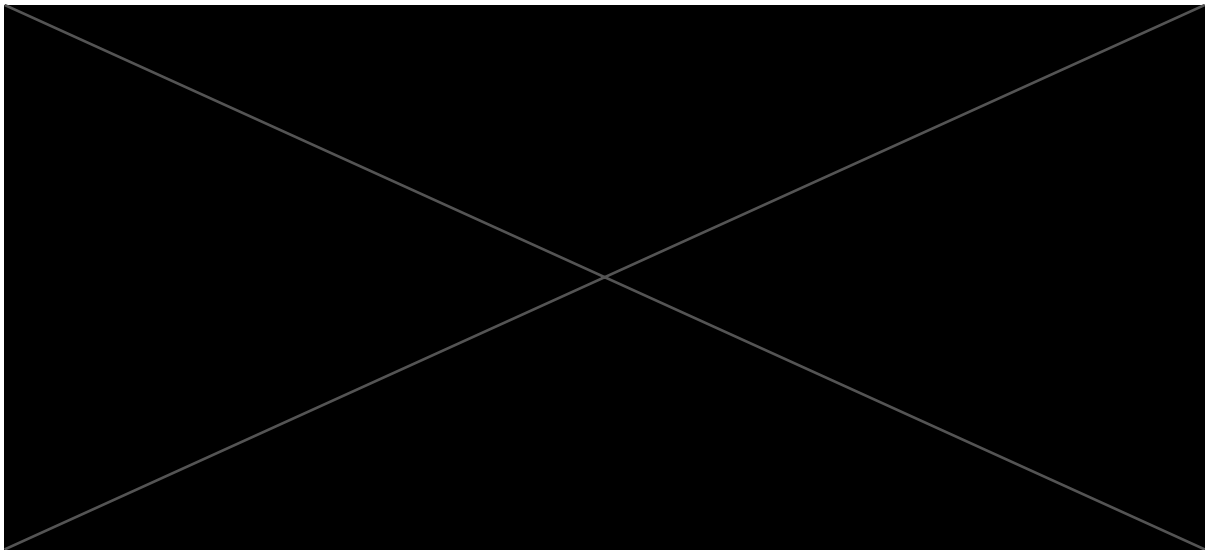
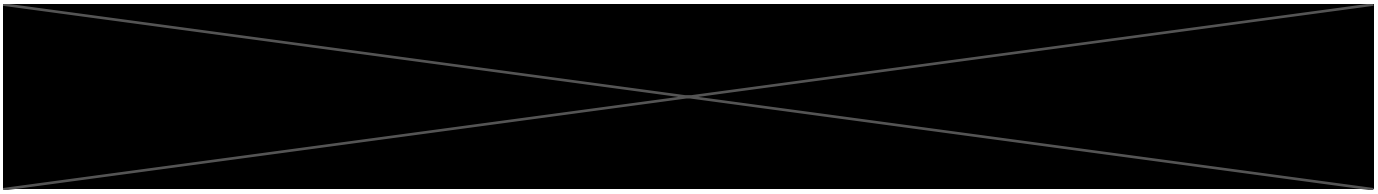




1.0.1 Step 1

[2]:
```
# Load and Display Datasets
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
import numpy as np

# Load the complaints by airport dataset
complaints_by_airport_df = pd.read_csv('complaints-by-airport.csv')
# Load the remaining datasets
iata_icao_df = pd.read_csv('iata-icao.csv')
complaints_by_subcategory_df = pd.read_csv('complaints-by-subcategory.csv')
complaints_by_category_df = pd.read_csv('complaints-by-category.csv')

# Display the first few rows of each dataset to explore
print("Complaints by Airport Dataset:")
display(complaints_by_airport_df.head())

print("\nIATA/ICAO Dataset:")
display(iata_icao_df.head())

print("\nComplaints by Subcategory Dataset:")
display(complaints_by_subcategory_df.head())
```

```
print("\nComplaints by Category Dataset:")
display(complaints_by_category_df.head())
```

```
Complaints by Airport Dataset:
```

	pdf_report_date	airport	year_month	count
0	2019-02	ABE	2015-01	0
1	2019-02	ABE	2015-02	0
2	2019-02	ABE	2015-03	0
3	2019-02	ABE	2015-04	0
4	2019-02	ABE	2015-05	2

```
IATA/ICAO Dataset:
```

	country_code	region_name	iata	icao	airport
0	AE	Abu Zaby	AAN	OMAL	Al Ain International Airport
1	AE	Abu Zaby	AUH	OMAA	Abu Dhabi International Airport
2	AE	Abu Zaby	AYM	NaN	Yas Island Seaplane Base
3	AE	Abu Zaby	AZI	OMAD	Al Bateen Executive Airport
4	AE	Abu Zaby	DHF	OMAM	Al Dhafra Air Base

	latitude	longitude
0	24.2617	55.6092
1	24.4330	54.6511
2	24.4670	54.6103
3	24.4283	54.4581
4	24.2482	54.5477

```
Complaints by Subcategory Dataset:
```

	pdf_report_date	airport	category
0	2019-02	ABE	Hazardous Materials Safety
1	2019-02	ABE	Mishandling of Passenger Property
2	2019-02	ABE	Hazardous Materials Safety
3	2019-02	ABE	Mishandling of Passenger Property
4	2019-02	ABE	Hazardous Materials Safety

	subcategory	year_month	count
0	General	2015-01	0
1	Damaged/Missing Items--Checked Baggage	2015-01	0
2	General	2015-02	0
3	Damaged/Missing Items--Checked Baggage	2015-02	0
4	General	2015-03	0

	clean_cat	clean_subcat
0	Hazardous Materials Safety	General
1	Mishandling of Passenger Property	*Damaged/Missing Items--Checked Baggage

```
2         Hazardous Materials Safety                                    General
3  Mishandling of Passenger Property  *Damaged/Missing Items--Checked Baggage
4         Hazardous Materials Safety                                    General

  clean_cat_status clean_subcat_status  is_category_prefix_removed
0         original            original                       False
1         original            original                       False
2         original            original                       False
3         original            original                       False
4         original            original                       False


Complaints by Category Dataset:

  pdf_report_date airport                          category year_month  \
0         2019-02     ABE         Hazardous Materials Safety    2015-01
1         2019-02     ABE  Mishandling of Passenger Property    2015-01
2         2019-02     ABE         Hazardous Materials Safety    2015-02
3         2019-02     ABE  Mishandling of Passenger Property    2015-02
4         2019-02     ABE         Hazardous Materials Safety    2015-03

   count                          clean_cat clean_cat_status
0      0         Hazardous Materials Safety         original
1      0  Mishandling of Passenger Property         original
2      0         Hazardous Materials Safety         original
3      0  Mishandling of Passenger Property         original
4      0         Hazardous Materials Safety         original
```

1.0.2 Step 2

[3]:
```
# Data Cleaning
# Remove duplicates, handle missing values, and clean data for each dataset

# Complaints by Airport Dataset
complaints_by_airport_df = complaints_by_airport_df.drop_duplicates()
complaints_by_airport_df = complaints_by_airport_df.replace([np.inf, -np.inf],␣
 ↪np.nan).dropna()
complaints_by_airport_df = complaints_by_airport_df.fillna(0)  # Replace␣
 ↪remaining NaN values with 0
print("\nCleaned Complaints by Airport Dataset:")
display(complaints_by_airport_df.head())

# IATA/ICAO Dataset
iata_icao_df = iata_icao_df.drop_duplicates()
iata_icao_df = iata_icao_df.dropna()
print("\nCleaned IATA/ICAO Dataset:")
display(iata_icao_df.head())
```

```
# Complaints by Subcategory Dataset
complaints_by_subcategory_df = complaints_by_subcategory_df.drop_duplicates()
complaints_by_subcategory_df = complaints_by_subcategory_df.replace([np.inf,␣
 ↪-np.inf], np.nan).dropna()
complaints_by_subcategory_df = complaints_by_subcategory_df.fillna(0)  #␣
 ↪Replace remaining NaN values with 0
print("\nCleaned Complaints by Subcategory Dataset:")
display(complaints_by_subcategory_df.head())

# Complaints by Category Dataset
complaints_by_category_df = complaints_by_category_df.drop_duplicates()
complaints_by_category_df = complaints_by_category_df.replace([np.inf, -np.
 ↪inf], np.nan).dropna()
complaints_by_category_df = complaints_by_category_df.fillna(0)  # Replace␣
 ↪remaining NaN values with 0
print("\nCleaned Complaints by Category Dataset:")
display(complaints_by_category_df.head())
```

```
Cleaned Complaints by Airport Dataset:
```

	pdf_report_date	airport	year_month	count
0	2019-02	ABE	2015-01	0
1	2019-02	ABE	2015-02	0
2	2019-02	ABE	2015-03	0
3	2019-02	ABE	2015-04	0
4	2019-02	ABE	2015-05	2

```
Cleaned IATA/ICAO Dataset:
```

	country_code	region_name	iata	icao	airport	latitude	longitude
0	AE	Abu Zaby	AAN	OMAL	Al Ain International Airport	24.2617	55.6092
1	AE	Abu Zaby	AUH	OMAA	Abu Dhabi International Airport	24.4330	54.6511
3	AE	Abu Zaby	AZI	OMAD	Al Bateen Executive Airport	24.4283	54.4581
4	AE	Abu Zaby	DHF	OMAM	Al Dhafra Air Base	24.2482	54.5477
5	AE	Abu Zaby	XSB	OMBY	Sir Bani Yas Airport	24.2836	52.5803

```
Cleaned Complaints by Subcategory Dataset:
```

	pdf_report_date	airport	category
0	2019-02	ABE	Hazardous Materials Safety

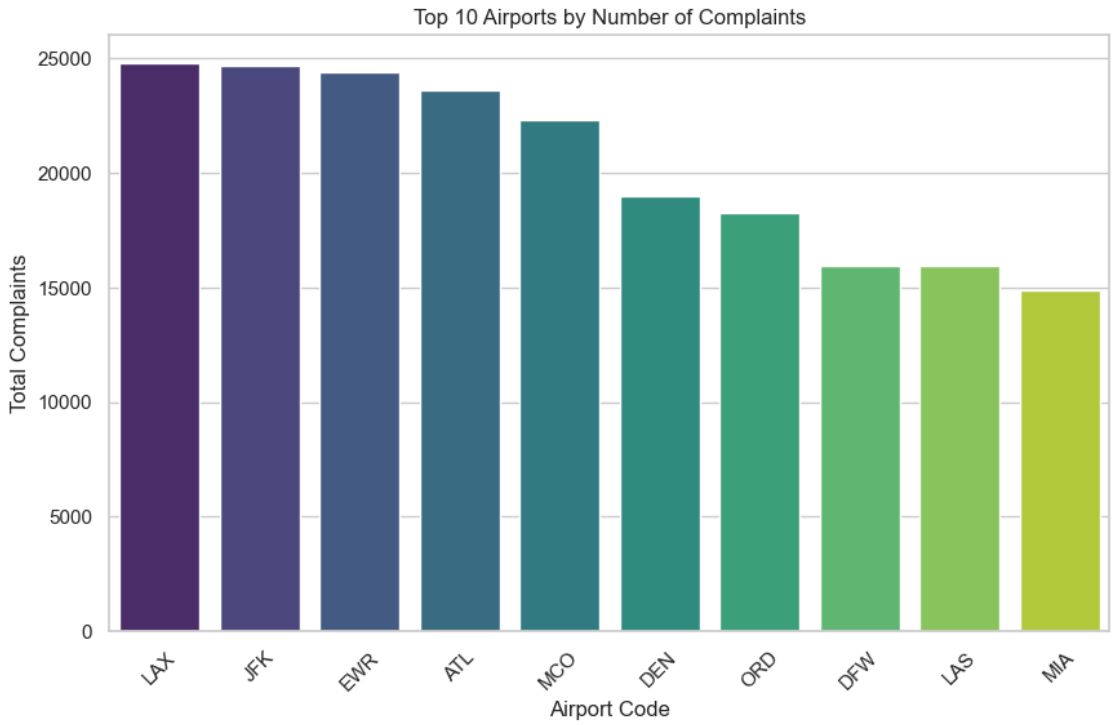
```
1         2019-02     ABE  Mishandling of Passenger Property
2         2019-02     ABE         Hazardous Materials Safety
3         2019-02     ABE  Mishandling of Passenger Property
4         2019-02     ABE         Hazardous Materials Safety

                            subcategory year_month  count  \
0                               General    2015-01      0
1  Damaged/Missing Items--Checked Baggage    2015-01      0
2                               General    2015-02      0
3  Damaged/Missing Items--Checked Baggage    2015-02      0
4                               General    2015-03      0

                         clean_cat                              clean_subcat  \
0         Hazardous Materials Safety                                   General
1  Mishandling of Passenger Property  *Damaged/Missing Items--Checked Baggage
2         Hazardous Materials Safety                                   General
3  Mishandling of Passenger Property  *Damaged/Missing Items--Checked Baggage
4         Hazardous Materials Safety                                   General

  clean_cat_status clean_subcat_status  is_category_prefix_removed
0         original            original                       False
1         original            original                       False
2         original            original                       False
3         original            original                       False
4         original            original                       False


Cleaned Complaints by Category Dataset:
  pdf_report_date airport                           category year_month  \
0         2019-02     ABE         Hazardous Materials Safety    2015-01
1         2019-02     ABE  Mishandling of Passenger Property    2015-01
2         2019-02     ABE         Hazardous Materials Safety    2015-02
3         2019-02     ABE  Mishandling of Passenger Property    2015-02
4         2019-02     ABE         Hazardous Materials Safety    2015-03

   count                         clean_cat clean_cat_status
0      0         Hazardous Materials Safety         original
1      0  Mishandling of Passenger Property         original
2      0         Hazardous Materials Safety         original
3      0  Mishandling of Passenger Property         original
4      0         Hazardous Materials Safety         original
```

1.0.3 Step 3

[4]:
```
# Visualization Setup
sns.set_theme(style="whitegrid")
```

1.0.4 Step 4

[18]:
```
# Visual 1: Visualization of the Complaints by Airport Dataset
plt.figure(figsize=(10, 6))
airport_complaints = complaints_by_airport_df.groupby('airport')['count'].sum().
 ↪sort_values(ascending=False).head(10)
sns.barplot(x=airport_complaints.index, y=airport_complaints.values,␣
 ↪palette='viridis')
plt.title('Top 10 Airports by Number of Complaints')
plt.xlabel('Airport Code')
plt.ylabel('Total Complaints')
plt.xticks(rotation=45)
plt.show()
```



Analysis: - This bar chart represents the top 10 airports that received the highest number of complaints. The x-axis displays airport codes, while the y-axis shows the total number of complaints. The colors in the plot help to distinguish between different airports.

- The key takeaway is which airports received the most complaints (most, LAX), giving insight into where potential operational or customer service issues may be prevalent.

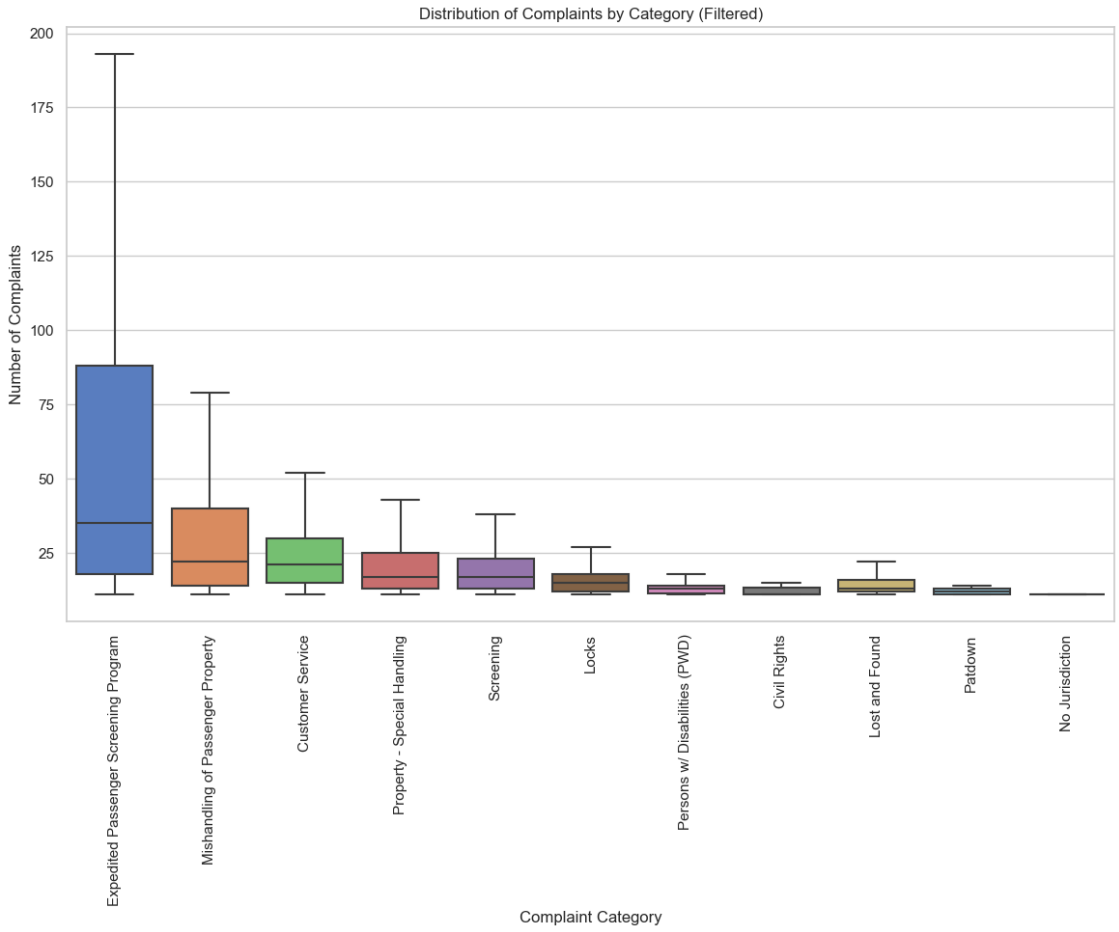
[25]:
```
# Geographical Distribution of Complaints - Airport Locations on a Map (US Only)
# Merge airport complaints with IATA/ICAO data
```

```
iata_complaints = complaints_by_airport_df.groupby('airport')['count'].sum().
 ↪reset_index()
merged_df = pd.merge(iata_complaints, iata_icao_df, left_on='airport',␣
 ↪right_on='iata')

# Filter for US airports only
us_airports_df = merged_df[merged_df['country_code'] == 'US']

# Plotting the geographical distribution of complaints for US only
fig = px.scatter_geo(us_airports_df,
                     lat='latitude',
                     lon='longitude',
                     size='count',
                     hover_name='airport_x',
                     title='Geographical Distribution of Complaints by Airport␣
 ↪(US Only)',
                     color='count',
                     color_continuous_scale='Viridis',
                     scope='usa',
                     projection='albers usa')
fig.show()
```

Analysis

- Airports with larger and darker markers are those that received a higher number of complaints. By examining the locations of these markers, it becomes apparent which airports might be facing significant operational or customer service issues.
- Airports that are major hubs or have higher passenger volumes might naturally have more complaints due to the sheer number of travelers. This visual helps pinpoint those specific airports, facilitating a deeper analysis to determine whether the complaints are proportionate to the airport's size and activity or indicative of underlying issues.

[20]:
```
# Visual 3: Box Plot of Complaints by Category (Cleaned)
plt.figure(figsize=(14, 8))
# Filter out categories with very few complaints to make the plot clearer
filtered_complaints_by_category_df =␣
 ↪complaints_by_category_df[complaints_by_category_df['count'] > 10]
sns.boxplot(data=filtered_complaints_by_category_df, x='category', y='count',␣
 ↪palette='muted', showfliers=False)
plt.title('Distribution of Complaints by Category (Filtered)')
plt.xlabel('Complaint Category')
plt.ylabel('Number of Complaints')
plt.xticks(rotation=90)
plt.show()
```

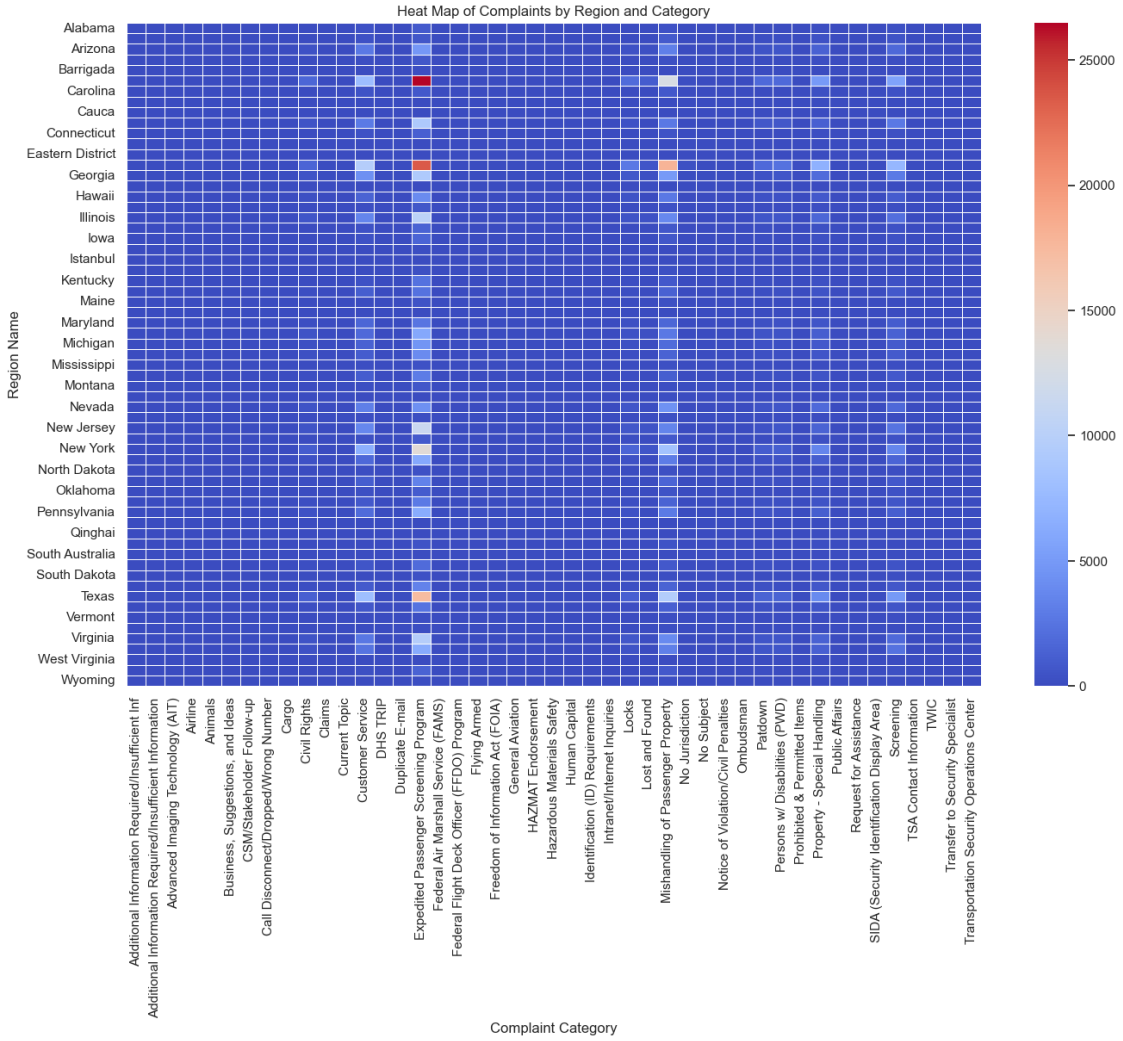


Analysis

- This box plot visualizes the distribution of complaints across different complaint categories. The y-axis indicates the number of complaints, and the x-axis shows the complaint categories.
- The box plot is filtered to include categories with a significant number of complaints, which helps to remove noise from categories with very few complaints. This visualization helps to identify categories with the most variability in complaints.

[21]:
```
# Visual 4: Heat Map of Complaints by Region and Category
region_category_complaints = pd.merge(complaints_by_category_df, iata_icao_df,␣
 ↪left_on='airport', right_on='iata')
region_category_pivot = region_category_complaints.
 ↪pivot_table(index='region_name', columns='category', values='count',␣
 ↪aggfunc='sum', fill_value=0)
plt.figure(figsize=(16, 10))
sns.heatmap(region_category_pivot, cmap='coolwarm', annot=False, linewidths=.5)
plt.title('Heat Map of Complaints by Region and Category')
plt.xlabel('Complaint Category')
```

```
plt.ylabel('Region Name')
plt.show()
```



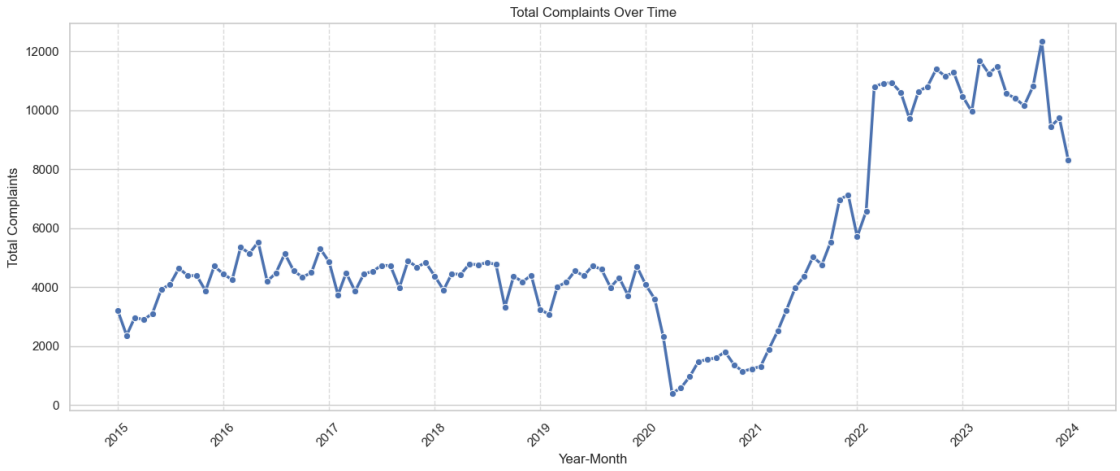
Analysis

- The heat map shows the distribution of complaints across different regions (y-axis) and complaint categories (x-axis). The color gradient represents the count of complaints in each cell, with darker colors indicating higher counts.
- This visual provides a high-level overview of which regions have the most complaints in certain categories, allowing for targeted interventions in those areas.

[22]:
```
# Visual 5: Trend Line of Complaints Over Time
plt.figure(figsize=(14, 6))
time_series = complaints_by_airport_df.groupby('year_month')['count'].sum().
↪reset_index()
```

```
time_series['year_month'] = pd.to_datetime(time_series['year_month'],␣
 ↪errors='coerce', format='%Y-%m')
time_series = time_series.dropna()  # Drop rows with invalid dates
time_series = time_series.sort_values('year_month')  # Ensure data is sorted by␣
 ↪date
sns.lineplot(data=time_series, x='year_month', y='count', marker='o',␣
 ↪color='b', linewidth=2.5)
plt.title('Total Complaints Over Time')
plt.xlabel('Year-Month')
plt.ylabel('Total Complaints')
plt.xticks(rotation=45)
plt.grid(axis='x', linestyle='--', alpha=0.7)
plt.tight_layout()
plt.show()
```

```
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning:

use_inf_as_na option is deprecated and will be removed in a future version.
Convert inf values to NaN before operating instead.

/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning:

use_inf_as_na option is deprecated and will be removed in a future version.
Convert inf values to NaN before operating instead.
```

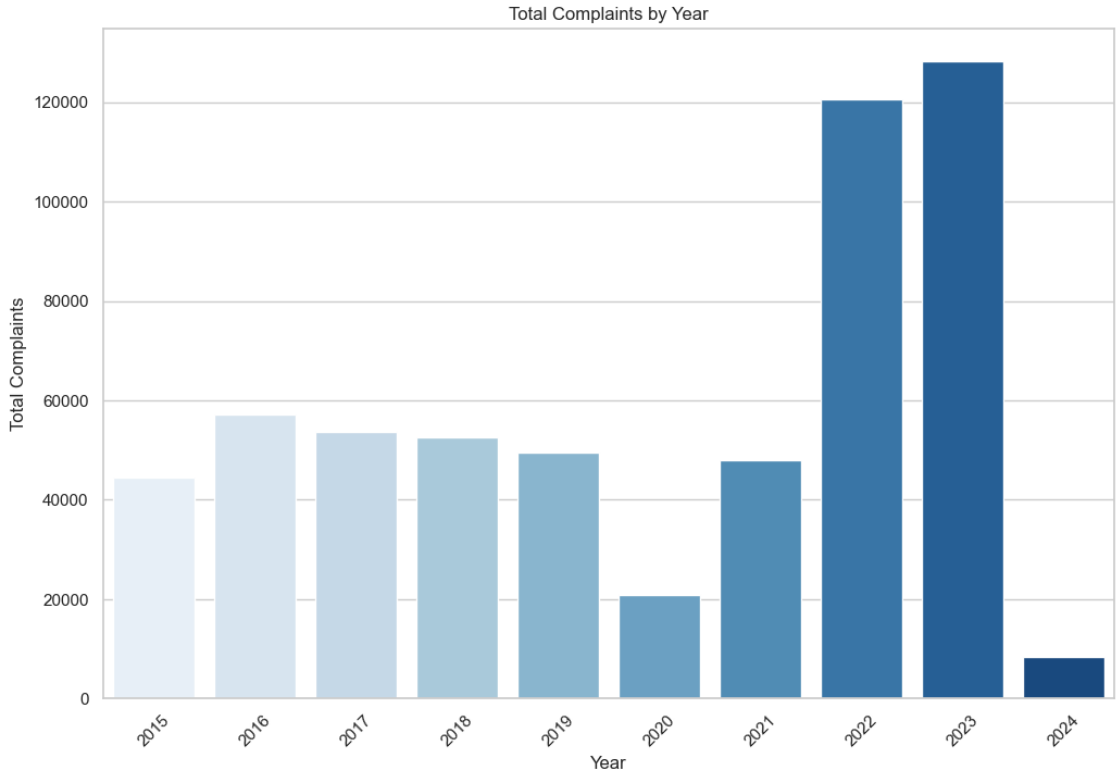


Analysis

- This line plot depicts the trend of total complaints over time. The x-axis represents the year and month, while the y-axis indicates the total number of complaints.

- The plot allows us to observe changes in the volume of complaints over time, highlighting any trends, seasonal variations, or notable spikes, such as the increase during certain years.

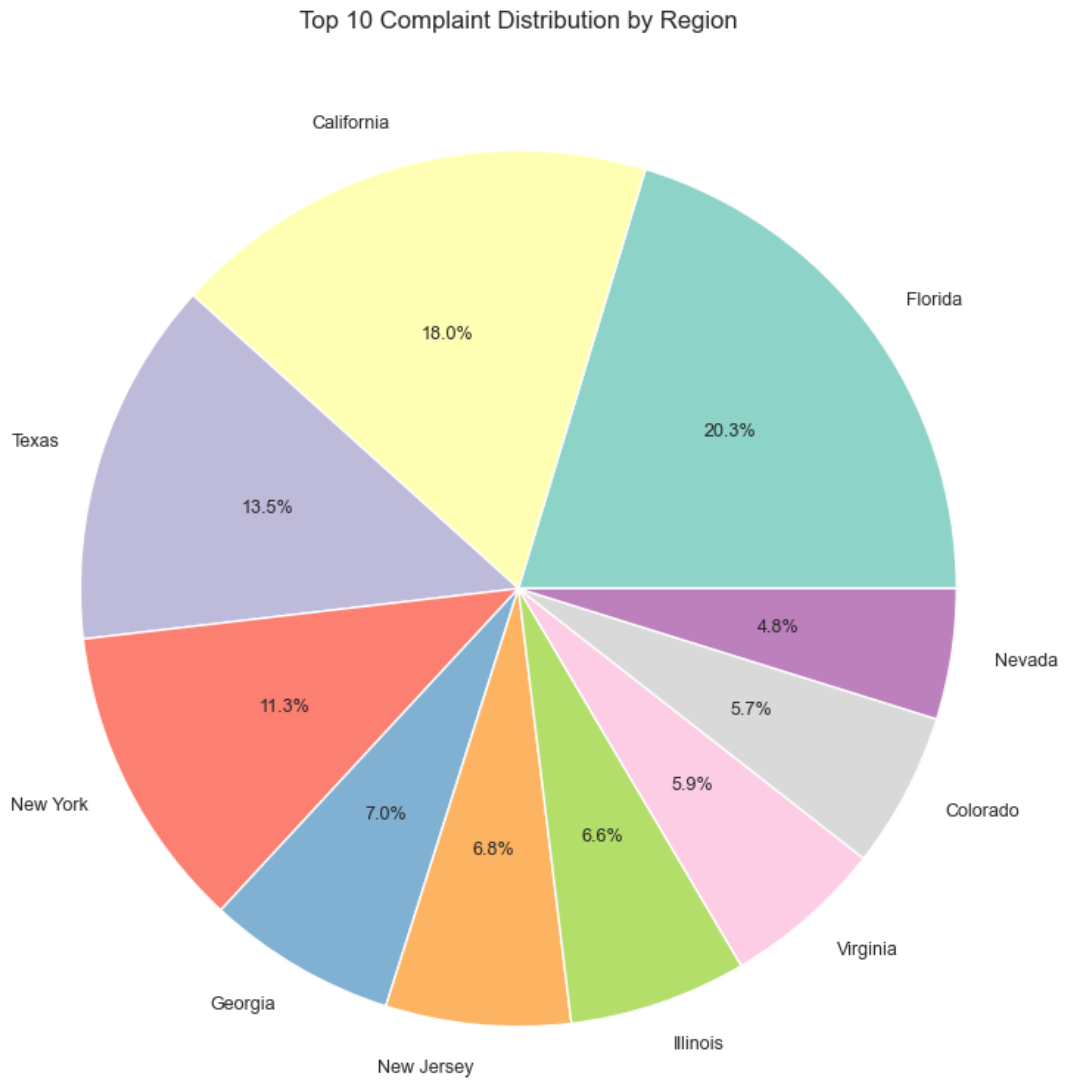
[23]:
```
# Visual 6: Bar Plot of Complaints by Year
complaints_by_airport_df['year'] = pd.
 ↪to_datetime(complaints_by_airport_df['year_month'], errors='coerce',␣
 ↪format='%Y-%m').dt.year
plt.figure(figsize=(12, 8))
year_complaints = complaints_by_airport_df.groupby('year')['count'].sum().
 ↪reset_index()
sns.barplot(data=year_complaints, x='year', y='count', palette='Blues')
plt.title('Total Complaints by Year')
plt.xlabel('Year')
plt.ylabel('Total Complaints')
plt.xticks(rotation=45)
plt.show()
```



Analysis

- This bar plot represents the total number of complaints received each year. The x-axis displays the years, while the y-axis shows the total number of complaints for each year.
- The visual helps to compare the volume of complaints from year to year, which is useful for assessing the effectiveness of complaint mitigation strategies or identifying problematic years.

[24]:
```
# Visual 7: Pie Chart of Complaint Distribution by Region (Simplified and Sorted Labels)
region_complaints = merged_df.groupby('region_name')['count'].sum().reset_index()
region_complaints = region_complaints.sort_values(by='count', ascending=False).head(10)  # Show only top 10 regions
plt.figure(figsize=(10, 8))
plt.pie(region_complaints['count'], labels=region_complaints['region_name'], autopct='%1.1f%%', colors=sns.color_palette('Set3'), textprops={'fontsize': 9}, wedgeprops={'linewidth': 1, 'edgecolor': 'white'})
plt.title('Top 10 Complaint Distribution by Region')
plt.tight_layout()
plt.show()
```



Analysis

- This pie chart shows the distribution of complaints among the top 10 regions. The chart segments represent different regions, and the percentage labels indicate the proportion of complaints contributed by each region.
- This visual provides insight into which regions generate the most complaints, allowing the identification of regions that may need targeted attention or improvements.

1.0.5 SUMMARY PAPER -

Audience: The intended audience for this analysis includes travelers who frequently use airports across the United States. The audience is likely not familiar with the specifics of airport operations or complaint categories, so the information needs to be presented in an accessible manner, avoiding technical jargon.

Purpose: The primary purpose of this analysis is to inform travelers about the common issues they may encounter at various airports and provide insights to help them make informed travel decisions. By highlighting the most frequently reported complaints, travelers can better understand what to expect and how to prepare. The goal is also to raise awareness about potential areas of concern, encouraging travelers to provide feedback to airports and TSA to improve the overall travel experience.

Call to Action: Travelers are encouraged to share their experiences and provide feedback whenever they encounter issues at airports. This feedback is crucial for improving airport services and addressing the most common issues reported, such as property mismanagement and screening delays. Additionally, travelers should consider planning their journeys with these insights in mind, avoiding airports with consistently high complaint rates if possible.

Medium: The information is presented in an infographic format, which is easy for travelers to understand at a glance. Infographics provide a visually engaging way to convey key points, using charts, graphs, and icons to highlight the most important data. This medium is effective for presenting information in airports, on travel websites, or on social media, where travelers can quickly absorb the key messages.

Design Choices: The infographic design follows Gestalt's principles for effective visual communication. **Color** is used to differentiate complaint categories and highlight key insights, with a simple color palette to avoid overwhelming the audience. **Text** is minimized and supplemented with icons and images to make the information digestible. **Alignment** and **spacing** are carefully managed to ensure clarity and readability. **Sizing** is used to draw attention to the most significant data points, such as the airports with the highest complaints or the most common complaint categories.

Ethical Considerations: The data was carefully processed to ensure accuracy and transparency. Duplicate records were removed, missing values were addressed, and only relevant data was included to provide clear and honest insights. The infographic is intended to inform travelers without causing undue alarm or bias. All sources of data are credible, and the presentation of findings is designed to be balanced, focusing on providing useful information rather than exaggerating issues.