Exploratory Data Analysis and Visualization Plan

# 1. Introduction

Exploratory Data Analysis (EDA) is the process of analyzing datasets to summarize their main characteristics using statistical and visual methods. EDA helps identify patterns, spot anomalies, test hypotheses, and check assumptions with the help of visualizations and statistical tools.

# 2. Objectives of EDA

- Understand the structure and distribution of the data  
- Detect missing values and outliers  
- Identify relationships between variables  
- Generate hypotheses for further analysis  
- Support data cleaning and preprocessing

# 3. Key Questions Driving EDA

- What are the types and distributions of features in the dataset?  
- Are there missing or duplicate records?  
- What are the central tendencies and dispersions (mean, median, std)?  
- Are there any outliers or anomalies?  
- What correlations exist between numeric variables?  
- Are there categorical distributions worth exploring?  
- Are there observable trends over time (for time-series data)?

# 4. Statistical Summarization Methods

- Shape and Structure: df.shape, df.info()  
- Summary Statistics: df.describe()  
- Missing Values: df.isnull().sum(), df.isna().mean()  
- Value Counts: df['column'].value\_counts()  
- Skewness/Kurtosis: df.skew(), df.kurt()  
- Correlation Matrix: df.corr()

# 5. Visualization Techniques

a. Univariate Analysis

* - Histogram (for distribution): sns.histplot(), plt.hist()  
  - Boxplot (for outliers): sns.boxplot()  
  - Bar plot (for categories): sns.countplot(), plt.bar()

b. Bivariate Analysis

* - Scatter Plot (relationships): sns.scatterplot()  
  - Pairplot (multi-variable analysis): sns.pairplot()  
  - Heatmap (correlation): sns.heatmap(corr\_matrix, annot=True)

c. Multivariate Analysis

* - Violin Plot: sns.violinplot()  
  - FacetGrid: sns.FacetGrid(data).map()  
  - Line Plot (for trends): sns.lineplot(), plt.plot()

# 6. Visualization Considerations

- Color Palette: Use sns.color\_palette() to ensure visual accessibility (e.g., colorblind-friendly palettes)  
- Clarity and Simplicity: Avoid clutter, label axes clearly, and use legends where needed  
- Interactivity (optional): Use Plotly or Altair for interactive dashboards

# 7. Example Code Structure

```python  
import pandas as pd  
import matplotlib.pyplot as plt  
import seaborn as sns  
  
df = pd.read\_csv('data.csv')  
print(df.describe())  
  
plt.figure(figsize=(10, 6))  
sns.heatmap(df.corr(), annot=True, cmap='coolwarm')  
plt.title('Correlation Heatmap')  
plt.show()  
  
sns.histplot(df['age'], kde=True)  
plt.title('Age Distribution')  
plt.show()  
```

# 8. Mock Analysis Plan (Hypothetical Example)

Dataset: Customer Purchase Dataset

- Identify top product categories (Bar Plot)  
- Analyze customer age distribution (Histogram)  
- Correlate income with purchase amount (Scatter Plot)  
- Explore purchase trends over months (Line Plot)  
- Detect anomalies in spending (Boxplot)

Possible Insights:  
- Younger users buy more during holiday seasons  
- Income correlates with high-value purchases  
- Outliers in spending highlight loyal or fraudulent users

# 9. Timeline for EDA Plan Execution

|  |  |
| --- | --- |
| Week | Activity |
| 1 | Load data, clean nulls, and summarize stats |
| 2 | Perform univariate and bivariate analysis |
| 3 | Create visualizations and identify patterns |
| 4 | Finalize EDA report and share insights |

# 10. Conclusion

This EDA and visualization plan provides a structured approach to understand the dataset and uncover hidden patterns. It uses core Python libraries like Matplotlib and Seaborn, and considers interactive options for advanced insights.