Name:-jay Vinod Bhandarkar

Divison:-CS5

Batch:-CS51

Roll No:-CS5-22

PRN:-202401100062

```
from google.colab import files
uploaded = files.upload()
```

Choose Files No file chosen Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

```
Saving Airline-Sentiment-2-w-AA.xlsm to Airline-Sentiment-2-w-AA (1).xlsm
```

```
file_path = '/content/Airline-Sentiment-2-w-AA.xlsm'
```

```
import pandas as pd
import numpy as np
```

```
df = pd.read_excel(file_path)
```

```
df.head()
```

	index	_unit_id	_golden	_unit_state	_trusted_judgments	_last_judgment_at	airline_sentiment	airline_sentiment:confidence	n
0	0	681448150	False	finalized	3	2/25/15 5:24	neutral	1.0000	
1	1	681448153	False	finalized	3	2/25/15 1:53	positive	0.3486	
2	2	681448156	False	finalized	3	2/25/15 10:01	neutral	0.6837	
3	3	681448158	False	finalized	3	2/25/15 3:05	negative	1.0000	
4	4	681448159	False	finalized	3	2/25/15 5:50	negative	1.0000	

5 rows × 21 columns

```
# Problem 1: How many tweets are there in total?
len(df)
```

```
14640
```

```
# Problem 2: How many unique airlines are mentioned?
df['airline'].nunique()
```

```
6
```

```
# Problem 3: What are the different airlines mentioned?
df['airline'].unique()
```

```
array(['Virgin America', 'United', 'Southwest', 'Delta', 'US Airways',
       'American'], dtype=object)
```

```
# Problem 4: What is the distribution (count) of sentiments?
df['airline_sentiment'].value_counts()
```

airline_sentiment	count
negative	9178
neutral	3099
positive	2363

dtype: int64

```
 Problem 5: What percentage of tweets are positive?
df['airline_sentiment'].value_counts(normalize=True)['positive'] * 100
```

```
Object `positive` not found.
np.float64(16.140710382513664)
```

```
# Problem 6: Find the airline with the most negative tweets.
df[df['airline_sentiment'] == 'negative']['airline'].value_counts().idxmax()
```

```
'United'
```

```
# Problem 7: Find the airline with the most positive tweets.
df[df['airline_sentiment'] == 'positive']['airline'].value_counts().idxmax()
```

```
'Southwest'
```

```
# Problem 8: Average retweet count overall.
df['retweet_count'].mean()
```

```
np.float64(0.08265027322404371)
```

```
# Problem 9: Average retweet count for negative tweets.
df[df['airline_sentiment'] == 'negative']['retweet_count'].mean()
```

```
np.float64(0.09337546306384834)
```

```
# Problem 10: Most common reason for negative sentiment.
df['negativereason'].value_counts().idxmax()
```

```
'Customer Service Issue'
```

```
# Problem 11: Number of tweets without a negative reason.
df['negativereason'].isna().sum()
```

```
np.int64(5462)
```

```
# Problem 12: Number of tweets with a provided location.
df['tweet_location'].notna().sum()
```

```
np.int64(9907)
```

```
# Problem 13: Number of tweets with geographic coordinates.
df['tweet_coord'].notna().sum()
```

```
np.int64(1019)
```

```
# Problem 14: Find the time range of tweets (earliest to latest).
df['tweet_created'] = pd.to_datetime(df['tweet_created'])
(df['tweet_created'].min(), df['tweet_created'].max())
```

```
<ipython-input-19-30ea89028339>:2: UserWarning: Could not infer format, so each element will be parsed individually, falling back to
  df['tweet_created'] = pd.to_datetime(df['tweet_created'])
(Timestamp('2015-02-16 23:36:00'), Timestamp('2015-02-24 11:53:00'))
```

```
# Problem 15: Find the top 5 timezones users belong to.
df['user_timezone'].value_counts().head(5)
```

user_timezone	count
Eastern Time (US & Canada)	3744
Central Time (US & Canada)	1931
Pacific Time (US & Canada)	1208
Quito	738
Atlantic Time (Canada)	497

Double-click (or enter) to edit

dtype: int64

```
# Problem 16: Which timezone has the most negative tweets?
df[df['airline_sentiment'] == 'negative']['user_timezone'].value_counts().idxmax()
```

```
'Eastern Time (US & Canada)'
```

```
# Problem 17: Confidence level distribution for sentiment labeling.
df['airline_sentiment:confidence'].describe()
```

	airline_sentiment:confidence
count	14640.000000
mean	0.900169
std	0.162830
min	0.335000
25%	0.692300
50%	1.000000
75%	1.000000
max	1.000000

dtype: float64

```
# Problem 18: Tweets with 100% confidence level (sentiment classification).
df[df['airline_sentiment:confidence'] == 1.0].shape[0]
```

```
10445
```

```
# Problem 19: Number of users who tweeted multiple times (by 'name').
df['name'].value_counts()[df['name'].value_counts() > 1].count()
```

```
np.int64(3000)
```

```
# Problem 20: How many tweets have no user timezone?
df['user_timezone'].isna().sum()
```

```
np.int64(4820)
```