eda

January 14, 2022

1 Exploratory Data Analysis

This will show us how we can do EDA using python

1.1 Three important steps to keep in mind are:

1- Understand data

2- Clean the data

3- Find a relationship between data

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

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

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

[ ]:
```
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
```

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

[ ]:
```
# Data set viewing
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
```

[ ]:
```
# rows and columns
ks.shape
```

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

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

[ ]:
```
     survived  pclass     sex   age  sibsp  parch   fare embarked   class  \
886         0       2    male  27.0      0      0  13.00        S  Second
887         1       1  female  19.0      0      0  30.00        S   First
888         0       3  female   NaN      1      2  23.45        S   Third
889         1       1    male  26.0      0      0  30.00        C   First
890         0       3    male  32.0      0      0   7.75        Q   Third

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

[ ]:
```
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
```

[ ]:
```
# unique values
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')
```

[ ]:
```
ks["sex"].unique()
```

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

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

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

[ ]:
```
# Assignment
# # For viewing all unique values of columns together
pd.Series({col:ks[col].unique() for col in ks})
```

[ ]:
```
survived                                                    [0, 1]
pclass                                                   [3, 1, 2]
sex                                                 [male, female]
age           [22.0, 38.0, 26.0, 35.0, nan, 54.0, 2.0, 27.0,…
sibsp                                        [1, 0, 3, 4, 2, 5, 8]
parch                                        [0, 1, 2, 5, 3, 4, 6]
fare          [7.25, 71.2833, 7.925, 53.1, 8.05, 8.4583, 51…
embarked                                            [S, C, Q, nan]
class         ['Third', 'First', 'Second']
Categories (3, ob…
who                                          [man, woman, child]
adult_male                                         [True, False]
deck          [NaN, 'C', 'E', 'G', 'D', 'A', 'B', 'F']
Categ…
embark_town          [Southampton, Cherbourg, Queenstown, nan]
alive                                                 [no, yes]
alone                                              [False, True]
dtype: object
```

Cleaning and filtering the daya

[ ]:
```
# find the missing values
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
```

[ ]:
```
# removing missing value colum (cleaning data)
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
```

[ ]:
```
# value droping extra 2 droped from embarked
ks_clean2= ks_clean.dropna()
```

[ ]:
```
ks_clean2.shape
```

[ ]:
```
(712, 14)
```

[ ]:
```
ks_clean2.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_clean2["sex"].value_counts()
```

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

[ ]:
```
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_clean2.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

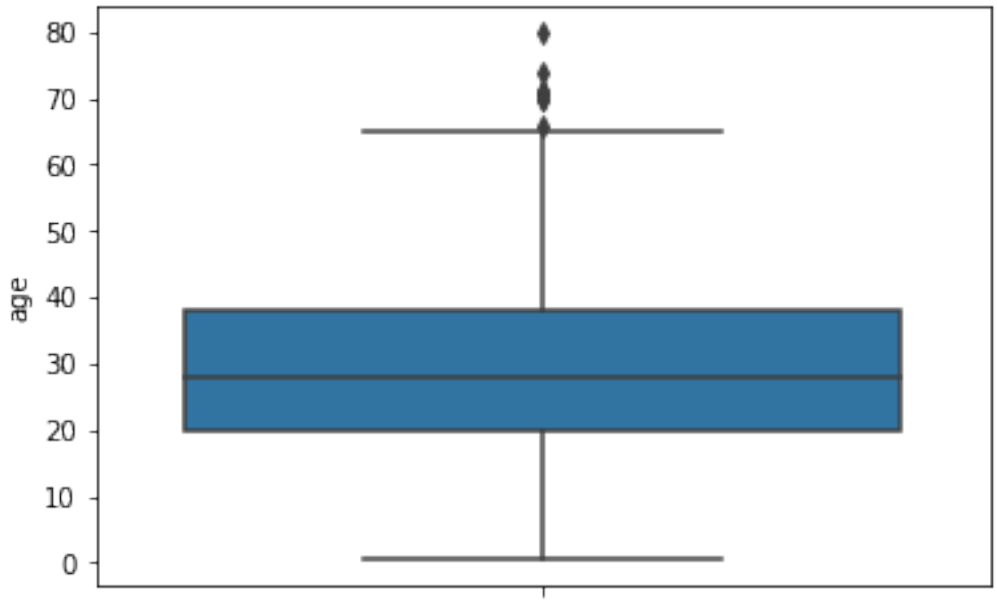
[ ]:
```
ks_clean2.columns
```

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

```
dtype='object')
```

[ ]:
```
sns.boxplot(y="age", data=ks_clean2 )
```

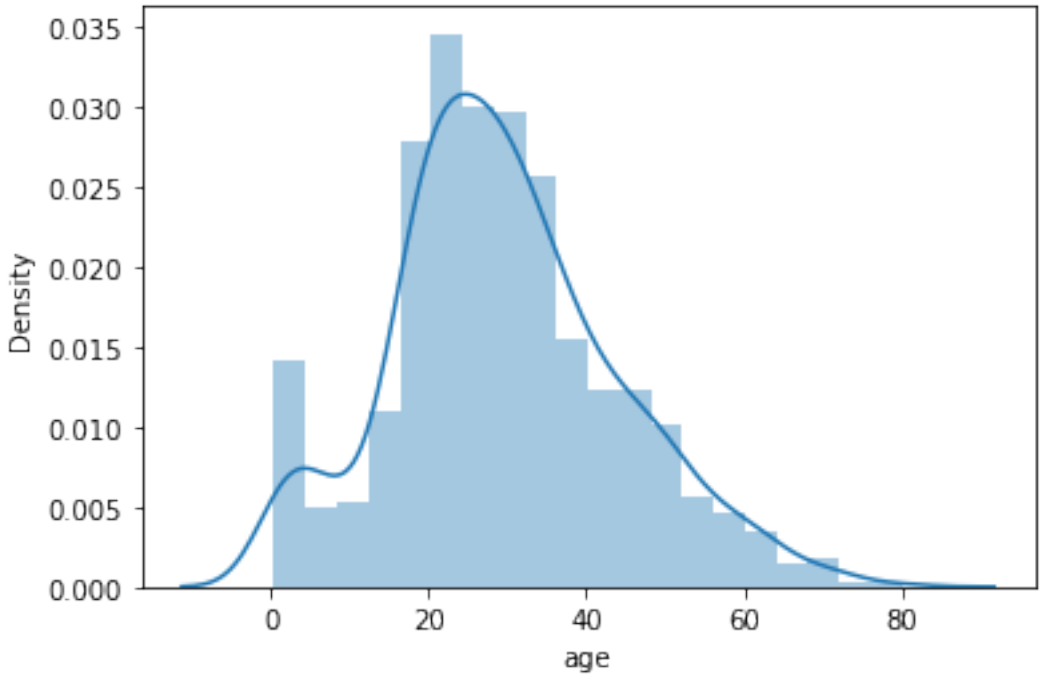
[ ]:
```
<AxesSubplot:ylabel='age'>
```



[ ]:
```
# bell curve check or normality check
sns.distplot(ks_clean2["age"] )
```

```
C:\Users\hp\AppData\Local\Programs\Python\Python310\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'>
```



1.2 outliers removal

[ ]:
```
# outliers removal
ks_clean2["age"].mean()
```

[ ]:
```
29.64209269662921
```

[ ]:
```
ks_clean2= ks_clean2[ks_clean2["age"]<68]
ks_clean2.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_clean2.shape
```

[ ]: (705, 14)

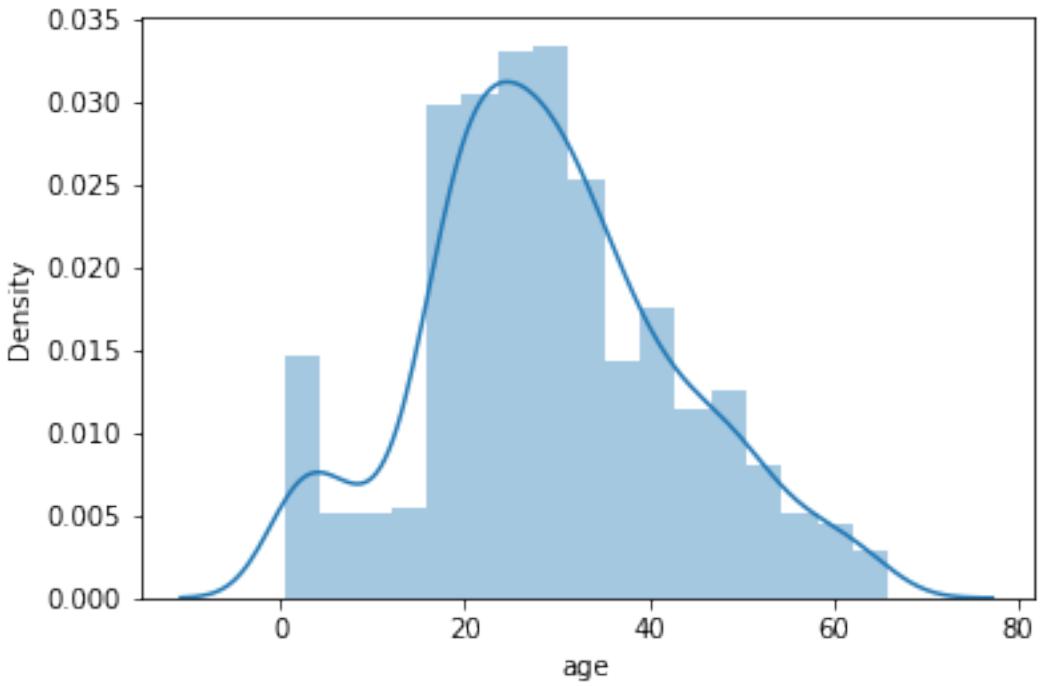
[ ]:
```
ks_clean2["age"].mean()
```

[ ]: 29.21797163120567

[ ]:
```
# bell curve check or normality check
sns.distplot(ks_clean2["age"] )
```

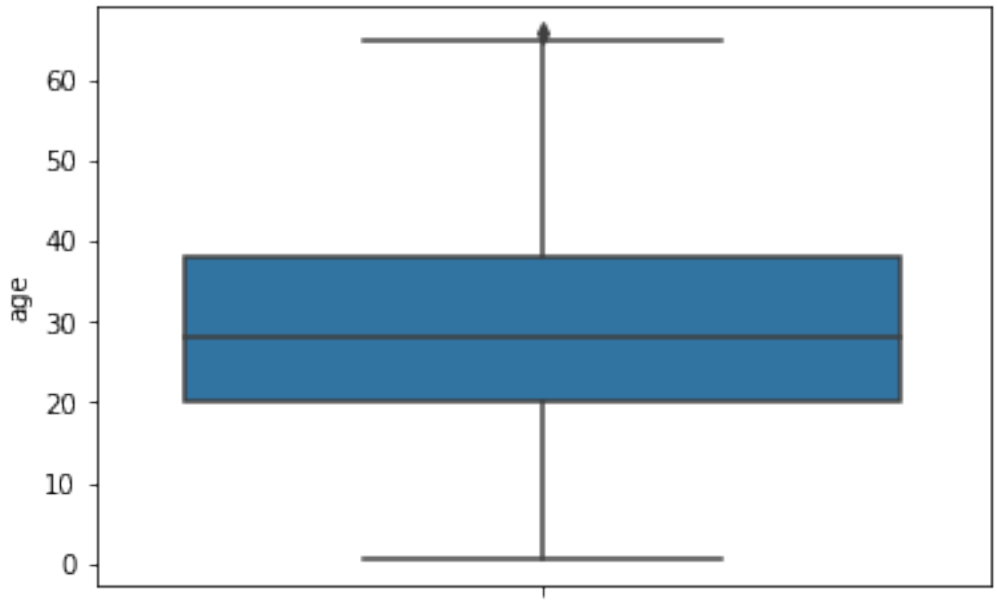
```
C:\Users\hp\AppData\Local\Programs\Python\Python310\lib\site-
packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a
deprecated function and will be removed in a future version. Please adapt your
code to use either `displot` (a figure-level function with similar flexibility)
or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

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



[ ]:
```
sns.boxplot(y="age", data=ks_clean2 )
```

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



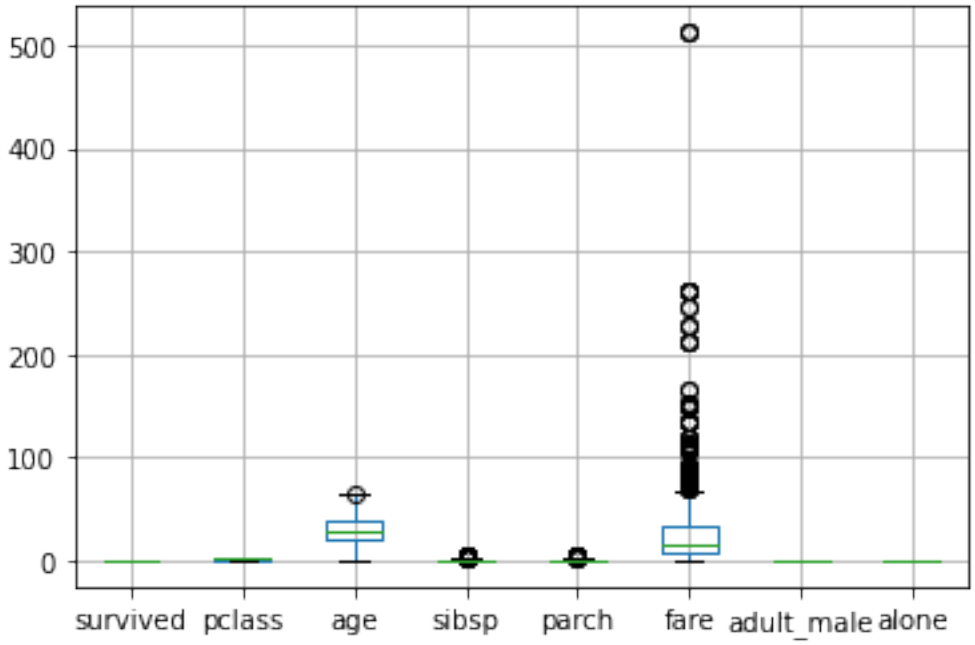
[ ]:
```
ks_clean2.head()
```

[ ]:
```
   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
```

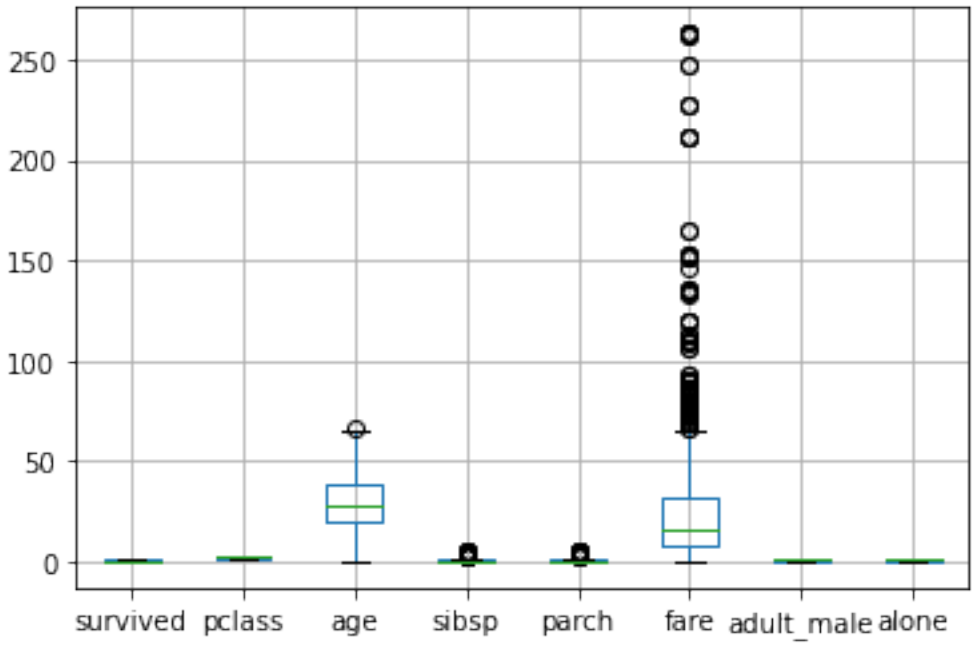
[ ]:
```
ks_clean2.boxplot()
```

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



[ ]:
```
ks_clean2= ks_clean2[ks_clean2["fare"]<300]
ks_clean2.boxplot()
```

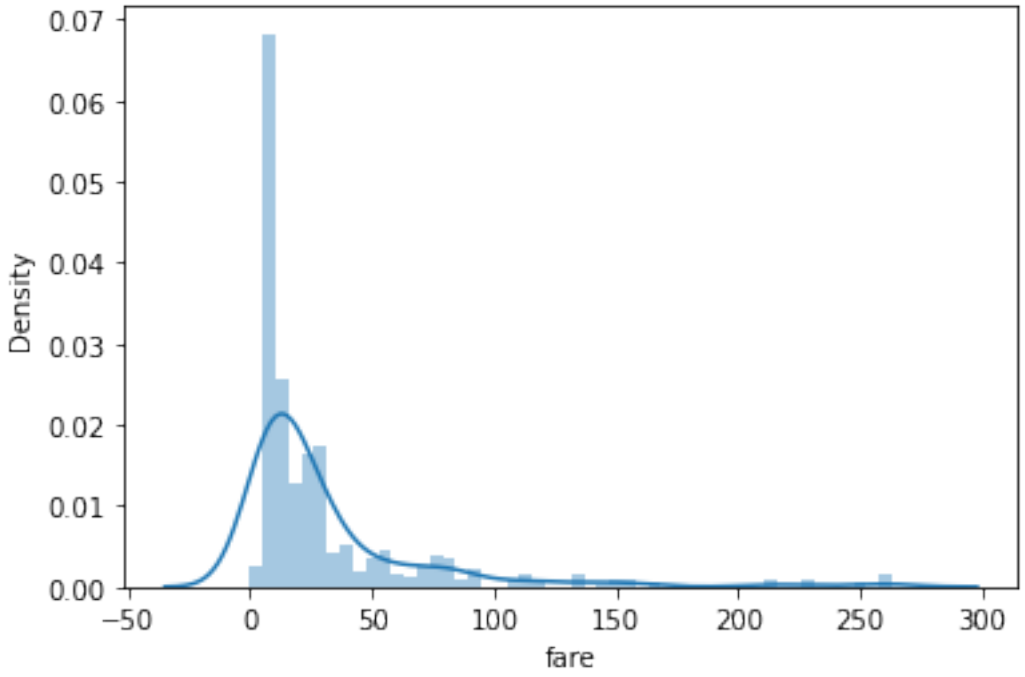
[ ]: <AxesSubplot:>



[ ]:
```
sns.distplot(ks_clean2["fare"] )
```

```
C:\Users\hp\AppData\Local\Programs\Python\Python310\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'>
```



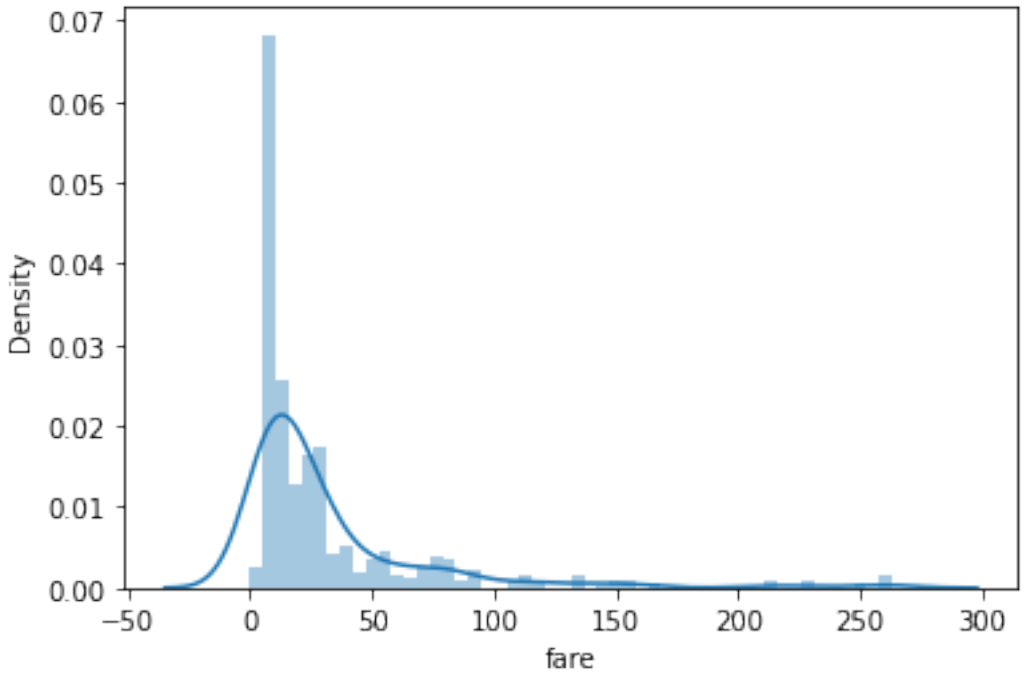
[ ]:
```
# Log Transformation
sns.distplot(ks_clean2["fare"] )
ks_clean2["fare_log"]=np.log(ks_clean2["fare"])
ks_clean2.head()
```

```
C:\Users\hp\AppData\Local\Programs\Python\Python310\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)
C:\Users\hp\AppData\Local\Programs\Python\Python310\lib\site-
packages\pandas\core\arraylike.py:364: RuntimeWarning: divide by zero
encountered in log
  result = getattr(ufunc, method)(*inputs, **kwargs)
```

[ ]:

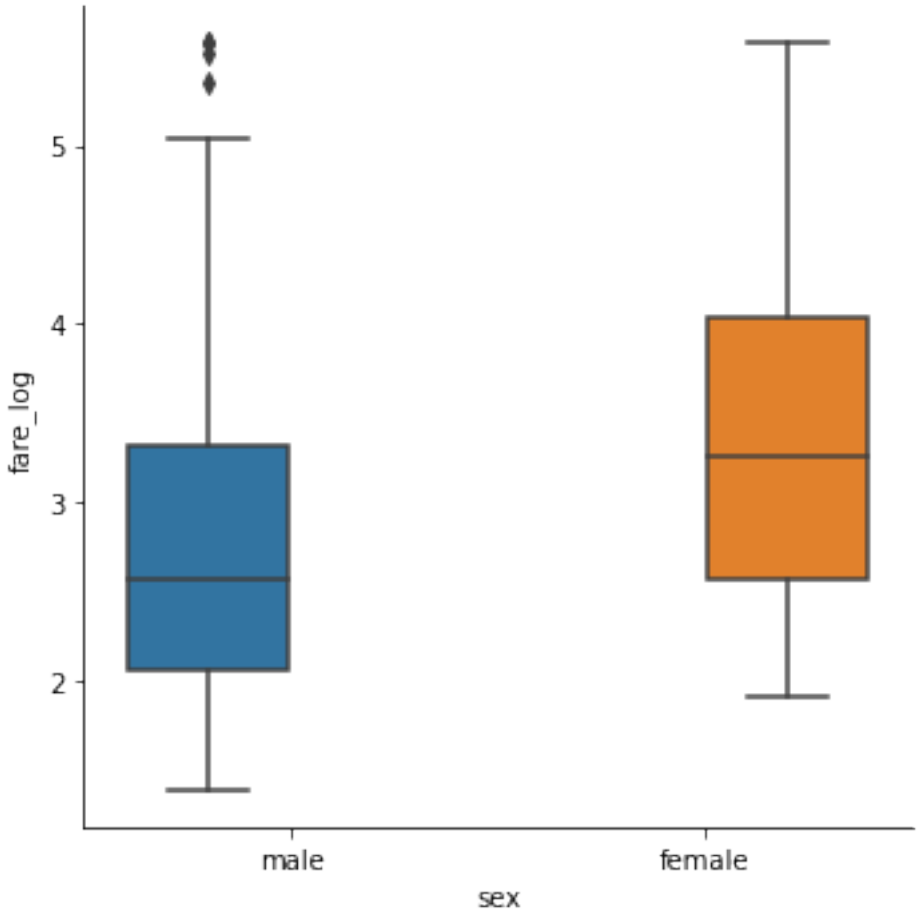
	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	\
0	0	3	male	22.0	1	0	7.2500	S	Third	
1	1	1	female	38.0	1	0	71.2833	C	First	
2	1	3	female	26.0	0	0	7.9250	S	Third	
3	1	1	female	35.0	1	0	53.1000	S	First	
4	0	3	male	35.0	0	0	8.0500	S	Third	

	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



[ ]:
```
sns.catplot(x="sex", y="fare_log", hue="sex", data=ks_clean2, kind= "box")
```

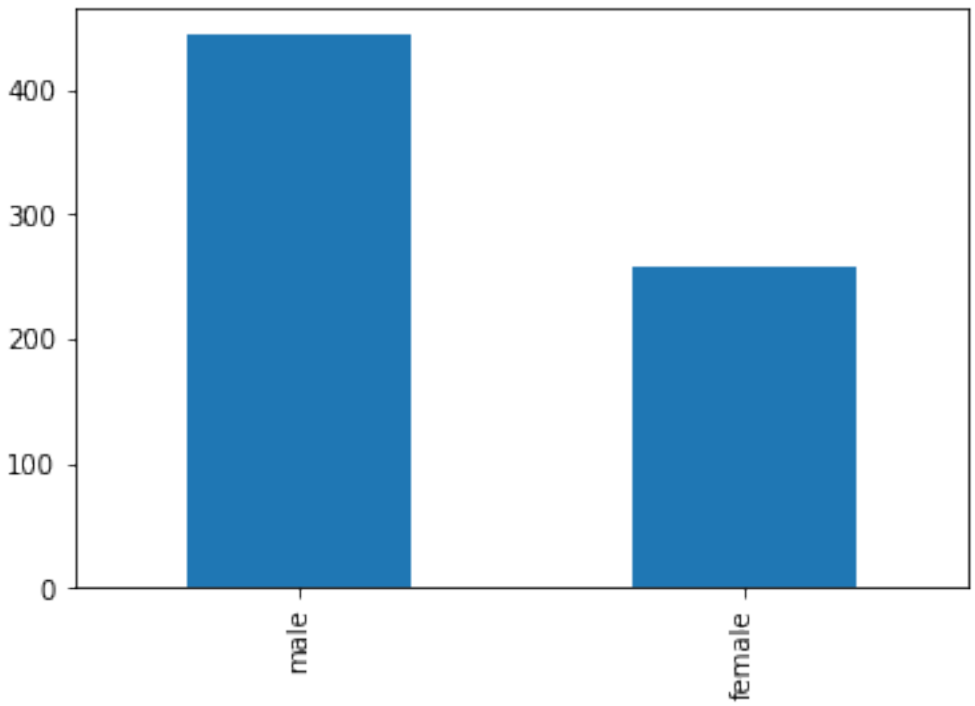
[ ]: <seaborn.axisgrid.FacetGrid at 0x1ac26812560>



[ ]:
```
ks_clean2.hist()
```

[ ]:
```
pd.value_counts(ks_clean2["sex"]).plot.bar()
```

[ ]: <AxesSubplot:>



[ ]:
```
ks_clean2.groupby(["sex","class"]).mean()
```

[ ]:
```
                survived  pclass        age     sibsp     parch        fare  \
sex    class
female First    0.963415     1.0  34.231707  0.560976  0.512195  103.696393
       Second   0.918919     2.0  28.722973  0.500000  0.621622   21.951070
       Third    0.460784     3.0  21.750000  0.823529  0.950980   15.875369
male   First    0.389474     1.0  40.067579  0.389474  0.336842   62.901096
       Second   0.153061     2.0  30.340102  0.377551  0.244898   21.221429
       Third    0.151394     3.0  26.143108  0.494024  0.258964   12.197757

                adult_male     alone
sex    class
female First      0.000000  0.353659
       Second     0.000000  0.405405
       Third      0.000000  0.372549
male   First      0.968421  0.526316
       Second     0.908163  0.632653
       Third      0.888446  0.737052
```

[ ]:
```
ks.groupby(["sex","class"]).mean()
```

[ ]:
```
                survived  pclass        age     sibsp     parch        fare  \
sex    class
```

```
female First   0.968085     1.0  34.611765  0.553191  0.457447  106.125798
       Second  0.921053     2.0  28.722973  0.486842  0.605263   21.970121
       Third   0.500000     3.0  21.750000  0.895833  0.798611   16.118810
male   First   0.368852     1.0  41.281386  0.311475  0.278689   67.226127
       Second  0.157407     2.0  30.740707  0.342593  0.222222   19.741782
       Third   0.135447     3.0  26.507589  0.498559  0.224784   12.661633
```

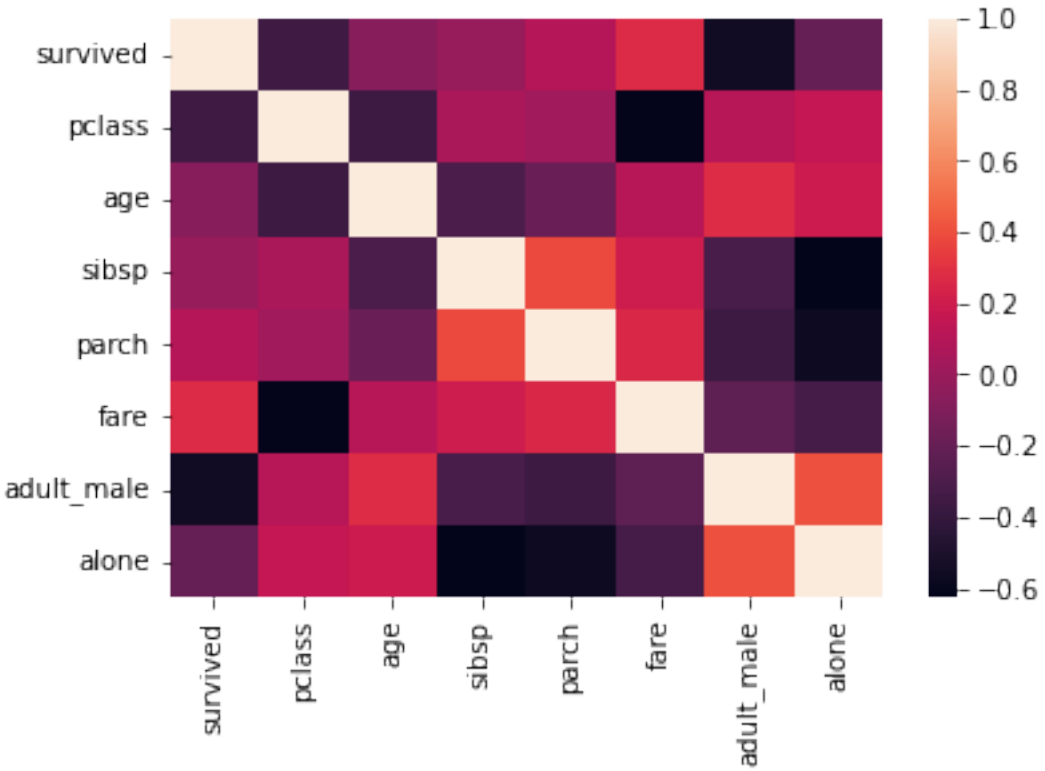
```
                adult_male     alone
sex    class
female First      0.000000  0.361702
       Second     0.000000  0.421053
       Third      0.000000  0.416667
male   First      0.975410  0.614754
       Second     0.916667  0.666667
       Third      0.919308  0.760807
```

1.3 Relationship

[ ]:
```
corr_ks_clean= ks_clean2.corr()
```

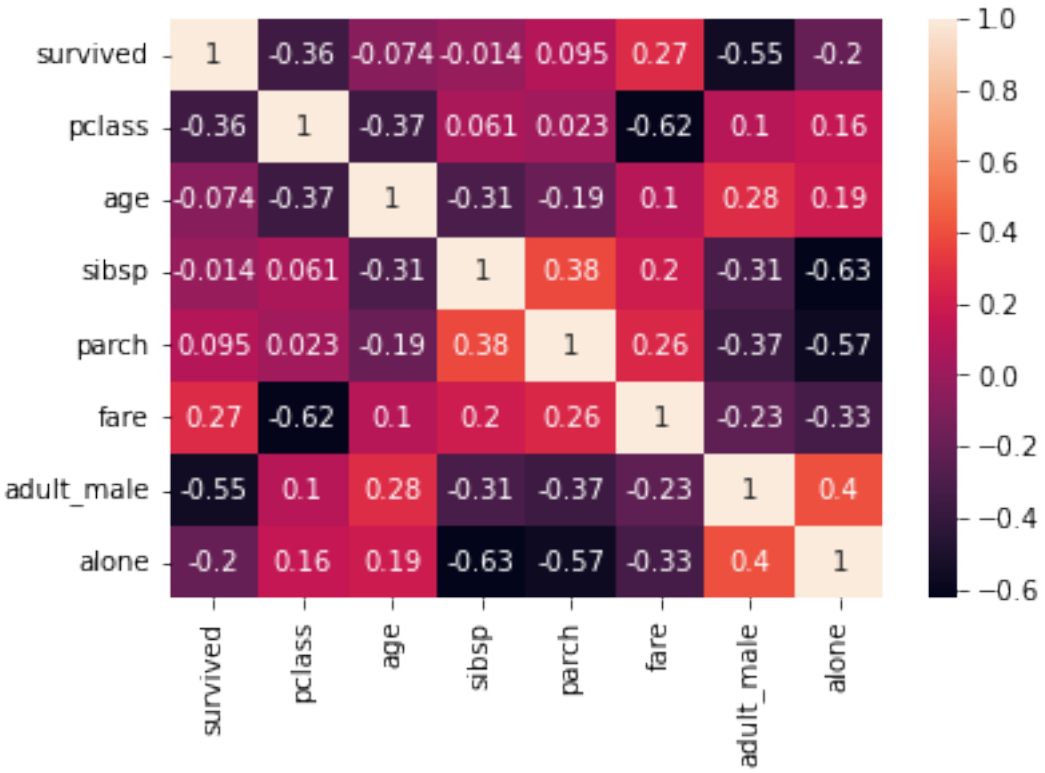
[ ]:
```
sns.heatmap(corr_ks_clean)
```

[ ]: <AxesSubplot:>



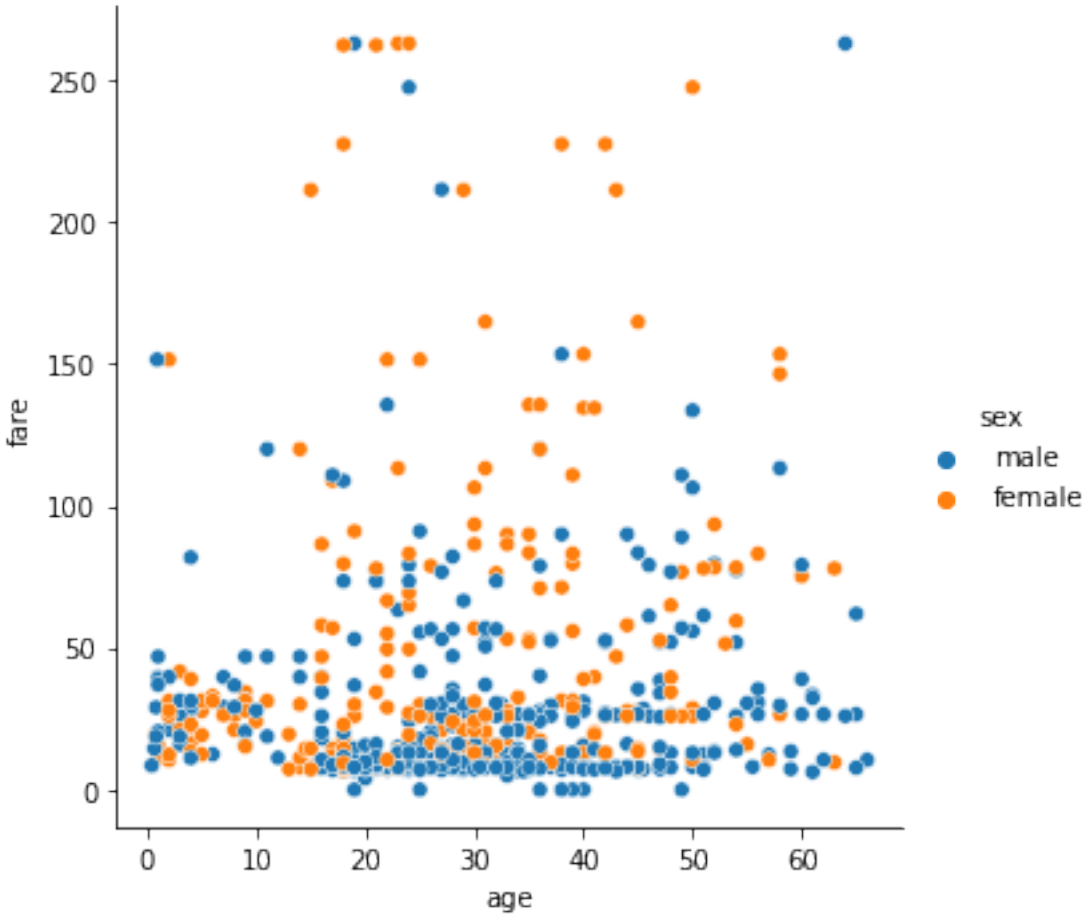
[ ]:
```
sns.heatmap(corr_ks_clean, annot=True)
```

[ ]: <AxesSubplot:>



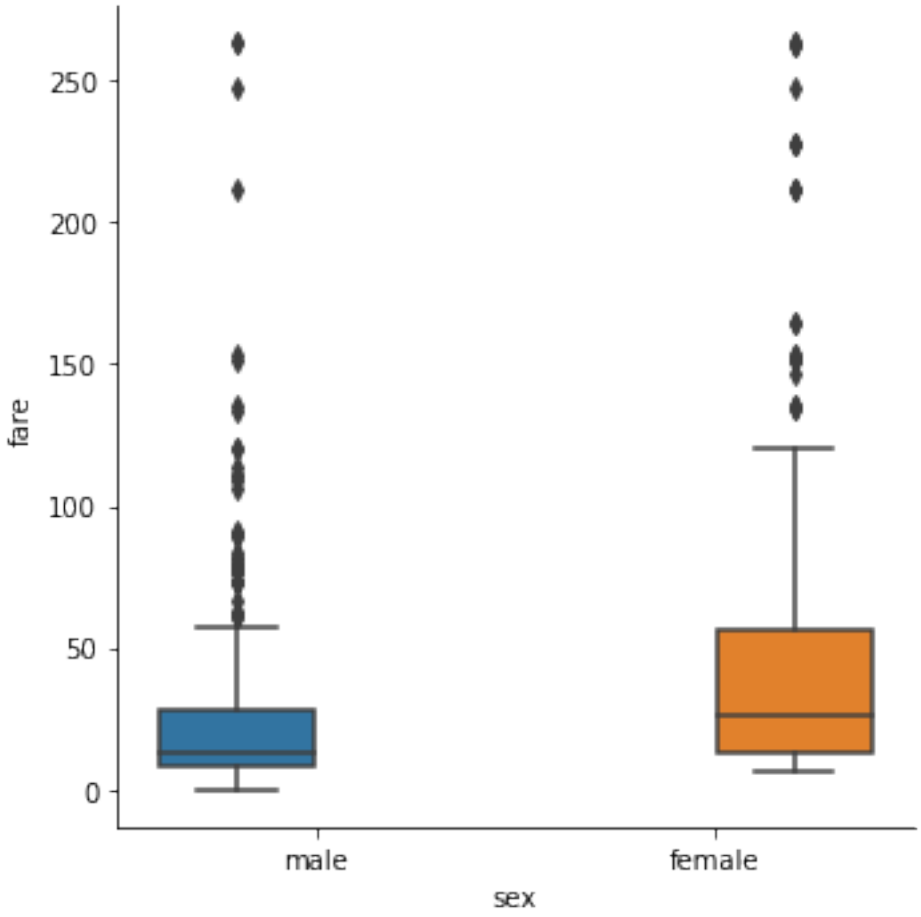
[ ]:
```
sns.relplot(x="age", y="fare", hue="sex", data=ks_clean2)
```

[ ]: <seaborn.axisgrid.FacetGrid at 0x284060bebc0>



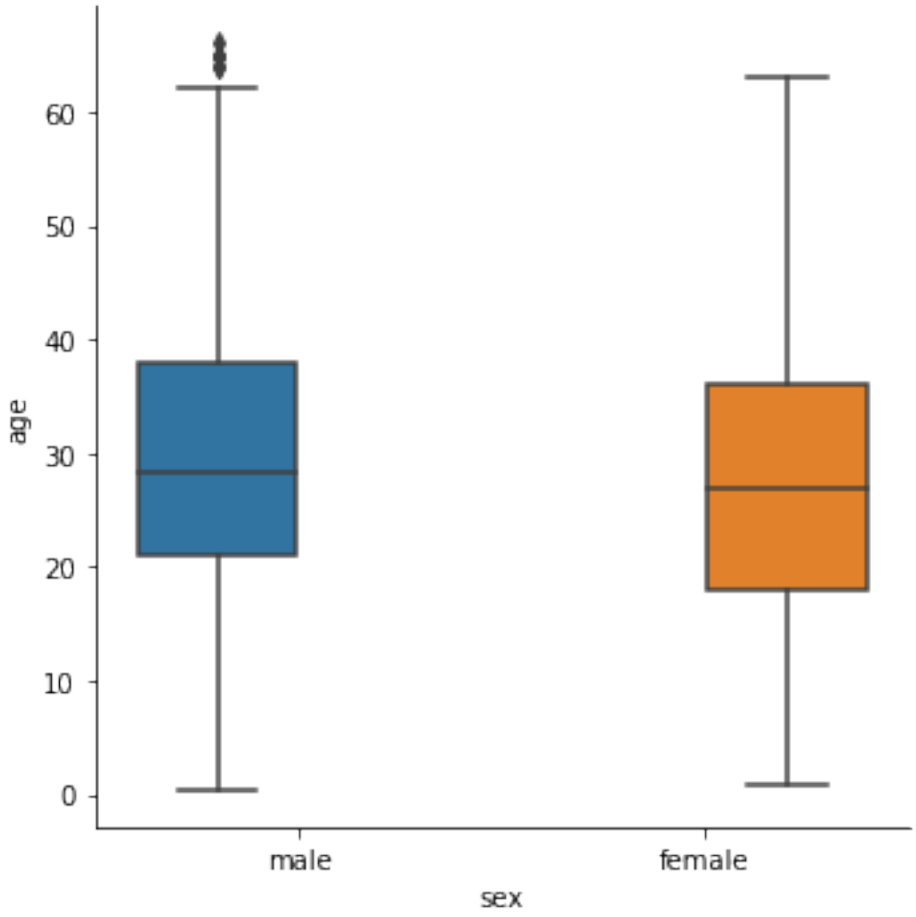
[ ]:
```
sns.catplot(x="sex", y="fare", hue="sex", data=ks_clean2, kind= "box")
```

[ ]:
```
<seaborn.axisgrid.FacetGrid at 0x28407413820>
```



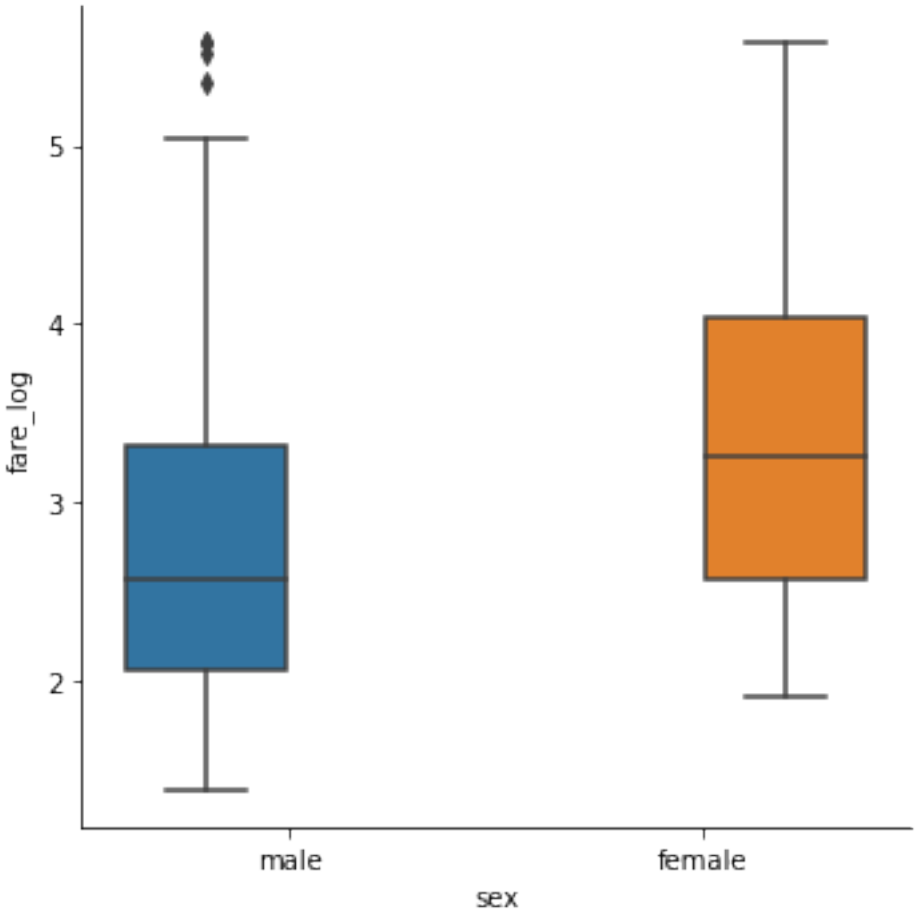
[ ]:
```
sns.catplot(x="sex", y="age", hue="sex", data=ks_clean2, kind= "box")
```

[ ]:
```
<seaborn.axisgrid.FacetGrid at 0x284074d7010>
```



[ ]:
```
sns.catplot(x="sex", y="fare_log", hue="sex", data=ks_clean2, kind= "box")
```

[ ]:
```
<seaborn.axisgrid.FacetGrid at 0x28405dfb8e0>
```


5
4
3
2
fare_log
male
female
sex