


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


df=pd.read_csv('/content/drive/MyDrive/titanic/gender_submission.csv')
df
```



	PassengerId	Survived
0	892	0
1	893	1
2	894	0
3	895	0
4	896	1
...	...	...
413	1305	0
414	1306	1
415	1307	0
416	1308	0
417	1309	0

418 rows × 2 columns

```
df_1=pd.read_csv('/content/drive/MyDrive/titanic/test.csv')
df_1
```



	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	892	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292	NaN	Q
1	893	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	363272	7.0000	NaN	S
2	894	2	Myles, Mr. Thomas Francis	male	62.0	0	0	240276	9.6875	NaN	Q
3	895	3	Wirz, Mr. Albert	male	27.0	0	0	315154	8.6625	NaN	S
4	896	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1	3101298	12.2875	NaN	S
...	...	...	...	...	...	...	...	...	...	...	...
413	1305	3	Spector, Mr. Woolf	male	NaN	0	0	A.5. 3236	8.0500	NaN	S
414	1306	1	Oliva y Ocana, Dona. Fermina	female	39.0	0	0	PC 17758	108.9000	C105	C
415	1307	3	Saether, Mr. Simon Sivertsen	male	38.5	0	0	SOTON/O.Q. 3101262	7.2500	NaN	S
416	1308	3	Ware, Mr. Frederick	male	NaN	0	0	359309	8.0500	NaN	S
417	1309	3	Peter, Master. Michael J	male	NaN	1	1	2668	22.3583	NaN	C

```
df_2=pd.read_csv('/content/drive/MyDrive/titanic/train.csv')
df_2
```

	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
...	...	...	...	...	...	...	...	...	...	...	...	...
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0000	NaN	S
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000	B42	S
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.4500	NaN	S
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0000	C148	C

```
df.shape
```

```
(418, 2)
```

```
df_1.shape
```

```
(418, 11)
```

```
df_2.shape
```

```
(891, 12)
```

```
# Check for missing values in each column
df_2.isnull().sum()
```

```
# Impute missing values in 'Age' with the median
df_2['Age'].fillna(df_2['Age'].median(), inplace=True)
```

```
# Drop rows with missing values in 'Embarked' (if any)
df_2.dropna(subset=['Embarked'], inplace=True)
```

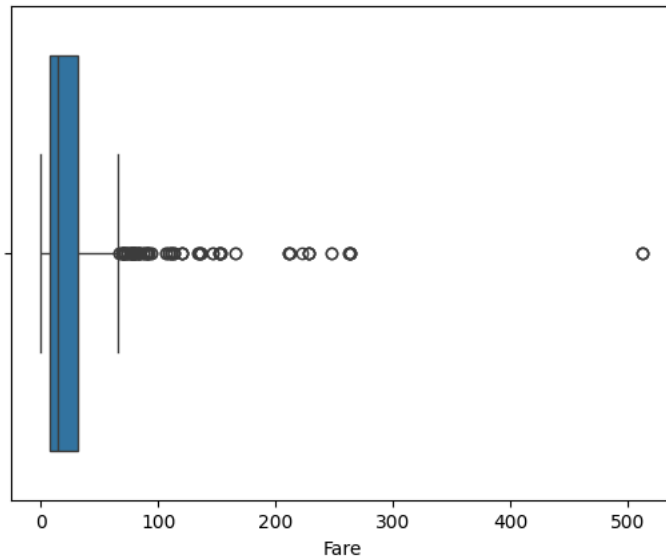
```
<ipython-input-9-25071d4129aa>:5: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained assignment. The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting
```

```
For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col]
```

```
df_2['Age'].fillna(df_2['Age'].median(), inplace=True)
```

```
# Visualize outliers using box plots
sns.boxplot(x=df_2['Fare'])
```

```
# Remove outliers using the IQR method
Q1 = df_2['Fare'].quantile(0.25)
Q3 = df_2['Fare'].quantile(0.75)
IQR = Q3 - Q1
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR
df_2 = df_2[(df_2['Fare'] >= lower_bound) & (df_2['Fare'] <= upper_bound)]
```



```
# Convert 'Sex' to a categorical variable
df_2['Sex'] = df_2['Sex'].astype('category')
```



```
<ipython-input-11-d563c4e039c4>:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

```
df_2['Sex'] = df_2['Sex'].astype('category')
```

```
df_2.describe()
```

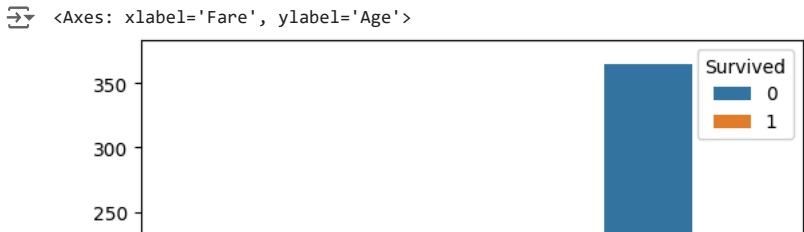


	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
count	775.000000	775.000000	775.000000	775.000000	775.000000	775.000000	775.000000
mean	445.806452	0.339355	2.480000	28.748710	0.437419	0.340645	17.822091
std	260.116285	0.473796	0.73439	12.782123	0.899838	0.785914	13.578085
min	1.000000	0.000000	1.00000	0.420000	0.000000	0.000000	0.000000
25%	213.500000	0.000000	2.00000	22.000000	0.000000	0.000000	7.895800
50%	450.000000	0.000000	3.00000	28.000000	0.000000	0.000000	13.000000
75%	670.500000	1.000000	3.00000	34.000000	1.000000	0.000000	26.000000
max	891.000000	1.000000	3.00000	80.000000	5.000000	6.000000	65.000000

```
# Histogram of 'Age'
plt.hist(df_2['Age'])
plt.xlabel('Age')
plt.ylabel('Frequency')
```

```
# Scatter plot of 'Fare' vs 'Age'
plt.scatter(df_2['Fare'], df_2['Age'])
plt.xlabel('Fare')
plt.ylabel('Age')
```

```
# Bar plot of 'Survived' by 'Pclass'
sns.countplot(x='Pclass', hue='Survived', data=df_2)
```



```
# Select only numerical features for correlation analysis
numerical_features = df_2.select_dtypes(include=['number'])

# Calculate correlation matrix for numerical features
correlation_matrix = numerical_features.corr()

# Visualize the correlation matrix using a heatmap
sns.heatmap(correlation_matrix, annot=True)
```

