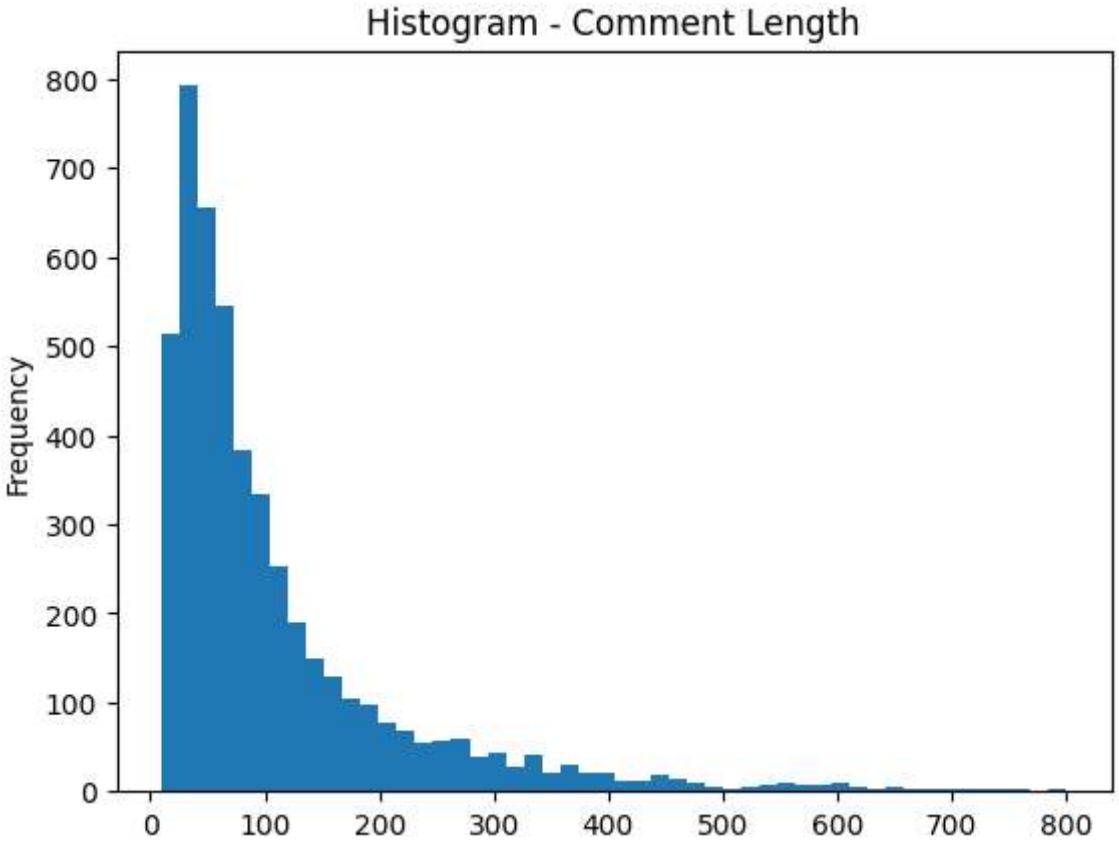
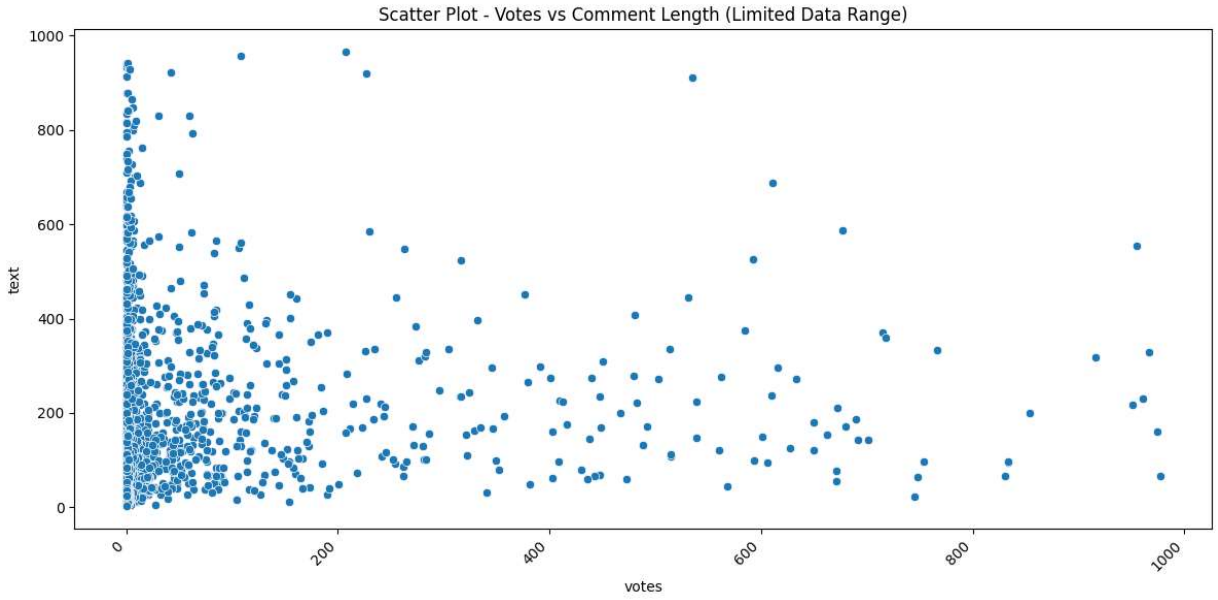
In [ ]:
```
import pandas as pd
from wordcloud import WordCloud
import matplotlib.pyplot as plt

# Load the cleaned data CSV
df = pd.read_csv('cleaned_youtube_comments.csv')

# Drop rows where 'text' is NaN
df = df.dropna(subset=['text'])

# Convert 'text' column to strings
df['text'] = df['text'].astype(str)

# Word Cloud
wordcloud = WordCloud(width=800, height=400, max_words=200, background_color='white
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.title('Word Cloud - Comment Text')
plt.show()
```

In [ ]:
```
import pandas as pd
import matplotlib.pyplot as plt

# Load the cleaned data CSV
df = pd.read_csv('cleaned_youtube_comments.csv')

# Convert 'text' column to strings, handling NaN values
df['text'] = df['text'].astype(str)

# Histogram with specified x-axis range
df['text'].apply(lambda x: len(str(x)) if not pd.isna(x) else 0).plot.hist(bins=50,
plt.show()
```



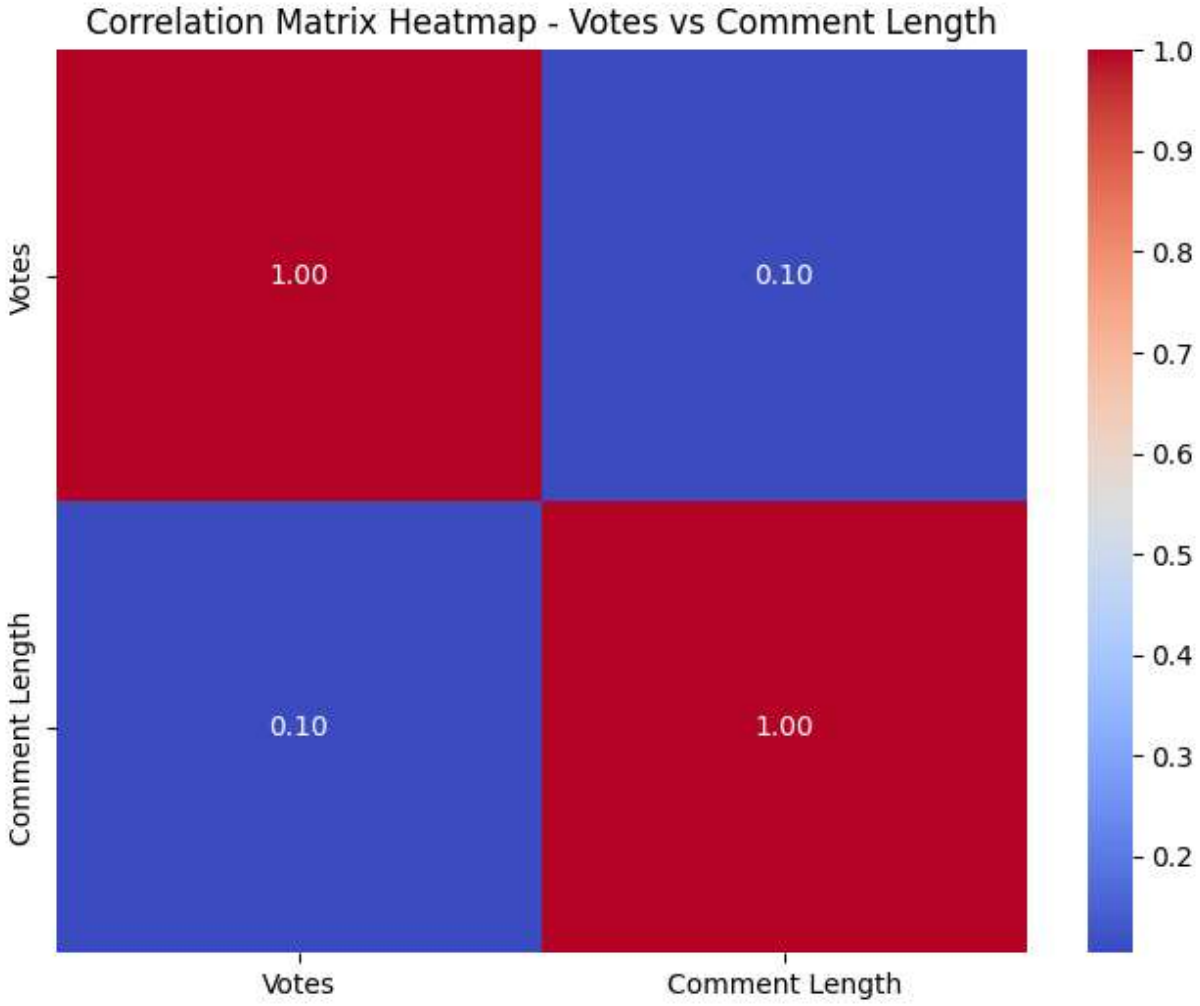
In [ ]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# Load the cleaned data CSV with 'votes' column
df = pd.read_csv('cleaned_youtube_comments.csv')

# Convert 'votes' column to numeric, handling errors and converting to NaN for non-
df['votes'] = pd.to_numeric(df['votes'], errors='coerce')

# Filter data within a specific range
filtered_df = df[(df['votes'].between(0, 5000, inclusive=True)) & (df['text'].apply

# Scatter Plot with limited data range
plt.figure(figsize=(12, 6))

sns.scatterplot(x='votes', y=filtered_df['text'].apply(lambda x: len(str(x)) if not

plt.title('Scatter Plot - Votes vs Comment Length (Limited Data Range)')
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()
```

```
C:\Users\etash\AppData\Local\Temp\ipykernel_5376\2834398278.py:12: FutureWarning: Bo
olean inputs to the `inclusive` argument are deprecated in favour of `both` or `neit
her`.
  filtered_df = df[(df['votes'].between(0, 5000, inclusive=True)) & (df['text'].appl
y(lambda x: len(str(x)) if not pd.isna(x) else 0) <= 1000)]
```



In [ ]:

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# Load the cleaned data CSV with 'votes' column
df = pd.read_csv('cleaned_youtube_comments.csv')

# Convert 'votes' column to numeric, handling errors and converting to NaN for non-
df['votes'] = pd.to_numeric(df['votes'], errors='coerce')

# Calculate correlation matrix
corr_matrix = df[['votes', 'text']].apply(lambda x: len(str(x['text'])) if not pd.i

# Heatmap
plt.figure(figsize=(8, 6))
sns.heatmap([[1, corr_matrix], [corr_matrix, 1]], annot=True, cmap='coolwarm', fmt=
plt.title('Correlation Matrix Heatmap - Votes vs Comment Length')
plt.show()
```



In [ ]:

```python
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# Load the cleaned data CSV with 'votes' column
df = pd.read_csv('cleaned_youtube_comments.csv')

# Convert 'votes' column to numeric, handling errors and converting to NaN for non-
df['votes'] = pd.to_numeric(df['votes'], errors='coerce')

# Filter data within a specific range
filtered_df = df[(df['votes'].between(0, 5000, inclusive=True)) & (df['text'].apply

# Bubble Chart
plt.figure(figsize=(12, 8))

sns.scatterplot(x='votes', y=filtered_df['text'].apply(lambda x: len(str(x)) if not

plt.title('Bubble Chart - Votes vs Comment Length (Limited Data Range)')
plt.xlabel('Votes')
plt.ylabel('Comment Length')
plt.tight_layout()
plt.show()
```

```
C:\Users\etash\AppData\Local\Temp\ipykernel_5376\2269404341.py:12: FutureWarning: Bo
olean inputs to the `inclusive` argument are deprecated in favour of `both` or `neit
her`.
  filtered_df = df[(df['votes'].between(0, 5000, inclusive=True)) & (df['text'].appl
y(lambda x: len(str(x)) if not pd.isna(x) else 0) <= 1000)]
```