Data Collection

```
import pandas as pd

# Load the dataset
data = pd.read_csv('/content/British_Airway_Review.csv')

Data Preview:
                                                reviews              date
\
0  ✅ Trip Verified |  I had the most fantastic BA...  1st August 2023

1  ✅ Trip Verified |  Couldn’t book in online. Ar...   31st July 2023

2  ✅ Trip Verified | London Heathrow to Mumbai in...   31st July 2023

3  ✅ Trip Verified | Keflavík, Iceland to London ...   31st July 2023

4  ✅ Trip Verified | Terrible Experience with Bri...   29th July 2023


          country       seat_type recommended  stars
route  \
0       Hong Kong  Business Class         yes      5  Heathrow to Las
Vegas
1  United Kingdom   Economy Class          no      3       Rome to
Heathrow
2         Iceland  Business Class         yes      3      Gatwick to
Venice
3         Iceland  Business Class         yes      5       London to
Luanda
4          Canada   Economy Class          no      5     Denver to
Heathrow

  type_of_traveller
0    Family Leisure
1      Solo Leisure
2      Solo Leisure
3    Couple Leisure
4    Family Leisure

Dataset Shape: (2500, 8)

Dataset Info:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2500 entries, 0 to 2499
Data columns (total 8 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   reviews            2500 non-null   object
```

```
 1   date                2500 non-null   object
 2   country             2500 non-null   object
 3   seat_type           2500 non-null   object
 4   recommended         2500 non-null   object
 5   stars               2500 non-null   int64
 6   route               2500 non-null   object
 7   type_of_traveller   2500 non-null   object
dtypes: int64(1), object(7)
memory usage: 156.4+ KB
None

# Display the first few rows of the dataset
print("Data Preview:")
print(data.head())

Data Preview:
                                             reviews             date
\
0  ✅ Trip Verified |  I had the most fantastic BA...  1st August 2023

1  ✅ Trip Verified |  Couldn’t book in online. Ar...   31st July 2023

2  ✅ Trip Verified | London Heathrow to Mumbai in...   31st July 2023

3  ✅ Trip Verified | Keflavík, Iceland to London ...   31st July 2023

4  ✅ Trip Verified | Terrible Experience with Bri...   29th July 2023


          country       seat_type recommended  stars
route  \
0       Hong Kong  Business Class         yes      5  Heathrow to Las
Vegas
1  United Kingdom   Economy Class          no      3       Rome to
Heathrow
2         Iceland  Business Class         yes      3      Gatwick to
Venice
3         Iceland  Business Class         yes      5       London to
Luanda
4          Canada   Economy Class          no      5     Denver to
Heathrow

  type_of_traveller
0    Family Leisure
1      Solo Leisure
2      Solo Leisure
3    Couple Leisure
4    Family Leisure

# Check dataset size
print("\nDataset Shape:", data.shape)
```

```
Dataset Shape: (2500, 8)
```

```
# Check dataset structure
print("\nDataset Info:")
print(data.info())
```

```
Dataset Info:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2500 entries, 0 to 2499
Data columns (total 8 columns):
 #   Column             Non-Null Count  Dtype 
---  ------             --------------  ----- 
 0   reviews            2500 non-null   object
 1   date               2500 non-null   object
 2   country            2500 non-null   object
 3   seat_type          2500 non-null   object
 4   recommended        2500 non-null   object
 5   stars              2500 non-null   int64 
 6   route              2500 non-null   object
 7   type_of_traveller  2500 non-null   object
dtypes: int64(1), object(7)
memory usage: 156.4+ KB
None
```

Data Inspection

```
# View statistical summary for numerical columns
print("\nDescriptive Statistics:")
print(data.describe())
```

```
Descriptive Statistics:
             stars
count  2500.000000
mean      4.272000
std       2.135378
min       1.000000
25%       3.000000
50%       3.000000
75%       5.000000
max       9.000000
```

```
# View unique values for categorical columns
categorical_columns = data.select_dtypes(include=['object']).columns
for col in categorical_columns:
    print(f"\nUnique Values in '{col}':")
    print(data[col].unique())
```

Unique Values in 'reviews':
["✅ Trip Verified | I had the most fantastic BA Flight today. The cabin crew in my seat zone 6D were of the best I have experienced. Although the 777 had the old style rear/front seating - it was comfortable and felt open and spacious. BA have done great things with the menu - it was nice not to have to eat breakfast food on the pre-arrival meal into HKG at 1730PM local time having left London at 2200 the night before. The dinner offering was equally ample and delicious. The inflight entertainment had a great collection of TV and current films and the inflight map was first rate. Previously not a huge BA fan - but after this 12 hour flight I could only recommend it without reserve. I flew the new BA suite in J class on a recent Delhi to London flight and enjoyed that product (although I felt it slightly more cramped) but I think BA have stepped up with really comfortable bedding and food and very friendly crew who engaged throughout the flight in the best way. Couldn't recommend BA more on this flight today"
 '✅ Trip Verified | Couldn’t book in online. Arrived at check in to find we had been bumped off due to overselling. No BA staff available. Very helpful Gatwick staff got us a bus to LHR and a flight to Toulouse. Had knock in effect on our car booking and sharing as the rest of family had been able yo board original flight. Airlines should be legally stopped from selling seats twice.'
 '✅ Trip Verified | London Heathrow to Mumbai in a Boeing 787-8 in Business Class. The lounge near Terminal 5, Gate B36 at Heathrow was outstanding in its service and offerings. It provides us just the right frame to relax in before boarding as the departure was delayed by almost 2 hours. The 787-8 on our flight featured the older Club World seating. Not the best in class but comfortable enough. I hear that the new Club Suites configuration is far superior. British Airways onboard service was outstanding in every respect. All in all, a very comfortable flight. One minor irritant: for some reason this aircraft was not fitted with WiFi. We got into Mumbai at 8 am, a civilized time to arrive.'
 ...
 'I travelled from London to Sydney via Singapore in British Airways first class. I found the service outstanding, food was good lots of choices and the wine list was also good. Seat was comfortable, but the thing I would fault is that the armrest does stick in to you when in the sleep position. Would fly them again.'
 "British Airways have just moved Cape Town flights to Terminal 3 at Heathrow and it's not a good move. The Concorde lounge at T3 is looking so shabby that it's a disgrace. Thankfully, on board in the new first class was better. The crew was young but did a good job. Seats 4E/F are good for a couple, but there's some lack of privacy. Dinner was excellent, which is down to steam ovens British Airways have finally fitted in first class galley. The combination of PJs, duvet and a 6 ft 6 flat bed meant that sleep actually occurred. The letdown was the breakfast which was mediocre as usual, although it was

```
good tea and coffee. But, as others have said, British Airways first class just isn't as good as Singapore and Emirates."
 "Bengaluru to Heathrow. My first long haul flight with British Airways and I am not at all impressed. The service levels were disgusting. No ear plugs, No eye shades even after requests.  Drink service only a showpiece. Asked for a Virgin Mary and was give iced tomato juice with lemon. When asked for other ingredients like Worcestershire sauce, tabasco, Salt and Pepper, I was told that this was the airline recipe. My request for some salt and pepper to spice my drink was refused saying that `we do not stock them in Economy'. When I requested for a talk with the Head of Staff on the aircraft, he was sorry and said the airline was trying to cut corners and the crew pleas were falling on deaf years.  Only vegetarian dinner option provided since they had apparently run out. No menus etc. IFE selection with dated screens and headsets.  Was a Low cost carrier experience that takes a passenger from Point A to Point B without any feel good moments."]

Unique Values in 'date':
['1st August 2023' '31st July 2023' '29th July 2023' ...
 '12th November 2015' '11th November 2015' '10th November 2015']

Unique Values in 'country':
['Hong Kong' 'United Kingdom' 'Iceland' 'Canada' 'Qatar' 'Spain'
'Germany'
 'United States' 'South Africa' 'Greece' 'Italy' 'Senegal'
 'United Arab Emirates' 'Romania' 'Australia' 'Cyprus' 'Chile'
'Sweden'
 'Ireland' 'Netherlands' 'Austria' 'France' 'India' 'Belgium'
 'New Zealand' 'Czech Republic' 'Malaysia' 'Singapore' 'Ghana'
 'Switzerland' 'Bermuda' 'Botswana' 'Brazil' 'Panama' 'Nigeria'
 'Russian Federation' 'Philippines' 'Bulgaria' 'Poland' 'Thailand'
 'Argentina' 'Mexico' 'Denmark' 'Saint Kitts and Nevis' 'Vietnam'
'Norway'
 'Jordan' 'Japan' 'Taiwan' 'China' 'Slovakia' 'Kuwait' 'Israel'
 'South Korea' 'Saudi Arabia' 'Hungary' 'Portugal' 'Cayman Islands'
 'Costa Rica' 'Egypt' 'Laos' 'Turkey' 'Indonesia' 'Bahrain'
 'Dominican Republic' 'Luxembourg' 'Finland']

Unique Values in 'seat_type':
['Business Class' 'Economy Class' 'Premium Economy' 'First Class']

Unique Values in 'recommended':
['yes' 'no']

Unique Values in 'route':
['Heathrow to Las Vegas' 'Rome to Heathrow' 'Gatwick to Venice'
 'London to Luanda' 'Denver to Heathrow' 'BKK to LHR' 'London to
Tampa'
 'London to Sydney' 'LHR to CPT' 'BLR to LHR']
```

```
Unique Values in 'type_of_traveller':
['Family Leisure' 'Solo Leisure' 'Couple Leisure' 'Business']

# Check for missing values
print("\nMissing Values:")
print(data.isnull().sum())


Missing Values:
reviews              0
date                 0
country              0
seat_type            0
recommended          0
stars                0
route                0
type_of_traveller    0
dtype: int64
```

Data Cleaning

```
# Handle missing values (example: filling missing dates with a
placeholder)
data['date'] = pd.to_datetime(data['date'], errors='coerce')
data['date'].fillna(pd.Timestamp('1970-01-01'), inplace=True)
# Remove rows with outlier ratings (assuming ratings are between 1 and
5)
data = data[(data['stars'] >= 1) & (data['stars'] <= 5)]

# Convert 'recommended' to binary values
data['recommended_encoded'] = data['recommended'].apply(lambda x: 1 if
x.lower() == 'yes' else 0)

# Display cleaned data info
print("\nCleaned Data Info:")
print(data.info())


Cleaned Data Info:
<class 'pandas.core.frame.DataFrame'>
Index: 2046 entries, 0 to 2499
Data columns (total 9 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   reviews              2046 non-null   object
 1   date                 2046 non-null   datetime64[ns]
 2   country              2046 non-null   object
 3   seat_type            2046 non-null   object
 4   recommended          2046 non-null   object
```

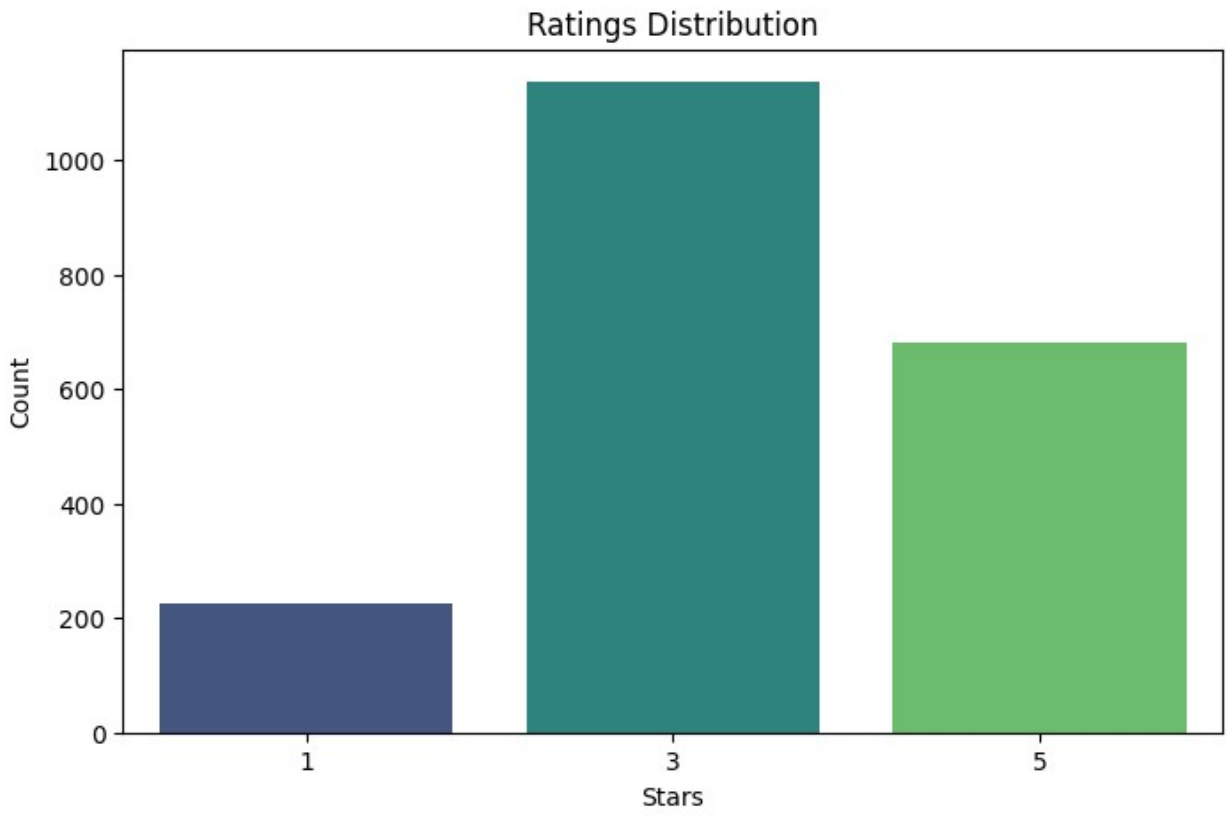
```
 5   stars                2046 non-null   int64
 6   route                2046 non-null   object
 7   type_of_traveller    2046 non-null   object
 8   recommended_encoded  2046 non-null   int64
dtypes: datetime64[ns](1), int64(2), object(6)
memory usage: 159.8+ KB
None

<ipython-input-12-9fbe86f7453c>:3: FutureWarning: A value is trying to
be set on a copy of a DataFrame or Series through chained assignment
using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never
work because the intermediate object on which we are setting values
always behaves as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try
using 'df.method({col: value}, inplace=True)' or df[col] =
df[col].method(value) instead, to perform the operation inplace on the
original object.


  data['date'].fillna(pd.Timestamp('1970-01-01'), inplace=True)
<ipython-input-12-9fbe86f7453c>:8: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#
returning-a-view-versus-a-copy
  data['recommended_encoded'] = data['recommended'].apply(lambda x: 1
if x.lower() == 'yes' else 0)
```

Univariate Analysis

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

# Plot ratings distribution
plt.figure(figsize=(8, 5))
sns.countplot(x='stars', data=data, palette='viridis')
plt.title("Ratings Distribution")
plt.xlabel("Stars")
plt.ylabel("Count")
plt.show()
```

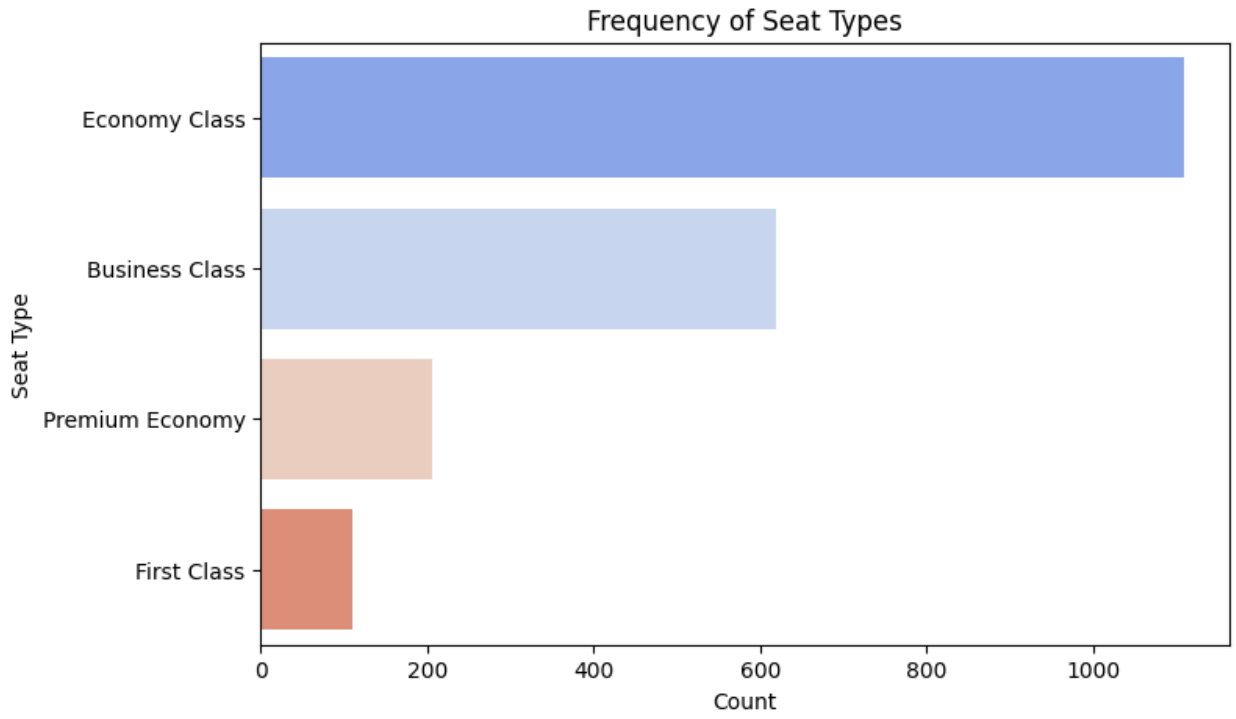
```
<ipython-input-13-9f0d0f56556b>:6: FutureWarning:

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.countplot(x='stars', data=data, palette='viridis')
```



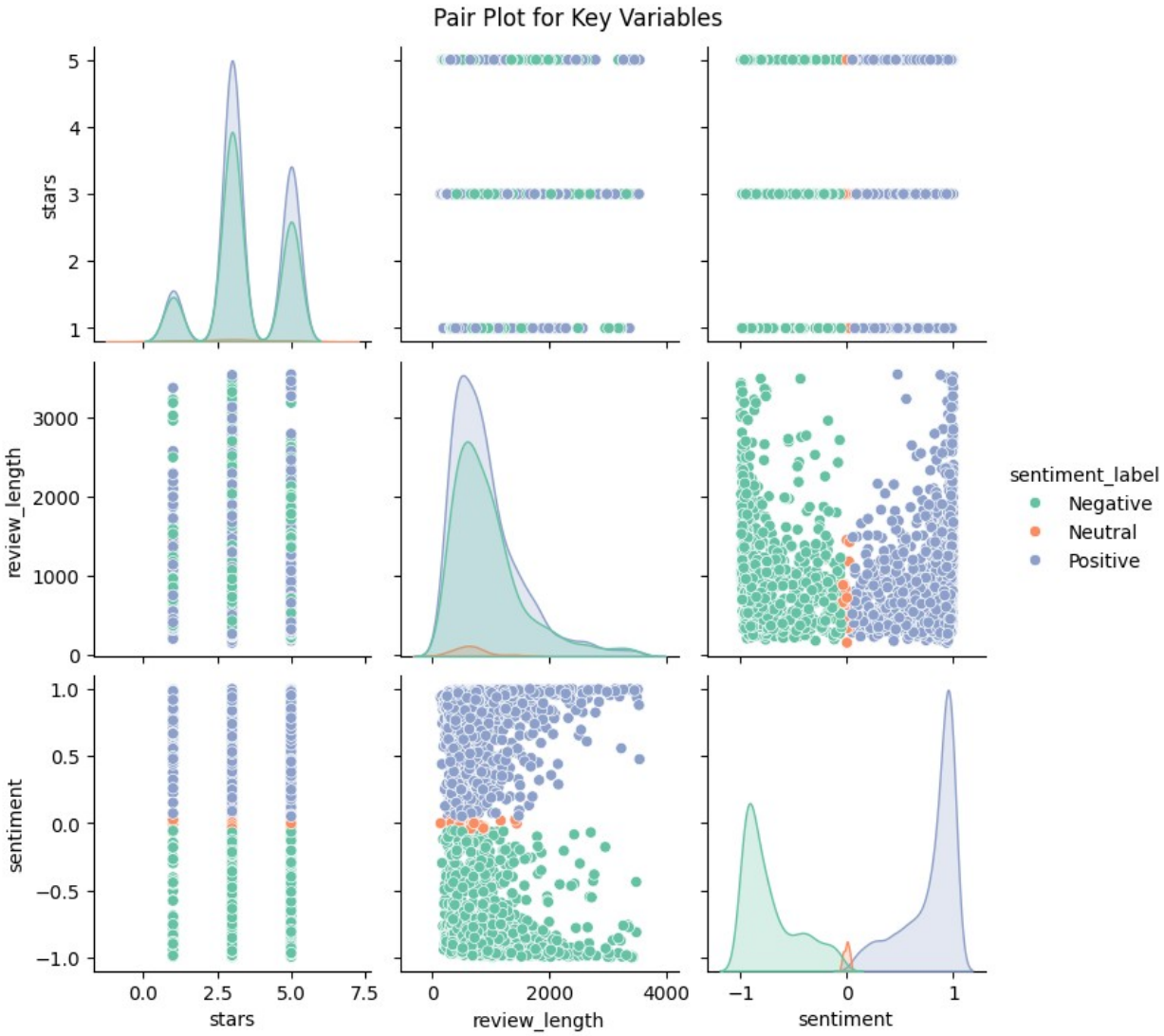

```
# Analyze the frequency of seat types
plt.figure(figsize=(8, 5))
sns.countplot(y='seat_type', data=data, palette='coolwarm',
order=data['seat_type'].value_counts().index)
plt.title("Frequency of Seat Types")
plt.ylabel("Seat Type")
plt.xlabel("Count")
plt.show()
```

```
<ipython-input-15-031142468d68>:3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be
removed in v0.14.0. Assign the `y` variable to `hue` and set
`legend=False` for the same effect.

  sns.countplot(y='seat_type', data=data, palette='coolwarm',
order=data['seat_type'].value_counts().index)
```



Multivariate Analysis

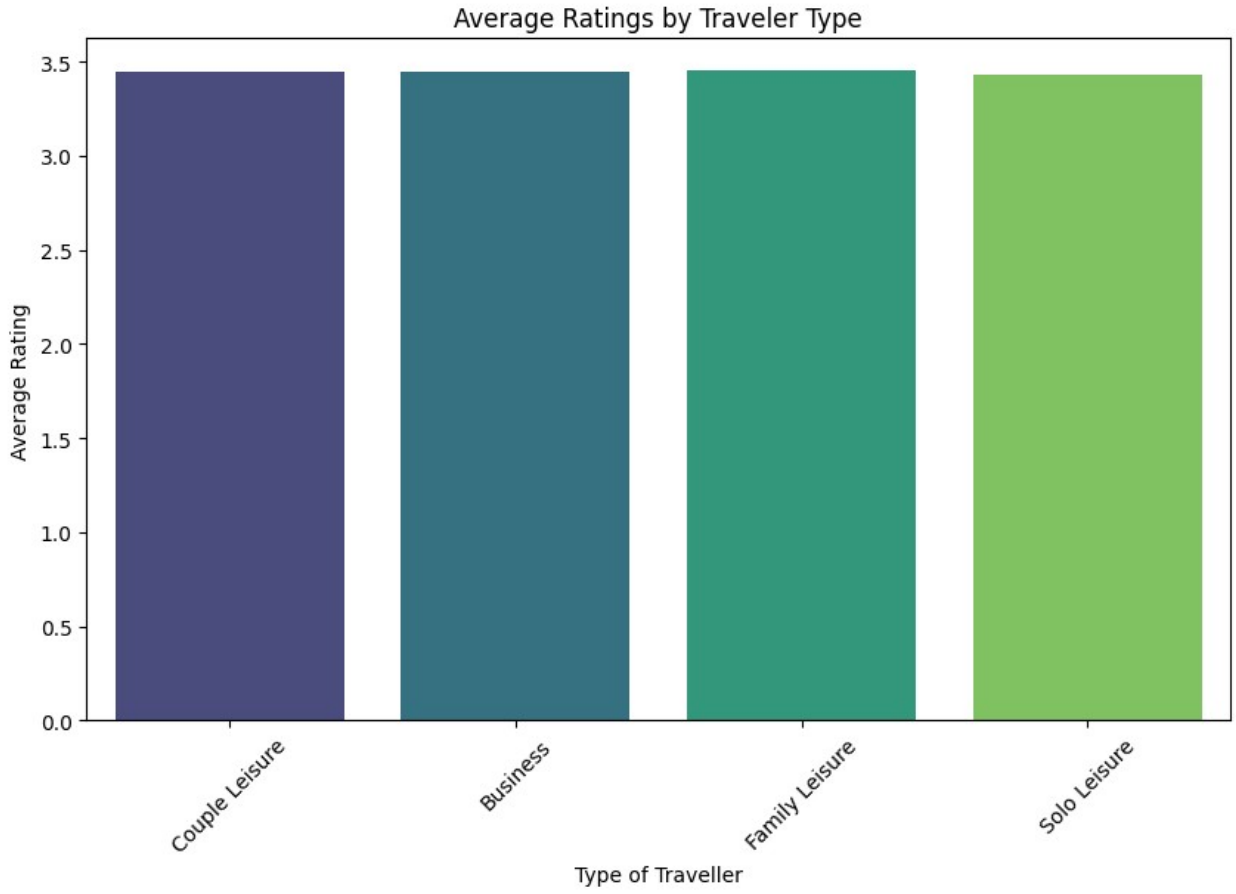
```
import matplotlib.pyplot as plt
import seaborn as sns

# 1. Pair Plot (Relationships Among Multiple Variables)
sns.pairplot(data, vars=['stars', 'review_length', 'sentiment'],
hue='sentiment_label', palette='Set2', diag_kind='kde')
plt.suptitle("Pair Plot for Key Variables", y=1.02)
plt.show()
```




```
# 2. Average Ratings by Traveler Type
plt.figure(figsize=(10, 6))
sns.barplot(x='type_of_traveller', y='stars', data=data, ci=None,
palette='viridis',
            order=data['type_of_traveller'].value_counts().index)
plt.title("Average Ratings by Traveler Type")
plt.xlabel("Type of Traveller")
plt.ylabel("Average Rating")
plt.xticks(rotation=45)
plt.show()
```

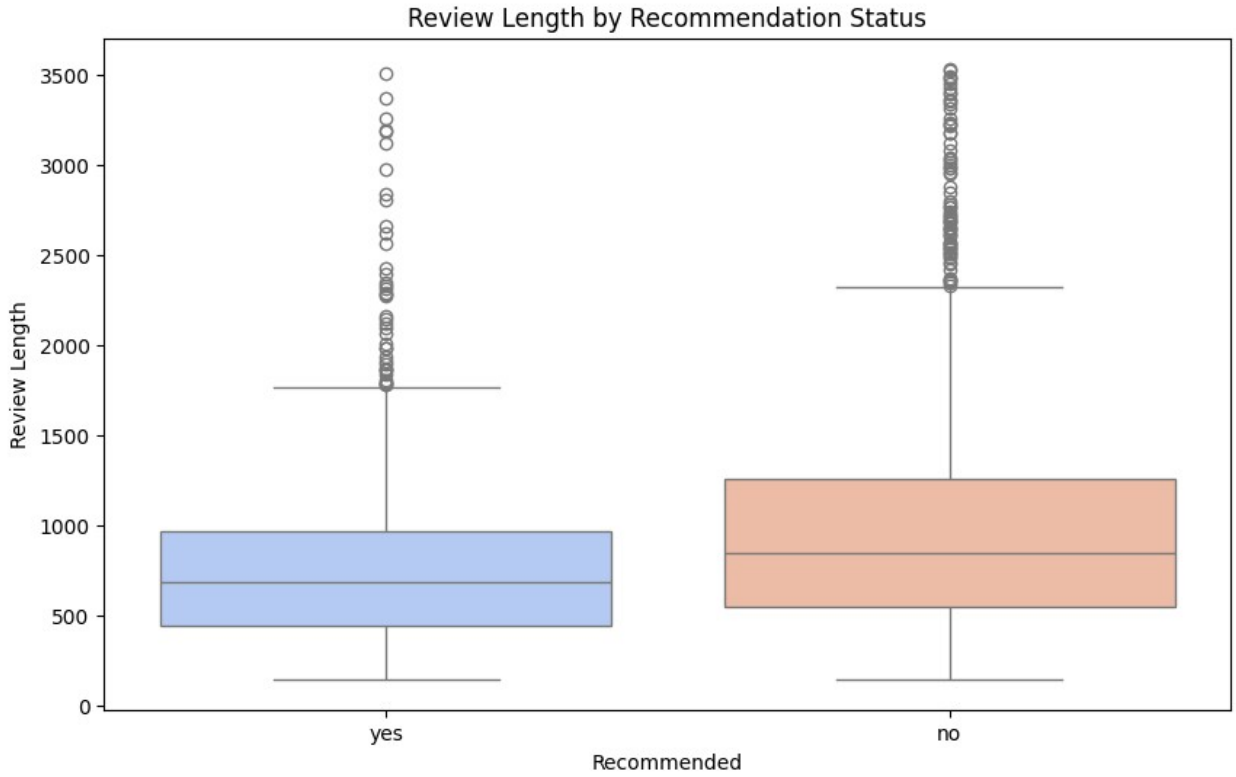
```
<ipython-input-29-f6cd15bb095c>:3: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same
effect.
```

```
  sns.barplot(x='type_of_traveller', y='stars', data=data, ci=None,
palette='viridis',
<ipython-input-29-f6cd15bb095c>:3: 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.barplot(x='type_of_traveller', y='stars', data=data, ci=None,
palette='viridis',
```

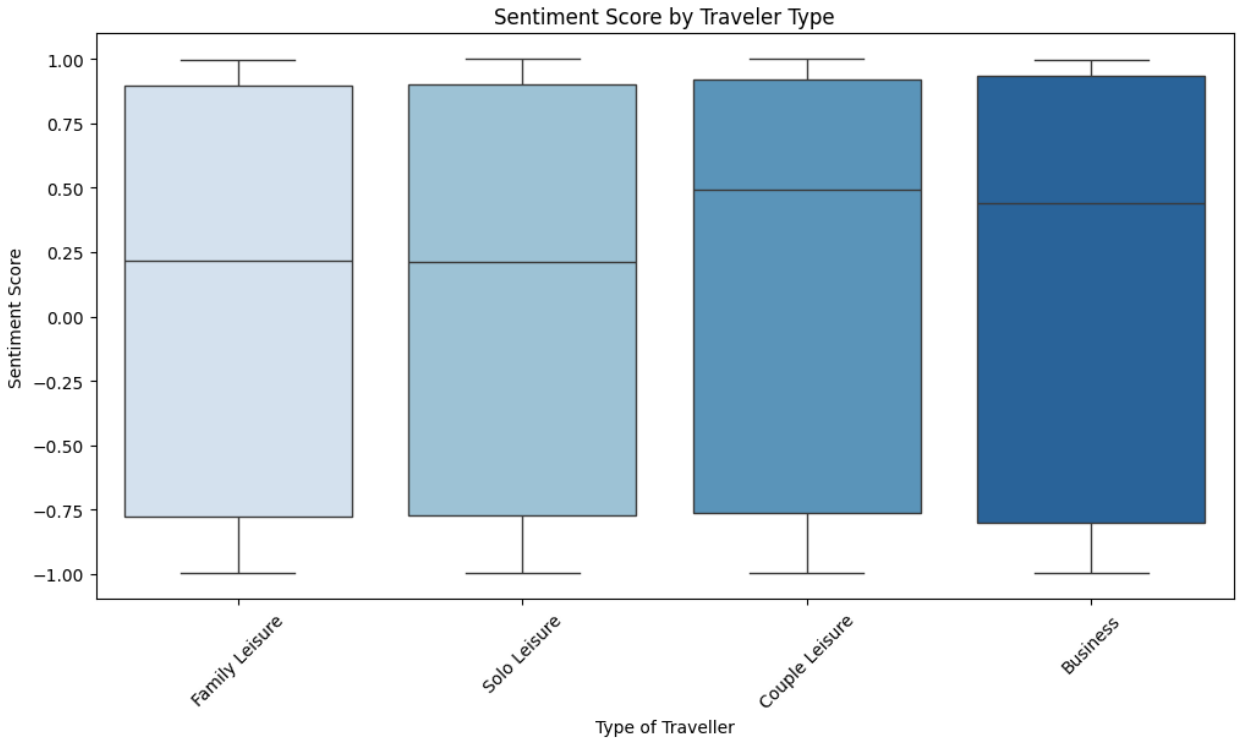



```
# 3. Average Review Length by Recommendation
plt.figure(figsize=(10, 6))
sns.boxplot(x='recommended', y='review_length', data=data,
palette='coolwarm')
plt.title("Review Length by Recommendation Status")
plt.xlabel("Recommended")
plt.ylabel("Review Length")
plt.show()

<ipython-input-30-616dfd5c5d63>:3: 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.boxplot(x='recommended', y='review_length', data=data,
palette='coolwarm')
```



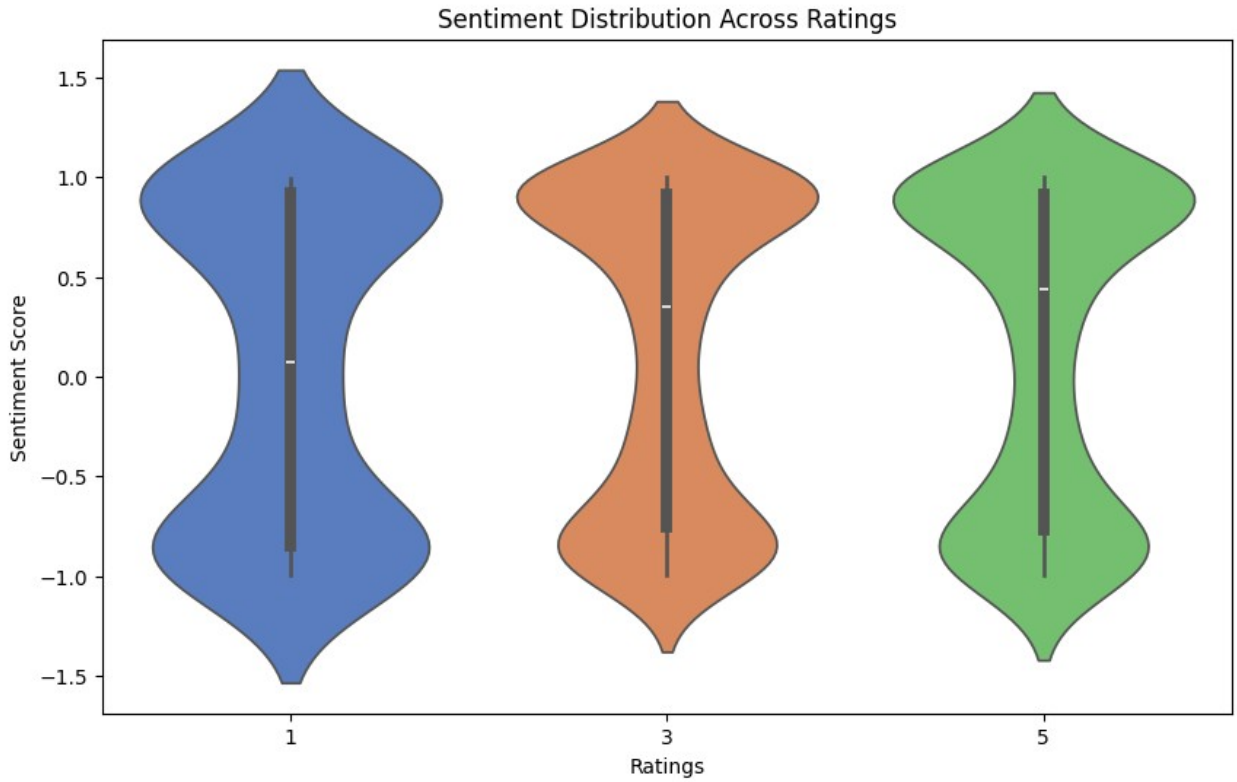

```
# 4. Sentiment by Traveler Type
plt.figure(figsize=(12, 6))
sns.boxplot(x='type_of_traveller', y='sentiment', data=data,
palette='Blues')
plt.title("Sentiment Score by Traveler Type")
plt.xlabel("Type of Traveller")
plt.ylabel("Sentiment Score")
plt.xticks(rotation=45)
plt.show()

<ipython-input-31-9a4ff24687b8>:3: 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.boxplot(x='type_of_traveller', y='sentiment', data=data,
palette='Blues')
```




```
# 5. Ratings by Route
top_routes = data['route'].value_counts().head(10).index
filtered_data = data[data['route'].isin(top_routes)]

plt.figure(figsize=(12, 6))
sns.boxplot(x='route', y='stars', data=filtered_data, palette='Set3')
plt.title("Ratings by Top 10 Routes")
plt.xlabel("Route")
plt.ylabel("Ratings")
plt.xticks(rotation=45)
plt.show()

# 6. Sentiment Distribution Across Ratings
plt.figure(figsize=(10, 6))
sns.violinplot(x='stars', y='sentiment', data=data, palette='muted',
scale='width')
plt.title("Sentiment Distribution Across Ratings")
plt.xlabel("Ratings")
plt.ylabel("Sentiment Score")
plt.show()
```

```
<ipython-input-33-7bdcf6a35e39>:3: 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.
```

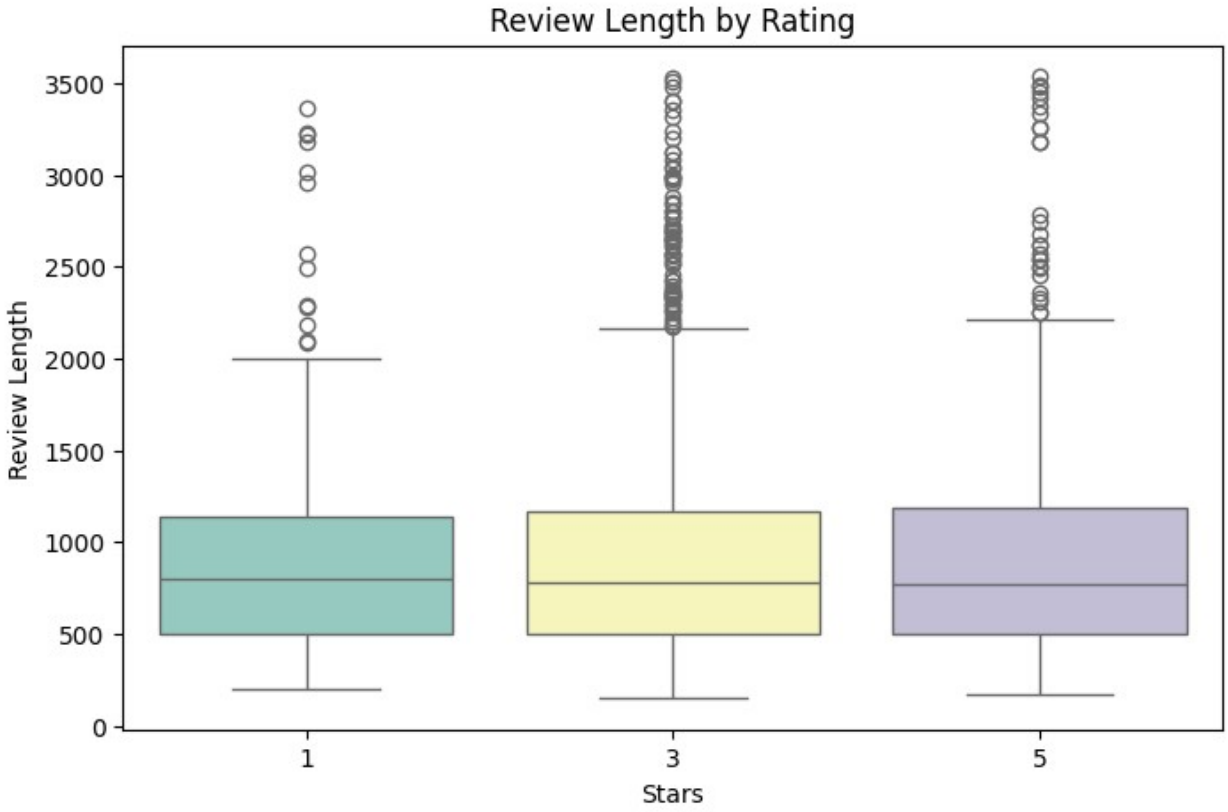
```
  sns.violinplot(x='stars', y='sentiment', data=data, palette='muted',
scale='width')
<ipython-input-33-7bdcf6a35e39>:3: FutureWarning:

The `scale` parameter has been renamed and will be removed in v0.15.0.
Pass `density_norm='width'` for the same effect.
  sns.violinplot(x='stars', y='sentiment', data=data, palette='muted',
scale='width')
```



Feature Analysis

```
# Create a new feature: Review Length
data['review_length'] = data['reviews'].apply(lambda x: len(str(x)))

# Analyze review length by rating
plt.figure(figsize=(8, 5))
sns.boxplot(x='stars', y='review_length', data=data, palette='Set3')
plt.title("Review Length by Rating")
plt.xlabel("Stars")
plt.ylabel("Review Length")
plt.show()

<ipython-input-19-fb46e7ab9c51>:6: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be
```

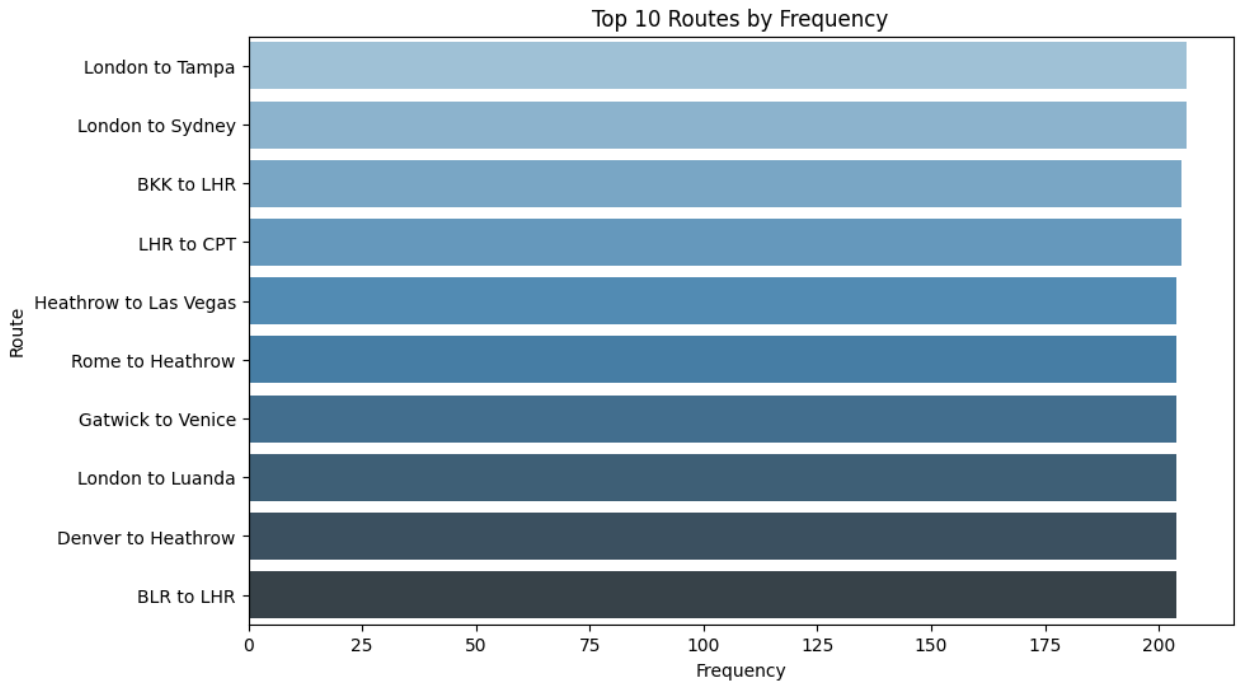
```
removed in v0.14.0. Assign the `x` variable to `hue` and set
`legend=False` for the same effect.

  sns.boxplot(x='stars', y='review_length', data=data, palette='Set3')
```



Data Visualization

```
# Top 10 most common routes
top_routes = data['route'].value_counts().head(10)
plt.figure(figsize=(10, 6))
sns.barplot(x=top_routes.values, y=top_routes.index,
palette='Blues_d')
plt.title("Top 10 Routes by Frequency")
plt.xlabel("Frequency")
plt.ylabel("Route")
plt.show()

<ipython-input-20-0a5019ace7d8>:4: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be
removed in v0.14.0. Assign the `y` variable to `hue` and set
`legend=False` for the same effect.
```

```
  sns.barplot(x=top_routes.values, y=top_routes.index,
palette='Blues_d')
```



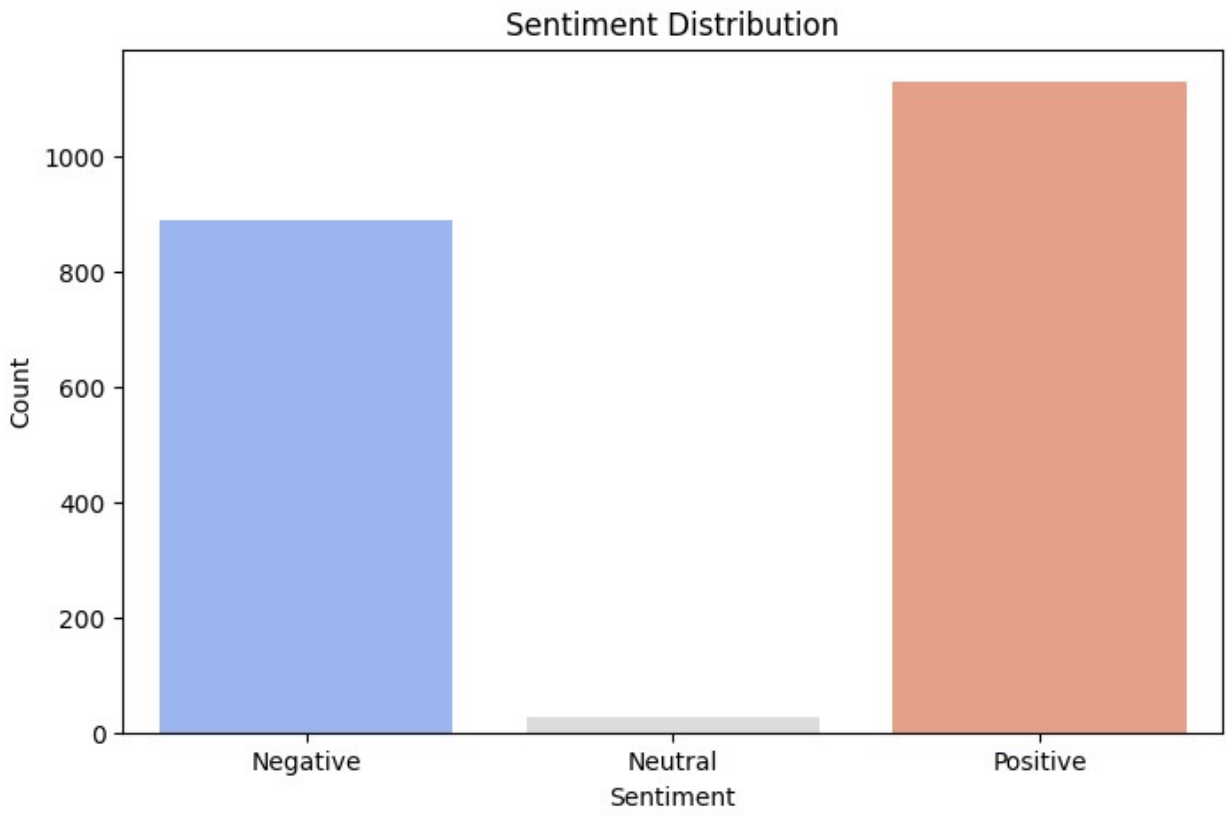

```
# Sentiment Distribution
from nltk.sentiment import SentimentIntensityAnalyzer
import nltk
nltk.download('vader_lexicon')
sia = SentimentIntensityAnalyzer()
data['sentiment'] = data['reviews'].apply(lambda x:
sia.polarity_scores(str(x))['compound'])
data['sentiment_label'] = pd.cut(data['sentiment'], bins=[-1, -0.05,
0.05, 1], labels=['Negative', 'Neutral', 'Positive'])
```

```
[nltk_data] Downloading package vader_lexicon to /root/nltk_data...
```

```
plt.figure(figsize=(8, 5))
sns.countplot(x='sentiment_label', data=data, palette='coolwarm')
plt.title("Sentiment Distribution")
plt.xlabel("Sentiment")
plt.ylabel("Count")
plt.show()
```

```
<ipython-input-22-d8b64c1a7e74>:2: 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.countplot(x='sentiment_label', data=data, palette='coolwarm')
```




```
# Calculate summary insights
summary = {
    "Total Reviews": len(data),
    "Average Rating": data['stars'].mean(),
    "Positive Sentiment Percentage": (data['sentiment_label'] ==
'Positive').mean() * 100,
    "Most Common Route": data['route'].mode()[0],
    "Most Frequent Seat Type": data['seat_type'].mode()[0]
}

# Display insights
print("\nSummary Insights:")
for key, value in summary.items():
    print(f"{key}: {value}")
```

```
Summary Insights:
Total Reviews: 2046
Average Rating: 3.444770283479961
Positive Sentiment Percentage: 55.2297165200391
```

```
Most Common Route: London to Sydney
Most Frequent Seat Type: Economy Class

# Export final cleaned dataset
data.to_csv('Cleaned_British_Airways_Data.csv', index=False)
print("\nCleaned data saved as 'Cleaned_British_Airways_Data.csv'")


Cleaned data saved as 'Cleaned_British_Airways_Data.csv'
```