Exploratory Data Analysis (EDA) - Titanic Dataset

Objective: Extract insights from the Titanic dataset using visual and statistical exploration.
Tools Used: Python, Pandas, Matplotlib, Seaborn
Dataset: Titanic.csv

1. Import Required Libraries

We will import the libraries needed for data manipulation and visualization.

In [1]:
```
# Install (if not already installed)
# !pip install pandas matplotlib seaborn

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

```
C:\Users\surut\anaconda3\Lib\site-packages\pandas\core\arrays\masked.py:60: UserWarning: Pandas
requires version '1.3.6' or newer of 'bottleneck' (version '1.3.5' currently installed).
  from pandas.core import (
```

2. Load the Dataset

We will load the Titanic dataset into a Pandas DataFrame.

In [2]:
```
# Load Titanic dataset
df = pd.read_csv("titanic.csv")  # Replace with your file path
df.head()
```

Out[2]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	892	0	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292	NaN	Q
1	893	1	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	363272	7.0000	NaN	S
2	894	0	2	Myles, Mr. Thomas Francis	male	62.0	0	0	240276	9.6875	NaN	Q
3	895	0	3	Wirz, Mr. Albert	male	27.0	0	0	315154	8.6625	NaN	S
4	896	1	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1	3101298	12.2875	NaN	S

3. Basic Data Exploration

We will explore:

- Summary statistics
- Data types
- Missing values
- Unique value counts for categorical columns

In [3]:
```
# Summary statistics for numeric columns
df.describe()

# Data types, null values, non-null counts
df.info()

# Value counts for important categorical columns
print(df['Sex'].value_counts())
print(df['Pclass'].value_counts())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 418 entries, 0 to 417
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   PassengerId  418 non-null    int64  
 1   Survived     418 non-null    int64  
 2   Pclass       418 non-null    int64  
 3   Name         418 non-null    object 
 4   Sex          418 non-null    object 
 5   Age          332 non-null    float64
 6   SibSp        418 non-null    int64  
 7   Parch        418 non-null    int64  
 8   Ticket       418 non-null    object 
 9   Fare         417 non-null    float64
 10  Cabin        91 non-null     object 
 11  Embarked     418 non-null    object 
dtypes: float64(2), int64(5), object(5)
memory usage: 39.3+ KB
Sex
male      266
female    152
Name: count, dtype: int64
Pclass
3    218
1    107
2     93
Name: count, dtype: int64
```

4. Pairplot Visualization

Pairplot shows relationships between multiple numeric variables, colored by survival status.

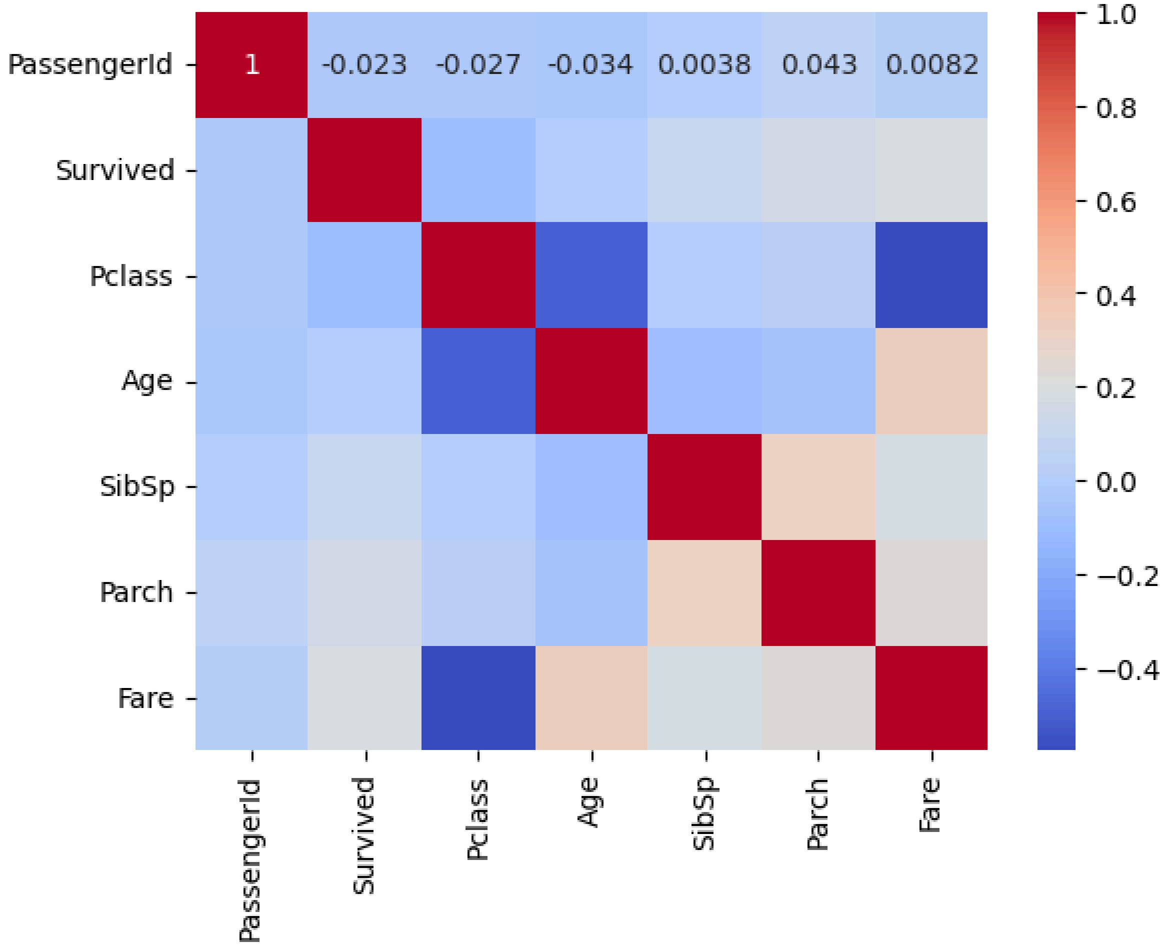
In [4]:
```
sns.pairplot(df.dropna(), hue='Survived')
plt.show()
```

```
C:\Users\surut\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_n
a option is deprecated and will be removed in a future version. Convert inf values to NaN before
operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
C:\Users\surut\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1075: FutureWarning: When groupin
g with a length-1 list-like, you will need to pass a length-1 tuple to get_group in a future ver
sion of pandas. Pass `(name,)` instead of `name` to silence this warning.
  data_subset = grouped_data.get_group(pd_key)
C:\Users\surut\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1075: FutureWarning: When groupin
g with a length-1 list-like, you will need to pass a length-1 tuple to get_group in a future ver
sion of pandas. Pass `(name,)` instead of `name` to silence this warning.
  data_subset = grouped_data.get_group(pd_key)
C:\Users\surut\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_n
a option is deprecated and will be removed in a future version. Convert inf values to NaN before
operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
C:\Users\surut\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1075: FutureWarning: When groupin
g with a length-1 list-like, you will need to pass a length-1 tuple to get_group in a future ver
sion of pandas. Pass `(name,)` instead of `name` to silence this warning.
  data_subset = grouped_data.get_group(pd_key)
C:\Users\surut\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1075: FutureWarning: When groupin
g with a length-1 list-like, you will need to pass a length-1 tuple to get_group in a future ver
sion of pandas. Pass `(name,)` instead of `name` to silence this warning.
  data_subset = grouped_data.get_group(pd_key)
C:\Users\surut\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_n
a option is deprecated and will be removed in a future version. Convert inf values to NaN before
operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
C:\Users\surut\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1075: FutureWarning: When groupin
g with a length-1 list-like, you will need to pass a length-1 tuple to get_group in a future ver
sion of pandas. Pass `(name,)` instead of `name` to silence this warning.
  data_subset = grouped_data.get_group(pd_key)
C:\Users\surut\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_n
a option is deprecated and will be removed in a future version. Convert inf values to NaN before
operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
C:\Users\surut\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1075: FutureWarning: When groupin
g with a length-1 list-like, you will need to pass a length-1 tuple to get_group in a future ver
sion of pandas. Pass `(name,)` instead of `name` to silence this warning.
  data_subset = grouped_data.get_group(pd_key)
C:\Users\surut\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1075: FutureWarning: When groupin
g with a length-1 list-like, you will need to pass a length-1 tuple to get_group in a future ver
sion of pandas. Pass `(name,)` instead of `name` to silence this warning.
  data_subset = grouped_data.get_group(pd_key)
C:\Users\surut\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_n
a option is deprecated and will be removed in a future version. Convert inf values to NaN before
operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
C:\Users\surut\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1075: FutureWarning: When groupin
g with a length-1 list-like, you will need to pass a length-1 tuple to get_group in a future ver
sion of pandas. Pass `(name,)` instead of `name` to silence this warning.
  data_subset = grouped_data.get_group(pd_key)
C:\Users\surut\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1075: FutureWarning: When groupin
g with a length-1 list-like, you will need to pass a length-1 tuple to get_group in a future ver
sion of pandas. Pass `(name,)` instead of `name` to silence this warning.
  data_subset = grouped_data.get_group(pd_key)
C:\Users\surut\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_n
a option is deprecated and will be removed in a future version. Convert inf values to NaN before
operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
C:\Users\surut\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1075: FutureWarning: When groupin
g with a length-1 list-like, you will need to pass a length-1 tuple to get_group in a future ver
sion of pandas. Pass `(name,)` instead of `name` to silence this warning.
  data_subset = grouped_data.get_group(pd_key)
```

5. Correlation Heatmap

The heatmap displays correlations between numerical variables.

In [5]:
```
corr = df.corr(numeric_only=True)  # For numerical columns
sns.heatmap(corr, annot=True, cmap='coolwarm')
plt.show()
```



6. Boxplot: Age vs Survived

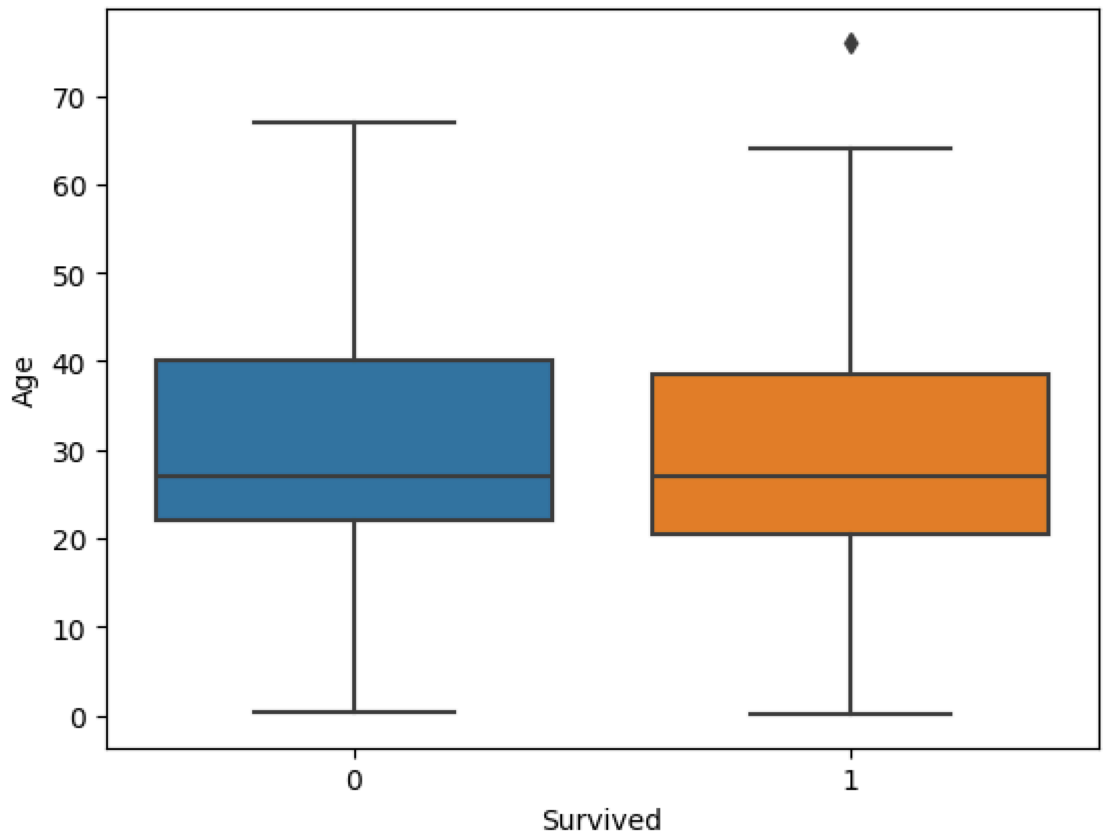
This plot shows the age distribution for survivors and non-survivors.

In [ ]:

```

```

In [6]:

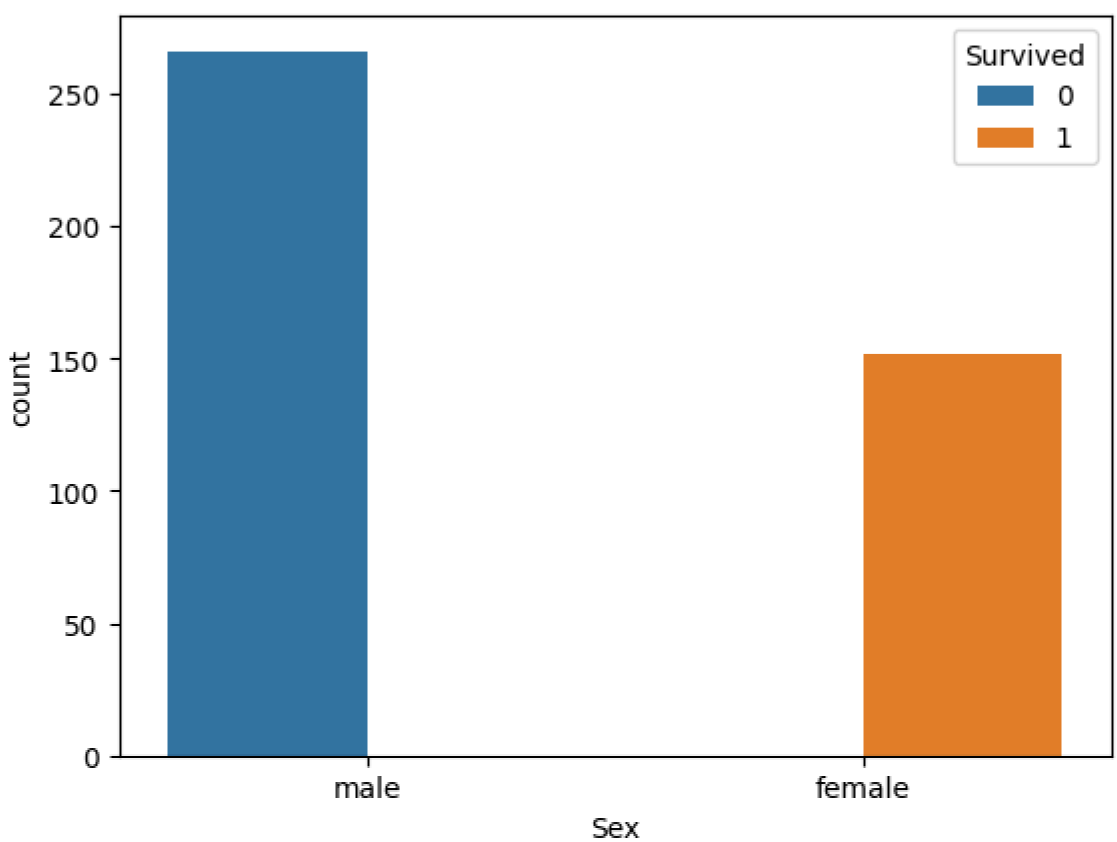
```
sns.boxplot(x='Survived', y='Age', data=df)
plt.show()
```



7. Countplot: Gender vs Survival

This plot compares survival counts between males and females.

In [7]:
```
sns.countplot(x='Sex', hue='Survived', data=df)
plt.show()
```



8. Histogram: Age Distribution

Histogram of passenger ages to observe distribution patterns.

9. Scatter Plot: Age vs Fare

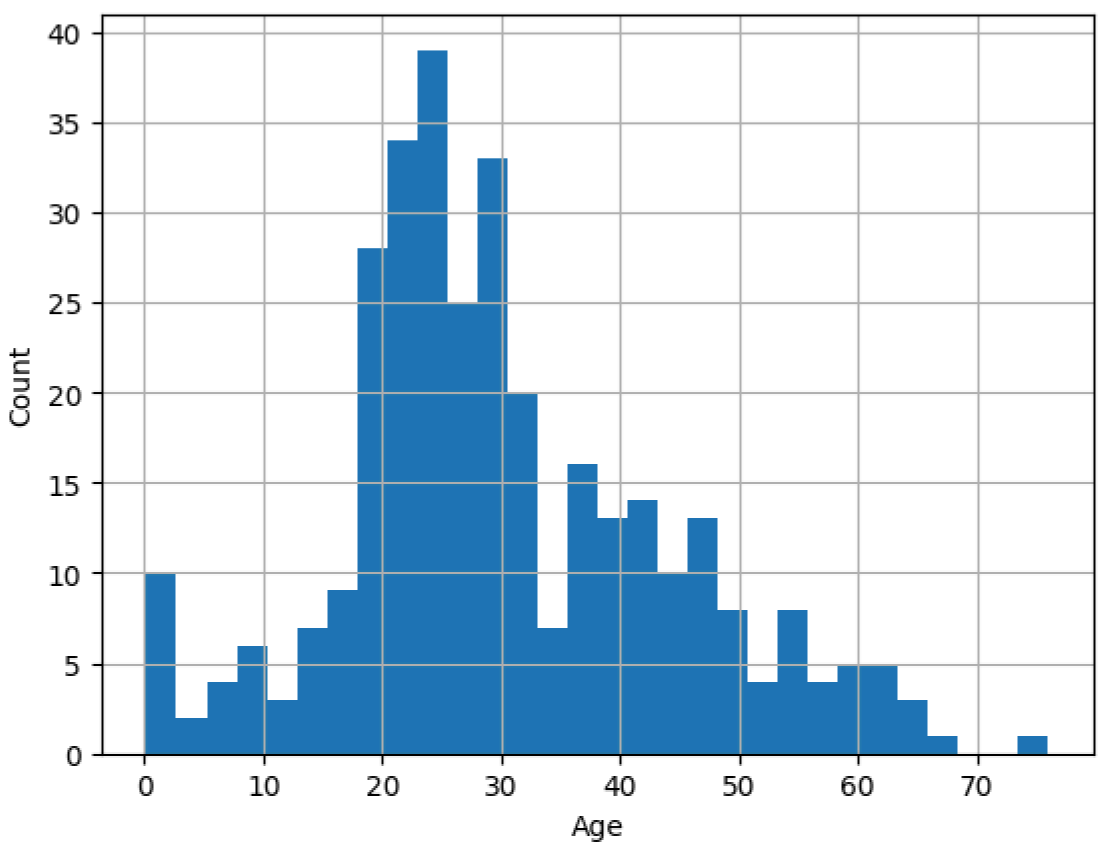
Scatter plot to check the relationship between passenger age and fare.

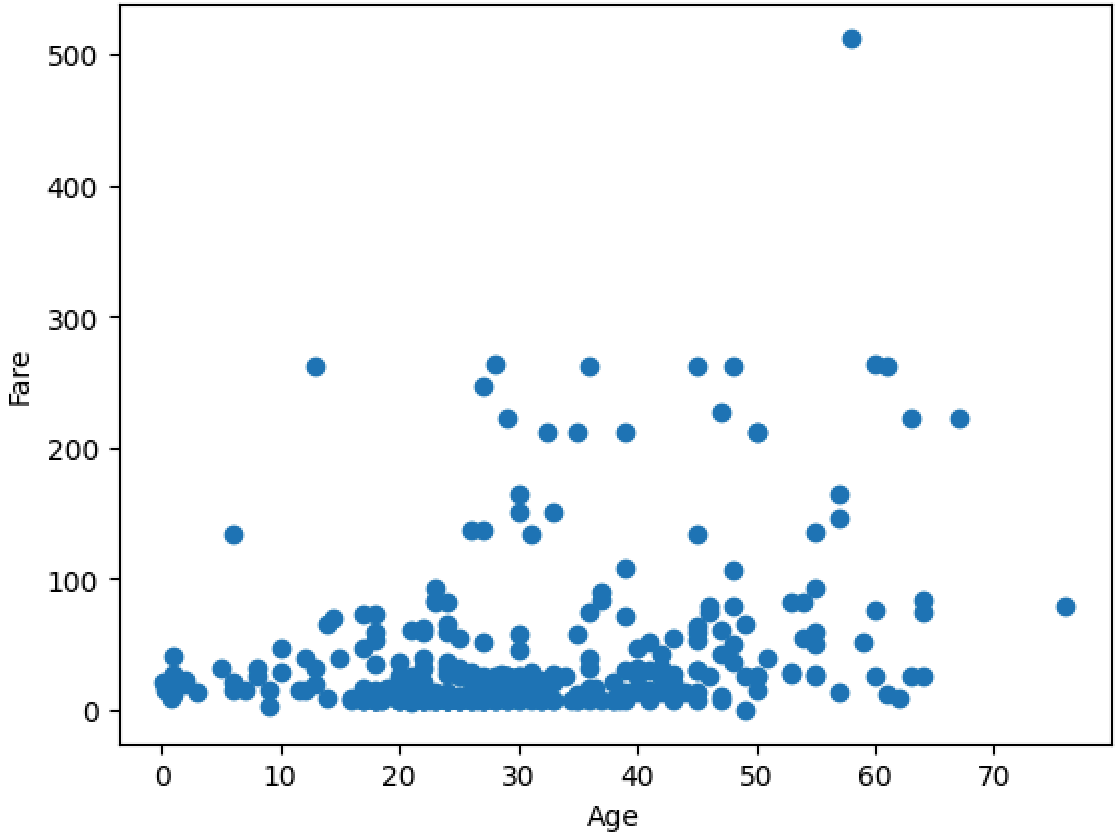
In [ ]:

```

```

In [8]:

```
# Histogram
df['Age'].hist(bins=30)
plt.xlabel('Age')
plt.ylabel('Count')
plt.show()

# Scatter Plot
plt.scatter(df['Age'], df['Fare'])
plt.xlabel('Age')
plt.ylabel('Fare')
plt.show()
```





10. Observations

- Females had a much higher survival rate than males.
- Passengers from higher classes had better chances of survival.
- Younger passengers had slightly higher survival chances.
- Higher fare amounts were correlated with higher survival rates.

11. Summary of Findings

The analysis reveals that:

1. **Gender** was a strong determinant of survival.
2. **Passenger class** influenced survival chances significantly.
3. Younger passengers had slightly higher chances of survival.
4. Higher fares were generally linked with higher survival rates.