Action List

Data exploration, cleaning, and analysis all done in R Studio.

1. Prep
   a. Download the dataset from Kaggle
   b. Import csv file into R Studio
   c. Load the necessary packages

2. Data Exploration
   a. Observing the data
      i. Number of columns and rows
      ii. Number of unique airlines
      iii. Number of tweets each user tweeted
      iv. Number of each sentiment
      v. Summary statistics of airline_sentiment_confidence and airline_sentiment_confidence
      vi. Number of missing variables

   b. Graphing the data
      i. Correlation plot to see which variables are most strongly correlated with airline sentiment confidence
      ii. Scatter plot of sentiment confidence vs negative reason confidence
      iii. Scatter plot of negative reason confidence vs airline sentiment confidence by airline sentiment
      iv. Scatter plot of negative reason confidence vs airline sentiment confidence by negative reason
      v. Histogram of sentiment confidence
      vi. Histogram of negative reason confidence
      vii. Bar graph of top negative reasons

3. Data Cleaning
   a. Drop unnecessary columns
      i. tweet_coord, negativereason_gold, airline_sentiment_gold, tweet_id
   b. Replace N/As in negativereason_confidence column with 0
   c. Create a new column in tweets that holds the number of @ characters in each tweet
   d. Create a new column in tweets that collapses the number of @s
      i. Any tweet with 3+ @s are grouped together
   e. Create a new column in tweets that stores the length of each tweet

4. Analysis
    a. Graphs
        i. Bar graph of airline sentiment based on number of @s
        ii. Bar graph of airline sentiment based on number of @s by airline
        iii. Bar graph of tweet length by company by sentiment
        iv. Density plot of tweet length by airline sentiment
        v. Create subset of tweet sentiment and airline
            1. Barplot of tweet sentiment by airline
    b. Multi-variable Linear Regression
        i. Create train (75%) and test (25%) data set
        ii. Create linear regression model with train data set
            1. Dependent variable: airline sentiment confidence
            2. Independent variables: negative reason confidence, retweet count, @ count, text length
        iii. Create predictions with train and test data set
        iv. Calculate RMSE for both train and test models
        v. Calculate R2 for train lm model
        vi. Check if heteroskedasticity is present
        vii. Check for multicollinearity

Code

```
# Kayelin Santa Elena
# MGSC 410
# HW 1

#-------------------------#
remove(list = ls())

getwd()

# import data
tweets <- read.csv("/Users/kksizzle/Desktop/MGSC 410/HW 1/Tweets.csv")

# get packages
library(ggplot2)
library(dplyr)
library(readr)
library(stringr)
library(ggthemes)
library(tidyr)
```

```
library(caret)
library(rsq)
library(olsrr)
library(RColorBrewer)
library(corrplot)

#--------------------------#

##############------------------ Data Exploration ------------------###############

# dimensions
dim(tweets)

# unique variables
unique(tweets$airline)

# how many tweets each user tweeted
summary(tweets$name)
a <- as.data.frame(table(tweets$name))

# how many of each sentiment
summary(tweets$airline_sentiment)

summary(tweets$airline_sentiment_confidence)
summary(tweets$airline_sentiment_confidence)

# check how many missing variables
sum(is.na(tweets))

### Which variables are most strongly correlated with airline sentiment confidence
# negative reason confidence and @ count

nums2 <- sapply(tweets, is.numeric) # names of numeric variables
cormat2 <- cor(tweets[,nums2], use="complete.obs")
print(cormat2[,"airline_sentiment_confidence"])
corrplot(cormat2)

#------------ Graphs --------------#

cbp2 <- c("#FDAE6B", "#999999", "#0072B2")
```

```
densityColor <- c('#f93822','#fedd00','#27e833')

# scatter plot of sentiment confidence vs negative reason confidence
ggplot(tweets, aes(airline_sentiment_confidence, negativereason_confidence)) +
  geom_point(color = "#FDAE6B") +
  geom_smooth(method = "lm", color = "#0072B2") +
  labs(x = "Airline Sentiment Confidence", y = "Negative Reason Confidence", title = "Airline
Sentiment Confidence vs Negative Reason Confidence")

# Negative Reason Confidence vs Airline Sentiment Confidence by Airline Sentiment
ggplot(tweets, aes(airline_sentiment_confidence, negativereason_confidence)) +
  geom_point(aes(color = tweets$airline_sentiment)) +
  scale_color_manual(values = cbp2, name = 'Airline\nSentiment') +
  labs(x = 'Airline Sentiment Confidence', y = 'Negative Reason Confidence')

# Negative Reason Confidence vs Airline Sentiment Confidence by Negative Reason
ggplot(tweets, aes(airline_sentiment_confidence, negativereason_confidence)) +
  geom_point(aes(color = tweets$negativereason)) +
  labs(x = 'Airline Sentiment Confidence', y = 'Negative Reason Confidence', color = "Negative
Reason")

# histogram of sentiment confidence
ggplot(tweets, aes(airline_sentiment_confidence)) +
  geom_histogram(bins=10, fill = "#FDAE6B") +
  labs(x = "Airline Sentiment Confidence", y = "Count", title = "Airline Sentiment Confidence")

# histogram of negative reason confidence
ggplot(tweets, aes(negativereason_confidence)) +
  geom_histogram(bins=10, fill = "#FDAE6B") +
  labs(x = "Negative Reason Confidence", y = "Count", title = "Negative Reason Confidence")

# bar graph top negative reasons
negtweets <- tweets %>% filter(negativereason != "")
ggplot(negtweets, aes(negativereason)) +
  geom_bar(fill = "#D55E00") +
  labs(x = "Negative Reason", y = "Count", title = "Top Negative Reasons") +
  theme(axis.text.x = element_text(angle=65, vjust=0.6))

##############------------------ Data Cleaning ----------------###############
```

```
# drop unnecessary columns
drop <- c('tweet_coord','negativereason_gold','airline_sentiment_gold','tweet_id')
tweets <- tweets[,!(names(tweets) %in% drop)]

# replace n/as with 0
tweets$negativereason_confidence[which(is.na(tweets$negativereason_confidence))] <- 0


# Create a variable holding the number of @ characters in each tweet
tweets$at_count <- sapply(tweets$text, function(x) str_count(x, '@'))

# Collapse number of @
tweets$at_count2[tweets$at_count == 1] <- '1'
tweets$at_count2[tweets$at_count == 2] <- '2'
tweets$at_count2[tweets$at_count %in% c(3:max(tweets$at_count))] <- '3+'

# Change to a factor variable
tweets$at_count2 <- factor(tweets$at_count2)

# Store the length of each tweet
tweets$text_length <- sapply(tweets$text, function(x) nchar(as.character(as.factor(x))))

##############------------------ More Graphs ------------------###############

sentBreaks  <- c('negative','neutral','positive')

# Airline Sentiment Based on Number of @s
ggplot(tweets, aes(x = at_count2, fill = airline_sentiment)) +
  geom_bar(position = 'fill') +
  scale_fill_manual(name   = 'Airline\nSentiment',
                    values = cbp2,
                    breaks = sentBreaks) +
  labs(x = 'Number of @s', y = 'Proportion', title = "Airline Sentiment Based on Number of @s")
+
  theme(text = element_text(size=12))

# Airline Sentiment Based on Number of @s by Airline
ggplot(tweets, aes(x = at_count2, fill = airline_sentiment)) +
  geom_bar(position = 'fill') +
  facet_wrap(~airline) +
```

```
  scale_fill_manual(name   = 'Airline\nSentiment',
                  values = cbp2,
                  breaks = sentBreaks) +
  labs(x = 'Number of @s', y = 'Proportion', title = "Airline Sentiment Based on Number of @s
by Airline") +
  theme(text = element_text(size=12))

# bar graph of tweet length by company by sentiment
ggplot(tweets, aes(x = text_length, fill = airline_sentiment)) +
  geom_bar() +
  facet_wrap(. ~ airline) +
  scale_fill_manual(name   = 'Airline\nSentiment',
                  values = cbp2,
                  breaks = sentBreaks)

# Density plot of tweet length by airline sentiment
ggplot(tweets, aes(x = text_length, fill = airline_sentiment)) +
  geom_density(alpha = 0.2) +
  facet_wrap(~airline, scale = 'free') +
  scale_fill_manual(name   = 'Airline\nSentiment',
                  values = densityColor,
                  breaks = sentBreaks) +
  labs(x = 'Tweet Length', y = "Density")


# Tweet Sentiment by Airline
airlineSentiment <- as.data.frame(table(tweets$airline, tweets$airline_sentiment))
colnames(airlineSentiment) <- c('Airline', 'Sentiment', 'Freq')

ggplot(airlineSentiment, aes(x=Airline, y=Freq, fill=Sentiment)) +
  scale_fill_manual(values = cbp2, name   = 'Airline\nSentiment') +
  labs(y = 'Number of Tweets', x = 'Airline', title = "Tweet Sentiment by Airline") +
  geom_bar(stat = 'identity')


##############------------------ Multi Variable Linear Regression
------------------###############

### test / train data
```

```
set.seed(410)
index <- sample(1:nrow(tweets),size=0.75*nrow(tweets),replace=FALSE)
train <- tweets[index,]
test <- tweets[-index,]


### linear regression model - predicting airline sentiment confidence
mod1_lm_train <- lm(airline_sentiment_confidence ~ negativereason_confidence +
            retweet_count + at_count + text_length,
              data = train)
summary(mod1_lm_train)
coefficients(mod1_lm_train)

mod2_lm_train <- lm(airline_sentiment_confidence ~ negativereason_confidence +
                retweet_count + at_count + text_length,
               data = test)
summary(mod2_lm_train)

### Predictions
#train
preds_train1 <- predict(mod1_lm_train)
preds_train_df1 <- data.frame(true = train$airline_sentiment_confidence, pred = preds_train1,
resid = mod1_lm_train$residuals)

#test
preds_test1 <-  predict(mod1_lm_train, newdata = test)
preds_test_df1 <- data.frame(true = test$airline_sentiment_confidence, pred = preds_test1)

### model accuracy: RMSE and R2
#There is not much of an overfitting issue since there is no big difference between the RSMEs.
#R2 is very low, the model didn't score too well.

# train RMSE
RMSE(preds_train_df1$pred, preds_train_df1$true)

# test RMSE
RMSE(preds_test_df1$pred, preds_test_df1$true)

# R2
rsq(mod1_lm_train)
```

```
rsq(mod2_lm_train)

### heteroskedasticity
ggplot(preds_train_df1, aes(pred, resid)) +
  geom_point(color = "#FDAE6B") +
  geom_smooth(method = "lm", color = "#0072B2") +
  labs(x = "Predicted Airline Sentiment Confidence", y = "Residual")
# There are signs of heteroskedasticity which may contribute to the low R2.

### collinearity
# VIF > 10 indicates problematic level of multicollinearity.
# There is a collinearity issue.
ols_vif_tol(mod1_lm_train)
```