

```
In [2]: 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
```

```
In [3]: sns.get_dataset_names()
```

```
Out[3]: ['anagrams',
'anscombe',
'attention',
'brain_networks',
'car_crashes',
'diamonds',
'dots',
'dowjones',
'exercise',
'flights',
'fmri',
'geyser',
'glue',
'healthexp',
'iris',
'mpg',
'penguins',
'planets',
'seaice',
'taxis',
'tips',
'titanic',
'anagrams',
'anagrams',
'anscombe',
'anscombe',
'attention',
'attention',
'brain_networks',
'brain_networks',
'car_crashes',
'car_crashes',
'diamonds',
'diamonds',
'diamonds',
'dots',
'dots',
'dowjones',
'dowjones',
'exercise',
'exercise',
'flights',
'flights',
'fmri',
'fmri',
'geyser',
'geyser',
'glue',
'glue',
'healthexp',
'healthexp',
'iris',
'iris',
'mpg',
'mpg',
'penguins',
'penguins',
'planets',
'planets',
'seaice',
'seaice',
'taxis',
'taxis',
'tips',
'tips',
'titanic',
'titanic',
'anagrams',
'anscombe',
'attention',
'brain_networks',
'car_crashes',
'diamonds',
'dots',
'dowjones',
'exercise',
'flights',
'fmri',
'geyser',
'glue',
'healthexp',
'iris',
'mpg',
'penguins',
'planets',
'seaice',
'taxis',
'tips',
'titanic']
```

```
In [4]: df=sns.load_dataset("titanic")
df
```

Out[4]:

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	deck	embark_town	alive	alone
0	0	3	male	22.0	1	0	7.2500	S	Third	man	True	NaN	Southampton	no	False
1	1	1	female	38.0	1	0	71.2833	C	First	woman	False	C	Cherbourg	yes	False
2	1	3	female	26.0	0	0	7.9250	S	Third	woman	False	NaN	Southampton	yes	True
3	1	1	female	35.0	1	0	53.1000	S	First	woman	False	C	Southampton	yes	False
4	0	3	male	35.0	0	0	8.0500	S	Third	man	True	NaN	Southampton	no	True
...
886	0	2	male	27.0	0	0	13.0000	S	Second	man	True	NaN	Southampton	no	True
887	1	1	female	19.0	0	0	30.0000	S	First	woman	False	B	Southampton	yes	True
888	0	3	female	NaN	1	2	23.4500	S	Third	woman	False	NaN	Southampton	no	False
889	1	1	male	26.0	0	0	30.0000	C	First	man	True	C	Cherbourg	yes	True
890	0	3	male	32.0	0	0	7.7500	Q	Third	man	True	NaN	Queenstown	no	True

891 rows × 15 columns

```
In [5]: df.shape
```

Out[5]: (891, 15)

```
In [6]: df.head()
```

Out[6]:

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	deck	embark_town	alive	alone
0	0	3	male	22.0	1	0	7.2500	S	Third	man	True	NaN	Southampton	no	False
1	1	1	female	38.0	1	0	71.2833	C	First	woman	False	C	Cherbourg	yes	False
2	1	3	female	26.0	0	0	7.9250	S	Third	woman	False	NaN	Southampton	yes	True
3	1	1	female	35.0	1	0	53.1000	S	First	woman	False	C	Southampton	yes	False
4	0	3	male	35.0	0	0	8.0500	S	Third	man	True	NaN	Southampton	no	True

```
In [7]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 15 columns):
#   Column      Non-Null Count  Dtype
---  -
0   survived    891 non-null    int64
1   pclass      891 non-null    int64
2   sex         891 non-null    object
3   age         714 non-null    float64
4   sibsp       891 non-null    int64
5   parch       891 non-null    int64
6   fare        891 non-null    float64
7   embarked    889 non-null    object
8   class       891 non-null    category
9   who         891 non-null    object
10  adult_male  891 non-null    bool
11  deck        203 non-null    category
12  embark_town 889 non-null    object
13  alive       891 non-null    object
14  alone       891 non-null    bool
dtypes: bool(2), category(2), float64(2), int64(4), object(5)
memory usage: 80.7+ KB
```

```
In [8]: df.describe()
```

Out[8]:

	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

```
In [9]: df.isna().sum()
```

```
Out[9]: survived      0
pclass      0
sex         0
age        177
sibsp      0
parch      0
fare       0
embarked    2
class      0
who        0
adult_male  0
deck       688
embark_town 2
alive      0
alone      0
dtype: int64
```

```
In [10]: df['age'] = df['age'].fillna(df['age'].mean())
```

```
In [11]: df.isna().sum()
```

```
Out[11]: survived      0
pclass      0
sex         0
age         0
sibsp      0
parch      0
fare       0
embarked    2
class      0
who        0
adult_male  0
deck       688
embark_town 2
alive      0
alone      0
dtype: int64
```

```
In [12]: def fun1(value):
         if (value == "male"):
             return 1
         else:
             return 0
```

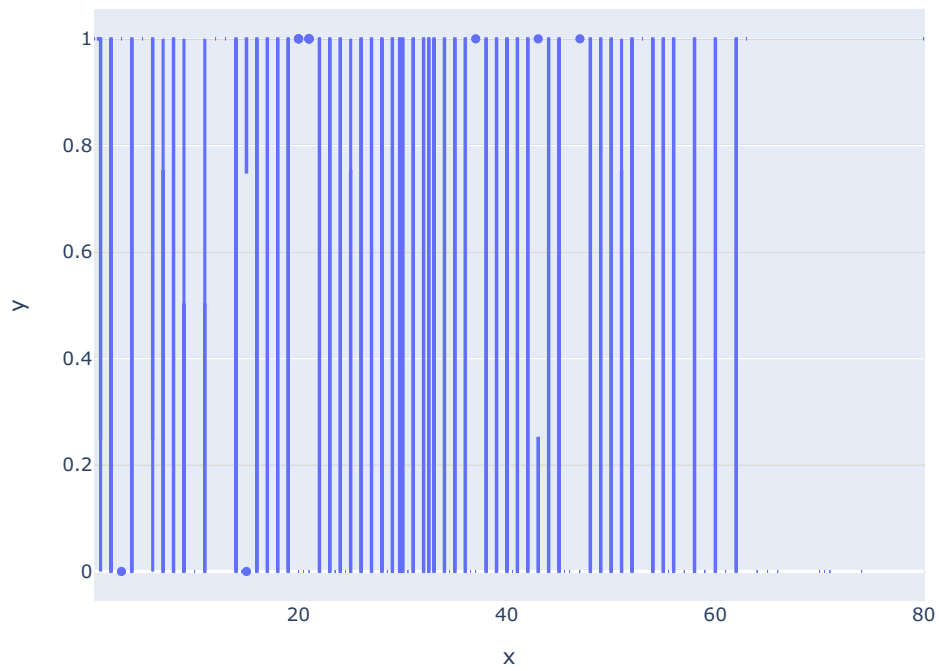
```
In [13]: def fun2(value):
         if (value == 's'):
             return 0
         elif (value == 'c'):
             return 1
         elif (value == 'q'):
             return 2
         else:
             return 0
```

```
In [14]: df['sex'] = df['sex'].apply(fun1)
df['embarked'] = df['embarked'].apply(fun2)
df = df.drop('deck', axis=1)
```

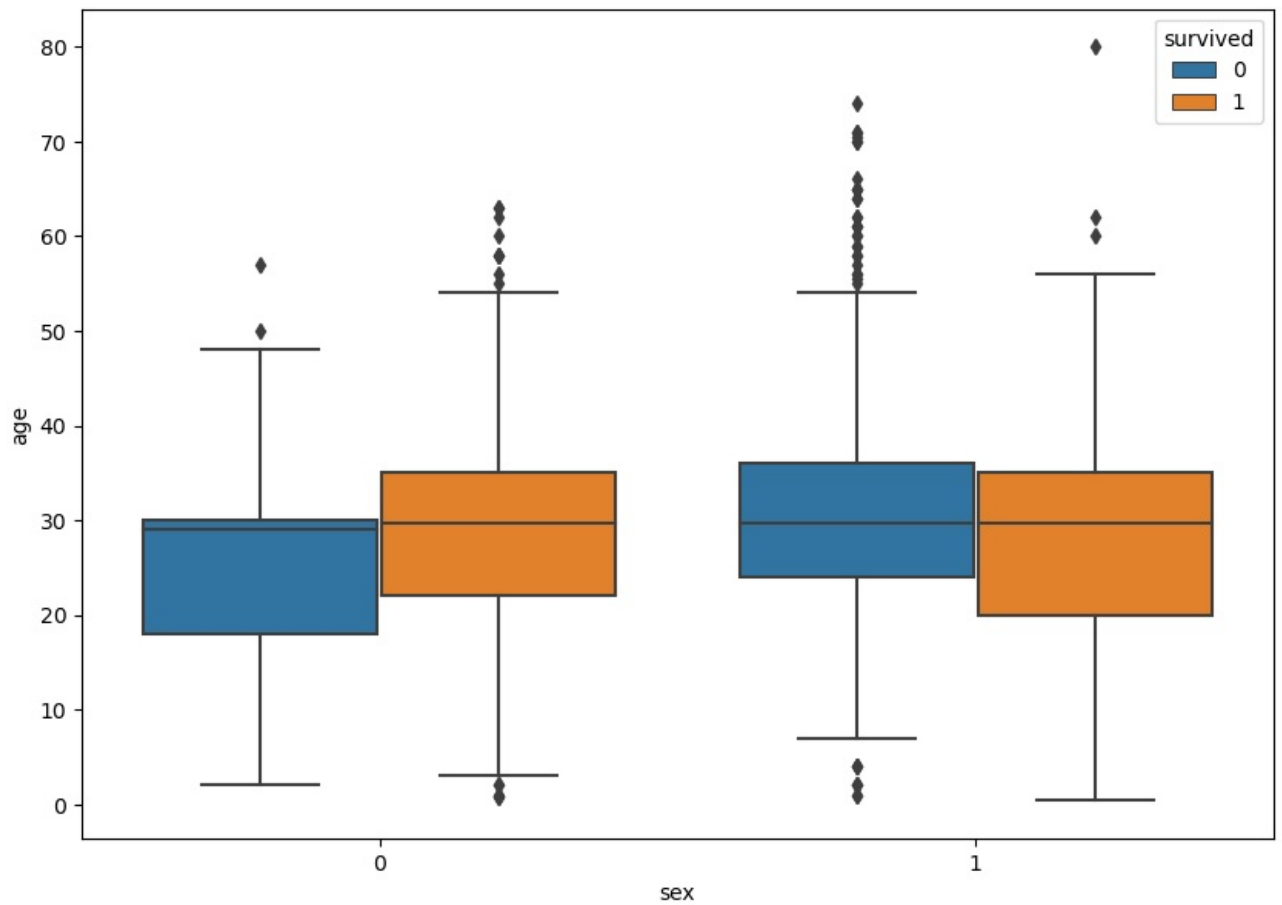
```
In [15]: df.shape
```

```
Out[15]: (891, 14)
```

```
In [16]: px.box(df['sex'], df['age'], df['survived'])
```



```
In [18]: plt.figure(figsize=(10,7))
sns.boxplot(x='sex', y='age',data=df,hue="survived")
plt.show()
```



```
In [ ]: # Kaushal Taware TE 13354
```