2. Data Visualization

Generate statistical summaries of the dataset.

Create visualizations using Matplotlib and Seaborn (e.g., histograms, scatter plots, heatmaps).

generate statistical summaries of a dataset and **create visualizations** using **Matplotlib** and **Seaborn** in Google Colab.

1. Generate Statistical Summaries of the Dataset

You can generate statistical summaries such as measures of central tendency, dispersion, and distribution of the dataset using Pandas.

```
import pandas as pd

# Load dataset (replace with your file path or URL)
url = 'https://raw.githubusercontent.com/openai/data/master/titanic.csv'
data = pd.read_csv(url)

# Generate statistical summary of the dataset
summary = data.describe()

# Display the summary
print(summary)
```

The `describe()` function in Pandas will give you:

- **Count**: The number of non-null values.
- **Mean**: The average value of numerical columns.
- **Standard deviation**: Measures the spread of the data.
- **Min, 25%, 50%, 75%, Max**: The min, quartiles, and max values.

If you want to get a summary of **categorical** variables as well, you can use:

```
# Summary of categorical columns (mode, unique values, etc.)
print(data.describe(include=['object']))
```

2. Create Visualizations Using Matplotlib and Seaborn

Visualizations using Matplotlib

Matplotlib is a powerful plotting library. Here’s how you can create basic visualizations like histograms, bar charts, and box plots.

Histogram

A histogram helps visualize the distribution of a numeric feature.

```
import matplotlib.pyplot as plt

# Plot histogram for 'Age' column
plt.hist(data['Age'].dropna(), bins=20, edgecolor='black', color='skyblue')
plt.title('Age Distribution')
plt.xlabel('Age')
plt.ylabel('Frequency')
plt.show()
```

Bar Chart

Bar charts are useful for categorical data, like the count of passengers by class or embarked location.

```
# Bar chart for 'Pclass' (Passenger class)
plt.figure(figsize=(6, 4))
data['Pclass'].value_counts().plot(kind='bar', color='lightcoral')
plt.title('Passenger Class Distribution')
plt.xlabel('Class')
plt.ylabel('Count')
plt.xticks(rotation=0)
plt.show()
```

Box Plot

A box plot is useful for visualizing the distribution and detecting outliers in a dataset.

```
# Box plot for 'Age' to visualize outliers and distribution
plt.figure(figsize=(6, 4))
plt.boxplot(data['Age'].dropna(), patch_artist=True,
boxprops=dict(facecolor='lightgreen'))
plt.title('Box Plot of Age')
plt.ylabel('Age')
plt.show()
```

Visualizations using Seaborn

Seaborn is built on top of Matplotlib and provides easier syntax and more aesthetically pleasing plots. It also offers additional functionalities for categorical data visualizations.

Pair Plot

A pair plot shows the relationships between multiple features in the dataset.

```
import seaborn as sns

# Pairplot to visualize relationships between numerical features
sns.pairplot(data[['Age', 'Fare', 'SibSp', 'Parch']].dropna())
plt.show()
```

Correlation Heatmap

A heatmap of correlations shows how features are correlated with each other.

```
# Correlation heatmap
correlation_matrix = data[['Age', 'Fare', 'SibSp', 'Parch']].corr()
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', cbar=True)
plt.title('Correlation Heatmap')
plt.show()
```

Count Plot

A count plot is useful for visualizing the distribution of a categorical variable.

```
# Count plot for 'Survived' (how many survived vs how many did not)
sns.countplot(x='Survived', data=data, palette='Set2')
plt.title('Survival Count')
plt.xlabel('Survived')
plt.ylabel('Count')
plt.show()
```

Violin Plot

A violin plot combines aspects of box plot and density plot, and is useful for visualizing the distribution of a continuous variable across categories.

```
# Violin plot for 'Age' across 'Pclass'
sns.violinplot(x='Pclass', y='Age', data=data, palette='muted')
plt.title('Age Distribution by Passenger Class')
plt.show()
```

Full Example of Statistical Summary and Visualization Code for Google Colab

```
# Importing necessary libraries
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# Load dataset
url = 'https://raw.githubusercontent.com/openai/data/master/titanic.csv'
data = pd.read_csv(url)

# 1. Generate Statistical Summary
print("Statistical Summary:")
print(data.describe())

# Summary of categorical columns
print("\nCategorical Summary:")
print(data.describe(include=['object']))
```

```
# 2. Data Visualizations using Matplotlib
# Histogram for 'Age'
plt.hist(data['Age'].dropna(), bins=20, edgecolor='black', color='skyblue')
plt.title('Age Distribution')
plt.xlabel('Age')
plt.ylabel('Frequency')
plt.show()

# Bar chart for 'Pclass'
plt.figure(figsize=(6, 4))
data['Pclass'].value_counts().plot(kind='bar', color='lightcoral')
plt.title('Passenger Class Distribution')
plt.xlabel('Class')
plt.ylabel('Count')
plt.xticks(rotation=0)
plt.show()

# Box plot for 'Age'
plt.figure(figsize=(6, 4))
plt.boxplot(data['Age'].dropna(), patch_artist=True,
boxprops=dict(facecolor='lightgreen'))
plt.title('Box Plot of Age')
plt.ylabel('Age')
plt.show()

# 3. Data Visualizations using Seaborn
# Pair plot to visualize relationships between numerical features
sns.pairplot(data[['Age', 'Fare', 'SibSp', 'Parch']].dropna())
plt.show()

# Correlation heatmap
correlation_matrix = data[['Age', 'Fare', 'SibSp', 'Parch']].corr()
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', cbar=True)
plt.title('Correlation Heatmap')
plt.show()

# Count plot for 'Survived'
sns.countplot(x='Survived', data=data, palette='Set2')
plt.title('Survival Count')
plt.xlabel('Survived')
plt.ylabel('Count')
plt.show()

# Violin plot for 'Age' by 'Pclass'
sns.violinplot(x='Pclass', y='Age', data=data, palette='muted')
plt.title('Age Distribution by Passenger Class')
plt.show()
```

Conclusion:

This code demonstrates how to:

1. Generate **statistical summaries** of a dataset using Pandas.
2. Create a variety of **visualizations** using **Matplotlib** and **Seaborn** to explore the dataset, including histograms, bar charts, box plots, pair plots, heatmaps, count plots, and violin plots.