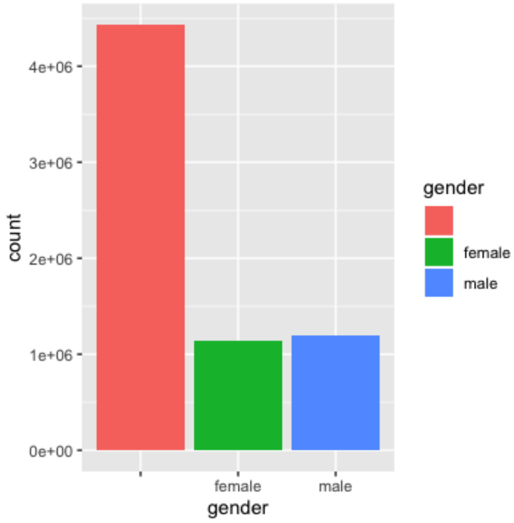
DATA ANALYTICS WITH R

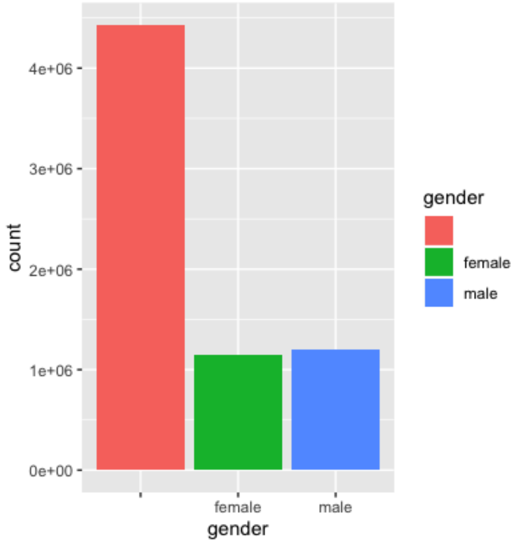
HOMEWORK 4

JASNEEK SINGH CHUGH
JC2433

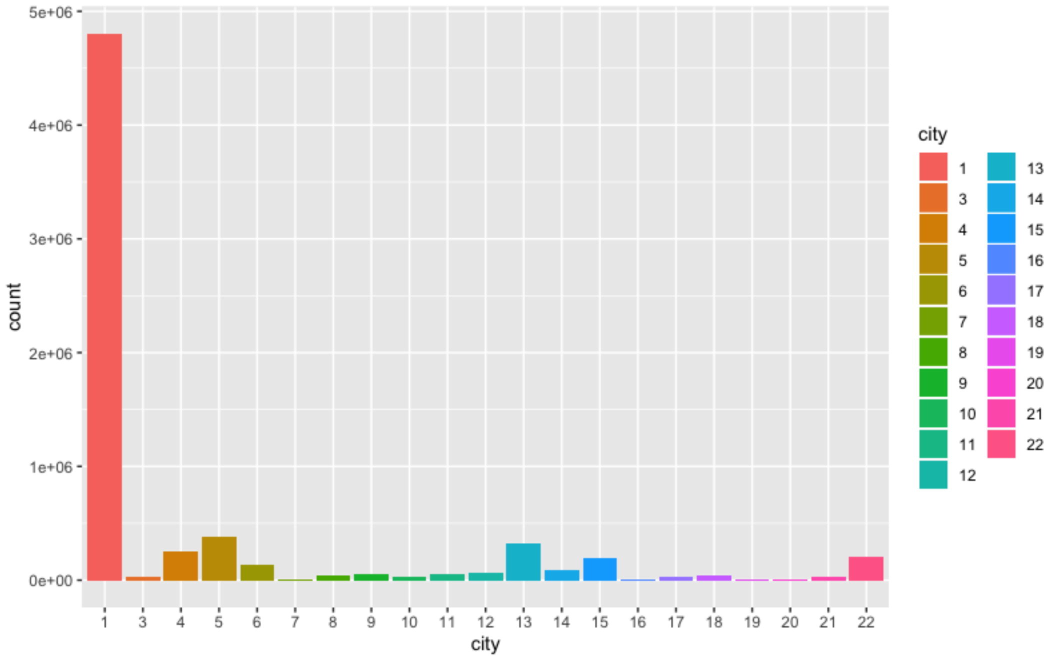
```
#Loading Required Packages/Libraries
> library(dplyr)
> library(data.table)
> library(ggplot2)
> library(rpart)
> library(caret)
> library(foreach)
> library(ROCR)
> library(pROC)
> library(lubridate)
> library(scales)
> library(grid)
> library(gridExtra)
> library(RColorBrewer)
> library(corrplot)

> #reading the required files
> trainData <- fread('data/train_v2.csv', sep = ",", header=T, stringsAsFactors = T)
> testData<- fread('data/sample_submission_v2.csv', sep = ",", header=T, stringsAsFactors =
T)
> members <- fread('data/members_v3.csv', sep = ",", header=T, stringsAsFactors = T)

> trans <- fread('data/transactions_v2.csv', sep = ",", header=T, stringsAsFactors = T)
> sum(is.na(trainData))
[1] 0
> sum(is.na(members))
[1] 0
> sum(is.na(trans))
[1] 0
> table(trainData$is_churn)

     0      1 
883630  87330 

> #Reformating

> trainData <- trainData %>%
+   mutate(is_churn = factor(is_churn))
> testData <- testData %>%
+   mutate(is_churn = factor(is_churn))
> trans <- trans %>%
+   mutate(pay_met = factor(payment_method_id),
+          auto_renew = factor(is_auto_renew),
+          is_cancel = factor(is_cancel),
+          trans_date = ymd(transaction_date),
+          exp_date = ymd(membership_expire_date))
> members <- members %>%
+   mutate(city = factor(city),
+          gender = factor(gender),
+          reg_via = factor(registered_via),
+          reg_init = ymd(registration_init_time))

> #EDA
> ggplot(data=trainData)+geom_bar(aes(x=is_churn, fill= is_churn))# The vast majority of users didn't churn
```



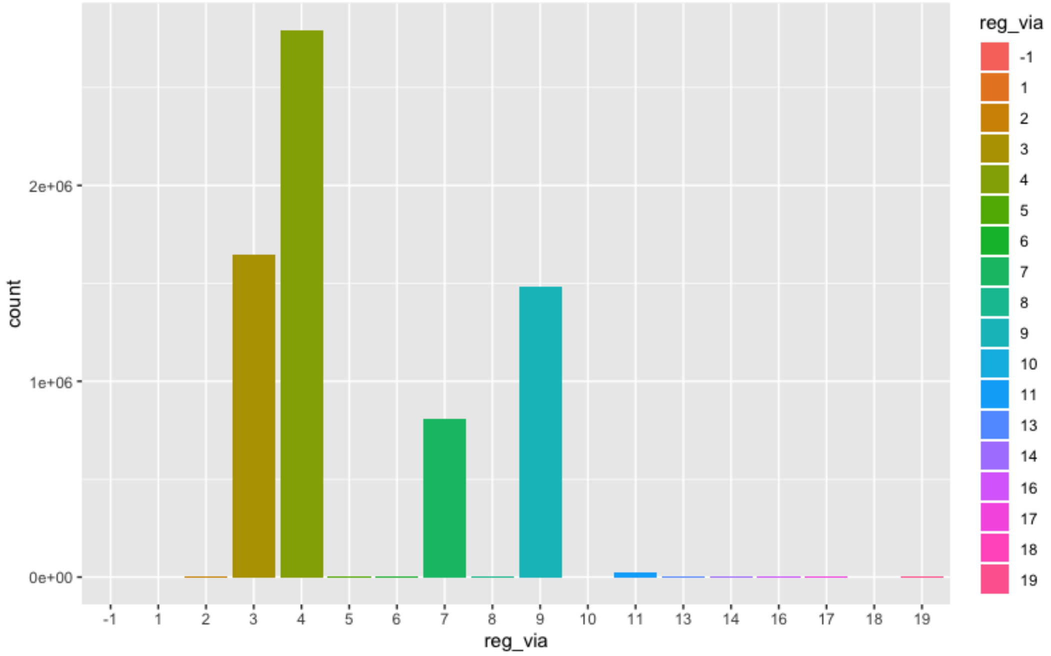
```
> ggplot(data=members)+geom_bar(aes(x=gender, fill= gender)) # there are blanks as well in Gender column
```



```
> ggplot(data=members)+geom_bar(aes(x=city, fill= city))
```

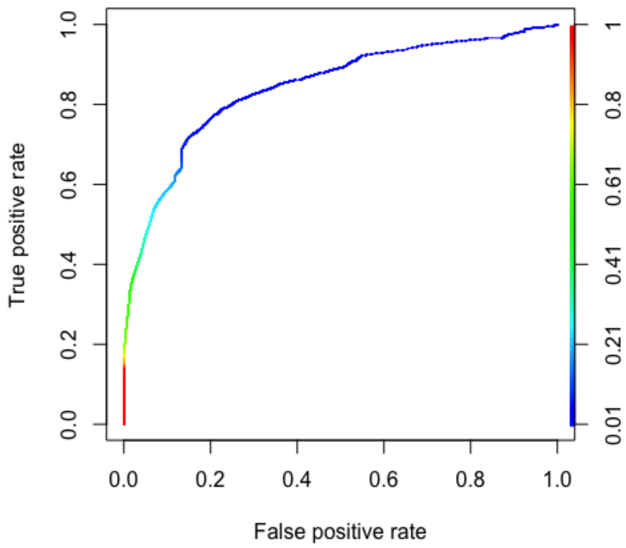


```
> ggplot(data=members)+geom_bar(aes(x=reg_via, fill= reg_via))
```



```
> # combining all datasets in one
> tr_data<- trainData %>%
+   inner_join(trans,by="msno") %>%
+   inner_join(members,by="msno")  #Inner join of trans data and members data to Train
data
> te_data<- testData %>%
+   inner_join(trans,by="msno") %>%
+   inner_join(members,by="msno")

> length( unique(tr_data$msno) ) # to make sure the data is joined correctly
[1] 825368
```

```
> table(tr_data$is_churn)

     0      1
911348 105635
> sum(is.na(tr_data)) #to check if any NA's are present in the data
[1] 0
> length(tr_data$msno)
[1] 1016983
> xtabs(~is_churn+gender,data=tr_data)
        gender
is_churn        female   male
       0 491615 199871 219862
       1  47255  27944  30436

> # Question 1.
> # logistic regression on data.
> tr_dataLength = nrow(tr_data)/2
> trainSet = tr_data[1:tr_dataLength,]
> testSet = tr_data[tr_dataLength:length(tr_data[,1]),]
> # selecting few features only
> columns = c("is_churn", "payment_plan_days", "plan_list_price" ,"city","auto_renew",
+          "trans_date", "exp_date", "is_cancel","reg_via")
> kk_train_data = trainSet %>%
+              select(columns)
> kk_test_data = testSet%>%
+          select(columns)
> sum(is.na(kk_train_data$is_churn))#to make sure there are no NA's in the data
[1] 0
> sum(is.na(kk_test_data$is_churn))
[1] 0
> #model
> churn_model = glm(is_churn ~ ., data = kk_train_data, family = binomial(link="logit"))
> churn_prob<-predict(churn_model, newdata=kk_test_data[-1], type="response")
> churn_pred <- ifelse(prob>=0.5,1,0)
> # Performance on test dataset
> #Accuracy
> accuracy = mean(kk_test_data$is_churn == churn_pred)
> accuracy
[1] 0.93158
> #Classification Error, it's calculated as 1-Accuracy
> class_error = 1 - accuracy
> class_error
[1] 0.06841996
> # Plotting ROC
> pred<-prediction(churn_prob, kk_test_data[1])
> plot(performance(pred, "tpr","fpr"), colorize=TRUE)
```



```
> #AUC- Area under the curve
> AUC=performance(pred,"auc")
> AUC@y.values[[1]]
[1] 0.8429289

> # Question 2.
#Used Random forest model for cross validation
> columns = c("is_churn", "payment_plan_days", "plan_list_price" ,"city","auto_renew",
+             "trans_date", "exp_date", "is_cancel","reg_via")
> tr = tr_data %>% select( columns) %>%
+   sample_n(1000)
> model<- train(is_churn ~ ., data = tr, method="rf", trControl=trainControl(method="cv",
number=5,  verboseIter =TRUE))
+ Fold1: mtry= 2
- Fold1: mtry= 2
+ Fold1: mtry=22
- Fold1: mtry=22
+ Fold1: mtry=43
- Fold1: mtry=43
+ Fold2: mtry= 2
- Fold2: mtry= 2
+ Fold2: mtry=22
- Fold2: mtry=22
+ Fold2: mtry=43
- Fold2: mtry=43
+ Fold3: mtry= 2
- Fold3: mtry= 2
+ Fold3: mtry=22
- Fold3: mtry=22
+ Fold3: mtry=43
```

```
- Fold3: mtry=43
+ Fold4: mtry= 2
- Fold4: mtry= 2
+ Fold4: mtry=22
- Fold4: mtry=22
+ Fold4: mtry=43
- Fold4: mtry=43
+ Fold5: mtry= 2
- Fold5: mtry= 2
+ Fold5: mtry=22
- Fold5: mtry=22
+ Fold5: mtry=43
- Fold5: mtry=43
Aggregating results
Selecting tuning parameters
Fitting mtry = 43 on full training set
> model
Random Forest

1000 samples
   8 predictor
   2 classes: '0', '1'

No pre-processing
Resampling: Cross-Validated (5 fold)
Summary of sample sizes: 800, 799, 800, 801, 800
Resampling results across tuning parameters:

  mtry  Accuracy   Kappa
   2    0.9170042  0.0000000
  22    0.9439743  0.5262622
  43    0.9489894  0.6113819

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was mtry = 43.
>  #the model was taking way too long to run the cross validation and resulted in interruption of R studio. Therefore, i have sampled the data to 1000 records and run cross validation on it.

> #Question3
> columns = c("is_churn", "payment_plan_days", "plan_list_price" ,"city","auto_renew",
+             "trans_date", "exp_date", "is_cancel","reg_via")
> kk_tr_data = tr_data %>%  select(columns)
> kk_te_data = te_data %>%  select(columns)
> churn_model = glm(is_churn ~ ., data = kk_tr_data, family = binomial(link="logit"))
> churn_prob<-predict(churn_model, newdata=kk_te_data[-1], type="response")
> churn_pred <- ifelse(prob>=0.5,1,0)
> kk_churn_prob<- data.frame(te_data, churn_prob)
> kk_churn_prob<- kk_churn_prob %>% select(c("msno", "churn_prob"))
> kk_churn_prob<- distinct(kk_churn_prob, msno, .keep_all = TRUE)
> write.csv(kk_churn_prob, "kk_churn_prob.csv", row.names = FALSE)  // the output will be
save as "kk_churn_prob.csv"
```