In [1]:
```
#import libraries

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

In [81]:
```
#Handling Missing Values

# Replace missing values in the 'DelayReason' column with 'Unknown'
flight_delays['DelayReason'] = flight_delays['DelayReason'].fillna('Unkno

# Verify if missing values are replaced
print(flight_delays['DelayReason'].isnull().sum())  # Should print 0
print(flight_delays['DelayReason'].value_counts())  # Check the counts, '

# Save the updated DataFrame back to the same CSV
flight_delays.to_csv('flight_delays_updated.csv', index=False)
```

```
0
DelayReason
Unknown                468873
Air Traffic Control    426488
Maintenance            426168
Weather                426098
Name: count, dtype: int64
```

In [83]:
```
#Loading datasets

flight_delays = pd.read_csv('flight_delays_updated.csv')
passenger_satisfaction = pd.read_csv('airline_passenger_satisfaction.csv'
```

In [145…
```
# Display first few rows and info for both datasets
print("Flight Delays Dataset:")
print(flight_delays.head())
print(flight_delays.info())

print("\nPassenger Satisfaction Dataset:")
print(passenger_satisfaction.head())
print(passenger_satisfaction.info())
```

```
Flight Delays Dataset:
   FlightID    Airline  FlightNumber Origin Destination  ScheduledDepartur
e  \
0         1     United          4558    ORD         MIA 2024-09-01 08:11:0
0
1         2      Delta          8021    LAX         MIA 2024-09-01 10:25:0
0
2         3  Southwest          7520    DFW         SFO 2024-09-01 16:53:0
0
3         4      Delta          2046    ORD         BOS 2024-09-01 14:44:0
0
4         5      Delta          6049    LAX         SEA 2024-09-01 01:51:0
0

    ActualDeparture  ScheduledArrival     ActualArrival  DelayMinutes  ...
\
0  2024-09-01 08:30  2024-09-01 12:11  2024-09-01 12:19             8  ...
1  2024-09-01 10:41  2024-09-01 13:25  2024-09-01 13:27             2  ...
2  2024-09-01 17:05  2024-09-01 17:53  2024-09-01 18:07            14  ...
3  2024-09-01 15:04  2024-09-01 18:44  2024-09-01 18:34           -10  ...
4  2024-09-01 02:08  2024-09-01 05:51  2024-09-01 06:15            24  ...

  Diverted  AircraftType  TailNumber Distance DelayCategory  ScheduledHour
\
0    False    Boeing 737      N71066     1031         Short              8
1     True   Airbus A320      N22657     1006         Short             10
2     True    Boeing 737      N95611     2980         Short             16
3    False    Boeing 777      N90029     1408         Short             14
4     True    Boeing 737      N27417     2298        Medium              1

   TimeOfDay       Route Month  MonthName
0    Morning  ORD -> MIA     9  September
1    Morning  LAX -> MIA     9  September
2  Afternoon  DFW -> SFO     9  September
3  Afternoon  ORD -> BOS     9  September
4      Night  LAX -> SEA     9  September

[5 rows x 22 columns]
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1747627 entries, 0 to 1747626
Data columns (total 22 columns):
 #   Column              Dtype
---  ------              -----
 0   FlightID            int64
 1   Airline             object
 2   FlightNumber        int64
 3   Origin              object
 4   Destination         object
 5   ScheduledDeparture  datetime64[ns]
 6   ActualDeparture     object
 7   ScheduledArrival    object
 8   ActualArrival       object
 9   DelayMinutes        int64
 10  DelayReason         object
 11  Cancelled           bool
 12  Diverted            bool
 13  AircraftType        object
 14  TailNumber          object
 15  Distance            int64
 16  DelayCategory       object
```

```
 17  ScheduledHour       int32
 18  TimeOfDay           object
 19  Route               object
 20  Month               int32
 21  MonthName           object
dtypes: bool(2), datetime64[ns](1), int32(2), int64(4), object(13)
memory usage: 256.7+ MB
None

Passenger Satisfaction Dataset:
   ID  Gender  Age Customer Type Type of Travel     Class  Flight Distance
\
0   1    Male   48    First-time       Business  Business              821
1   2  Female   35     Returning       Business  Business              821
2   3    Male   41     Returning       Business  Business              853
3   4    Male   50     Returning       Business  Business             1905
4   5  Female   49     Returning       Business  Business             3470

   Departure Delay  Arrival Delay  Departure and Arrival Time Convenience
\
0                2            5.0                                       3
1               26           39.0                                       2
2                0            0.0                                       4
3                0            0.0                                       2
4                0            1.0                                       3

   ...  Food and Drink  In-flight Service  In-flight Wifi Service  \
0  ...               5                  5                       3
1  ...               3                  5                       2
2  ...               5                  3                       4
3  ...               4                  5                       2
4  ...               4                  3                       3

   In-flight Entertainment  Baggage Handling             Satisfaction  \
0                        5                 5  Neutral or Dissatisfied
1                        5                 5                Satisfied
2                        3                 3                Satisfied
3                        5                 5                Satisfied
4                        3                 3                Satisfied

   SatisfactionCategory     AgeGroup  DistanceCategory  Month
0                   Low        Adult       Medium Haul      1
1                  High  Young Adult       Medium Haul      1
2                  High        Adult       Medium Haul      1
3                  High        Adult         Long Haul      1
4                  High        Adult         Long Haul      1

[5 rows x 28 columns]
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 129880 entries, 0 to 129879
Data columns (total 28 columns):
 #   Column                                  Non-Null Count   Dtype
---  ------                                  --------------   -----
 0   ID                                      129880 non-null  int64
 1   Gender                                  129880 non-null  object
 2   Age                                     129880 non-null  int64
 3   Customer Type                           129880 non-null  object
 4   Type of Travel                          129880 non-null  object
 5   Class                                   129880 non-null  object
 6   Flight Distance                         129880 non-null  int64
```

```
 7   Departure Delay                          129880 non-null  int64
 8   Arrival Delay                            129487 non-null  float64
 9   Departure and Arrival Time Convenience   129880 non-null  int64
 10  Ease of Online Booking                   129880 non-null  int64
 11  Check-in Service                         129880 non-null  int64
 12  Online Boarding                          129880 non-null  int64
 13  Gate Location                            129880 non-null  int64
 14  On-board Service                         129880 non-null  int64
 15  Seat Comfort                             129880 non-null  int64
 16  Leg Room Service                         129880 non-null  int64
 17  Cleanliness                              129880 non-null  int64
 18  Food and Drink                           129880 non-null  int64
 19  In-flight Service                        129880 non-null  int64
 20  In-flight Wifi Service                   129880 non-null  int64
 21  In-flight Entertainment                  129880 non-null  int64
 22  Baggage Handling                         129880 non-null  int64
 23  Satisfaction                             129880 non-null  object
 24  SatisfactionCategory                     129880 non-null  object
 25  AgeGroup                                 129880 non-null  object
 26  DistanceCategory                         129880 non-null  object
 27  Month                                    129880 non-null  int32
dtypes: float64(1), int32(1), int64(18), object(8)
memory usage: 27.3+ MB
None
```

In [147…

```
#Categorize Delay Durations (Flight Delays)

# Categorize delays into short, medium, and long
def categorize_delay(minutes):
    if minutes <= 15:
        return 'Short'
    elif minutes <= 60:
        return 'Medium'
    else:
        return 'Long'

flight_delays['DelayCategory'] = flight_delays['DelayMinutes'].apply(cate

# Verify the new column
print(flight_delays['DelayCategory'].value_counts())
```

```
DelayCategory
Short     1108310
Medium     639317
Name: count, dtype: int64
```

In [149…

```
#Categorize Satisfaction Levels (Passenger Satisfaction)

# Simplify satisfaction levels
def categorize_satisfaction(satisfaction):
    if satisfaction == 'Satisfied':
        return 'High'
    elif satisfaction == 'Neutral or Dissatisfied':
        return 'Low'
    else:
        return 'Unknown'

passenger_satisfaction['SatisfactionCategory'] = passenger_satisfaction['
```

```
# Verify the new column
print(passenger_satisfaction['SatisfactionCategory'].value_counts())
```

```
SatisfactionCategory
Low     73452
High    56428
Name: count, dtype: int64
```

In [151…

```
#Categorize Age Groups (Passenger Satisfaction)

# Categorize Age into groups
def age_group(age):
    if age < 18:
        return 'Child'
    elif age <= 35:
        return 'Young Adult'
    elif age <= 60:
        return 'Adult'
    else:
        return 'Senior'

passenger_satisfaction['AgeGroup'] = passenger_satisfaction['Age'].apply(

# Verify the new column
print(passenger_satisfaction['AgeGroup'].value_counts())
```

```
AgeGroup
Adult          68081
Young Adult    41898
Senior         10054
Child           9847
Name: count, dtype: int64
```

In [153…

```
# Check for long delays (> 60 minutes)
print(flight_delays[flight_delays['DelayMinutes'] > 60].shape[0])
```

```
0
```

In [155…

```
print(flight_delays['DelayMinutes'].describe())
```

```
count    1.747627e+06
mean     9.999179e+00
std      1.183112e+01
min     -1.000000e+01
25%      0.000000e+00
50%      1.000000e+01
75%      2.000000e+01
max      3.000000e+01
Name: DelayMinutes, dtype: float64
```

Feature Engineering for Both Datasets

In [157…

```
#Flight Delays

# Categorize Delay Durations
def categorize_delay(minutes):
    if minutes <= 15:
        return 'Short'
    else:
        return 'Medium'
```

```
flight_delays['DelayCategory'] = flight_delays['DelayMinutes'].apply(cate

# Check results
print(flight_delays['DelayCategory'].value_counts())
```

```
DelayCategory
Short     1108310
Medium     639317
Name: count, dtype: int64
```

In [159…

```
#Passenger Satisfaction

#Categorize Satisfaction Levels
def categorize_satisfaction(satisfaction):
    if satisfaction == 'Satisfied':
        return 'High'
    elif satisfaction == 'Neutral or Dissatisfied':
        return 'Low'
    else:
        return 'Unknown'

passenger_satisfaction['SatisfactionCategory'] = passenger_satisfaction['

#Categorize Age Groups
def age_group(age):
    if age < 18:
        return 'Child'
    elif age <= 35:
        return 'Young Adult'
    elif age <= 60:
        return 'Adult'
    else:
        return 'Senior'

passenger_satisfaction['AgeGroup'] = passenger_satisfaction['Age'].apply(

#Check results
print(passenger_satisfaction['SatisfactionCategory'].value_counts())
print(passenger_satisfaction['AgeGroup'].value_counts())
```

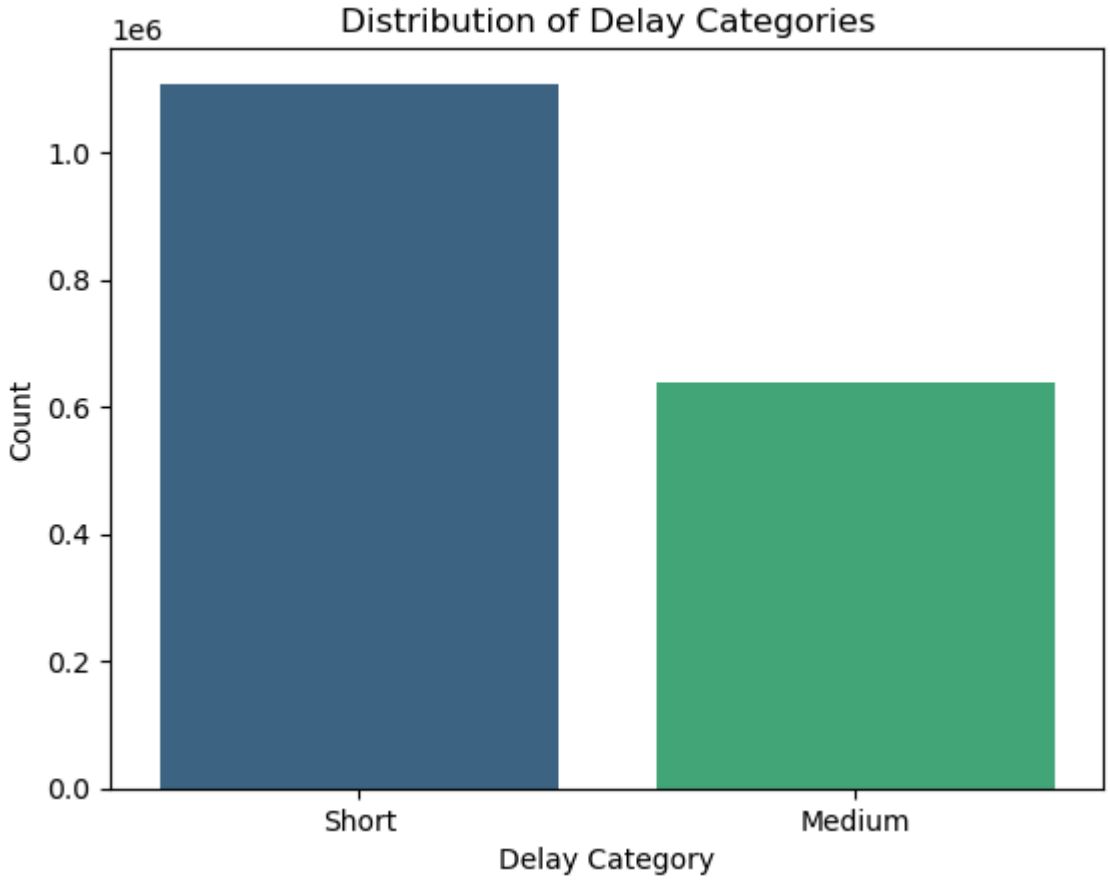
```
SatisfactionCategory
Low     73452
High    56428
Name: count, dtype: int64
AgeGroup
Adult          68081
Young Adult    41898
Senior         10054
Child           9847
Name: count, dtype: int64
```

Exploratory Data Analysis (Visualizations)

In [161…

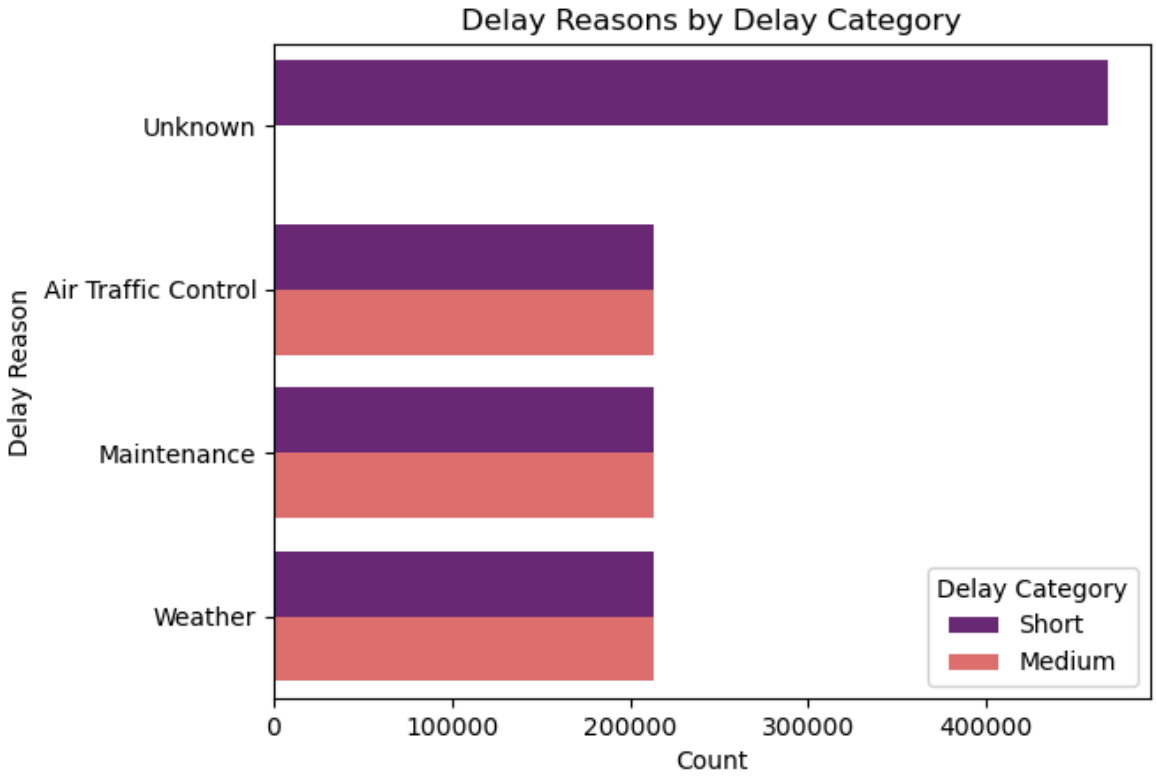
```
#Flight Delays

sns.countplot(x='DelayCategory', data=flight_delays, palette='viridis')
plt.title('Distribution of Delay Categories')
plt.xlabel('Delay Category')
plt.ylabel('Count')
plt.show()
```

```
/var/folders/rv/yn_d7x695mj2hcfz7vw9cnk40000gn/T/ipykernel_83813/1714507700.py: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.countplot(x='DelayCategory', data=flight_delays, palette='viridis')
```



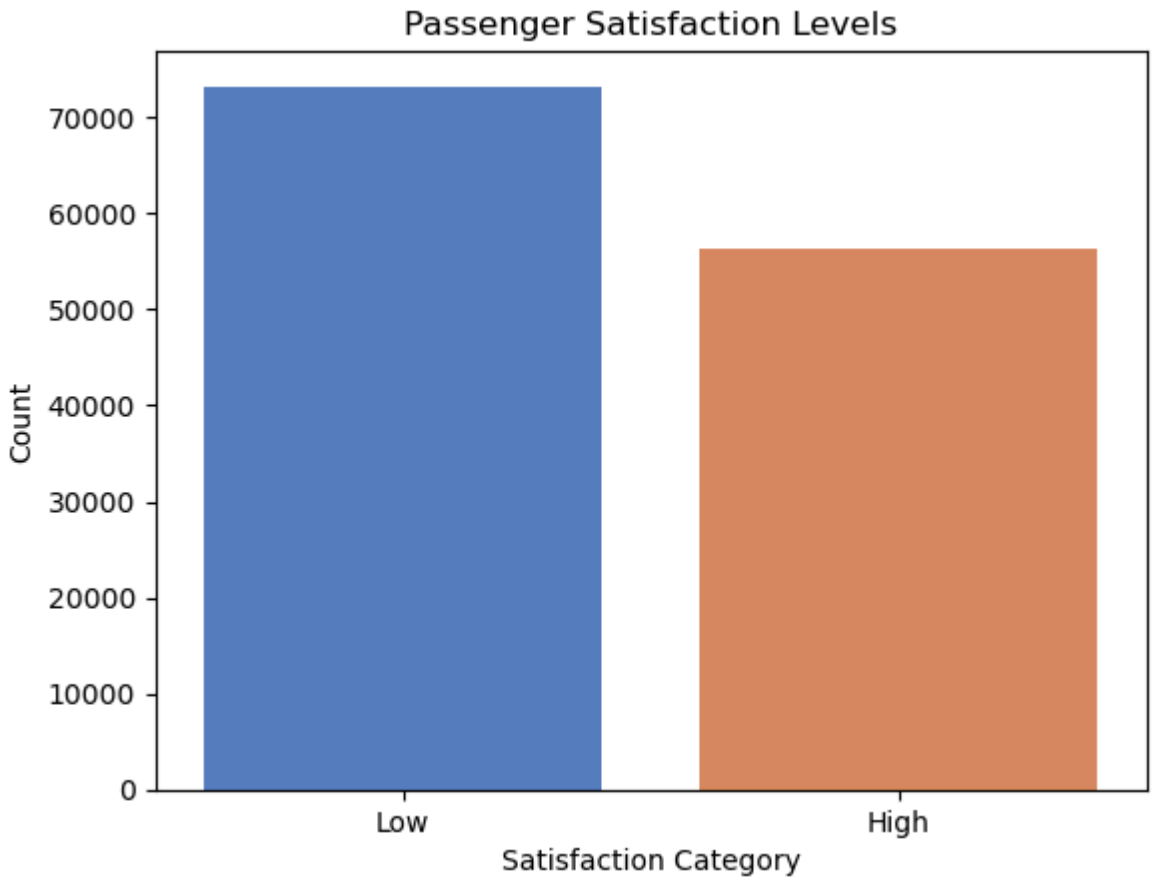
In [107…

```
#Delay Reasons by Category

sns.countplot(
    y='DelayReason',
    hue='DelayCategory',
    data=flight_delays,
    palette='magma',
    order=flight_delays['DelayReason'].value_counts().index
)
plt.title('Delay Reasons by Delay Category')
plt.xlabel('Count')
plt.ylabel('Delay Reason')
plt.legend(title='Delay Category')
plt.show()
```



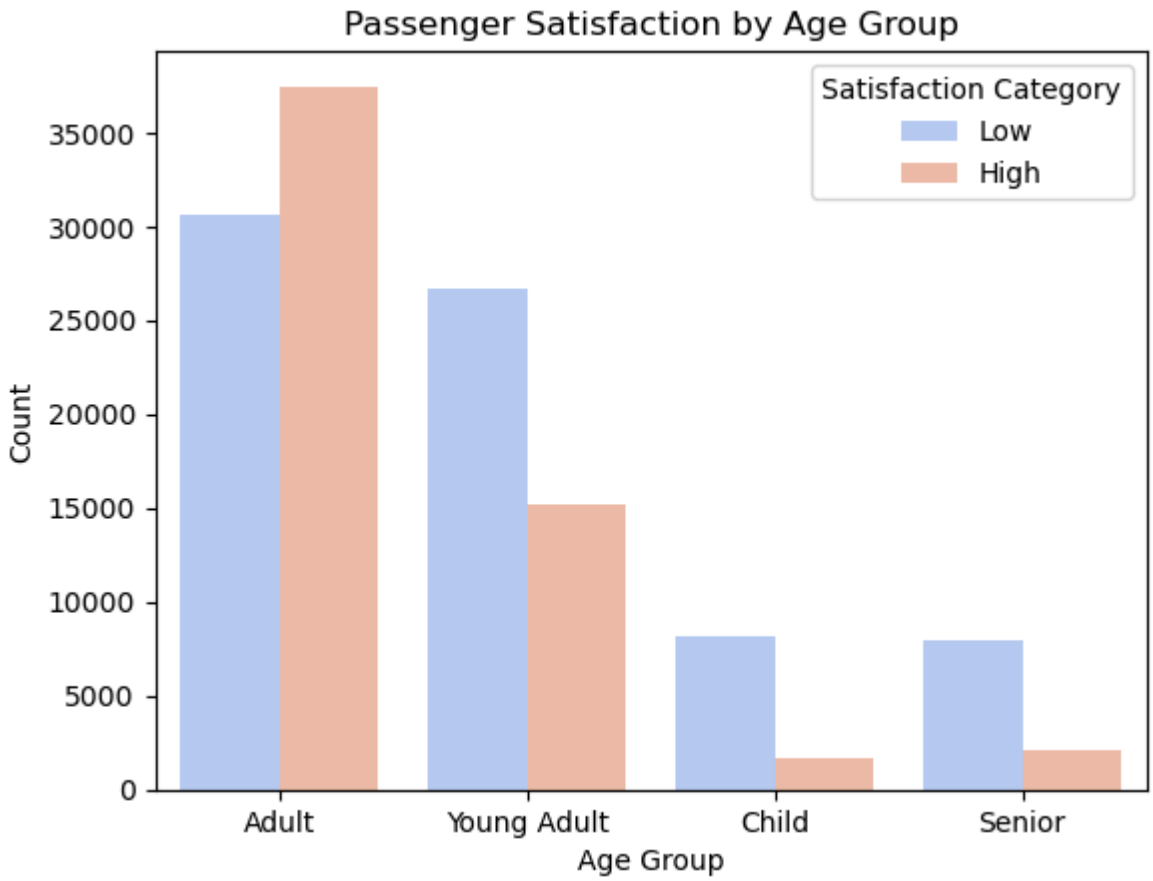
In [37]:
```
#Passenger Satisfaction

#Satisfaction Levels

sns.countplot(x='SatisfactionCategory', data=passenger_satisfaction, pale
plt.title('Passenger Satisfaction Levels')
plt.xlabel('Satisfaction Category')
plt.ylabel('Count')
plt.show()
```

```
/var/folders/rv/yn_d7x695mj2hcfz7vw9cnk40000gn/T/ipykernel_83813/62159578
7.py:5: FutureWarning:

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

  sns.countplot(x='SatisfactionCategory', data=passenger_satisfaction, pal
ette='muted')
```



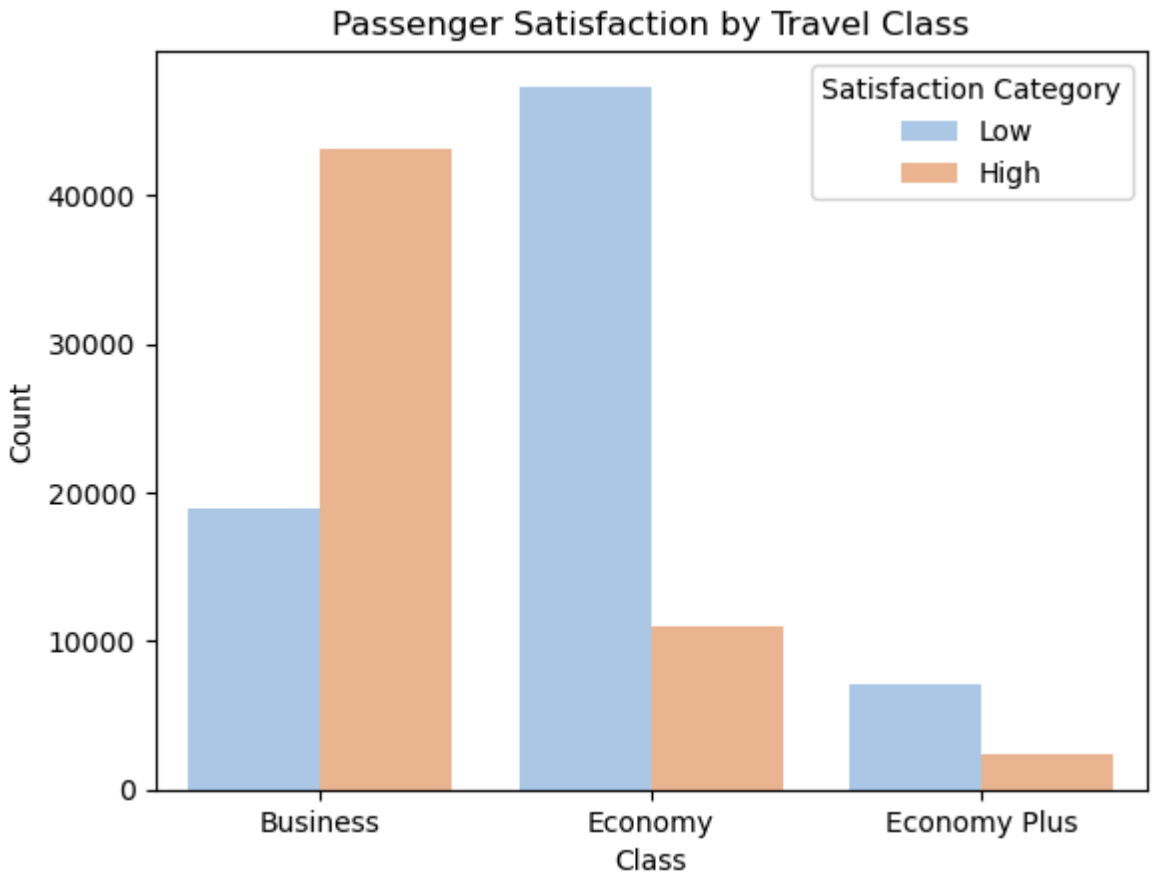
In [109…

```
#Satisfaction by Age Group

sns.countplot(x='AgeGroup', hue='SatisfactionCategory', data=passenger_sa
plt.title('Passenger Satisfaction by Age Group')
plt.xlabel('Age Group')
plt.ylabel('Count')
plt.legend(title='Satisfaction Category')
plt.show()
```



In [111…

```
#Satisfaction by Travel Class

sns.countplot(x='Class', hue='SatisfactionCategory', data=passenger_satis
plt.title('Passenger Satisfaction by Travel Class')
plt.xlabel('Class')
plt.ylabel('Count')
plt.legend(title='Satisfaction Category')
plt.show()
```

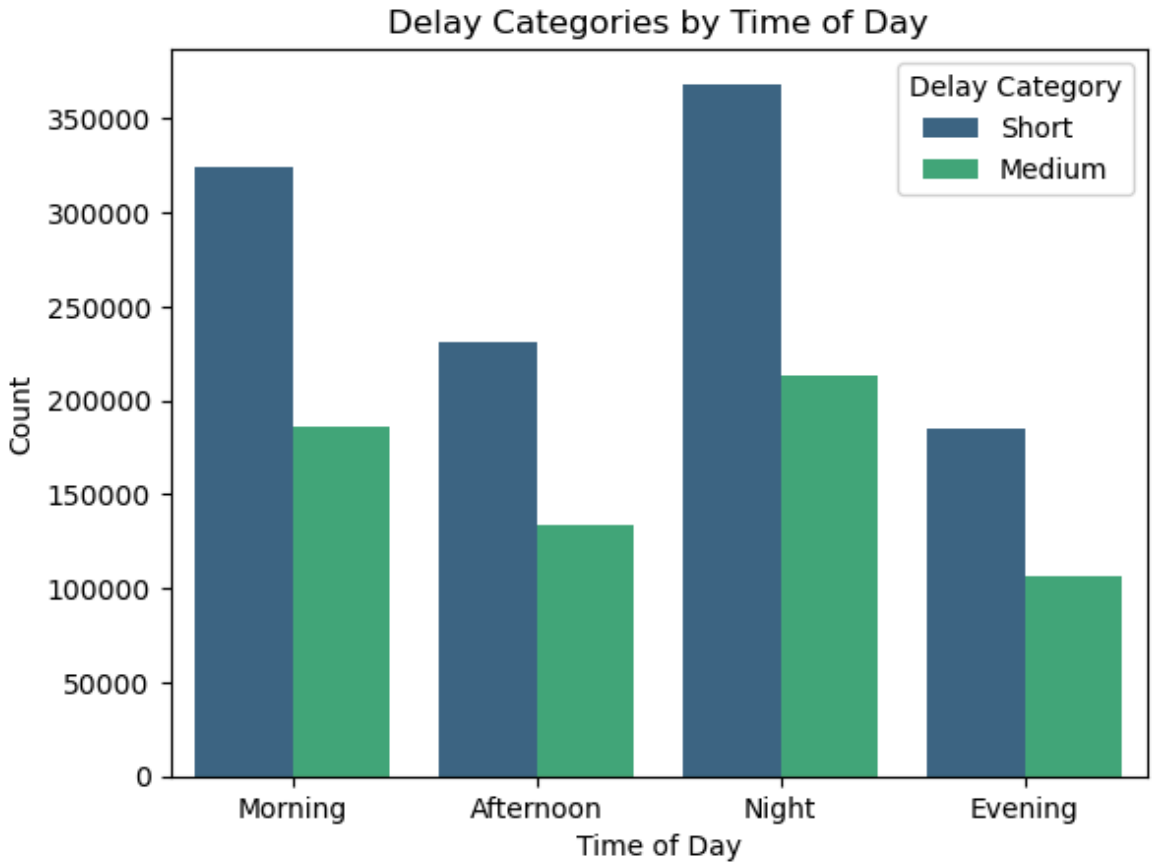


In [113…
```
#Average delay minutes by airline
avg_delay_by_airline = flight_delays.groupby('Airline')['DelayMinutes'].m

#Satisfaction by airline
satisfaction_by_airline = passenger_satisfaction.groupby('Class')['Satisf

print("Average Delay by Airline:\n", avg_delay_by_airline)
print("\nSatisfaction by Airline:\n", satisfaction_by_airline)
```

```
Average Delay by Airline:
 Airline
American Airlines     9.986951
Delta                 9.998505
United               10.002781
Southwest            10.008476
Name: DelayMinutes, dtype: float64

Satisfaction by Airline:
 SatisfactionCategory      High       Low
Class
Business               0.694434  0.305566
Economy                0.187673  0.812327
Economy Plus           0.246414  0.753586
```

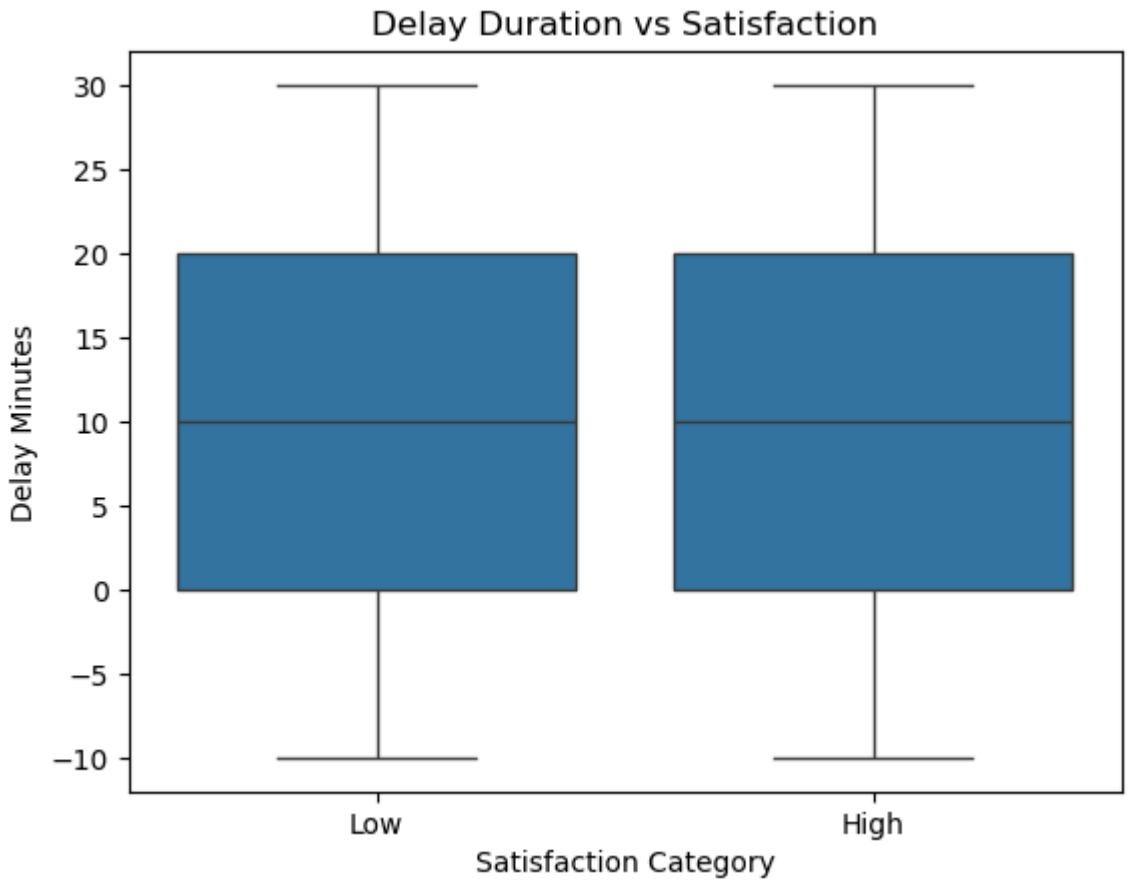
In [115…
```
#Delay Patterns by Time of Day

# Categorize Scheduled Departure Times into time periods
def time_of_day(hour):
    if 5 <= hour < 12:
        return 'Morning'
    elif 12 <= hour < 17:
        return 'Afternoon'
    elif 17 <= hour < 21:
```

```
        return 'Evening'
    else:
        return 'Night'

flight_delays['ScheduledHour'] = pd.to_datetime(flight_delays['ScheduledD
flight_delays['TimeOfDay'] = flight_delays['ScheduledHour'].apply(time_of

# Plot delay categories by time of day
sns.countplot(x='TimeOfDay', hue='DelayCategory', data=flight_delays, pal
plt.title('Delay Categories by Time of Day')
plt.xlabel('Time of Day')
plt.ylabel('Count')
plt.legend(title='Delay Category')
plt.show()
```



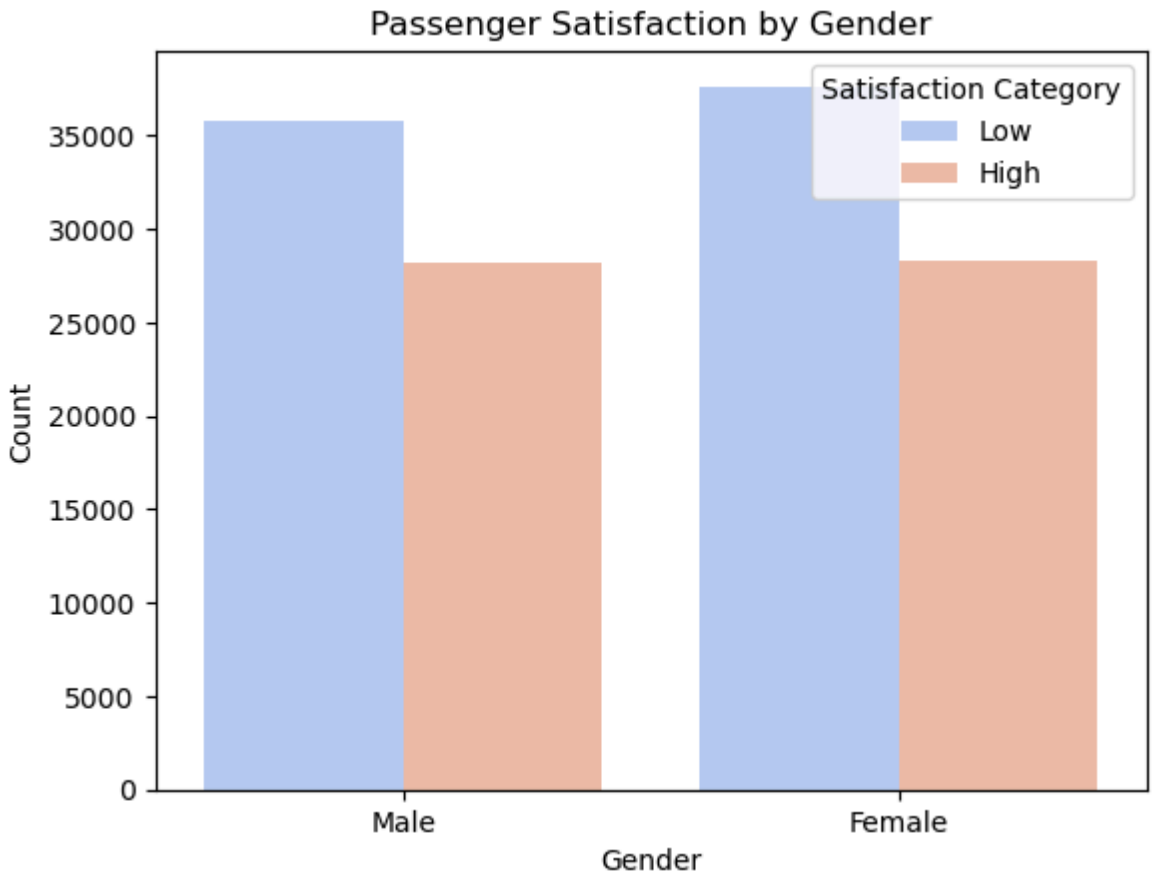
In [117…

```
#Satisfaction and Delay Correlation

#Merge datasets to analyze satisfaction vs delays
merged_data = pd.merge(
    flight_delays[['FlightID', 'DelayMinutes']],
    passenger_satisfaction[['ID', 'SatisfactionCategory']],
    left_on='FlightID',
    right_on='ID',
    how='inner'
)

#Analyze correlation between delays and satisfaction
sns.boxplot(x='SatisfactionCategory', y='DelayMinutes', data=merged_data)
plt.title('Delay Duration vs Satisfaction')
plt.xlabel('Satisfaction Category')
plt.ylabel('Delay Minutes')
plt.show()
```



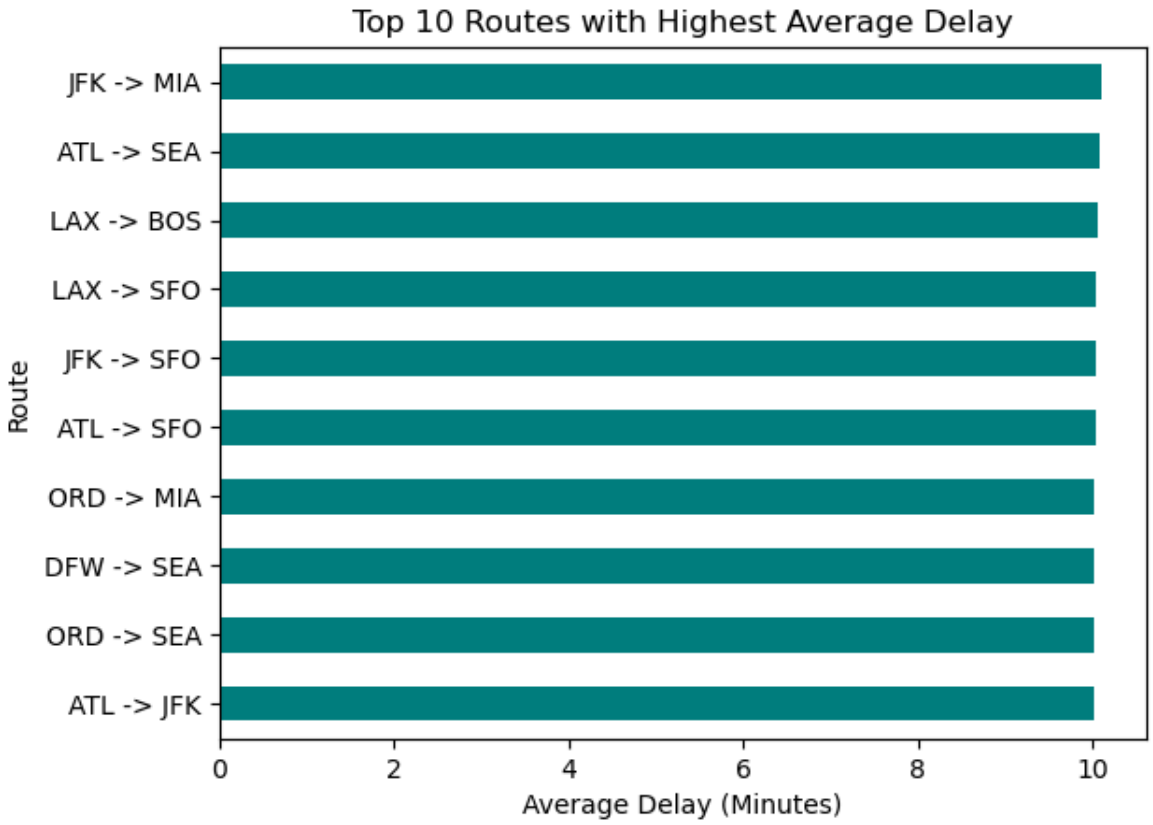
In [119…

```
#Analyze Satisfaction by Gender

#Satisfaction by Gender
sns.countplot(x='Gender', hue='SatisfactionCategory', data=passenger_sati
plt.title('Passenger Satisfaction by Gender')
plt.xlabel('Gender')
plt.ylabel('Count')
plt.legend(title='Satisfaction Category')
plt.show()
```



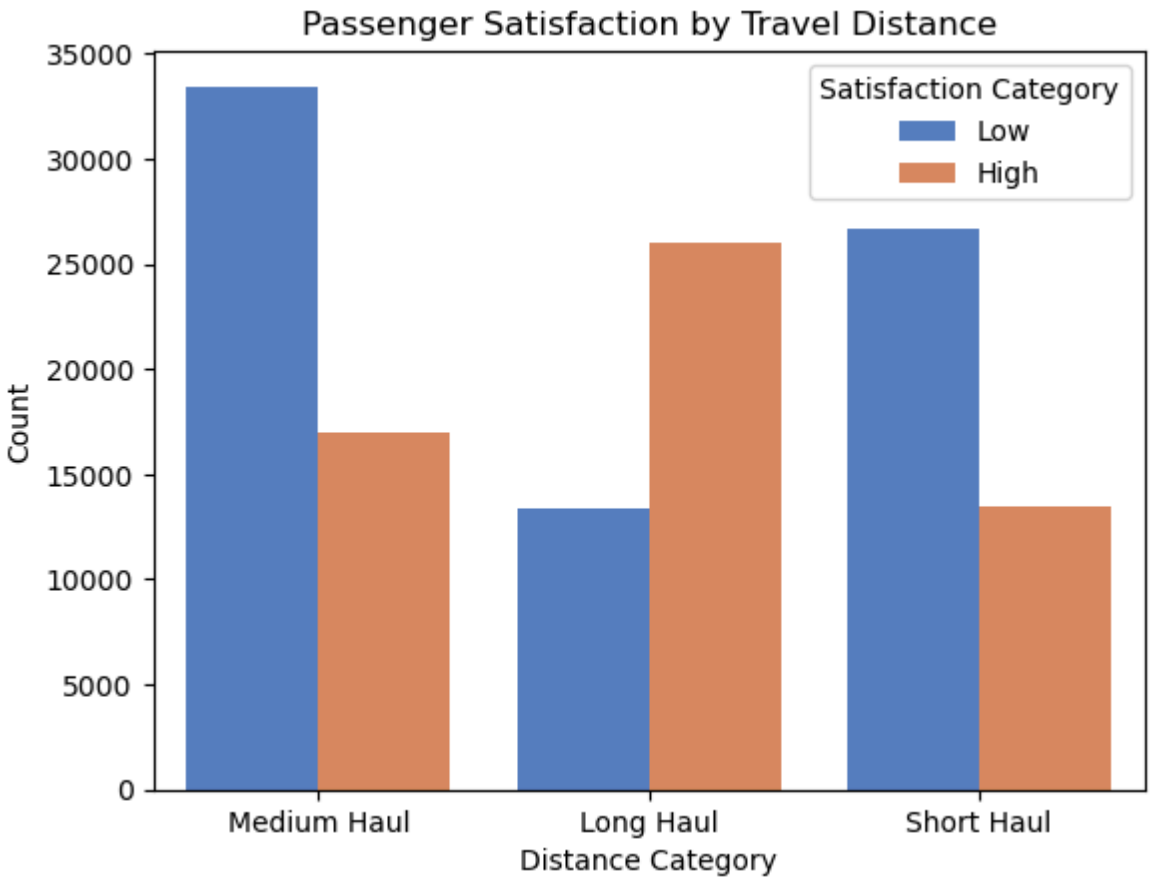
In [121…

```
#Delays by Route

#Group by Origin-Destination and calculating average delay
flight_delays['Route'] = flight_delays['Origin'] + ' -> ' + flight_delays
avg_delay_by_route = flight_delays.groupby('Route')['DelayMinutes'].mean(

#Visualizing top 10 routes with highest average delays
avg_delay_by_route.plot(kind='barh', color='teal')
plt.title('Top 10 Routes with Highest Average Delay')
plt.xlabel('Average Delay (Minutes)')
plt.ylabel('Route')
plt.gca().invert_yaxis()
plt.show()
```



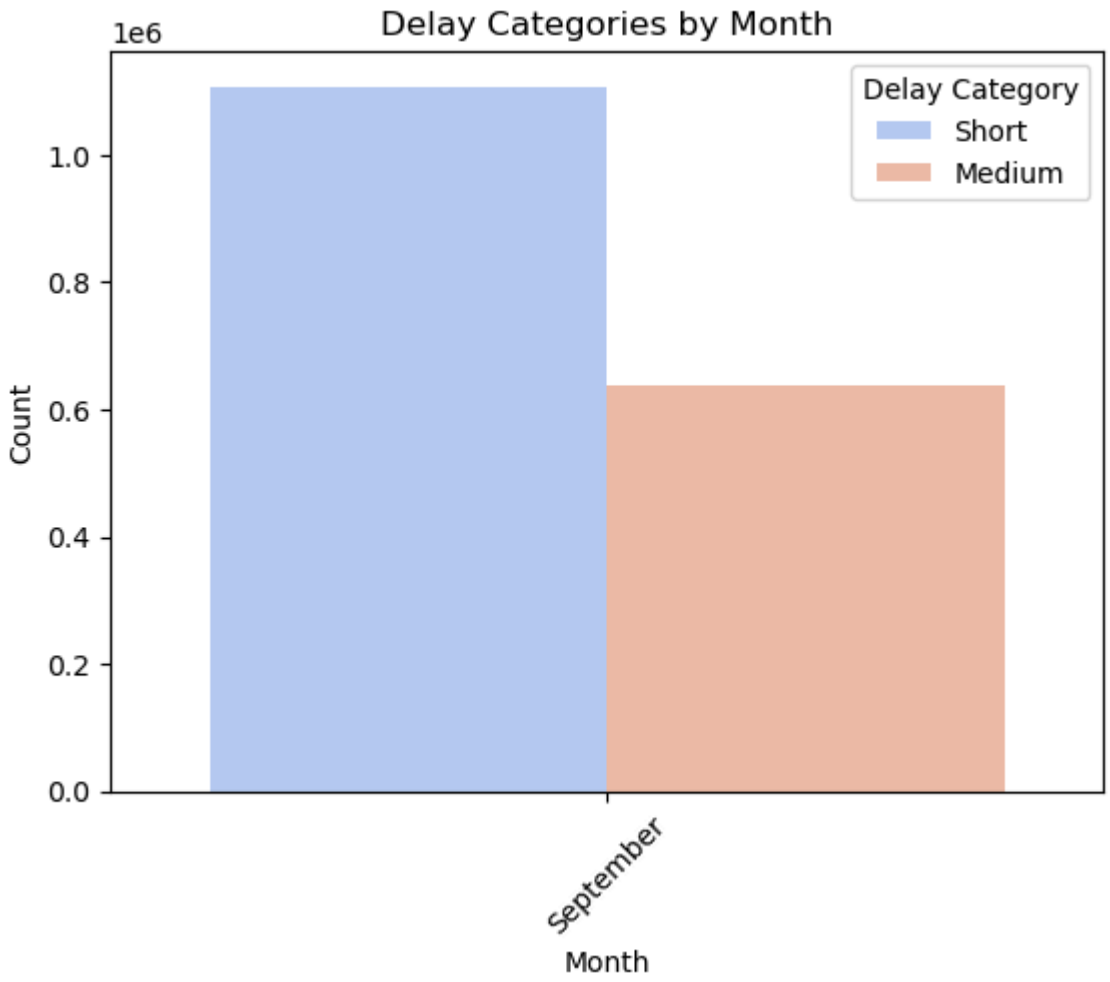
In [123…

```
#Passenger Satisfaction by Travel Distance

#Categorize Flight Distance
def distance_category(distance):
    if distance < 500:
        return 'Short Haul'
    elif distance < 1500:
        return 'Medium Haul'
    else:
        return 'Long Haul'

passenger_satisfaction['DistanceCategory'] = passenger_satisfaction['Flig

#Satisfaction by Distance Category
sns.countplot(x='DistanceCategory', hue='SatisfactionCategory', data=pass
plt.title('Passenger Satisfaction by Travel Distance')
plt.xlabel('Distance Category')
plt.ylabel('Count')
plt.legend(title='Satisfaction Category')
plt.show()
```



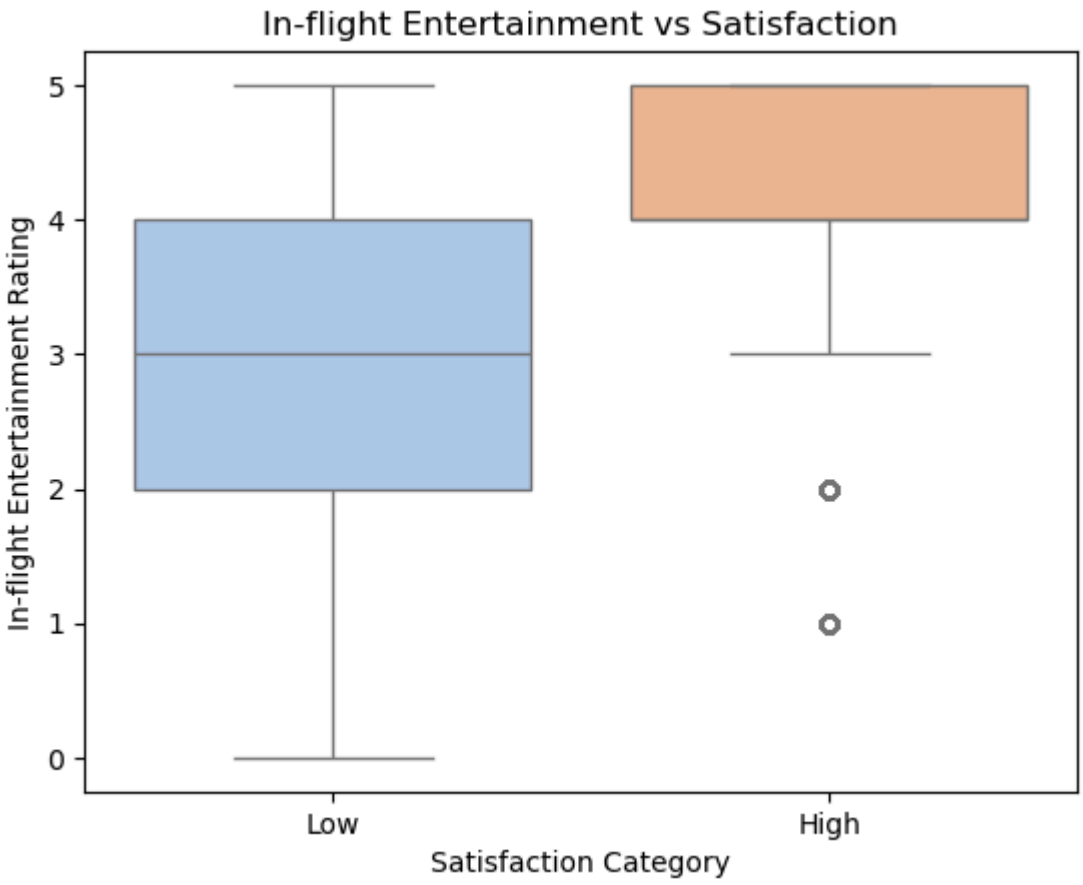
In [141…

```
#Seasonality of Delays

import calendar

#Extract month from Scheduled Departure
flight_delays['Month'] = pd.to_datetime(flight_delays['ScheduledDeparture

#Map month numbers to month names
flight_delays['MonthName'] = flight_delays['Month'].apply(lambda x: calen

#Analyze delays by month using month names
sns.countplot(x='MonthName', hue='DelayCategory', data=flight_delays, pal
plt.title('Delay Categories by Month')
plt.xlabel('Month')
plt.ylabel('Count')
plt.legend(title='Delay Category')
plt.xticks(rotation=45)
plt.show()
```



In [127…
```
#Satisfaction and Specific Flight Features

#Analyze Satisfaction by In-flight Entertainment Rating
sns.boxplot(x='SatisfactionCategory', y='In-flight Entertainment', data=p
plt.title('In-flight Entertainment vs Satisfaction')
plt.xlabel('Satisfaction Category')
plt.ylabel('In-flight Entertainment Rating')
plt.show()
```

```
/var/folders/rv/yn_d7x695mj2hcfz7vw9cnk40000gn/T/ipykernel_83813/317227108
0.py:4: FutureWarning:

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

  sns.boxplot(x='SatisfactionCategory', y='In-flight Entertainment', data=
passenger_satisfaction, palette='pastel')
```

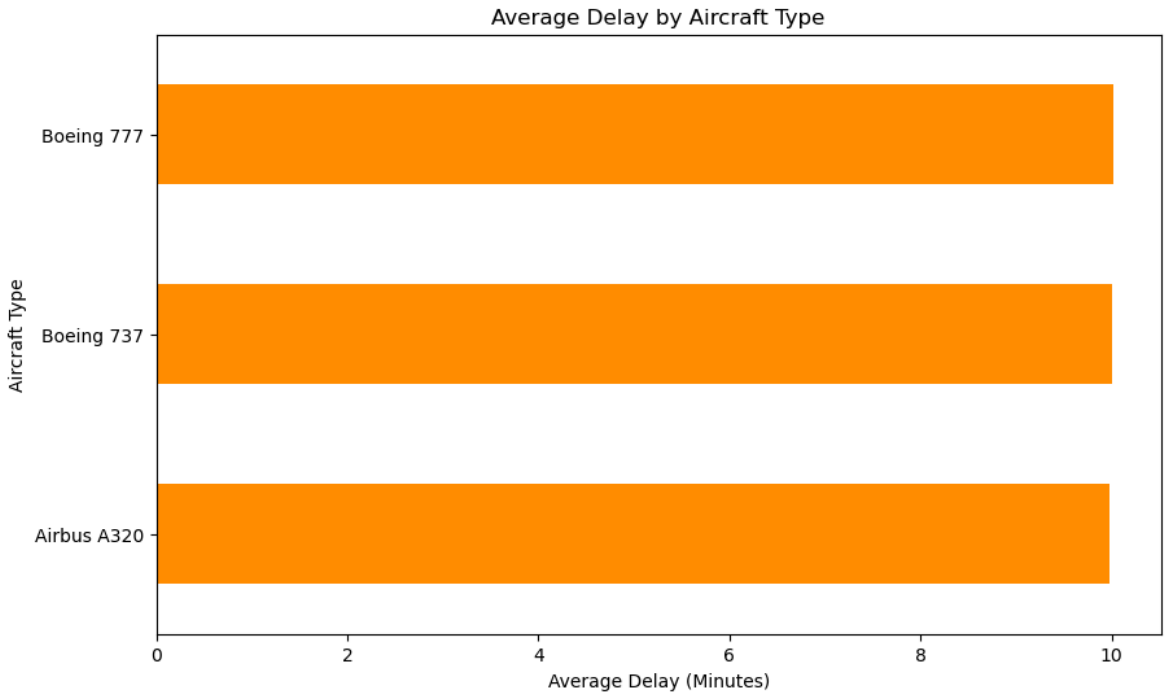


In [129…

```
#Word Cloud for Delay Reasons

from wordcloud import WordCloud

#Generate Word Cloud
delay_reasons_text = ' '.join(flight_delays['DelayReason'])
wordcloud = WordCloud(width=800, height=400, background_color='white').ge

#Display Word Cloud
plt.figure(figsize=(10, 5))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.title('Most Common Delay Reasons')
plt.show()
```

Delay Trends by Aircraft Type

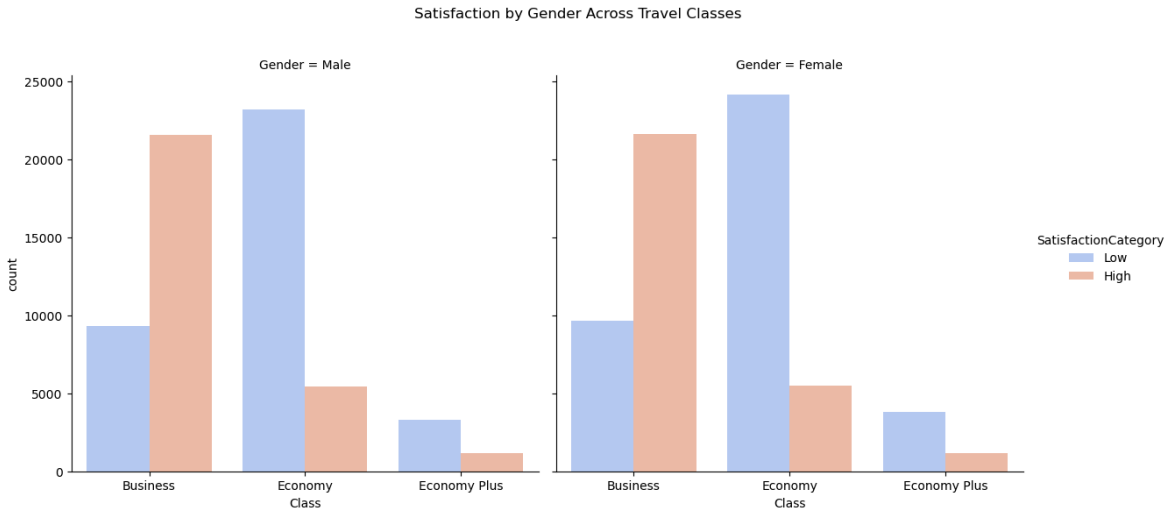
In [131…
```
# Average delay by Aircraft Type
avg_delay_by_aircraft = flight_delays.groupby('AircraftType')['DelayMinut

#Visualize
avg_delay_by_aircraft.plot(kind='barh', color='darkorange', figsize=(10,
plt.title('Average Delay by Aircraft Type')
plt.xlabel('Average Delay (Minutes)')
plt.ylabel('Aircraft Type')
plt.gca().invert_yaxis()
plt.show()
```



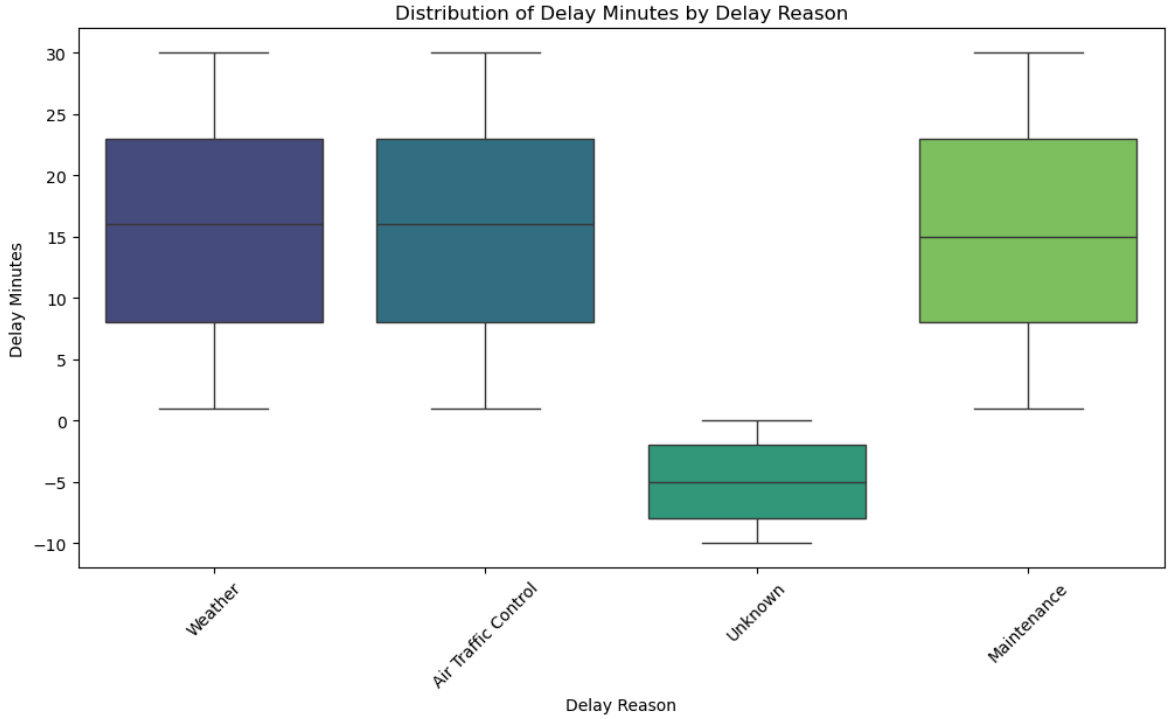
Satisfaction and Gender Differences Across Travel Classes

In [133…
```
#Satisfaction by Gender and Class
sns.catplot(x='Class', hue='SatisfactionCategory', col='Gender',
            data=passenger_satisfaction, kind='count', palette='coolwarm'
```

```
plt.subplots_adjust(top=0.85)
plt.suptitle('Satisfaction by Gender Across Travel Classes')
plt.show()

#Boxplot for Delay Minutes by Delay Reason
plt.figure(figsize=(12, 6))
sns.boxplot(x='DelayReason', y='DelayMinutes', data=flight_delays, palett
plt.title('Distribution of Delay Minutes by Delay Reason')
plt.xlabel('Delay Reason')
plt.ylabel('Delay Minutes')
plt.xticks(rotation=45)
plt.show()
```

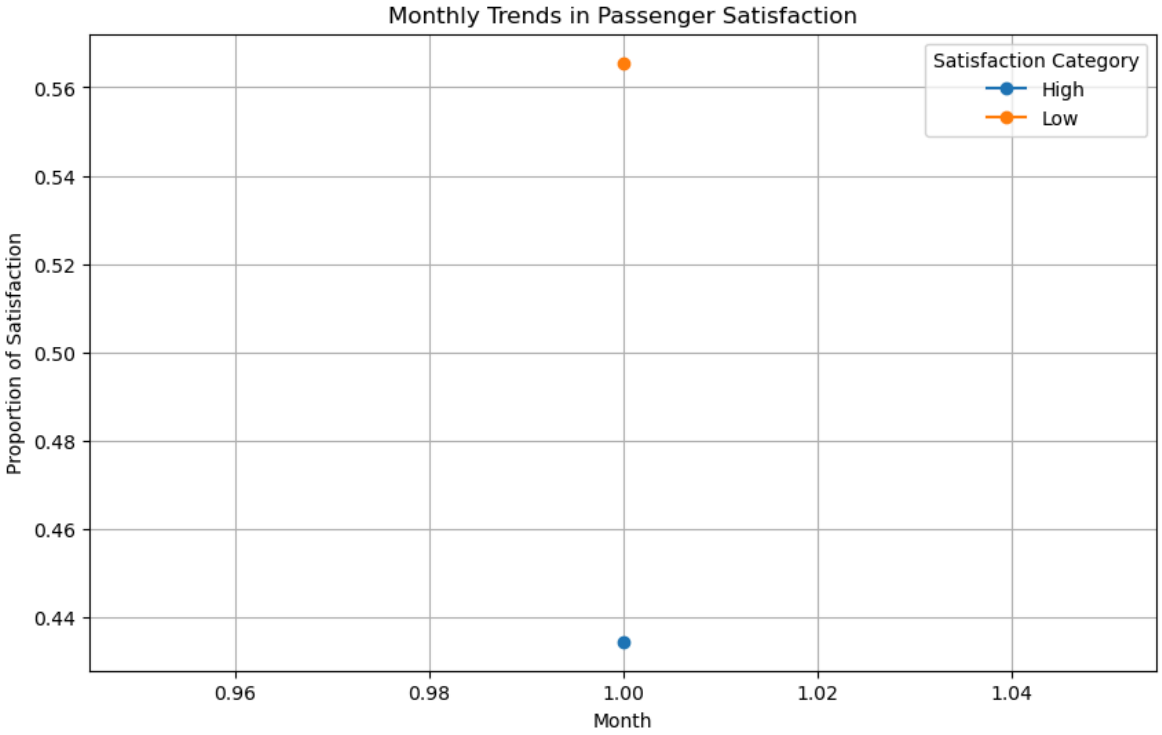



```
/var/folders/rv/yn_d7x695mj2hcfz7vw9cnk40000gn/T/ipykernel_83813/372492686
4.py:13: FutureWarning:

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

  sns.boxplot(x='DelayReason', y='DelayMinutes', data=flight_delays, palet
te='viridis')
```



In [135…

```
#Monthly Satisfaction Trends

#Add Month to passenger satisfaction data based on Scheduled Departure
passenger_satisfaction['Month'] = pd.to_datetime(passenger_satisfaction['

#Satisfaction trends by month
monthly_satisfaction = passenger_satisfaction.groupby('Month')['Satisfact

#Visualize
monthly_satisfaction.plot(kind='line', figsize=(10, 6), marker='o')
plt.title('Monthly Trends in Passenger Satisfaction')
plt.xlabel('Month')
plt.ylabel('Proportion of Satisfaction')
plt.legend(title='Satisfaction Category')
plt.grid()
plt.show()
```



In [ ]: