Homework1_MGSC410

October 5, 2023

1 MGSC 410 Homework 1 - Twitter US Airline Sentiment

[ ]:

```
import warnings
warnings.filterwarnings('ignore')

# data and plotting
import pandas as pd
import numpy as np
import random
from plotnine import *
from tabulate import tabulate

import sklearn
import matplotlib.pyplot as plt
import matplotlib.colors as mcolors
%matplotlib inline

from wordcloud import WordCloud,STOPWORDS
from wordcloud import ImageColorGenerator
```

1.1 Data Preprocessing/Assessment

[ ]:

```
# Loading in and previewing the data
data = pd.read_csv('https://raw.githubusercontent.com/ryanking916/Data/main/
 ↪Tweets.csv')
data.head()
```

[ ]:

```
             tweet_id airline_sentiment  airline_sentiment_confidence  \
0  570306133677760513           neutral                        1.0000
1  570301130888122368          positive                        0.3486
2  570301083672813571           neutral                        0.6837
3  570301031407624196          negative                        1.0000
4  570300817074462722          negative                        1.0000

  negativereason  negativereason_confidence         airline  \
0            NaN                        NaN  Virgin America
1            NaN                     0.0000  Virgin America
```

```
2            NaN                         NaN  Virgin America
3     Bad Flight                      0.7033  Virgin America
4     Can't Tell                      1.0000  Virgin America

  airline_sentiment_gold        name negativereason_gold  retweet_count  \
0                    NaN     cairdin                 NaN              0
1                    NaN    jnardino                 NaN              0
2                    NaN  yvonnalynn                 NaN              0
3                    NaN    jnardino                 NaN              0
4                    NaN    jnardino                 NaN              0

                                                text tweet_coord  \
0                @VirginAmerica What @dhepburn said.         NaN
1  @VirginAmerica plus you've added commercials t…         NaN
2  @VirginAmerica I didn't today… Must mean I n…         NaN
3  @VirginAmerica it's really aggressive to blast…         NaN
4  @VirginAmerica and it's a really big bad thing…         NaN

               tweet_created tweet_location               user_timezone
0  2015-02-24 11:35:52 -0800            NaN  Eastern Time (US & Canada)
1  2015-02-24 11:15:59 -0800            NaN  Pacific Time (US & Canada)
2  2015-02-24 11:15:48 -0800      Lets Play  Central Time (US & Canada)
3  2015-02-24 11:15:36 -0800            NaN  Pacific Time (US & Canada)
4  2015-02-24 11:14:45 -0800            NaN  Pacific Time (US & Canada)
```

[ ]:

```python
# Printing the shape of our current dataframe
print("The shape of the dataframe is: ", data.shape)
```

```
The shape of the dataframe is:  (14640, 15)
```

[ ]:

```python
# Checking for null values
data.isnull().sum()
```

[ ]:

```
tweet_id                            0
airline_sentiment                   0
airline_sentiment_confidence        0
negativereason                   5462
negativereason_confidence        4118
airline                             0
airline_sentiment_gold          14600
name                                0
negativereason_gold             14608
retweet_count                       0
text                                0
tweet_coord                     13621
tweet_created                       0
tweet_location                   4733
```

```
user_timezone                    4820
dtype: int64
```

Since there are so many null values in the categories: **airline_sentiment_gold**, **negativereason_gold**, and **tweet_cord**, we will delete those columns

[ ]:

```
# Deleting columns that are not needed
del data['airline_sentiment_gold']
del data['negativereason_gold']
del data['tweet_coord']
```

[ ]:

```
# Changing tweet_created from date time to date
data['tweet_created'] = pd.to_datetime(data['tweet_created']).dt.date
```

[ ]:

```
filtered_data = data[data['airline'] == 'Delta']
jetblue_count = filtered_data['text'].str.contains('JetBlue', case=False,␣
 ↪na=False).sum()

print(f'Number of texts containing "JetBlue" with airline "Delta":␣
 ↪{jetblue_count}')
```

```
Number of texts containing "JetBlue" with airline "Delta": 2218
```

[ ]:

```
# Changing 'Delta' to 'JetBlue'
data.loc[data['airline'] == 'Delta', 'airline'] = 'JetBlue'
```

[ ]:

```
data.head()
```

[ ]:

```
             tweet_id airline_sentiment  airline_sentiment_confidence  \
0  570306133677760513           neutral                        1.0000
1  570301130888122368          positive                        0.3486
2  570301083672813571           neutral                        0.6837
3  570301031407624196          negative                        1.0000
4  570300817074462722          negative                        1.0000

  negativereason  negativereason_confidence         airline        name  \
0            NaN                        NaN  Virgin America     cairdin
1            NaN                     0.0000  Virgin America    jnardino
2            NaN                        NaN  Virgin America  yvonnalynn
3     Bad Flight                     0.7033  Virgin America    jnardino
4     Can't Tell                     1.0000  Virgin America    jnardino

   retweet_count                                               text  \
0              0                @VirginAmerica What @dhepburn said.
1              0  @VirginAmerica plus you've added commercials t…
2              0  @VirginAmerica I didn't today… Must mean I n…
3              0  @VirginAmerica it's really aggressive to blast…
```

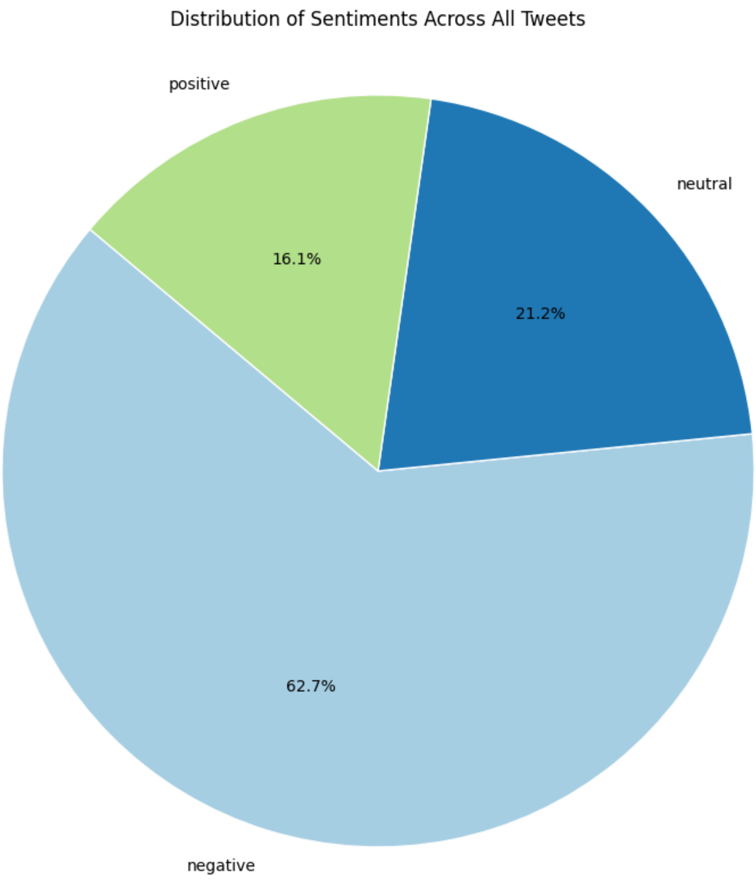
```
4             0  @VirginAmerica and it's a really big bad thing…

  tweet_created tweet_location               user_timezone
0    2015-02-24            NaN  Eastern Time (US & Canada)
1    2015-02-24            NaN  Pacific Time (US & Canada)
2    2015-02-24      Lets Play  Central Time (US & Canada)
3    2015-02-24            NaN  Pacific Time (US & Canada)
4    2015-02-24            NaN  Pacific Time (US & Canada)
```

1.2 Data Understanding

1.2.1 Count of Sentiments per Airline

[ ]:

```
# Counting the number of each sentiment
sentiment_counts = data['airline_sentiment'].value_counts()

# Defining a color palette
colors = plt.cm.Paired(range(len(sentiment_counts)))

# Creating a pie chart
plt.figure(figsize=(10, 8))  # making the plot a bit larger

# Drawing the pie chart
plt.pie(sentiment_counts, labels=sentiment_counts.index,
        autopct='%1.1f%%', startangle=140, colors=colors,␣
 ↪wedgeprops=dict(edgecolor='w'))

# Title
plt.title('Distribution of Sentiments Across All Tweets', pad=20)  # pad␣
 ↪adjusts the position of the title.

# Ensuring the pie chart is a circle
plt.axis('equal')

# Displaying the plot
plt.tight_layout()
plt.show()
```

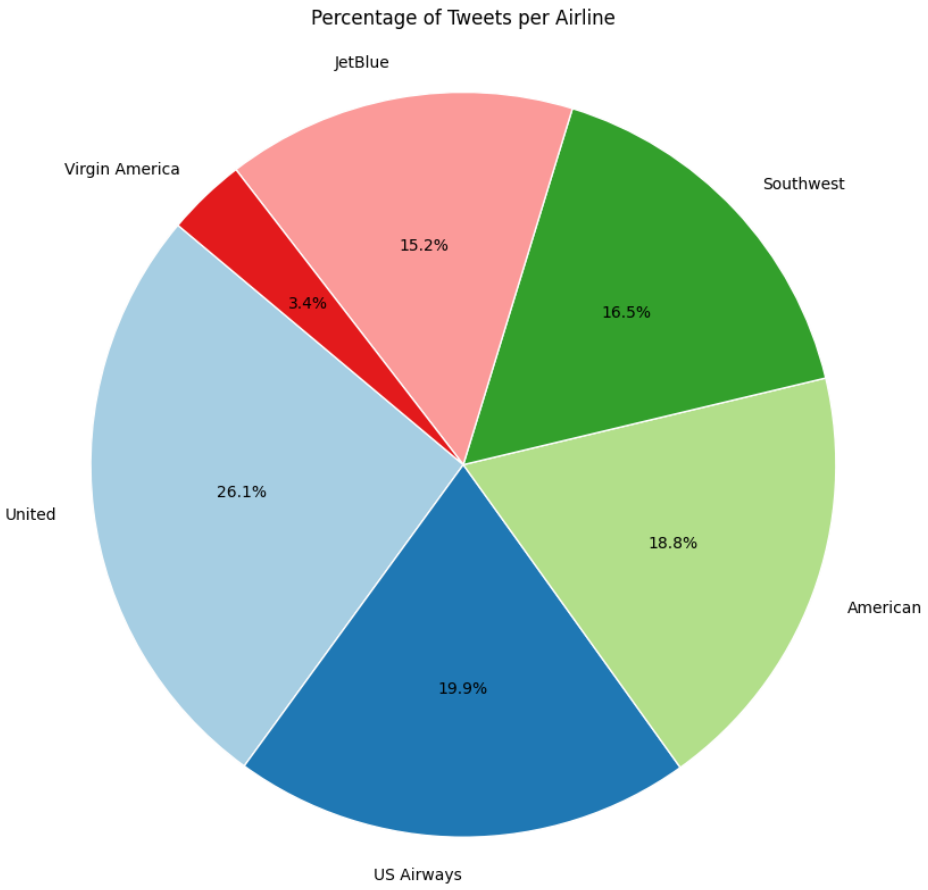


```
[ ]: data['airline'].value_counts()
```

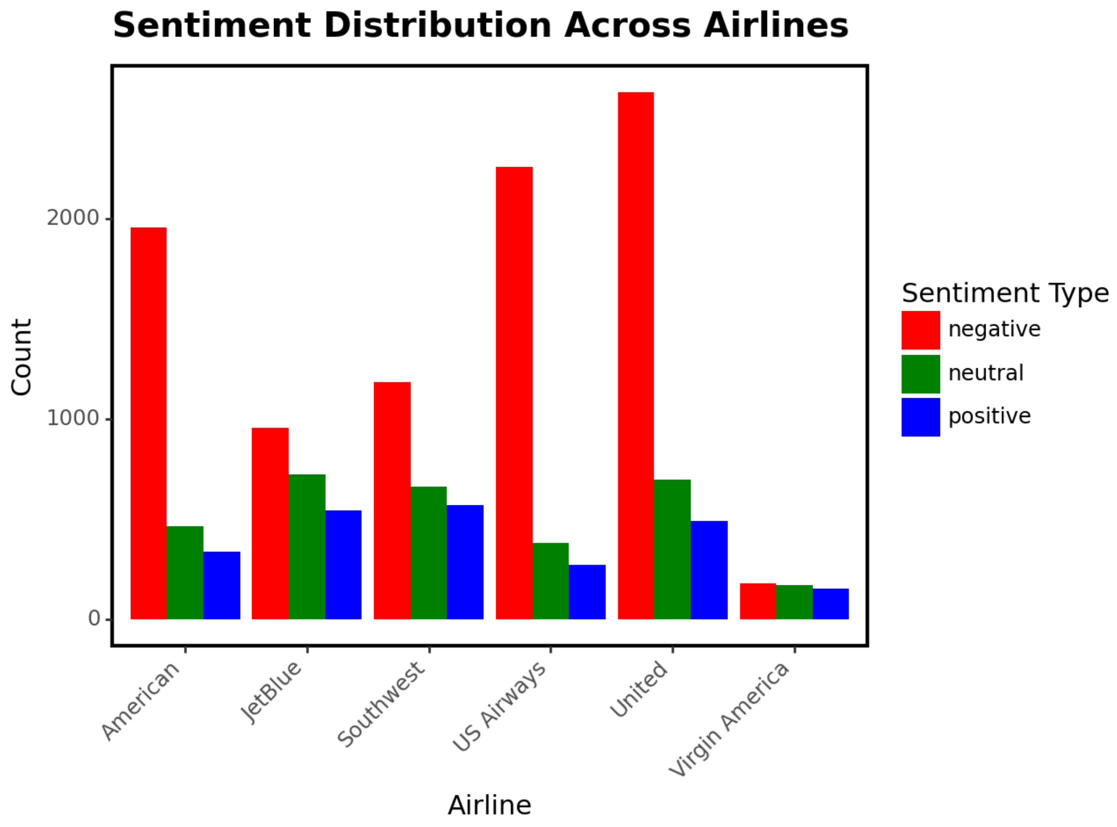
```
[ ]: United            3822
     US Airways        2913
     American          2759
     Southwest         2420
     JetBlue           2222
     Virgin America     504
     Name: airline, dtype: int64
```

```
[ ]: # Calculating the count of each airline's tweets
     airline_counts = data['airline'].value_counts().reset_index()
     airline_counts.columns = ['airline', 'count']

     # Defining a color palette
     colors = plt.cm.Paired(range(len(airline_counts)))

     # Creating a pie chart
     plt.figure(figsize=(10, 8))  # making the plot a bit larger
```

```
# Drawing the pie chart
plt.pie(airline_counts['count'], labels=airline_counts['airline'],
        autopct='%1.1f%%', startangle=140, colors=colors,␣
 ↪wedgeprops=dict(edgecolor='w'))
# Title
plt.title('Percentage of Tweets per Airline', pad=20)  # pad adjusts the␣
 ↪position of the title.
# Ensuring the pie chart is a circle
plt.axis('equal')

# Displaying the plot
plt.tight_layout()
plt.show()
```



```
[ ]: plot = (
         ggplot(data, aes(x='airline', fill='airline_sentiment')) +
         geom_bar(stat='count', position='dodge', show_legend=True) +
         labs(title='Sentiment Distribution Across Airlines',
              x='Airline',
              y='Count',
              fill='Sentiment Type') +
         theme(
             axis_text_x=element_text(rotation=45, hjust=1),
             plot_title=element_text(size=14, face="bold"),
             panel_grid_major=element_blank(),  # removes major grid
             panel_grid_minor=element_blank(),  # removes minor grid
             panel_background=element_blank(),   # removes background
             panel_border=element_rect(colour="black", fill=None, size=1.5)  # adds border around plot
         ) +
         scale_fill_manual(values=['red', 'green', 'blue'])  # specify colors
     )

     print(plot)
```



The graphs produced above display that American, US Airways, and United airlines mainly get negative reactions from passengers. Southwest, Virgin America, and Delta airlines are more balanced but still have the most sentiments in the negative columns.

1.2.2 Insights

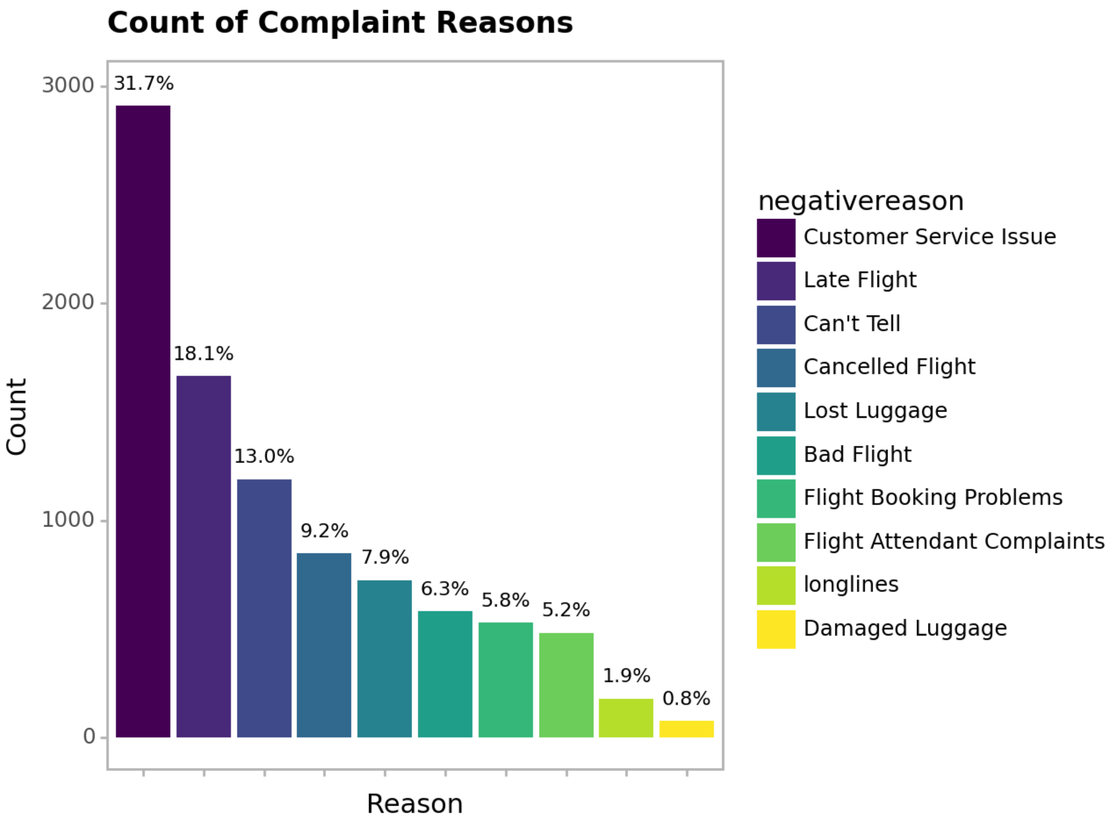
People normally tend to give more weight to negative experiences than positive ones. This phenomenon, known as negativity bias,is most likely the reason why individuals tend to share thier negatives experiences with their Twitter audience.

Another reason for why the majority of the sentiments are negative is traveling with airlines is very expensive so people have high expectations. When these expectations are not met, it can lead to dissatisfaction and negative feedback from passengers. A lot of times there is one specific reason that upsets passengers which will be discussed in the next section.

1.2.3 Count of Complaint Reasons

[ ]:

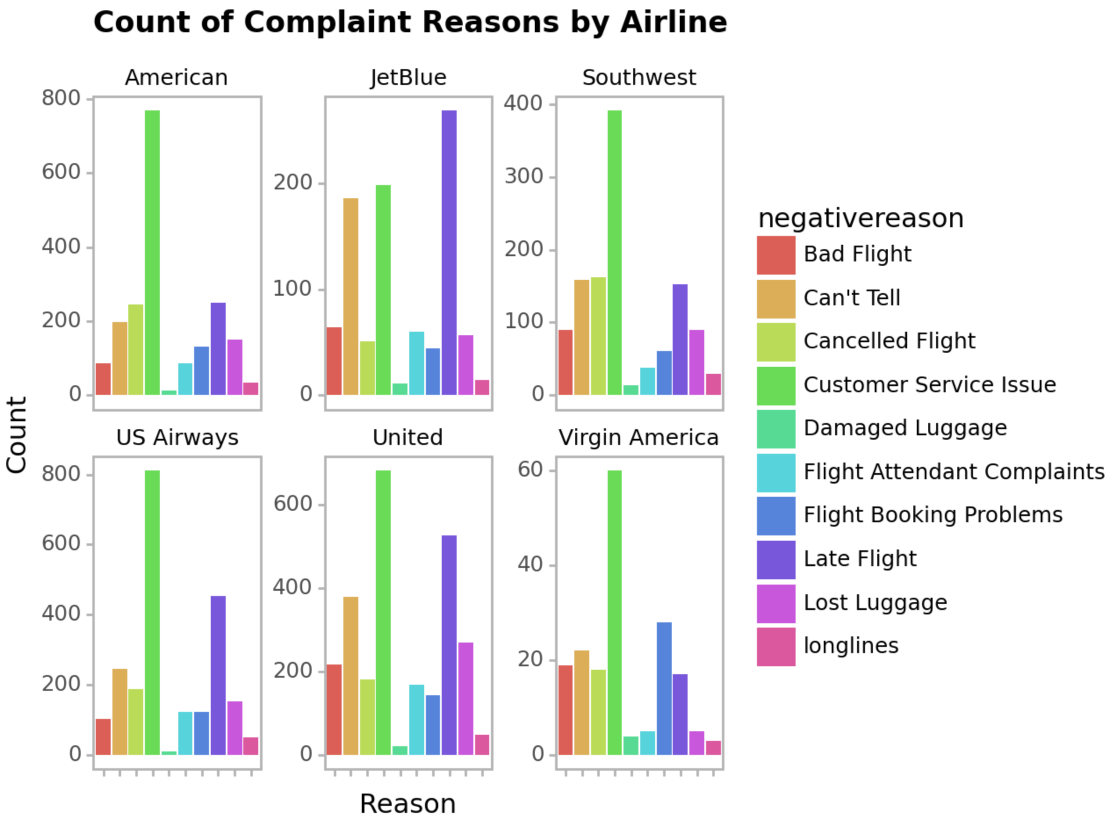
```
# Adding a new column to data that contains the percentage values.
total = reason_counts['counts'].sum()
reason_counts['percentage'] = (reason_counts['counts'] / total) * 100

# Create the plot
plot2 = (
    ggplot(reason_counts, aes(x='negativereason', y='counts',␣
 ↪fill='negativereason')) +
    geom_bar(stat='identity') +
    geom_text(
        aes(label='round(percentage, 1).astype(str) + "%"'),  # add percentage␣
 ↪sign
        va='bottom',  # vertical alignment
        nudge_y=reason_counts['counts'].max() * 0.02,  # adjust nudging to␣
 ↪avoid overlap with bars
        size=8  # adjust size of the text
    ) +
    labs(title='Count of Complaint Reasons', x='Reason', y='Count') +
    theme_light() +
    theme(axis_text_x=element_blank(),
          panel_grid_major=element_blank(),
          panel_grid_minor=element_blank(),
          plot_title=element_text(size=12, face="bold")
    )
)

print(plot2)
```



Takeaways - Customer service, late flight, and cancelled flight are the three main reasons why customers wrote negative tweets towards the airlines.

[ ]:

```
# Filter
reason_counts = data.groupby(['airline', 'negativereason']).size().
 ↪reset_index(name='counts')

# Now using ggplot to create the visualization
plot2 = (ggplot(reason_counts, aes(x='negativereason', y='counts',␣
 ↪fill='negativereason')) +
        geom_bar(stat='identity') +
        facet_wrap('~ airline', scales='free_y') +
        labs(title='Count of Complaint Reasons by Airline', x='Reason',␣
 ↪y='Count') +
        theme_light() +
        theme(axis_text_x=element_blank(),
              strip_text_x=element_text(size=9,color="black"),
              panel_grid_major=element_blank(),
              panel_grid_minor=element_blank(),
              plot_title=element_text(size=12, face="bold"),
              strip_background=element_blank()))

print(plot2)
```



The number one total negative reason throughout all airlines is customer service. This reason dominates the others for every airline except Delta. Delta's main complaint is the late flights. United is also another airline that struggles with late flights.

1.2.4 Negative Sentiments & Dates

[ ]:
```
# Using groupby to get the info from the dataset that we want
day_data = data.groupby(['tweet_created','airline','airline_sentiment']).size()

# Displaying information
day_data.unstack()
```

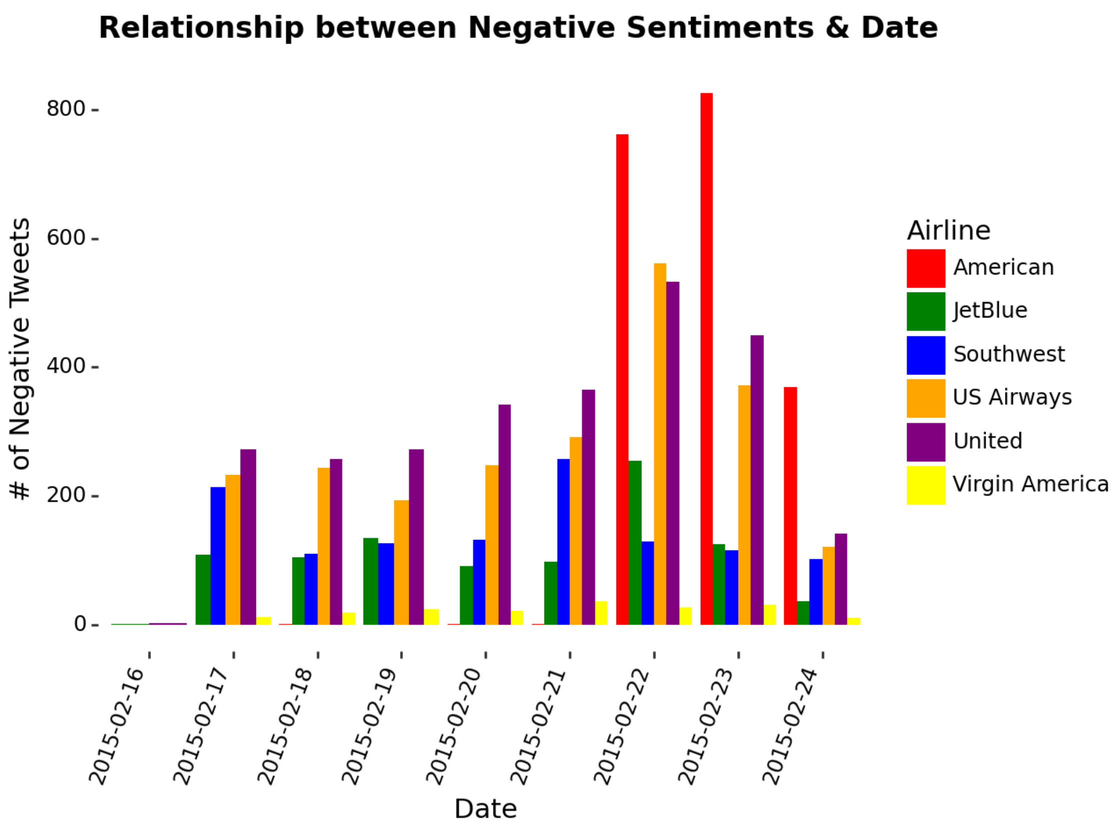
[ ]:
```
airline_sentiment              negative  neutral  positive
tweet_created airline
```

```
2015-02-16    JetBlue              1.0      1.0       NaN
              United               2.0      NaN       NaN
2015-02-17    JetBlue            108.0     86.0      69.0
              Southwest          213.0     85.0      86.0
              US Airways         233.0     30.0      48.0
              United             272.0     75.0      49.0
              Virgin America      12.0     21.0      21.0
2015-02-18    American             1.0      NaN       NaN
              JetBlue            105.0     86.0      77.0
              Southwest          110.0    106.0      76.0
              US Airways         244.0     32.0      41.0
              United             257.0     90.0      59.0
              Virgin America      19.0     21.0      20.0
2015-02-19    American             NaN      NaN       1.0
              JetBlue            135.0     70.0      78.0
              Southwest          127.0     94.0      96.0
              US Airways         193.0     54.0      32.0
              United             272.0     85.0      69.0
              Virgin America      24.0     26.0      20.0
2015-02-20    American             1.0      NaN       NaN
              JetBlue             91.0     90.0      70.0
              Southwest          132.0    110.0      77.0
              US Airways         248.0     52.0      33.0
              United             342.0     99.0      85.0
              Virgin America      21.0     32.0      17.0
2015-02-21    American             1.0      NaN       NaN
              JetBlue             98.0     79.0      66.0
              Southwest          257.0     60.0      53.0
              US Airways         291.0     39.0      30.0
              United             365.0     88.0      53.0
              Virgin America      37.0     12.0      28.0
2015-02-22    American           762.0    132.0      94.0
              JetBlue            255.0     76.0      77.0
              Southwest          129.0     77.0      73.0
              US Airways         561.0     60.0      27.0
              United             532.0    102.0      69.0
              Virgin America      27.0     16.0      10.0
2015-02-23    American           826.0    178.0     137.0
              JetBlue            125.0    195.0      71.0
              Southwest          116.0     83.0      77.0
              US Airways         372.0     74.0      42.0
              United             449.0    109.0      83.0
              Virgin America      31.0     37.0      23.0
2015-02-24    American           369.0    153.0     104.0
              JetBlue             37.0     40.0      36.0
              Southwest          102.0     49.0      32.0
              US Airways         121.0     40.0      16.0
```

```
United              142.0      49.0       25.0
Virgin America       10.0       6.0       13.0
```

[ ]:

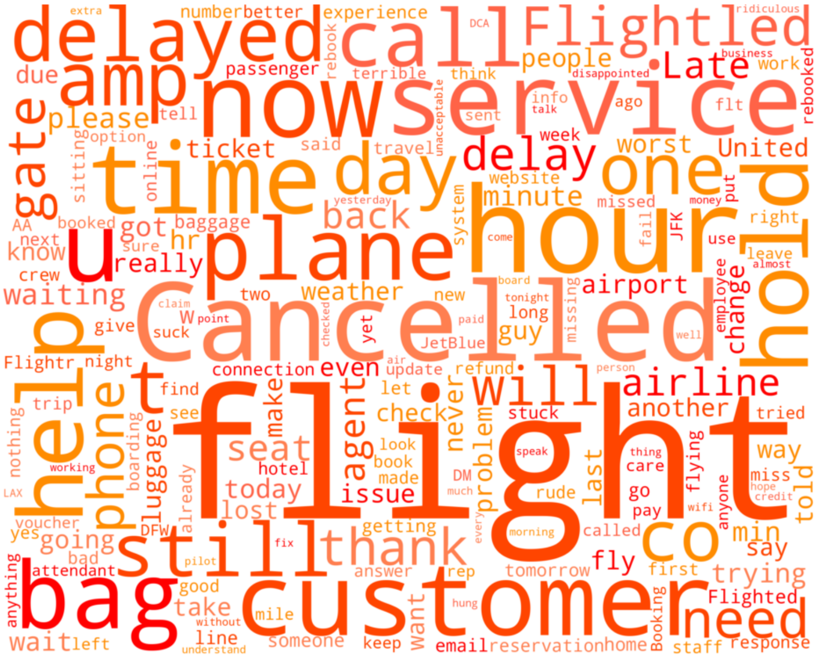
```
# Filter only sentiments that are negative
day_data = day_data.loc(axis=0)[:,:,'negative']

# Convert the Series to a DataFrame and rename the count column
day_data = day_data.reset_index().rename(columns={0: 'negative_count'})

# Plotting the graph
plot = (ggplot(day_data, aes(x='tweet_created', y='negative_count',␣
 ↪fill='airline')) +
        geom_bar(stat="identity", position="dodge") +
        theme(axis_text_x=element_text(rotation=70, hjust=1)) +
        labs(title='Relationship between Negative Sentiments & Date', x='Date',␣
 ↪y='# of Negative Tweets', fill='Airline') +
        scale_fill_manual(values=['red', 'green', 'blue', 'orange', 'purple',␣
 ↪'yellow']) +
        theme(panel_background=element_blank(),
        panel_grid_major=element_blank(),
        panel_grid_minor=element_blank(),
        axis_text=element_text(color="black"),
        axis_text_x=element_text(rotation=70, hjust=1, color="black"),
        axis_text_y=element_text(color="black"),
        plot_title=element_text(size=12, face="bold"),
        legend_title=element_text()
        )
)

print(plot)
```



[ ]:

```
# Filtering data to get rows where the sentiment is negative
negative_sentiment_data = data[data['airline_sentiment']=='negative']
words = ' '.join(negative_sentiment_data['text'])

# Cleaned the words to remove words starting with '@' basically removing␣
  ↪airline names that were tagged
text = " ".join([word for word in words.split()
                if not word.startswith('@')
                 ])

# Generating word cloud from negative sentiments
word_cloud = WordCloud(collocations=False,background_color='white', width=2500,
                      height=2000).generate(text)

def red_shades_color_func(*args, **kwargs):
    red_shades = ["#FF0000", "#FF4500", "#FF6347", "#FF7F50", "#FF8C00"]
    return random.choice(red_shades)

# Generating word cloud from negative sentiments
```

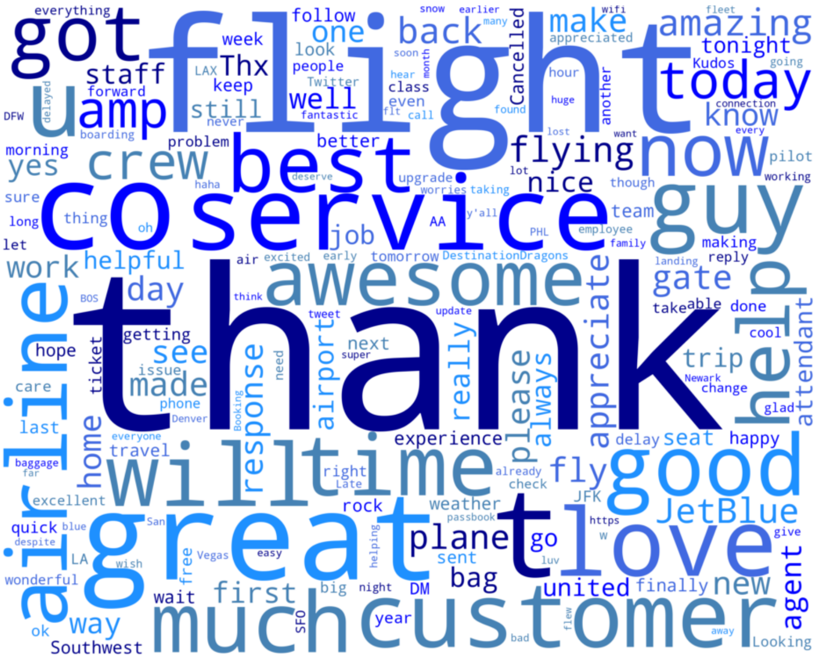
```
word_cloud = WordCloud(collocations=False, background_color='white', width=2500,
                       height=2000).generate(text)

# Applying the red_shades_color_func to the word cloud
word_cloud.recolor(color_func=red_shades_color_func)

# Displaying word cloud
plt.imshow(word_cloud, interpolation="bilinear")
plt.axis('off')
plt.show()
```



Common Words: delayed, service, flight, cancelled, customer, hour, time, help, hold, bag,weather, lategate, call

```
[ ]: # Filtering data to get rows where the sentiment is positive
positive_sentiment_data = data[data['airline_sentiment']=='positive']
words = ' '.join(positive_sentiment_data['text'])

# Cleaned the words to remove words starting with '@' (removing airline names␣
 ↪that were tagged)
text = " ".join([word for word in words.split() if not word.startswith('@')])
```

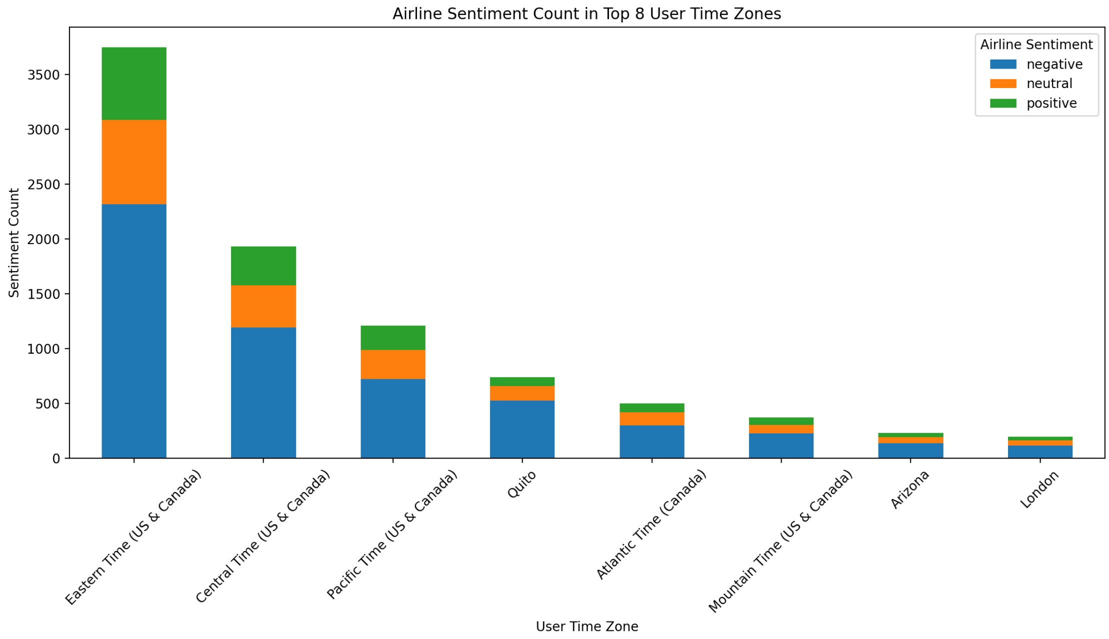
```
# Define a function that returns various shades of blue
def blue_shades_color_func(*args, **kwargs):
    blue_shades = ["#0000FF", "#00008B", "#1E90FF", "#4169E1", "#4682B4"]
    return random.choice(blue_shades)

# Generating word cloud from positive sentiments
word_cloud = WordCloud(collocations=False, background_color='white',␣
 ↪width=2500, height=2000).generate(text)

# Applying the blue_shades_color_func to the word cloud
word_cloud.recolor(color_func=blue_shades_color_func)

# Displaying word cloud
plt.imshow(word_cloud, interpolation="bilinear")
plt.axis('off')
plt.show()
```



Common Words: Thank, time, flight, great, awesome, help, love, service, Crew, cancelled, good, best, appreciate, guy,got, staff

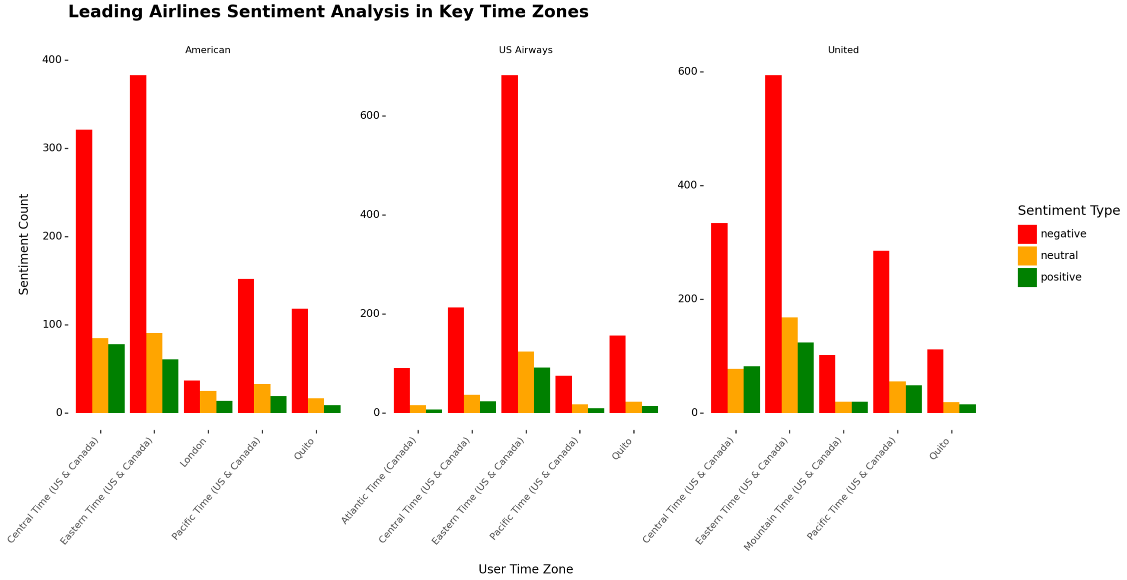
[ ]:

```
# Get the top 8 user timezones in terms of frequency
top_timezones = data['user_timezone'].value_counts().head(8).index

# Filter data for only these top 8 timezones
filtered_data = data[data['user_timezone'].isin(top_timezones)]

# Pivot the data to get sentiment counts per timezone
pivot_data = pd.crosstab(index=filtered_data['user_timezone'],␣
  ↪columns=filtered_data['airline_sentiment'])

# Add a "Total" column and sort by it
pivot_data['Total'] = pivot_data.sum(axis=1)
pivot_data = pivot_data.sort_values(by='Total', ascending=False)

# Plotting
ax = pivot_data[['negative', 'neutral', 'positive']].plot(kind='bar',␣
  ↪stacked=True, figsize=(12, 7))

plt.title('Airline Sentiment Count in Top 8 User Time Zones')
plt.xlabel('User Time Zone')
plt.ylabel('Sentiment Count')
plt.xticks(rotation=45)
plt.tight_layout()
plt.legend(title='Airline Sentiment')
plt.grid(False)
plt.show()
```



[ ]:

```
# Filtering to get top 3 airlines
selected_airlines = ['United', 'US Airways', 'American']
filtered_data = data[data['airline'].isin(selected_airlines)]

# Get the top 5 time zones for each airline based on sentiment counts
top_timezones_data = pd.DataFrame()

for airline in selected_airlines:
    top_zones = (
        filtered_data[filtered_data['airline'] == airline]
        .groupby('user_timezone')['airline_sentiment']
        .count()
        .nlargest(5)
        .index
    )
    airline_data = filtered_data[
        (filtered_data['airline'] == airline) & (filtered_data['user_timezone'].
 ↪isin(top_zones))
    ]
    top_timezones_data = pd.concat([top_timezones_data, airline_data])


plot2 = (
    ggplot(top_timezones_data, aes(x='user_timezone',␣
 ↪fill='airline_sentiment')) +
    geom_bar(stat='count', position='dodge', show_legend=True) +
    facet_wrap('~ airline', scales='free', ncol=3) +
    labs(title='Leading Airlines Sentiment Analysis in Key Time Zones',
         x='User Time Zone',
         y='Sentiment Count',
         fill='Sentiment Type') +
    theme(
        axis_text_x=element_text(rotation=50, hjust=1, size=8),
        strip_text_x=element_text(size=8, color="black"),
        axis_text_y=element_text(color="black"),
        plot_title=element_text(size=14, face="bold"),
        strip_background=element_blank(),
        figure_size=(13, 7),
        axis_title=element_text(size=10),
        subplots_adjust={'wspace': 0.25},
        # Add/modify these to remove gridlines and have a blank background
        panel_background=element_blank(),  # No background
        panel_grid_major=element_blank(),  # No major gridlines
        panel_grid_minor=element_blank(),  # No minor gridlines
        panel_border=element_blank(),  # No border
    ) +
    scale_fill_manual(values=['red', 'orange', 'green'])
)

# Display plot
plot2
```



```
[ ]: <Figure Size: (1300 x 700)>
```

```
[ ]: # doesn't show this cells output when downloading PDF
!pip install gwpy &> /dev/null

# installing necessary files
!apt-get install texlive texlive-xetex texlive-latex-extra pandoc
!sudo apt-get update
!sudo apt-get install texlive-xetex texlive-fonts-recommended␣
 ↪texlive-plain-generic

# installing pypandoc
!pip install pypandoc

# connecting your google drive
from google.colab import drive
drive.mount('/content/drive')

# copying your file over. Change "Class6-Completed.ipynb" to whatever your file␣
 ↪is called (see top of notebook)
!cp "drive/My Drive/Colab Notebooks/Homework1_MGSC410.ipynb" ./
```

```
# Again, replace "Class6-Completed.ipynb" to whatever your file is called (see␣
 ↪top of notebook)
!jupyter nbconvert --to PDF "Homework1_MGSC410.ipynb"
```

```
Reading package lists… Done
Building dependency tree… Done
Reading state information… Done
pandoc is already the newest version (2.9.2.1-3ubuntu2).
pandoc set to manually installed.
The following additional packages will be installed:
  dvisvgm fonts-droid-fallback fonts-lato fonts-lmodern fonts-noto-mono
  fonts-texgyre fonts-urw-base35 libapache-pom-java libcommons-logging-java
  libcommons-parent-java libfontbox-java libfontenc1 libgs9 libgs9-common
  libidn12 libijs-0.35 libjbig2dec0 libkpathsea6 libpdfbox-java libptexenc1
  libruby3.0 libsynctex2 libteckit0 libtexlua53 libtexluajit2 libwoff1
  libzzip-0-13 lmodern poppler-data preview-latex-style rake ruby
  ruby-net-telnet ruby-rubygems ruby-webrick ruby-xmlrpc ruby3.0
  rubygems-integration t1utils teckit tex-common tex-gyre texlive-base
  texlive-binaries texlive-fonts-recommended texlive-latex-base
  texlive-latex-recommended texlive-pictures texlive-plain-generic tipa
  xfonts-encodings xfonts-utils
Suggested packages:
  fonts-noto fonts-freefont-otf | fonts-freefont-ttf libavalon-framework-java
  libcommons-logging-java-doc libexcalibur-logkit-java liblog4j1.2-java
  poppler-utils ghostscript fonts-japanese-mincho | fonts-ipafont-mincho
  fonts-japanese-gothic | fonts-ipafont-gothic fonts-arphic-ukai
  fonts-arphic-uming fonts-nanum ri ruby-dev bundler debhelper gv
  | postscript-viewer perl-tk xpdf | pdf-viewer xzdec
  texlive-fonts-recommended-doc texlive-latex-base-doc python3-pygments
  icc-profiles libfile-which-perl libspreadsheet-parseexcel-perl
  texlive-latex-extra-doc texlive-latex-recommended-doc texlive-luatex
  texlive-pstricks dot2tex prerex texlive-pictures-doc vprerex
  default-jre-headless tipa-doc
The following NEW packages will be installed:
  dvisvgm fonts-droid-fallback fonts-lato fonts-lmodern fonts-noto-mono
  fonts-texgyre fonts-urw-base35 libapache-pom-java libcommons-logging-java
  libcommons-parent-java libfontbox-java libfontenc1 libgs9 libgs9-common
  libidn12 libijs-0.35 libjbig2dec0 libkpathsea6 libpdfbox-java libptexenc1
  libruby3.0 libsynctex2 libteckit0 libtexlua53 libtexluajit2 libwoff1
  libzzip-0-13 lmodern poppler-data preview-latex-style rake ruby
  ruby-net-telnet ruby-rubygems ruby-webrick ruby-xmlrpc ruby3.0
  rubygems-integration t1utils teckit tex-common tex-gyre texlive texlive-base
  texlive-binaries texlive-fonts-recommended texlive-latex-base
  texlive-latex-extra texlive-latex-recommended texlive-pictures
  texlive-plain-generic texlive-xetex tipa xfonts-encodings xfonts-utils
0 upgraded, 55 newly installed, 0 to remove and 18 not upgraded.
Need to get 182 MB of archives.
```