a.Read the dataset and identify the right feature

Read train dataset

train = fread("blogData_train.csv")

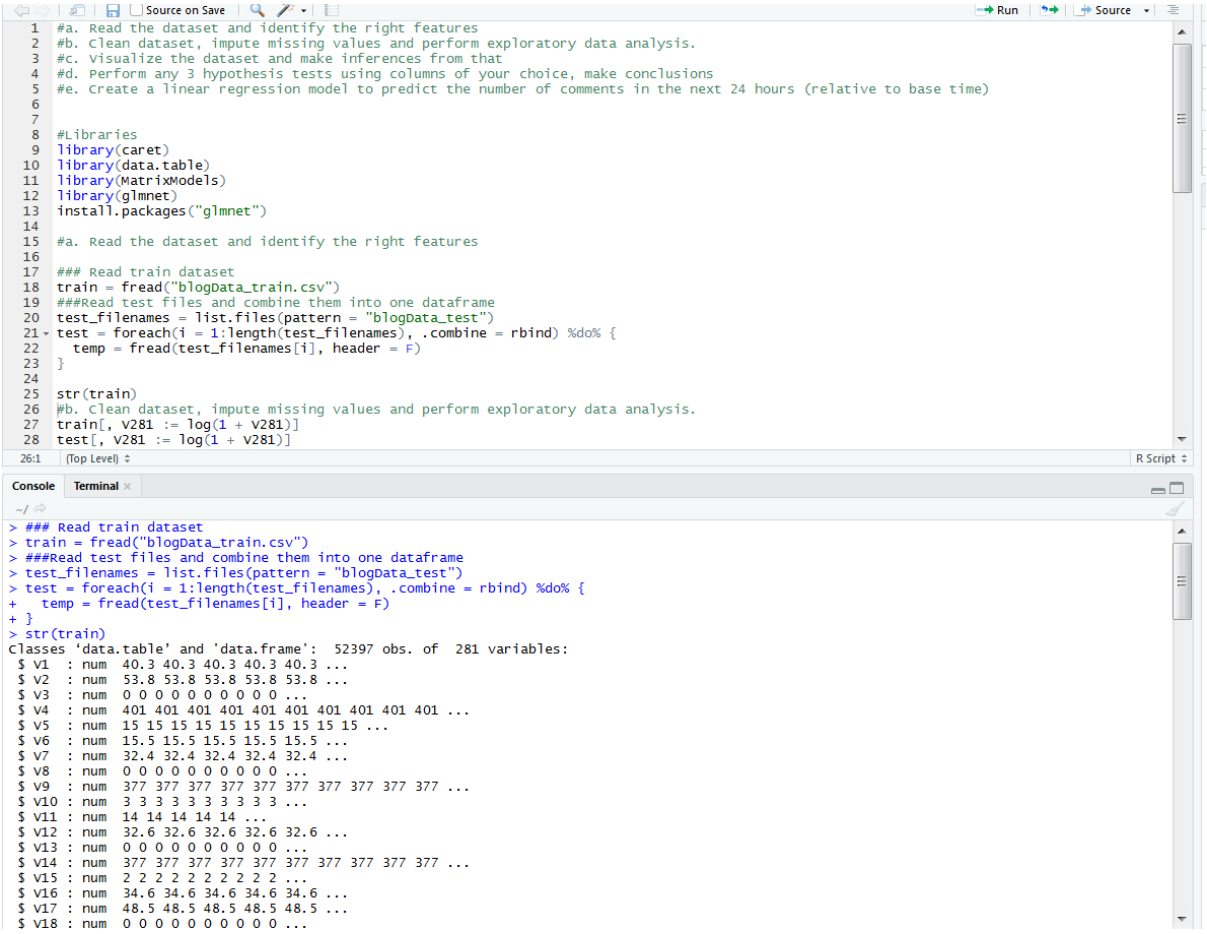
###Read test files and combine them into one dataframe

test_filenames = list.files(pattern = "blogData_test")

test = foreach(i = 1:length(test_filenames), .combine = rbind) %do% {

  temp = fread(test_filenames[i], header = F)

}



```
#a. Read the dataset and identify the right features
#b. Clean dataset, impute missing values and perform exploratory data analysis.
#c. Visualize the dataset and make inferences from that
#d. Perform any 3 hypothesis tests using columns of your choice, make conclusions
#e. Create a linear regression model to predict the number of comments in the next 24 hours (relative to base time)


#Libraries
library(caret)
library(data.table)
library(MatrixModels)
library(glmnet)
install.packages("glmnet")

#a. Read the dataset and identify the right features

### Read train dataset
train = fread("blogData_train.csv")
###Read test files and combine them into one dataframe
test_filenames = list.files(pattern = "blogData_test")
test = foreach(i = 1:length(test_filenames), .combine = rbind) %do% {
  temp = fread(test_filenames[i], header = F)
}

str(train)
#b. Clean dataset, impute missing values and perform exploratory data analysis.
train[, V281 := log(1 + V281)]
test[, V281 := log(1 + V281)]
```

```
> ### Read train dataset
> train = fread("blogData_train.csv")
> ###Read test files and combine them into one dataframe
> test_filenames = list.files(pattern = "blogData_test")
> test = foreach(i = 1:length(test_filenames), .combine = rbind) %do% {
+   temp = fread(test_filenames[i], header = F)
+ }
> str(train)
Classes 'data.table' and 'data.frame':  52397 obs. of  281 variables:
 $ V1  : num  40.3 40.3 40.3 40.3 40.3 ...
 $ V2  : num  53.8 53.8 53.8 53.8 53.8 ...
 $ V3  : num  0 0 0 0 0 0 0 0 0 0 ...
 $ V4  : num  401 401 401 401 401 401 401 401 401 401 ...
 $ V5  : num  15 15 15 15 15 15 15 15 15 15 ...
 $ V6  : num  15.5 15.5 15.5 15.5 15.5 ...
 $ V7  : num  32.4 32.4 32.4 32.4 32.4 ...
 $ V8  : num  0 0 0 0 0 0 0 0 0 0 ...
 $ V9  : num  377 377 377 377 377 377 377 377 377 377 ...
 $ V10 : num  3 3 3 3 3 3 3 3 3 3 ...
 $ V11 : num  14 14 14 14 14 ...
 $ V12 : num  32.6 32.6 32.6 32.6 32.6 ...
 $ V13 : num  0 0 0 0 0 0 0 0 0 0 ...
 $ V14 : num  377 377 377 377 377 377 377 377 377 377 ...
 $ V15 : num  2 2 2 2 2 2 2 2 2 2 ...
 $ V16 : num  34.6 34.6 34.6 34.6 34.6 ...
 $ V17 : num  48.5 48.5 48.5 48.5 48.5 ...
 $ V18 : num  0 0 0 0 0 0 0 0 0 0 ...
```

b.Clean dataset, impute missing values and perform exploratory data

#b. Clean dataset, impute missing values and perform exploratory data analysis.

train[, V281 := log(1 + V281)]

test[, V281 := log(1 + V281)]

drop continous variables without variation

drop = c(8, 13, 28, 33, 38, 40, 43, 50, 278)

train[, (drop) := NULL]

test[, (drop) := NULL]

```
test_filenames = list.files(pattern = "blogData_test")
test = foreach(i = 1:length(test_filenames), .combine = rbind) %do% {
  temp = fread(test_filenames[i], header = F)
}

str(train)
#b. Clean dataset, impute missing values and perform exploratory data analysis.
train[, V281 := log(1 + V281)]
test[, V281 := log(1 + V281)]
# drop continous variables without variation
drop = c(8, 13, 28, 33, 38, 40, 43, 50, 278)
train[, (drop) := NULL]
test[, (drop) := NULL]

#c. Visualize the dataset and make inferences from that

view(train)

#Our target variable for prediction is V281(which is the number of comments in the next 24hrs)
#Other variables are statistcal information (mean,sd,min,max)  of some variables on the oroginal blog posts

#d. Perform any 3 hypothesis tests using columns of your choice, make conclusions

chisq.test(table(train$V1, train$V2))

t.test(train$V4, train$V280, paired=TRUE)


#e. Create a linear regression model to predict the number of comments in the next 24 hours (relative to base time)
mse = function(y_hat, y) {
  mse = mean((y - y_hat)^2)
  return(mse)
}
# create design matrices

train_x = model.Matrix(X0.0.57 ~ . - 1, data = blogTrain, sparse = F)
train_x_sparse = model.Matrix(X0.0.57 ~ . - 1, data = blogTrain, sparse = T)
train_y = blogTrain$X0.0.57

test_x = model.Matrix(V281 ~ . - 1, data = blogTest, sparse = F)
test_y = blogTest$V281

# Linear Model Using LASSO
mdl_lasso = cv.glmnet(train_x_sparse, train_y, family = "gaussian", alpha = 1)
pred_lasso = predict(mdl_lasso, newx = test_x)
```

```
0:11  (Top Level)
nsole  Terminal
#b. Clean dataset, impute missing values and perform exploratory data analysis.
train[, V281 := log(1 + V281)]
test[, V281 := log(1 + V281)]
# drop continous variables without variation
drop = c(8, 13, 28, 33, 38, 40, 43, 50, 278)
train[, (drop) := NULL]
test[, (drop) := NULL]
```

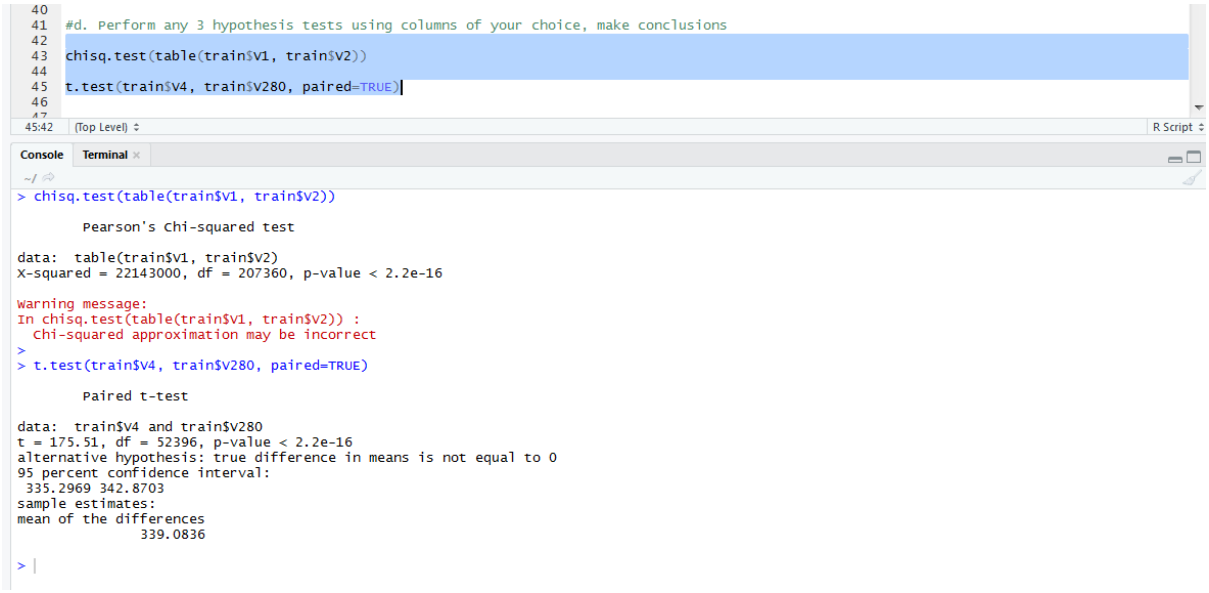
c.Visualize the dataset and make inferences from that

View(train)

#Our target variable for prediction is V281(which is the number of comments in the next 24hrs)

#Other variables are statistcal information (mean,sd,min,max) of some variables on the oroginal blog posts

d.Perform any 3 hypothesis tests using columns of your choice, make



```
#d. Perform any 3 hypothesis tests using columns of your choice, make conclusions

chisq.test(table(train$V1, train$V2))

t.test(train$V4, train$V280, paired=TRUE)
```

```
> chisq.test(table(train$V1, train$V2))

        Pearson's Chi-squared test

data:  table(train$V1, train$V2)
X-squared = 22143000, df = 207360, p-value < 2.2e-16

Warning message:
In chisq.test(table(train$V1, train$V2)) :
  Chi-squared approximation may be incorrect
>
> t.test(train$V4, train$V280, paired=TRUE)

        Paired t-test

data:  train$V4 and train$V280
t = 175.51, df = 52396, p-value < 2.2e-16
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 335.2969 342.8703
sample estimates:
mean of the differences
              339.0836

>
```

P-Value is insignificant on both t.test and chi-square and less than o.o5 so we fail to reject null hypothesis

e.Create a linear regression model to predict the number of comments in the next 24 hours (relative to base time)

Linear Model Using LASSO

I used Lasso and got a success rate of 0.6345 as seen below

Steps Taken below:

#e. Create a linear regression model to predict the number of comments in the next 24 hours (relative to base time)

train = fread("bf-Train.csv")

test = fread("bf-Test.csv")

mse = function(y_hat, y) {

  mse = mean((y - y_hat)^2)

  return(mse)

}

create design matrices

train_x = model.Matrix(V281 ~ . - 1, data = train, sparse = F)

train_x_sparse = model.Matrix(V281 ~ . - 1, data = train, sparse = T)

train_y = train$V281

test_x = model.Matrix(V281 ~ . - 1, data = test, sparse = F)

test_y = test$V281

Linear Model Using LASSO

mdl_lasso = cv.glmnet(train_x_sparse, train_y, family = "gaussian", alpha = 1)

pred_lasso = predict(mdl_lasso, newx = test_x)

mse(pred_lasso, test_y)