Understand the data

Clean the data

Find a relationship between data

In [4]:

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

Explore the Data

In [6]:

```
kashti = sns.load_dataset('titanic')
kashti
```

Out[6]:

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adu
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 [7]:

```
kashti.describe()
```

Out[7]:

	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 [8]:

```
kashti.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 [9]:

```
kashti.head()
```

Out[9]:

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

In [10]:

```
kashti.shape
```

Out[10]:

```
(891, 15)
```

Unique Values

In [12]:

```
kashti.nunique()
```

Out[12]:

```
survived         2
pclass           3
sex              2
age             88
sibsp            7
parch            7
fare           248
embarked         3
class            3
who              3
adult_male       2
deck             7
embark_town      3
alive            2
alone            2
dtype: int64
```

In [13]:

```
kashti.columns
```

Out[13]:

```
Index(['survived', 'pclass', 'sex', 'age', 'sibsp', 'parch', 'fare',
       'embarked', 'class', 'who', 'adult_male', 'deck', 'embark_town',
       'alive', 'alone'],
      dtype='object')
```

In [14]:

```
kashti['sex'].unique()
```

Out[14]:

```
array(['male', 'female'], dtype=object)
```

In [15]:

```
kashti['fare'].unique()
```

```
Out[15]: array([  7.25  ,  71.2833,   7.925 ,  53.1   ,   8.05  ,   8.4583,
               51.8625,  21.075 ,  11.1333,  30.0708,  16.7   ,  26.55  ,
               31.275 ,   7.8542,  16.    ,  29.125 ,  13.    ,  18.    ,
                7.225 ,  26.    ,   8.0292,  35.5   ,  31.3875, 263.    ,
                7.8792,   7.8958,  27.7208, 146.5208,   7.75  ,  10.5   ,
               82.1708,  52.    ,   7.2292,  11.2417,   9.475 ,  21.    ,
               41.5792,  15.5   ,  21.6792,  17.8   ,  39.6875,   7.8   ,
               76.7292,  61.9792,  27.75  ,  46.9   ,  80.    ,  83.475 ,
               27.9   ,  15.2458,   8.1583,   8.6625,  73.5   ,  14.4542,
               56.4958,   7.65  ,  29.    ,  12.475 ,   9.    ,   9.5   ,
                7.7875,  47.1   ,  15.85  ,  34.375 ,  61.175 ,  20.575 ,
               34.6542,  63.3583,  23.    ,  77.2875,   8.6542,   7.775 ,
               24.15  ,   9.825 ,  14.4583, 247.5208,   7.1417,  22.3583,
                6.975 ,   7.05  ,  14.5   ,  15.0458,  26.2833,   9.2167,
               79.2   ,   6.75  ,  11.5   ,  36.75  ,   7.7958,  12.525 ,
               66.6   ,   7.3125,  61.3792,   7.7333,  69.55  ,  16.1   ,
               15.75  ,  20.525 ,  55.    ,  25.925 ,  33.5   ,  30.6958,
               25.4667,  28.7125,   0.    ,  15.05  ,  39.    ,  22.025 ,
               50.    ,   8.4042,   6.4958,  10.4625,  18.7875,  31.    ,
              113.275 ,  27.    ,  76.2917,  90.    ,   9.35  ,  13.5   ,
                7.55  ,  26.25  ,  12.275 ,   7.125 ,  52.5542,  20.2125,
               86.5   , 512.3292,  79.65  , 153.4625, 135.6333,  19.5   ,
               29.7   ,  77.9583,  20.25  ,  78.85  ,  91.0792,  12.875 ,
                8.85  , 151.55  ,  30.5   ,  23.25  ,  12.35  , 110.8833,
              108.9   ,  24.    ,  56.9292,  83.1583, 262.375 ,  14.    ,
              164.8667, 134.5   ,   6.2375,  57.9792,  28.5   , 133.65  ,
               15.9   ,   9.225 ,  35.    ,  75.25  ,  69.3   ,  55.4417,
              211.5   ,   4.0125, 227.525 ,  15.7417,   7.7292,  12.    ,
              120.    ,  12.65  ,  18.75  ,   6.8583,  32.5   ,   7.875 ,
               14.4   ,  55.9   ,   8.1125,  81.8583,  19.2583,  19.9667,
               89.1042,  38.5   ,   7.725 ,  13.7917,   9.8375,   7.0458,
                7.5208,  12.2875,   9.5875,  49.5042,  78.2667,  15.1   ,
                7.6292,  22.525 ,  26.2875,  59.4   ,   7.4958,  34.0208,
               93.5   , 221.7792, 106.425 ,  49.5   ,  71.    ,  13.8625,
                7.8292,  39.6   ,  17.4   ,  51.4792,  26.3875,  30.    ,
               40.125 ,   8.7125,  15.    ,  33.    ,  42.4   ,  15.55  ,
               65.    ,  32.3208,   7.0542,   8.4333,  25.5875,   9.8417,
                8.1375,  10.1708, 211.3375,  57.    ,  13.4167,   7.7417,
                9.4833,   7.7375,   8.3625,  23.45  ,  25.9292,   8.6833,
                8.5167,   7.8875,  37.0042,   6.45  ,   6.95  ,   8.3   ,
                6.4375,  39.4   ,  14.1083,  13.8583,  50.4958,   5.    ,
                9.8458,  10.5167])
```

```
In [16]: kashti['alive'].unique()
```

```
Out[16]: array(['no', 'yes'], dtype=object)
```

Cleaning and Filtering the Data

Find the missing values inside

```
In [19]: kashti.isnull()
```

Out[19]:

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male
0	False	False	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...	...	...	...
886	False	False	False	False	False	False	False	False	False	False	False
887	False	False	False	False	False	False	False	False	False	False	False
888	False	False	False	True	False	False	False	False	False	False	False
889	False	False	False	False	False	False	False	False	False	False	False
890	False	False	False	False	False	False	False	False	False	False	False

891 rows × 15 columns

In [20]:

```
kashti.isnull().sum()
```

Out[20]:

```
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 [21]:

```
kashti_clean = kashti.drop(['deck'], axis =1)
kashti_clean
```

Out[21]:

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adu
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 × 14 columns

In [22]:
```python
kashti_clean = kashti.drop(['age'], axis =1)
kashti_clean
```

Out[22]:

	survived	pclass	sex	sibsp	parch	fare	embarked	class	who	adult_ma
0	0	3	male	1	0	7.2500	S	Third	man	Tru
1	1	1	female	1	0	71.2833	C	First	woman	Fal
2	1	3	female	0	0	7.9250	S	Third	woman	Fal
3	1	1	female	1	0	53.1000	S	First	woman	Fal
4	0	3	male	0	0	8.0500	S	Third	man	Tru
...	...	...	...	...	...	...	...	...	...	
886	0	2	male	0	0	13.0000	S	Second	man	Tru
887	1	1	female	0	0	30.0000	S	First	woman	Fal
888	0	3	female	1	2	23.4500	S	Third	woman	Fal
889	1	1	male	0	0	30.0000	C	First	man	Tru
890	0	3	male	0	0	7.7500	Q	Third	man	Tru

891 rows × 14 columns

In [23]:
```python
kashti_clean.head()
```

Out[23]:

	survived	pclass	sex	sibsp	parch	fare	embarked	class	who	adult_male	c
0	0	3	male	1	0	7.2500	S	Third	man	True	I
1	1	1	female	1	0	71.2833	C	First	woman	False	
2	1	3	female	0	0	7.9250	S	Third	woman	False	I
3	1	1	female	1	0	53.1000	S	First	woman	False	
4	0	3	male	0	0	8.0500	S	Third	man	True	I

In [24]:
```python
kashti_clean =kashti.drop(['deck'], axis =1)
kashti_clean
```

Out[24]:

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adı
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 × 14 columns

In [25]:
```python
kashti_clean.isnull().sum()
```

Out[25]:

```
survived         0
pclass           0
sex              0
age            177
sibsp            0
parch            0
fare             0
embarked         2
class            0
who              0
adult_male       0
embark_town      2
alive            0
alone            0
dtype: int64
```

In [26]:

```
kashti_cleaned = kashti_clean.drop(['age'] ,axis =1)
kashti_cleaned
```

Out[26]:

	survived	pclass	sex	sibsp	parch	fare	embarked	class	who	adult_ma
0	0	3	male	1	0	7.2500	S	Third	man	Tru
1	1	1	female	1	0	71.2833	C	First	woman	Fal
2	1	3	female	0	0	7.9250	S	Third	woman	Fal
3	1	1	female	1	0	53.1000	S	First	woman	Fal
4	0	3	male	0	0	8.0500	S	Third	man	Tru
...	...	...	...	...	...	...	...	...	...	
886	0	2	male	0	0	13.0000	S	Second	man	Tru
887	1	1	female	0	0	30.0000	S	First	woman	Fal
888	0	3	female	1	2	23.4500	S	Third	woman	Fal
889	1	1	male	0	0	30.0000	C	First	man	Tru
890	0	3	male	0	0	7.7500	Q	Third	man	Tru

891 rows × 13 columns

In [27]:

```
kashti_cleaned.isnull().sum()
```

Out[27]:

```
survived       0
pclass         0
sex            0
sibsp          0
parch          0
fare           0
embarked       2
class          0
who            0
adult_male     0
embark_town    2
alive          0
alone          0
dtype: int64
```

In [31]:

```
kashti.shape
```

Out[31]:

```
(891, 15)
```

In [33]:

```
kashti_cleaned.shape
```

Out[33]:

```
(891, 13)
```

In [35]:

```
kashti.describe()
```

Out[35]:

	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 [37]:

```
kashti_cleaned.describe()
```

Out[37]:

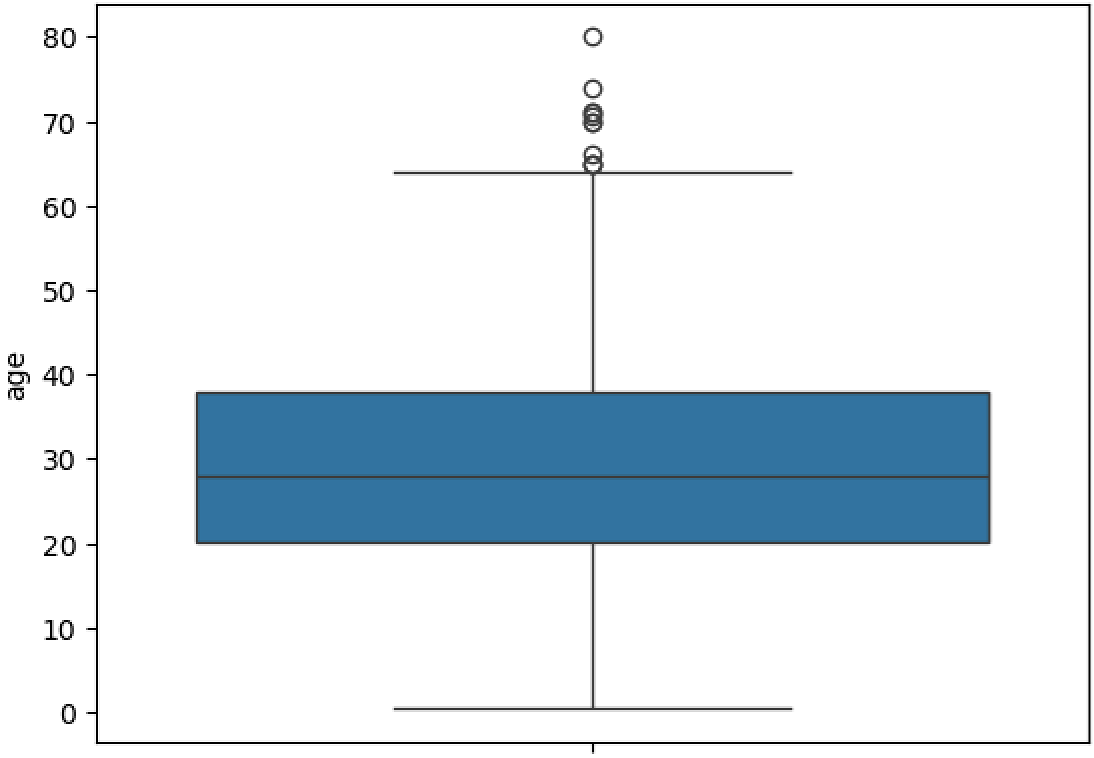
	survived	pclass	sibsp	parch	fare
count	891.000000	891.000000	891.000000	891.000000	891.000000
mean	0.383838	2.308642	0.523008	0.381594	32.204208
std	0.486592	0.836071	1.102743	0.806057	49.693429
min	0.000000	1.000000	0.000000	0.000000	0.000000
25%	0.000000	2.000000	0.000000	0.000000	7.910400
50%	0.000000	3.000000	0.000000	0.000000	14.454200
75%	1.000000	3.000000	1.000000	0.000000	31.000000
max	1.000000	3.000000	8.000000	6.000000	512.329200

In [41]:

```
sns.boxplot(y ='age', data = kashti)
```

Out[41]:

```
<Axes: ylabel='age'>
```

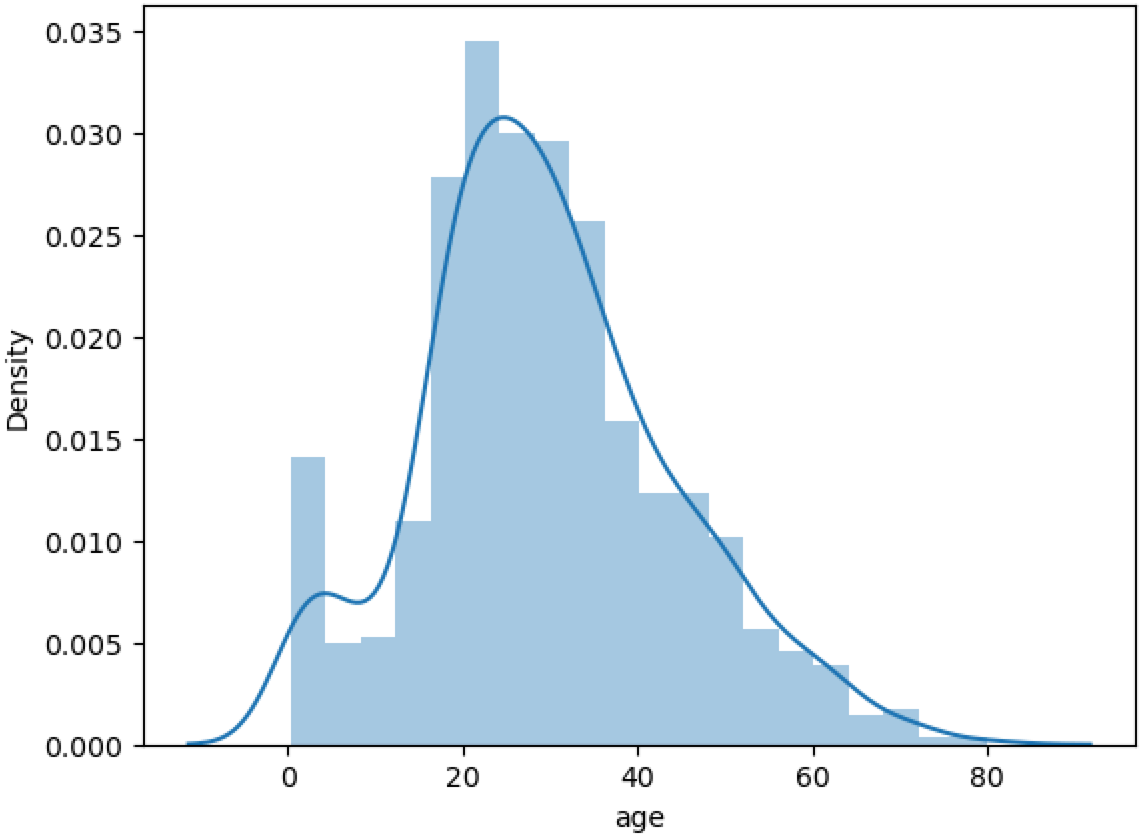


In [43]:

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

```
C:\Users\PC VISION\AppData\Local\Temp\ipykernel_18024\3786974353.py:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

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

```
Out[43]: <Axes: xlabel='age', ylabel='Density'>
```



```
In [45]: kashti['age'].mean()
```

```
Out[45]: 29.69911764705882
```

```
In [47]: kashti['age'].mode()
```

```
Out[47]: 0    24.0
         Name: age, dtype: float64
```

```
In [49]: kashti['age'].median()
```

```
Out[49]: 28.0
```

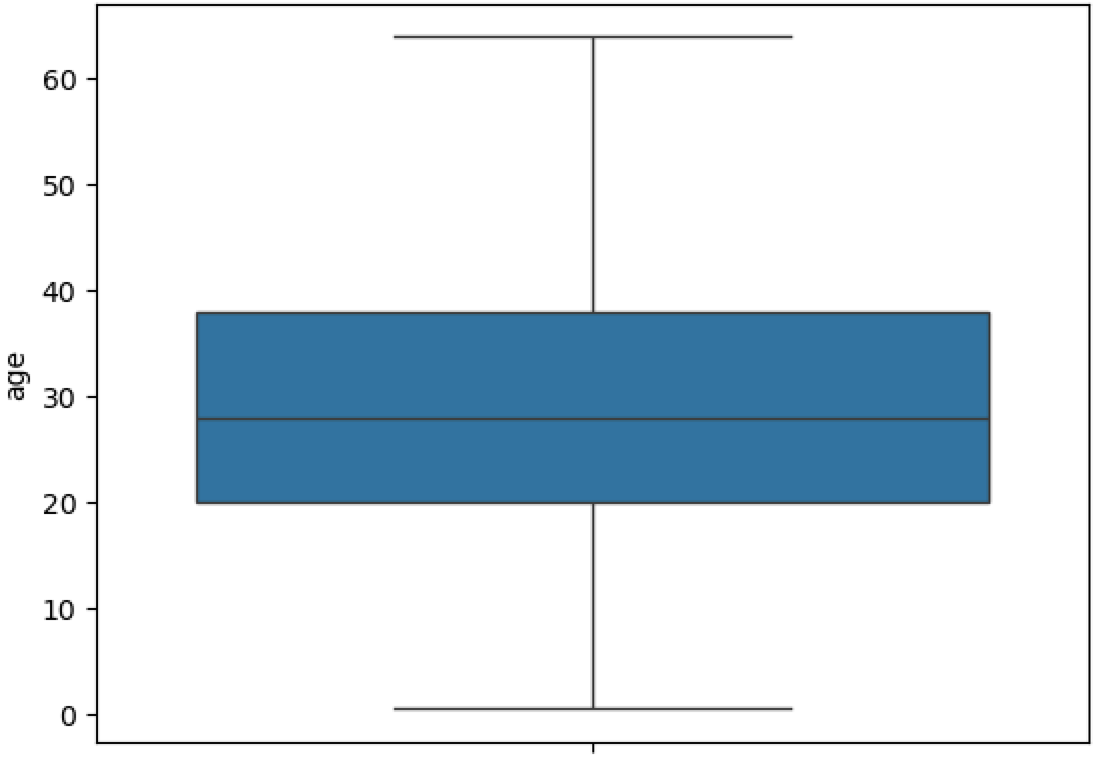
```
In [59]: kashti = kashti[kashti['age'] < 69]
```

In [61]:
```
kashti['age'].mean()
```

Out[61]:
```
29.07207681365576
```

In [63]:
```
sns.boxplot(kashti['age'])
```

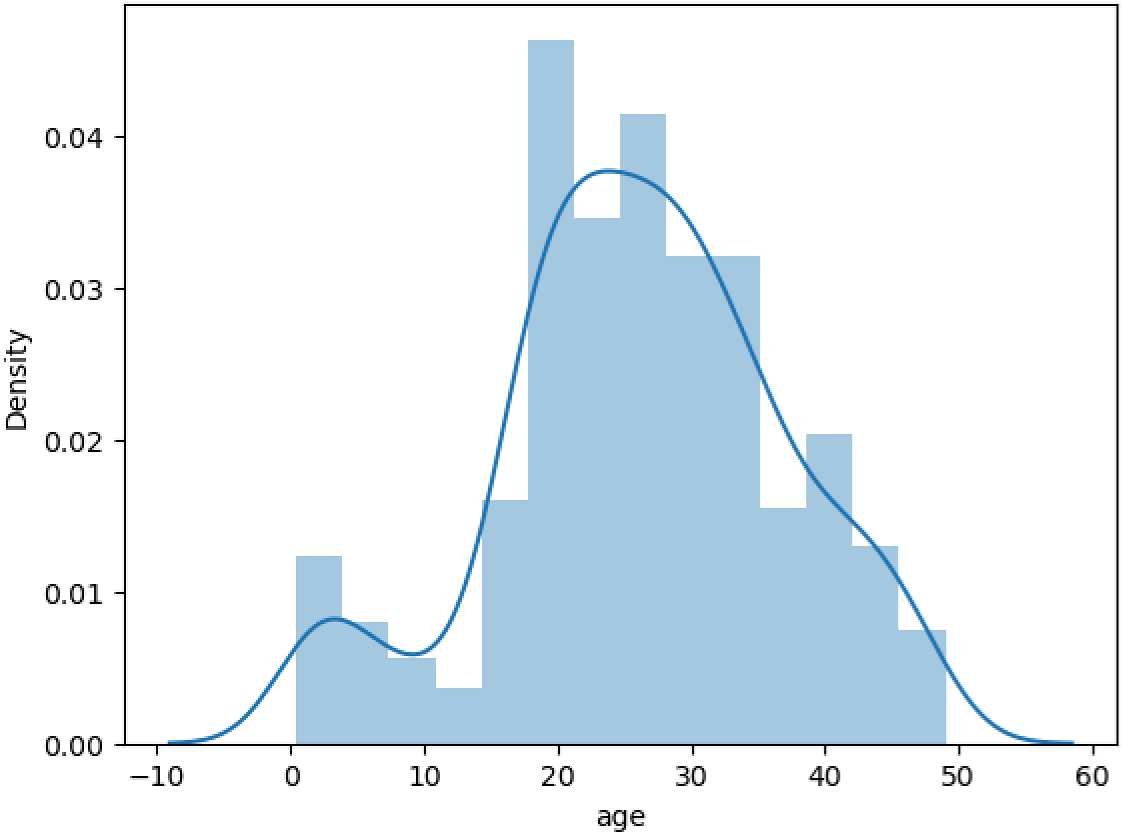
Out[63]:
```
<Axes: ylabel='age'>
```



In [117…
```
kashti_cleaned = kashti_cleaned[kashti_cleaned['age'] < 50]
sns.distplot(kashti_cleaned['age'])
```

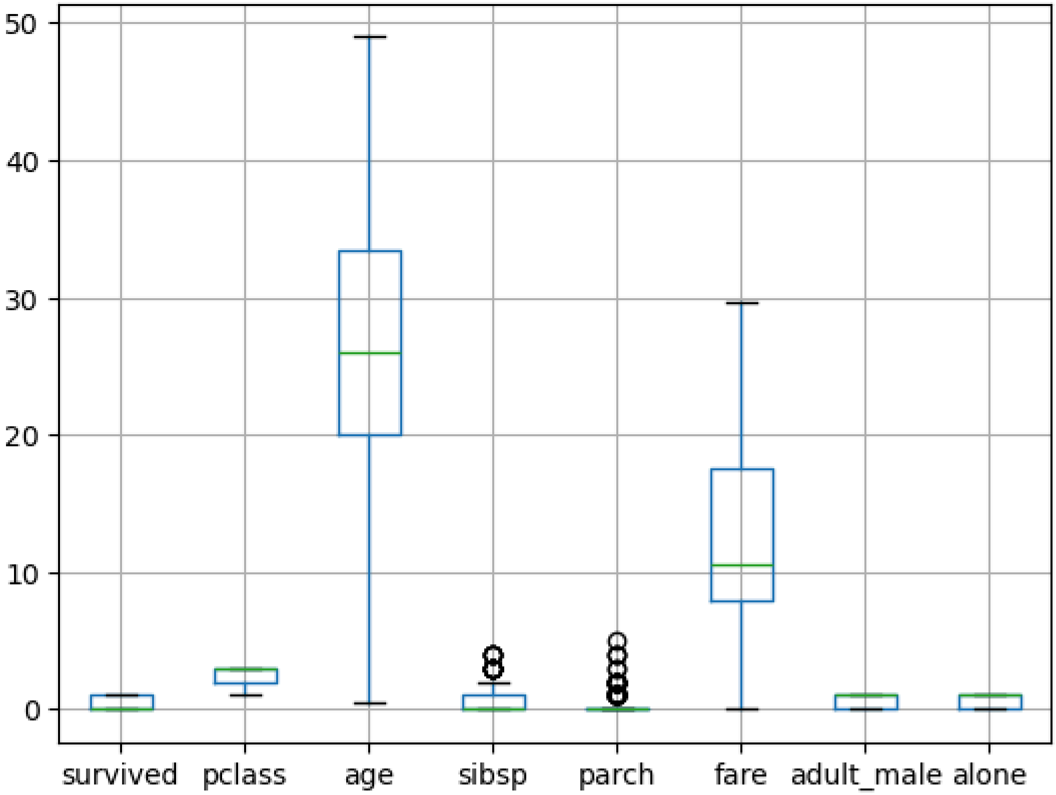
```
C:\Users\PC VISION\AppData\Local\Temp\ipykernel_18024\3980623757.py:2: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

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

Out[117…
```
<Axes: xlabel='age', ylabel='Density'>
```



```
In [119…  kashti_cleaned.boxplot()
```

```
Out[119…  <Axes: >
```



In [113…

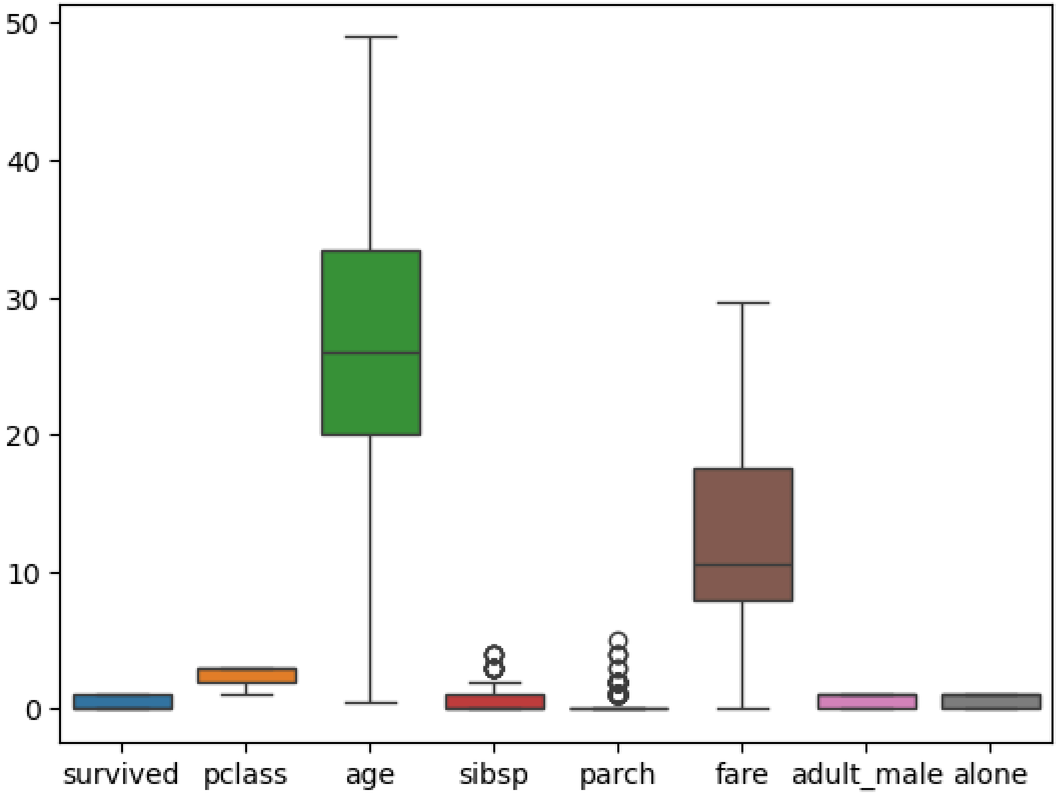
```
kashti_cleaned = kashti_cleaned[kashti_cleaned['fare'] < 30 ]
```

In [121…

```
sns.boxplot(kashti_cleaned)
```

Out[121…

```
<Axes: >
```



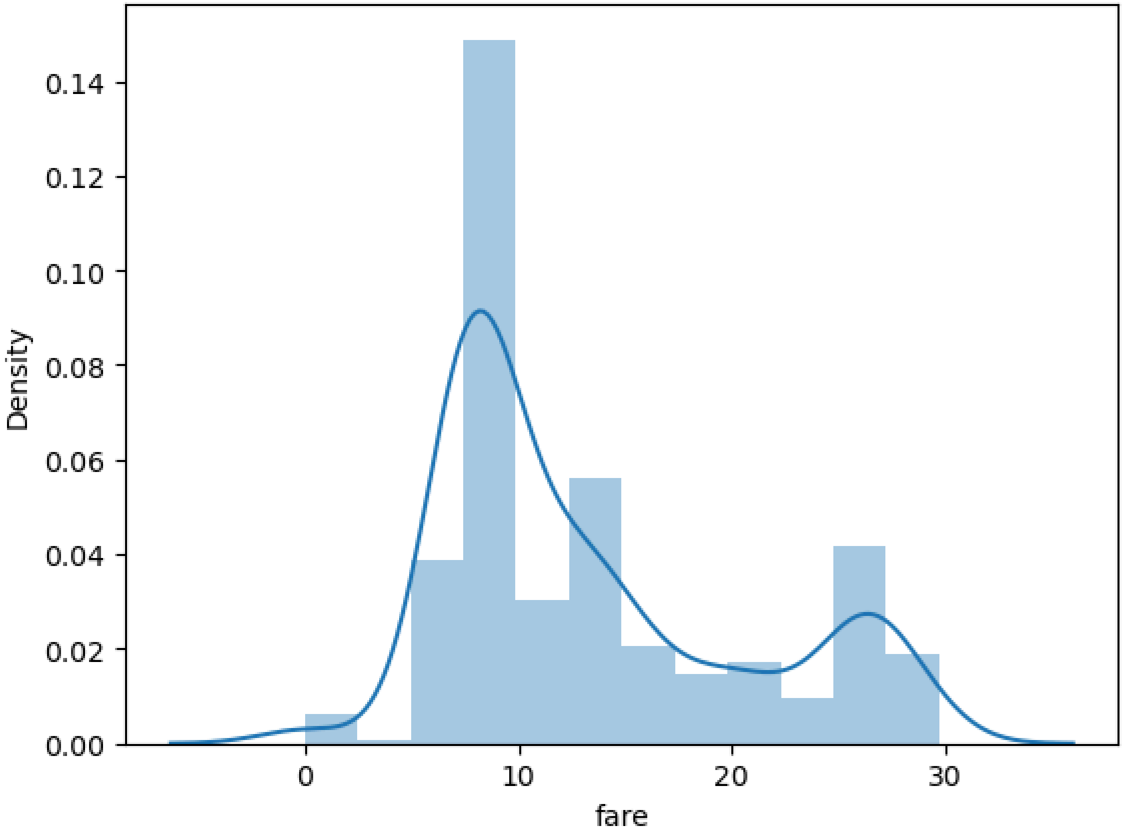
In [125…

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

```
C:\Users\PC VISION\AppData\Local\Temp\ipykernel_18024\2905693554.py:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

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

Out[125…

```
<Axes: xlabel='fare', ylabel='Density'>
```

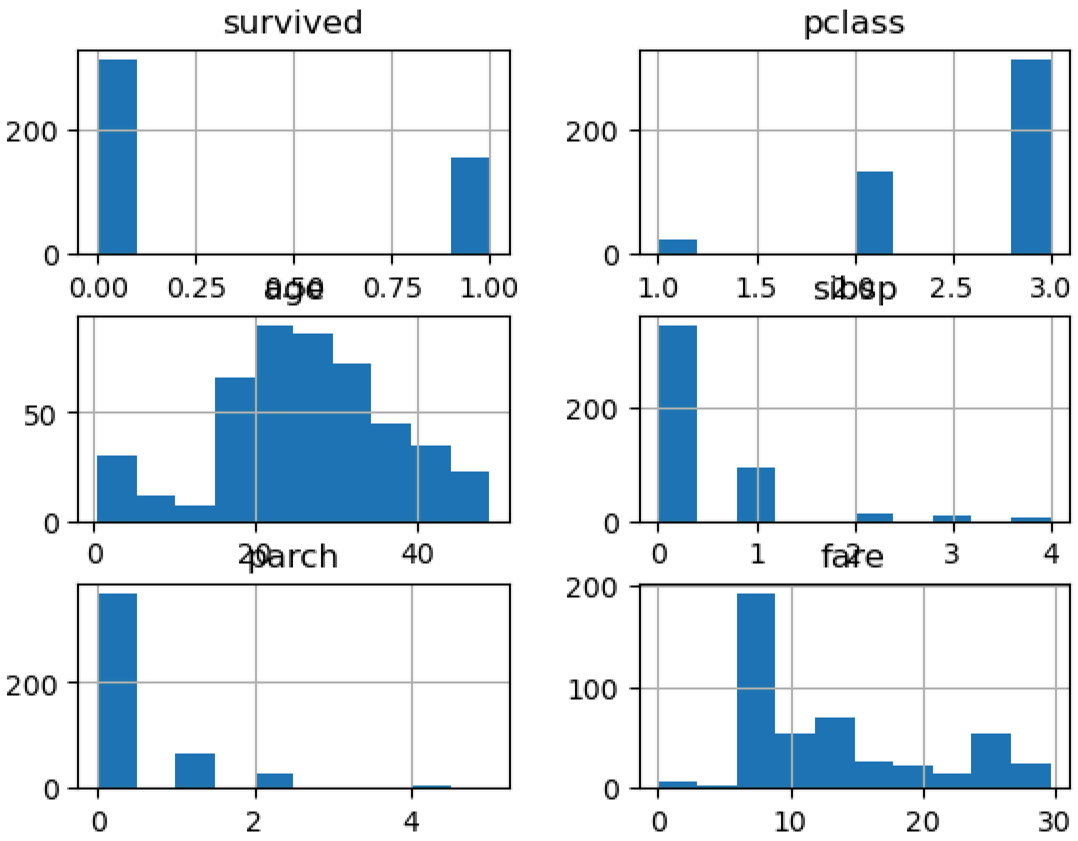


In [131…

```
kashti_cleaned.hist( )
```

Out[131…

```
array([[<Axes: title={'center': 'survived'}>,
        <Axes: title={'center': 'pclass'}>],
       [<Axes: title={'center': 'age'}>,
        <Axes: title={'center': 'sibsp'}>],
       [<Axes: title={'center': 'parch'}>,
        <Axes: title={'center': 'fare'}>]], dtype=object)
```

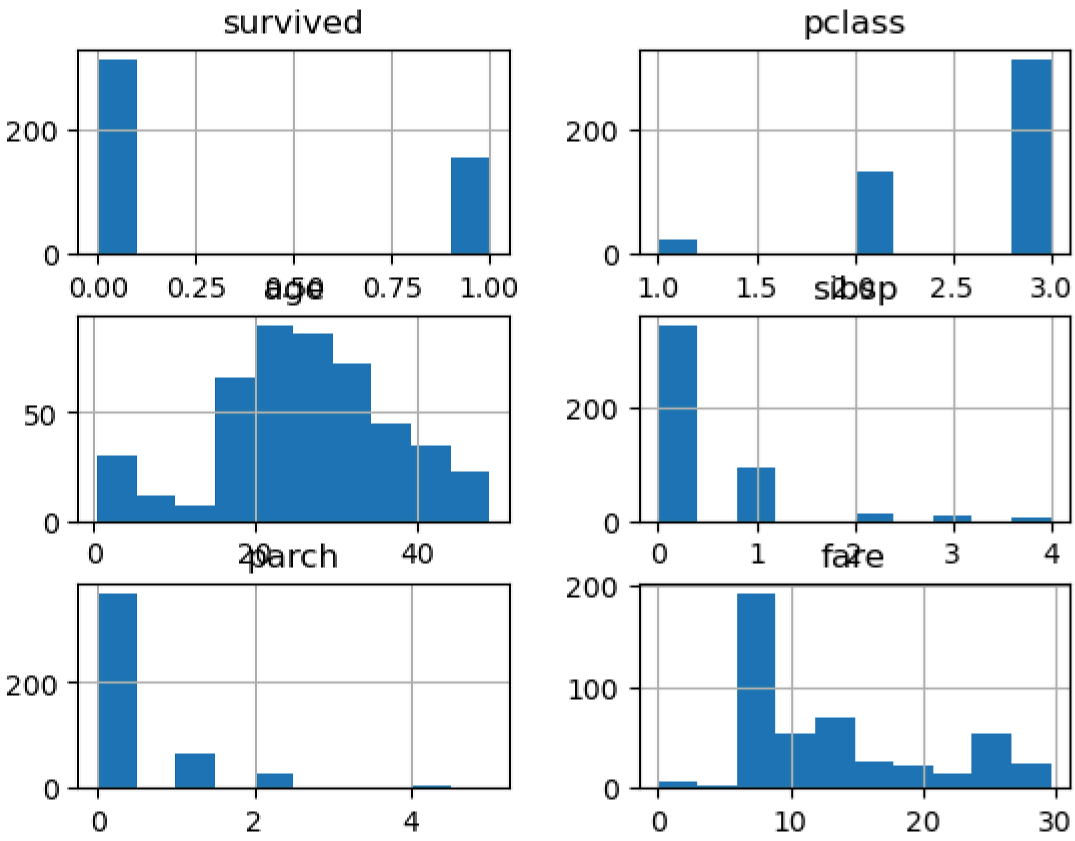


In [131…

```
kashti_cleaned.hist( )
```

Out[131…

```
array([[<Axes: title={'center': 'survived'}>,
        <Axes: title={'center': 'pclass'}>],
       [<Axes: title={'center': 'age'}>,
        <Axes: title={'center': 'sibsp'}>],
       [<Axes: title={'center': 'parch'}>,
        <Axes: title={'center': 'fare'}>]], dtype=object)
```



In [145…

```
kashti_cleaned.groupby(['sex','class']).mean(numeric_only=True)
```

```
C:\Users\PC VISION\AppData\Local\Temp\ipykernel_18024\2306802719.py:1: FutureWarnin
g: The default of observed=False is deprecated and will be changed to True in a futu
re version of pandas. Pass observed=False to retain current behavior or observed=Tru
e to adopt the future default and silence this warning.
  kashti_cleaned.groupby(['sex','class']).mean(numeric_only=True)
```

Out[145…

		survived	pclass	age	sibsp	parch	fare	adult_male	
sex	**class**								
female	**First**	1.000000	1.0	40.000000	0.000000	0.500000	26.465625	0.000000	0.7
	Second	0.913793	2.0	27.405172	0.448276	0.500000	19.157759	0.000000	0.4
	Third	0.500000	3.0	21.221591	0.579545	0.647727	13.112927	0.000000	0.4
male	**First**	0.473684	1.0	38.605263	0.000000	0.052632	21.516226	1.000000	0.9
	Second	0.150685	2.0	27.956575	0.301370	0.191781	16.673515	0.917808	0.0
	Third	0.151111	3.0	26.019644	0.311111	0.168889	9.909887	0.924444	0.7

In [147…

```
kashti.groupby(['sex' , 'class']).mean(numeric_only = True)
```

```
C:\Users\PC VISION\AppData\Local\Temp\ipykernel_18024\1968306831.py:1: FutureWarnin
g: The default of observed=False is deprecated and will be changed to True in a futu
re version of pandas. Pass observed=False to retain current behavior or observed=Tru
e to adopt the future default and silence this warning.
  kashti.groupby(['sex' , 'class']).mean(numeric_only = True)
```

Out[147…

		survived	pclass	age	sibsp	parch	fare	adult_male
sex	class							
female	First	0.964286	1.0	34.607143	0.547619	0.500000	103.132193	0.000000
	Second	0.918919	2.0	28.722973	0.500000	0.621622	21.951070	0.000000
	Third	0.460784	3.0	21.750000	0.823529	0.950980	15.875369	0.000000
male	First	0.397849	1.0	39.531398	0.397849	0.333333	63.301881	0.967742
	Second	0.154639	2.0	29.972474	0.381443	0.247423	21.331959	0.907216
	Third	0.152000	3.0	25.987680	0.496000	0.260000	12.215548	0.888000

In [155…

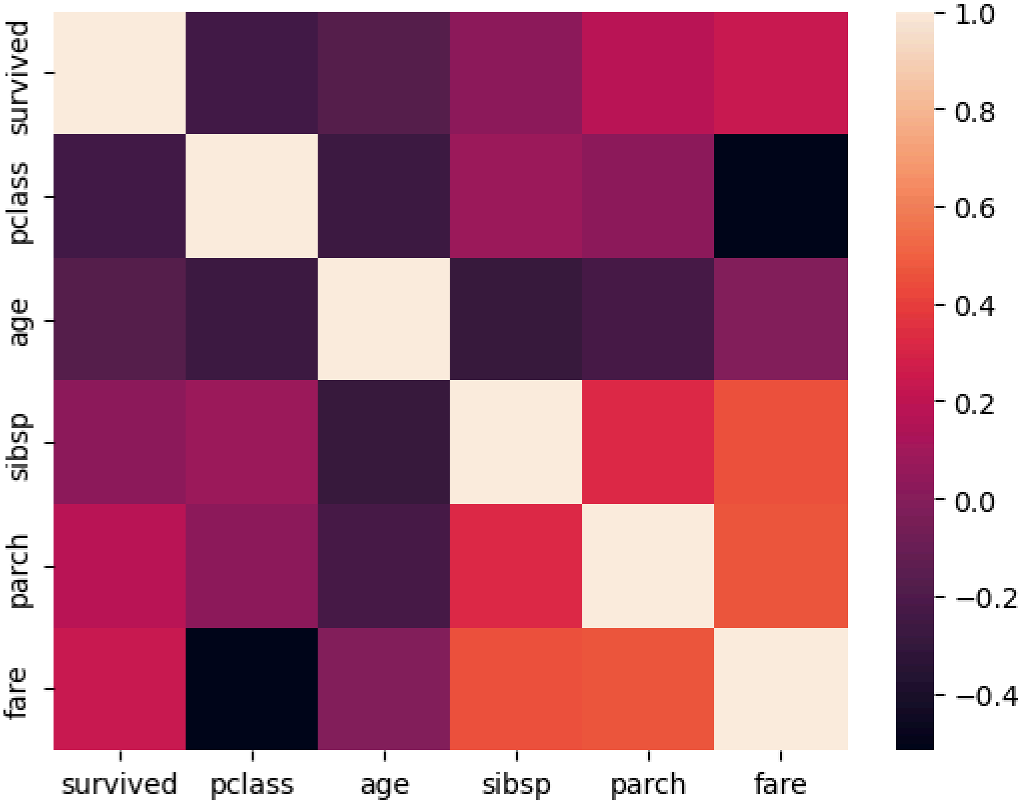
```python
kashti_cleaned_corr = kashti_cleaned.select_dtypes(include='number').corr()
```

In [157…

```python
sns.heatmap(kashti_cleaned_corr)
```

Out[157…

```
<Axes: >
```



In [ ]: