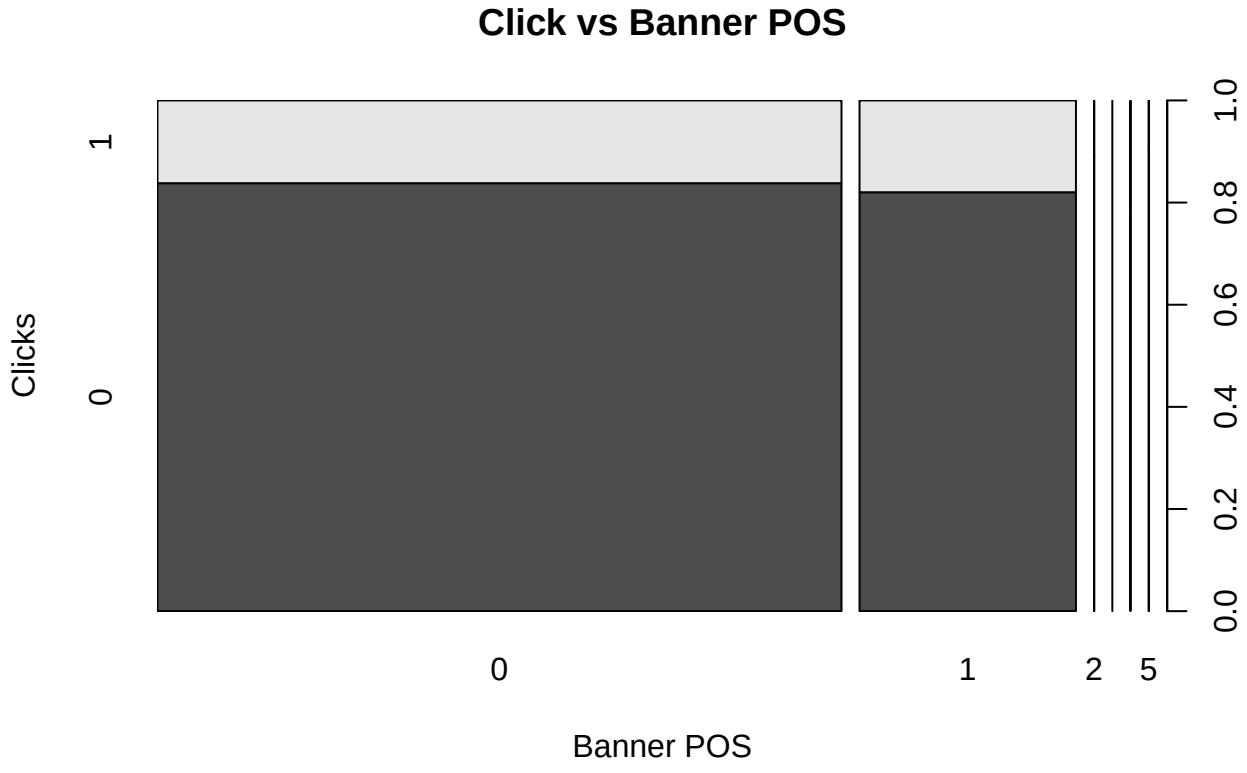
```
plot(device_type, click, xlab = "Device type", ylab= "Clicks", main = "Click vs Device type")
```




```
# device_type = 1 captures maximum traffic.

# Click vs Banner POS

table(banner_pos, click)
```

```
##            click
## banner_pos      0      1
##          0 444964  86059
##          1 137814  30208
##          2    216     26
##          3      5      0
##          4     81     23
##          5    325     35
##          7    136    108
```

```
plot(banner_pos, click, xlab = "Banner POS", ylab= "Clicks", main = "Click vs Banner POS")
```




```
# Almost all the traffic to the advertisement page comes via the banner_pos 0 and 1 with both
# of them having almost equal conversion rate.


# Segmenting train and test dataset.

sample_2 <- sample(nrow(ctr_set1), nrow(ctr_set1)*0.7)
train <- ctr_set1[-sample_2, ]

#Dataset on which to build model
test <- ctr_set1[-sample_2, ]


# Analysis methodology

# Using ANOVA, finding out the significant variables and building the model using
#logistic regression

# Anova

anova_1 <- aov(as.numeric(click) ~ C1+C15+C16+C18+banner_pos+device_conn_type+
device_type+site_category+app_category, data = train)

summary(anova_1)
```

```
##                      Df Sum Sq Mean Sq F value   Pr(>F)
```

```
## C1                    6     75   12.47  93.805  < 2e-16 ***
## C15                   6    428   71.39 536.953  < 2e-16 ***
## C16                   2     91   45.47 341.987  < 2e-16 ***
## C18                   3    369  122.98 924.956  < 2e-16 ***
## banner_pos            5     27    5.37  40.382  < 2e-16 ***
## device_conn_type      3     26    8.80  66.175  < 2e-16 ***
## device_type           1      0    0.01   0.069    0.793
## site_category        18    159    8.86  66.612  < 2e-16 ***
## app_category         22     14    0.62   4.632 3.02e-12 ***
## Residuals        209934  27913    0.13
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#C1, C15, C16, C18, banner_pos, device_conn_type, site_category, app_category came out
#to be significant. Hence we can proceed with logistic regression having the above as
#independent varaibles.


# Logistic regression

log_1 <- glm(click ~ C1+C15+C16+C18+banner_pos+device_conn_type+site_category+app_category,
data = train, family = binomial)
summary(log_1)
```

```
##
## Call:
## glm(formula = click ~ C1 + C15 + C16 + C18 + banner_pos + device_conn_type +
##     site_category + app_category, family = binomial, data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.2915  -0.6284  -0.5336  -0.4574   2.8060
##
## Coefficients: (6 not defined because of singularities)
##                         Estimate Std. Error z value Pr(>|z|)
## (Intercept)          -15.796683  88.632991  -0.178 0.858546
## C11002                13.275249  88.633223   0.150 0.880940
## C11005                11.909552  88.632250   0.134 0.893110
## C11007                11.299446  88.632915   0.127 0.898556
## C11008                12.809996  88.633752   0.145 0.885084
## C11010                11.417824  88.632250   0.129 0.897498
## C11012                11.310568  88.632562   0.128 0.898456
## C15300                 0.916675   0.127925   7.166 7.74e-13 ***
## C15320                 0.705103   0.107326   6.570 5.04e-11 ***
## C15480                -0.252829   0.756975  -0.334 0.738380
## C15728                -0.620928   0.180420  -3.442 0.000578 ***
## C15768                 1.569497   1.139300   1.378 0.168328
## C151024               14.369553 535.411234   0.027 0.978589
## C1650                 -0.562411   0.094636  -5.943 2.80e-09 ***
## C1690                       NA         NA      NA       NA
## C16250                 0.217841   0.123752   1.760 0.078355 .
## C16320                      NA         NA      NA       NA
## C16480                      NA         NA      NA       NA
## C16768                      NA         NA      NA       NA
```

```
## C161024                       NA         NA      NA       NA
## C181                    -0.397420   0.052733  -7.536 4.83e-14 ***
## C182                     0.731554   0.018567  39.400  < 2e-16 ***
## C183                    -0.011445   0.015249  -0.751 0.452951
## banner_pos1              0.214421   0.019599  10.940  < 2e-16 ***
## banner_pos2              1.301480   0.403164   3.228 0.001246 **
## banner_pos3            -10.516460 304.156930  -0.035 0.972418
## banner_pos4              0.277495   0.505108   0.549 0.582747
## banner_pos5                   NA         NA      NA       NA
## banner_pos7              1.905629   0.265823   7.169 7.57e-13 ***
## device_conn_type2       -0.143116   0.023250  -6.156 7.48e-10 ***
## device_conn_type3       -0.542174   0.061811  -8.771  < 2e-16 ***
## device_conn_type5       -1.102292   0.319359  -3.452 0.000557 ***
## site_category28905ebd    2.285248   0.359350   6.359 2.03e-10 ***
## site_category335d28a8    1.036980   0.378625   2.739 0.006166 **
## site_category3e814130    1.858981   0.359859   5.166 2.39e-07 ***
## site_category42a36e14    2.520954   0.943614   2.672 0.007549 **
## site_category50e219e0    0.868303   0.548187   1.584 0.113205
## site_category5378d028   -9.821627 378.593038  -0.026 0.979303
## site_category70fb0e29    1.542463   0.433399   3.559 0.000372 ***
## site_category72722551    0.625061   0.436443   1.432 0.152095
## site_category75fa27f6    1.652120   0.379905   4.349 1.37e-05 ***
## site_category76b2941d    0.490716   0.422583   1.161 0.245549
## site_category8fd0aea4   -9.810183 535.411288  -0.018 0.985381
## site_categorya818d37a -11.257135  74.112013  -0.152 0.879271
## site_categorybcf865d9    2.169330 235.769950   0.009 0.992659
## site_categoryc0dd3be3    1.067297   0.471991   2.261 0.023743 *
## site_categorydedf689d    1.882146   0.447837   4.203 2.64e-05 ***
## site_categorye787de0e    2.216973   0.901292   2.460 0.013902 *
## site_categoryf028772b    1.878716   0.358957   5.234 1.66e-07 ***
## site_categoryf66779e6    0.007544   0.385005   0.020 0.984366
## app_category09481d60     1.193096   0.426290   2.799 0.005129 **
## app_category0bfbc358   -11.578930 218.580911  -0.053 0.957753
## app_category0f2161f8     0.956003   0.414498   2.306 0.021088 *
## app_category0f9a328c     1.393544   0.588968   2.366 0.017978 *
## app_category18b1e0be    -8.767382 225.242507  -0.039 0.968951
## app_category2281a340     1.301027 148.913890   0.009 0.993029
## app_category4681bb9d     2.036852   0.705205   2.888 0.003873 **
## app_category4ce2e9fc     1.582509   0.472866   3.347 0.000818 ***
## app_category75d80bbe     1.262505   0.465838   2.710 0.006725 **
## app_category79f0b860     1.341244 199.213036   0.007 0.994628
## app_category879c24eb     0.961822   0.491229   1.958 0.050231 .
## app_category8ded1f7a     0.682718   0.416923   1.638 0.101523
## app_category8df2e842     2.082848   0.821161   2.536 0.011198 *
## app_categorya3c42688     1.564941   0.717176   2.182 0.029103 *
## app_categorya7fd01ec     3.535279   1.485849   2.379 0.017346 *
## app_categorya86a3e89   -10.278055 124.390351  -0.083 0.934148
## app_categorycef3e649     0.889987   0.415259   2.143 0.032096 *
## app_categoryd1327cf5     0.434983   0.431335   1.008 0.313235
## app_categorydc97ec06     1.274343   0.495901   2.570 0.010177 *
## app_categoryf95efa07     1.144717   0.415690   2.754 0.005891 **
## app_categoryfc6fa53d    -0.150484   0.587740  -0.256 0.797922
## app_category4b7ade46     1.374183 542.697860   0.003 0.997980
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 188924  on 210000  degrees of freedom
## Residual deviance: 181019  on 209935  degrees of freedom
## AIC: 181151
##
## Number of Fisher Scoring iterations: 12
```

```
# Only the variables C15, C18, banner_pos,device_conn_type,site_category came out to be
# significant in the first trial of logistic regression. Iterating the same process with
# significant idvs.

log_2 <- glm(click ~ C15+C18+banner_pos+device_conn_type+site_category, data = train,
family = binomial)
summary(log_2)
```

```
##
## Call:
## glm(formula = click ~ C15 + C18 + banner_pos + device_conn_type +
##     site_category, family = binomial, data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.2467  -0.5936  -0.5272  -0.4805   2.8131
##
## Coefficients:
##                          Estimate Std. Error z value Pr(>|z|)
## (Intercept)            -3.760523   0.362772 -10.366  < 2e-16 ***
## C15300                  0.904446   0.059135  15.294  < 2e-16 ***
## C15320                  0.112164   0.051143   2.193 0.028299 *
## C15480                 -0.370777   0.764386  -0.485 0.627631
## C15728                 -0.830564   0.179128  -4.637 3.54e-06 ***
## C15768                  1.060580   1.072278   0.989 0.322619
## C151024                12.787069 324.743791   0.039 0.968591
## C181                   -0.432229   0.052278  -8.268  < 2e-16 ***
## C182                    0.823564   0.017615  46.753  < 2e-16 ***
## C183                   -0.004706   0.015160  -0.310 0.756228
## banner_pos1             0.118248   0.017863   6.620 3.60e-11 ***
## banner_pos2             1.301837   0.402969   3.231 0.001235 **
## banner_pos3           -10.453945 184.644055  -0.057 0.954850
## banner_pos4             0.205884   0.506434   0.407 0.684348
## banner_pos5            -0.144041   0.306706  -0.470 0.638614
## banner_pos7             1.991736   0.242729   8.206 2.29e-16 ***
## device_conn_type2      -0.192687   0.022920  -8.407  < 2e-16 ***
## device_conn_type3      -0.654119   0.060560 -10.801  < 2e-16 ***
## device_conn_type5      -1.237855   0.308219  -4.016 5.92e-05 ***
## site_category28905ebd   2.194044   0.359276   6.107 1.02e-09 ***
## site_category335d28a8   1.016136   0.378582   2.684 0.007274 **
## site_category3e814130   1.842672   0.359776   5.122 3.03e-07 ***
## site_category42a36e14   2.660042   0.952198   2.794 0.005213 **
## site_category50e219e0   1.745174   0.359309   4.857 1.19e-06 ***
## site_category5378d028  -8.917704 229.628750  -0.039 0.969022
```

```
## site_category70fb0e29   1.451910   0.433246   3.351 0.000805 ***
## site_category72722551   0.566273   0.436412   1.298 0.194437
## site_category75fa27f6   1.628101   0.379877   4.286 1.82e-05 ***
## site_category76b2941d   0.394970   0.422489   0.935 0.349859
## site_category8fd0aea4  -8.912998 324.743895  -0.027 0.978104
## site_categorya818d37a -10.243035  45.218328  -0.227 0.820794
## site_categorybcf865d9  -8.819539 132.460698  -0.067 0.946914
## site_categoryc0dd3be3   0.971297   0.471689   2.059 0.039476 *
## site_categorydedf689d   1.533836   0.439887   3.487 0.000489 ***
## site_categorye787de0e   2.164682   0.898314   2.410 0.015965 *
## site_categoryf028772b   1.822709   0.358945   5.078 3.82e-07 ***
## site_categoryf66779e6  -0.091571   0.384926  -0.238 0.811964
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 188924  on 210000  degrees of freedom
## Residual deviance: 181576  on 209964  degrees of freedom
## AIC: 181650
##
## Number of Fisher Scoring iterations: 11
```

```
# Prediction- . Test$scr is the predicted variable of click. We use .16 as the cutoff,
# as >.16 implies a click will be made.the scale is given on the scale of the response
# variable of the glm object

test$scr <- predict(log_2, test, type = "response")

a <- data.frame(test$click, test$scr)

#rounding up the predicted probability (cutoff=0.16)
test$round <- ifelse((test$scr > 0.16), 1, 0)

test$tpr <- ifelse(test$round == 1 , (ifelse( (test$round == test$click), 1, 0)), 0)

#calculating false positive rate
test$fpr <- ifelse(test$round ==1 , ifelse( (test$round != test$click), 1, 0), 0)

#calculating true negative rate i.e correctly identified 0's
test$tnr <-  ifelse(test$round ==0 , ifelse( (test$round == test$click), 1, 0), 0)

#True-positives
TP <- sum(test$tpr)

#True-negatives
TN <- sum(test$tnr)

#Calculating accuracy
acc <- (TP + TN) /nrow(test)
acc
```

```
## [1] 0.6629921
```

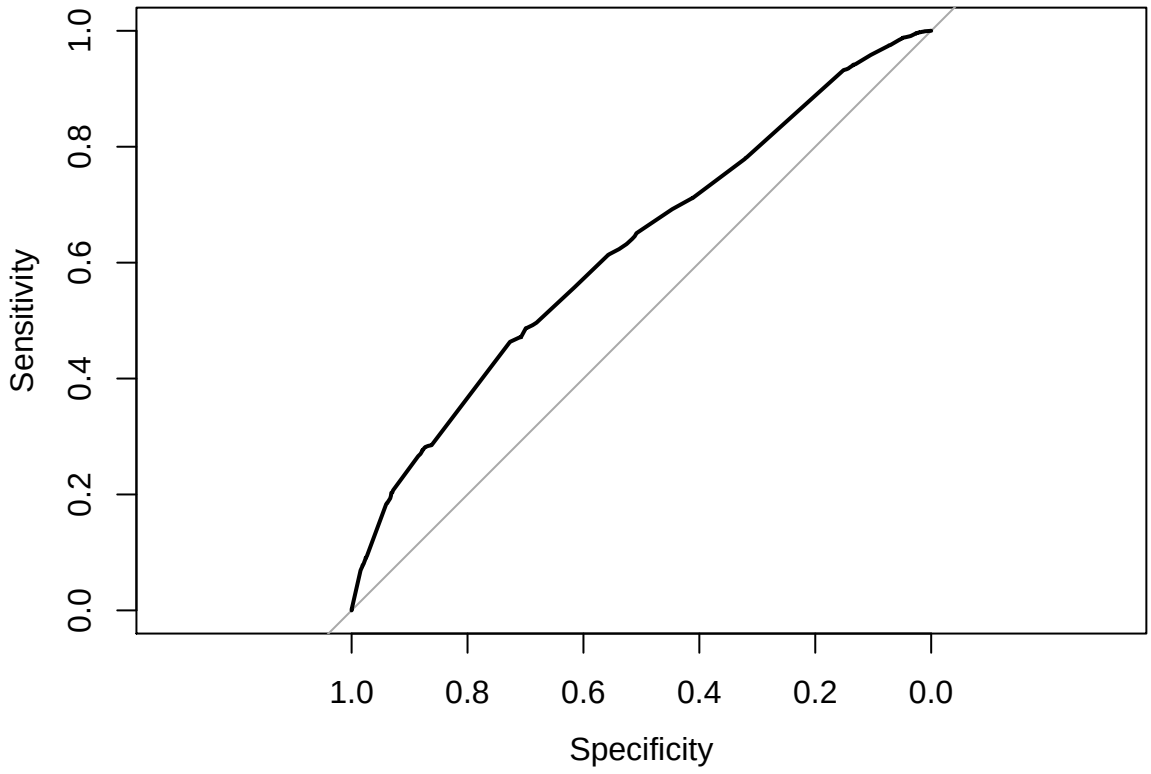
```r
# Accuracy of 66% implies it is a good model

#plotting roc curve

library(pROC)
```

```
## Type 'citation("pROC")' for a citation.
##
## Attaching package: 'pROC'
##
## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```r
roc_1 <- roc (click ~ scr, data = test)
roc_1
```

```
##
## Call:
## roc.formula(formula = click ~ scr, data = test)
##
## Data: scr in 175098 controls (click 0) < 34903 cases (click 1).
## Area under the curve: 0.6226
```

```r
plot(roc_1)
```



```
##
## Call:
## roc.formula(formula = click ~ scr, data = test)
##
## Data: scr in 175098 controls (click 0) < 34903 cases (click 1).
## Area under the curve: 0.6226
```

```
#Specificity is the % of false positives and sensitivity the % true positive
#The plot shows a straight line as no predicted values and our test as the curved line
# We can now set the specificity of our own test. Considering the low cost of mobile advertising
# This can be relatively high around .7.

#----OUTCOME-----
# The objective of the project was to build a predictive model which helps to predict
# whether a mobile ad will be clicked and accordingly target those who have a high porobability
# to click. Targetting the people who have a higher frequency to click will obviously help to
# increase the overall clickthrough rate and hence conversion. It will also help to increase
# efficiency since we will be paying for clicks which has a higher conversion chance. This will
# thus help to improve performance in online advertising and marketing. The dataset includes 36
# hours of click data containing metrics related to the app, site, device, time of activity etc.
#
# The model is built on 80% train dataset and validated against the 20% test dataset.
# This is a good learning exercise on the side of predictive modeling and has real world
# applications. Models can be built on similar lines to predict frauds, chances of default on
# insurance premium, credit going bad, customer churn, propensity to buy etc. Predictive modeling
# helps to target more relevant customers and improve efficiency.
```