VIT Applied Data Science 2023

Assignment 2

Ishaan Singh Bains
ishaansingh.bains2020@vitstudent.ac.in

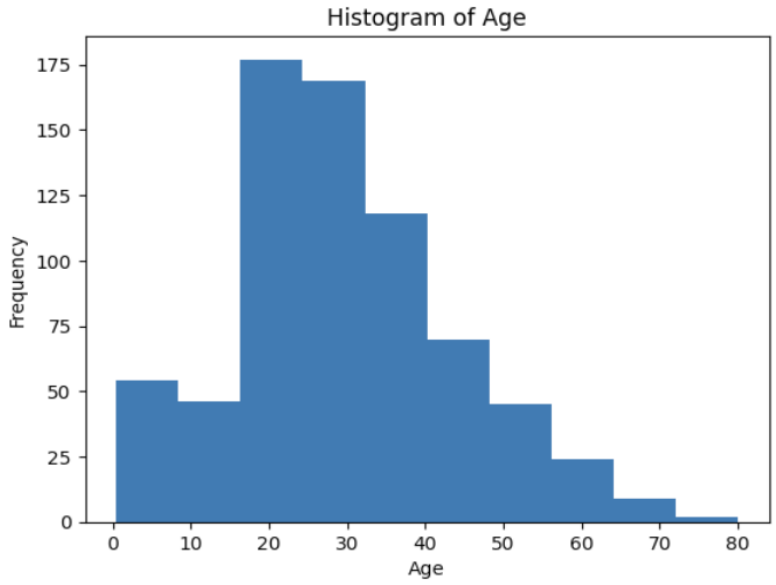
1. Download the dataset: Dataset
2. Load the dataset.

In [18]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
data = pd.read_csv(r'C:\Study\SmartBridge\Assignments\CSV\titanic.csv')
```

3. Perform Below Visualizations.

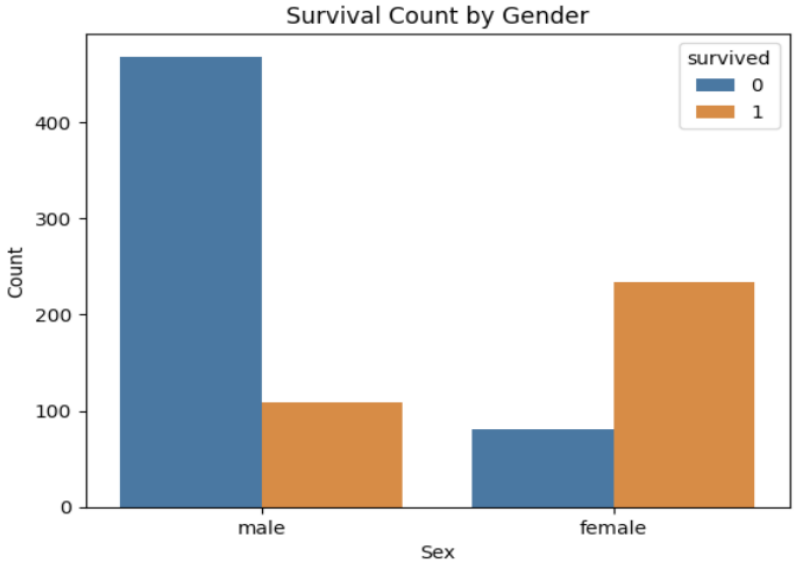
- Univariate Analysis

In [19]:
```
# Univariate Analysis - Histogram of Age
plt.hist(data["age"].dropna(), bins=10)
plt.xlabel("Age")
plt.ylabel("Frequency")
plt.title("Histogram of Age")
plt.show()
```



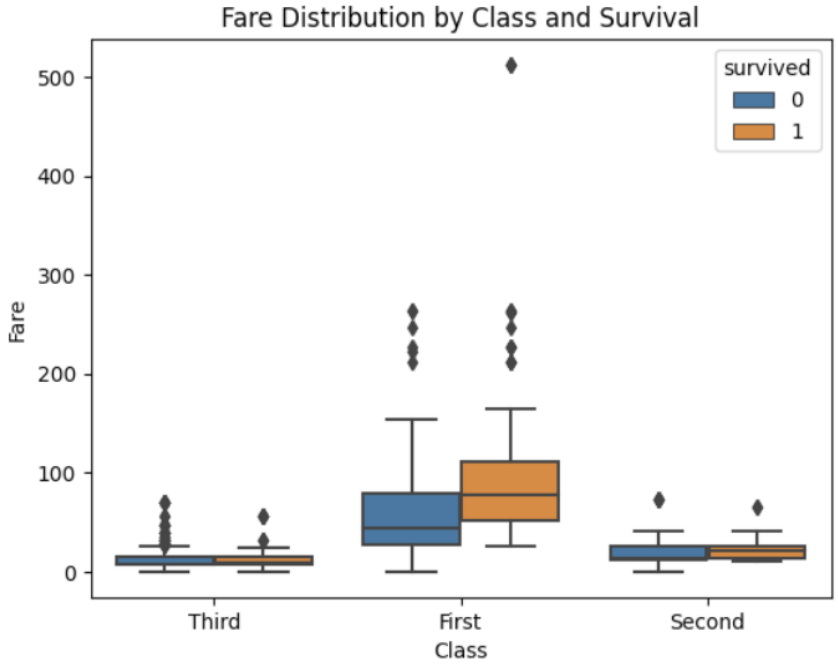
- Bi - Variate Analysis

In [20]:
```
# Bivariate Analysis - Bar Plot of Survival by Gender
sns.countplot(x="sex", hue="survived", data=data)
plt.xlabel("Sex")
plt.ylabel("Count")
plt.title("Survival Count by Gender")
plt.show()
```



- Multi - Variate Analysis

In [21]:
```
# Multivariate Analysis - Box Plot of Fare by Class and Survival
sns.boxplot(x="class", y="fare", hue="survived", data=data)
plt.xlabel("Class")
plt.ylabel("Fare")
plt.title("Fare Distribution by Class and Survival")
plt.show()
```



4. Perform descriptive statistics on the dataset.

In [22]:
```
# Perform descriptive statistics
statistics = data.describe()
print(statistics)
```

	survived	pclass	age	sibsp	parch	fare
count	891.000000	891.000000	714.000000	891.000000	891.000000	891.000000
mean	0.383838	2.308642	29.699118	0.523008	0.381594	32.204208
std	0.486592	0.836071	14.526497	1.102743	0.806057	49.693429
min	0.000000	1.000000	0.420000	0.000000	0.000000	0.000000
25%	0.000000	2.000000	20.125000	0.000000	0.000000	7.910400
50%	0.000000	3.000000	28.000000	0.000000	0.000000	14.454200
75%	1.000000	3.000000	38.000000	1.000000	0.000000	31.000000
max	1.000000	3.000000	80.000000	8.000000	6.000000	512.329200

5. Handle the Missing values.

```
# Handle missing values
missing_cols = ["age", "fare"]
imputer = SimpleImputer(strategy="mean")
data[missing_cols] = imputer.fit_transform(data[missing_cols])
```

6. Find the outliers and replace the outliers

In [52]:
```
# Find and replace outliers
numeric_cols = ["age", "fare"]
z_scores = np.abs(zscore(data[numeric_cols]))
threshold = 3
outliers = np.where(z_scores > threshold)
data[numeric_cols] = np.where(z_scores > threshold, np.nan, data[numeric_cols])
data[numeric_cols] = imputer.fit_transform(data[numeric_cols])
```

7. Check for Categorical columns and perform encoding.

In [60]:
```
# Check for categorical columns
categorical_columns = data.select_dtypes(include=["object"]).columns

# Perform encoding using pandas get_dummies function
data_encoded = pd.get_dummies(data, columns=categorical_columns)
```

8. Split the data into dependent and independent variables.

In [61]:
```
# Split the data into dependent and independent variables
X = data_encoded.drop("survived", axis=1)  # Independent variables (features)
y = data_encoded["survived"]  # Dependent variable (target)
```

9. Scale the independent variables

In [62]:
```
# Scale the independent variables
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
```

10. Split the data into training and testing

In [63]:
```
# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2, random_state=
```

In [65]:
```
print("Shape of X_train:", X_train.shape)
print("Shape of X_test:", X_test.shape)
print("Shape of y_train:", y_train.shape)
print("Shape of y_test:", y_test.shape)
```

```
Shape of X_train: (712, 23)
Shape of X_test: (179, 23)
Shape of y_train: (712,)
Shape of y_test: (179,)
```