```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
import warnings
warnings.filterwarnings("ignore")
%matplotlib inline
```

Load data and basic stats

```
df = pd.read_csv("titanic.csv")
```

```
df.shape
```

```
(891, 12)
```

```
df.head()
```

	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

Next steps: Generate code with df | View recommended plots | New interactive sheet

```
df.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
```

```
df.describe()
```

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
count	891.000000	891.000000	891.000000	714.000000	891.000000	891.000000	891.000000
mean	446.000000	0.383838	2.308642	29.699118	0.523008	0.381594	32.204208
std	257.353842	0.486592	0.836071	14.526497	1.102743	0.806057	49.693429
min	1.000000	0.000000	1.000000	0.420000	0.000000	0.000000	0.000000
25%	223.500000	0.000000	2.000000	20.125000	0.000000	0.000000	7.910400
50%	446.000000	0.000000	3.000000	28.000000	0.000000	0.000000	14.454200
75%	668.500000	1.000000	3.000000	38.000000	1.000000	0.000000	31.000000
max	891.000000	1.000000	3.000000	80.000000	8.000000	6.000000	512.329200

```
df.isna().sum()
```

	0
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

```
df["Age"] = df["Age"].fillna(df["Age"].mean())
```

```
df.isna().sum()
```

	0
PassengerId	0
Survived	0
Pclass	0
Name	0
Sex	0
Age	0
SibSp	0
Parch	0
Ticket	0
Fare	0
Cabin	687
Embarked	2

dtype: int64

Visualization

```
def fun1(value):
    if (value == "male"):
        return 1
    else:
        return 0
```

```
def fun2(value):
    if (value == 'S'):
        return 0
    elif (value == 'C'):
        return 1
    elif (value == 'Q'):
        return 2
    else:
        return 0
```

```
df["Sex"] = df["Sex"].apply(fun1)
```

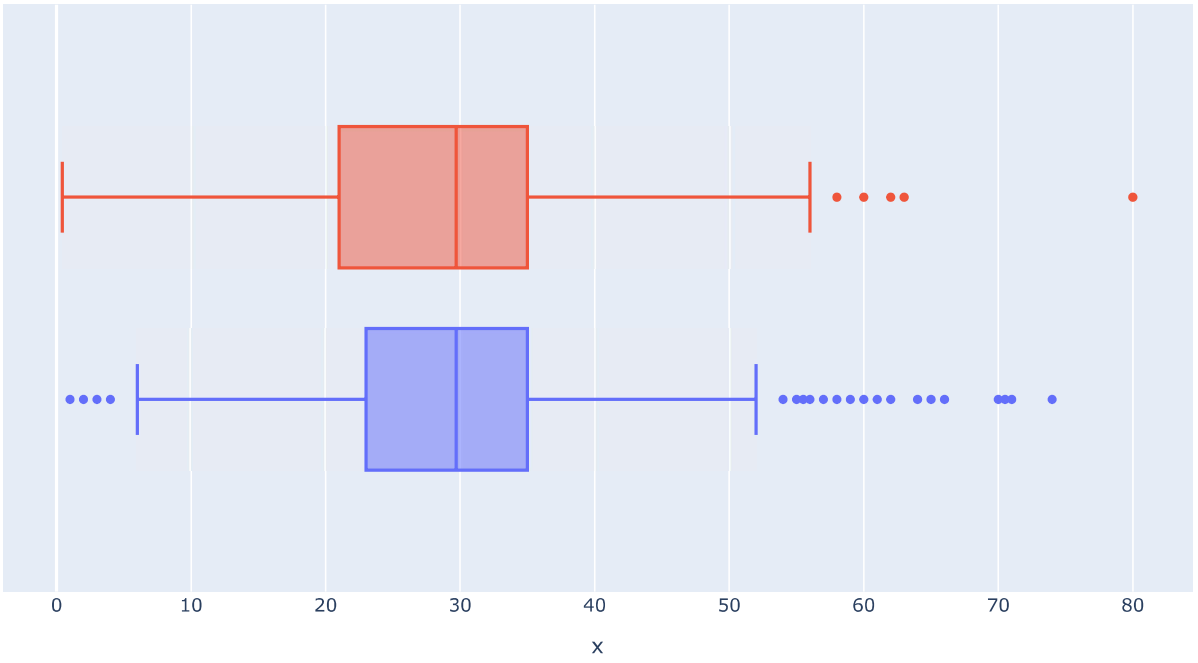
```
df["Embarked"] = df["Embarked"].apply(fun2)
```

```
df = df.drop("Cabin", axis=1)
```

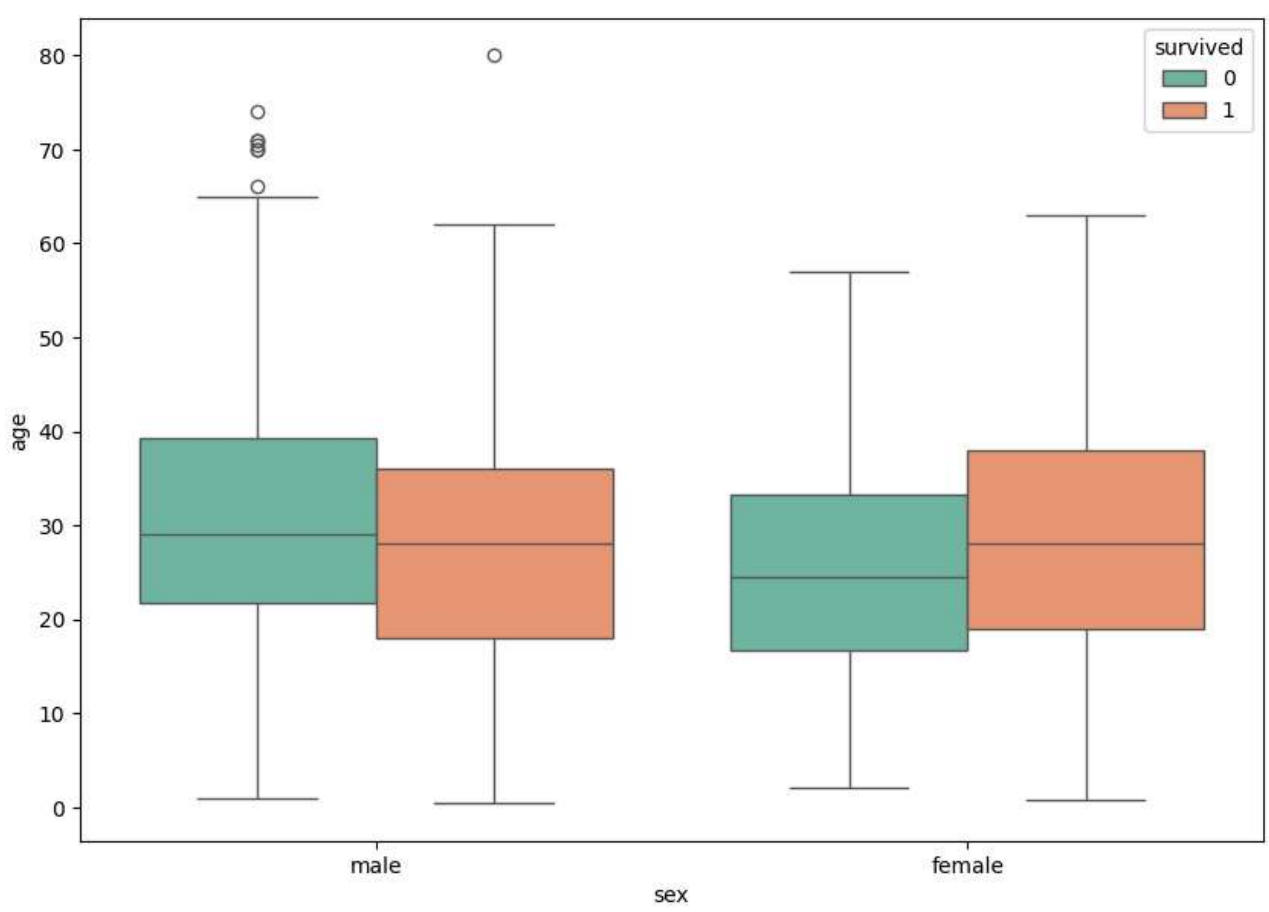
```
df.shape
```

```
(891, 11)
```

```
px.box(df["Sex"], df["Age"], color=df["Survived"])
```




```
plt.figure(figsize=(10,7))
box = sns.boxplot(data=titanic_clean, x='sex', y='age', hue='survived', palette='Set2')
plt.show()
```




```
# Import necessary libraries
import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd

# Load Titanic dataset
titanic = sns.load_dataset('titanic')

# Display first few rows
titanic.head()
```

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	deck	embark_town	ali
0	0	3	male	22.0	1	0	7.2500	S	Third	man	True	NaN	Southampton	
1	1	1	female	38.0	1	0	71.2833	C	First	woman	False	C	Cherbourg	y
2	1	3	female	26.0	0	0	7.9250	S	Third	woman	False	NaN	Southampton	y
3	1	1	female	35.0	1	0	53.1000	S	First	woman	False	C	Southampton	y
4	0	3	male	35.0	0	0	8.0500	S	Third	man	True	NaN	Southampton	

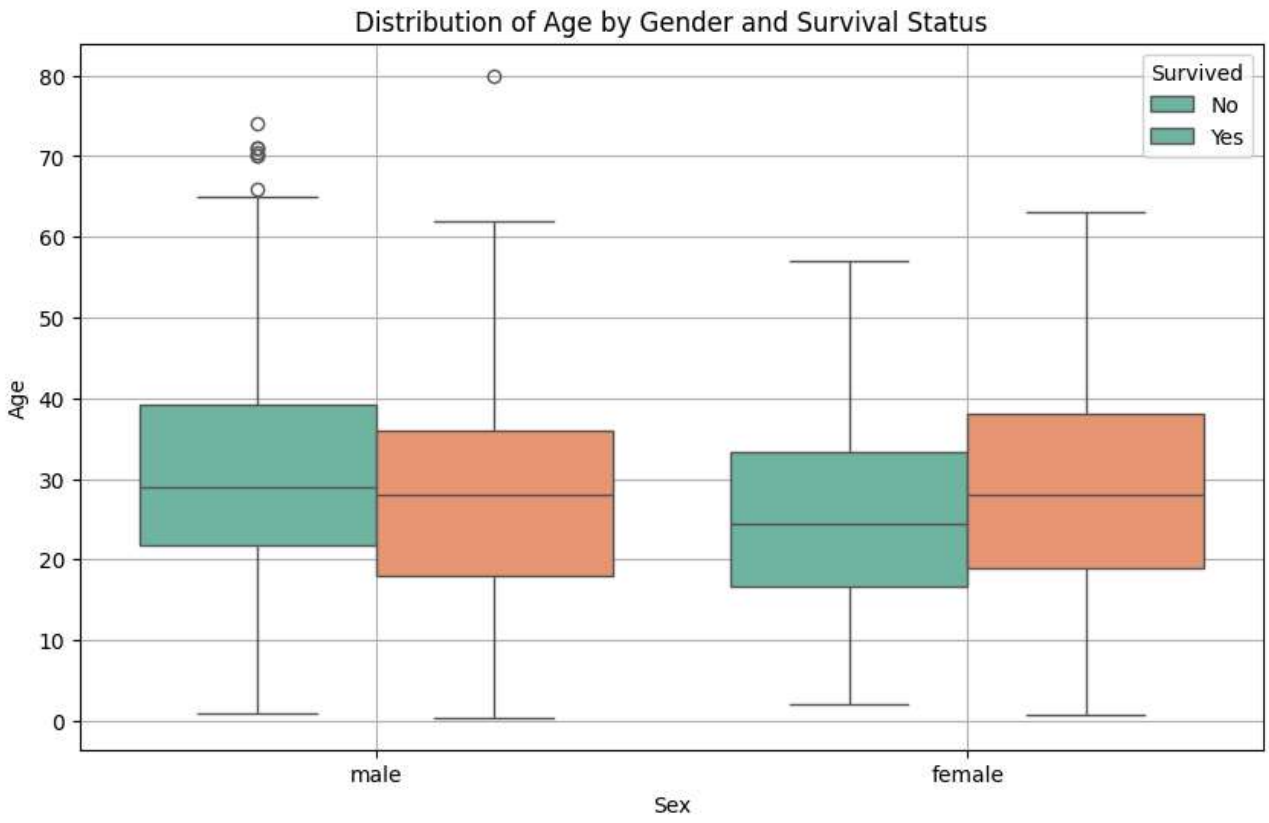
Next steps: Generate code with titanic | View recommended plots | New interactive sheet

```
# Check for missing values in 'age' and 'sex' columns
titanic[['age', 'sex', 'survived']].isnull().sum()
```

```
0
```

```
# Drop rows with missing values in 'age' or 'sex'
titanic_clean = titanic.dropna(subset=['age', 'sex', 'survived'])

# Plotting box plot
plt.figure(figsize=(10, 6))
sns.boxplot(data=titanic_clean, x='sex', y='age', hue='survived', palette='Set2')
plt.title('Distribution of Age by Gender and Survival Status')
plt.xlabel('Sex')
plt.ylabel('Age')
plt.legend(title='Survived', labels=['No', 'Yes'])
plt.grid(True)
plt.show()
```




```
Start coding or generate with AI.
```