PRACTICAL 1

Name:	Harsh Shah	**Semester:**	VII	**Division:**	6
Roll No.:	21BCP359	**Date:**	22-07-24	**Batch:**	G11
Aim:	Discuss and analyze different data visualization tools.				

Objective

The objective of this lab assignment is to explore and analyse various data visualization tools used for representing and understanding complex datasets. Through this assignment, you will gain insights into the strengths, weaknesses, and practical applications of different visualization tools.

Introduction to Data Visualization

Data visualization is the graphical representation of information and data. By using visual elements like charts, graphs, and maps, data visualization tools provide an accessible way to see and understand trends, outliers, and patterns in data. It's an essential skill in the data analysis process, helping to make complex data more understandable, insightful, and actionable.

Key Types of Data Visualizations:

- **Charts**: Including bar charts, line charts, pie charts, and histograms, these are used to compare different categories or show changes over time.
- **Graphs**: Such as scatter plots and area graphs, these help in understanding relationships between variables.
- **Maps**: Geographic data can be visualized using maps to highlight spatial patterns and distributions.
- **Infographics**: These combine various visualization elements to tell a story or provide a comprehensive overview of data.
- **Dashboards**: Interactive platforms that allow users to manipulate data and visuals to explore different perspectives and insights.

Data Visualisation Tools

- **Microsoft Excel**: A widely used tool that offers basic to advanced charting and graphing capabilities.
- **Python Libraries**: Libraries like Matplotlib, Seaborn, and Plotly are popular among data scientists for creating a wide range of static, animated, and interactive visualizations.
- **Tableau**: A powerful tool for creating interactive and shareable dashboards.
- **Power BI**: A business analytics service by Microsoft that provides interactive visualizations and business intelligence capabilities.
- **Google Data Studio**: A free tool that transforms your data into informative, easy-to-read, and shareable dashboards and reports.
- **R/JS Libraries**: Libraries like ggplot2(R), plotly(R), D3.js(JS), plotly(JS) are popular among for creating a wide range of static, animated, and interactive visualizations.

Brief Overview of Tool Capabilities and Features

a) Matplotlib (Python)

- Matplotlib is a versatile plotting library in Python that produces publication-quality figures in a variety of formats and interactive environments.
- Wide Range of Plots: Supports line plots, scatter plots, bar charts, histograms, pie charts, error bars, and more.
- Subplots and Axes: Allows creation of complex, multi-plot layouts.
- Use Cases: Scientific Research, Financial Analysis

b) Seaborn (Python)

- Seaborn is a statistical data visualization library based on Matplotlib, providing a high-level interface for drawing attractive and informative statistical graphics.
- Complex Plots Simplified: Simplifies the creation of complex visualizations such as heatmaps, time series, violin plots, and pair plots.
- Statistical Plotting: Integrates with pandas Data Frames and supports statistical aggregation and plotting.
- Use Cases: Exploratory Data Analysis (EDA), Statistical Data Visualization

c) Power BI

- Power BI is a business analytics service by Microsoft that delivers insights to enable fast, informed decisions.
- Interactive Dashboards: Allows creation of interactive and shareable dashboards.
- Data Connectivity: Connects to a wide variety of data sources, including databases, web services, and Excel spreadsheets.
- Data Modelling: Offers robust data modelling and transformation capabilities.
- Real-Time Analytics: Supports real-time data visualization and streaming analytics.
- Use Cases: Business Intelligence and Reporting, Real-Time Data Monitoring

d) Plotly (Python)

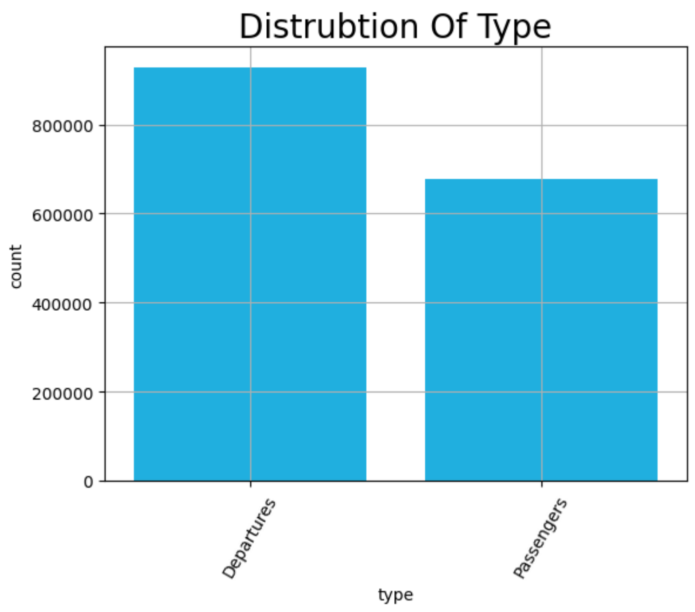
- Plotly is an interactive graphing library for Python, known for its high-quality and interactive plots.
- Interactive Graphs: Produces interactive plots that can be embedded in web applications or dashboards.
- Dash by Plotly: Integrated with Dash, a framework for building interactive web applications with Python.
- Ease of Use: User-friendly API and integration with pandas for quick and easy plotting.
- Cross-Platform: Plots can be viewed in Jupyter Notebooks, standalone HTML files, or web applications.
- Use Cases: Interactive Web Applications, 3D Data Visualization

Code Snippets

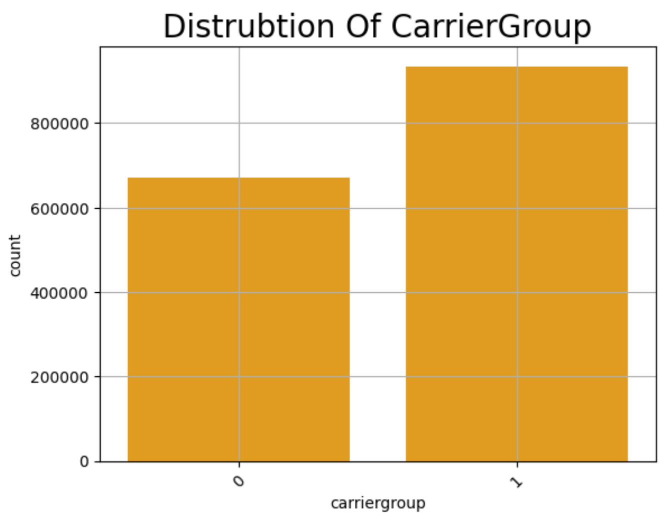
Dataset Used: U.S. International Air Traffic data (1990-2020) - Kaggle

1. Using Seaborn and Matplotlib

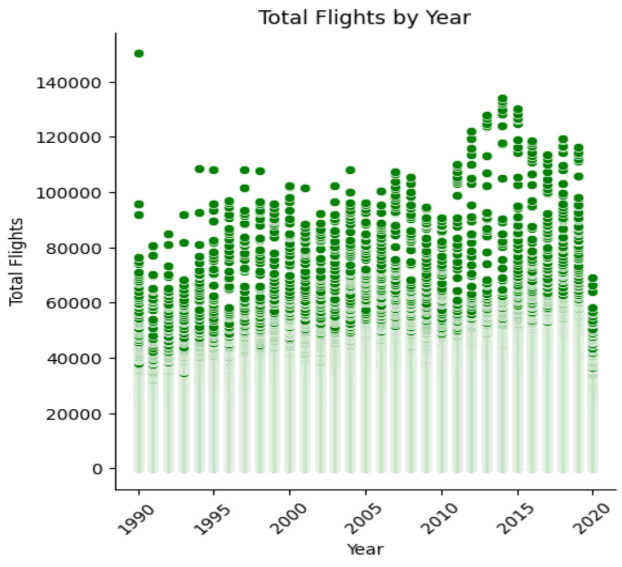
```
sns.countplot(x='type',data=full_data, color="deepskyblue")
plt.title('Distrubtion Of Type',fontsize=20 , color="black")
plt.grid(True)
plt.xticks(rotation=60)
plt.show()
```



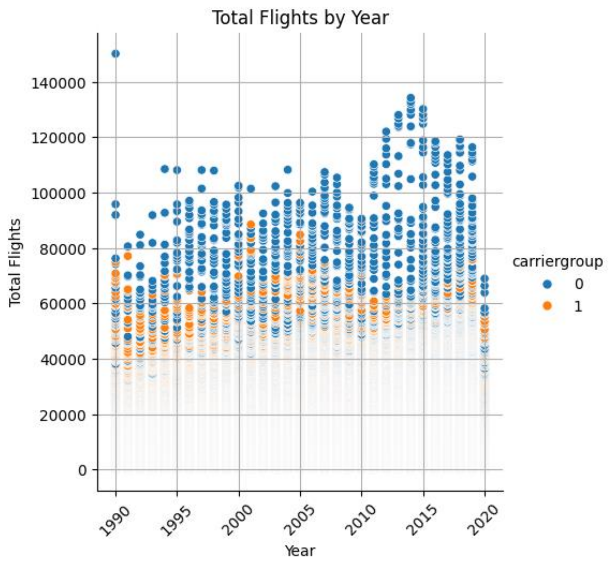
```
sns.countplot(x='carriergroup',data=full_data, color="orange")
plt.title('Distrubtion Of CarrierGroup',fontsize=20)
plt.grid(True)
plt.xticks(rotation=45)
plt.show()
```



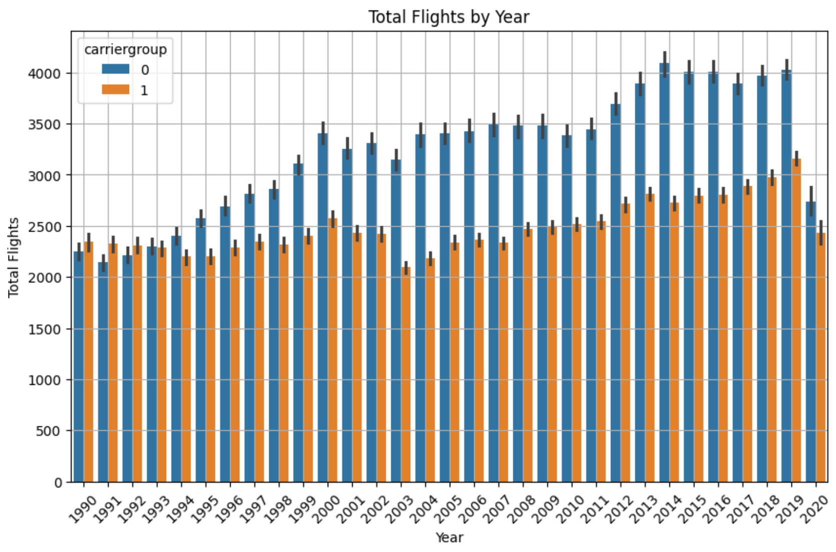
```
plt.figure(figsize=(10, 6))
sns.relplot(x='Year', y='Total', data=full_data ,  color='deepskyblue')
plt.xlabel('Year')
plt.ylabel('Total Flights')
plt.title('Total Flights by Year')
plt.xticks(rotation=45)
plt.show()
```



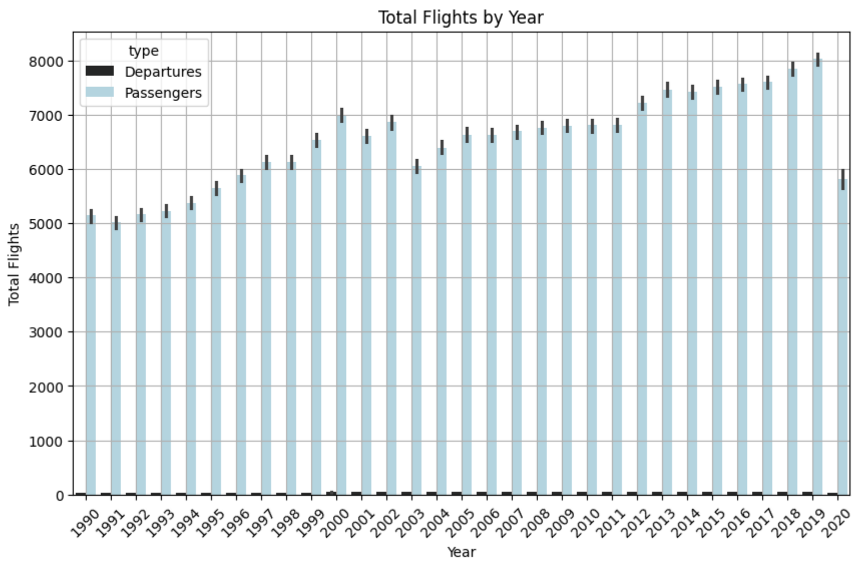
```
plt.figure(figsize=(10, 6))
sns.relplot(x='Year', y='Total',hue='carriergroup',data=full_data, color='green')
plt.xlabel('Year')
plt.ylabel('Total Flights')
plt.title('Total Flights by Year')
plt.grid(True)
plt.xticks(rotation=45)
plt.show()
```



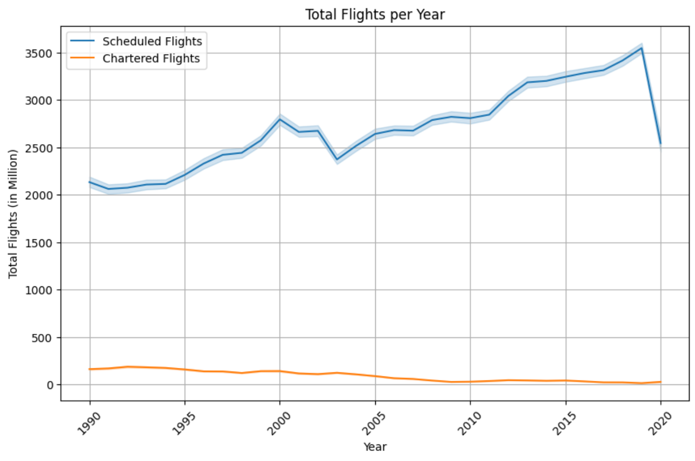
```
plt.figure(figsize=(10, 6))
sns.barplot(x='Year', y='Total',hue='carriergroup',data=full_data)
plt.xlabel('Year')
plt.ylabel('Total Flights')
plt.title('Total Flights by Year')
plt.grid(True)
plt.xticks(rotation=45)
plt.show()
```



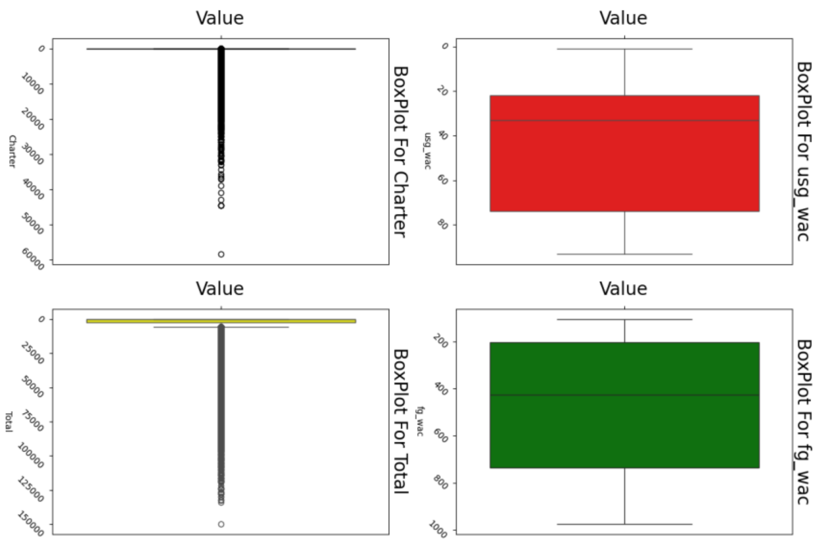
```
plt.figure(figsize=(10, 6))
sns.barplot(x='Year', y='Total',hue='type',data=full_data, color='lightblue' )
plt.xlabel('Year')
plt.ylabel('Total Flights')
plt.title('Total Flights by Year')
plt.xticks(rotation=45)
plt.grid(True)
plt.show()
```



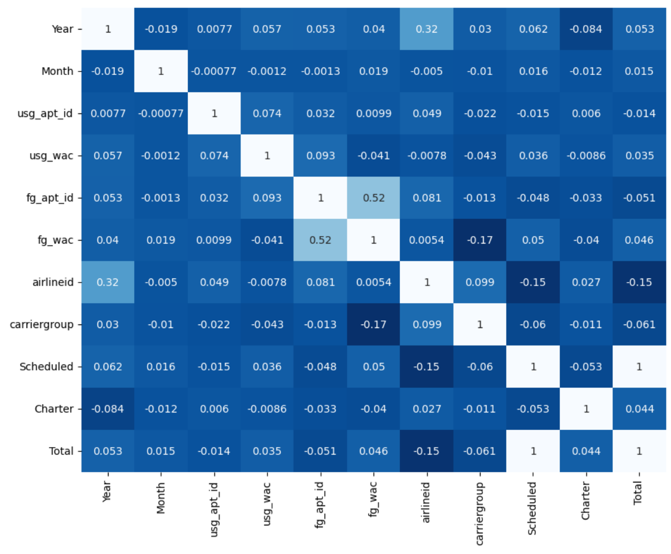
```
plt.figure(figsize=(10, 6))
sns.lineplot(x='Year', y='Scheduled', data=full_data, label='Scheduled Flights')
sns.lineplot(x='Year', y='Charter', data=full_data, label='Chartered Flights')
sns.color_palette("flare", as_cmap=True)
plt.title('Total Flights per Year')
plt.xlabel('Year')
plt.ylabel('Total Flights (in Million)')
plt.xticks(rotation=45)
plt.grid(True)
plt.legend()
plt.show()
```



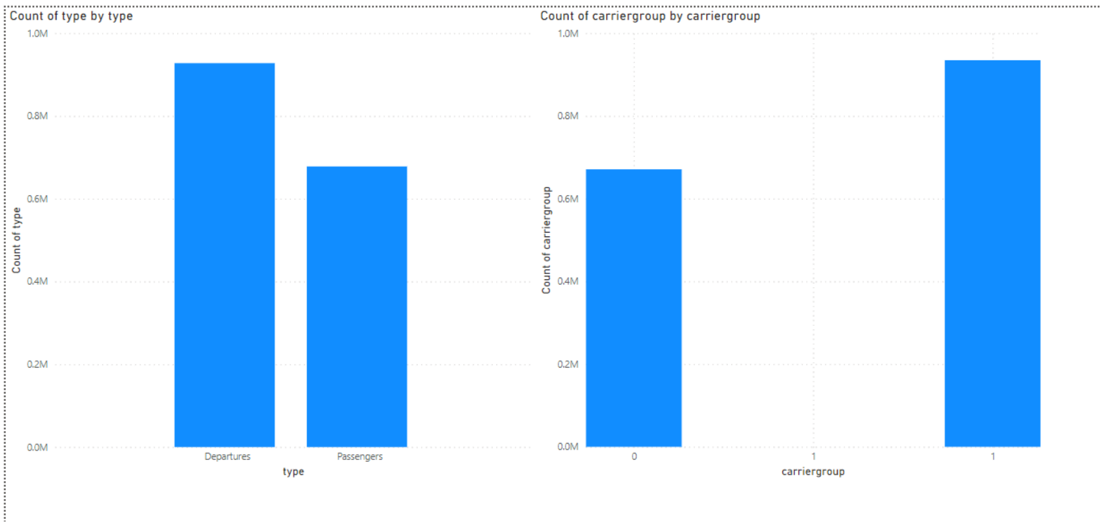
```
plt.figure(figsize=(10,15))
x=1
for i,j in zip(['usg_wac','fg_wac','Charter','Total'],['red','green','black','yellow']):
  plt.subplot(2,2,x)
  sns.boxplot(x=i,data=full_data,color=j)
  plt.ylabel('Value',fontsize=20)
  plt.title(f'BoxPlot For {i}',fontsize=20)
  plt.xticks(rotation=45)
  x+=1
```

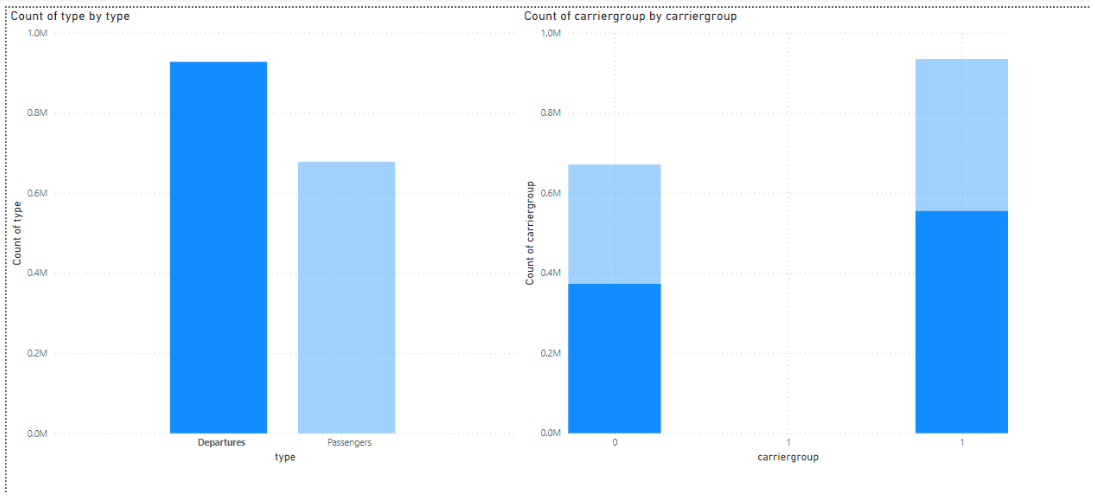


```
numerical_data = full_data.select_dtypes(include=['number'])
correlation_matrix = numerical_data.corr()
plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, cmap='Blues_r', annot=True, cbar=False)
```

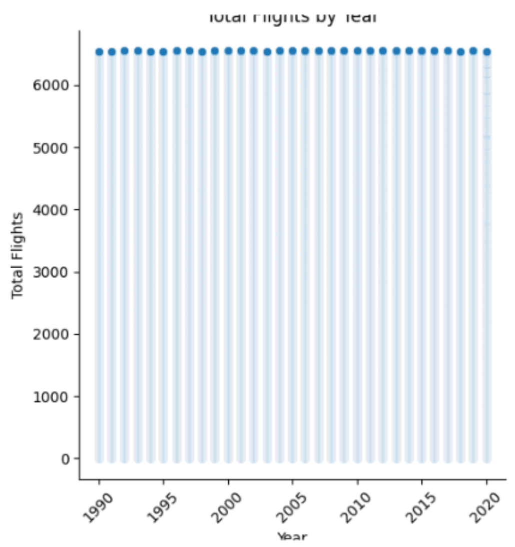


2. Using POWERBI

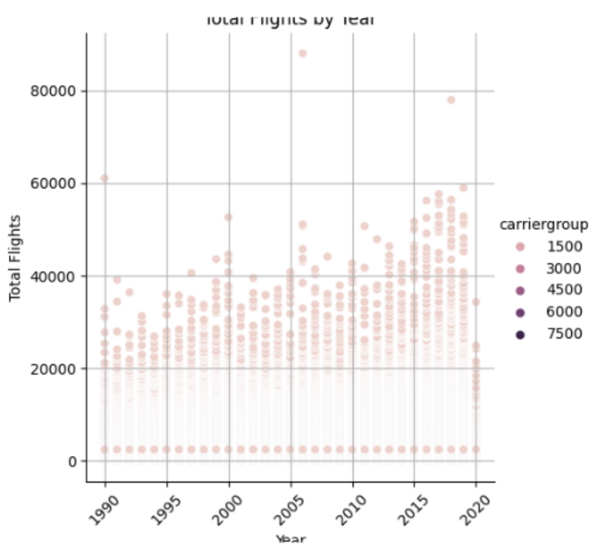




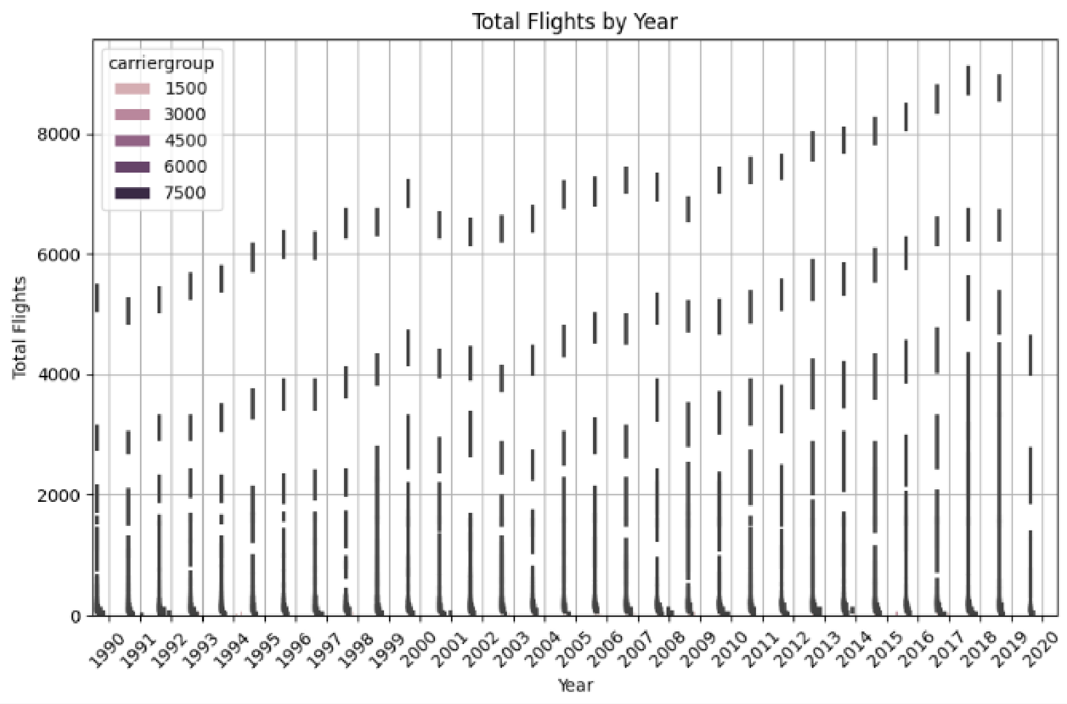
Total and Year



Count of carriergroup, Year and Total

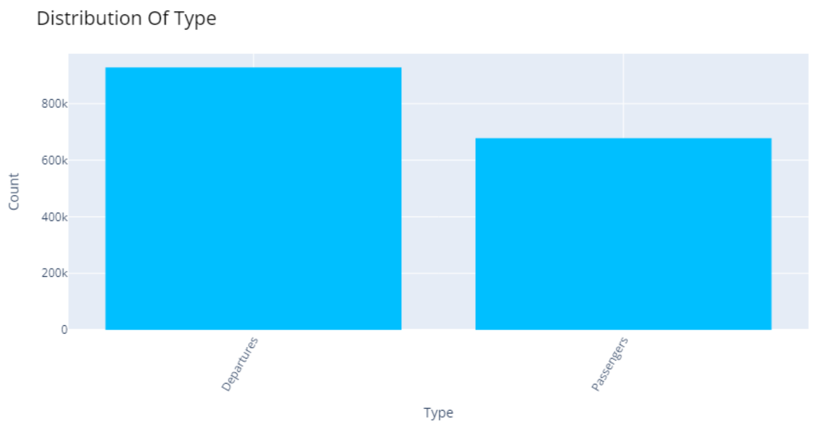


Count of carriergroup, Total and Year

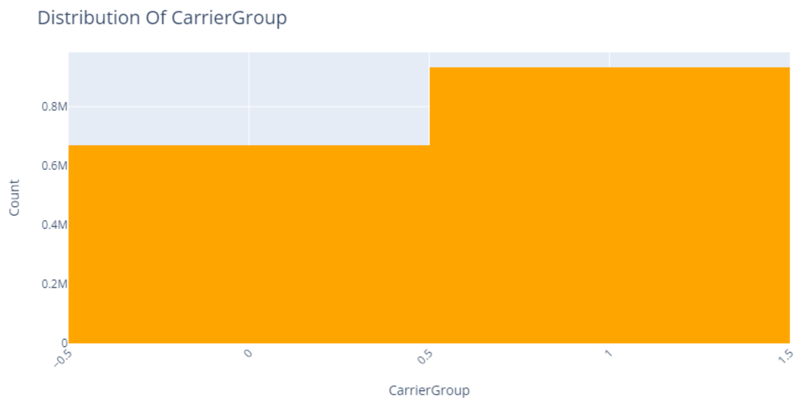


3. Using Plotly (Python)

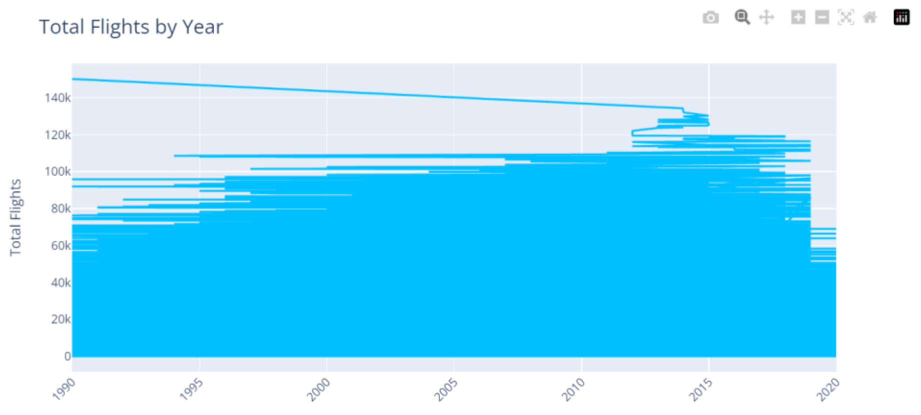
```
fig = px.histogram(full_data, x='type', color_discrete_sequence=['deepskyblue'])
fig.update_layout(
    title='Distribution Of Type',
    title_font_size=20,
    title_font_color='black',
    xaxis_title='Type',
    yaxis_title='Count',
    xaxis_tickangle=-60,
    xaxis_showgrid=True,
    yaxis_showgrid=True
)
fig.show()
```



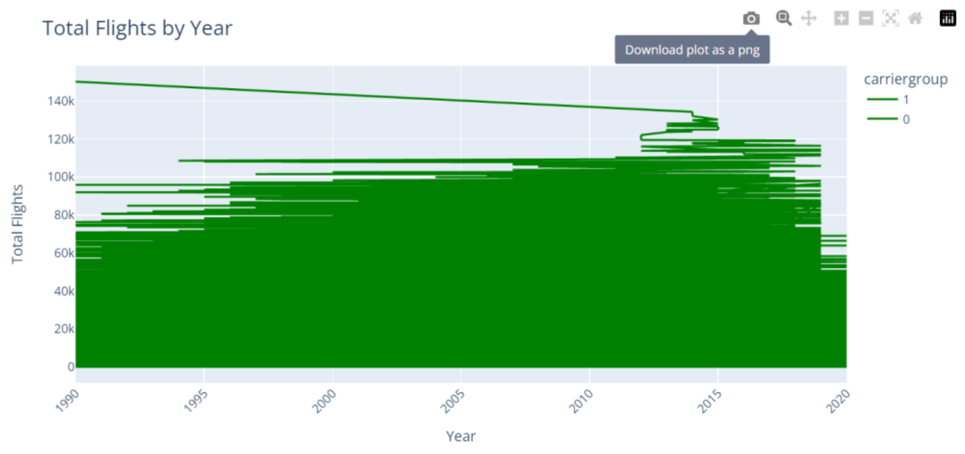
```
fig = px.histogram(full_data, x='carriergroup', color_discrete_sequence=['orange'])
fig.update_layout(
    title='Distribution Of CarrierGroup',
    title_font_size=20,
    xaxis_title='CarrierGroup',
    yaxis_title='Count',
    xaxis_tickangle=-45,
    xaxis_showgrid=True,
    yaxis_showgrid=True
)
fig.show()
```



```
fig = px.line(full_data, x='Year', y='Total', line_shape='linear', color_discrete_sequence=['deepskyblue'])
fig.update_layout(
    title='Total Flights by Year',
    title_font_size=20,
    xaxis_title='Year',
    yaxis_title='Total Flights',
    xaxis_tickangle=-45
)
fig.show()
```



```
fig = px.line(full_data, x='Year', y='Total', color='carriergroup', line_shape='linear',
color_discrete_sequence=['green'])
fig.update_layout(
    title='Total Flights by Year',
    title_font_size=20,
    xaxis_title='Year',
    yaxis_title='Total Flights',
    xaxis_tickangle=-45,
    xaxis_showgrid=True,
    yaxis_showgrid=True
)
fig.show()
```

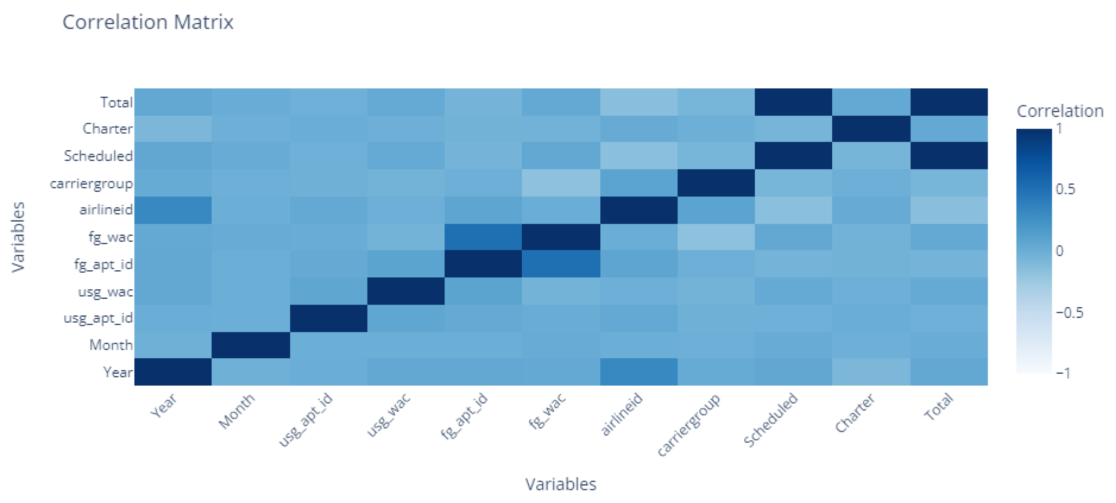


```
# Compute the correlation matrix
numerical_data = full_data.select_dtypes(include=['number'])
correlation_matrix = numerical_data.corr()

# Create the heatmap
fig = go.Figure(data=go.Heatmap(
   z=correlation_matrix.values,
   x=correlation_matrix.columns,
   y=correlation_matrix.columns,
   colorscale='Blues',
   colorbar_title='Correlation',
   zmid=0
))

# Update layout
fig.update_layout(
   title='Correlation Matrix',
   xaxis_title='Variables',
   yaxis_title='Variables',
   xaxis_nticks=len(correlation_matrix.columns),
   yaxis_nticks=len(correlation_matrix.columns),
   xaxis_tickangle=-45
)

fig.show()
```


Correlation Matrix
Total
Charter
Scheduled
carriergroup
airlineid
fg_wac
fg_apt_id
usg_wac
usg_apt_id
Month
Year
Variables
Year
Month
usg_apt_id
usg_wac
fg_apt_id
fg_wac
airlineid
carriergroup
Scheduled
Charter
Total
Variables
Correlation