Bansilal Ramnath Agarwal Charitable Trust's

VISHWAKARMA INSTITUTE OF INFORMATION TECHNOLOGY, PUNE-48

Department of Information Technology

ITUA40201: DATA SCIENCE AND ANALYTICS

Assignment-1

Shreyas Shripad Kulkarni

BTech A Division

Roll No.: 431048

PRN: 22010443

AIM: Data Exploration and Visualization Exercise

OBJECTIVE: Perform exploratory data analysis and create visualizations to gain insights from a given dataset.

TASKS:

1) Load a dataset (e.g., customer sales data, stock market data).
2) Perform data cleaning and pre-processing.
3) Conduct descriptive statistical analysis.
4) Create visualizations (e.g., bar charts, scatter plots, box plots) to explore relationships and patterns in the data.
5) Interpret the findings and present the insights.

THEORY:

- **DATA PREPROCESSING**: Data preprocessing is a fundamental and critical step in the data preparation phase of any data analysis. It involves a series of operations and techniques to clean, transform, and organize raw data into a format that is suitable for analysis or modelling. Data preprocessing aims to improve the quality, consistency, and usability of data, making it ready for further exploration and application.

Key Aspects of Data Preprocessing:

- ⌘ Data Cleaning
- ⌘ Data Transformation
- ⌘ Data Reduction
- ⌘ Data Integration
- ⌘ Feature Selection
- ⌘ Handling Imbalanced Data

Φ **DATA CLEANING**: Data cleaning is the process of fixing or removing incorrect, corrupted, incorrectly formatted, duplicate, or incomplete data within a dataset. Data cleaning is the process that removes data that does not belong in your dataset.

There are several other data cleaning techniques:

1. Handling Missing Values:
   - ∆ Drop rows or columns with missing values: You can use the `dropna()` function to remove rows or columns with missing values.
   - ∆ Fill missing values with a specific value: You can use the `fillna()` function to replace missing values with a specified value.
   - ∆ Fill missing values with mean, median, or mode: You can use the `fillna()` function with the mean, median, or mode of the column to replace missing values.
2. Removing Duplicates:
   - ∆ Use the `drop_duplicates()` function to remove duplicate rows from the dataset.
3. Handling Outliers:
   - ∆ Identify outliers using statistical methods such as z-score or IQR (Interquartile Range).
   - ∆ Remove outliers by filtering the dataset based on the identified outliers.

- **DESCRIPTIVE STATISTICAL ANALYSIS**: Descriptive statistical analysis is a branch of statistics that focuses on summarizing and presenting data in a meaningful and interpretable way. Descriptive statistics are fundamental in data analysis as they provide a clear and concise snapshot of the data, making it easier to understand and interpret.

IMPLEMENTATION:

In this assignment we will be using the standard dataset (Titanic).

Step 1: Load Dataset.

431048_DSA_Assg1_Shreyas_

```
[2]: import numpy as np
     import pandas as pd
```

```
[4]: dataset = pd.read_csv('titanic.csv')
     dataset.head()
```

[4]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S

```
[6]: dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   PassengerId  891 non-null    int64
 1   Survived     891 non-null    int64
 2   Pclass       891 non-null    int64
 3   Name         891 non-null    object
 4   Sex          891 non-null    object
 5   Age          714 non-null    float64
 6   SibSp        891 non-null    int64
 7   Parch        891 non-null    int64
 8   Ticket       891 non-null    object
 9   Fare         891 non-null    float64
 10  Cabin        204 non-null    object
 11  Embarked     889 non-null    object
dtypes: float64(2), int64(5), object(5)
memory usage: 83.7+ KB
```

Step 2: Data Preprocessing & Data Cleaning

```
[8]: dataset.isna().sum()
     #This shows the number of NULL values in the Dataset
```

```
[8]: PassengerId      0
     Survived         0
     Pclass           0
     Name             0
     Sex              0
     Age            177
     SibSp            0
     Parch            0
     Ticket           0
     Fare             0
     Cabin          687
     Embarked         2
     dtype: int64
```

- Drop rows or columns with missing values: You can use the `dropna()` function to remove rows or columns with missing values.

```
[26]: #Method 1 to Clean Dataset
      #1.1 Drop rows or columns with missing values
      dataset.dropna(inplace=True)
      dataset.dropna(axis=1, inplace=True)
      dataset.isna().sum()
```

```
[26]: PassengerId    0
      Survived       0
      Pclass         0
      Name           0
      Sex            0
      Age            0
      SibSp          0
      Parch          0
      Ticket         0
      Fare           0
      Cabin          0
      Embarked       0
      dtype: int64
```

∆ Fill missing values with a specific value: You can use the `fillna()` function to replace missing values with a specified value.

```
[28]: #1.2 Fill missing values with mean, median, or mode
dataset['Fare'].fillna(dataset['Fare'].mean(), inplace=True)
dataset['Age'].fillna(dataset['Age'].median(), inplace=True)
dataset['Embarked'].fillna(dataset['Embarked'].mode()[0], inplace=True)
dataset['Cabin'].fillna(dataset['Cabin'].mode()[0], inplace=True)

dataset.isna().sum()
```

```
[28]: PassengerId    0
Survived       0
Pclass         0
Name           0
Sex            0
Age            0
SibSp          0
Parch          0
Ticket         0
Fare           0
Cabin          0
Embarked       0
dtype: int64
```

∆ Fill missing values with mean, median, or mode: You can use the `fillna()` function with the mean, median, or mode of the column to replace missing values.

```
[110]: #1.3 Fill the NULL/Missing Values with a specific value
dataset.Cabin = dataset.Cabin.fillna("0")
print(dataset.isnull().sum())
```

```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age            177
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin            0
Embarked         2
dtype: int64
```

Step 3: Descriptive statistical analysis

```
[234]: #Descriptive Statistical Analysis
       #1.1 Summary Statistics for Numeric Columns
       numeric_summary = dataset.describe()
       print(numeric_summary)
```

```
       PassengerId    Survived      Pclass         Age       SibSp  \
count   891.000000  891.000000  891.000000  714.000000  891.000000
mean    446.000000    0.383838    2.308642   29.699118    0.523008
std     257.353842    0.486592    0.836071   14.526497    1.102743
min       1.000000    0.000000    1.000000    0.420000    0.000000
25%     223.500000    0.000000    2.000000   20.125000    0.000000
50%     446.000000    0.000000    3.000000   28.000000    0.000000
75%     668.500000    1.000000    3.000000   38.000000    1.000000
max     891.000000    1.000000    3.000000   80.000000    8.000000

            Parch        Fare
count  891.000000  891.000000
mean     0.381594   32.204208
std      0.806057   49.693429
min      0.000000    0.000000
25%      0.000000    7.910400
50%      0.000000   14.454200
75%      0.000000   31.000000
max      6.000000  512.329200
```

```
[255]: # 1.2 Summary Statistics for Categorical Columns
       categorical_summary_sex = dataset["Sex"].value_counts()
       categorical_summary_embarked = dataset["Embarked"].value_counts()

       print("Sex Distribution:")
       print(categorical_summary_sex)
       print("\nEmbarked Distribution:")
       print(categorical_summary_embarked)
```

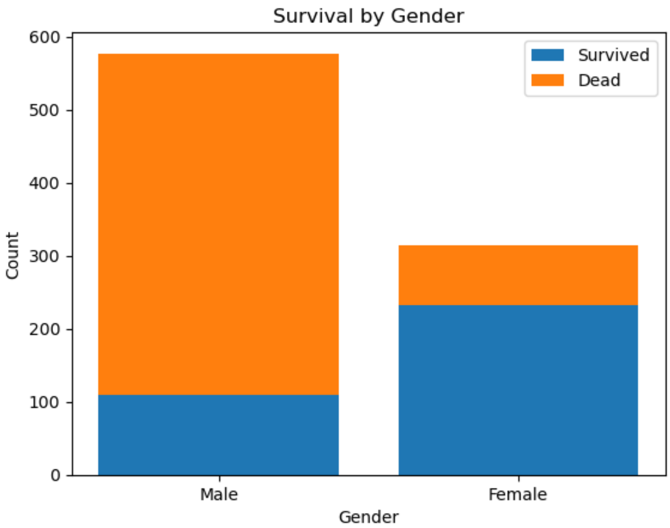
```
Sex Distribution:
male      577
female    314
Name: Sex, dtype: int64

Embarked Distribution:
S    644
C    168
Q     77
Name: Embarked, dtype: int64
```

```
[238]: #1.3 Count the number of missing values
       dataset.isna().sum()
```

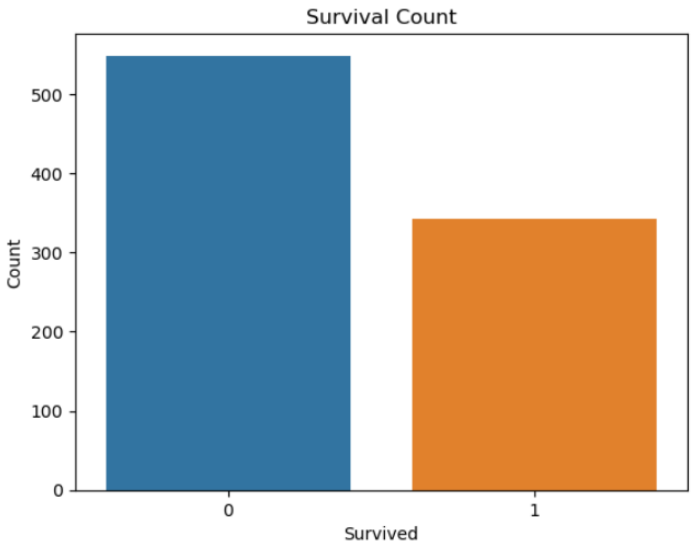
```
[238]: PassengerId      0
       Survived         0
       Pclass           0
       Name             0
       Sex              0
       Age            177
       SibSp            0
       Parch            0
       Ticket           0
       Fare             0
       Cabin          687
       Embarked         2
       dtype: int64
```

Step 4: Create visualizations

[162]:
```
dataset = sns.load_dataset('titanic')
male = dataset[dataset['sex'] == 'male']
female = dataset[dataset['sex'] == 'female']

male_survived = male[male['survived'] == 1].shape[0]
male_dead = male[male['survived'] == 0].shape[0]
female_survived = female[female['survived'] == 1].shape[0]
female_dead = female[female['survived'] == 0].shape[0]

plt.bar(['Male', 'Female'], [male_survived, female_survived], label='Survived')
plt.bar(['Male', 'Female'], [male_dead, female_dead], bottom=[male_survived, female_survived], label='Dead')
plt.title('Survival by Gender')
plt.xlabel('Gender')
plt.ylabel('Count')
plt.legend()
plt.show()
```

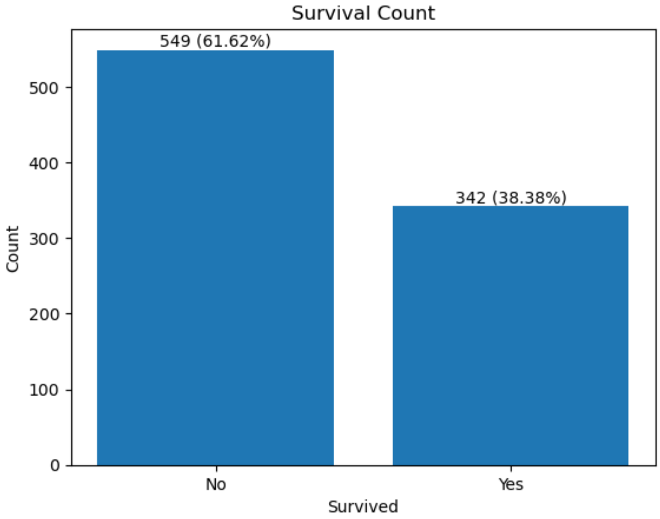


[138]:
```
dataset = sns.load_dataset('titanic')
sns.countplot(x='survived', data=dataset)
plt.xlabel('Survived')
plt.ylabel('Count')
plt.title('Survival Count')
plt.show()
# 0-> Not Survived
# 1-> Survived
```

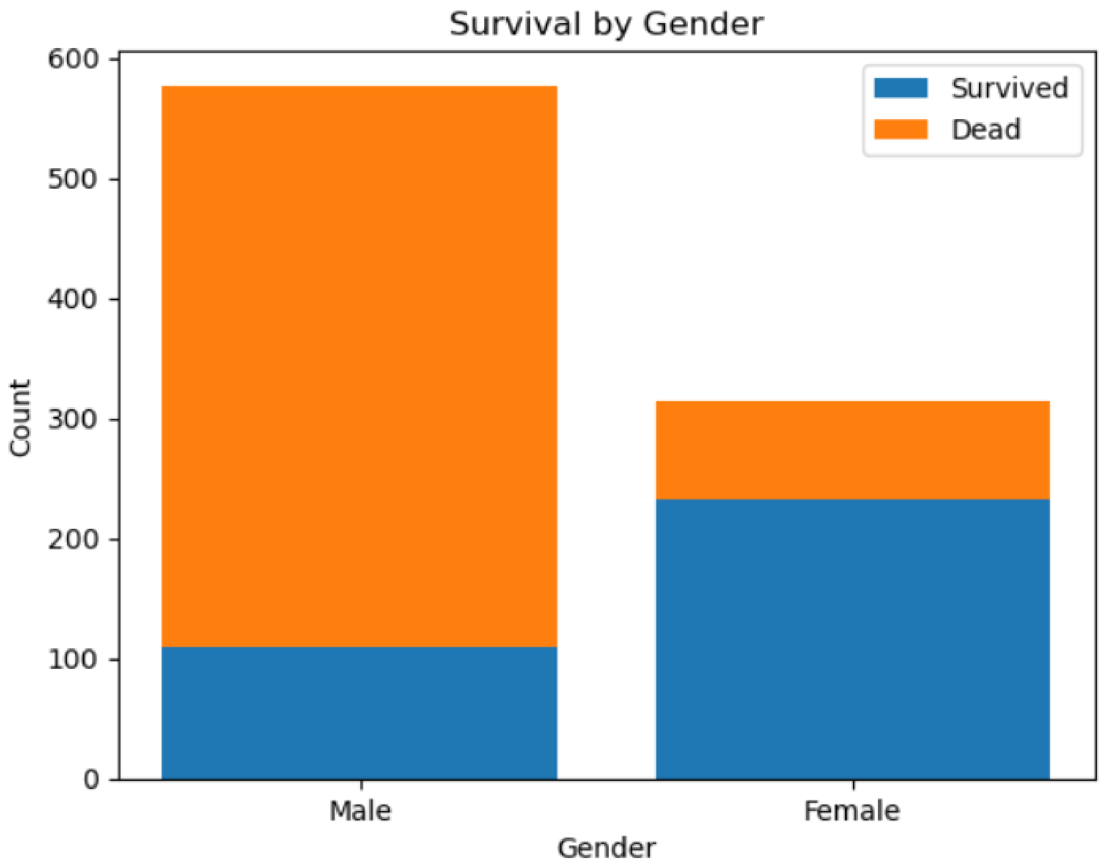


Step 5: Interpret Findings and present Insights

[154]:
```
dataset = pd.read_csv('titanic.csv')

survived_count = dataset['Survived'].value_counts()

survival_percentage = survived_count / len(dataset) * 100

plt.bar(survived_count.index, survived_count.values)
plt.xlabel('Survived')
plt.ylabel('Count')
plt.title('Survival Count')
plt.xticks(survived_count.index, ['No', 'Yes'])

plt.text(0, survived_count[0], f'{survived_count[0]} ({survival_percentage[0]:.2f}%)', ha='center', va='bottom')
plt.text(1, survived_count[1], f'{survived_count[1]} ({survival_percentage[1]:.2f}%)', ha='center', va='bottom')

plt.show()
```



[218]:
```
dataset = sns.load_dataset('titanic')
male = dataset[dataset['sex'] == 'male']
female = dataset[dataset['sex'] == 'female']

male_survived = male[male['survived'] == 1].shape[0]
male_dead = male[male['survived'] == 0].shape[0]
female_survived = female[female['survived'] == 1].shape[0]
female_dead = female[female['survived'] == 0].shape[0]

plt.bar(['Male', 'Female'], [male_survived, female_survived], label='Survived')
plt.bar(['Male', 'Female'], [male_dead, female_dead], bottom=[male_survived, female_survived], label='Dead')
plt.title('Survival by Gender')
plt.xlabel('Gender')
plt.ylabel('Count')
plt.legend()
plt.show()

print('Males Survived = ',male_survived)
print("Males Dead = ",male_dead)
print('Females Survived = ',female_survived)
print('Females Dead = ',female_dead)

percentage_male_survived=round((100*(male_survived)/(male_survived+male_dead)),2)
print('Male Survived = ',percentage_male_survived,'%')

percentage_female_survived=round((100*(female_survived)/(female_survived+female_dead)),2)
print('Female Survived = ',percentage_female_survived,'%')
```




```
Males Survived =  109
Males Dead =  468
Females Survived =  233
Females Dead =  81
Male Survived =  18.89 %
Female Survived =  74.2 %
```

Findings & Insights from the Dataset:

- 109 Males & 233 Females Survived.
- 468 Males & 81 Females Dead
- 18.89 % Males & 74.2 % Females Survived
- 38.38 % People Survived

CONCLUSION: We have learnt, understood and performed exploratory data analysis and created visualizations to gain insights from titanic dataset.