Statistics_ASS_06

March 14, 2022

1 Exploratory Data Analisi

This will show us how we can do EDA using Python ### Three Important steps to keep in mind are - Understand the data - Clean the Data - Find the Relationship b/w the Data

[ ]:
```
# EDA Data
# Understand the data
# clean the Data
# find the relationship between the data

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

[ ]:
```
kashti= sns.load_dataset("titanic")
kashti.head(4)
```

[ ]:

	Unnamed: 0	survived	pclass	sex	age	sibsp	parch	fare	embarked	\
0	0	0	3	male	22.0	1	0	7.2500	S	
1	1	1	1	female	38.0	1	0	71.2833	C	
2	2	1	3	female	26.0	0	0	7.9250	S	
3	3	1	1	female	35.0	1	0	53.1000	S	

	class	who	adult_male	deck	embark_town	alive	alone
0	Third	man	True	NaN	Southampton	no	False
1	First	woman	False	C	Cherbourg	yes	False
2	Third	woman	False	NaN	Southampton	yes	True
3	First	woman	False	C	Southampton	yes	False

[ ]:
```
kashti.to_csv("kashti1.csv")
```

[ ]:
```
kashti.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 16 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
```

```
 0   Unnamed: 0   891 non-null    int64
 1   survived     891 non-null    int64
 2   pclass       891 non-null    int64
 3   sex          891 non-null    object
 4   age          714 non-null    float64
 5   sibsp        891 non-null    int64
 6   parch        891 non-null    int64
 7   fare         891 non-null    float64
 8   embarked     889 non-null    object
 9   class        891 non-null    category
 10  who          891 non-null    object
 11  adult_male   891 non-null    bool
 12  deck         203 non-null    category
 13  embark_town  889 non-null    object
 14  alive        891 non-null    object
 15  alone        891 non-null    bool
dtypes: bool(2), category(2), float64(2), int64(5), object(5)
memory usage: 87.6+ KB
```

[ ]:
```
ks = kashti
```

[ ]:
```
ks.head(5)
```

[ ]:
```
   Unnamed: 0  survived  pclass     sex   age  sibsp  parch     fare embarked  \
0           0         0       3    male  22.0      1      0   7.2500        S
1           1         1       1  female  38.0      1      0  71.2833        C
2           2         1       3  female  26.0      0      0   7.9250        S
3           3         1       1  female  35.0      1      0  53.1000        S
4           4         0       3    male  35.0      0      0   8.0500        S

   class    who  adult_male deck  embark_town alive  alone
0  Third    man        True  NaN  Southampton    no  False
1  First  woman       False    C    Cherbourg   yes  False
2  Third  woman       False  NaN  Southampton   yes   True
3  First  woman       False    C  Southampton   yes  False
4  Third    man        True  NaN  Southampton    no   True
```

[ ]:
```
ks.shape
# Rows and coloums
```

[ ]:
```
(891, 16)
```

[ ]:
```
ks.tail(5)
# End 5 rows and coloumns
```

[ ]:
```
     Unnamed: 0  survived  pclass     sex   age  sibsp  parch   fare embarked  \
886         886         0       2    male  27.0      0      0  13.00        S
887         887         1       1  female  19.0      0      0  30.00        S
```

```
888          888         0       3  female   NaN      1      2  23.45        S
889          889         1       1    male  26.0      0      0  30.00        C
890          890         0       3    male  32.0      0      0   7.75        Q

      class    who  adult_male deck  embark_town alive  alone
886  Second    man        True  NaN  Southampton    no   True
887   First  woman       False    B  Southampton   yes   True
888   Third  woman       False  NaN  Southampton    no  False
889   First    man        True    C    Cherbourg   yes   True
890   Third    man        True  NaN   Queenstown    no   True
```

[ ]:
```
ks.describe()
```

[ ]:
```
       Unnamed: 0    survived      pclass         age       sibsp       parch  \
count  891.000000  891.000000  891.000000  714.000000  891.000000  891.000000
mean   445.000000    0.383838    2.308642   29.699118    0.523008    0.381594
std    257.353842    0.486592    0.836071   14.526497    1.102743    0.806057
min      0.000000    0.000000    1.000000    0.420000    0.000000    0.000000
25%    222.500000    0.000000    2.000000   20.125000    0.000000    0.000000
50%    445.000000    0.000000    3.000000   28.000000    0.000000    0.000000
75%    667.500000    1.000000    3.000000   38.000000    1.000000    0.000000
max    890.000000    1.000000    3.000000   80.000000    8.000000    6.000000

             fare
count  891.000000
mean    32.204208
std     49.693429
min      0.000000
25%      7.910400
50%     14.454200
75%     31.000000
max    512.329200
```

[ ]:
```
# This is how we know how many unique values
ks.nunique()
```

[ ]:
```
Unnamed: 0     891
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
```

[ ]:
```
# coloumns names
ks.columns
```

[ ]:
```
Index(['Unnamed: 0', 'survived', 'pclass', 'sex', 'age', 'sibsp', 'parch',
       'fare', 'embarked', 'class', 'who', 'adult_male', 'deck', 'embark_town',
       'alive', 'alone'],
      dtype='object')
```

[ ]:
```
# to see a unique value in a coloumn
ks['sex'].unique()
```

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

[ ]:
```
ks['class'].unique()
```

[ ]:
```
['Third', 'First', 'Second']
Categories (3, object): ['Third', 'First', 'Second']
```

[ ]:
```
val = ks[['sex','class','who']].values
np.unique(val)
```

[ ]:
```
array(['First', 'Second', 'Third', 'child', 'female', 'male', 'man',
       'woman'], dtype=object)
```

1.1 Data Cleaning and Filtering

[ ]:
```
# Cleaning and filtering the data
# Find missing values
ks.isnull().sum()
```

[ ]:
```
Unnamed: 0       0
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
```

[ ]:
```
# Removing missing values in data and  the Cleaning Data
ks_clean = ks.drop(['deck'],axis=1)
ks_clean.head(5)
```

[ ]:
```
   Unnamed: 0  survived  pclass     sex   age  sibsp  parch     fare embarked  \
0           0         0       3    male  22.0      1      0   7.2500        S
1           1         1       1  female  38.0      1      0  71.2833        C
2           2         1       3  female  26.0      0      0   7.9250        S
3           3         1       1  female  35.0      1      0  53.1000        S
4           4         0       3    male  35.0      0      0   8.0500        S

   class    who  adult_male  embark_town alive  alone
0  Third    man        True  Southampton    no  False
1  First  woman       False    Cherbourg   yes  False
2  Third  woman       False  Southampton   yes   True
3  First  woman       False  Southampton   yes  False
4  Third    man        True  Southampton    no   True
```

[ ]:
```
ks_clean.isnull().sum()
```

[ ]:
```
Unnamed: 0       0
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
```

[ ]:
```
ks_clean.shape
```

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

[ ]:
```
ks_clean = ks_clean.dropna()
```

[ ]:
```
ks_clean.shape
```

[ ]:
```
(712, 15)
```

[ ]:
```
ks_clean.isnull().sum()
```

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

[ ]:
```
ks.shape
```

[ ]:
```
(891, 16)
```

[ ]:
```
ks_clean.shape
```

[ ]:
```
(712, 15)
```

[ ]:
```
ks_clean['age'].value_counts()
```

[ ]:
```
24.00    30
22.00    27
18.00    26
28.00    25
19.00    25
         ..
55.50     1
74.00     1
0.92      1
70.50     1
12.00     1
Name: age, Length: 88, dtype: int64
```

[ ]:
```
ks.describe()
```

[ ]:
```
       Unnamed: 0    survived      pclass         age       sibsp       parch  \
count  891.000000  891.000000  891.000000  714.000000  891.000000  891.000000
mean   445.000000    0.383838    2.308642   29.699118    0.523008    0.381594
std    257.353842    0.486592    0.836071   14.526497    1.102743    0.806057
min      0.000000    0.000000    1.000000    0.420000    0.000000    0.000000
25%    222.500000    0.000000    2.000000   20.125000    0.000000    0.000000
50%    445.000000    0.000000    3.000000   28.000000    0.000000    0.000000
75%    667.500000    1.000000    3.000000   38.000000    1.000000    0.000000
max    890.000000    1.000000    3.000000   80.000000    8.000000    6.000000

             fare
count  891.000000
mean    32.204208
std     49.693429
min      0.000000
25%      7.910400
50%     14.454200
75%     31.000000
max    512.329200
```

[ ]:
```
ks_clean.describe()
```

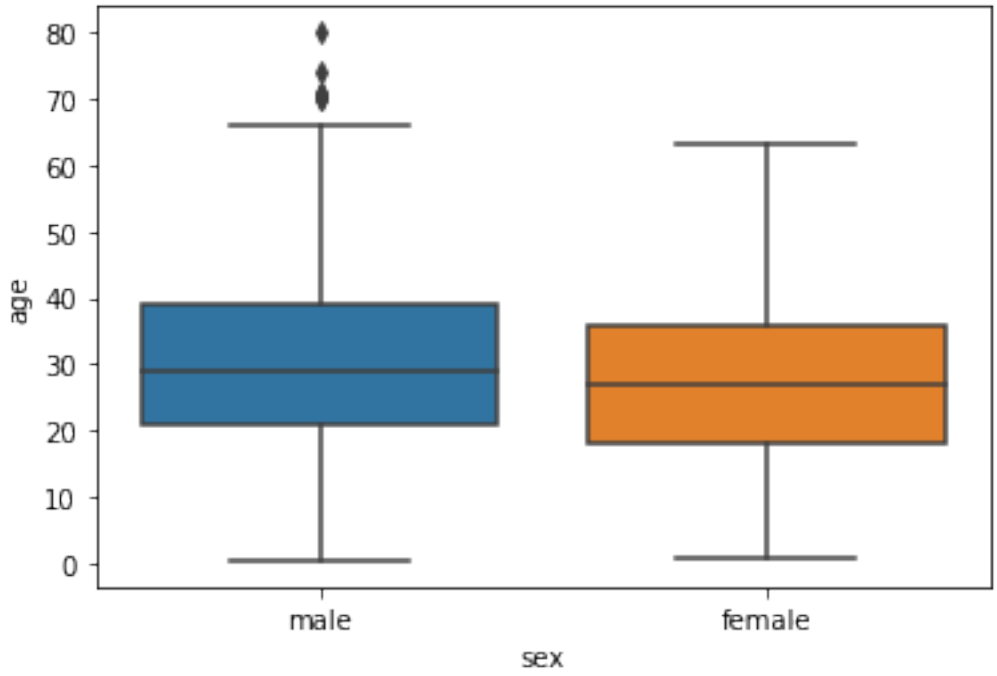
[ ]:
```
       Unnamed: 0    survived      pclass         age       sibsp       parch  \
count  712.000000  712.000000  712.000000  712.000000  712.000000  712.000000
mean   447.589888    0.404494    2.240169   29.642093    0.514045    0.432584
std    258.683191    0.491139    0.836854   14.492933    0.930692    0.854181
min      0.000000    0.000000    1.000000    0.420000    0.000000    0.000000
25%    221.750000    0.000000    1.000000   20.000000    0.000000    0.000000
50%    444.000000    0.000000    2.000000   28.000000    0.000000    0.000000
75%    676.250000    1.000000    3.000000   38.000000    1.000000    1.000000
max    890.000000    1.000000    3.000000   80.000000    5.000000    6.000000

             fare
count  712.000000
mean    34.567251
std     52.938648
min      0.000000
25%      8.050000
50%     15.645850
75%     33.000000
max    512.329200
```

[ ]:
```
ks_clean.columns
```

[ ]:
```
Index(['Unnamed: 0', 'survived', 'pclass', 'sex', 'age', 'sibsp', 'parch',
       'fare', 'embarked', 'class', 'who', 'adult_male', 'embark_town',
       'alive', 'alone'],
      dtype='object')
```

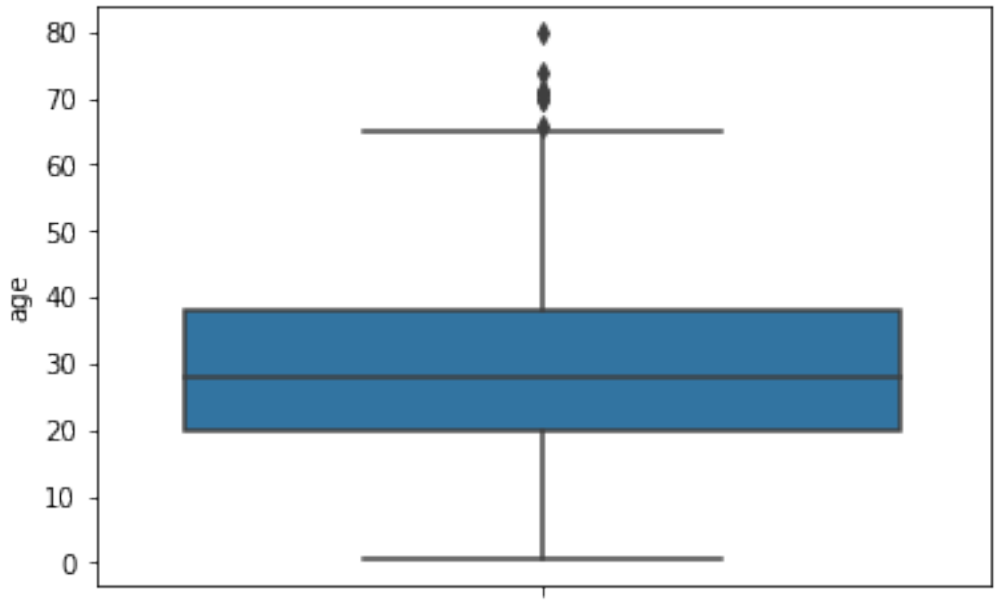
[ ]:
```
sns.boxplot(x='sex',y='age', data=ks_clean)
```

[ ]:
```
<AxesSubplot:xlabel='sex', ylabel='age'>
```



[ ]:
```
sns.boxplot(y='age', data=ks_clean)
```

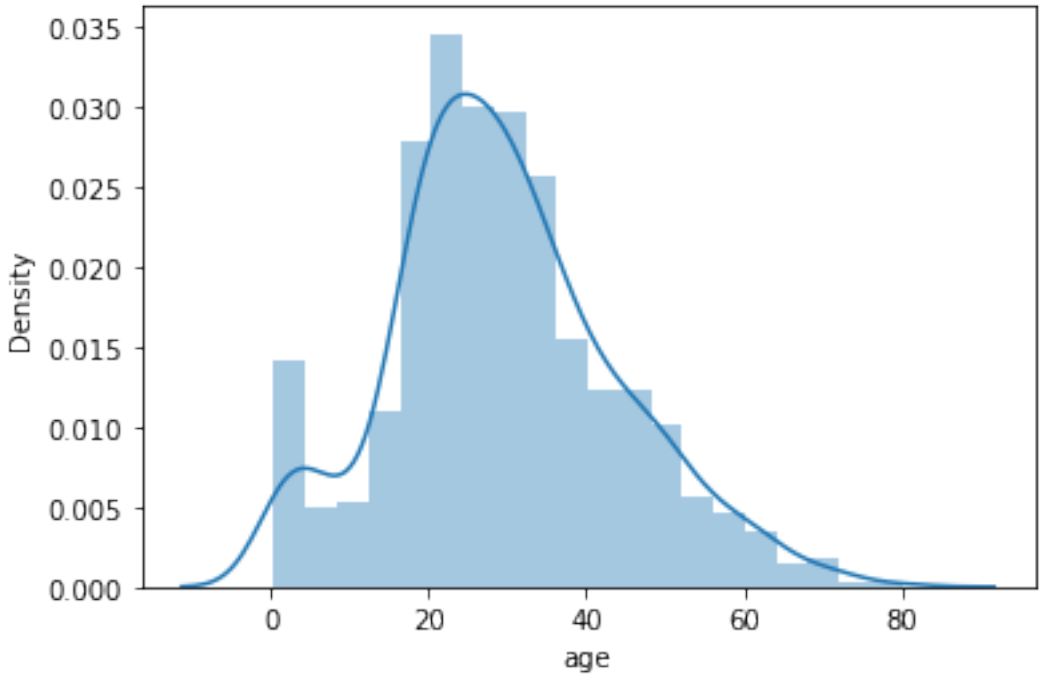
[ ]:
```
<AxesSubplot:ylabel='age'>
```



[ ]:
```
sns.distplot(ks_clean['age'])
```

```
C:\Users\Sartaj\AppData\Local\Programs\Python\Python39\lib\site-
packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a
deprecated function and will be removed in a future version. Please adapt your
code to use either `displot` (a figure-level function with similar flexibility)
or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

[ ]:
```
<AxesSubplot:xlabel='age', ylabel='Density'>
```



[ ]:
```
# Outliers remover
ks_clean['age'].mean()
```

[ ]:
```
29.64209269662921
```

[ ]:
```
ks_clean= ks_clean[ks_clean['age']<60]
```

[ ]:
```
ks_clean.head(5)
```

[ ]:
```
   Unnamed: 0  survived  pclass     sex   age  sibsp  parch     fare embarked  \
0           0         0       3    male  22.0      1      0   7.2500        S
1           1         1       1  female  38.0      1      0  71.2833        C
2           2         1       3  female  26.0      0      0   7.9250        S
3           3         1       1  female  35.0      1      0  53.1000        S
4           4         0       3    male  35.0      0      0   8.0500        S

   class    who  adult_male  embark_town alive  alone  fare_log
0  Third    man        True  Southampton    no  False  1.981001
1  First  woman       False    Cherbourg   yes  False  4.266662
2  Third  woman       False  Southampton   yes   True  2.070022
3  First  woman       False  Southampton   yes  False  3.972177
4  Third    man        True  Southampton    no   True  2.085672
```

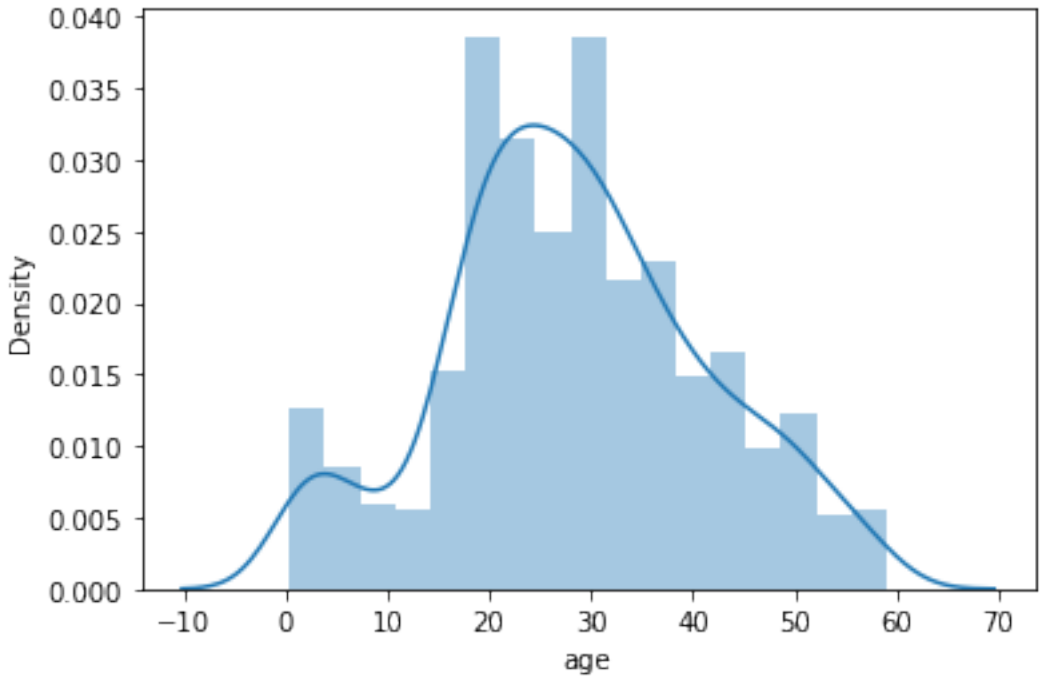
[ ]:
```
ks_clean.shape
```

[ ]: (684, 16)

[ ]:
```
sns.distplot(ks_clean['age'])
```

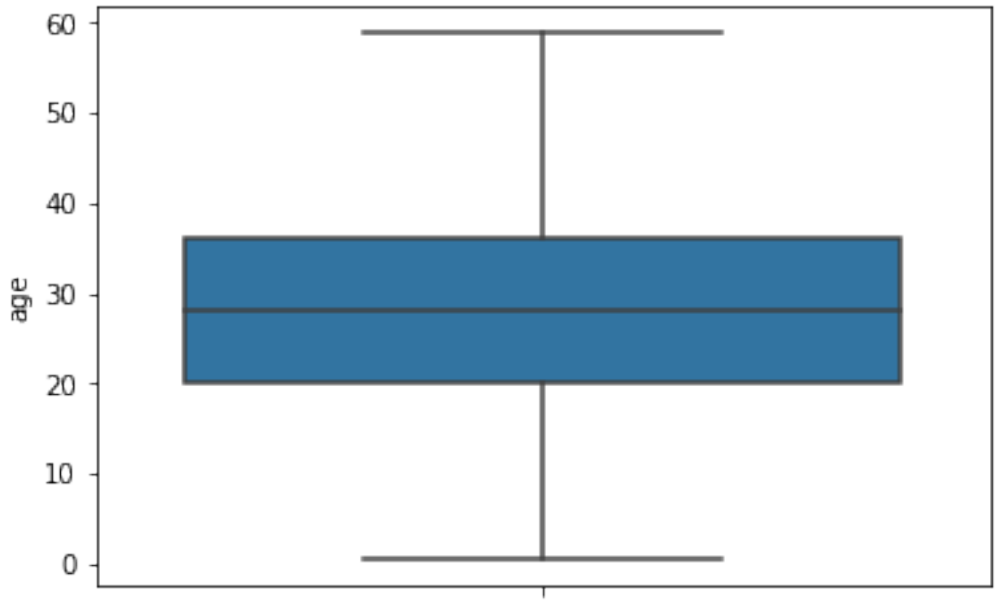
```
C:\Users\Sartaj\AppData\Local\Programs\Python\Python39\lib\site-
packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a
deprecated function and will be removed in a future version. Please adapt your
code to use either `displot` (a figure-level function with similar flexibility)
or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

[ ]: <AxesSubplot:xlabel='age', ylabel='Density'>



[ ]:
```
sns.boxplot(y='age',data=ks_clean)
```

[ ]: <AxesSubplot:ylabel='age'>



[ ]:
```
ks_clean.head(5)
```

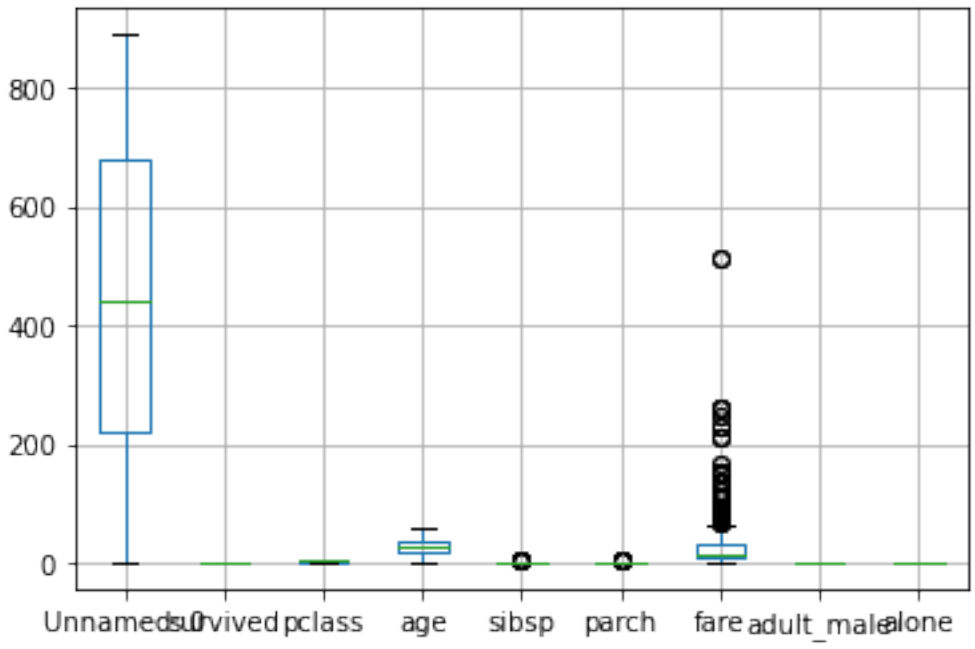
[ ]:

	Unnamed: 0	survived	pclass	sex	age	sibsp	parch	fare	embarked	\
0	0	0	3	male	22.0	1	0	7.2500	S	
1	1	1	1	female	38.0	1	0	71.2833	C	
2	2	1	3	female	26.0	0	0	7.9250	S	
3	3	1	1	female	35.0	1	0	53.1000	S	
4	4	0	3	male	35.0	0	0	8.0500	S	

	class	who	adult_male	embark_town	alive	alone
0	Third	man	True	Southampton	no	False
1	First	woman	False	Cherbourg	yes	False
2	Third	woman	False	Southampton	yes	True
3	First	woman	False	Southampton	yes	False
4	Third	man	True	Southampton	no	True

[ ]:
```
ks_clean.boxplot()
```

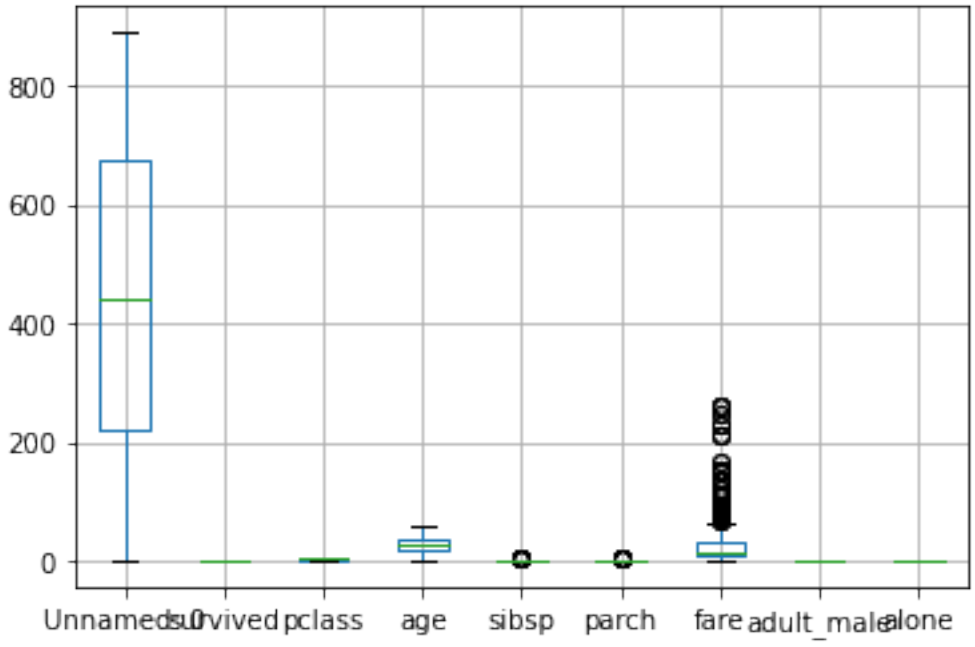
[ ]: <AxesSubplot:>



[ ]:
```
ks_clean = ks_clean[ks_clean['fare']<300]
```

[ ]:
```
ks_clean.boxplot()
```

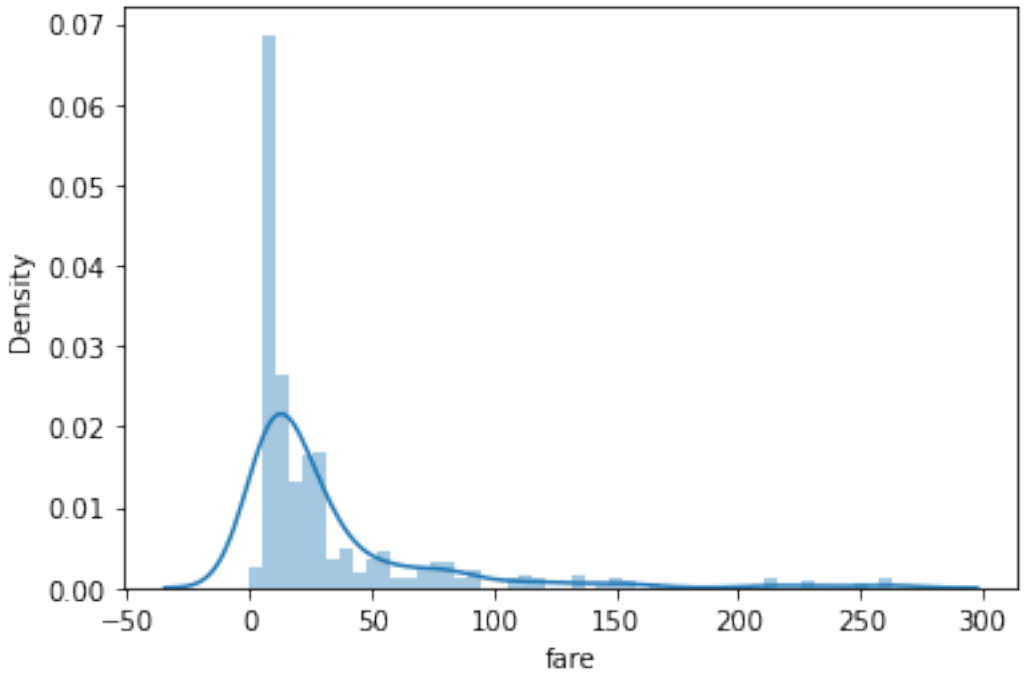
[ ]: <AxesSubplot:>



[ ]:
```
sns.distplot(ks_clean['fare'])
```

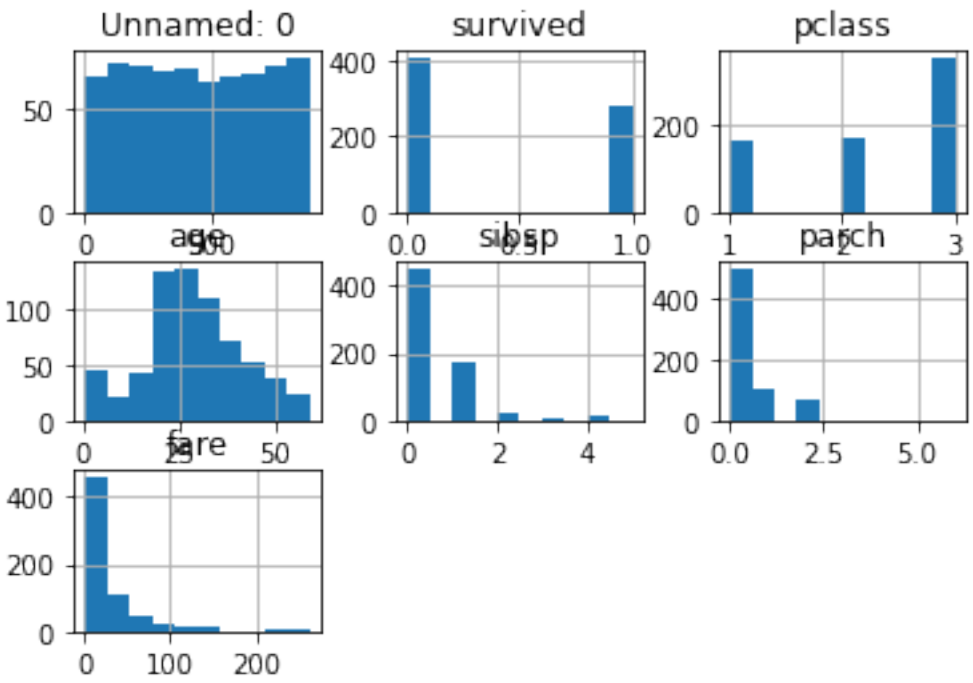
```
C:\Users\Sartaj\AppData\Local\Programs\Python\Python39\lib\site-
packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a
deprecated function and will be removed in a future version. Please adapt your
code to use either `displot` (a figure-level function with similar flexibility)
or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

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



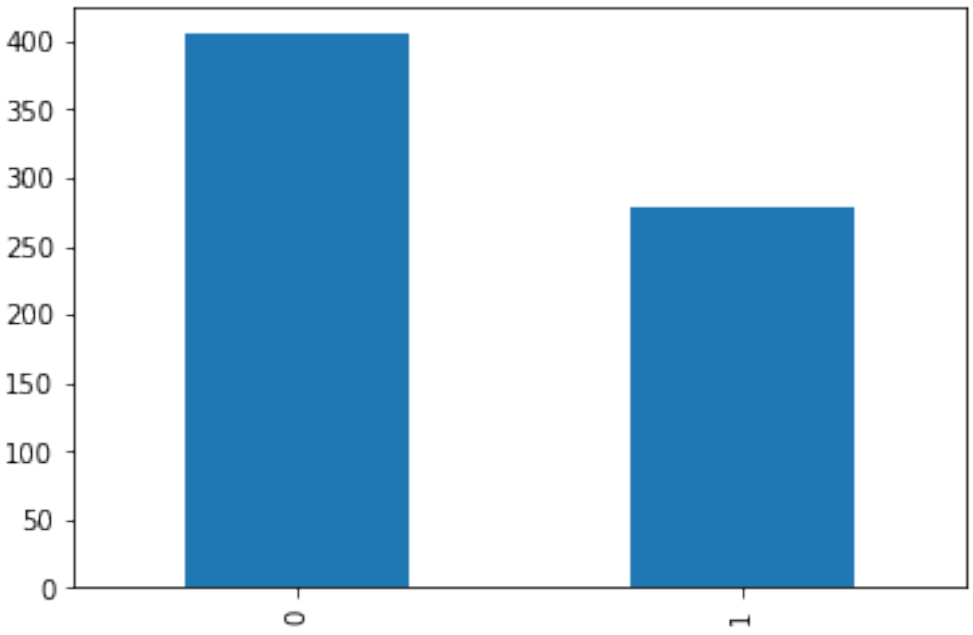
[ ]:
```
ks_clean.hist()
```

[ ]:
```
array([[<AxesSubplot:title={'center':'Unnamed: 0'}>,
        <AxesSubplot:title={'center':'survived'}>,
        <AxesSubplot:title={'center':'pclass'}>],
       [<AxesSubplot:title={'center':'age'}>,
        <AxesSubplot:title={'center':'sibsp'}>,
        <AxesSubplot:title={'center':'parch'}>],
       [<AxesSubplot:title={'center':'fare'}>, <AxesSubplot:>,
        <AxesSubplot:>]], dtype=object)
```



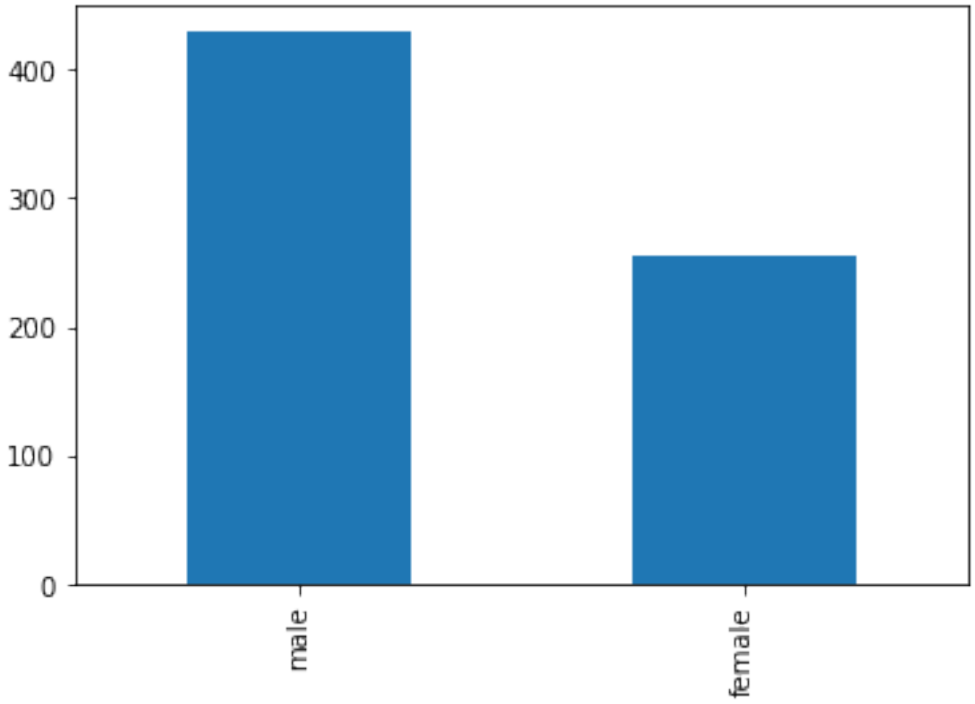
[ ]:
```
pd.value_counts(ks_clean['survived']).plot.bar()
```

[ ]: <AxesSubplot:>



[ ]:
```
pd.value_counts(ks_clean['sex']).plot.bar()
```

[ ]: <AxesSubplot:>



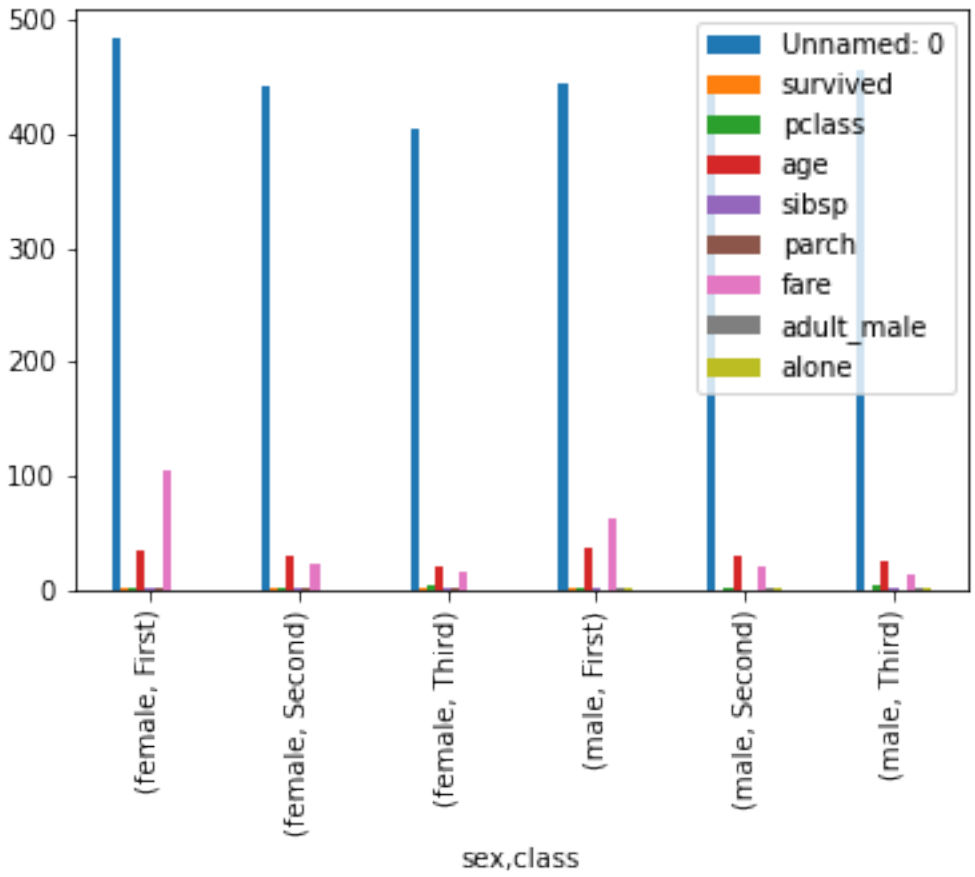
[ ]:
```
ks_clean.groupby(['sex','class']).mean()
```

[ ]:

sex	class	Unnamed: 0	survived	pclass	age	sibsp	parch	fare	adult_male	alone
female	First	484.750000	0.962500	1.0	33.550000	0.550000	0.525000	104.373699	0.000000	0.362500
	Second	441.905405	0.918919	2.0	28.722973	0.500000	0.621622	21.951070	0.000000	0.405405
	Third	404.732673	0.455446	3.0	21.341584	0.831683	0.960396	15.937625	0.000000	0.366337
male	First	444.541176	0.423529	1.0	37.440235	0.411765	0.305882	63.216519	0.964706	0.505882
	Second	447.631579	0.147368	2.0	29.319263	0.378947	0.242105	21.260000	0.905263	0.631579
	Third	455.196787	0.152610	3.0	25.847068	0.497992	0.261044	12.239556	0.887550	0.734940

[ ]:
```
ks_clean.groupby(['sex','class']).mean().plot.bar()
```

[ ]: <AxesSubplot:xlabel='sex,class'>



[ ]:
```
ks1 = kashti
ks1.groupby(['sex','class','who']).mean()
```

[ ]:

sex	class	who	Unnamed: 0	survived	pclass	age	sibsp
female	First	child	473.666667	0.666667	1.0	10.333333	0.666667
		man	NaN	NaN	NaN	NaN	NaN
		woman	468.032967	0.978022	1.0	35.500000	0.549451
	Second	child	394.600000	1.000000	2.0	6.600000	0.700000
		man	NaN	NaN	NaN	NaN	NaN
		woman	449.303030	0.909091	2.0	32.179688	0.454545
	Third	child	415.700000	0.533333	3.0	7.100000	1.533333
		man	NaN	NaN	NaN	NaN	NaN
		woman	394.263158	0.491228	3.0	27.854167	0.728070
male	First	child	517.333333	1.000000	1.0	5.306667	0.666667

```
                man    453.151261  0.352941     1.0  42.382653  0.302521
                woman         NaN       NaN     NaN        NaN       NaN
        Second  child  462.555556  1.000000     2.0   2.258889  0.888889
                man    445.545455  0.080808     2.0  33.588889  0.292929
                woman         NaN       NaN     NaN        NaN       NaN
        Third   child  435.250000  0.321429     3.0   6.515000  2.821429
                man    456.206897  0.119122     3.0  28.995556  0.294671
                woman         NaN       NaN     NaN        NaN       NaN
```

```
                          parch        fare  adult_male     alone
sex    class  who
female First  child  1.666667  160.962500         0.0  0.000000
              man         NaN         NaN         NaN       NaN
              woman  0.417582  104.317995         0.0  0.373626
       Second child  1.300000   29.240000         0.0  0.000000
              man         NaN         NaN         NaN       NaN
              woman  0.500000   20.868624         0.0  0.484848
       Third  child  1.100000   19.023753         0.0  0.166667
              man         NaN         NaN         NaN       NaN
              woman  0.719298   15.354351         0.0  0.482456
male   First  child  2.000000  117.802767         0.0  0.000000
              man    0.235294   65.951086         1.0  0.630252
              woman       NaN         NaN         NaN       NaN
       Second child  1.222222   27.306022         0.0  0.000000
              man    0.131313   19.054124         1.0  0.727273
              woman       NaN         NaN         NaN       NaN
       Third  child  1.321429   27.716371         0.0  0.035714
              man    0.128527   11.340213         1.0  0.824451
              woman       NaN         NaN         NaN       NaN
```

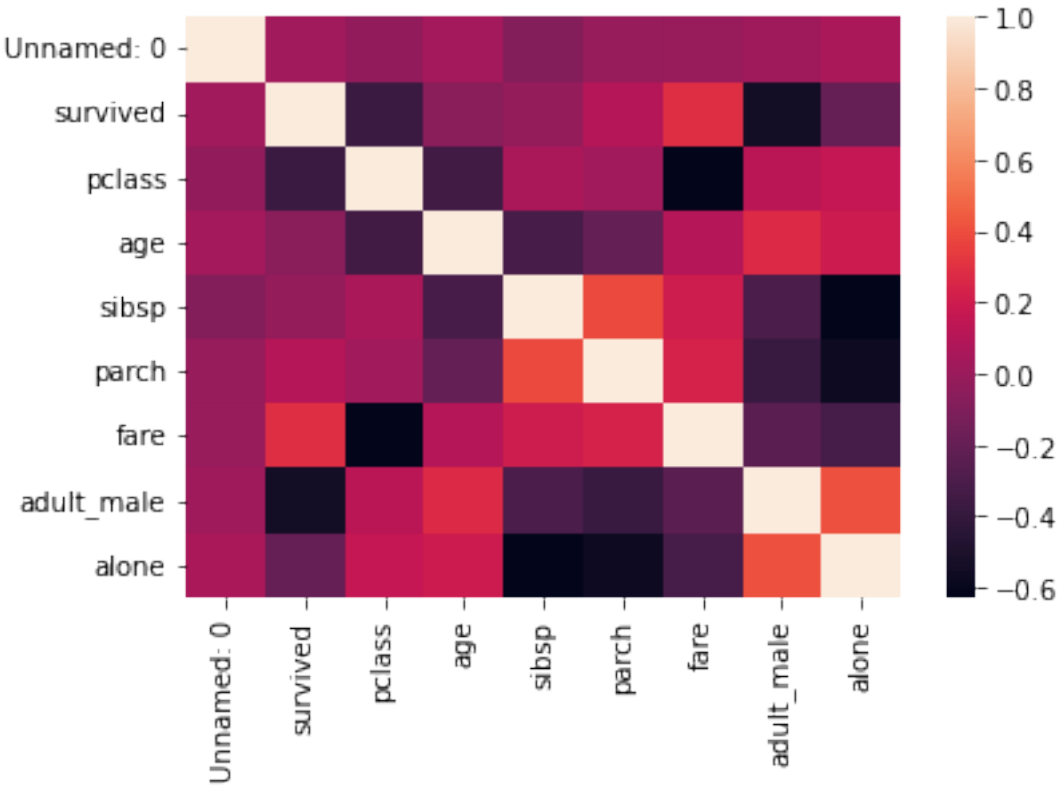
[ ]:
```
# Finding the relation ship
cor_ks_clean =ks_clean.corr()
cor_ks_clean
```

[ ]:
```
             Unnamed: 0  survived    pclass       age     sibsp     parch  \
Unnamed: 0     1.000000  0.024824 -0.034906  0.039508 -0.086529 -0.014790
survived       0.024824  1.000000 -0.376913 -0.062820 -0.021580  0.101012
pclass        -0.034906 -0.376913  1.000000 -0.342623  0.059466  0.027224
age            0.039508 -0.062820 -0.342623  1.000000 -0.318082 -0.202076
sibsp         -0.086529 -0.021580  0.059466 -0.318082  1.000000  0.381742
parch         -0.014790  0.101012  0.027224 -0.202076  0.381742  1.000000
fare          -0.008363  0.284657 -0.626093  0.091596  0.195031  0.234899
adult_male     0.019542 -0.550647  0.120975  0.265445 -0.309017 -0.379839
alone          0.063871 -0.196596  0.159555  0.190447 -0.627571 -0.574854
fare_log      -0.013633  0.344567 -0.767125  0.116841  0.321557  0.332127

                fare  adult_male     alone  fare_log
```

```
Unnamed: 0 -0.008363    0.019542  0.063871 -0.013633
survived    0.284657   -0.550647 -0.196596  0.344567
pclass     -0.626093    0.120975  0.159555 -0.767125
age         0.091596    0.265445  0.190447  0.116841
sibsp       0.195031   -0.309017 -0.627571  0.321557
parch       0.234899   -0.379839 -0.574854  0.332127
fare        1.000000   -0.240071 -0.326577  0.870383
adult_male -0.240071    1.000000  0.402767 -0.315856
alone      -0.326577    0.402767  1.000000 -0.494042
fare_log    0.870383   -0.315856 -0.494042  1.000000
```

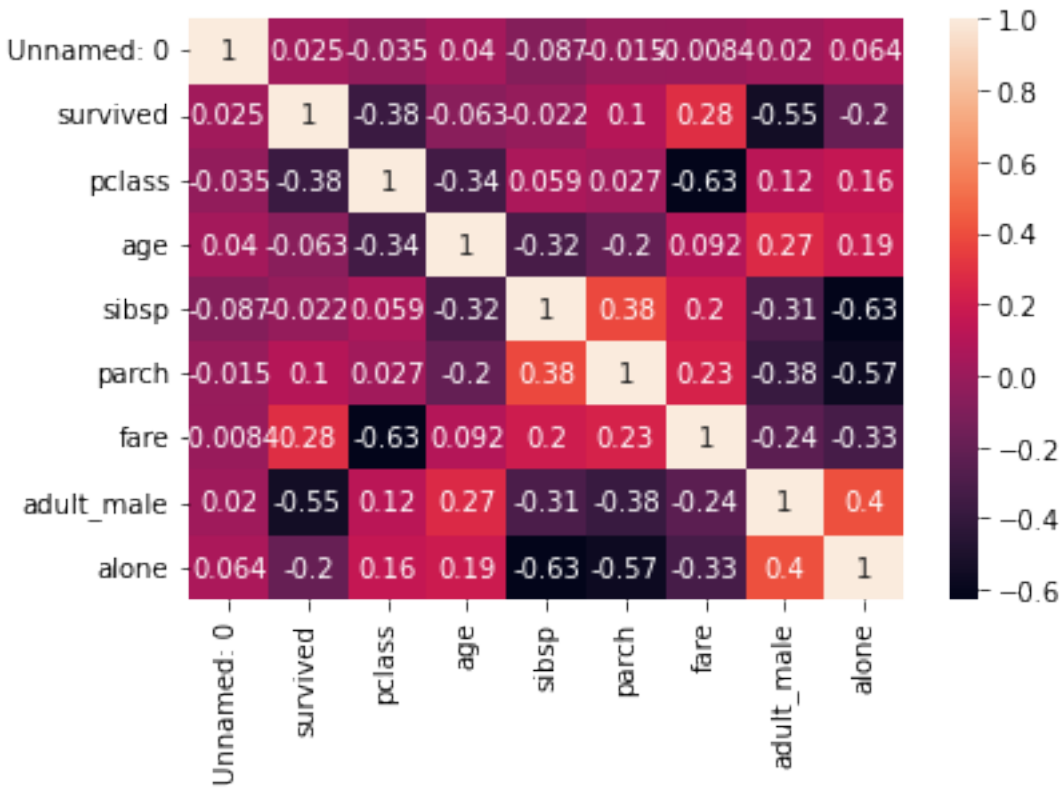
[ ]:
```
sns.heatmap(cor_ks_clean)
```

[ ]: <AxesSubplot:>



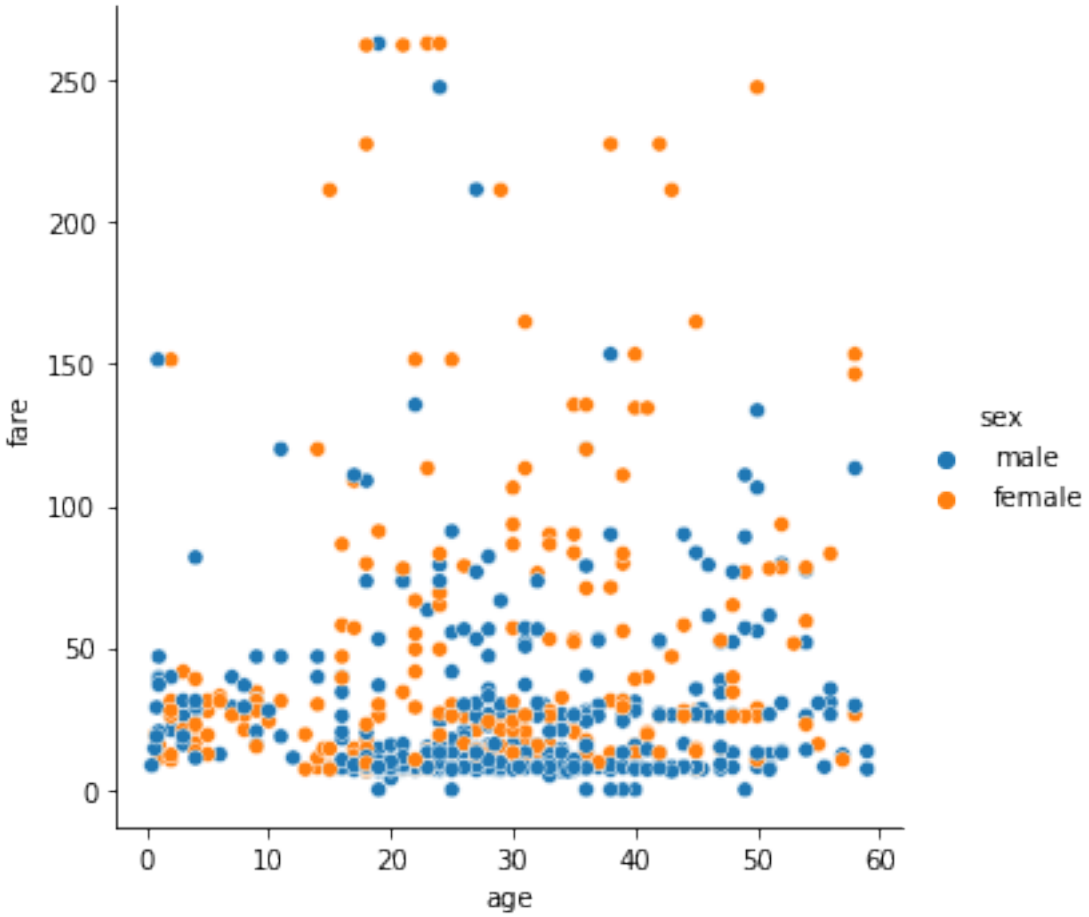
[ ]:
```
sns.heatmap(cor_ks_clean , annot=True)
```

[ ]: <AxesSubplot:>



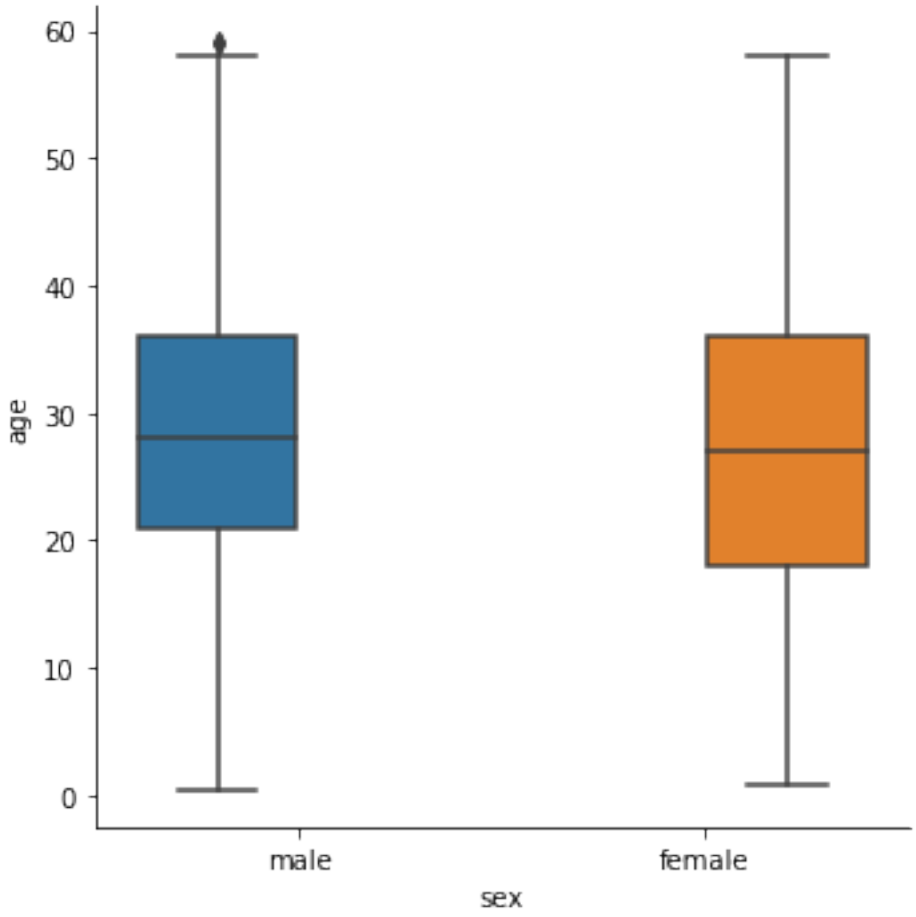
[ ]:
```
sns.relplot(x='age',y='fare',hue='sex',data=ks_clean)
```

[ ]:
```
<seaborn.axisgrid.FacetGrid at 0x1dcb0d44070>
```



[ ]:
```
sns.catplot(x='sex',y='age',hue='sex',data=ks_clean, kind='box')
```

[ ]:
```
<seaborn.axisgrid.FacetGrid at 0x1dcafbece20>
```



[ ]:
```
# Log transformation
ks_clean['fare_log']=np.log(ks_clean['fare'])
ks_clean.head(5)
```

```
C:\Users\Sartaj\AppData\Local\Programs\Python\Python39\lib\site-
packages\pandas\core\arraylike.py:358: RuntimeWarning: divide by zero
encountered in log
  result = getattr(ufunc, method)(*inputs, **kwargs)
```

[ ]:

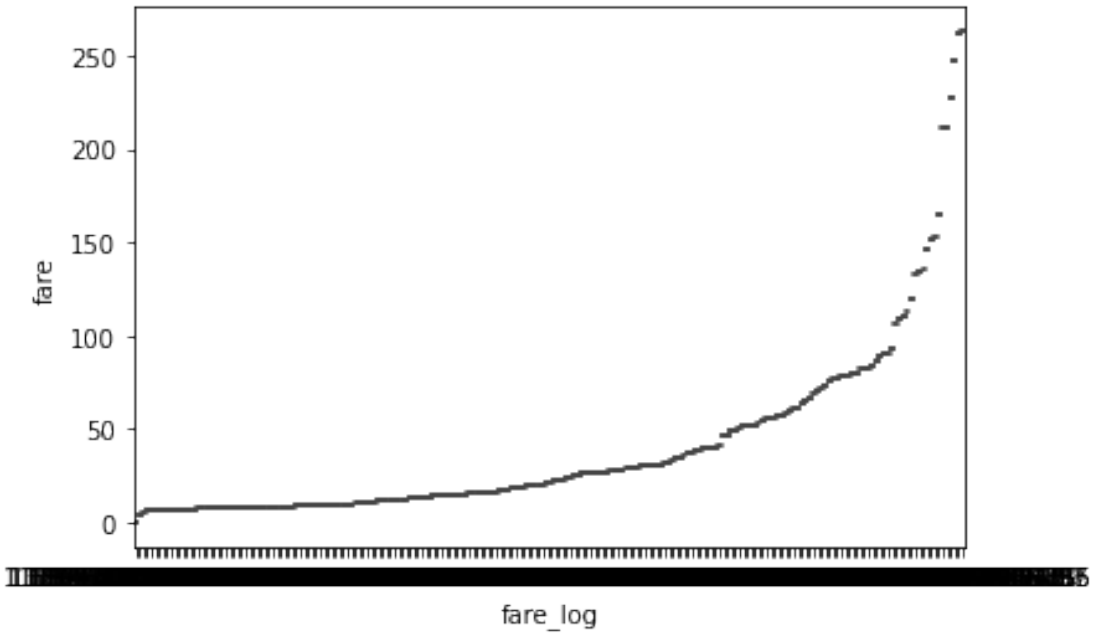
	Unnamed: 0	survived	pclass	sex	age	sibsp	parch	fare	embarked	\
0	0	0	3	male	22.0	1	0	7.2500	S	
1	1	1	1	female	38.0	1	0	71.2833	C	
2	2	1	3	female	26.0	0	0	7.9250	S	
3	3	1	1	female	35.0	1	0	53.1000	S	
4	4	0	3	male	35.0	0	0	8.0500	S	

	class	who	adult_male	embark_town	alive	alone	fare_log
0	Third	man	True	Southampton	no	False	1.981001

```
1  First  woman       False    Cherbourg   yes  False  4.266662
2  Third  woman       False  Southampton   yes   True  2.070022
3  First  woman       False  Southampton   yes  False  3.972177
4  Third    man        True  Southampton    no   True  2.085672
```

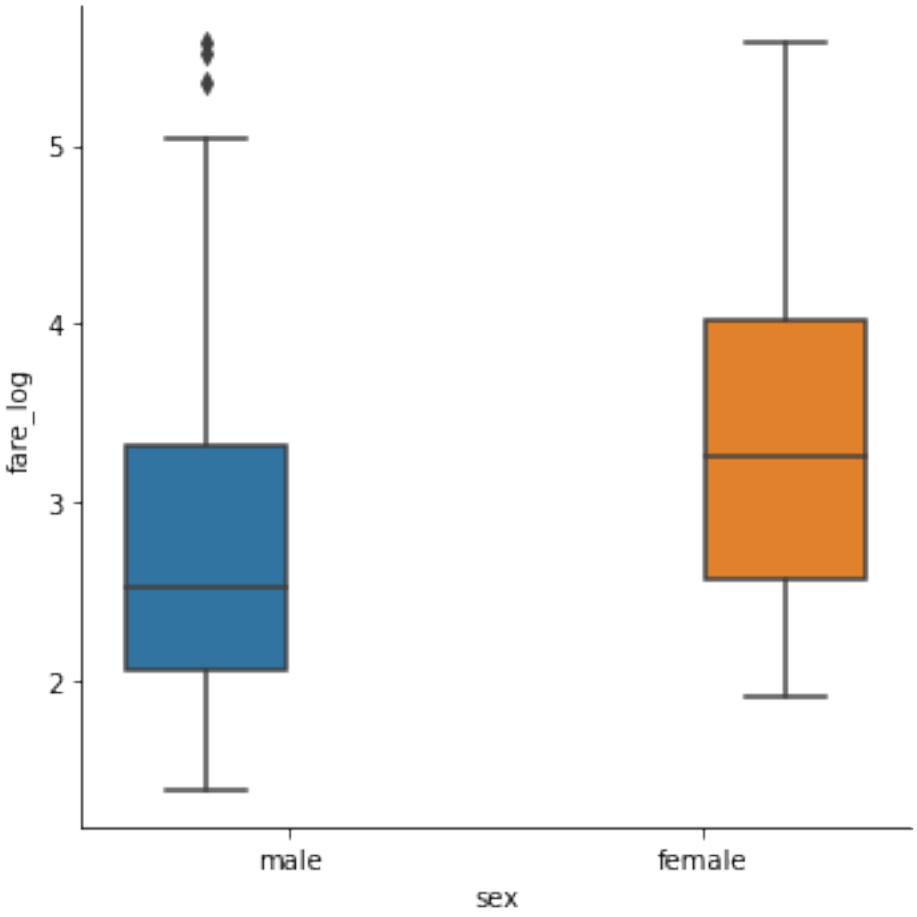
[ ]:
```
sns.boxplot(x='fare_log',y='fare',data=ks_clean)
```

[ ]:
```
<AxesSubplot:xlabel='fare_log', ylabel='fare'>
```



[ ]:
```
sns.catplot(x='sex',y='fare_log',hue='sex',data=ks_clean,kind='box')
```

[ ]:
```
<seaborn.axisgrid.FacetGrid at 0x1dcb238ad90>
```


5
4
3
2
fare_log
male
female
sex

[ ]: