Data Visualization I

1. Use the inbuilt dataset 'titanic'. The dataset contains 891 rows and contains information about the passengers who boarded the unfortunate Titanic ship. Use the Seaborn library to see if we can find any patterns in the data.
2. Write a code to check how the price of the ticket (column name: 'fare') for each passenger is distributed by plotting a histogram

In [18]:

```
import seaborn as sns
import pandas as pd

titanic = sns.load_dataset("titanic")

titanic
```

Out[18]:

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult
0	0	3	male	22.0	1	0	7.2500	S	Third	man	
1	1	1	female	38.0	1	0	71.2833	C	First	woman	
2	1	3	female	26.0	0	0	7.9250	S	Third	woman	
3	1	1	female	35.0	1	0	53.1000	S	First	woman	
4	0	3	male	35.0	0	0	8.0500	S	Third	man	
...	...	...	...	...	...	...	...	...	...	...	
886	0	2	male	27.0	0	0	13.0000	S	Second	man	
887	1	1	female	19.0	0	0	30.0000	S	First	woman	
888	0	3	female	NaN	1	2	23.4500	S	Third	woman	
889	1	1	male	26.0	0	0	30.0000	C	First	man	
890	0	3	male	32.0	0	0	7.7500	Q	Third	man	

891 rows × 15 columns

In [19]:

```
titanic.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 [20]:

```
x=titanic["fare"]
x
```

Out[20]:

```
0       7.2500
1      71.2833
2       7.9250
3      53.1000
4       8.0500
        ...   
886    13.0000
887    30.0000
888    23.4500
889    30.0000
890     7.7500
Name: fare, Length: 891, dtype: float64
```

In [21]:

```
#titanic.iloc[:,"fare"]
```

In [22]:

```
titanic.describe()
```

Out[22]:

	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 [23]:

```
#First Part
```

Data Cleanup

In [24]:

```
#inform us about empty fileds in column
titanic.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 [25]:

```
#Dropping the not required columns
titanic_cleaned = titanic.drop(['pclass','embarked','deck','embark_town'],axis=1)
titanic_cleaned.head(15)
```

Out[25]:

	survived	sex	age	sibsp	parch	fare	class	who	adult_male	alive	alone
0	0	male	22.0	1	0	7.2500	Third	man	True	no	False
1	1	female	38.0	1	0	71.2833	First	woman	False	yes	False
2	1	female	26.0	0	0	7.9250	Third	woman	False	yes	True
3	1	female	35.0	1	0	53.1000	First	woman	False	yes	False
4	0	male	35.0	0	0	8.0500	Third	man	True	no	True
5	0	male	NaN	0	0	8.4583	Third	man	True	no	True
6	0	male	54.0	0	0	51.8625	First	man	True	no	True
7	0	male	2.0	3	1	21.0750	Third	child	False	no	False
8	1	female	27.0	0	2	11.1333	Third	woman	False	yes	False
9	1	female	14.0	1	0	30.0708	Second	child	False	yes	False
10	1	female	4.0	1	1	16.7000	Third	child	False	yes	False
11	1	female	58.0	0	0	26.5500	First	woman	False	yes	True
12	0	male	20.0	0	0	8.0500	Third	man	True	no	True
13	0	male	39.0	1	5	31.2750	Third	man	True	no	False
14	0	female	14.0	0	0	7.8542	Third	child	False	no	True

In [26]:

```
titanic_cleaned.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 11 columns):
 #   Column      Non-Null Count  Dtype   
---  ------      --------------  -----   
 0   survived    891 non-null    int64   
 1   sex         891 non-null    object  
 2   age         714 non-null    float64 
 3   sibsp       891 non-null    int64   
 4   parch       891 non-null    int64   
 5   fare        891 non-null    float64 
 6   class       891 non-null    category
 7   who         891 non-null    object  
 8   adult_male  891 non-null    bool    
 9   alive       891 non-null    object  
 10  alone       891 non-null    bool    
dtypes: bool(2), category(1), float64(2), int64(3), object(3)
memory usage: 58.6+ KB
```

In [27]:

```
titanic_cleaned.isnull().sum()
```

Out[27]:

```
survived        0
sex             0
age           177
sibsp           0
parch           0
fare            0
class           0
who             0
adult_male      0
alive           0
alone           0
dtype: int64
```

In [28]:

```
titanic_cleaned.corr(method='pearson')
```

Out[28]:

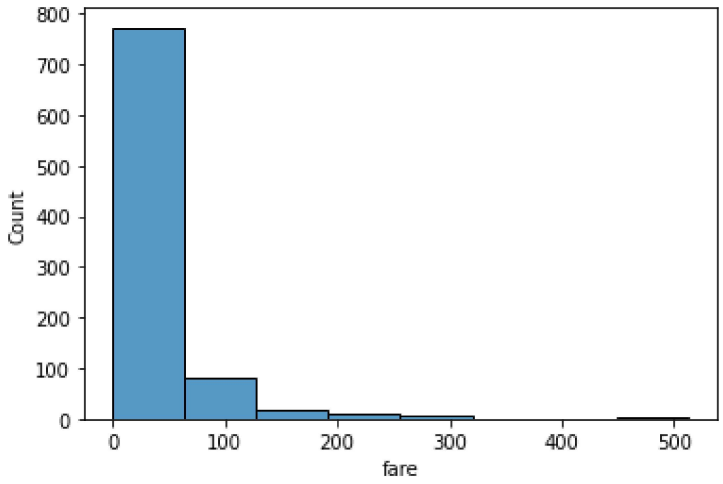
	survived	age	sibsp	parch	fare	adult_male	alone
survived	1.000000	-0.077221	-0.035322	0.081629	0.257307	-0.557080	-0.203367
age	-0.077221	1.000000	-0.308247	-0.189119	0.096067	0.280328	0.198270
sibsp	-0.035322	-0.308247	1.000000	0.414838	0.159651	-0.253586	-0.584471
parch	0.081629	-0.189119	0.414838	1.000000	0.216225	-0.349943	-0.583398
fare	0.257307	0.096067	0.159651	0.216225	1.000000	-0.182024	-0.271832
adult_male	-0.557080	0.280328	-0.253586	-0.349943	-0.182024	1.000000	0.404744
alone	-0.203367	0.198270	-0.584471	-0.583398	-0.271832	0.404744	1.000000

In [29]:

```
sns.histplot(data=titanic,x="fare",bins=8)
```

Out[29]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fef9a15b710>
```



In [30]:

```
sns.histplot(data=titanic,x="fare",binwidth=10)
```

Out[30]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fef9a14d8d0>
```