

```
In [63]: # I
import numpy as np
import necessary Libraries
import pandas as pd

# Load the data from the first sheet
df = pd.read_csv("C:/Users/USER/Downloads/jacques irakoze.csv", encoding='latin1')

df.head()
```

Out[63]:	Name	username	Comments
0	Baziryiki	fearlesslesgon1	What about the drivers who used to operate wit...
1	Nshimiyimiana jean de Dieu	NshimiyimianaJd4	This is a good idea, and we, the beneficiaries...
2	GasigwaOfficial	NyaweMotar	Ahaaaaaaa!! I feel you're joking for nothing...
3	Daniel HAKUZWEYEUZU	DanielHAKUZWEYEUZU	BUGESERA-MUSENYI SECTOR residents are happy wi...
4	sankara	karanowa_070336	Trips one after another will double. I see, an...

```
In [30]:
```

Columns in the DataFrame: ['Name' 'username' 'Comments']

```
In [24]: #pip install vaderSentiment
```

```
In [64]: from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer
import pandas as pd
import re

# Initialize analyzer
analyzer = SentimentIntensityAnalyzer()
```

```

In [32]: # Clean column names
df.columns = df.columns.str.strip()

# Filter valid comments
df_sample = df[df["Comments"].notnull()].copy()
df_sample["Comments"] = df_sample["Comments"].str.strip()
df_sample = df_sample[df_sample["Comments"] != ""]

# Remove non-ASCII characters
df_sample["Comments"] = df_sample["Comments"].apply(lambda x: re.sub(r'[^\x00-\x7F]+', '', x))

# Apply sentiment analysis
df_sample["Sentiment"] = df_sample["Comments"].apply(lambda x: analyzer.polarity_scores(x)["compound"])

# Optional: Label as Positive, Negative, or Neutral
def label_sentiment(score):
    if score >= 0.05:
        return "Positive"
    elif score <= -0.05:
        return "Negative"
    else:
        return "Neutral"

df_sample["Sentiment_Label"] = df_sample["Sentiment"].apply(label_sentiment)

# View result
print(df_sample[["Comments", "Sentiment", "Sentiment_Label"]].head())

```

	Comments	Sentiment	Sentiment_Label
0	What about the drivers who used to operate wit...	0.1670	Positive
1	This is a good idea, and we, the beneficiaries...	0.8689	Positive
2	Ahaaaaaaa!! I feel youre joking for nothing.	0.3578	Positive
3	BUGESERA-MUSENYI SECTOR residents are happy wi...	0.8885	Positive
4	Trips one after another will double, I see, an...	-0.5773	Negative

In [27]:

```

In [65]: import re

def clean_text(text):
    # Convert to Lowercase
    text = text.lower()
    # Remove emojis (Unicode ranges for emojis)
    text = re.sub(r'[\U0001F600-\U0001F64F\U0001F300-\U0001F5FF\U0001F680-\U0001F6FF\U0001F700-\U0001F77F\U0001F780-\U0001F7FF\U0001F800-\U0001F8FF\U0001F900-\U0001F9FF\U0001FA00-\U0001FA6F\U0001FA70-\U0001FAFF\U00002700-\U000027BF]', '', text)
    # Remove special characters and punctuation
    text = re.sub(r'["\W\S]', '', text)
    # Remove extra whitespace
    text = ' '.join(text.split())
    # Handle missing or empty text
    if not text:
        text = 'na'
    return text

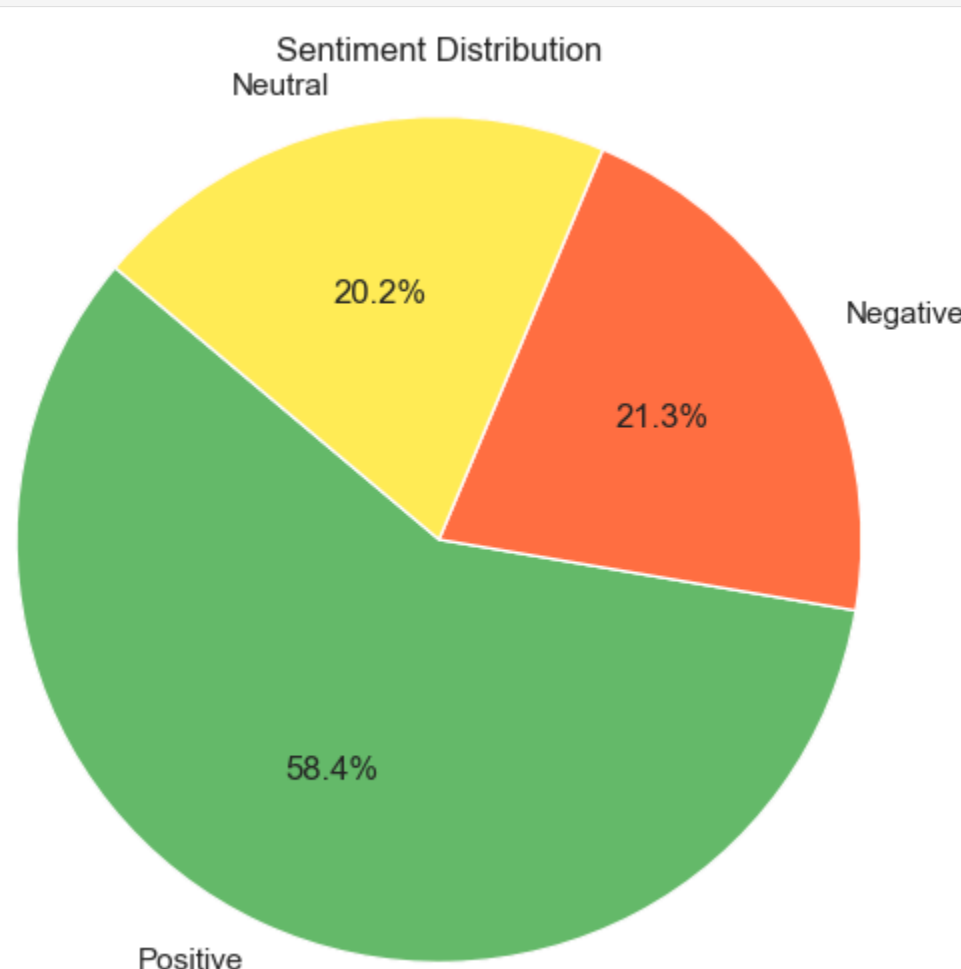
# Apply cleaning to the English Translation column
df_clean['Cleaned Translated Comments (English)'] = df_clean['Translated Comments (English)'].apply(clean_text)

```

```
In [28]: import matplotlib.pyplot as plt

# Count sentiment labels
sentiment_counts = df_sample["Sentiment_Label"].value_counts()

# Pie chart
plt.figure(figsize=(6, 6))
plt.pie(
    sentiment_counts,
    labels=sentiment_counts.index,
    autopct="%1.1f%%",
    startangle=140,
    colors=["#66b666", "#ff7043", "#ffee58"] # Positive, Negative, Neutral
)
plt.title("Sentiment Distribution")
plt.axis("equal")
plt.show()
```

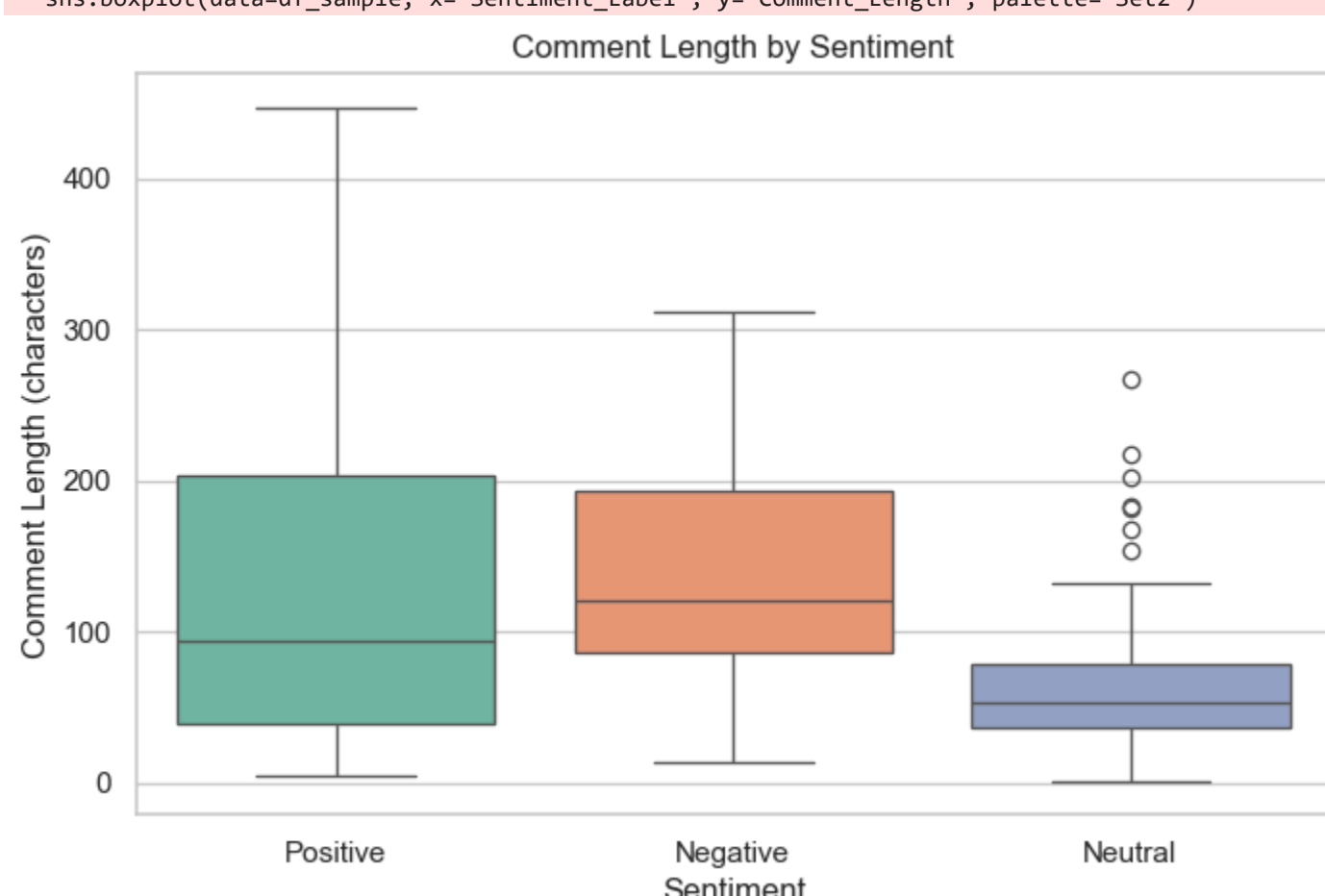


```
In [66]: # Add comment length
df_sample['Comment_Length'] = df_sample['Comments'].apply(len)

# Boxplot: Comment Length vs Sentiment
import seaborn as sns

plt.figure(figsize=(8, 5))
sns.boxplot(data=df_sample, x='Sentiment_Label', y='Comment_Length', palette='Set2')
plt.title("Comment Length by Sentiment")
plt.xlabel("Sentiment")
plt.ylabel("Comment Length (characters)")
plt.show()
```

```
C:\Users\USER\AppData\Local\Temp\ipykernel_17780\2794679441.py:8: FutureWarning:
Passing 'palette' without assigning 'hue' is deprecated and will be removed in v0.14.0. Assign the 'x' variable to 'hue' and set 'legend=False' for the same effect.
sns.bovplot(data=df_sample, x='Sentiment Label', y='Comment Length', palette=Set2)
```



```
In [ ]: # Install in Jupyter notebook
        #!pip install wordcloud
```

```
In [53]: #fr
```

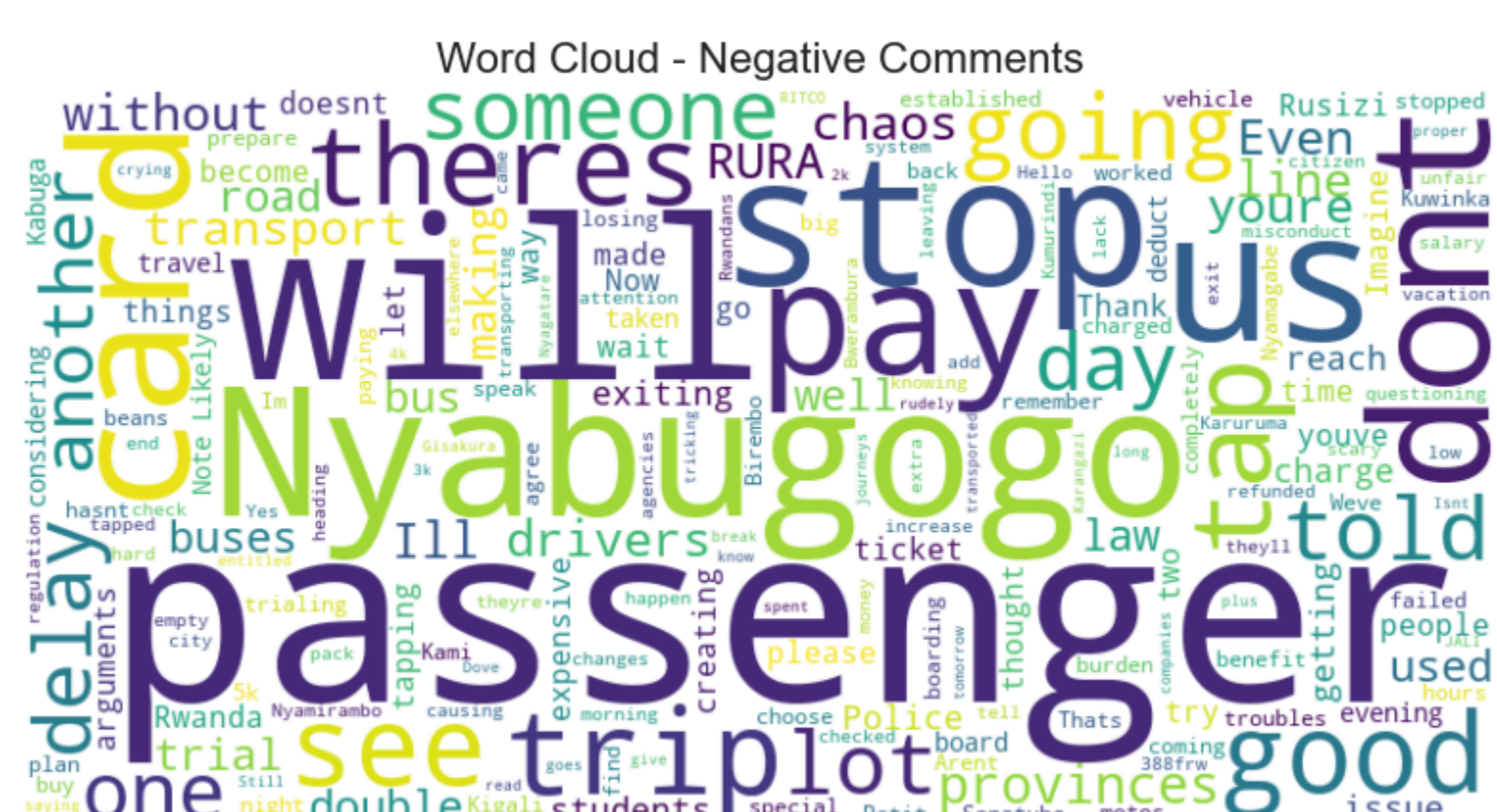
```
In [37]: #pip install wordcloud
```

```
In [67]: from wordcloud import WordCloud

def generate_wordcloud(data, title):
    text = ' '.join(data)
    wordcloud = WordCloud(width=800, height=400, background_color='white').generate(text)

    plt.figure(figsize=(10, 5))
    plt.imshow(wordcloud, interpolation='bilinear')
    plt.axis('off')
    plt.title(title, fontsize=16)
    plt.show()

# Generate word clouds
for label in ['Positive', 'Negative', 'Neutral']:
    comments = df_sample[df_sample['Sentiment_Label'] == label]['Comments']
    generate_wordcloud(comments, f'Word Cloud - {label} Comments')
```



Word Cloud - Neutral Comments

