

Experiment-2

Aim:

Data Visualization/ Exploratory data Analysis using Matplotlib and Seaborn.

1. Create bar graph, contingency table using any 2 features.
2. Plot Scatter plot, box plot, Heatmap using seaborn.
3. Create histogram and normalized Histogram.
4. Describe what this graph and table indicates.
5. Handle outlier using box plot and Inter quartile range.

Introduction:

Exploratory Data Analysis (EDA), introduced by John Tukey in the 1970s, is the first step in analyzing datasets to summarize their key characteristics using statistical and visual techniques. It helps understand data patterns, detect anomalies, and prepare the data for machine learning models.

Why Perform EDA?

EDA is essential for:

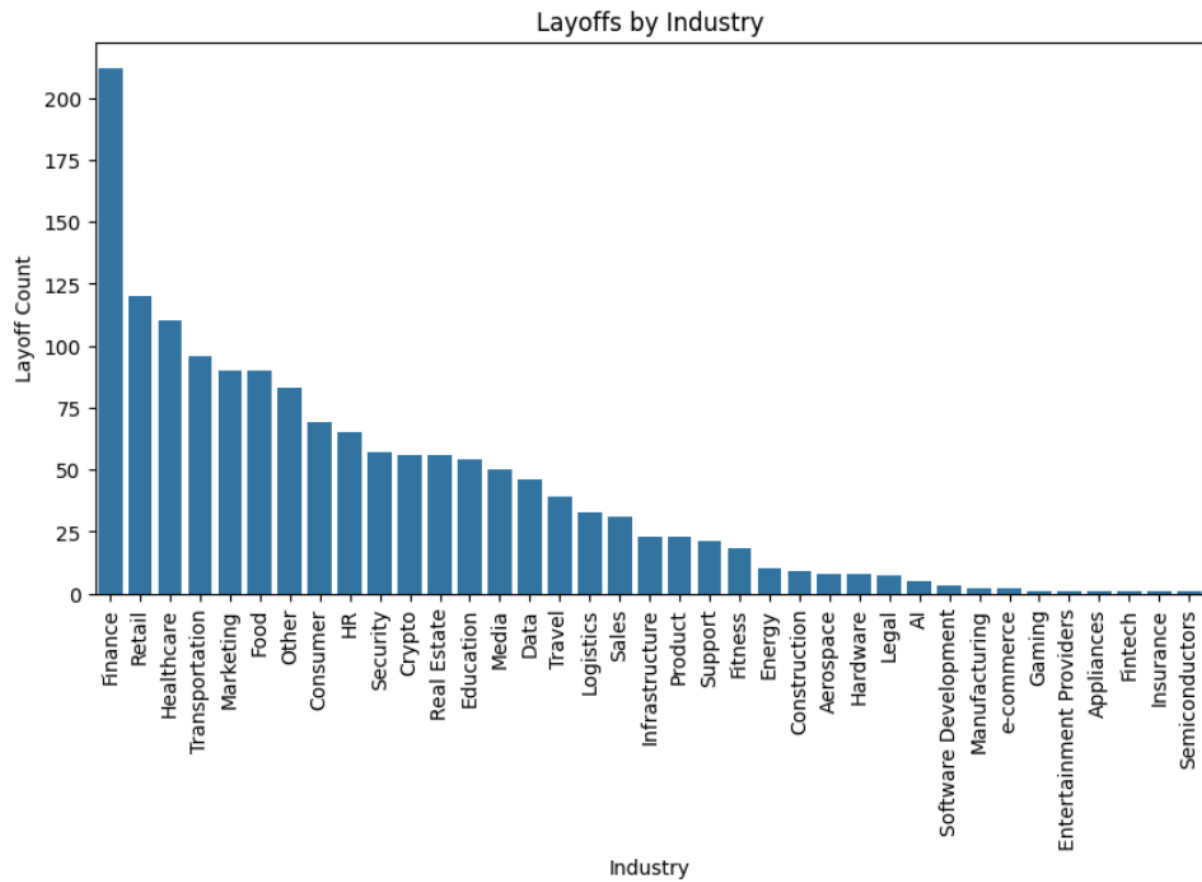
- Identifying key features and trends in the data.
- Detecting correlations between variables.
- Assessing data quality and handling missing values.
- Determining the need for data preprocessing.
- Communicating insights effectively using visual tools.

Common EDA Techniques:

- Histograms and frequency distributions to analyze data distribution.
- Box plots to identify outliers and data spread.
- Scatter plots to observe relationships between variables.
- Heatmaps to visualize correlations between features.
- Bar charts and pie charts for categorical data analysis

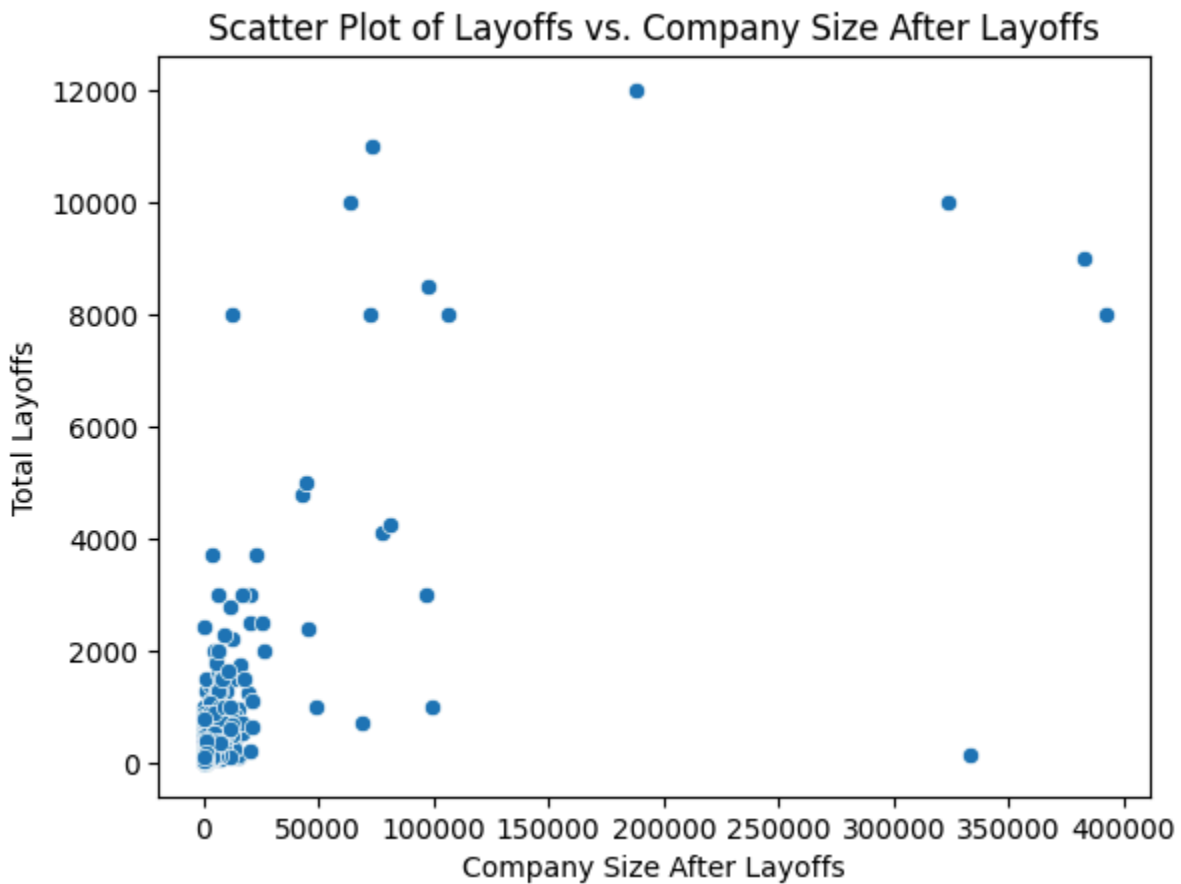
1)Bar Graph

```
plt.figure(figsize=(10, 5))
sns.barplot(x=df['Industry'].value_counts().index, y=df['Industry'].value_counts().values)
plt.xticks(rotation=90)
plt.xlabel('Industry')
plt.ylabel('Layoff Count')
plt.title('Layoffs by Industry')
plt.show()
```



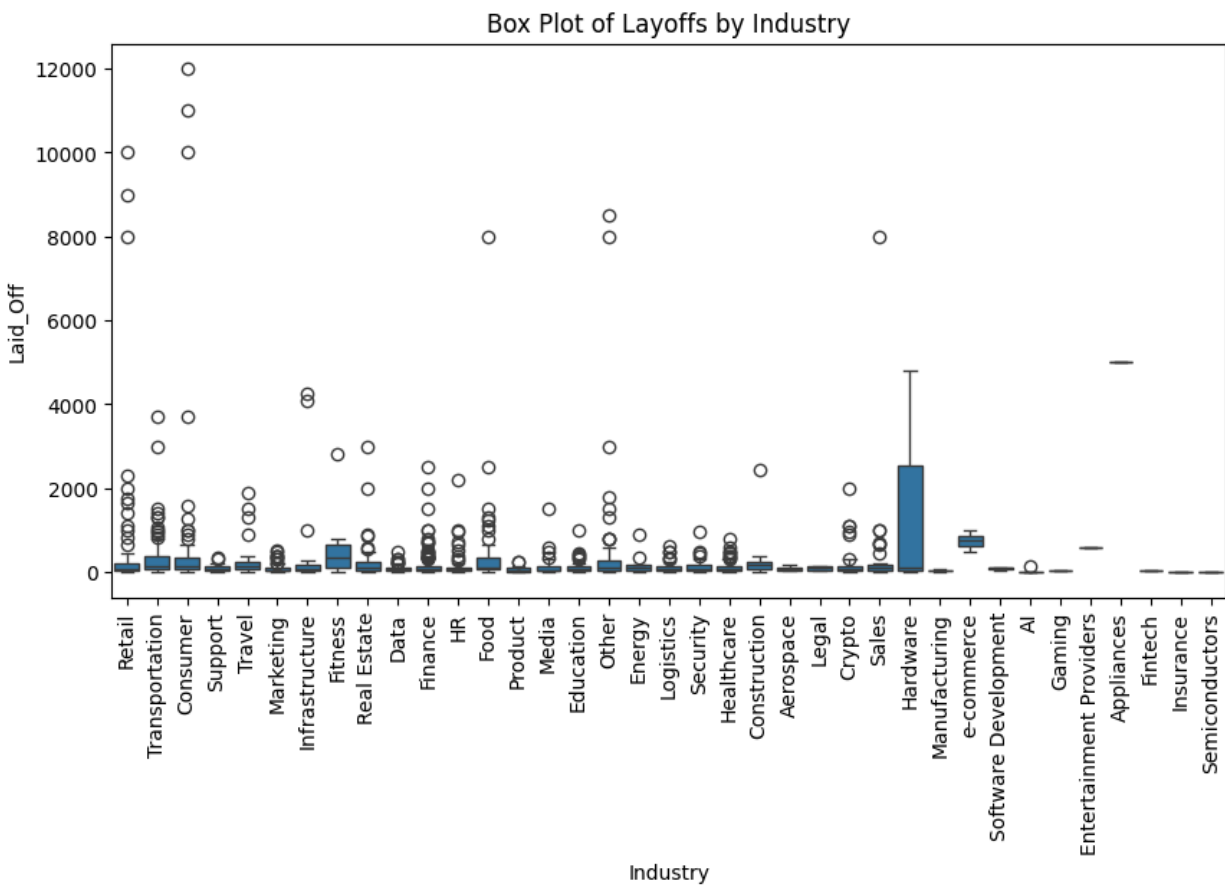
2)Scatter plot

```
▶ sns.scatterplot(x=df['Company_Size_after_layoffs'], y=df['Laid_Off'])  
plt.xlabel('Company Size After Layoffs')  
plt.ylabel('Total Layoffs')  
plt.title('Scatter Plot of Layoffs vs. Company Size After Layoffs')  
plt.show()
```



3)Box Plot

```
[ ] plt.figure(figsize=(10, 5))
sns.boxplot(x='Industry', y='Laid_Off', data=df)
plt.xticks(rotation=90)
plt.title('Box Plot of Layoffs by Industry')
plt.show()
```



4)Heat Map

```
numeric_df = df.select_dtypes(include=['number'])

plt.figure(figsize=(8, 6))
sns.heatmap(numeric_df.corr(), annot=True, cmap='coolwarm', fmt=".2f")
plt.title("Heatmap of Feature Correlations")
plt.show()
```

