In [1]:

```
### Externship assignment 1
### Selva Manooj M
### 20MID0189
### VIT VELLORE
```

In [3]:

```
"""

Vellore Titanic Ship Case Study: Perform Below Tasks to
complete the assignment:-

1. Download the dataset: Dataset
2. Load the dataset.
3. Perform Below Visualizations.
    ● Univariate Analysis
    ● Bi - Variate Analysis
    ● Multi - VariateAnalysis
4. Perform descriptive statistics on the dataset.
5. Handle the Missing values.
6. Find the outliers and replace the outliers
7. Check for Categorical columns and perform encoding.
8. Split the data into dependent and independent variables.
9. Scale the independent variables
10. Split the data into training and testing

"""
```

Out[3]:

```
'\n\nVellore Titanic Ship Case Study: Perform Below Tasks to\ncomplete the
assignment:-\n\n1. Download the dataset: Dataset\n2. Load the dataset.\n3.
Perform Below Visualizations. \n    ● Univariate Analysis \n    ● Bi - Var
iate Analysis \n    ● Multi - VariateAnalysis\n4. Perform descriptive stat
istics on the dataset.\n5. Handle the Missing values.\n6. Find the outlier
s and replace the outliers\n7. Check for Categorical columns and perform e
ncoding.\n8. Split the data into dependent and independent variables.\n9.
Scale the independent variables\n10. Split the data into training and test
ing\n\n'
```

In [4]:

```
#1. Download the dataset: Dataset
```

In [5]:

```
#2. Load the dataset.
```

In [6]:

```
import pandas as pd
Titanic = pd.read_csv('C:/Users/imsel/Downloads/titanic.csv')
Titanic
```

Out[6]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Far
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.250
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.283
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.925
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.100
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.050
...	...	...	...	...	...	...	...	...	...	.
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.000
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.000
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.450
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.000
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.750

891 rows × 12 columns

In [7]:

```
Titanic.head(15)
```

Out[7]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	F
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0
5	6	0	3	Moran, Mr. James	male	NaN	0	0	330877	8.4
6	7	0	1	McCarthy, Mr. Timothy J	male	54.0	0	0	17463	51.8
7	8	0	3	Palsson, Master. Gosta Leonard	male	2.0	3	1	349909	21.0
8	9	1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27.0	0	2	347742	11.1
9	10	1	2	Nasser, Mrs. Nicholas (Adele Achem)	female	14.0	1	0	237736	30.0
10	11	1	3	Sandstrom, Miss. Marguerite Rut	female	4.0	1	1	PP 9549	16.7
11	12	1	1	Bonnell, Miss. Elizabeth	female	58.0	0	0	113783	26.5
12	13	0	3	Saundercock, Mr. William Henry	male	20.0	0	0	A/5. 2151	8.0
13	14	0	3	Andersson, Mr. Anders Johan	male	39.0	1	5	347082	31.2
14	15	0	3	Vestrom, Miss. Hulda Amanda Adolfina	female	14.0	0	0	350406	7.8

In [10]:

```
Titanic.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
```

In [15]:

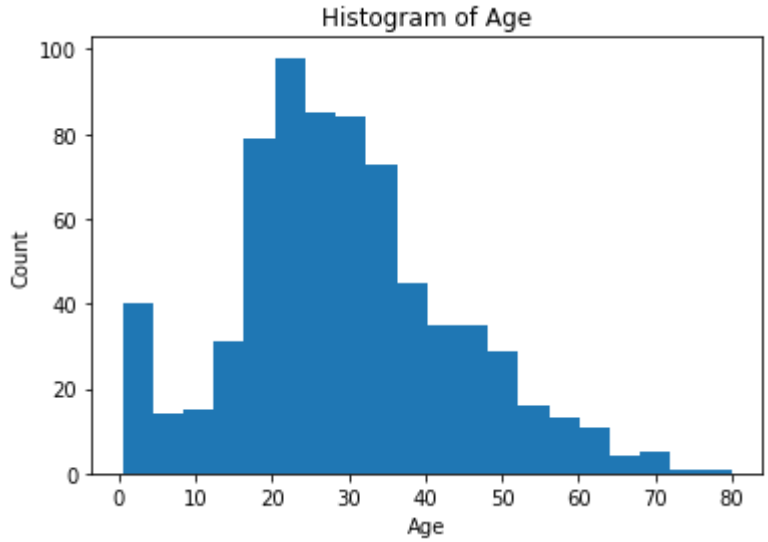
```
# 3.Visualisation
```

In [18]:

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

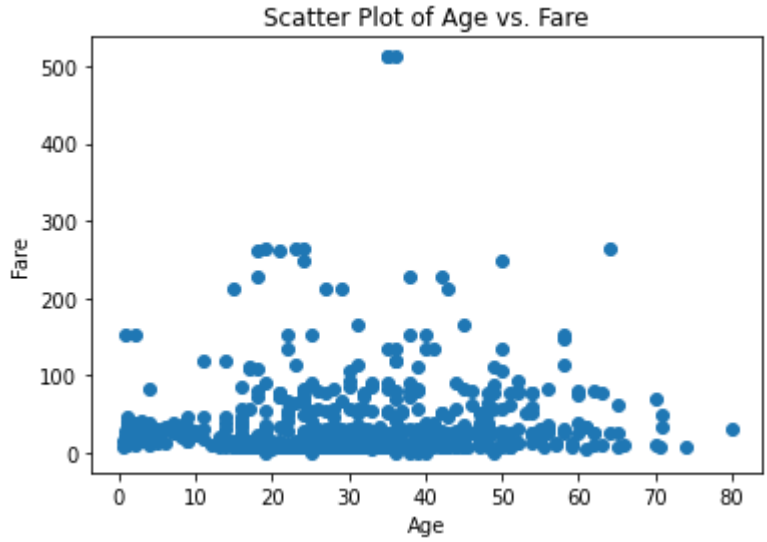
In [20]:

```
# Univariate Analysis
# Example: Histogram of Age
plt.hist(titanic_data['Age'].dropna(), bins=20)
plt.xlabel('Age')
plt.ylabel('Count')
plt.title('Histogram of Age')
plt.show()
```



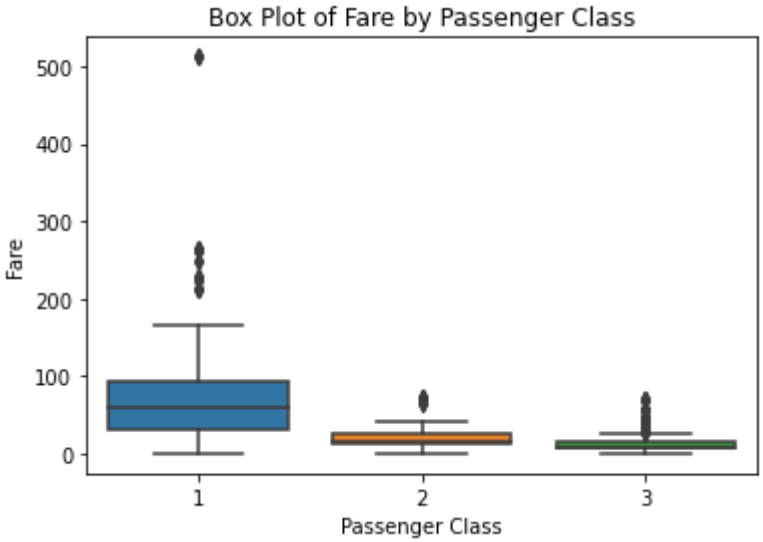
In [21]:

```
# Bi-Variate Analysis
# Example: Scatter plot of Age vs. Fare
plt.scatter(titanic_data['Age'], titanic_data['Fare'])
plt.xlabel('Age')
plt.ylabel('Fare')
plt.title('Scatter Plot of Age vs. Fare')
plt.show()
```



In [22]:

```
# Multi-Variate Analysis
# Example: Box plot of Fare by Passenger Class
sns.boxplot(x='Pclass', y='Fare', data=titanic_data)
plt.xlabel('Passenger Class')
plt.ylabel('Fare')
plt.title('Box Plot of Fare by Passenger Class')
plt.show()
```



In [23]:

```
#4. Perform descriptive statistics on the dataset.
```

In [24]:

```
# Perform descriptive statistics
statistics = titanic_data.describe()

# Display the descriptive statistics
print(statistics)
```

```
       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
```

In [ ]:

```
# 5. Handle the Missing values.
```

In [26]:

```
# Check for missing values
missing_values = titanic_data.isnull().sum()
print("missing values before handling")
print(missing_values)


titanic_data['Age'].fillna(titanic_data['Age'].mean(), inplace=True)
titanic_data['Fare'].fillna(titanic_data['Fare'].mean(), inplace=True)

titanic_data.dropna(subset=['Cabin', 'Embarked'], inplace=True)

# Verify if missing values have been handled
print("missing values after handling")
missing_values_after_handling = titanic_data.isnull().sum()
print(missing_values_after_handling)
```

```
missing values before handling
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
missing values after handling
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
```

In [27]:

```
# 6. Find the outliers and replace the outliers
```

In [29]:

```
from scipy.stats import zscore
import numpy as np

outlier_threshold = 3

numerical_columns = ['Age', 'Fare']
z_scores = np.abs(zscore(titanic_data[numerical_columns]))
outlier_indices = np.where(z_scores > outlier_threshold)
titanic_data[numerical_columns] = np.where(z_scores > outlier_threshold, titanic_data[num
outlier_threshold_value = titanic_data[numerical_columns].mean() + (outlier_threshold * t
titanic_data[numerical_columns] = np.where(titanic_data[numerical_columns] > outlier_thre

z_scores_after_replacement = np.abs(zscore(titanic_data[numerical_columns]))
outliers_after_replacement = np.where(z_scores_after_replacement > outlier_threshold)

print("Indices of replaced outliers:", outliers_after_replacement)
```

```
Indices of replaced outliers: (array([  7,  15,  67,  79,  95, 170], dtype
=int64), array([1, 1, 1, 1, 1, 1], dtype=int64))
```

In [31]:

```
# 7. Check for Categorical columns and perform encoding.
```

In [30]:

```
categorical_columns = titanic_data.select_dtypes(include=['object']).columns
encoded_data = pd.get_dummies(titanic_data, columns=categorical_columns)

print(encoded_data.head())
```

```
    PassengerId  Survived  Pclass   Age  SibSp  Parch     Fare  \
1             2         1       1  38.0      1      0  71.2833
3             4         1       1  35.0      1      0  53.1000
6             7         0       1  54.0      0      0  51.8625
10           11         1       3   4.0      1      1  16.7000
11           12         1       1  58.0      0      0  26.5500

    Name_Allen, Miss. Elisabeth Walton  Name_Allison, Master. Hudson Trevo
r  \
1                                    0
0
3                                    0
0
6                                    0
0
10                                   0
0
11                                   0
0

    Name_Allison, Miss. Helen Loraine  ...  Cabin_F G73  Cabin_F2  Cabin_F
33  \
1                                   0  ...            0         0
0
3                                   0  ...            0         0
0
6                                   0  ...            0         0
0
10                                  0  ...            0         0
0
11                                  0  ...            0         0
0

    Cabin_F38  Cabin_F4  Cabin_G6  Cabin_T  Embarked_C  Embarked_Q  Embark
ed_S
1           0         0         0        0           1           0
0
3           0         0         0        0           0           0
1
6           0         0         0        0           0           0
1
10          0         0         1        0           0           0
1
11          0         0         0        0           0           0
1

[5 rows x 501 columns]
```

In [32]:

```
# 8. Split the data into dependent and independent variables.
```

In [33]:

```
X = titanic_data.drop("Survived", axis=1)
y = titanic_data["Survived"]

print(X.head())
print(y.head())
```

```
    PassengerId  Pclass                                               Name
\
1             2       1  Cumings, Mrs. John Bradley (Florence Briggs Th...
3             4       1       Futrelle, Mrs. Jacques Heath (Lily May Peel)
6             7       1                            McCarthy, Mr. Timothy J
10           11       3                  Sandstrom, Miss. Marguerite Rut
11           12       1                         Bonnell, Miss. Elizabeth

       Sex   Age  SibSp  Parch    Ticket     Fare Cabin Embarked
1   female  38.0      1      0  PC 17599  71.2833   C85        C
3   female  35.0      1      0    113803  53.1000  C123        S
6     male  54.0      0      0     17463  51.8625   E46        S
10  female   4.0      1      1   PP 9549  16.7000    G6        S
11  female  58.0      0      0    113783  26.5500  C103        S
1     1
3     1
6     0
10    1
11    1
Name: Survived, dtype: int64
```

In [34]:

```
# 9. Scale the independent variable
```

In [38]:

```
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()
X_encoded = pd.get_dummies(X)
```

In [35]:

```
# 10. Split the data into training and testing
```

In [40]:

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Display the shapes of the training and testing sets
print("X_train shape:", X_train.shape)
print("y_train shape:", y_train.shape)
print("X_test shape:", X_test.shape)
print("y_test shape:", y_test.shape)
```

```
X_train shape: (161, 11)
y_train shape: (161,)
X_test shape: (41, 11)
y_test shape: (41,)
```