WEEK-2 Assignment

Mir Mohammed Zain - 20BDS0211

Q1. Download the dataset: **Dataset**

```
# downloaded the dataset
```

```
# importing libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

Q2. Load the dataset.

```
data = pd.read_csv('titanic.csv')
```

```
data
```

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

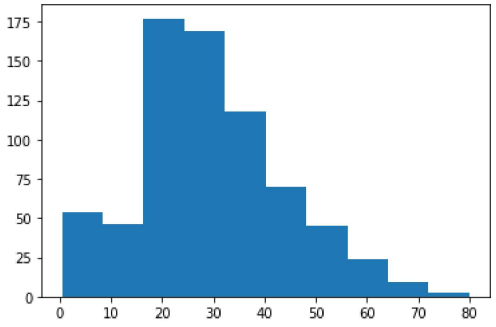
891 rows × 15 columns

**Q3. Perform Below Visualizations.

● Univariate Analysis**

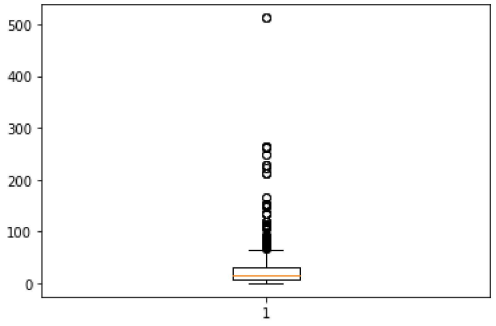
```
plt.hist(data['age'])
```

```
D:\anaconda\lib\site-packages\numpy\lib\histograms.py:839: RuntimeWarning: invalid value encountered ir
  keep = (tmp_a >= first_edge)
D:\anaconda\lib\site-packages\numpy\lib\histograms.py:840: RuntimeWarning: invalid value encountered ir
  keep &= (tmp_a <= last_edge)
(array([ 54.,  46., 177., 169., 118.,  70.,  45.,  24.,   9.,   2.]),
 array([ 0.42 ,  8.378, 16.336, 24.294, 32.252, 40.21 , 48.168, 56.126,
        64.084, 72.042, 80.   ]),
 <a list of 10 Patch objects>)
```



```
plt.boxplot(data['fare'])
```

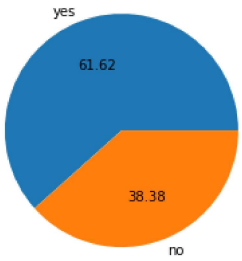
```
{'whiskers': [<matplotlib.lines.Line2D at 0x258fea10d00>,
  <matplotlib.lines.Line2D at 0x258fea210a0>],
 'caps': [<matplotlib.lines.Line2D at 0x258fea21400>,
  <matplotlib.lines.Line2D at 0x258fea21760>],
 'boxes': [<matplotlib.lines.Line2D at 0x258fea109a0>],
 'medians': [<matplotlib.lines.Line2D at 0x258fea21ac0>],
 'fliers': [<matplotlib.lines.Line2D at 0x258fea21dc0>],
 'means': []}
```




```
living = data['alive'].value_counts()
```

```
plt.pie(living,autopct='%.2f',labels = ['yes','no'])
```

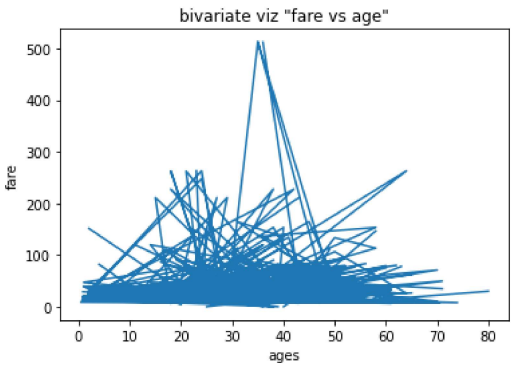
```
([<matplotlib.patches.Wedge at 0x258fea6efa0>,
  <matplotlib.patches.Wedge at 0x258fea7c4c0>],
 [Text(-0.3925749350994583, 1.0275626113924428, 'yes'),
  Text(0.3925750313068116, -1.0275625746369201, 'no')],
 [Text(-0.21413178278152267, 0.5604886971231505, '61.62'),
  Text(0.21413183525826085, -0.5604886770746836, '38.38')])
```



● Bi - Variate Analysis

```
plt.plot(data['age'],data['fare'])
plt.xlabel('ages')
plt.ylabel('fare')
plt.title('bivariate viz "fare vs age"')
```

```
Text(0.5, 1.0, 'bivariate viz "fare vs age"')
```

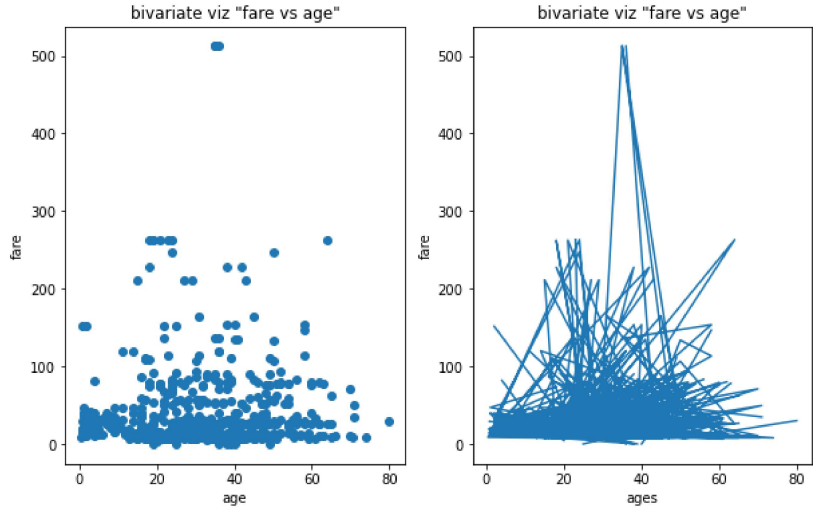



```
plt.figure(figsize=(10,6))

plt.subplot(1,2,1)
plt.scatter(data['age'],data['fare'])
plt.xlabel('age')
plt.ylabel('fare')
plt.title('bivariate viz "fare vs age"')
```

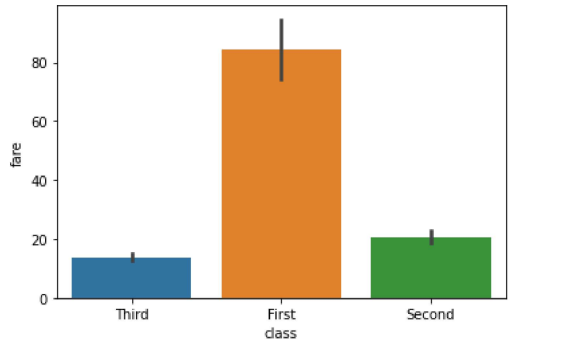
```
plt.subplot(1,2,2)
plt.plot(data['age'],data['fare'])
plt.xlabel('ages')
plt.ylabel('fare')
plt.title('bivariate viz "fare vs age"')
```

```
Text(0.5, 1.0, 'bivariate viz "fare vs age"')
```




```
sns.barplot(data['class'],data['fare'])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x258fecd84c0>
```

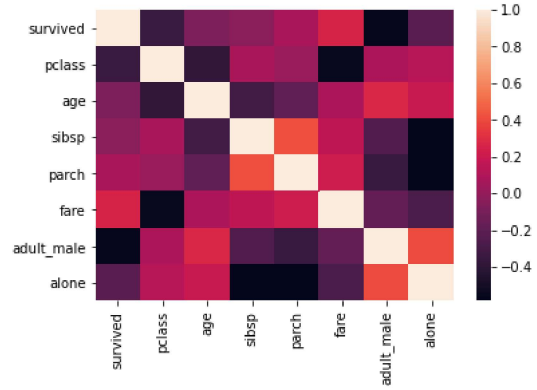


● Multi - Variate Analysis

```
heat = data.corr()
```

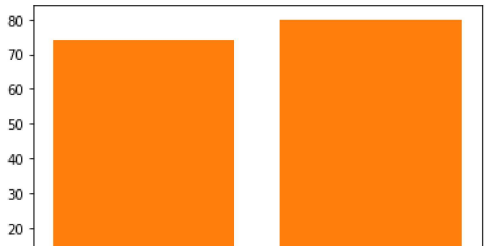
```
sns.heatmap(heat)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x258febe5880>
```



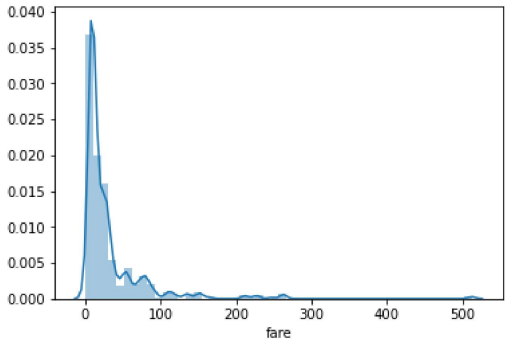

```
plt.bar(data['survived'],data['pclass'])
plt.bar(data['survived'],data['age'])
```

```
<BarContainer object of 891 artists>
```



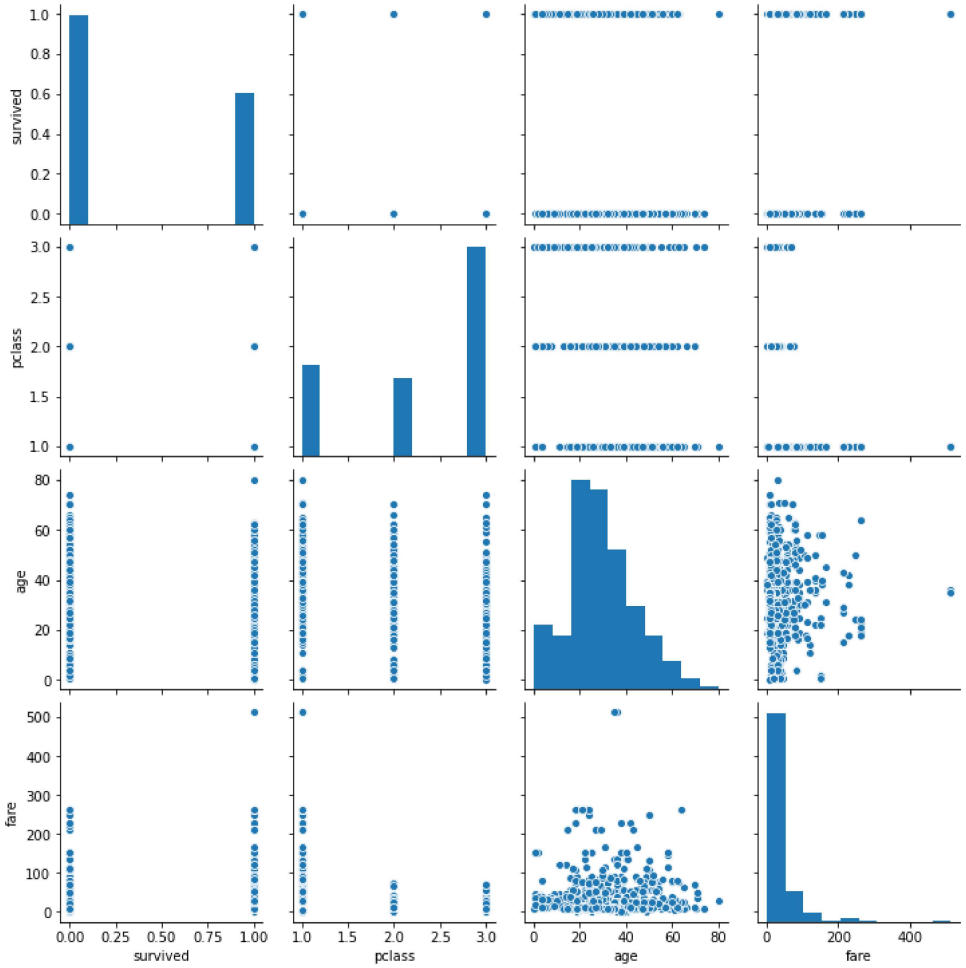
```
sns.distplot(data.fare)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x25880098ca0>
```




```
sns.pairplot(data=data[['survived','pclass','age','fare']])
```

```
<seaborn.axisgrid.PairGrid at 0x25882352f40>
```



Q4. Perform descriptive statistics on the dataset.

```
print(data.describe())
print('median')
print(data.median())
print('mode')
print(data.mode())
```

```
print(data.kurt())
print('printing quartile')
quantile=data.quantile(q=[0.75,0.25])
print(quantile)
print(quantile.iloc[0])
print(data.quantile(0.5))
print(quantile.iloc[1])
```

```
            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
median
survived       0.0000
pclass         3.0000
age           28.0000
sibsp          0.0000
parch          0.0000
fare          14.4542
adult_male     1.0000
alone          1.0000
dtype: float64
mode
   survived  pclass   sex   age  sibsp  parch  fare embarked  class  who  \
0         0       3  male  24.0      0      0  8.05        S  Third  man

   adult_male deck  embark_town alive  alone
0        True    C  Southampton    no   True
survived      -1.775005
pclass        -1.280015
age            0.178274
sibsp         17.880420
parch          9.778125
fare          33.398141
adult_male    -1.827345
alone         -1.827345
dtype: float64
printing quartile
      survived  pclass     age  sibsp  parch     fare  adult_male  alone
0.75       1.0     3.0  38.000    1.0    0.0  31.0000         1.0    1.0
0.25       0.0     2.0  20.125    0.0    0.0   7.9104         0.0    0.0
survived       1.0
pclass         3.0
age           38.0
sibsp          1.0
parch          0.0
fare          31.0
adult_male     1.0
alone          1.0
Name: 0.75, dtype: float64
survived       0.0000
pclass         3.0000
age           28.0000
sibsp          0.0000
parch          0.0000
fare          14.4542
adult_male     1.0000
alone          1.0000
Name: 0.5, dtype: float64
survived       0.0000
pclass         2.0000
```

Q5. Handle the Missing values.

```
data.isnull()
```

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	deck	embark_town	alive	alone
0	False	False	False	False	False	False	False	False	False	False	False	True	False	False	False
1	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False	False	True	False	False	False
3	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False

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

```
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
```

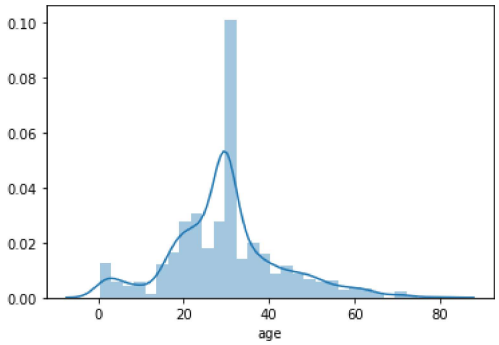
```
data['age'].fillna(data['age'].mean(),inplace=True)
```

```
data['alive'].value_counts()
```

```
no     549
yes    342
Name: alive, dtype: int64
```

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

```
<matplotlib.axes._subplots.AxesSubplot at 0x2588032fee0>
```



Q6. Find the outliers and replace the outliers

```
plt.boxplot(data['fare'])
```

```
{'whiskers': [<matplotlib.lines.Line2D at 0x25882e1b220>,
```

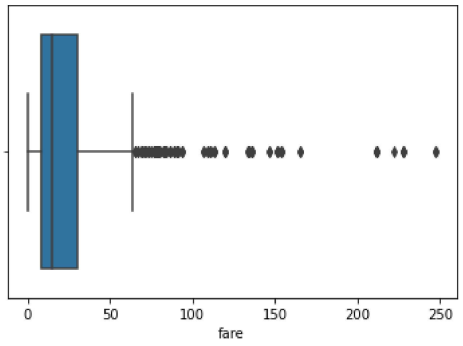
```
perc99=data.fare.quantile(0.99)
perc99
```

```
249.00622000000033
```

```
 'fliers': [<matplotlib.lines.Line2D at 0x25882e252e0>],
```

```
data=data[data.fare<=perc99]
sns.boxplot(data.fare)
# the value 'above 500' is being removed
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x25882e323d0>
```



Q7. Check for Categorical columns and perform encoding.

```
data.head()
```

	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

```
from sklearn.preprocessing import LabelEncoder


#LabelEncoder

le = LabelEncoder()


data.alone = le.fit_transform(data.alone)
data.sex = le.fit_transform(data.sex)
data['class'] = le.fit_transform(data['class'])
data.who = le.fit_transform(data.who)
data.alive = le.fit_transform(data.alive)
data.adult_male = le.fit_transform(data.adult_male)
data.deck = le.fit_transform(data.adult_male)
```

```
D:\anaconda\lib\site-packages\pandas\core\generic.py:5303: 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
  self[name] = value
<ipython-input-29-9e27b9a3781e>:3: 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
  data['class'] = le.fit_transform(data['class'])
```

```
data['alone'].head()
```

```
0    0
1    0
2    1
3    0
4    1
Name: alone, dtype: int64
```

```
onehot_data = pd.get_dummies(data,columns = ['embark_town'])
```

```
onehot_data.head()
```

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	deck	alive	alone	embark_town_Cherbourg	em
0	0	3	1	22.0	1	0	7.2500	S	2	1	1	1	0	0	0	
1	1	1	0	38.0	1	0	71.2833	C	0	2	0	0	1	0	1	
2	1	3	0	26.0	0	0	7.9250	S	2	2	0	0	1	1	0	
3	1	1	0	35.0	1	0	53.1000	S	0	2	0	0	1	0	0	
4	0	3	1	35.0	0	0	8.0500	S	2	1	1	1	0	1	0	

Q8. Split the data into dependent and independent variables.

```
y = onehot_data['alive']
```

```
X=onehot_data.drop(columns=['alive','embarked','survived'],axis=1)
```

```
X.head()
```

	pclass	sex	age	sibsp	parch	fare	class	who	adult_male	deck	alone	embark_town_Cherbourg	embark_town_Queenstown	emba
0	3	1	22.0	1	0	7.2500	2	1	1	1	0	0	0	
1	1	0	38.0	1	0	71.2833	0	2	0	0	0	1	0	
2	3	0	26.0	0	0	7.9250	2	2	0	0	1	0	0	
3	1	0	35.0	1	0	53.1000	0	2	0	0	0	0	0	
4	3	1	35.0	0	0	8.0500	2	1	1	1	1	0	0	

Q9. Scale the independent variables

```
name = X.columns
name
```

```
Index(['pclass', 'sex', 'age', 'sibsp', 'parch', 'fare', 'class', 'who',
       'adult_male', 'deck', 'alone', 'embark_town_Cherbourg',
       'embark_town_Queenstown', 'embark_town_Southampton'],
      dtype='object')
```

```
from sklearn.preprocessing import MinMaxScaler
```

```
#MinMaxScaler
```

```
scale=MinMaxScaler()
```

```
x_scaled = scale.fit_transform(X)
x_scaled
```

```
array([[1.        , 1.        , 0.27117366, ..., 0.        , 0.        ,
        1.        ],
       [0.        , 0.        , 0.4722292 , ..., 1.        , 0.        ,
        0.        ],
       [1.        , 0.        , 0.32143755, ..., 0.        , 0.        ,
        1.        ],
       ...,
       [1.        , 0.        , 0.36792055, ..., 0.        , 0.        ,
        1.        ],
       [0.        , 1.        , 0.32143755, ..., 1.        , 0.        ,
        0.        ],
       [1.        , 1.        , 0.39683338, ..., 0.        , 1.        ,
        0.        ]])
```

```
X=pd.DataFrame(x_scaled,columns=name)
```

```
X
```

	pclass	sex	age	sibsp	parch	fare	class	who	adult_male	deck	alone	embark_town_Cherbourg	embark_town_Queens
0	1.0	1.0	0.271174	0.125	0.000000	0.029290	1.0	0.5	1.0	1.0	0.0	0.0	
1	0.0	0.0	0.472229	0.125	0.000000	0.287989	0.0	1.0	0.0	0.0	0.0	1.0	
2	1.0	0.0	0.321438	0.000	0.000000	0.032018	1.0	1.0	0.0	0.0	1.0	0.0	
3	0.0	0.0	0.434531	0.125	0.000000	0.214527	0.0	1.0	0.0	0.0	0.0	0.0	
4	1.0	1.0	0.434531	0.000	0.000000	0.032523	1.0	0.5	1.0	1.0	1.0	0.0	
...	...	...	...	...	...	...	...	...	...	...	...	...	
877	0.5	1.0	0.334004	0.000	0.000000	0.052521	0.5	0.5	1.0	1.0	1.0	0.0	
878	0.0	0.0	0.233476	0.000	0.000000	0.121202	0.0	1.0	0.0	0.0	1.0	0.0	
879	1.0	0.0	0.367921	0.125	0.333333	0.094740	1.0	1.0	0.0	0.0	0.0	0.0	
880	0.0	1.0	0.321438	0.000	0.000000	0.121202	0.0	0.5	1.0	1.0	1.0	1.0	
881	1.0	1.0	0.396833	0.000	0.000000	0.031310	1.0	0.5	1.0	1.0	1.0	0.0	

882 rows × 14 columns

Q10. Split the data into training and testing

```
from sklearn.model_selection import train_test_split

x_train,x_test,y_train,y_test = train_test_split(X,y,test_size = 20,random_state = 0)

print(x_train.head())
print(y_train.head())
print(x_test.head())
print(y_test.head())
```

```
     pclass  sex       age  sibsp     parch      fare  class  who  adult_male  \
686     1.0  1.0  0.308872   0.00  0.000000  0.029189    1.0  0.5         1.0
14      1.0  0.0  0.170646   0.00  0.000000  0.031731    1.0  0.0         0.0
267     0.0  1.0  0.367921   0.00  0.000000  0.125242    0.0  0.5         1.0
145     1.0  0.0  0.107816   0.25  0.333333  0.138877    1.0  0.0         0.0
667     0.5  1.0  0.384267   0.00  0.000000  0.052521    0.5  0.5         1.0

     deck  alone  embark_town_Cherbourg  embark_town_Queenstown  \
686   1.0    1.0                    1.0                     0.0
14    0.0    1.0                    0.0                     0.0
267   1.0    1.0                    0.0                     0.0
145   0.0    0.0                    0.0                     0.0
667   1.0    1.0                    0.0                     0.0

     embark_town_Southampton
686                      0.0
14                       1.0
267                      1.0
145                      1.0
667                      1.0
693    0
14     0
270    0
147    0
673    1
Name: alive, dtype: int32
     pclass  sex       age  sibsp  parch      fare  class  who  adult_male  \
150     1.0  1.0  0.692134    0.0    0.0  0.032523    1.0  0.5         1.0
406     1.0  1.0  0.367921    0.0    0.0  0.027708    1.0  0.5         1.0
513     1.0  1.0  0.396833    0.0    0.0  0.031900    1.0  0.5         1.0
101     1.0  1.0  0.409399    0.0    0.0  0.034964    1.0  0.5         1.0
584     1.0  1.0  0.434531    0.0    0.0  0.028785    1.0  0.5         1.0

     deck  alone  embark_town_Cherbourg  embark_town_Queenstown  \
150   1.0    1.0                    0.0                     0.0
406   1.0    1.0                    0.0                     1.0
513   1.0    1.0                    0.0                     0.0
101   1.0    1.0                    0.0                     0.0
584   1.0    1.0                    0.0                     0.0

     embark_town_Southampton
150                      1.0
406                      0.0
513                      1.0
101                      1.0
584                      1.0
152    0
411    0
```

```
519    0
103    0
590    0
Name: alive, dtype: int32
```