EDA

January 15, 2022

1 Exploratory Data Analysis (EDA)

1.0.1 Three important steps

- Understand the data
- Clean the data
- Find a relationship between data

[ ]:
```
# important libraries
import pandas as pd
import numpy as np
import matplotlib as plt
import seaborn as sns
```

[ ]:
```
kashti= sns.load_dataset('titanic')
```

[ ]:
```
kashti.to_csv('kashti.csv')
```

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

[ ]:
```
ks.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
```

[ ]:
```
#to look headings/type of data
ks.head()
```

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

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

[ ]:
```
#To look rows and column of data
ks.shape
```

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

[ ]:
```
#give mean,std,median of numerical value
ks.describe()
```

[ ]:
```
         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
```

[ ]:
```
# to find unique value (type, categorial, numeric, etc)
ks.nunique()
```

[ ]:
```
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
```

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

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

[ ]:
```
#Unique value of specific value
ks['sex'].unique()
```

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

[ ]:
```
ks[["who","survived","age","fare"]].nunique()
```

[ ]:
```
who           3
survived      2
age          88
fare        248
dtype: int64
```

[ ]:
```
pd.concat([ks['survived'],ks['age'],ks['class'],ks['survived']]).unique()
```

[ ]:
```
array([0, 1, 22.0, 38.0, 26.0, 35.0, nan, 54.0, 2.0, 27.0, 14.0, 4.0,
       58.0, 20.0, 39.0, 55.0, 31.0, 34.0, 15.0, 28.0, 8.0, 19.0, 40.0,
       66.0, 42.0, 21.0, 18.0, 3.0, 7.0, 49.0, 29.0, 65.0, 28.5, 5.0,
       11.0, 45.0, 17.0, 32.0, 16.0, 25.0, 0.83, 30.0, 33.0, 23.0, 24.0,
       46.0, 59.0, 71.0, 37.0, 47.0, 14.5, 70.5, 32.5, 12.0, 9.0, 36.5,
       51.0, 55.5, 40.5, 44.0, 61.0, 56.0, 50.0, 36.0, 45.5, 20.5, 62.0,
       41.0, 52.0, 63.0, 23.5, 0.92, 43.0, 60.0, 10.0, 64.0, 13.0, 48.0,
       0.75, 53.0, 57.0, 80.0, 70.0, 24.5, 6.0, 0.67, 30.5, 0.42, 34.5,
       74.0, 'Third', 'First', 'Second'], dtype=object)
```

2 Cleaning and filtering the data

[ ]:
```
# find missing values inside
ks.isnull()
```

[ ]:
```
     survived  pclass    sex    age  sibsp  parch   fare  embarked  class  \
0       False   False  False  False  False  False  False     False  False
1       False   False  False  False  False  False  False     False  False
2       False   False  False  False  False  False  False     False  False
3       False   False  False  False  False  False  False     False  False
4       False   False  False  False  False  False  False     False  False
..        …     …    …    …    …    …       …    …
886     False   False  False  False  False  False  False     False  False
887     False   False  False  False  False  False  False     False  False
888     False   False  False   True  False  False  False     False  False
889     False   False  False  False  False  False  False     False  False
890     False   False  False  False  False  False  False     False  False

       who  adult_male   deck  embark_town  alive  alone
0    False       False   True        False  False  False
1    False       False  False        False  False  False
2    False       False   True        False  False  False
3    False       False  False        False  False  False
4    False       False   True        False  False  False
..     …         …    …          …    …    …
886  False       False   True        False  False  False
887  False       False  False        False  False  False
888  False       False   True        False  False  False
889  False       False  False        False  False  False
890  False       False   True        False  False  False

[891 rows x 15 columns]
```

[ ]:
```
ks.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
```

[ ]:
```
# drop/remove whole missing value column
ks_clean= ks.drop(['deck'], axis=1)
ks_clean.head()
```

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

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

[ ]:
```
ks_clean.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
embark_town      2
alive            0
alone            0
dtype: int64
```

[ ]:
```
891-177
```

[ ]:
```
714
```

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

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

```
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_clean.shape
```

```
(712, 14)
```

```
ks.shape
```

```
(891, 15)
```

```
ks_clean['sex'].value_counts()
```

```
male      453
female    259
Name: sex, dtype: int64
```

```
#difference between both data before and after cleaning
ks.describe()
```

```
         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
```

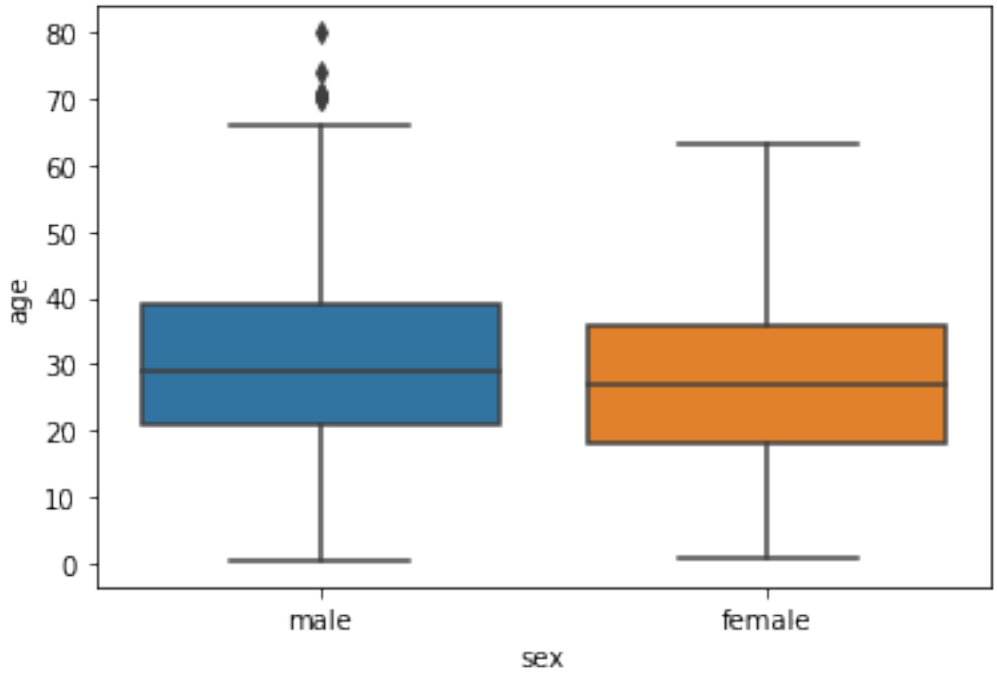
```
ks_clean.describe()
```

```
         survived      pclass         age       sibsp       parch        fare
count  712.000000  712.000000  712.000000  712.000000  712.000000  712.000000
mean     0.404494    2.240169   29.642093    0.514045    0.432584   34.567251
```

std	0.491139	0.836854	14.492933	0.930692	0.854181	52.938648
min	0.000000	1.000000	0.420000	0.000000	0.000000	0.000000
25%	0.000000	1.000000	20.000000	0.000000	0.000000	8.050000
50%	0.000000	2.000000	28.000000	0.000000	0.000000	15.645850
75%	1.000000	3.000000	38.000000	1.000000	1.000000	33.000000
max	1.000000	3.000000	80.000000	5.000000	6.000000	512.329200

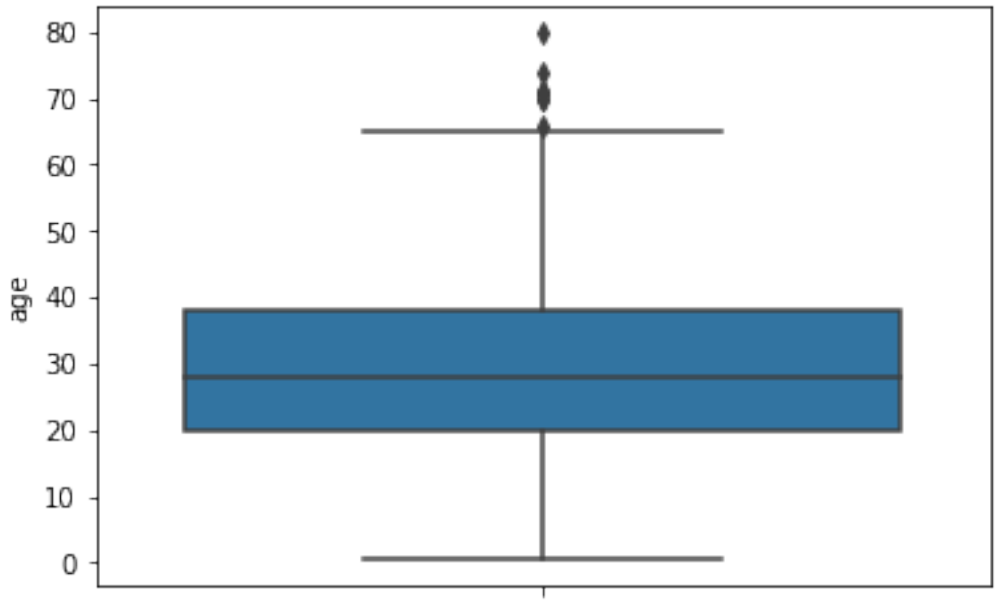
[ ]:
```
#analysing outliers
sns.boxplot(x='sex', y='age', data=ks_clean)
```

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



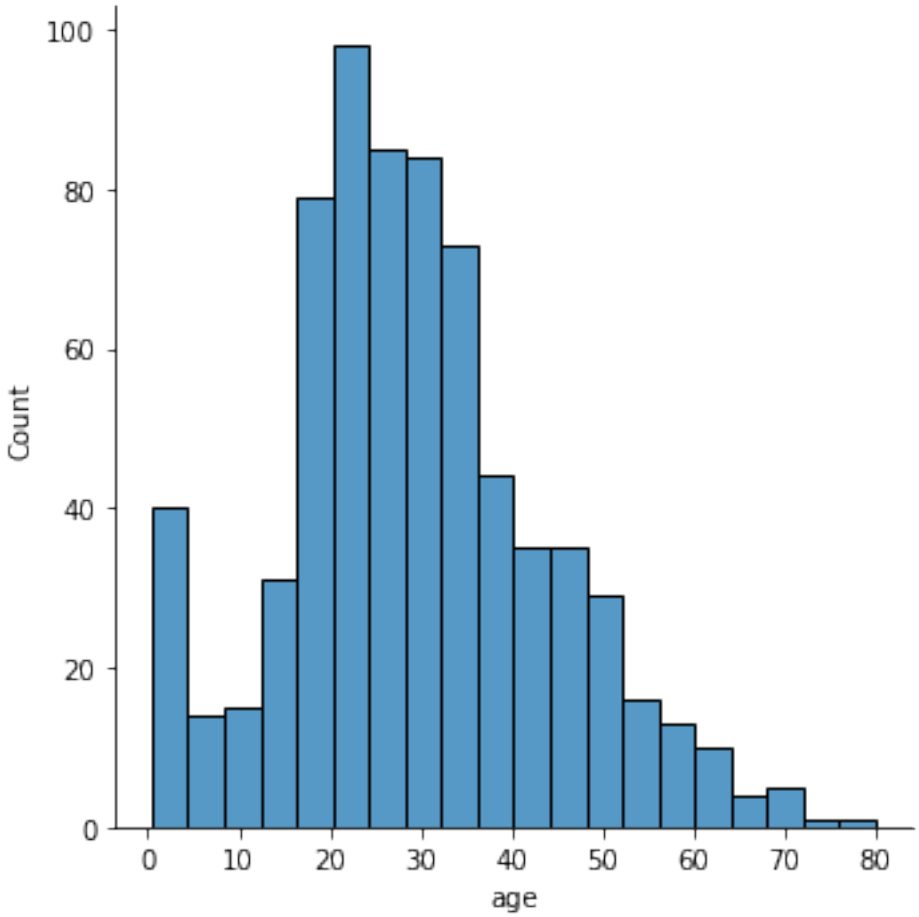
[ ]:
```
sns.boxplot(y='age', data=ks_clean)
```

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



[ ]:
```
#displot or distplot to look bell curve and outlier effects
sns.displot(ks_clean['age'])
```

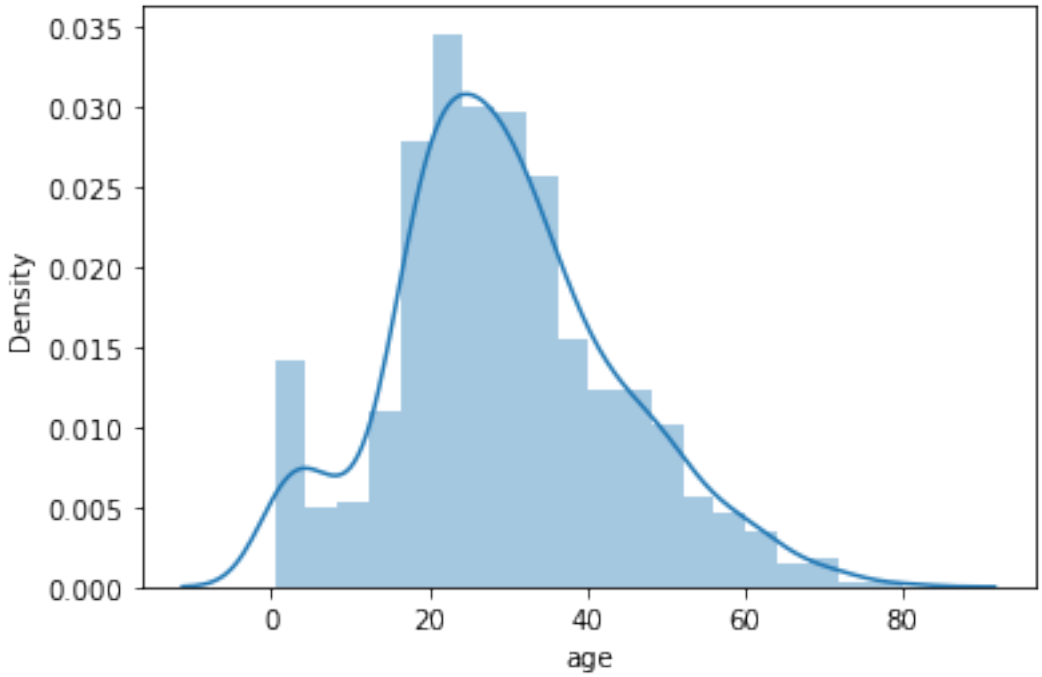
[ ]: <seaborn.axisgrid.FacetGrid at 0x1980753b580>



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

```
C:\Anaconda\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'>
```



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

[ ]:
```
29.64209269662921
```

[ ]:
```
#remove outlier from specific column like giving range
ks1=ks_clean[ks_clean['age']<68]
ks1.head()
```

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

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

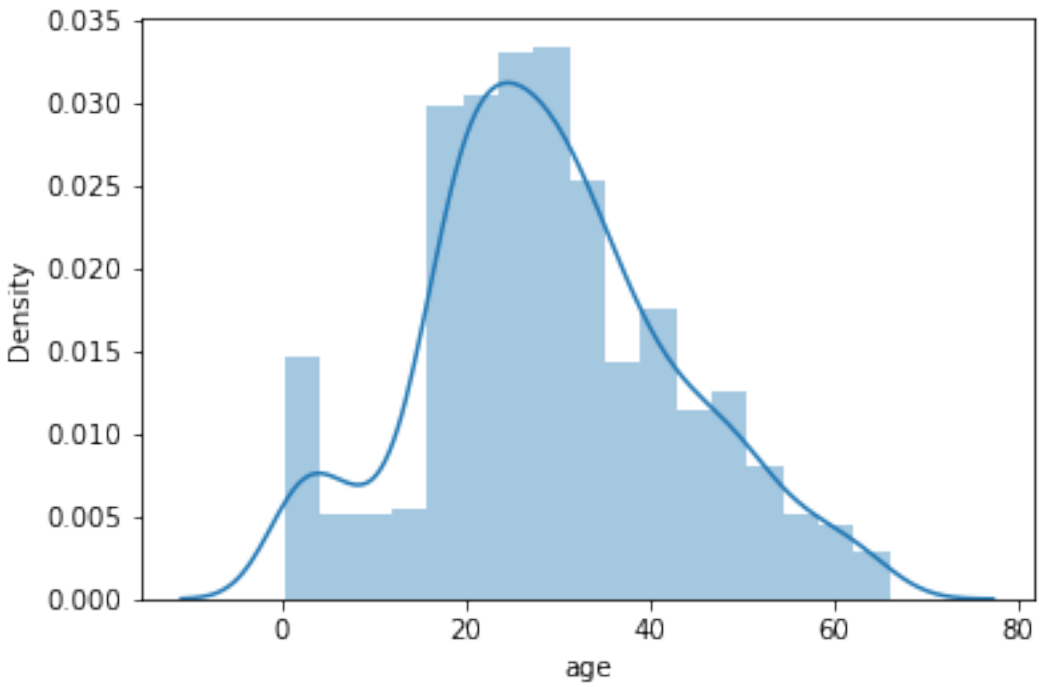
[ ]:
```
ks1.shape
```

[ ]:
```
(705, 14)
```

[ ]:
```
sns.distplot(ks1['age'])
```

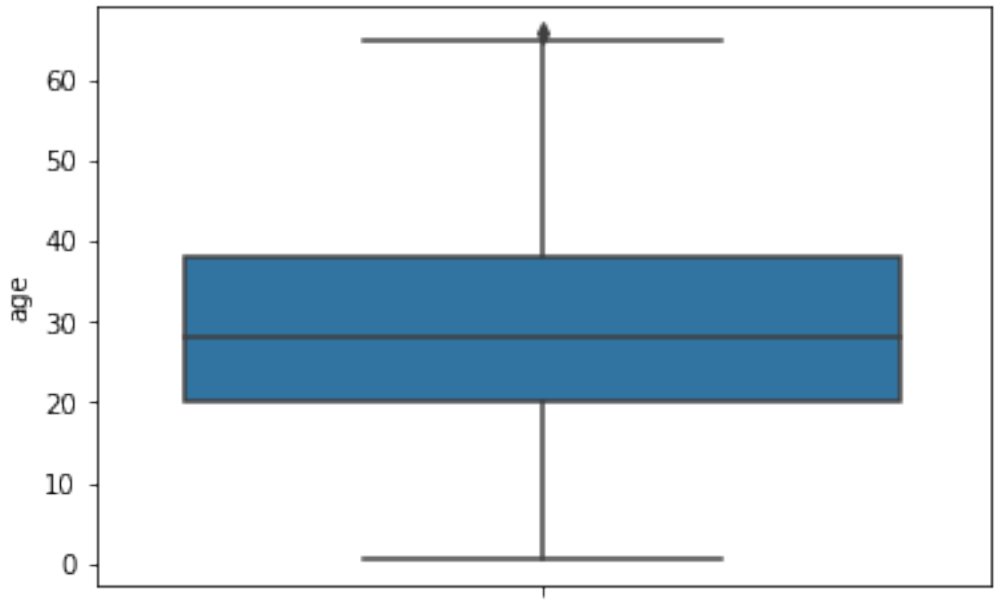
```
C:\Anaconda\lib\site-packages\seaborn\distributions.py:2619: FutureWarning:
`distplot` is a deprecated function and will be removed in a future version.
Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).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

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



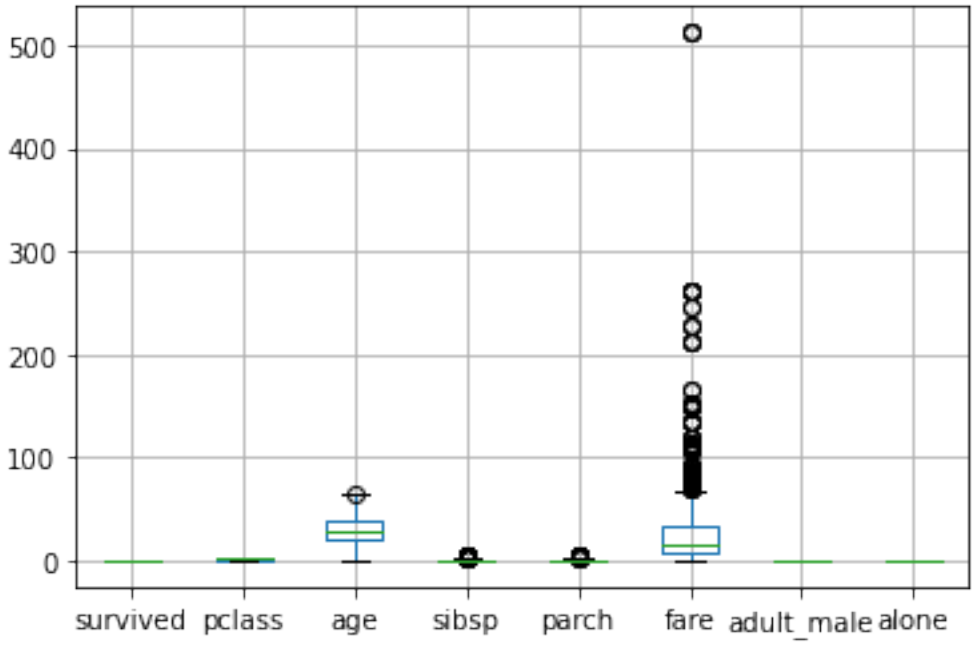
[ ]:
```
sns.boxplot(y='age', data=ks1)
```

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



[ ]:
```
ks1.boxplot()
```

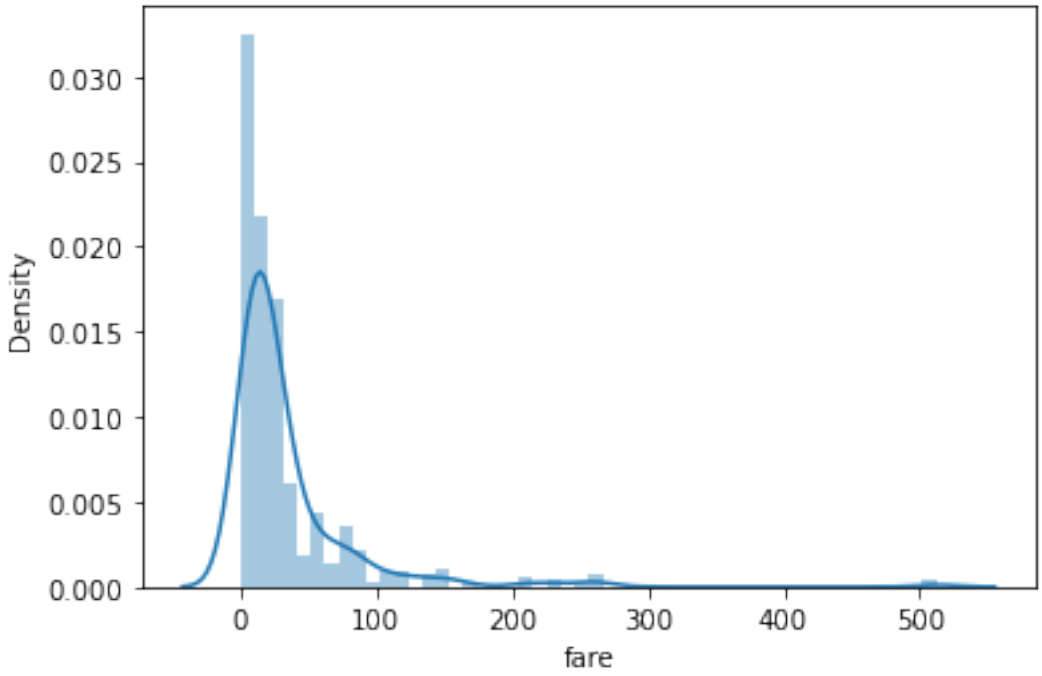
[ ]: <AxesSubplot:>



[ ]:
```
sns.distplot(ks1['fare'])
```

```
C:\Anaconda\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'>
```



[ ]:
```
#log transformation
ks1['fare_log']=np.log(ks1['fare'])
```

```
C:\Anaconda\lib\site-packages\pandas\core\arraylike.py:364: RuntimeWarning:
divide by zero encountered in log
  result = getattr(ufunc, method)(*inputs, **kwargs)
C:\Users\masha\AppData\Local\Temp/ipykernel_106036/3103475165.py:1:
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-a-copy
  ks1['fare_log']=np.log(ks1['fare'])
```

[ ]:
```
ks1.head()
```

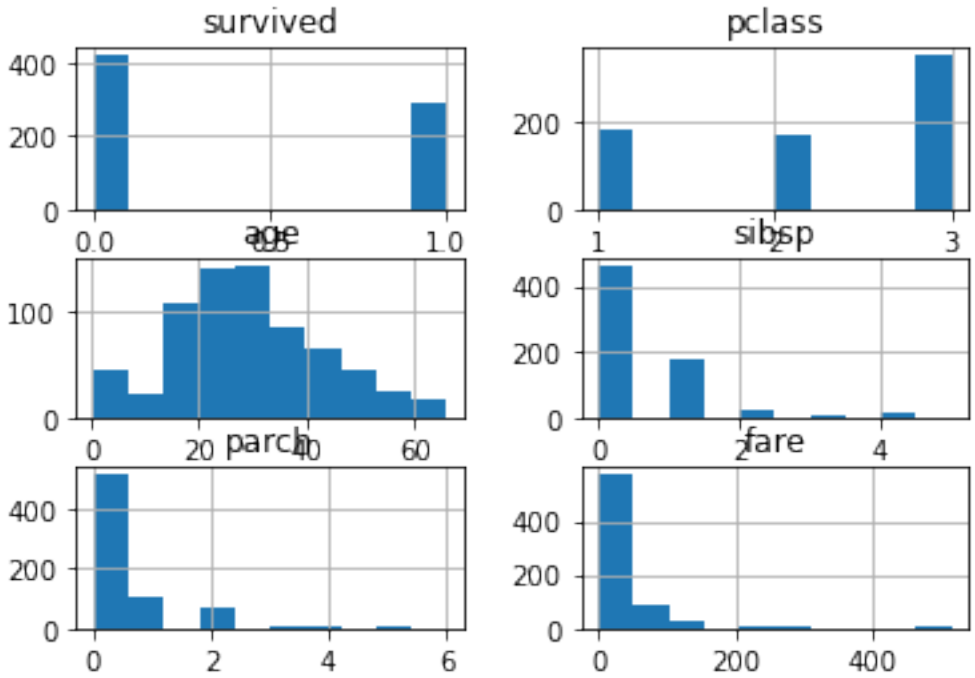
[ ]:

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

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

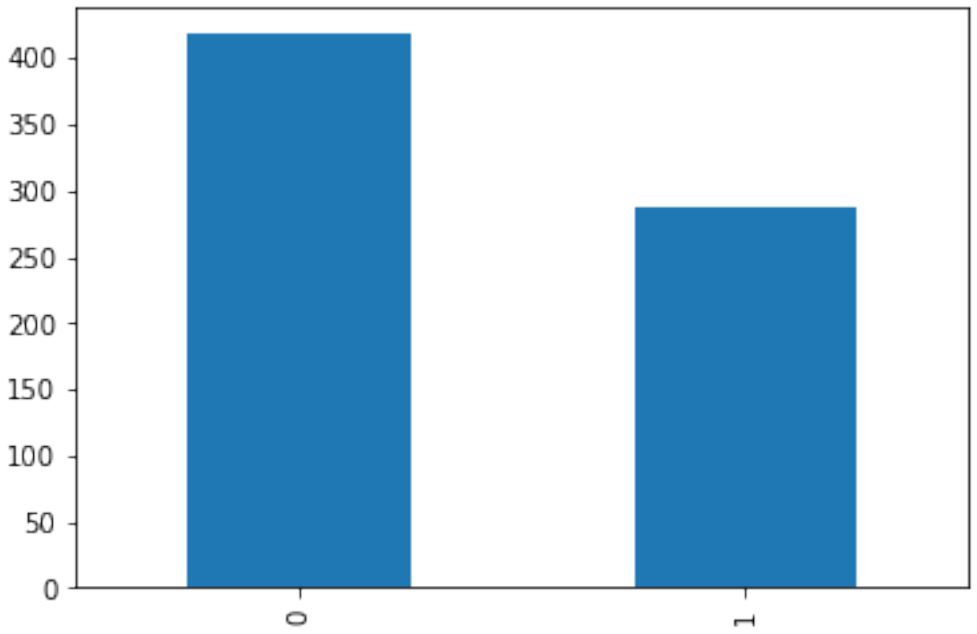
[ ]:
```
#histogram of whole dataset
ks1.hist()
```

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



[ ]:
```
#using pandas function to draw specific bar plot
pd.value_counts(ks1['survived']).plot.bar()
```

[ ]: <AxesSubplot:>



[ ]:
```
#groupby function
ks1.groupby(['sex','class']).mean()
```

[ ]:

sex	class	survived	pclass	age	sibsp	parch	fare	adult_male	alone
female	First	0.963855	1.0	34.240964	0.554217	0.506024	108.619680	0.000000	0.361446
	Second	0.918919	2.0	28.722973	0.500000	0.621622	21.951070	0.000000	0.405405
	Third	0.460784	3.0	21.750000	0.823529	0.950980	15.875369	0.000000	0.372549
male	First	0.402062	1.0	39.973402	0.381443	0.340206	72.167655	0.969072	0.525773
	Second	0.153061	2.0	30.340102	0.377551	0.244898	21.221429	0.908163	0.632653
	Third	0.151394	3.0	26.143108	0.494024	0.258964	12.197757	0.888446	0.737052

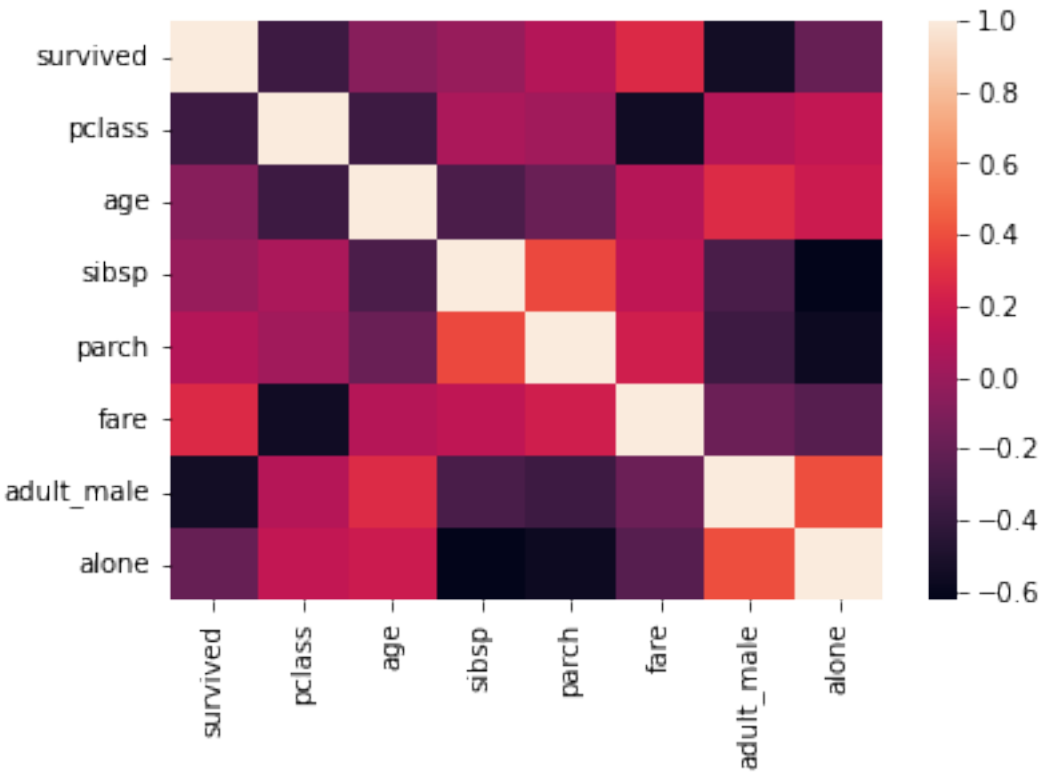
[ ]:
```
# relationship in data
# finding co-relation in the data through matrix
ks2= ks1.corr()
ks2
```

[ ]:
```
            survived    pclass       age     sibsp     parch      fare  \
survived    1.000000 -0.361441 -0.071804 -0.017289  0.094449  0.266954
pclass     -0.361441  1.000000 -0.366032  0.064561  0.023157 -0.554566
age        -0.071804 -0.366032  1.000000 -0.309617 -0.186213  0.100263
sibsp      -0.017289  0.064561 -0.309617  1.000000  0.381577  0.138697
parch       0.094449  0.023157 -0.186213  0.381577  1.000000  0.205546
fare        0.266954 -0.554566  0.100263  0.138697  0.205546  1.000000
adult_male -0.550780  0.101061  0.274782 -0.311226 -0.364533 -0.177542
alone      -0.199052  0.153622  0.187088 -0.628019 -0.575487 -0.261454

            adult_male     alone
survived     -0.550780 -0.199052
pclass        0.101061  0.153622
age           0.274782  0.187088
sibsp        -0.311226 -0.628019
parch        -0.364533 -0.575487
fare         -0.177542 -0.261454
adult_male    1.000000  0.398833
alone         0.398833  1.000000
```

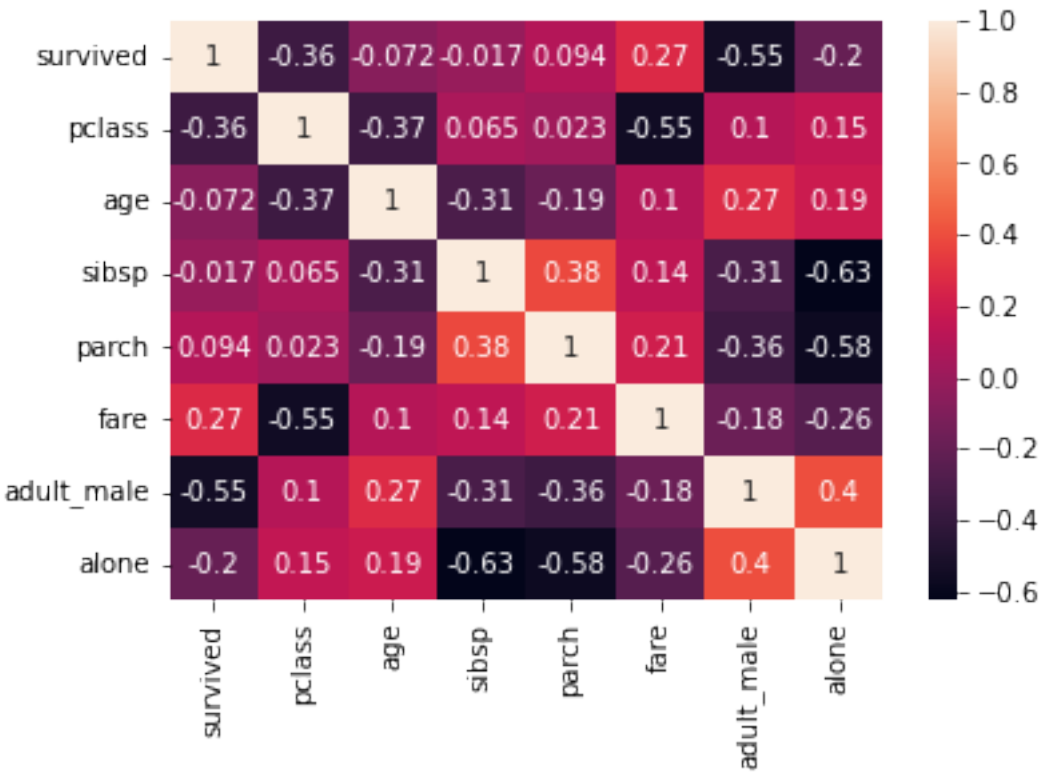
[ ]:
```
# heatmap of correlation matrix
sns.heatmap(ks2)
```

[ ]:
```
<AxesSubplot:>
```



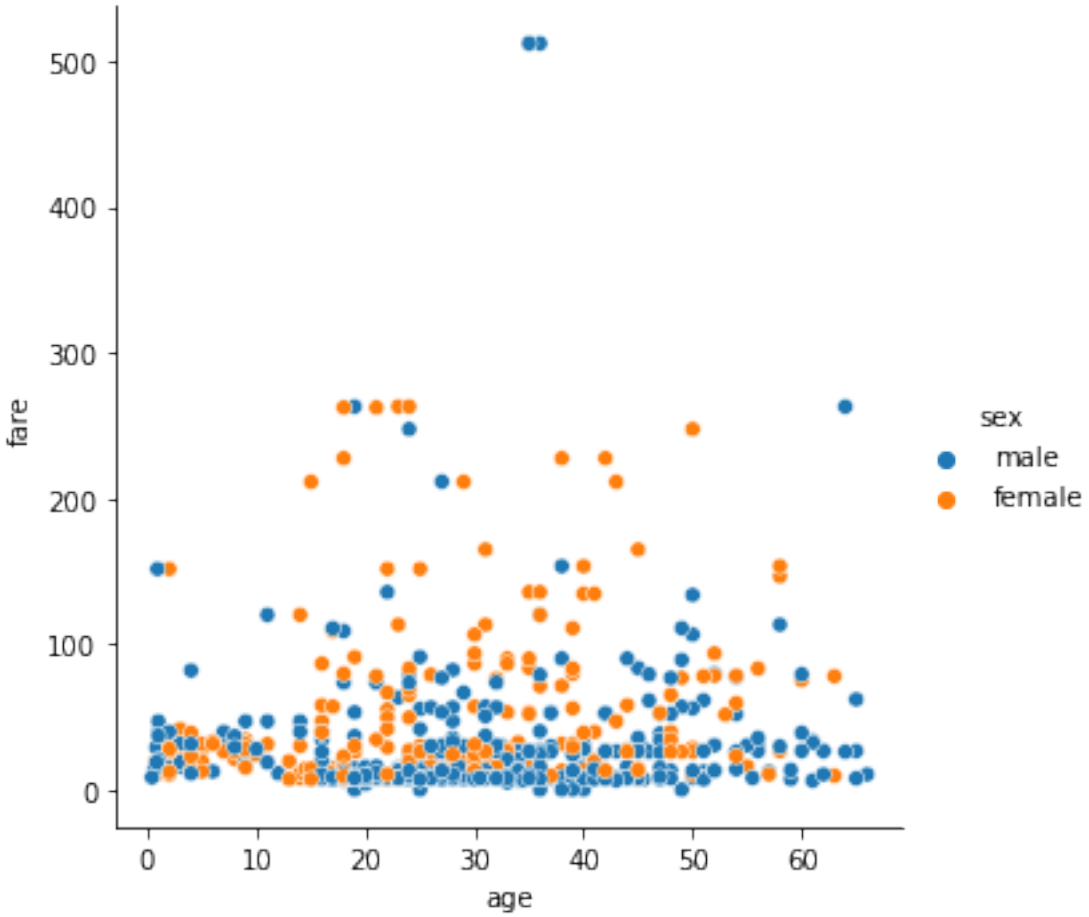
[ ]:
```
#with annotation in the box
sns.heatmap(ks2, annot=True)
```

[ ]: <AxesSubplot:>



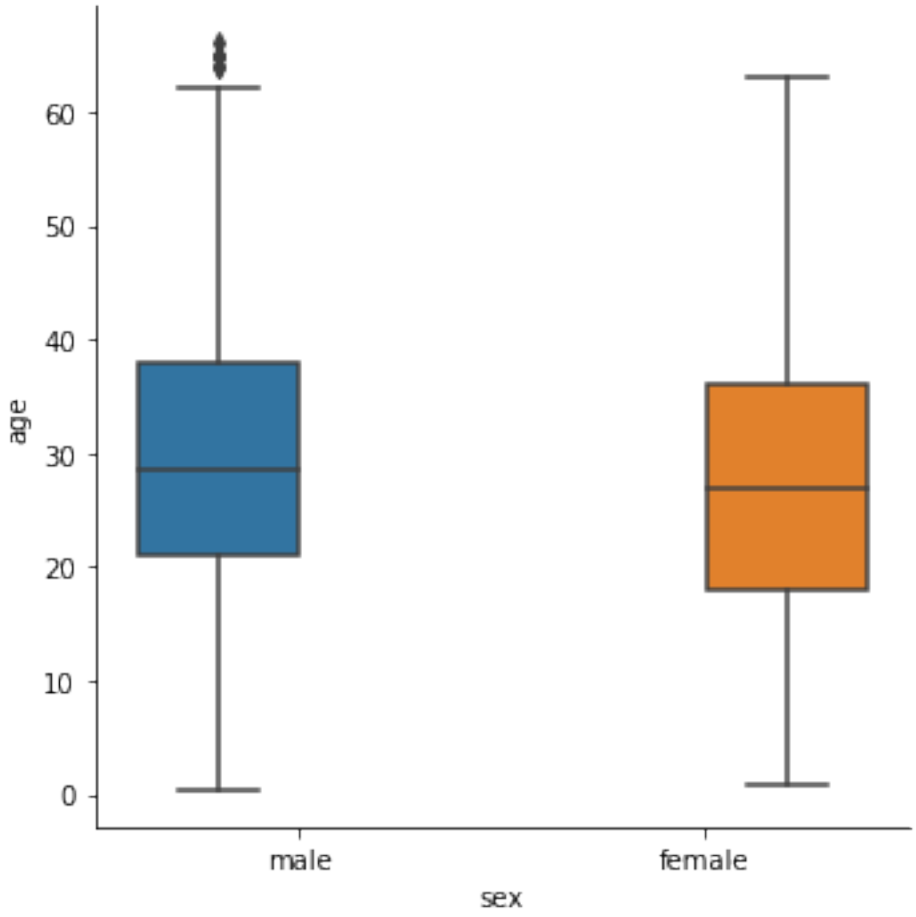
[ ]:
```
sns.relplot(x='age',y='fare', hue='sex', data=ks1)
```

[ ]:
```
<seaborn.axisgrid.FacetGrid at 0x19808c8e610>
```



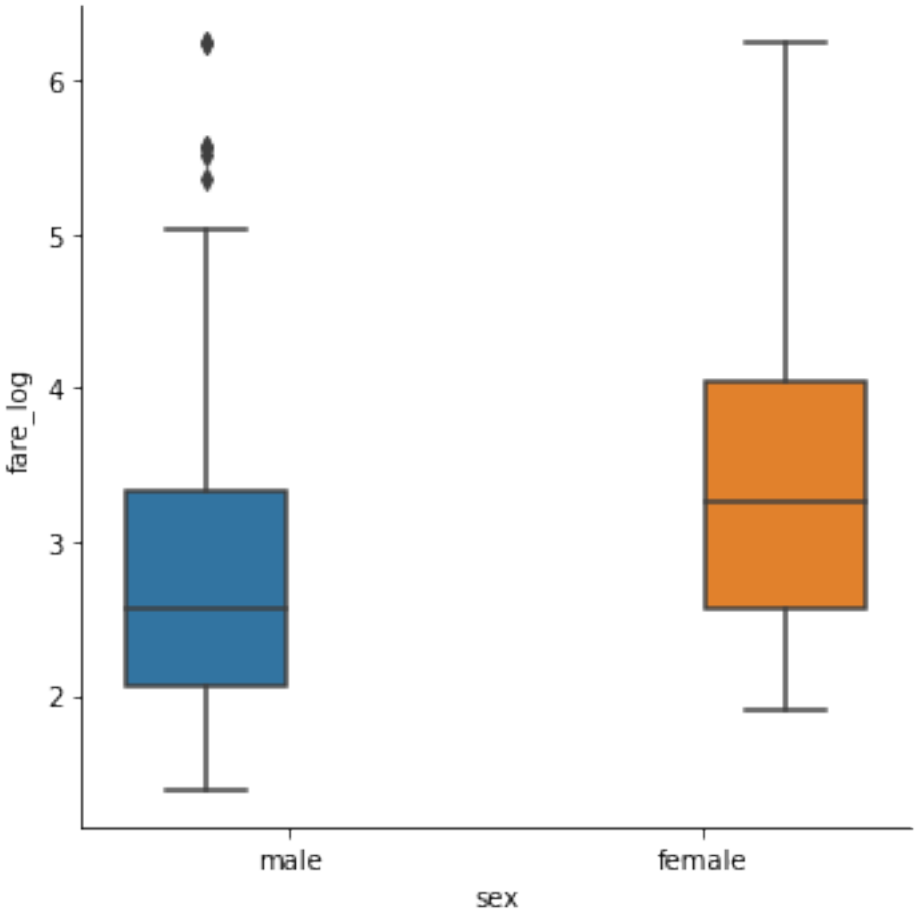
[ ]:
```
# Catogery plot
sns.catplot(x='sex', y='age', hue='sex', data=ks1, kind='box')
```

[ ]: <seaborn.axisgrid.FacetGrid at 0x19807a0c340>



[ ]:
```
# after adding calculating and adding log fare, plot looks much better
sns.catplot(x='sex', y='fare_log', hue='sex', data=ks1, kind='box')
```

[ ]: <seaborn.axisgrid.FacetGrid at 0x198090b4820>


6
5
4
3
2
fare_log
male
female
sex