stats exp 3

Experiment Number: 3

AIM: Load an open-source dataset into a pandas DataFrame and perform Exploratory Data Analysis (EDA).

Software Used: Python with Pandas, NumPy, SciPy, Matplotlib, and Seaborn libraries in a Jupyter Notebook or similar environment.

Theory:

This experiment dives into Exploratory Data Analysis (EDA), a critical step in understanding and extracting insights from data. It emphasizes using Python libraries like Pandas for data manipulation and Matplotlib/Seaborn for visualization.

- **Pandas:** A powerful library for data manipulation and analysis, providing DataFrames for efficient handling of tabular data.
- **EDA:** An iterative process involving:
  - **Data Cleaning:** Handling missing values, outliers, and inconsistencies.
  - **Univariate Analysis:** Examining individual variables for distributions, central tendencies, and dispersion.
  - **Bivariate/Multivariate Analysis:** Exploring relationships between two or more variables.
  - **Data Visualization:** Creating charts and graphs to gain insights and communicate findings.
- **Matplotlib:** A versatile plotting library for creating static, interactive, and animated visualizations.
- **Seaborn:** Built on Matplotlib, Seaborn provides a high-level interface for creating statistically informative and visually appealing plots.

Dataset Information:

You need to select an open-source dataset. Refer to the sources mentioned in Experiment 2 (Kaggle, UCI ML Repository, Google Dataset Search) to find a suitable dataset.

Example Dataset (Choose your own):

- **Dataset Name:** Titanic Dataset
- **Source:** Kaggle
- **Description:** This dataset contains information about passengers on the Titanic, including survival status, age, gender, ticket class, fare, etc. It's often used for classification tasks (predicting survival).

Code (Illustrative - adapt to your chosen dataset):

Python

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Load the dataset (assuming it's in a CSV file)
data = pd.read_csv('titanic.csv')

# Display basic information
print(data.head())
print(data.info())
print(data.describe())

# Separate numerical and categorical variables
cat_cols = data.select_dtypes(include=['object']).columns
num_cols = data.select_dtypes(include=np.number).columns.tolist()
print("Categorical Variables:", cat_cols)
print("Numerical Variables:", num_cols)

# Univariate analysis (histograms and box plots for numerical variables)
for col in num_cols:
    print(col)
    print('Skew :', round(data[col].skew(), 2))
    plt.figure(figsize=(15, 4))
    plt.subplot(1, 2, 1)
    data[col].hist(grid=False)
    plt.ylabel('count')
    plt.subplot(1, 2, 2)
    sns.boxplot(x=data[col])
    plt.show()

# Univariate analysis (count plots for categorical variables)
for col in cat_cols:
    plt.figure(figsize=(10, 4))
    sns.countplot(x=data[col])
    plt.show()

# Bivariate analysis (example: correlation heatmap for numerical variables)
plt.figure(figsize=(10, 8))
```

```
sns.heatmap(data.corr(), annot=True, cmap='coolwarm')
plt.show()
```

Output (Will vary based on your dataset):

(Output will include the first few rows of the data, column information, summary statistics, histograms, box plots, count plots, and a correlation heatmap.)

Result:

The code performs a comprehensive EDA, including loading the data, displaying basic information, separating numerical and categorical variables, and conducting univariate analysis with visualizations. It also includes an example of bivariate analysis with a correlation heatmap.

Learning Outcomes:

- Loaded and explored a real-world dataset using Pandas.
- Performed univariate analysis to understand the distribution and characteristics of individual variables.
- Created visualizations (histograms, box plots, count plots) to gain insights from the data.
- Conducted bivariate analysis (correlation heatmap) to explore relationships between variables.
- Gained a deeper understanding of EDA techniques and their importance in data analysis.
- Enhanced skills in using Pandas, Matplotlib, and Seaborn for data manipulation and visualization.