

678 Midterm Project Main

Qianhui Rong

11/12/2018

```
#Library
library(MASS)
library(dplyr)

library(magrittr)
library(ggplot2)
library(cluster)
library(gridExtra)

library(lme4)

library(party)

library(MLmetrics)

library(logistf)
library(brglm)

library(merTools)

#Load data
data_whole <- read.table(file = 'content_train.tsv', sep = '\t', header =
TRUE)

#Add two columns: total_spend and total_no
data_whole%<>% mutate(total_spend=express.total.spend+metro.total.spend+
superstore.total.spend+extra.total.spend+fandf.total.spend+
                        petrol.total.spend+direct.total.spend)
data_whole%<>% mutate(total_no=express.no.transactions+metro.no.transactions+
                        superstore.no.transactions+extra.no.transactions+
                        fandf.no.transactions+
                        petrol.no.transactions+direct.no.transactions)

#Seperate the data into train set and test set (80:20 split)
#Random generate 100000*0.2 test set
set.seed(2018)
test_index <- sort(sample(nrow(data_whole), nrow(data_whole)*.2))
data_test <- data_whole[test_index,]
data_train <- data_whole[-test_index,]
```

A. Abstract

This analysis project is conducted based on the Tesco purchase history data. The goal is to build an appropriate model to predict the advertisements' click-through-rate for each of the customers in Tesco's database. This has been done with multinomial model, logistic model and multilevel mixed effect model. A problem named "seperation problem" or "rare event problem" has arose during modeling process. Some methods have been used to tackle this problem, but their performance is not up to the expectation, which leaves a wide exploring space for future analysis on this data.

B. Introduction

Click-through_rate(CTR) is the percentage rate at which people click on a particular ad when online. A critical step to improve CTR is to target the right group of customers for each kind of advertisement. Therefore, exploring the relation between features of customers (for example, demographic characteristics and purchase preferences) and whether they have clicked through this advertisement is important.

The most common and efficient method is logistic regression, predicting the probability of a certain group of people clicking a link. Also, there are some machine learning methods popular for this problem: Gradient Boosting Decision Tree, Factorization Mechines, etc.

C. Method

1. Data source

The data is obtained from <https://www.kaggle.com/linkonabe/tesco-marketing-content>. It's shared by Tesco company, which is a supermarket market leader in the UK.

The company releases some marketing content cards each year, and in this dataset, we have 9 of them marked as content_1 to content_9. For each content card, 1 means the customer clicked on the card, 0 means the customer viewed the card but didn't click, NA means the user was never shown the card.

The dataset also shares its customers' purchase history in different kinds of shop. There are seven kinds of store:

- Tesco Express: neighbourhood convenience shops
- Tesco Metro: located in city centres beside railway stations
- Tesco Superstore: standard large supermarkets
- Tesco Extra: larger, mainly out-of-town hypermarkets
- Tesco F&F: online store selling Tesco's own clothes brand

-Tesco Direct: online store selling groceries, homewares, electronics, etc.

-Tesco Petrol: grocery store in petrol stations

Some customers' demographic features are also provided in the data: gender and county. Affluency is a more informative variable than county. It's a broad categorisation of how affluent the customer is based on their postcode.

2. Model used

Models selected to explain the content_1 variable are:

-- Multinomial Model: using content_1 without conversion or pre-processing, and keep its three levels(0,1,NA);

For the following two models, preprocessing is necessary. I've first tried out converting NA to 0 and leave 1 as 1, so content_1 becomes "absence of click" vs. "presence of click", but this makes the "seperation" problem more severe(in appendix). Then I switched to eliminating the NA directly in both training set and testing set, in order to make "ones" in content_1 to have a higher proportion, compared to "zeros" in content_1(in main result section).

-- Logistic Model

-- Multilevel Mixed Effect Model

D. Results

Part I: EDA

EDA on Affluency

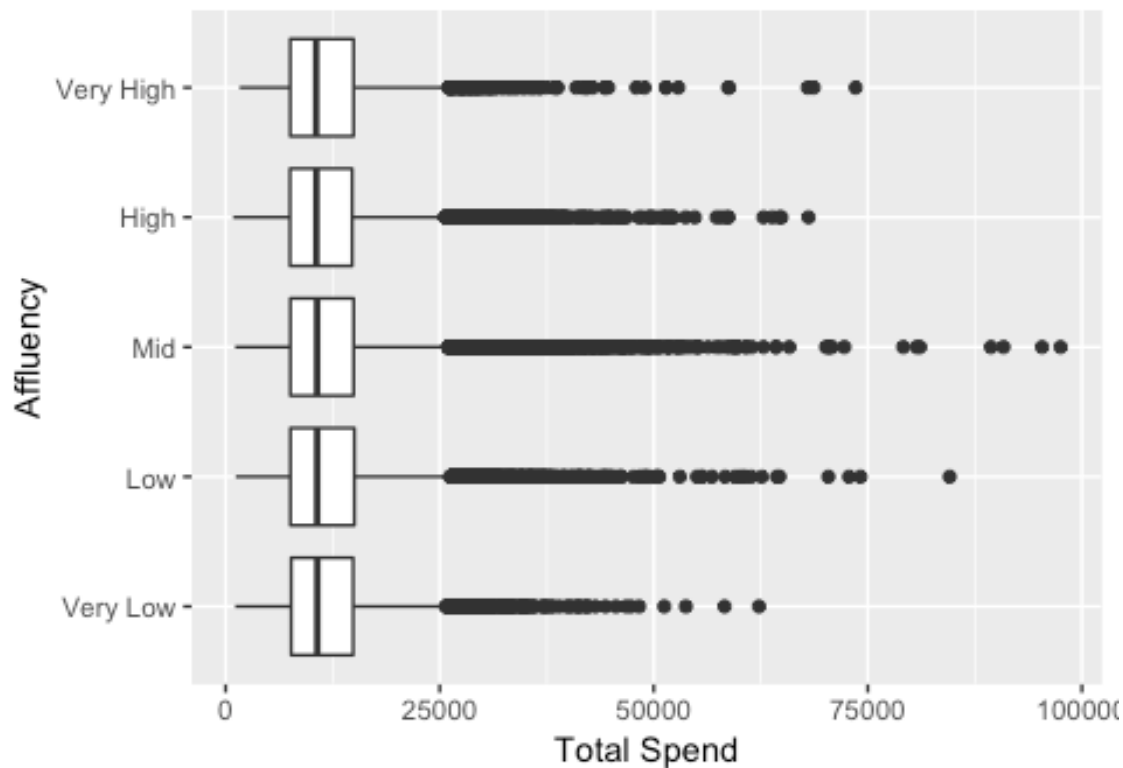
About affluency, I'm expecting a higher amount of transactions and a higher number of transactions in regions of higher degree of affluency.

1. Total amount of transactions vs. Affluency

```
data_train %>% group_by(affluency) %>%  
  ggplot(aes(x=as.factor(affluency),y=total_spend))+ geom_boxplot()+  
  scale_x_discrete(limits=c("Very Low","Low","Mid","High","Very High"))+  
  xlab("Affluency") +  
  ylab("Total Spend") +  
  coord_flip()+  
  labs(title="Total Spend Amount vs. Affluency",subtitle="No apparent  
difference in total spend range for each affluency class")
```

Total Spend Amount vs. Affluency

No apparent difference in total spend range for each affluency



#There are some points located in higher spend amount for Mid class affluency, but these points are trivial to the whole data size.

#Double check if those five bar charts are that close

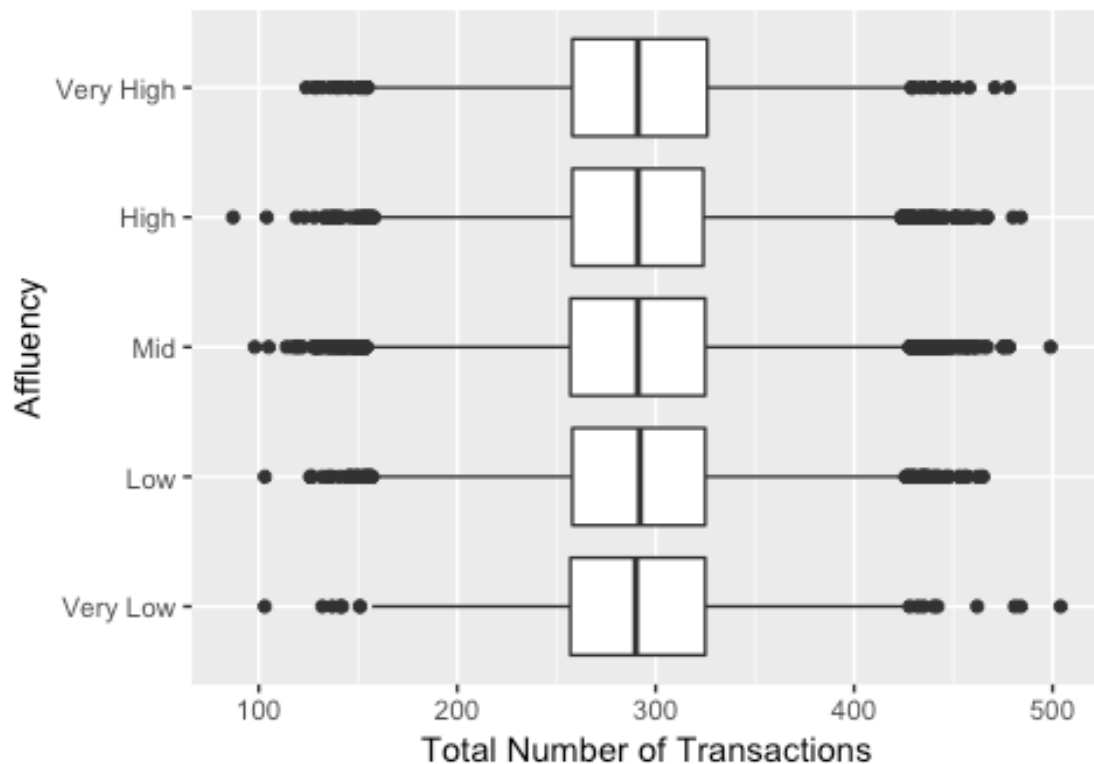
```
#c(236496799/19772,242749658/20133,601268557/50105,60063877/5009,59859247/4981)
```

2. Total Number of transactions vs. Affluency

```
data_train %>% group_by(affluency) %>%
  ggplot(aes(x=as.factor(affluency),y=total_no))+ geom_boxplot()+
  scale_x_discrete(limits=c("Very Low","Low","Mid","High","Very High"))+
  xlab("Affluency") +
  ylab("Total Number of Transactions") +
  coord_flip()+
  labs(title="Total Number of Transactions vs. Affluency",subtitle="No
apparent difference in number of transactions range for each affluency
class")
```

Total Number of Transactions vs. Affluency

No apparent difference in number of transactions range for each



3. Clicking in Different Content (1~9) vs. Affluency

```
#Group data
plot_mat <- matrix(0,nrow =45,ncol=3)
colnames(plot_mat) <- c("Content","Affluency","Count")
plot_mat[,1] <-
c(rep("content_1",5),rep("content_2",5),rep("content_3",5),rep("content_4",5),
,
rep("content_5",5),rep("content_6",5),rep("content_7",5),rep("content_8",5),
rep("content_9",5))
plot_mat[,2] <- rep(c("High","Low","Mid","Very High","Very Low"),9)

mat<- matrix(0,nrow = 9,ncol=5)
for (i in 1:9){
  vh <- 0
  h <- 0
  m <- 0
  l <- 0
  vl <- 0
  vh <- length(which(data_train[which(data_train[,1+i]==1),26]=="Very
High")==TRUE)
```

```

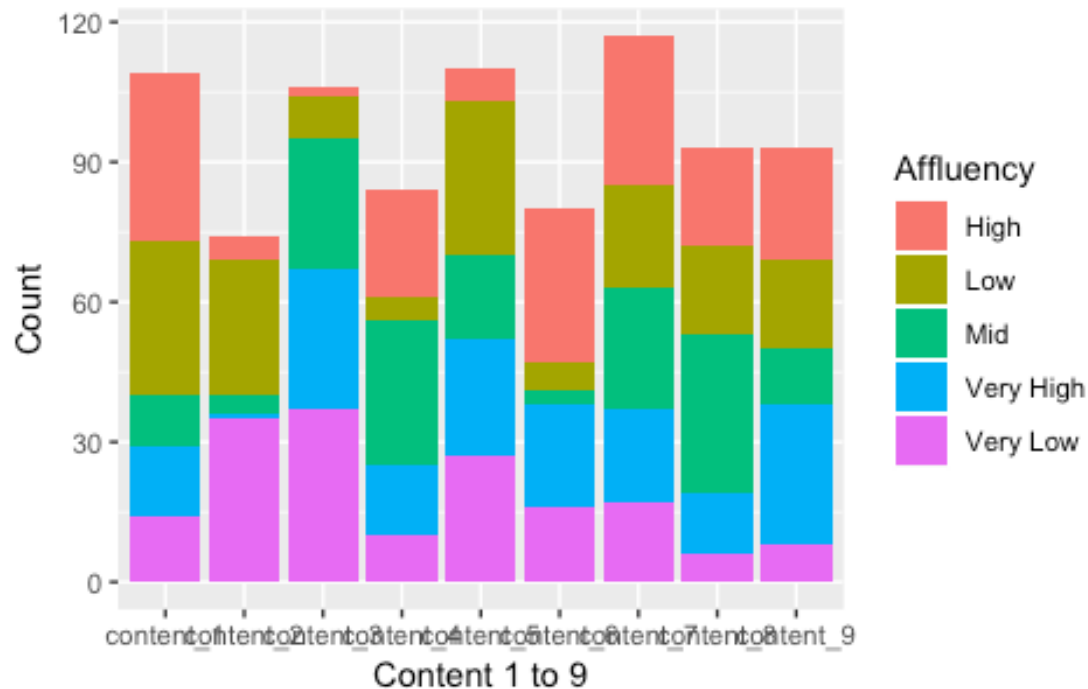
h <- length(which(data_train[which(data_train[,1+i]==1),26]=="High")==TRUE)
m <- length(which(data_train[which(data_train[,1+i]==1),26]=="Mid")==TRUE)
l <- length(which(data_train[which(data_train[,1+i]==1),26]=="Low")==TRUE)
vl <- length(which(data_train[which(data_train[,1+i]==1),26]=="Very
Low")==TRUE)
mat[i,] <- c(h,l,m,vh,vl)
}
mat <- t(mat)
mat <- as.data.frame(mat)
rownames(mat) <- c("High","Low","Mid","Very High","Very Low")
colnames(mat) <- c("1","2","3","4","5","6","7","8","9")
plot_mat[,3] <-
c(mat$`1`,mat$`2`,mat$`3`,mat$`4`,mat$`5`,mat$`6`,mat$`7`,mat$`8`,mat$`9`)
plot_mat <- as.data.frame(plot_mat)

#Plot
plot_mat$Count <- as.numeric(plot_mat$Count)
ggplot(data = plot_mat)+
  aes(x=as.factor(Content),y=Count,fill=as.factor(Affluency))+
  geom_col(position = 'stack')+
  xlab("Content 1 to 9")+
  scale_fill_discrete(name="Affluency")+
  labs(title="Count number of clicking each content vs.
Affluency",subtitle=paste(strwrap("The most clicked contents are content 7
and content1, and the rest of them don't have an evident difference on count.
People from different affluency regions may have some preferences towards
certain contents, but hard to make sure on it.", 90), collapse="\n"))

```

Count number of clicking each content vs. Affluency

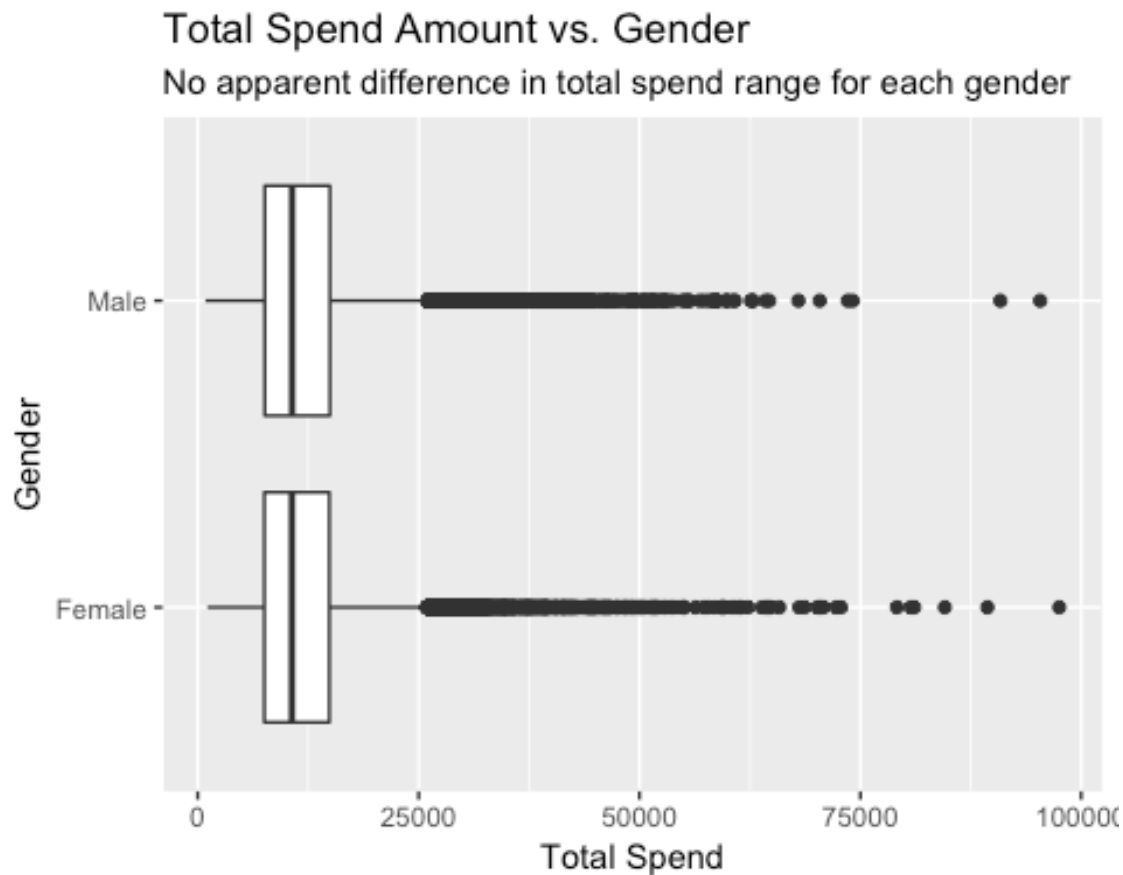
The most clicked contents are content 7 and content 1, and the res evident difference on count. People from different affluency region preferences towards certain contents, but hard to make sure on it.



EDA on Gender

1. Total amount of transactions vs. Gender

```
data_train %>% group_by(gender) %>%  
  ggplot(aes(x=as.factor(gender), y=total_spend)) + geom_boxplot() +  
  xlab("Gender") +  
  ylab("Total Spend") +  
  coord_flip() +  
  labs(title="Total Spend Amount vs. Gender", subtitle="No apparent difference  
in total spend range for each gender")
```

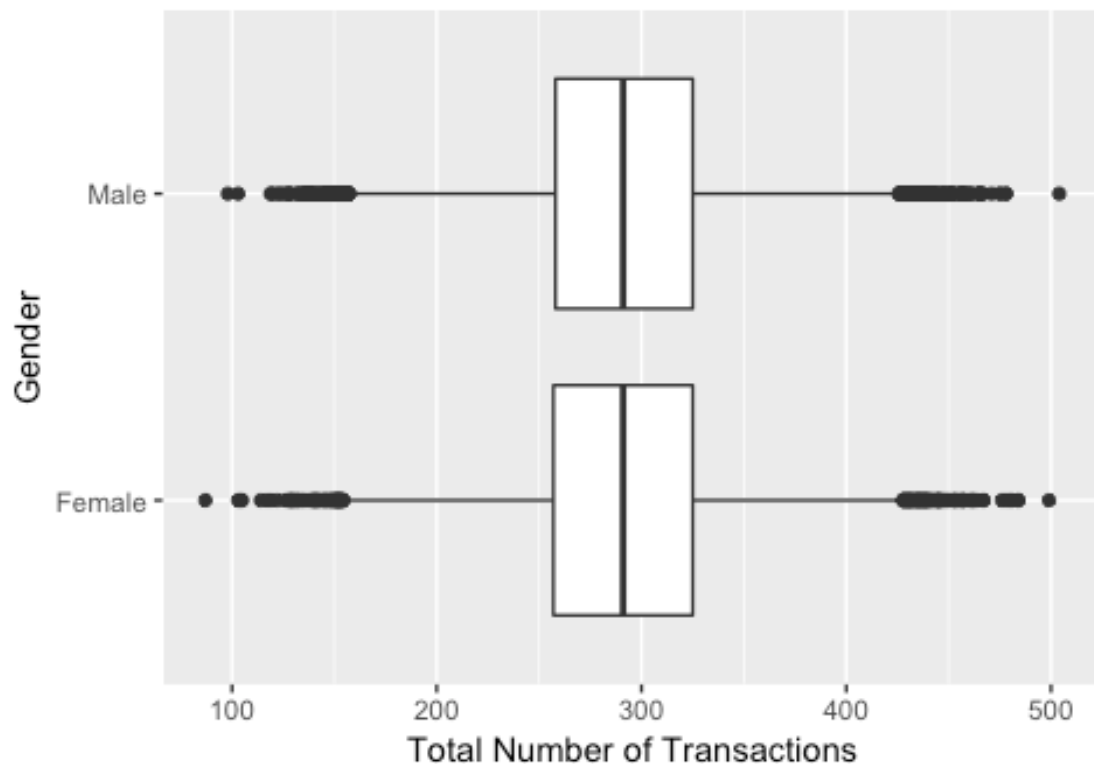


2. Total Number of transactions vs. Gender

```
data_train %>% group_by(gender) %>%
  ggplot(aes(x=as.factor(gender),y=total_no))+ geom_boxplot()+
  xlab("Gender") +
  ylab("Total Number of Transactions") +
  coord_flip()+
  labs(title="Total Number of Transactions vs. Gender",subtitle="No apparent
difference in number of transactions range for each gender")
```


Total Number of Transactions vs. Gender

No apparent difference in number of transactions range for each



3. Clicking in Different Content (1~9) vs. Gender

```
#Group data
plot_mat <- matrix(0,nrow =18,ncol=3)
colnames(plot_mat) <- c("Content","Gender","Count")
plot_mat[,1] <-
c(rep("content_1",2),rep("content_2",2),rep("content_3",2),rep("content_4",2),
,
rep("content_5",2),rep("content_6",2),rep("content_7",2),rep("content_8",2),
  rep("content_9",2))
plot_mat[,2] <- rep(c("Male","Female"),9)

mat<- matrix(0,nrow = 9,ncol=2)
for (i in 1:9){
  f <- 0
  m <- 0
  m <- length(which(data_train[which(data_train[,1+i]==1),25]=="Male")==TRUE)
  f <-
length(which(data_train[which(data_train[,1+i]==1),25]=="Female")==TRUE)
  mat[i,] <- c(m,f)
}
```

```

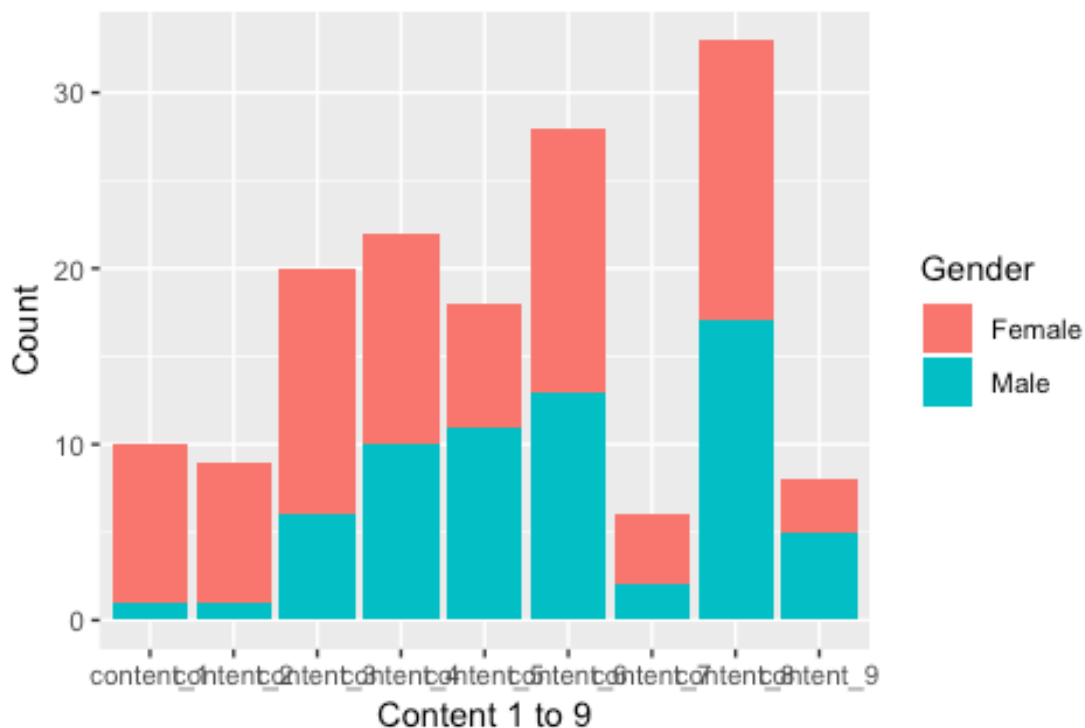
mat <- t(mat)
mat <- as.data.frame(mat)
rownames(mat) <- c("Male", "Female")
colnames(mat) <- c("1", "2", "3", "4", "5", "6", "7", "8", "9")
plot_mat[,3] <-
c(mat$`1`,mat$`2`,mat$`3`,mat$`4`,mat$`5`,mat$`6`,mat$`7`,mat$`8`,mat$`9`)
plot_mat <- as.data.frame(plot_mat)

#Plot
plot_mat$Count <- as.numeric(plot_mat$Count)
ggplot(data = plot_mat)+
  aes(x=as.factor(Content),y=Count,fill=as.factor(Gender))+
  geom_col(position = 'stack')+
  xlab("Content 1 to 9")+
  scale_fill_discrete(name="Gender")+
  labs(title="Count number of clicking each content vs. Gender",
        subtitle=paste(strwrap("Certain contents are male dominant(5,9) but most of
        them are female dominant(1,2,3,7). We can expect contents have gender-
        featured.", 90), collapse="\n"))

```

Count number of clicking each content vs. Gender

Certain contents are male dominant(5,9) but most of them are female dominant(1,2,3,7). We can expect contents have gender-featured.



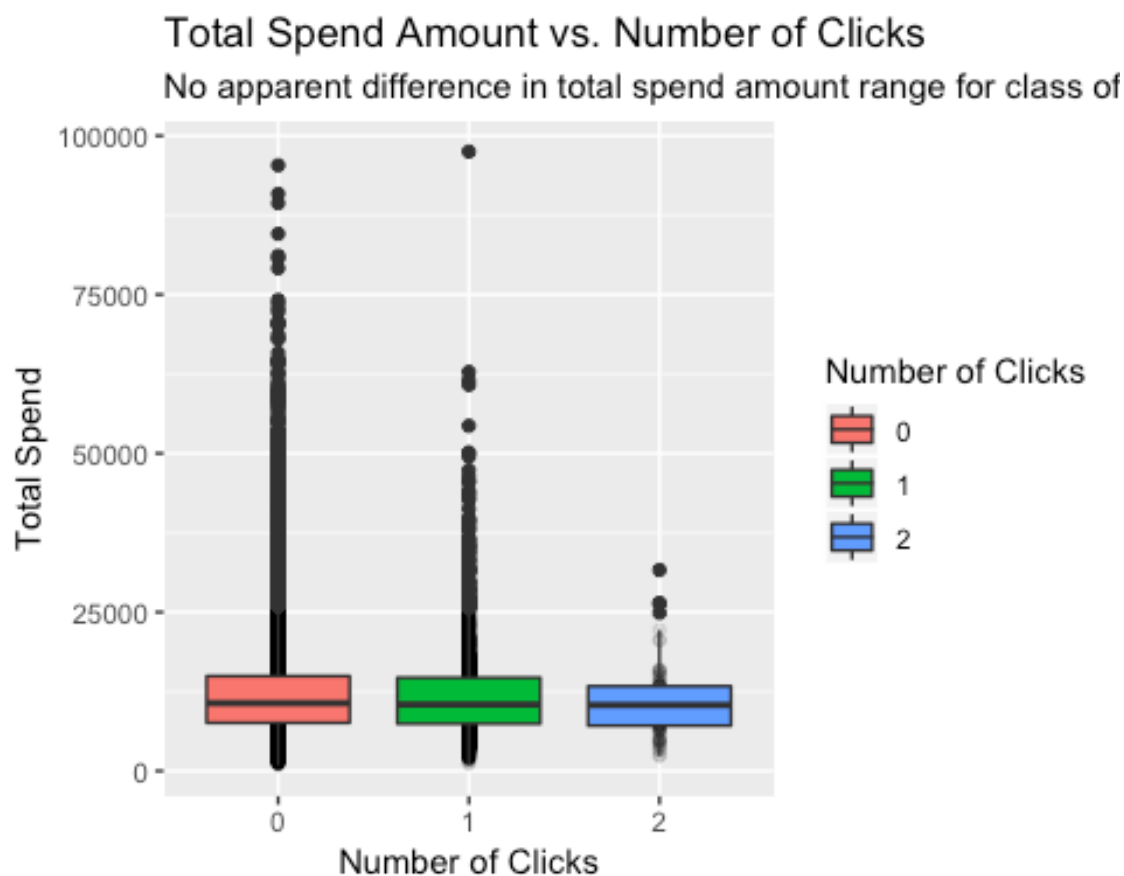
EDA on Contents(Clicked/Ignored/Not Informed)

```
#Calculate the number of click for each customer
for (i in 1:80000){data_train[i,30]<-
sum(as.numeric(data_train[i,2:10]),na.rm = TRUE)}
colnames(data_train)[30] <- "No_Click"
```

1. Plot of amount of transactions and contents

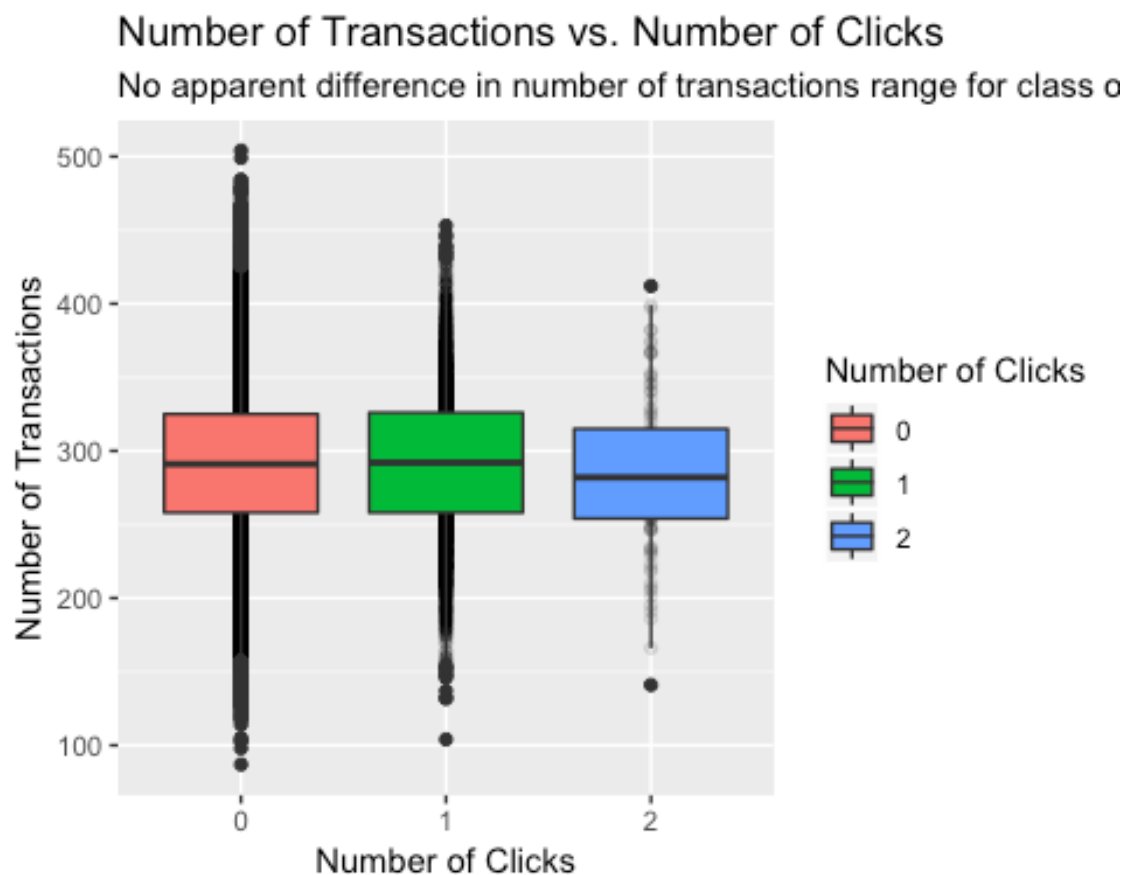
#Maximum number of clicks for a customer is two, and minimum number of clicks for a customer is zero.

```
ggplot(data=data_train)+
  aes(x=as.factor(No_Click),y=total_spend,fill=as.factor(No_Click))+
  geom_point(alpha=0.1)+
  geom_boxplot()+
  xlab("Number of Clicks")+
  ylab("Total Spend")+
  labs(title="Total Spend Amount vs. Number of Clicks",subtitle="No apparent difference in total spend amount range for class of number of clicks")+
  guides(fill=guide_legend(title="Number of Clicks"))
```



2. Plot of number of transactions and contents

```
ggplot(data=data_train)+  
  aes(x=as.factor(No_Click),y=total_no,fill=as.factor(No_Click))+  
  geom_point(alpha=0.1)+  
  geom_boxplot()+  
  xlab("Number of Clicks")+  
  ylab("Number of Transactions")+  
  labs(title="Number of Transactions vs. Number of Clicks",subtitle="No  
apparent difference in number of transactions range for class of  
clicks")+  
  guides(fill=guide_legend(title="Number of Clicks"))
```



Part II: Modeling

In this Modeling section, I'll use the data deleted content_1=NA observations.

i. Multinomial Model for Contents

The content_1 is originally a three-level response variable, and the first relevant model that came to my mind is multinomial model.

I fit a multinomial model taking content one's clicking history (0, 1, NA: three levels) as response variable. In this case, the three possible responses are: not informed(NA) / informed but didn't click(0) / clicked(1) have an order.

To avoid problems, convert "NA" to "0", "0" to "1", "1" to "2". "0" means the customers are not informed, "1" means the customers didn't clicked, "2" means the customers have clicked.

```
for (i in 1:9){
data_train[,1+i]<- ifelse(is.na(data_train[,1+i])==TRUE,0,
                           ifelse(data_train[,1+i]=="0",1,
                                   ifelse(data_train[,1+i]=="1",2,NA)))
data_train[,1+i] <- as.factor(data_train[,1+i])
}

#Proceed the same value modification for test data
for (i in 1:9){
data_test[,1+i]<- ifelse(is.na(data_test[,1+i])==TRUE,0,
                        ifelse(data_test[,1+i]=="0",1,
                                ifelse(data_test[,1+i]=="1",2,NA)))
data_test[,1+i] <- as.factor(data_test[,1+i])
}

#Model Fitting
ord_logit <-
polr(content_1~as.factor(gender)+as.factor(affluency)+log(express.total.spend
+1)+log(express.no.transactions+1)+log(metro.total.spend+1)+log(metro.no.trans
actions+1)+log(superstore.total.spend+1)+log(superstore.no.transactions+1)+l
og(extra.total.spend+1)+log(extra.no.transactions+1)+log(fandf.total.spend+1)
+log(fandf.no.transactions+1)+log(petrol.total.spend+1)+log(petrol.no.transac
tions+1)+log(direct.total.spend+1)+log(direct.no.transactions+1),data=data_tr
ain,Hess = T)
summary(ord_logit)

## Call:
## polr(formula = content_1 ~ as.factor(gender) + as.factor(affluency) +
##       log(express.total.spend + 1) + log(express.no.transactions +
##       1) + log(metro.total.spend + 1) + log(metro.no.transactions +
```

```
##      1) + log(superstore.total.spend + 1) + log(superstore.no.transactions
+
##      1) + log(extra.total.spend + 1) + log(extra.no.transactions +
##      1) + log(fandf.total.spend + 1) + log(fandf.no.transactions +
##      1) + log(petrol.total.spend + 1) + log(petrol.no.transactions +
##      1) + log(direct.total.spend + 1) + log(direct.no.transactions +
##      1), data = data_train, Hess = T)
##
## Coefficients:
##
##              Value Std. Error t value
## as.factor(gender)Male      -0.009182   0.014160  -0.6484
## as.factor(affluency)Low     -0.025490   0.022410  -1.1375
## as.factor(affluency)Mid     -0.041716   0.018812  -2.2175
## as.factor(affluency)Very High -0.051694   0.035447  -1.4583
## as.factor(affluency)Very Low  0.047587   0.035519   1.3398
## log(express.total.spend + 1)   0.002352   0.006007   0.3916
## log(express.no.transactions + 1) -0.015385   0.014972  -1.0276
## log(metro.total.spend + 1)    0.004984   0.005947   0.8380
## log(metro.no.transactions + 1)  0.001624   0.014995   0.1083
## log(superstore.total.spend + 1) 0.004705   0.005863   0.8024
## log(superstore.no.transactions + 1) -0.003470   0.015277  -0.2271
## log(extra.total.spend + 1)    0.006325   0.005874   1.0768
## log(extra.no.transactions + 1) -0.017368   0.015246  -1.1392
## log(fandf.total.spend + 1)    0.001960   0.005772   0.3396
## log(fandf.no.transactions + 1) -0.001951   0.011345  -0.1719
## log(petrol.total.spend + 1)   -0.001046   0.006002  -0.1743
## log(petrol.no.transactions + 1) -0.002153   0.011115  -0.1937
## log(direct.total.spend + 1)   -0.008313   0.005524  -1.5049
## log(direct.no.transactions + 1)  0.015327   0.011846   1.2939
##
## Intercepts:
##      Value Std. Error t value
## 0|1  0.1053  0.1127   0.9343
## 1|2  5.2436  0.1231  42.5889
##
## Residual Deviance: 114792.93
## AIC: 114834.93
```

#Residual Plots

```
resid_multinom <- model.matrix(~factor(content_1)-1, data=data_train)-
fitted(ord_logit)
par(mfrow= c(2,2))
for (i in 1:3) {
  binnedplot(fitted(ord_logit)[,i], resid_multinom[,i], cex.main=1.3,
main="Binned residual plot", nclass = 20) }
```

#LogLoss

```

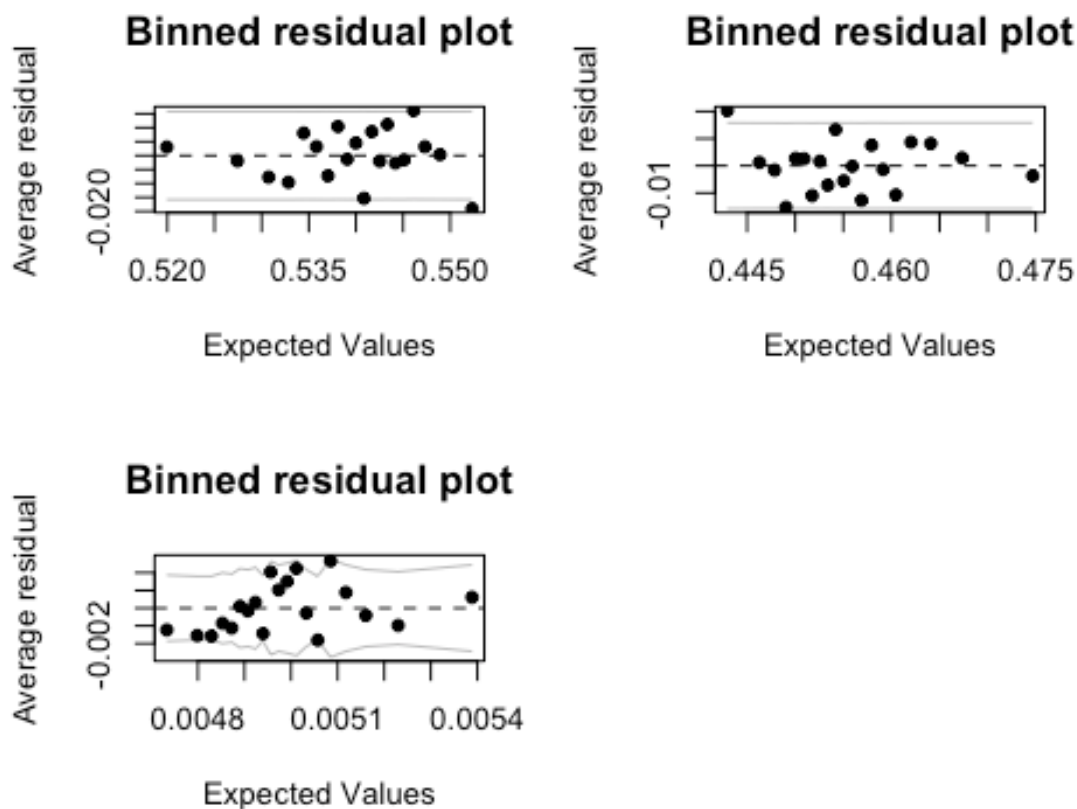
LogLoss(y_pred=ord_logit$fitted.values,
y_true=as.numeric(data_train$content_1))

## [1] 3.059993

#Predict the test data
data_test[,-c(1:10,28,29)] -> data_test_multinom
predict_multinom_prob <- predict(ord_logit, newdata = data_test_multinom,
"probs")
predict_multinom_class <- predict(ord_logit, newdata = data_test_multinom)
#Calculate the probability of mis-prediction
mean(as.character(predict_multinom_class) !=
as.character(data_test$content_1))

## [1] 0.45865

```



For this multinomial model:

1. The residual plots for three levels look acceptable, with most of the points located between two curves.

2. Prediction based on this model is a little bit better than random guess, about 55% of successful prediction rate. But the problem is that the model automatically take all predictions equal to 0.

To simplify the problem, I then narrow the response's three levels to two levels by deleting NA observations in order to increase the ones proportion compared to the zeros.

"0" means the customers didn't clicked, "1" means the customers have clicked.

```
#Reload original data
data_whole <- read.table(file = 'content_train.tsv', sep = '\t', header =
TRUE)
data_whole%<>% mutate(total_spend=express.total.spend+metro.total.spend+
superstore.total.spend+extra.total.spend+fandf.total.spend+
petrol.total.spend+direct.total.spend)
data_whole%<>% mutate(total_no=express.no.transactions+metro.no.transactions+
superstore.no.transactions+extra.no.transactions+
fandf.no.transactions+
petrol.no.transactions+direct.no.transactions)
set.seed(2018)
test_index <- sort(sample(nrow(data_whole), nrow(data_whole)*.2))
data_test <- data_whole[test_index,]
data_train <- data_whole[-test_index,]

#Preprocessing the data: deleting all NA observations
data_train_nona <- data_train %>%
  filter(is.na(content_1)==FALSE)

data_test_nona <- data_test %>%
  filter(is.na(content_1)==FALSE)

data_train_nona$content_1 <- as.factor(data_train_nona$content_1)

nrow(subset(data_train_nona,data_train_nona$content_1=="0"))
## [1] 36450

nrow(subset(data_train_nona,data_train_nona$content_1=="1"))
## [1] 399

#Now for content 1, there are 36450 ones and 399 twos.
```


ii. Logistic Model

```
simp_logit <-  
glm(content_1~as.factor(gender)+as.factor(affluency)+log(express.total.spend+  
1)+log(express.no.transactions+1)+log(metro.total.spend+1)+log(metro.no.trans  
actions+1)+log(superstore.total.spend+1)+log(superstore.no.transactions+1)+lo  
g(extra.total.spend+1)+log(extra.no.transactions+1)+log(fandf.total.spend+1)+  
log(fandf.no.transactions+1)+log(petrol.total.spend+1)+log(petrol.no.transact  
ions+1)+log(direct.total.spend+1)+log(direct.no.transactions+1),data=data_tra  
in_nona,family = binomial)  
summary(simp_logit)  
  
##  
## Call:  
## glm(formula = content_1 ~ as.factor(gender) + as.factor(affluency) +  
##      log(express.total.spend + 1) + log(express.no.transactions +  
##      1) + log(metro.total.spend + 1) + log(metro.no.transactions +  
##      1) + log(superstore.total.spend + 1) + log(superstore.no.transactions  
##      +  
##      1) + log(extra.total.spend + 1) + log(extra.no.transactions +  
##      1) + log(fandf.total.spend + 1) + log(fandf.no.transactions +  
##      1) + log(petrol.total.spend + 1) + log(petrol.no.transactions +  
##      1) + log(direct.total.spend + 1) + log(direct.no.transactions +  
##      1), family = binomial, data = data_train_nona)  
##  
## Deviance Residuals:  
##      Min       1Q   Median       3Q      Max   
## -0.2752  -0.1772  -0.1278  -0.1064   3.3331   
##  
## Coefficients:  
##              Estimate Std. Error z value Pr(>|z|)   
## (Intercept)      -3.997e+00  7.995e-01  -4.999 5.76e-07   
## as.factor(gender)Male      -1.022e+00  1.135e-01  -9.003  < 2e-16   
## as.factor(affluency)Low     -1.541e-01  1.574e-01  -0.979  0.327   
## as.factor(affluency)Mid     -1.411e-01  1.293e-01  -1.091  0.275   
## as.factor(affluency)Very High  6.683e-02  2.358e-01   0.283  0.777   
## as.factor(affluency)Very Low  -1.137e-01  2.492e-01  -0.456  0.648   
## log(express.total.spend + 1)  7.209e-02  4.486e-02   1.607  0.108   
## log(express.no.transactions + 1) -4.331e-02  1.099e-01  -0.394  0.694   
## log(metro.total.spend + 1)    1.410e-02  4.261e-02   0.331  0.741   
## log(metro.no.transactions + 1)  7.861e-03  1.084e-01   0.073  0.942   
## log(superstore.total.spend + 1)  4.233e-02  4.289e-02   0.987  0.324   
## log(superstore.no.transactions + 1) -1.346e-02  1.121e-01  -0.120  0.904   
## log(extra.total.spend + 1)     5.914e-03  4.177e-02   0.142  0.887   
## log(extra.no.transactions + 1) -1.558e-01  1.017e-01  -1.532  0.126   
## log(fandf.total.spend + 1)    -9.066e-03  4.079e-02  -0.222  0.824   
## log(fandf.no.transactions + 1)  1.284e-05  7.994e-02   0.000  1.000   
## log(petrol.total.spend + 1)    -3.335e-02  4.242e-02  -0.786  0.432   
## log(petrol.no.transactions + 1)  7.001e-02  7.881e-02   0.888  0.374   
## log(direct.total.spend + 1)     1.239e-02  3.931e-02   0.315  0.753   
## log(direct.no.transactions + 1) -6.420e-02  8.394e-02  -0.765  0.444
```

```

##
## (Intercept) ***
## as.factor(gender)Male ***
## as.factor(affluency)Low
## as.factor(affluency)Mid
## as.factor(affluency)Very High
## as.factor(affluency)Very Low
## log(express.total.spend + 1)
## log(express.no.transactions + 1)
## log(metro.total.spend + 1)
## log(metro.no.transactions + 1)
## log(superstore.total.spend + 1)
## log(superstore.no.transactions + 1)
## log(extra.total.spend + 1)
## log(extra.no.transactions + 1)
## log(fandf.total.spend + 1)
## log(fandf.no.transactions + 1)
## log(petrol.total.spend + 1)
## log(petrol.no.transactions + 1)
## log(direct.total.spend + 1)
## log(direct.no.transactions + 1)
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##    Null deviance: 4405.1  on 36848  degrees of freedom
## Residual deviance: 4303.0  on 36829  degrees of freedom
## AIC: 4343
##
## Number of Fisher Scoring iterations: 8

#Evaluate the model
#Anova test to show if adding each variable makes sense
anova(simp_logit)

## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: content_1
##
## Terms added sequentially (first to last)
##
##
##
## Df Deviance Resid. Df Resid. Dev
## NULL 36848 4405.1
## as.factor(gender) 1 91.465 36847 4313.6

```

| | | | | |
|--|---|-------|-------|--------|
| ## as.factor(affluency) | 4 | 1.993 | 36843 | 4311.7 |
| ## log(express.total.spend + 1) | 1 | 2.610 | 36842 | 4309.0 |
| ## log(express.no.transactions + 1) | 1 | 0.145 | 36841 | 4308.9 |
| ## log(metro.total.spend + 1) | 1 | 0.189 | 36840 | 4308.7 |
| ## log(metro.no.transactions + 1) | 1 | 0.005 | 36839 | 4308.7 |
| ## log(superstore.total.spend + 1) | 1 | 1.086 | 36838 | 4307.6 |
| ## log(superstore.no.transactions + 1) | 1 | 0.017 | 36837 | 4307.6 |
| ## log(extra.total.spend + 1) | 1 | 0.503 | 36836 | 4307.1 |
| ## log(extra.no.transactions + 1) | 1 | 2.270 | 36835 | 4304.8 |
| ## log(fandf.total.spend + 1) | 1 | 0.132 | 36834 | 4304.7 |
| ## log(fandf.no.transactions + 1) | 1 | 0.000 | 36833 | 4304.7 |
| ## log(petrol.total.spend + 1) | 1 | 0.018 | 36832 | 4304.7 |
| ## log(petrol.no.transactions + 1) | 1 | 0.777 | 36831 | 4303.9 |
| ## log(direct.total.spend + 1) | 1 | 0.305 | 36830 | 4303.6 |
| ## log(direct.no.transactions + 1) | 1 | 0.587 | 36829 | 4303.0 |

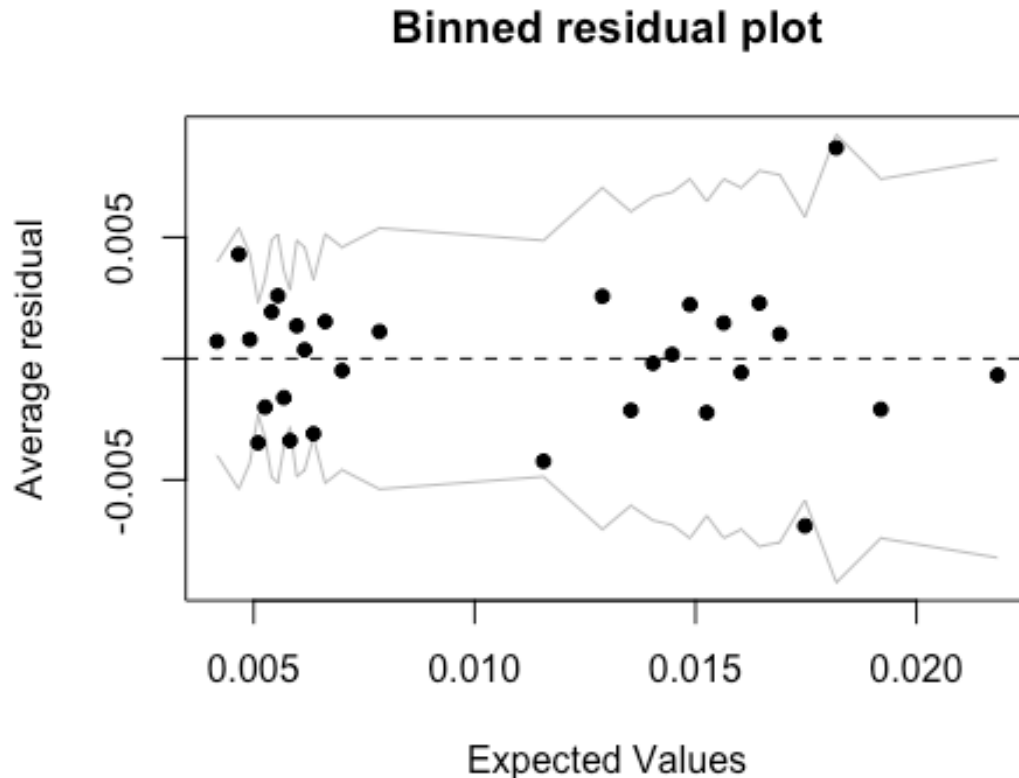
#LogLoss to see the difference between fitted values and actual values

```
LogLoss(y_pred=simp_logit$fitted.values,
y_true=as.numeric(data_train_nona$content_1))
```

```
[1] 4.709136
```

#Binned residual plot

```
binnedplot(x=simp_logit$fitted.values,y=resid(simp_logit,type="response"),nclass = 30)
```



#Error table

```
T_A <- mean(simp_logit$fitted.values<0.5 & data_train_nona$content_1==0)
T_B <- mean(simp_logit$fitted.values>0.5 & data_train_nona$content_1==0)
T_C <- mean(simp_logit$fitted.values<0.5 & data_train_nona$content_1==1)
T_D <- mean(simp_logit$fitted.values>0.5 & data_train_nona$content_1==1)
error_table <- matrix(c(T_A,T_B,T_C,T_D),byrow=TRUE,nrow = 2)
rownames(error_table) <- c("y=0","y=1")
colnames(error_table) <- c("y_hat=0","y_hat=1")
error_table
```

```
##          y_hat=0 y_hat=1
## y=0 0.98917203      0
## y=1 0.01082797      0
```

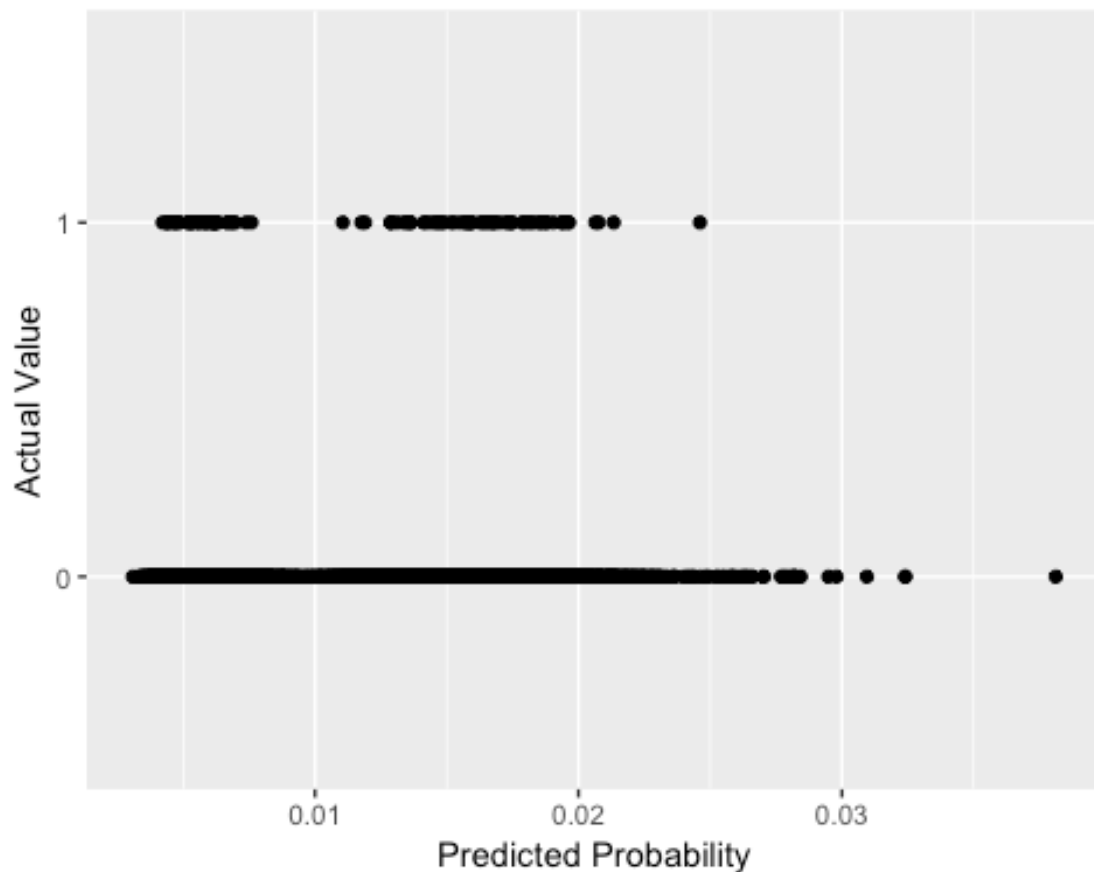
#Predictions

```
data_test[, -c(1:10,28,29)] %>% filter(is.na(data_test$content_1)==FALSE)->
data_test_glm
predict_glm_class <- predict(simp_logit, newdata = data_test_glm)
probs_glm <- exp(predict_glm_class)/(1+exp(predict_glm_class)) #probability
of 1
data_test_use <- data_test %>% filter(is.na(data_test$content_1)==FALSE)
ggplot(data_test_use)+
```

```

aes(x=probs_glm,y=as.factor(data_test_use$content_1))+
geom_point()+
xlab("Predicted Probability")+ylab("Actual Value")

```



For this logistic model:

1. Comparing the null deviance and residual deviance, we can see that the model has been improved by adding new variables.
2. From the ANOVA test, even though the deviance is not improving drastically, as I added each variable, the deviance does decrease (but of small amount).
3. LogLoss is a value that we want to minimize signifying that the model is more accurate. The LogLoss in this model is higher than that of multinomial model.
4. Binned residual plot here looks normal, with all points between lines and symmetrically distributed above and below zero.
5. The error table is telling a big problem, as the one I've observed from the multinomial model before. The model is having a really high successful prediction rate, 99.5%. Simply because that it is predicting all the test observations to be zero, because the original training dataset has only less than 0.05% which has $content_1 = 1$. The number of

customers who have clicked through is really low. I'll deal with this in the next section "Rare event problem".

6. The prediction plot also illustrate the last problem. We have the predicted probability not higher than 0.02, which we can approximate to a probability of zero, but there are a number of points of one.

To deal with logistic regression's "rare event problem": We only have 399 "ones" in data_train for content_1, which is relatively rare, about 1%. I'll try two most popular methods to deal with this kind of separation problem:

a. brglm: Bias Reduction in Binomial-Response Generalized Linear Models

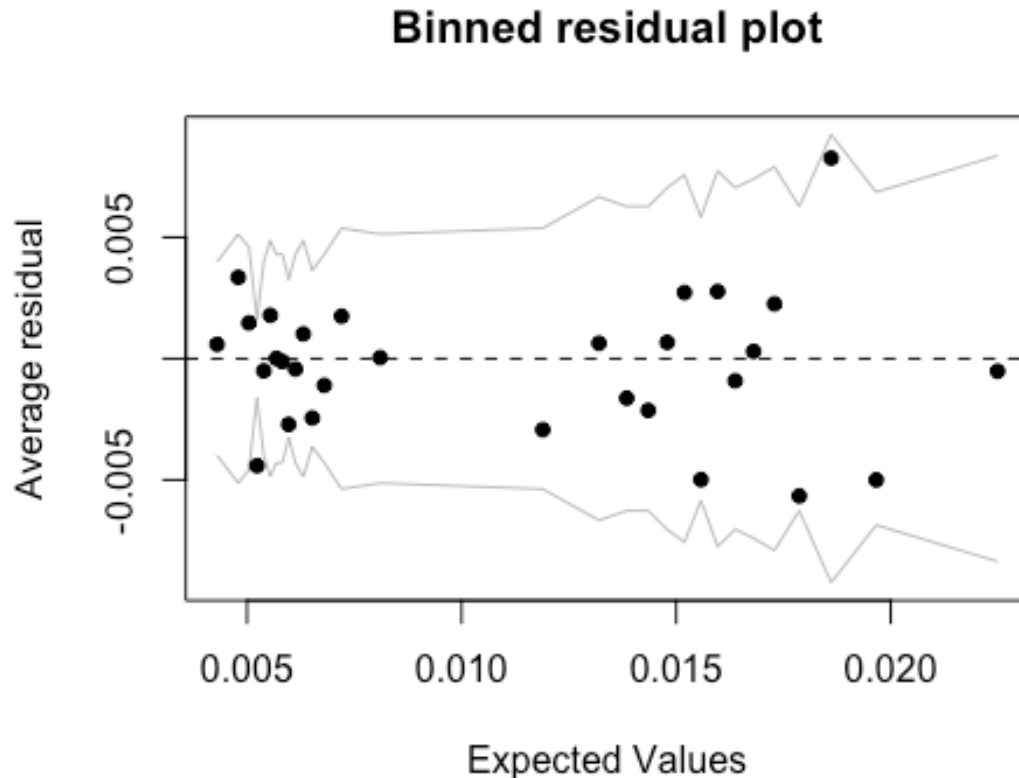
```
brglm_logit <-  
brglm(content_1~as.factor(gender)+as.factor(affluency)+log(express.total.spend+1)+log(express.no.transactions+1)+log(metro.total.spend+1)+log(metro.no.transactions+1)+log(superstore.total.spend+1)+log(superstore.no.transactions+1)+log(extra.total.spend+1)+log(extra.no.transactions+1)+log(fandf.total.spend+1)+log(fandf.no.transactions+1)+log(petrol.total.spend+1)+log(petrol.no.transactions+1)+log(direct.total.spend+1)+log(direct.no.transactions+1),data=data_train_nona,family = binomial)  
summary(brglm_logit)  
  
##  
## Call:  
## brglm(formula = content_1 ~ as.factor(gender) + as.factor(affluency) +  
##      log(express.total.spend + 1) + log(express.no.transactions +  
##      1) + log(metro.total.spend + 1) + log(metro.no.transactions +  
##      1) + log(superstore.total.spend + 1) + log(superstore.no.transactions  
##      +  
##      1) + log(extra.total.spend + 1) + log(extra.no.transactions +  
##      1) + log(fandf.total.spend + 1) + log(fandf.no.transactions +  
##      1) + log(petrol.total.spend + 1) + log(petrol.no.transactions +  
##      1) + log(direct.total.spend + 1) + log(direct.no.transactions +  
##      1), family = binomial, data = data_train_nona)  
##  
##  
## Coefficients:  
##  
##              Estimate Std. Error z value Pr(>|z|)  
## (Intercept)      -3.8393754   0.7860728  -4.884 1.04e-06  
## as.factor(gender)Male      -1.0185855   0.1120838  -9.088 < 2e-16  
## as.factor(affluency)Low      -0.1531079   0.1555150  -0.985  0.325  
## as.factor(affluency)Mid      -0.1438760   0.1278681  -1.125  0.261  
## as.factor(affluency)Very High   0.0820907   0.2317607   0.354  0.723  
## as.factor(affluency)Very Low   -0.0945375   0.2444770  -0.387  0.699  
## log(express.total.spend + 1)    0.0707130   0.0443016   1.596  0.110  
## log(express.no.transactions + 1) -0.0483935   0.1082230  -0.447  0.655
```

```

## log(metro.total.spend + 1)      0.0127495  0.0420744  0.303  0.762
## log(metro.no.transactions + 1)  0.0027111  0.1067234  0.025  0.980
## log(superstore.total.spend + 1)  0.0408004  0.0423360  0.964  0.335
## log(superstore.no.transactions + 1) -0.0181540  0.1103970 -0.164  0.869
## log(extra.total.spend + 1)      0.0043457  0.0412364  0.105  0.916
## log(extra.no.transactions + 1)  -0.1596766  0.1002229 -1.593  0.111
## log(fandf.total.spend + 1)      -0.0102178  0.0402931 -0.254  0.800
## log(fandf.no.transactions + 1)  -0.0003046  0.0789415 -0.004  0.997
## log(petrol.total.spend + 1)     -0.0344171  0.0418980 -0.821  0.411
## log(petrol.no.transactions + 1)  0.0694901  0.0778077  0.893  0.372
## log(direct.total.spend + 1)     0.0112770  0.0388238  0.290  0.771
## log(direct.no.transactions + 1) -0.0644028  0.0829091 -0.777  0.437
##
## (Intercept)                    ***
## as.factor(gender)Male          ***
## as.factor(affluency)Low
## as.factor(affluency)Mid
## as.factor(affluency)Very High
## as.factor(affluency)Very Low
## log(express.total.spend + 1)
## log(express.no.transactions + 1)
## log(metro.total.spend + 1)
## log(metro.no.transactions + 1)
## log(superstore.total.spend + 1)
## log(superstore.no.transactions + 1)
## log(extra.total.spend + 1)
## log(extra.no.transactions + 1)
## log(fandf.total.spend + 1)
## log(fandf.no.transactions + 1)
## log(petrol.total.spend + 1)
## log(petrol.no.transactions + 1)
## log(direct.total.spend + 1)
## log(direct.no.transactions + 1)
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 4316.3  on 36848  degrees of freedom
## Residual deviance: 4303.3  on 36829  degrees of freedom
## Penalized deviance: 4196.796
## AIC:  4343.3

#Residual Plot
binnedplot(x=brglm_logit$fitted.values,y=resid(brglm_logit,type="response"),n
class = 30)

```



#Error table

```
T_A <- mean(fitted(brglm_logit)<0.5 & data_train_nona$content_1==0)
T_B <- mean(fitted(brglm_logit)>0.5 & data_train_nona$content_1==0)
T_C <- mean(fitted(brglm_logit)<0.5 & data_train_nona$content_1==1)
T_D <- mean(fitted(brglm_logit)>0.5 & data_train_nona$content_1==1)
error_table <- matrix(c(T_A,T_B,T_C,T_D),byrow=TRUE,nrow = 2)
rownames(error_table) <- c("y=0","y=1")
colnames(error_table) <- c("y_hat=0","y_hat=1")
error_table
```

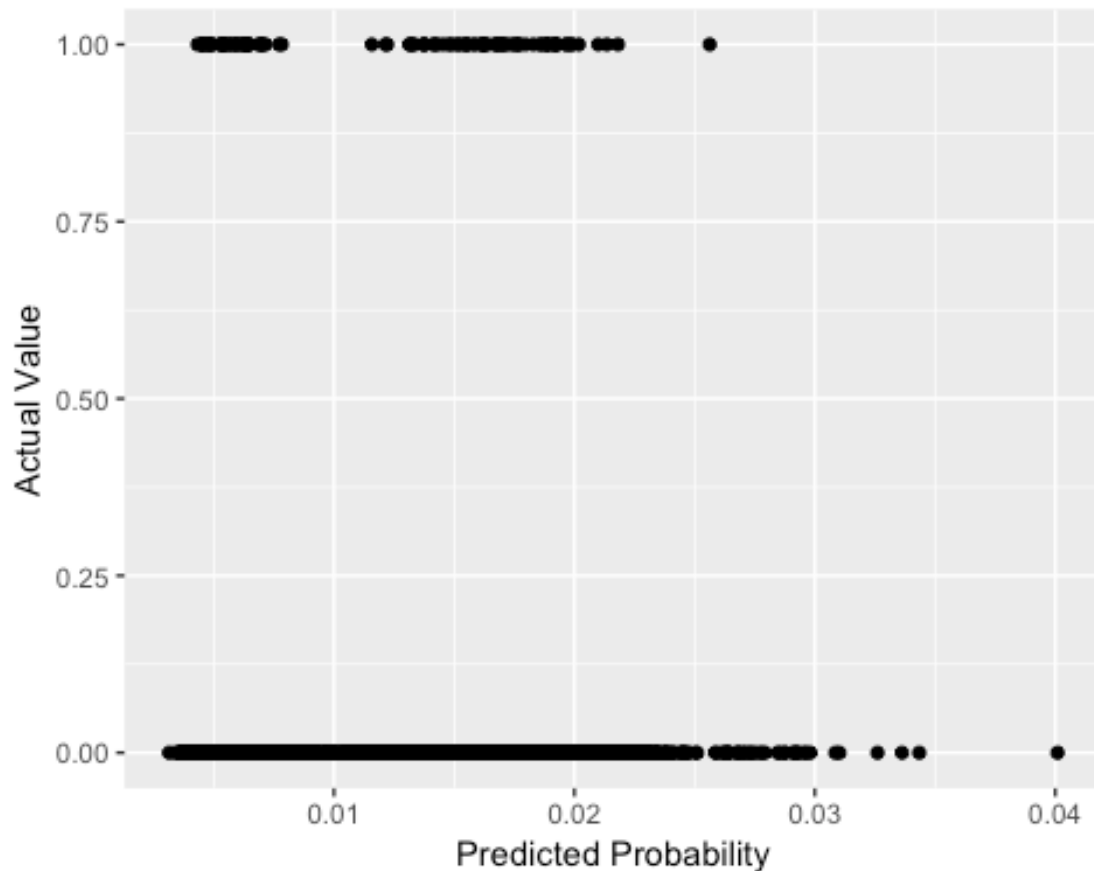
```
##          y_hat=0 y_hat=1
## y=0 0.98917203      0
## y=1 0.01082797      0
```

#Predictions

```
data_test[, -c(1:10,28,29)] %>% filter(is.na(data_test$content_1)==FALSE)->
data_test_glm
predict_brglm_class <- predict(brglm_logit, newdata = data_test_glm)
probs_brglm <- exp(predict_brglm_class)/(1+exp(predict_brglm_class))
#probability of 1
ggplot(data_test_use)+
  aes(x=probs_brglm,y=data_test_use$content_1)+
```



```
geom_point()+
xlab("Predicted Probability")+ylab("Actual Value")
```



From this brglm model:

1. We're seeing similar model outputs in summary, given that I've kept all the same independent variables.
2. The error table shows similar results and the same problem as before; So does the prediction test.
3. This model is not a good remedy for our problem in the context.

b. logistf: Firth's bias-Reduced penalized-likelihood logistic regression

```
stf_logit <-
logistf(content_1~as.factor(gender)+as.factor(affluency)+log(express.total.spend+1)+log(express.no.transactions+1)+log(metro.total.spend+1)+log(metro.no.transactions+1)+log(superstore.total.spend+1)+log(superstore.no.transactions+1)+log(extra.total.spend+1)+log(extra.no.transactions+1)+log(fandf.total.spend+1)+log(fandf.no.transactions+1)+log(petrol.total.spend+1)+log(petrol.no.transactions+1)+log(direct.total.spend+1)+log(direct.no.transactions+1), data=data
```

```

_train_nona)
summary(stf_logit)

## logistf(formula = content_1 ~ as.factor(gender) + as.factor(affluency) +
##      log(express.total.spend + 1) + log(express.no.transactions +
##      1) + log(metro.total.spend + 1) + log(metro.no.transactions +
##      1) + log(superstore.total.spend + 1) + log(superstore.no.transactions
+
##      1) + log(extra.total.spend + 1) + log(extra.no.transactions +
##      1) + log(fandf.total.spend + 1) + log(fandf.no.transactions +
##      1) + log(petrol.total.spend + 1) + log(petrol.no.transactions +
##      1) + log(direct.total.spend + 1) + log(direct.no.transactions +
##      1), data = data_train_nona)
##
## Model fitted by Penalized ML
## Confidence intervals and p-values by Profile Likelihood Profile Likelihood
Profile Likelihood Profile Likelihood Profile Likelihood Profile Likelihood
Profile Likelihood Profile Likelihood Profile Likelihood Profile Likelihood
Profile Likelihood Profile Likelihood Profile Likelihood Profile Likelihood
Profile Likelihood Profile Likelihood Profile Likelihood Profile Likelihood
Profile Likelihood Profile Likelihood
##
##               coef      se(coef)   lower 0.95
## (Intercept)      8.265762178 1.45816964 -0.09361282
## as.factor(gender)Male -2.681733939 0.37329880 -10.43159221
## as.factor(affluency)Low -0.310983464 0.28502558 -2.66173129
## as.factor(affluency)Mid -0.276789113 0.24866147 -2.59579494
## as.factor(affluency)Very High 0.117597328 0.50685485 -2.55784097
## as.factor(affluency)Very Low -0.266772222 0.42741972 -2.83075634
## log(express.total.spend + 1) 0.114124275 0.06890675 -0.35193519
## log(express.no.transactions + 1) -0.055585517 0.16922005 -1.22191244
## log(metro.total.spend + 1) 0.025300095 0.07165286 -0.47323907
## log(metro.no.transactions + 1) 0.025451917 0.17870317 -1.16856227
## log(superstore.total.spend + 1) 0.071798575 0.06842217 -0.39767633
## log(superstore.no.transactions + 1) -0.020901485 0.17636157 -1.21484583
## log(extra.total.spend + 1) 0.008335429 0.07174061 -0.48635474
## log(extra.no.transactions + 1) -0.360011040 0.21619233 -1.76266431
## log(fandf.total.spend + 1) -0.012393396 0.07078352 -0.49853942
## log(fandf.no.transactions + 1) -0.001244196 0.13941440 -0.86768331
## log(petrol.total.spend + 1) -0.056701713 0.07515365 -0.57787407
## log(petrol.no.transactions + 1) 0.122186055 0.13816305 -0.70516307
## log(direct.total.spend + 1) 0.019781378 0.06624631 -0.43390172
## log(direct.no.transactions + 1) -0.114712087 0.14342820 -1.03047011
##               upper 0.95      Chisq      p
## (Intercept)      16.3907622      Inf 0.000000e+00
## as.factor(gender)Male -0.7652501 0.0000000 1.000000e+00
## as.factor(affluency)Low 1.5473366 50.0962979 1.463829e-12
## as.factor(affluency)Mid 1.0843469 60.3368684 7.993606e-15
## as.factor(affluency)Very High 8.0082223 0.0000000 1.000000e+00
## as.factor(affluency)Very Low 7.6238528 19.1047779 1.237347e-05

```

```
## log(express.total.spend + 1)      0.5145738  0.0000000 1.000000e+00
## log(express.no.transactions + 1)  0.8865293 41.0247899 1.503105e-10
## log(metro.total.spend + 1)        0.4260381  0.0000000 1.000000e+00
## log(metro.no.transactions + 1)    1.0287588  0.0000000 1.000000e+00
## log(superstore.total.spend + 1)   0.4580335  0.0000000 1.000000e+00
## log(superstore.no.transactions + 1) 0.9865037 16.3986653 5.132135e-05
## log(extra.total.spend + 1)        0.4047644  0.0000000 1.000000e+00
## log(extra.no.transactions + 1)    0.7645834      Inf 0.000000e+00
## log(fandf.total.spend + 1)        0.3926898  4.4521282 3.485778e-02
## log(fandf.no.transactions + 1)    0.9210663  0.4232211 5.153341e-01
## log(petrol.total.spend + 1)       0.3685193 13.4863913 2.402999e-04
## log(petrol.no.transactions + 1)   1.0710366  0.0000000 1.000000e+00
## log(direct.total.spend + 1)       0.3993480  0.0000000 1.000000e+00
## log(direct.no.transactions + 1)   0.8004202 21.2177824 4.099434e-06
##
## Likelihood ratio test=36.94853 on 19 df, p=0.008053394, n=36849
## Wald test = 63.4694 on 19 df, p = 1.08019e-06
##
```

#Error table

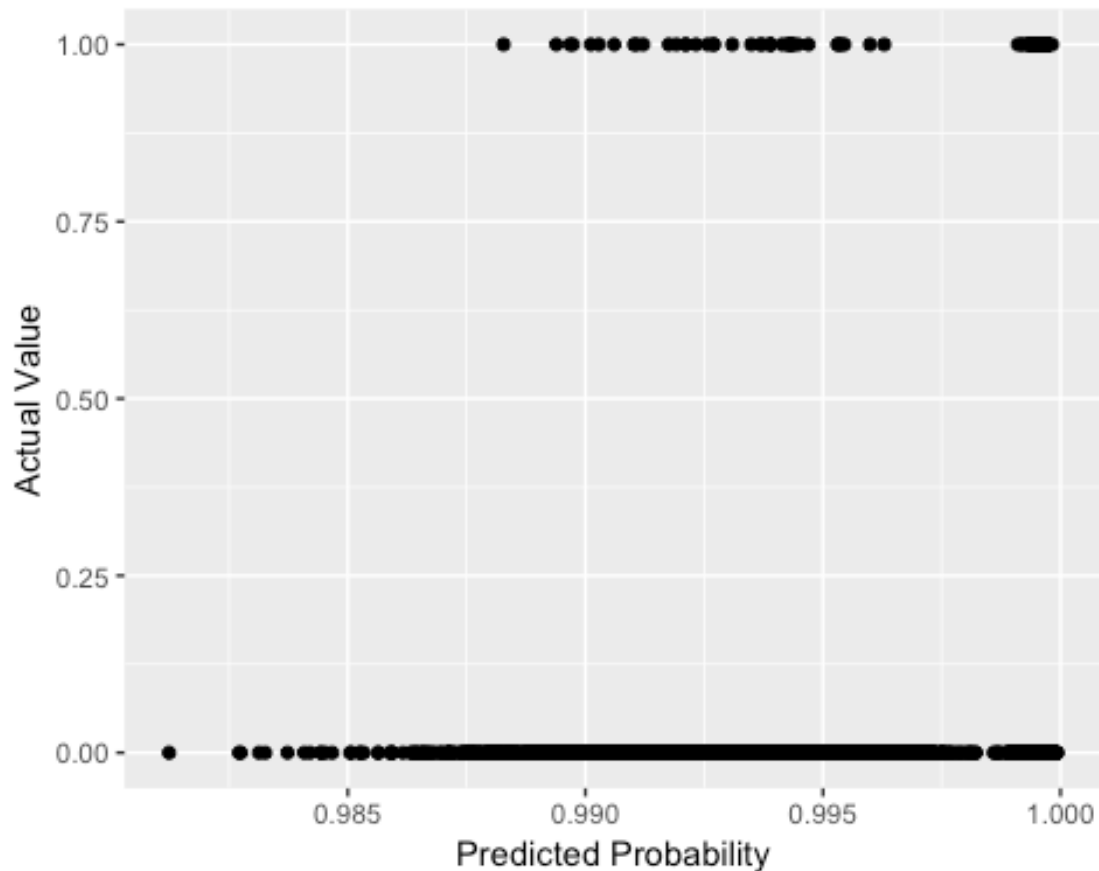
```
T_A <- mean(stf_logit$predict<0.5 & data_train_nona$content_1==0)
T_B <- mean(stf_logit$predict>0.5 & data_train_nona$content_1==0)
T_C <- mean(stf_logit$predict<0.5 & data_train_nona$content_1==1)
T_D <- mean(stf_logit$predict>0.5 & data_train_nona$content_1==1)
error_table <- matrix(c(T_A,T_B,T_C,T_D),byrow=TRUE,nrow = 2)
rownames(error_table) <- c("y=0","y=1")
colnames(error_table) <- c("y_hat=0","y_hat=1")
error_table
```

```
##      y_hat=0    y_hat=1
## y=0          0 0.98917203
## y=1          0 0.01082797
```

#Predictions

```
betas <- coef(stf_logit)
formula <-
content_1~as.factor(gender)+as.factor(affluency)+log(express.total.spend+1)+log(
express.no.transactions+1)+log(metro.total.spend+1)+log(metro.no.transacti
ons+1)+log(superstore.total.spend+1)+log(superstore.no.transactions+1)+log(ex
tra.total.spend+1)+log(extra.no.transactions+1)+log(fandf.total.spend+1)+log(
fandf.no.transactions+1)+log(petrol.total.spend+1)+log(petrol.no.transactions
+1)+log(direct.total.spend+1)+log(direct.no.transactions+1)
data_test_stf <- data_test[, -c(1,3:10,28,29)] %>%
filter(is.na(data_test$content_1)==FALSE)
X <- model.matrix(formula, data=data_test_stf)
pi.obs <- 1 / (1 + exp(-X %*% betas))
ggplot(data_test_use)+
aes(x=pi.obs,y=data_test_use$content_1)+
```

```
geom_point()+  
xlab("Predicted Probability")+ylab("Actual Value")
```



From this logitstf model:

1. The error table has the totally opposite result compared to the logistic model, which is not normal.
2. Prediction plot shows the same conclusion: the predicted probability higher than 0.99 but the majority of actual points are zeros.

In general, these two models designed to tackle separation problem didn't work out in this context, and I will be open to explore other solutions to this problem.

iii. Multilevel Mixed Effect Model for Contents

It would also make sense to try multilevel mixed effect model. I'll put gender and affluency, the only two factor variables in the data as random effects.

```

mix_logit <- glmer(content_1~(1|gender)+(1|affluency)+
  scale(express.total.spend)+scale(metro.total.spend)+
  scale(superstore.total.spend)+scale(extra.total.spend)+
  scale(fandf.total.spend)+scale(petrol.total.spend)+
  scale(direct.total.spend),
  data=data_train_nona,family = binomial,REML=FALSE)

## Warning: extra argument(s) 'REML' disregarded

summary(mix_logit)

## Generalized linear mixed model fit by maximum likelihood (Laplace
## Approximation) [glmerMod]
## Family: binomial ( logit )
## Formula:
## content_1 ~ (1 | gender) + (1 | affluency) + scale(express.total.spend) +
##   scale(metro.total.spend) + scale(superstore.total.spend) +
##   scale(extra.total.spend) + scale(fandf.total.spend) +
##   scale(petrol.total.spend) +
##   scale(direct.total.spend)
## Data: data_train_nona
##
##      AIC      BIC   logLik deviance df.resid
##  4336.1   4421.3  -2158.1   4316.1     36839
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -0.2012 -0.1267 -0.0874 -0.0769  14.9243
##
## Random effects:
## Groups      Name      Variance Std.Dev.
## affluency (Intercept) 2.542e-10 1.594e-05
## gender      (Intercept) 2.544e-01 5.043e-01
## Number of obs: 36849, groups:  affluency, 5; gender, 2
##
## Fixed effects:
##
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -4.642385    0.361219  -12.852   <2e-16 ***
## scale(express.total.spend)    0.046845    0.046970    0.997    0.319
## scale(metro.total.spend)   -0.036449    0.052916   -0.689    0.491
## scale(superstore.total.spend) 0.053987    0.044876    1.203    0.229
## scale(extra.total.spend)   -0.095496    0.059605   -1.602    0.109
## scale(fandf.total.spend)   -0.007329    0.051341   -0.143    0.886
## scale(petrol.total.spend)    0.034821    0.047397    0.735    0.463
## scale(direct.total.spend)   -0.055414    0.055294   -1.002    0.316
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##              (Intr) scl(xp..) scl(m..) scl(s..) scl(xt..) scl(f..) scl(p..)

```

```
## scl(xprs..) -0.006
## scl(mtr.t.) 0.005 0.009
## scl(sprs..) -0.007 -0.006 -0.010
## scl(xtr.t.) 0.013 0.000 0.006 0.000
## scl(fndf..) 0.001 0.001 -0.005 -0.011 0.003
## scl(ptrl..) -0.005 0.000 -0.004 0.004 0.016 0.003
## scl(drct..) 0.008 -0.001 0.005 -0.006 0.002 0.008 -0.002
```

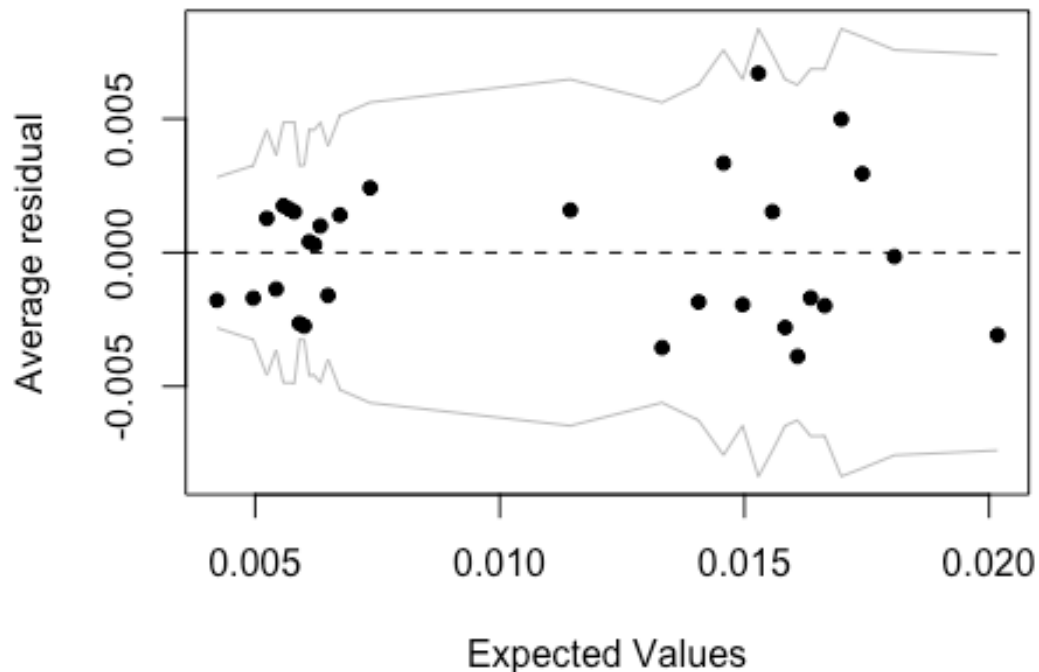
```
ranef(mix_logit)
```

```
## $affluency
## (Intercept)
## High 2.159202e-09
## Low -1.029648e-09
## Mid -1.937542e-09
## Very High 8.860451e-10
## Very Low -7.216275e-11
##
## $gender
## (Intercept)
## Female 0.4997983
## Male -0.4939002
```

```
#Residual Plot
```

```
binnedplot(fitted(mix_logit), resid(mix_logit, type = "response"), nclass = 30)
```

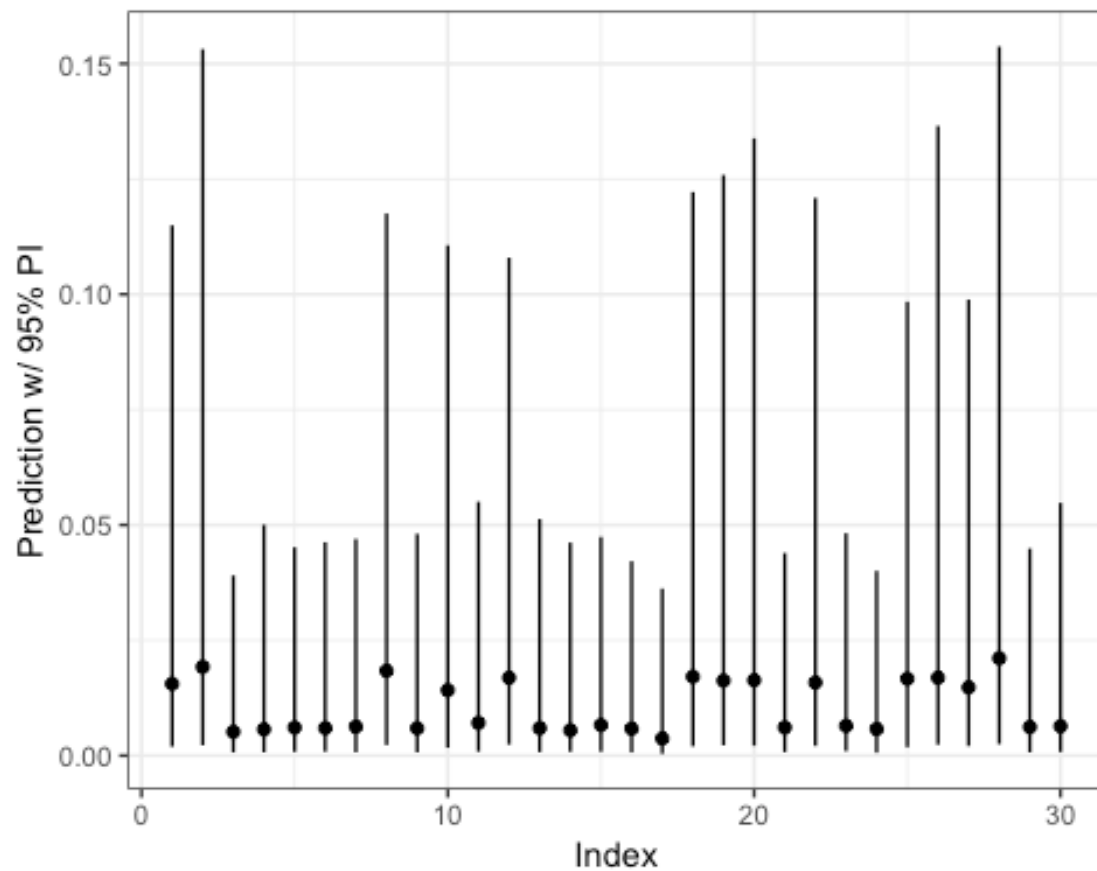
Binned residual plot



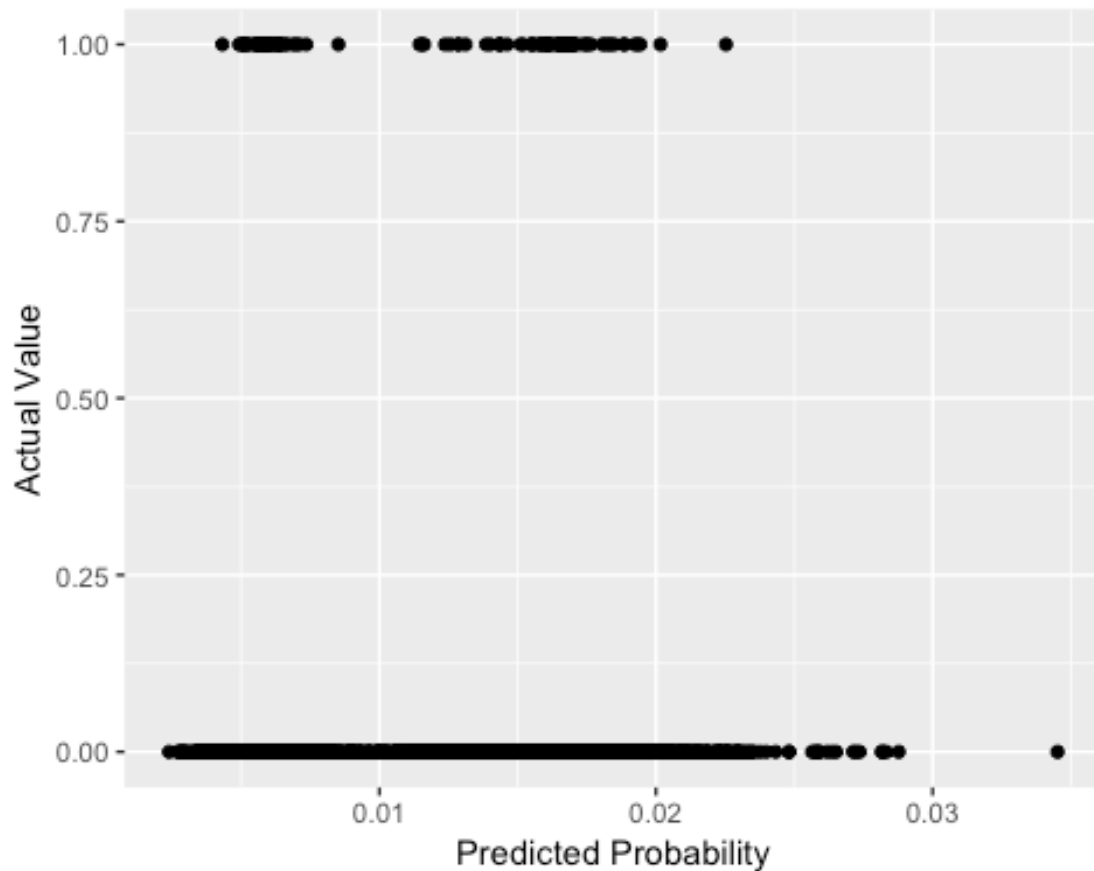
```
#Predictions
PI <- predictInterval(merMod = mix_logit, newdata = data_test_use,
  level = 0.95,
  stat = "median", type="linear.prediction",
  include.resid.var = TRUE)

## Warning: executing %dopar% sequentially: no parallel backend registered

probs_glmr <- matrix(nrow = 9173, ncol=3) #Transfer value to probability
for(i in 1:3){
  probs_glmr[,i] <- exp(PI[,i])/(1+exp(PI[,i])) #probability of 1
}
colnames(probs_glmr) <- colnames(PI)
probs_glmr <- as.data.frame(probs_glmr)
#Plot only on the first 30 observations
ggplot(aes(x=1:30, y=fit, ymin=lwr, ymax=upr), data=probs_glmr[1:30,]) +
  geom_point() +
  geom_linerange() +
  labs(x="Index", y="Prediction w/ 95% PI") + theme_bw()
```



```
#Plot predicted value vs. actual value
ggplot(data=data_test_use,aes(x=probs_glm[,1],y=data_test_use$content_1))+
  geom_point()+
  xlab("Predicted Probability")+ylab("Actual Value")
```

#Error table

```
T_A <- mean(probs_glmmer[,1]<0.5 & data_train_nona$content_1==0)

## Warning in probs_glmmer[, 1] < 0.5 & data_train_nona$content_1 == 0: longer
## object length is not a multiple of shorter object length

T_B <- mean(probs_glmmer[,1]>0.5 & data_train_nona$content_1==0)

## Warning in probs_glmmer[, 1] > 0.5 & data_train_nona$content_1 == 0: longer
## object length is not a multiple of shorter object length

T_C <- mean(probs_glmmer[,1]<0.5 & data_train_nona$content_1==1)

## Warning in probs_glmmer[, 1] < 0.5 & data_train_nona$content_1 == 1: longer
## object length is not a multiple of shorter object length

T_D <- mean(probs_glmmer[,1]>0.5 & data_train_nona$content_1==1)

## Warning in probs_glmmer[, 1] > 0.5 & data_train_nona$content_1 == 1: longer
## object length is not a multiple of shorter object length

error_table <- matrix(c(T_A,T_B,T_C,T_D),byrow=TRUE,nrow = 2)
rownames(error_table) <- c("y=0", "y=1")
colnames(error_table) <- c("y_hat=0", "y_hat=1")
error_table
```

```
##      y_hat=0 y_hat=1
## y=0 0.98917203      0
## y=1 0.01082797      0
```

From this multilevel mixed effect model:

1. The affluency as a random effect doesn't have a high influence on the model, with its random effect coefficients really close to zero. While the gender effect relatively has a large difference.
2. Binned residual plot of this model looks normal.
3. The predictions made on test set and the error table seem that it's still suffering the same problem left before, known as "seperation" problem. The three models (multinomial, logistic and multilevel) are showing exactly same results in error table and predictions.

E. Discussion

i. Implitation

1. From all the results and the interpretations of the models fitted above, it's not reseasonable to use these models to predict clicking rate; while the models do explain the clicking rate to some extend.
2. I've also thought of sampling a data of 50% zeros in content_1 and 50% ones in content_1, but this data modification will lose the proportion's information and it doesn't make sense because one of the most important messages conveyed from the data is the proportion of ones in content clicking choice.
- 3.

ii. Limitation

1. The "rare event" problem is not resolved in this project, which will constraint the models proceeding predictions.
2. The data is random simulated by the company with some latent assumptions, which are not disclosed to data users. These assumptions are supposed to be discovered by learning results, but in my project, they have not been identified. This can potentially be solved when I've learnt more about machine learning.

iii. Future Directions

1. I may continue on looking for other methods which could resolve the separation problem in this dataset.
2. The data's structure or other latent relations may violate the assumptions of regressions I'm applying in the project. Therefore, some other more advanced learning methods may be more appropriate for this data's predicting problem.
3. After finding a more proper and efficient way to make predictions, I will continue on predicting content_2 to content_9.

Appendix:

I. Other EDA Plots

1. Total amount of transactions vs. Affluency

```
#Group data
plot_amt <- data_train %>%
  dplyr::select(affluency, express.total.spend, metro.total.spend,
superstore.total.spend, extra.total.spend, fandf.total.spend,
                petrol.total.spend, direct.total.spend) %>%
  group_by(affluency) %>% summarise(Express=sum(express.total.spend),
                                   Metro=sum(metro.total.spend),
                                   Superstore=sum(superstore.total.spend),
                                   Extrastore=sum(extra.total.spend),
                                   Fandf=sum(fandf.total.spend),
                                   Petrol=sum(petrol.total.spend),
                                   Direct=sum(direct.total.spend))

plot_mat <- matrix(0, nrow = 35, ncol = 3)
plot_mat[,1] <- rep(c("High", "Low", "Mid", "Very High", "Very Low"), 7)
plot_mat[,2] <- c(rep(x="Express", 5), rep(x="Metro", 5),
                 rep(x="Superstore", 5), rep(x="Extrastore", 5),
                 rep(x="Fandf", 5), rep(x="Petrol", 5), rep(x="Direct", 5))
plot_mat[,3] <-
rbind(plot_amt$Express, plot_amt$Metro, plot_amt$Superstore, plot_amt$Extrastore
,
      plot_amt$Fandf, plot_amt$Petrol, plot_amt$Direct)
plot_mat <- as.data.frame(plot_mat)
colnames(plot_mat) <- c("Affluency", "Store.Type", "Total.Amount")
plot_mat$Total.Amount <- as.numeric(plot_mat$Total.Amount)

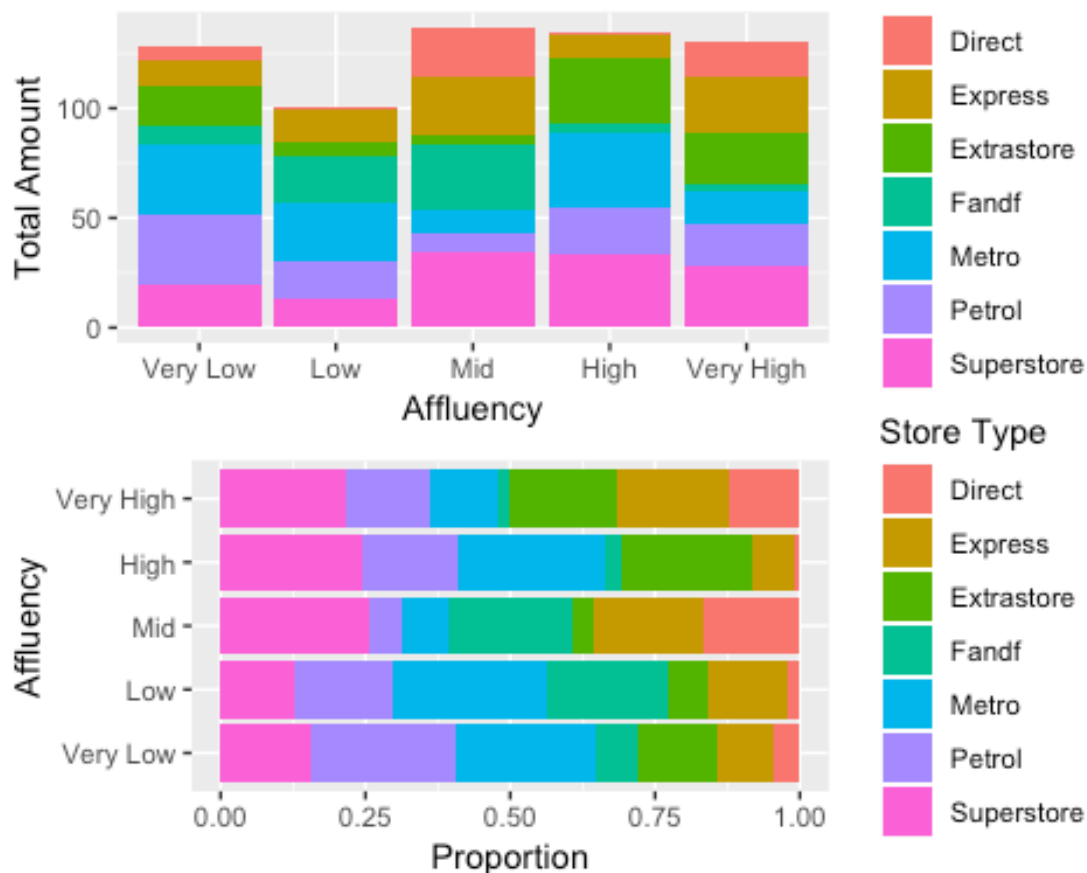
#Plot
```

```

g1 <-
  ggplot(data = plot_mat)+
  aes(x=as.factor(Affluency),y=Total.Amount,fill=as.factor(Store.Type))+
  geom_col(position = 'stack')+
  scale_x_discrete(limits=c("Very Low","Low","Mid","High","Very High"))+
  xlab("Affluency")+
  ylab("Total Amount")+
  scale_fill_discrete(name="Store Type")
g2 <-
  ggplot(data = plot_mat)+
  aes(x=as.factor(Affluency),y=Total.Amount,fill=as.factor(Store.Type))+
  geom_col(stat="identity", position="fill")+
  scale_x_discrete(limits=c("Very Low","Low","Mid","High","Very High"))+
  xlab("Affluency")+
  ylab("Proportion")+
  scale_fill_discrete(name="Store Type")+
  coord_flip()

## Warning: Ignoring unknown parameters: stat
grid.arrange(g1, g2)

```



2. Number of transactions vs. Affluency

```
#Group data
data_train %>%
dplyr::select(affluency, express.no.transactions, metro.no.transactions,

superstore.no.transactions, extra.no.transactions, fandf.no.transactions,
               petrol.no.transactions, direct.no.transactions) %>%
  group_by(affluency) %>% summarise(Express=sum(express.no.transactions),
                                   Metro=sum(metro.no.transactions),

Superstore=sum(superstore.no.transactions),
               Extrastore=sum(extra.no.transactions),
               Fandf=sum(fandf.no.transactions),
               Petrol=sum(petrol.no.transactions),
               Direct=sum(direct.no.transactions))-

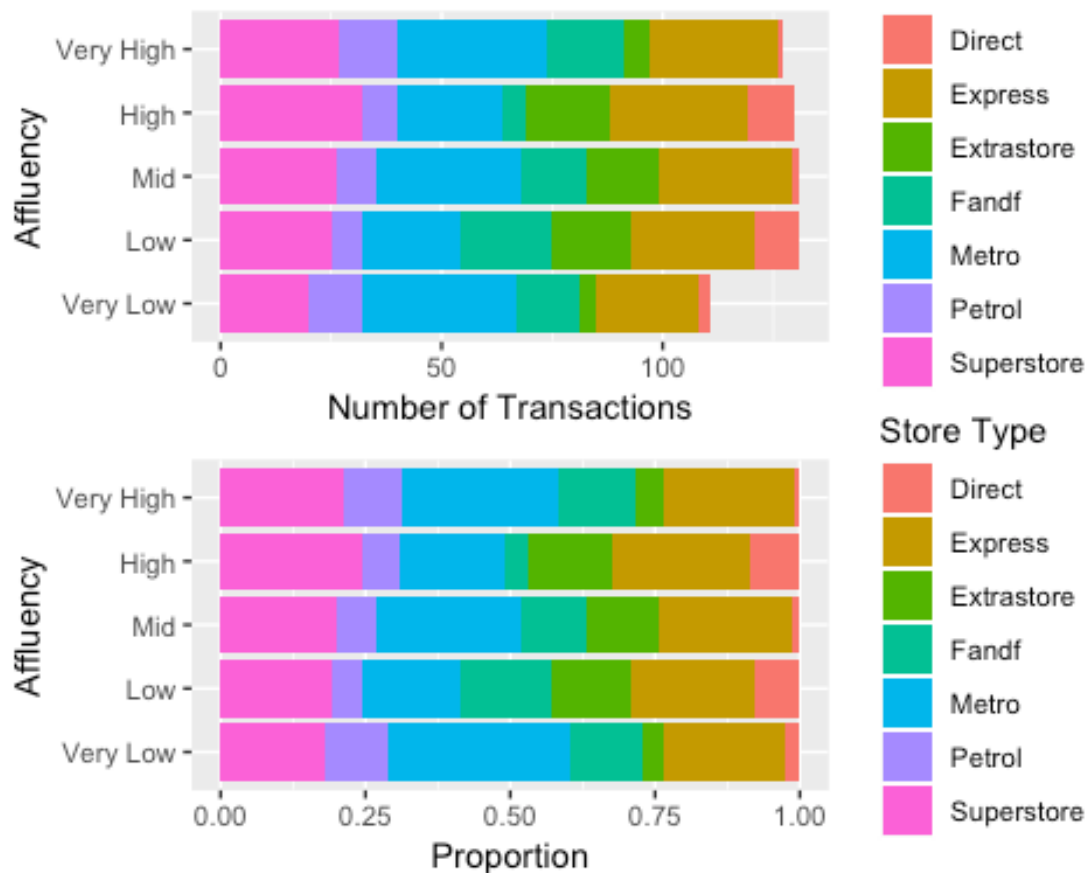
>plot_amt

plot_mat <- matrix(0, nrow = 35, ncol = 3)
plot_mat[,1] <- rep(c("High", "Low", "Mid", "Very High", "Very Low"), 7)
plot_mat[,2] <- c(rep(x="Express", 5), rep(x="Metro", 5),
                  rep(x="Superstore", 5), rep(x="Extrastore", 5),
                  rep(x="Fandf", 5), rep(x="Petrol", 5), rep(x="Direct", 5))
plot_mat[,3] <-
rbind(plot_amt$Express, plot_amt$Metro, plot_amt$Superstore, plot_amt$Extrastore
,
       plot_amt$Fandf, plot_amt$Petrol, plot_amt$Direct)
plot_mat <- as.data.frame(plot_mat)
colnames(plot_mat) <- c("Affluency", "Store.Type", "No.Amount")
plot_mat$No.Amount <- as.numeric(plot_mat$No.Amount)

#Plot
g1 <-
  ggplot(data = plot_mat)+
  aes(x=as.factor(Affluency), y=No.Amount, fill=as.factor(Store.Type))+
  geom_col(position = 'stack')+
  scale_x_discrete(limits=c("Very Low", "Low", "Mid", "High", "Very High"))+
  xlab("Affluency")+
  ylab("Number of Transactions")+
  scale_fill_discrete(name="Store Type")+
  coord_flip()
g2 <-
  ggplot(data = plot_mat)+
  aes(x=as.factor(Affluency), y=No.Amount, fill=as.factor(Store.Type))+
  geom_col(stat="identity", position="fill")+
  scale_x_discrete(limits=c("Very Low", "Low", "Mid", "High", "Very High"))+
  xlab("Affluency")+
  ylab("Proportion")+
  scale_fill_discrete(name="Store Type")+
  coord_flip()
```

```
## Warning: Ignoring unknown parameters: stat
```

```
grid.arrange(g1, g2)
```



3. Total amount of transactions vs. Gender

```
#Group data
```

```
data_train %>% dplyr::select(gender, express.total.spend, metro.total.spend,
superstore.total.spend, extra.total.spend, fandf.total.spend,
                        petrol.total.spend, direct.total.spend) %>%
  group_by(gender) %>% summarise(Express=sum(express.total.spend),
                                Metro=sum(metro.total.spend),
                                Superstore=sum(superstore.total.spend),
                                Extrastore=sum(extra.total.spend),
                                Fandf=sum(fandf.total.spend),
                                Petrol=sum(petrol.total.spend),
                                Direct=sum(direct.total.spend))->plot_amt

plot_mat <- matrix(0, nrow = 14, ncol = 3)
plot_mat[,1] <- rep(c("Male", "Female"), 7)
plot_mat[,2] <- c(rep(x="Express", 2), rep(x="Metro", 2),
```

```

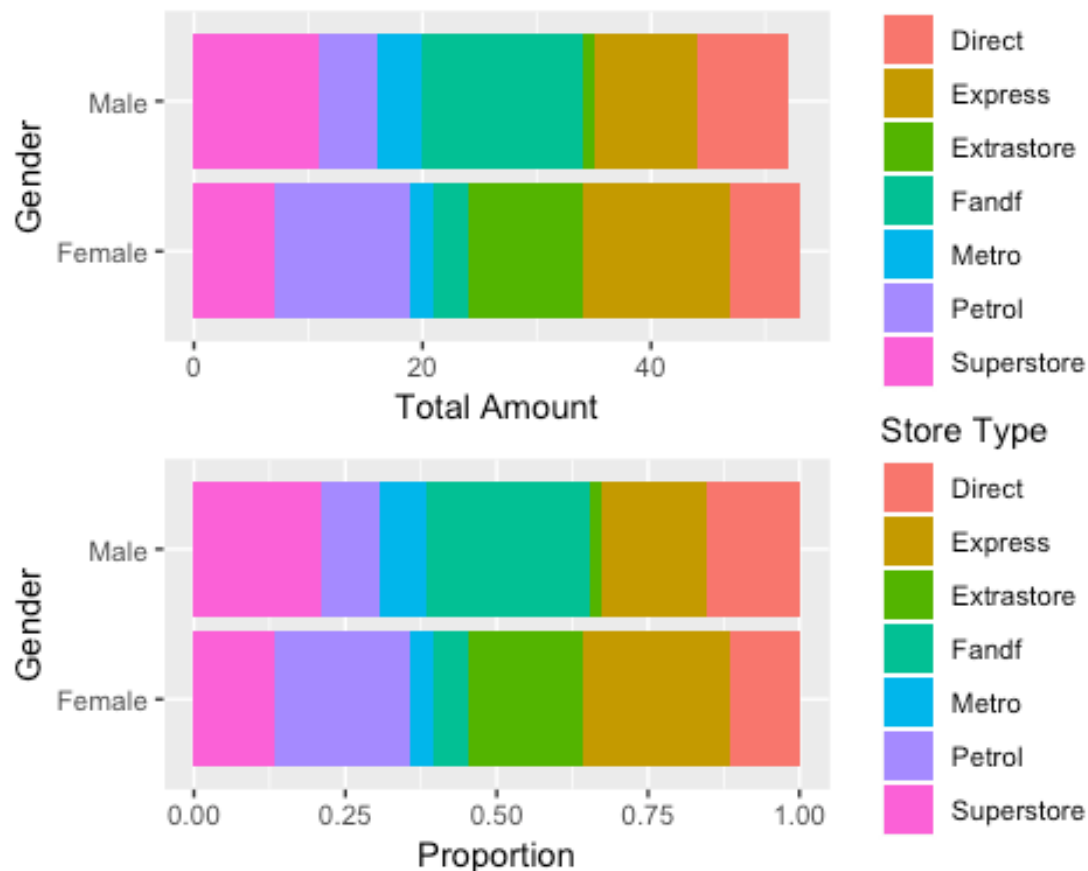
      rep(x="Superstore",2),rep(x="Extrastore",2),
      rep(x="Fandf",2),rep(x="Petrol",2),rep(x="Direct",2))
plot_mat[,3] <-
rbind(plot_amt$Express,plot_amt$Metro,plot_amt$Superstore,plot_amt$Extrastore
,
      plot_amt$Fandf,plot_amt$Petrol,plot_amt$Direct)
plot_mat <- as.data.frame(plot_mat)
colnames(plot_mat) <- c("Gender","Store.Type","Total.Amount")
plot_mat$Total.Amount <- as.numeric(plot_mat$Total.Amount)

#Plot
g1 <-
  ggplot(data = plot_mat)+
  aes(x=as.factor(Gender),y=Total.Amount,fill=as.factor(Store.Type))+
  geom_col(position = 'stack')+
  xlab("Gender")+
  ylab("Total Amount")+
  scale_fill_discrete(name="Store Type")+
  coord_flip()
g2 <-
  ggplot(data = plot_mat)+
  aes(x=as.factor(Gender),y=Total.Amount,fill=as.factor(Store.Type))+
  geom_col(stat="identity", position="fill")+
  xlab("Gender")+
  ylab("Proportion")+
  scale_fill_discrete(name="Store Type")+
  coord_flip()

## Warning: Ignoring unknown parameters: stat

grid.arrange(g1, g2)

```



4. Number of transactions vs. Gender

```
#Group data
data_train %>% select(gender,express.no.transactions,metro.no.transactions,
superstore.no.transactions,extra.no.transactions,fandf.no.transactions,
                    petrol.no.transactions,direct.no.transactions) %>%
  group_by(gender) %>% summarise(Express=sum(express.no.transactions),
                                Metro=sum(metro.no.transactions),

Superstore=sum(superstore.no.transactions),
                                Extrastore=sum(extra.no.transactions),
                                Fandf=sum(fandf.no.transactions),
                                Petrol=sum(petrol.no.transactions),
                                Direct=sum(direct.no.transactions))-

>plot_amt

plot_mat <- matrix(0,nrow = 14,ncol = 3)
plot_mat[,1] <- rep(c("Male","Female"),7)
plot_mat[,2] <- c(rep(x="Express",2),rep(x="Metro",2),
                  rep(x="Superstore",2),rep(x="Extrastore",2),
                  rep(x="Fandf",2),rep(x="Petrol",2),rep(x="Direct",2))
```



```

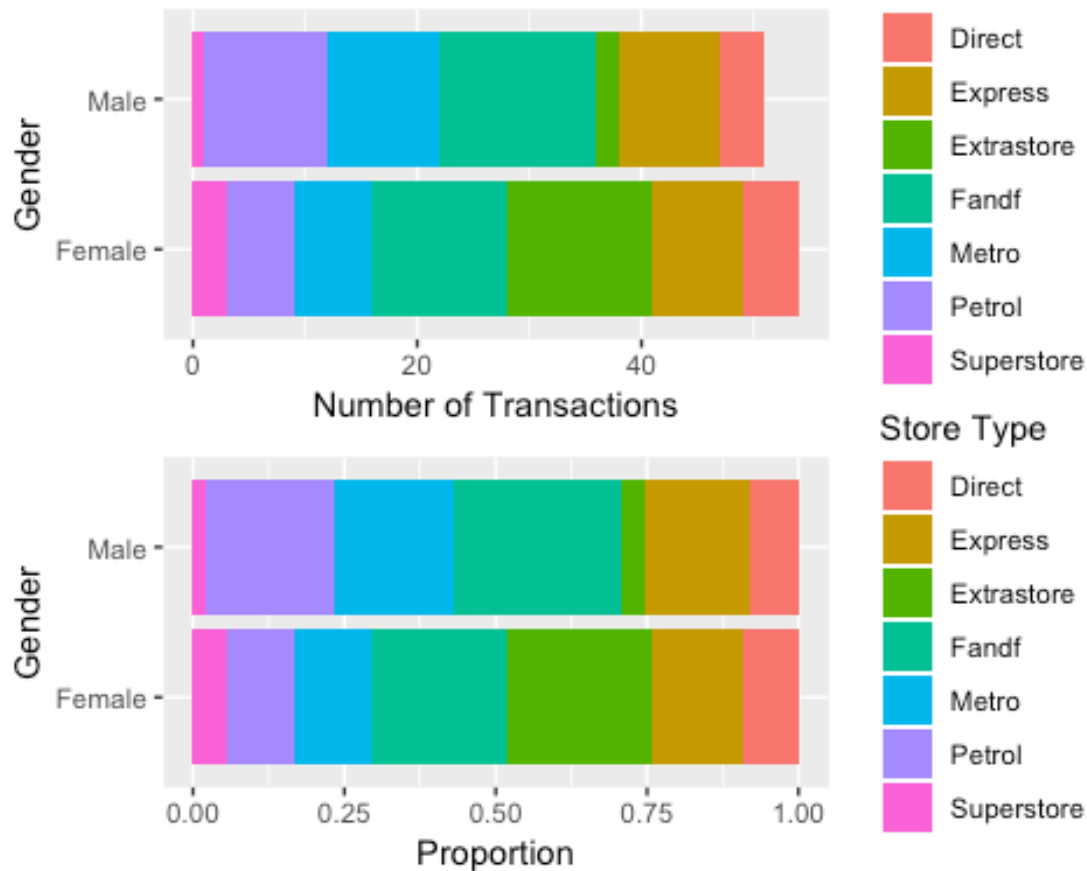
plot_mat[,3] <-
rbind(plot_amt$Express,plot_amt$Metro,plot_amt$Superstore,plot_amt$Extrastore
,
        plot_amt$Fandf,plot_amt$Petrol,plot_amt$Direct)
plot_mat <- as.data.frame(plot_mat)
colnames(plot_mat) <- c("Gender","Store.Type","No.Amount")
plot_mat$No.Amount <- as.numeric(plot_mat$No.Amount)

#Plot
g1 <-
  ggplot(data = plot_mat)+
  aes(x=as.factor(Gender),y=No.Amount,fill=as.factor(Store.Type))+
  geom_col(position = 'stack')+
  xlab("Gender")+
  ylab("Number of Transactions")+
  scale_fill_discrete(name="Store Type")+
  coord_flip()
g2 <-
  ggplot(data = plot_mat)+
  aes(x=as.factor(Gender),y=No.Amount,fill=as.factor(Store.Type))+
  geom_col(stat="identity", position="fill")+
  xlab("Gender")+
  ylab("Proportion")+
  scale_fill_discrete(name="Store Type")+
  coord_flip()

## Warning: Ignoring unknown parameters: stat

grid.arrange(g1, g2)

```



II. Another Method on Data Preprocessing

I've first use the data with all observations to fit all three regression models, which means the NAs have not been removed. The results were similar: all three model's explaining abilities on the content_1 clicking rate were good but the models predicted each output (no matter the values of variables) to be zero, because of the seperation problem, which means that the training dataset has overwhelming amount of zeros compared to ones for content_1. I then think of deleting NA at the beginning, because after deleting the NAs, the proportion of ones will increase a little bit (but finally found out that this was not sufficient).

III. Statistical Learning Methods Used in the Project

For this project, after getting these results, I've applied some learning methods which can potentially be helpful:

1. Gradient Boosting Desicion Tree: <https://www.r-bloggers.com/gradient-boosting-in-r/>

It was used to deal with the “rare event” problem, but I had a hard time interpreting the results and I was not sure about if the whole process was correct or not, so I didn’t put it into the report.

2. Neural Network: <https://www.r-bloggers.com/fitting-a-neural-network-in-r-neuralnet-package/>

Depending on my understanding, neural network can handle some latent, complexe and multi-layer relations, which may be useful on identifying the company’s initial assumptions on the simulations. But I think I need to spend more time learning its concepts and package to have a whole picture understanding before applying it to data.

References:

[1]: https://turi.com/learn/gallery/notebooks/click_through_rate_prediction_intro.html

[2]: <https://www3.nd.edu/~rwilliam/stats3/rareevents.pdf>

[3]: <https://www.r-bloggers.com/example-8-15-firth-logistic-regression/>

[4]: <https://www.r-bloggers.com/making-sense-of-logarithmic-loss/>

[5]: <https://cran.r-project.org/web/packages/brglm/brglm.pdf>