```
import matplotlib.pyplot as plt
import seaborn as sns
from wordcloud import WordCloud
import nltk
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
import string


nltk.download('punkt')
nltk.download('stopwords')
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
True
```

```
# Load the dataset
import pandas as pd
df = pd.read_csv('/content/Restaurant_Reviews.tsv', sep='\t')
df.head()
```

	Review	Liked
0	Wow... Loved this place.	1
1	Crust is not good.	0
2	Not tasty and the texture was just nasty.	0
3	Stopped by during the late May bank holiday of...	1
4	The selection on the menu was great and so wer	1

Next steps: Generate code with df | View recommended plots | New interactive sheet

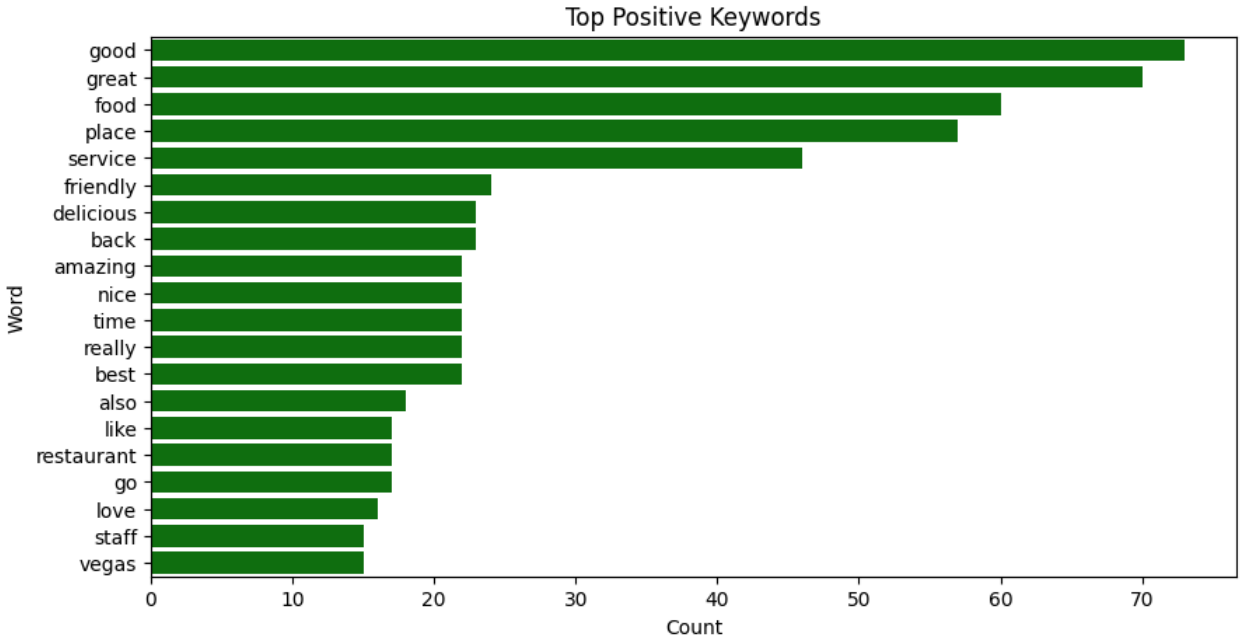
```
nltk.download('punkt')
nltk.download('stopwords')
nltk.download('punkt_tab')

df = pd.read_csv('/content/Restaurant_Reviews.tsv', sep='\t')

print("Shape:", df.shape)
df.head()

stop_words = set(stopwords.words('english'))
def clean_text(text):
    text = text.lower()
    tokens = word_tokenize(text)
```

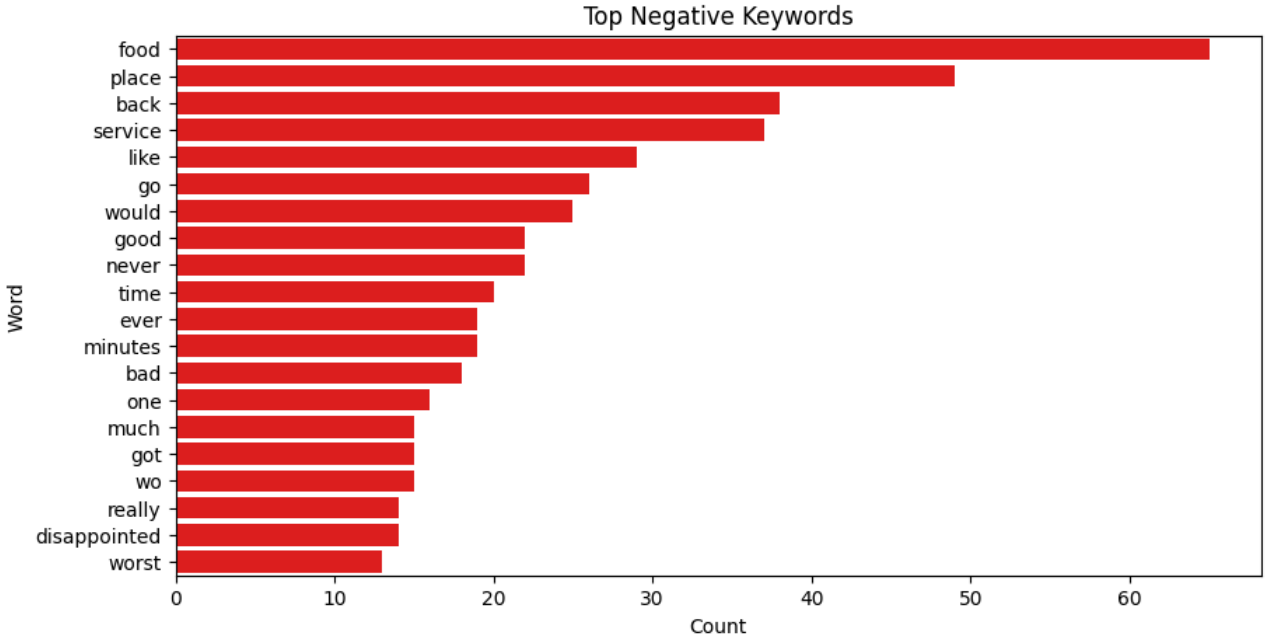
```
    cleaned = [word for word in tokens if word.isalpha() and word not in stop_words]
    return " ".join(cleaned)


df['cleaned_review'] = df['Review'].apply(clean_text)
df[['Review', 'cleaned_review']].head()
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package punkt_tab to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt_tab.zip.
Shape: (1000, 2)
```

	Review	cleaned_review
0	Wow... Loved this place.	wow loved place
1	Crust is not good.	crust good
2	Not tasty and the texture was just nasty.	tasty texture nasty
3	Stopped by during the late May bank holiday of...	stopped late may bank holiday rick steve recom...
4	The selection on the menu was great and so wer	selection menu great prices

```
from collections import Counter


positive_reviews = df[df['Liked'] == 1]['cleaned_review']
negative_reviews = df[df['Liked'] == 0]['cleaned_review']

positive_words = " ".join(positive_reviews).split()
negative_words = " ".join(negative_reviews).split()

positive_freq = Counter(positive_words).most_common(20)
negative_freq = Counter(negative_words).most_common(20)

pos_df = pd.DataFrame(positive_freq, columns=['Word', 'Count'])
neg_df = pd.DataFrame(negative_freq, columns=['Word', 'Count'])



plt.figure(figsize=(10,5))
sns.barplot(data=pos_df, x='Count', y='Word', color='green')
plt.title('Top Positive Keywords')
plt.show()


plt.figure(figsize=(10,5))
sns.barplot(data=neg_df, x='Count', y='Word', color='red')
plt.title('Top Negative Keywords')
plt.show()
```





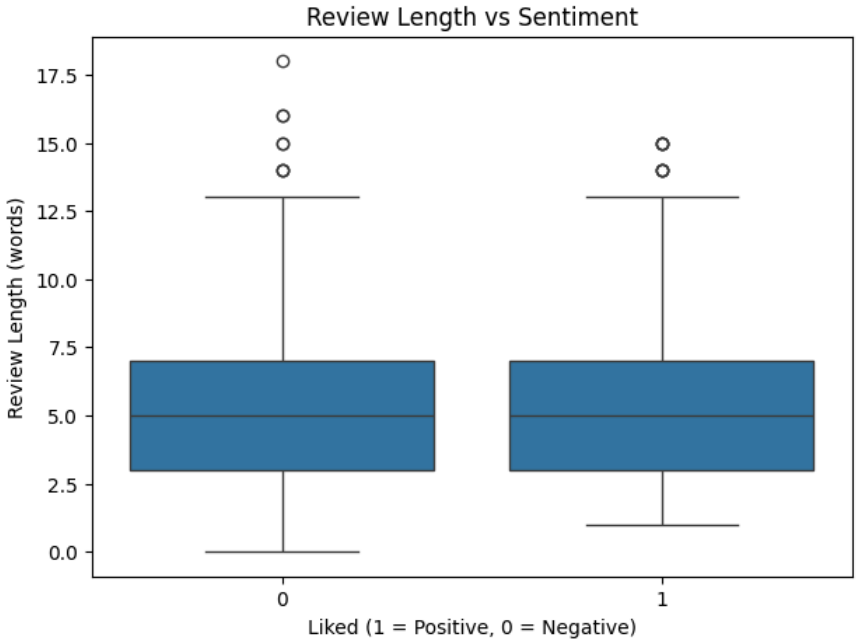

```
df['review_length'] = df['cleaned_review'].apply(lambda x: len(x.split()))
```

```
avg_len = df['review_length'].mean()
print(f"Average Review Length: {avg_len:.2f} words")
```

```
Average Review Length: 5.45 words
```

```
plt.figure(figsize=(7,5))
sns.boxplot(x='Liked', y='review_length', data=df)
plt.title('Review Length vs Sentiment')
plt.xlabel('Liked (1 = Positive, 0 = Negative)')
plt.ylabel('Review Length (words)')
plt.show()
```

```
correlation = df['Liked'].corr(df['review_length'])
print(f"Correlation between review length and sentiment (Liked): {correlation:.2f}")
```




```
Correlation between review length and sentiment (Liked): -0.04
```

```
from wordcloud import WordCloud
import matplotlib.pyplot as plt

custom_stopwords = set([
    "place", "one", "go", "got", "get", "would", "really", "back",
    "restaurant", "food", "service", "also", "time"
```

```
])


def clean_text(text):
    tokens = word_tokenize(text.lower())
    cleaned = [word for word in tokens if word.isalpha() and word not in stop_words and word not in custom_stopwords]
    return " ".join(cleaned)

df['cleaned_review'] = df['Review'].apply(clean_text)
positive_reviews = df[df['Liked'] == 1]['cleaned_review']
negative_reviews = df[df['Liked'] == 0]['cleaned_review']

positive_text = " ".join(positive_reviews)
negative_text = " ".join(negative_reviews)

wordcloud_pos = WordCloud(
    width=900, height=450,
    background_color='white',
    colormap='Greens',
    contour_color='black',
    contour_width=1,
    max_words=100,
    prefer_horizontal=0.95
).generate(positive_text)

plt.figure(figsize=(10, 5))
plt.imshow(wordcloud_pos, interpolation='bilinear')
plt.axis('off')
plt.title('🌟 Most Common Words in Positive Reviews', fontsize=16, fontweight='bold')
plt.show()

wordcloud_neg = WordCloud(
    width=900, height=450,
    background_color='white',
    colormap='Reds',
    contour_color='black',
    contour_width=1,
    max_words=100,
    prefer_horizontal=0.95
).generate(negative_text)

plt.figure(figsize=(10, 5))
plt.imshow(wordcloud_neg, interpolation='bilinear')
plt.axis('off')
plt.title('⚠️ Most Common Words in Negative Reviews', fontsize=16, fontweight='bold')
plt.show()
```


Most Common Words in Positive Reviews
good
great
friendly
delicious
nice
amazing
best
staff
loved
chicken
steak
vegas
price
first
awesome
fantastic
perfect
excellent
love
always
menu
atmosphere
fresh
night
taste
salad
well
Most Common Words in Negative Reviews
never
good
bad
minutes
much
worst
wo
terrible
think
better
slow
bland
overpriced
cold
wait
quality
eat
came
going
table
disappointed
waited
horrible
probably
experience
chicken
avoid
lunch
ordered
us
way
rude
eating
felt
definitely
mediocre
disappointing
little
sushi
coming
want
vegas
salad
worth
know
next
took
poor
hot
star
meat
still
people
waste
another
nothing
customer
asked
server
pretty
take
either
hard
impressed
money
taste
waiter
many
meal
sauce
management
waiting
give
say
night
thing
bit
times
burger