8. Data Visualization I

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

In [1]:

```
import seaborn as sns
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import warnings
warnings.filterwarnings('ignore')
```

```
In [2]: sns.get_dataset_names()

Out[2]: ['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',
         '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 [3]:
```
data=sns.load_dataset('titanic')
```

In [4]: data

Out[4]:
```
survivedpclasssexagesibspparchfareembarkedclasswhoadult_mal
______________________________________________________________________________
        0    0     3     male     22.0   1      0      7.2500   S      Third    man      Tru
        1    1     1     female   38.0   1      0      71.2833  C      First    woman    Fals
        2    1     3     female   26.0   0      0      7.9250   S      Third    woman    Fals
3    1 1 female 35.0 1 0 53.1000 S First woman Fals 4 0 3 male 35.0 0 0 8.0500 S Third man Tru ... ... ... ... ... ... ... ... ... ... ...
  886        0     2     male     27.0   0      0      13.0000  S      Second   man      Tru
  887        1     1     female   19.0   0      0      30.0000  S      First    woman    Fals
  888        03femaleNaN1223.4500SThirdwomanFals
  889        1     1     male     26.0   0      0      30.0000  C      First    man      Tru
  890        0     3     male     32.0   0      0      7.7500   Q      Third    man      Tru
```

Out[5]:

891 rows × 15 columns

In [5]:

```
data.head()
```

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

In [6]: data.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 [7]:
```
data.isnull().sum()
```

Out[7]:
```
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 [8]:
```
data['age'].mean()
```

Out[8]: 29.69911764705882

In [9]:
```
data['age'] = data['age'].fillna(np.mean(data['age']))
```

In [10]:
```
data['deck'].mode()
```

Out[10]:
```
0    C
Name: deck, dtype: category
                             Categories (7, object): ['A', 'B', 'C', 'D', 'E', 'F', 'G']
```

In [11]:

```
data['deck'] = data['deck'].fillna(data['deck'].mode()[0])
```

In [12]:

```
data['embarked'].mode()
```

Out[12]:

```
0    S
Name: embarked, dtype: object
```

In [13]:

```
data['embarked'] = data['embarked'].fillna(data['embarked'].mode()[0])
```

In [14]:

```
data['embark_town'].mode()
```

Out[14]:

```
0    Southampton
Name: embark_town, dtype: object
```

In [15]:

```
data['embark_town'] = data['embark_town'].fillna(data['embark_town'].mode()[0])
```

In [16]:

```
data.isnull().sum()
```

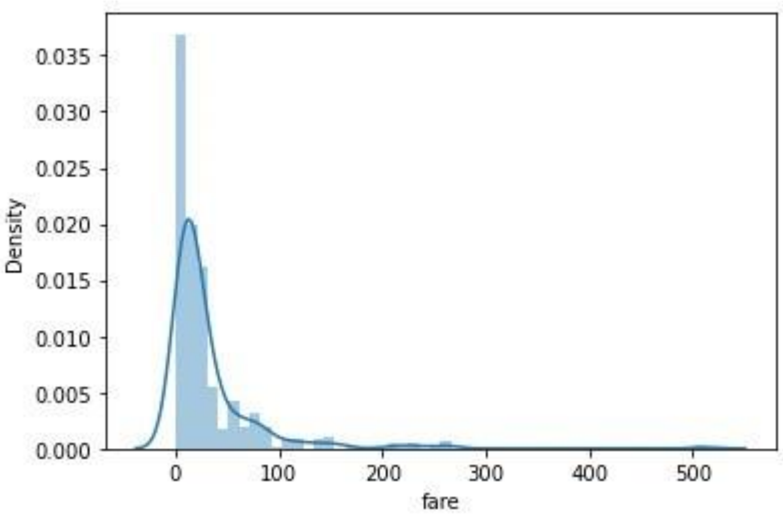
Out[16]:

```
survived        0
pclass          0
sex         0 age       0
sibsp        0 parch
0 fare        0 embarked
0 class        0 who
0 adult_male    0 deck
0 embark_town    0 alive
0 alone        0 dtype:
int64
```

In [17]:

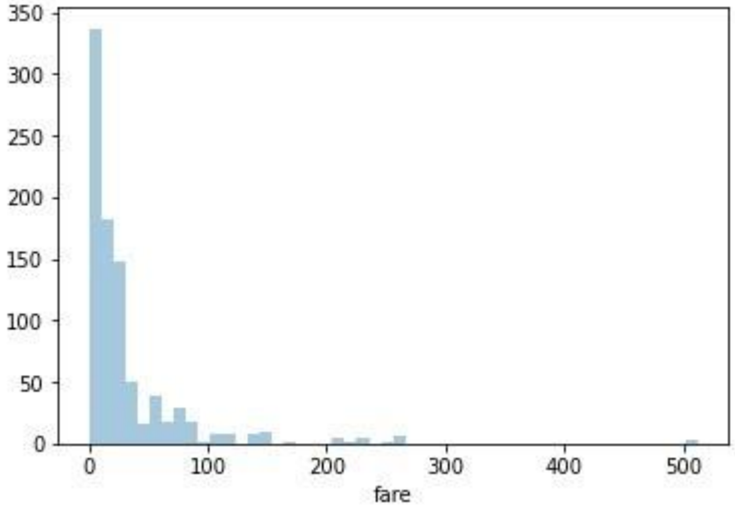
```
sns.distplot(data['fare'])
```

Out[17]: <AxesSubplot:xlabel='fare', ylabel='Density'>



In [18]:
```
sns.distplot(data['fare'], kde=False)
```

Out[18]:
```
<AxesSubplot:xlabel='fare'>
```

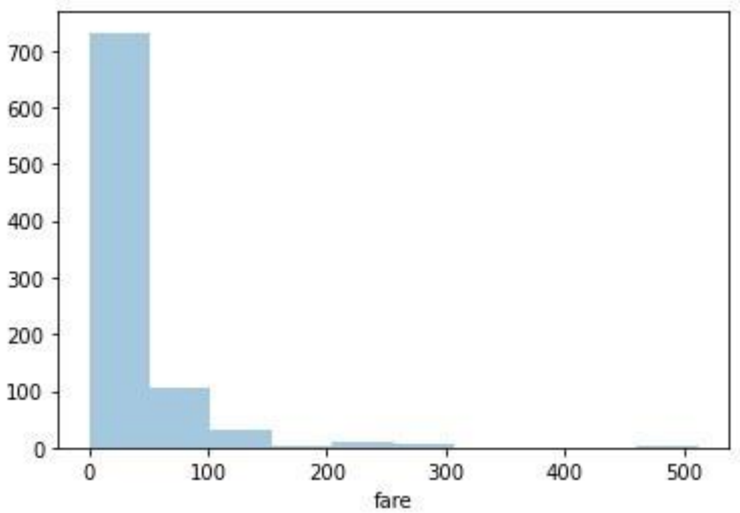


In [19]:

```
sns.distplot(data['fare'], kde=False, bins=10)
```

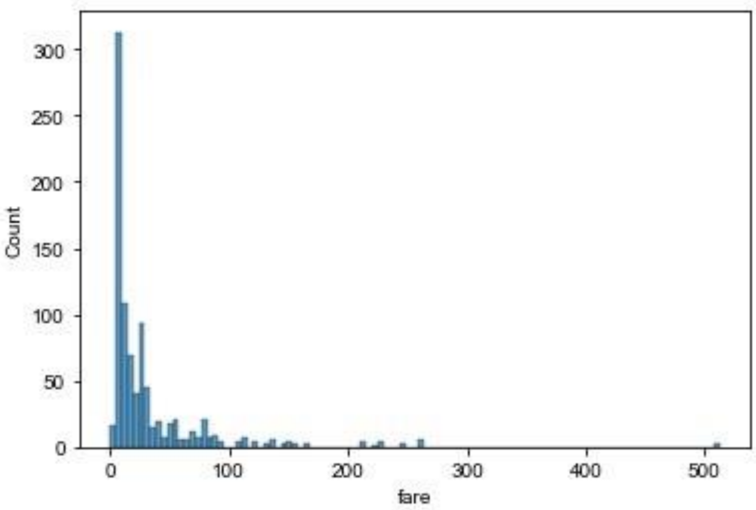
Out[19]:

```
<AxesSubplot:xlabel='fare'>
```



In [20]:

```
sns.histplot(x='fare' , data=data)
sns.set (rc={'figure.figsize':(5,5)})
```



In [ ]:

In [ ]:

In [ ]:

In [ ]: