Exploratory Data Analysis

This will show us how can we do EDA using python

Three important steps to keep in mind are

1- understand the data

2- clean the data

3- find a relationship between the data

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

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

In [ ]:
```
ship.to_csv('ship.csv')
```

In [ ]:
```
ship.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 [ ]:
```
sh=ship
sh1=ship
```

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

Out[ ]:

sex	class	survived	pclass	age	sibsp	parch	fare	adult_male	alone
female	First	0.968085	1.0	34.611765	0.553191	0.457447	106.125798	0.000000	0.361702
	Second	0.921053	2.0	28.722973	0.486842	0.605263	21.970121	0.000000	0.421053
	Third	0.500000	3.0	21.750000	0.895833	0.798611	16.118810	0.000000	0.416667
male	First	0.368852	1.0	41.281386	0.311475	0.278689	67.226127	0.975410	0.614754
	Second	0.157407	2.0	30.740707	0.342593	0.222222	19.741782	0.916667	0.666667
	Third	0.135447	3.0	26.507589	0.498559	0.224784	12.661633	0.919308	0.760807

In [ ]:
```
sh.head()
```

Out[ ]:

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	deck
0	0	3	male	22.0	1	0	7.2500	S	Third	man	True	NaN
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
3	1	1	female	35.0	1	0	53.1000	S	First	woman	False	C
4	0	3	male	35.0	0	0	8.0500	S	Third	man	True	NaN

In [ ]:
```
sh.tail()
```

Out[ ]:

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	de
886	0	2	male	27.0	0	0	13.00	S	Second	man	True	Na
887	1	1	female	19.0	0	0	30.00	S	First	woman	False	
888	0	3	female	NaN	1	2	23.45	S	Third	woman	False	Na
889	1	1	male	26.0	0	0	30.00	C	First	man	True	
890	0	3	male	32.0	0	0	7.75	Q	Third	man	True	Na

In [ ]:
```
sh.shape
```

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

In [ ]:
```
sh.nunique()
```

Out[ ]:
```
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 [ ]:
```
sh.describe()
```

Out[ ]:

	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 [ ]:
```
sh.columns
```

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

In [ ]:
```
sh["sex"].unique()
```

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

In [ ]:
```
sh["who"].unique()
```

Out[ ]:
```
array(['man', 'woman', 'child'], dtype=object)
```

In [ ]:
```
sh["deck"].unique()
```

Out[ ]:
```
[NaN, 'C', 'E', 'G', 'D', 'A', 'B', 'F']
Categories (7, object): ['A', 'B', 'C', 'D', 'E', 'F', 'G']
```

In [ ]:
```
#sh(["deck", "who"].unique())
```

Cleaning and filtering the data

In [ ]:
```
# find missing values
sh.isnull()
```

Out[ ]:

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

891 rows × 15 columns

In [ ]:
```
sh.isnull().sum()
```

Out[ ]:
```
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 [ ]:
```
# removing missing values/cleaning data
sh_clean=sh.drop(['deck'], axis=1)
sh_clean.head()
```

Out[ ]:

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	emba
0	0	3	male	22.0	1	0	7.2500	S	Third	man	True	Soutl
1	1	1	female	38.0	1	0	71.2833	C	First	woman	False	Cl
2	1	3	female	26.0	0	0	7.9250	S	Third	woman	False	Soutl
3	1	1	female	35.0	1	0	53.1000	S	First	woman	False	Soutl
4	0	3	male	35.0	0	0	8.0500	S	Third	man	True	Soutl

In [ ]:
```
sh_clean.isnull().sum()
```

Out[ ]:
```
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 [ ]:
```
sh_clean.shape
```

Out[ ]:
```
(891, 14)
```

In [ ]:
```
sh_clean=sh_clean.dropna()
```

In [ ]:
```
sh_clean.shape
```

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

In [ ]:
```
sh_clean.isnull().sum()
```

Out[ ]:
```
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
```

In [ ]:
```
sh.shape
```

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

In [ ]:
```
sh_clean.shape
```

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

In [ ]:
```
sh_clean['sex'].value_counts()
```

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

In [ ]:
```
sh_clean['age'].value_counts()
```

Out[ ]:
```
24.00    30
22.00    27
18.00    26
19.00    25
28.00    25
         ..
36.50     1
55.50     1
0.92      1
23.50     1
74.00     1
Name: age, Length: 88, dtype: int64
```

In [ ]:
```
sh.describe()
```

Out[ ]:

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

Out[ ]:

	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

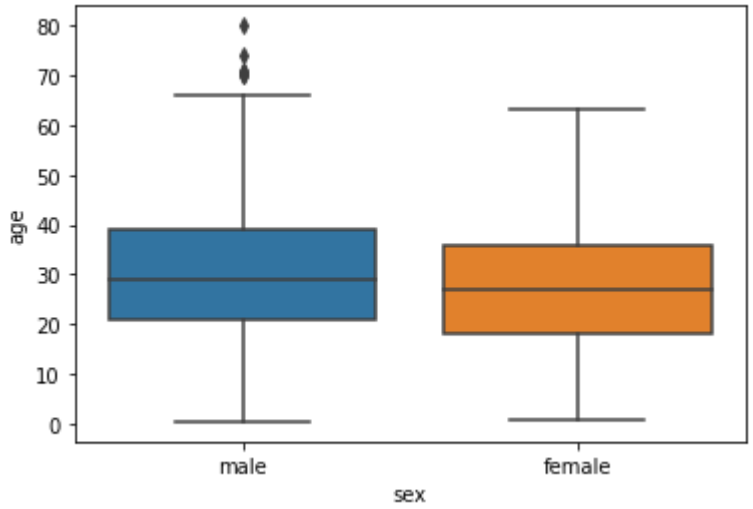
Findind outliers

In [ ]:
```
sh_clean.columns
```

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

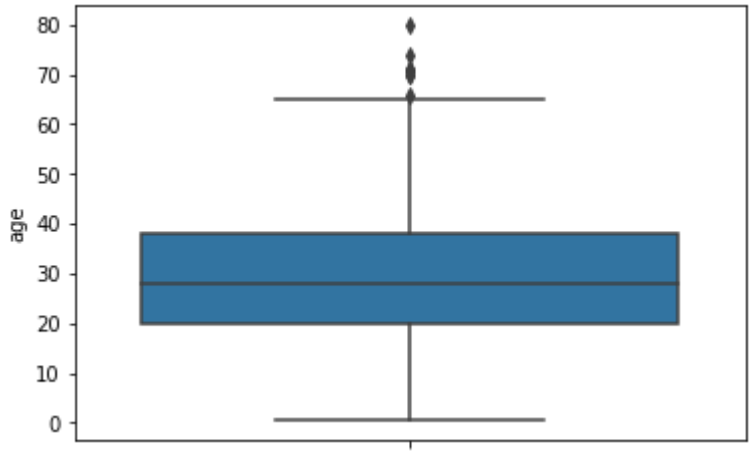
In [ ]:
```
sns.boxplot(x="sex", y="age", data=sh_clean)
```

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



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

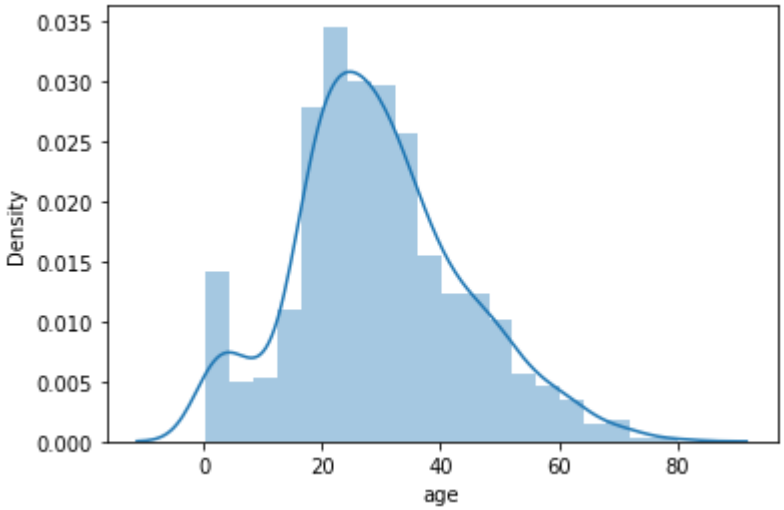
Out[ ]:
```
<AxesSubplot:ylabel='age'>
```



In [ ]:
```
sns.distplot(sh_clean['age'])
```

```
c:\Users\m s\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)
```

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



In [ ]:
```
# out liers remover
sh_clean['age'].mean()
```

Out[ ]:
```
29.64209269662921
```

In [ ]:
```
sh['age']=sh_clean['age']<66
sh['age'].mean()
```

Out[ ]:
```
0.9887640449438202
```

In [ ]:
```
sh_clean=sh_clean[sh_clean['age']<66]
sh_clean
```

Out[ ]:

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	e
0	0	3	male	22.0	1	0	7.2500	S	Third	man	True	
1	1	1	female	38.0	1	0	71.2833	C	First	woman	False	
2	1	3	female	26.0	0	0	7.9250	S	Third	woman	False	
3	1	1	female	35.0	1	0	53.1000	S	First	woman	False	
4	0	3	male	35.0	0	0	8.0500	S	Third	man	True	
...	...	...	...	...	...	...	...	...	...	...	...	
885	0	3	female	39.0	0	5	29.1250	Q	Third	woman	False	
886	0	2	male	27.0	0	0	13.0000	S	Second	man	True	
887	1	1	female	19.0	0	0	30.0000	S	First	woman	False	
889	1	1	male	26.0	0	0	30.0000	C	First	man	True	
890	0	3	male	32.0	0	0	7.7500	Q	Third	man	True	

704 rows × 14 columns

In [ ]:
```
sh_clean.shape
```

Out[ ]:
```
(704, 14)
```

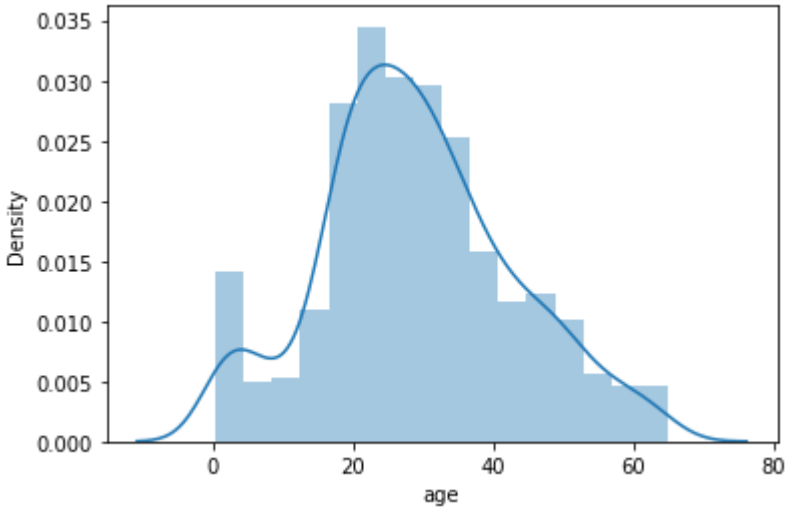
In [ ]:
```
sh_clean['age'].mean()
```

Out[ ]:
```
29.16572443181818
```

In [ ]:
```
sns.distplot(sh_clean['age'])
```

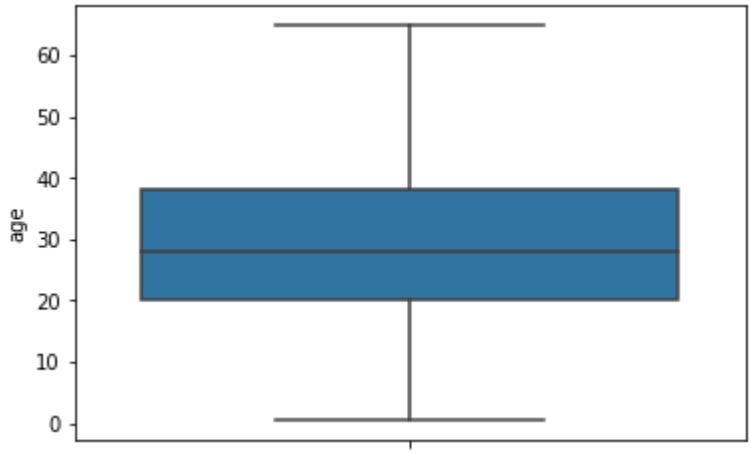
```
c:\Users\m s\AppData\Local\Programs\Python\Python310\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot`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

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



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

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



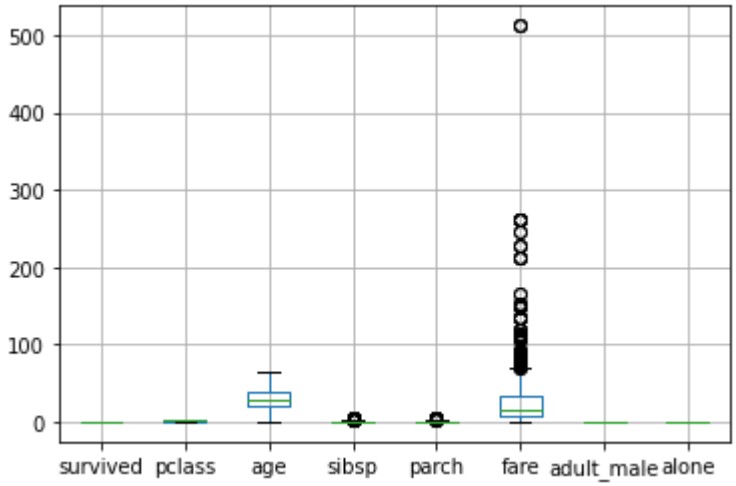
In [ ]:
```
sh_clean.head()
```

Out[ ]:

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	emba
0	0	3	male	22.0	1	0	7.2500	S	Third	man	True	Soutl
1	1	1	female	38.0	1	0	71.2833	C	First	woman	False	Cl
2	1	3	female	26.0	0	0	7.9250	S	Third	woman	False	Soutl
3	1	1	female	35.0	1	0	53.1000	S	First	woman	False	Soutl
4	0	3	male	35.0	0	0	8.0500	S	Third	man	True	Soutl

In [ ]:
```
sh_clean.boxplot()
```

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



In [ ]:
```
sh_clean=sh_clean[sh_clean['fare']<300]
sh_clean
```

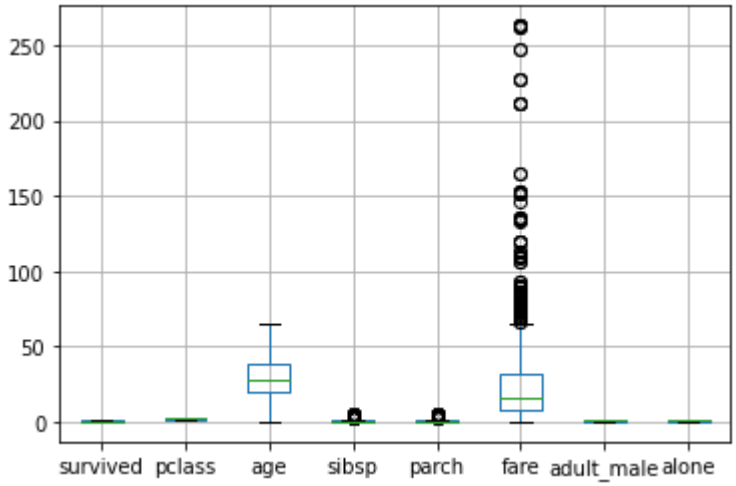
Out[ ]:

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	e
0	0	3	male	22.0	1	0	7.2500	S	Third	man	True	
1	1	1	female	38.0	1	0	71.2833	C	First	woman	False	
2	1	3	female	26.0	0	0	7.9250	S	Third	woman	False	
3	1	1	female	35.0	1	0	53.1000	S	First	woman	False	
4	0	3	male	35.0	0	0	8.0500	S	Third	man	True	
...	...	...	...	...	...	...	...	...	...	...	...	...
885	0	3	female	39.0	0	5	29.1250	Q	Third	woman	False	
886	0	2	male	27.0	0	0	13.0000	S	Second	man	True	
887	1	1	female	19.0	0	0	30.0000	S	First	woman	False	
889	1	1	male	26.0	0	0	30.0000	C	First	man	True	
890	0	3	male	32.0	0	0	7.7500	Q	Third	man	True	

701 rows × 14 columns

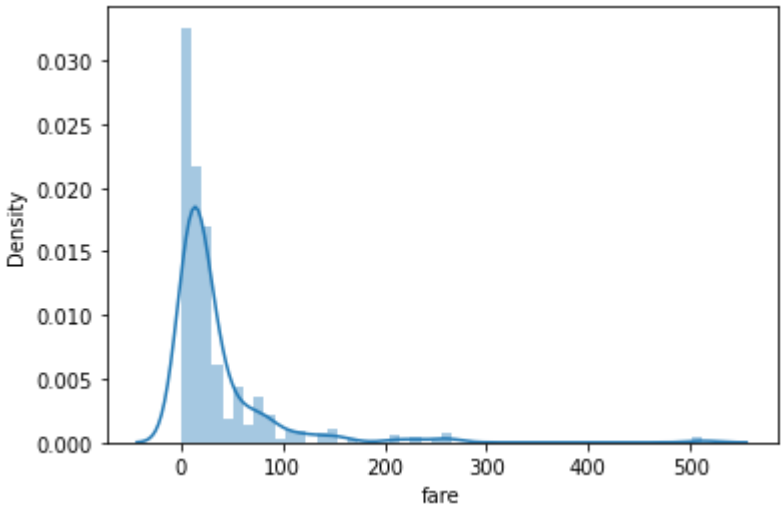
In [ ]:
```
sh_clean.boxplot()
```

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



In [ ]:
```
sns.distplot(sh_clean['fare'])
sh_clean['fare_log']=np.log(sh_clean['fare'])
```

```
c:\Users\m s\AppData\Local\Programs\Python\Python310\lib\site-packages\seaborn\distri
butions.py:2619: FutureWarning: `distplot` is a deprecated function and will be remov
ed in a future version. Please adapt your code to use either `displot` (a figure-leve
l function with similar flexibility) or `histplot` (an axes-level function for histog
rams).
  warnings.warn(msg, FutureWarning)
c:\Users\m s\AppData\Local\Programs\Python\Python310\lib\site-packages\pandas\core\ar
raylike.py:397: RuntimeWarning: divide by zero encountered in log
  result = getattr(ufunc, method)(*inputs, **kwargs)
```



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

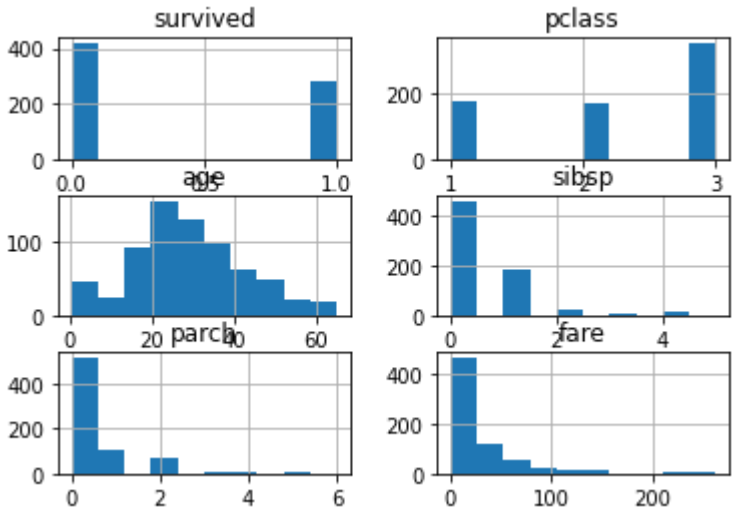
```
c:\Users\m s\AppData\Local\Programs\Python\Python310\lib\site-packages\pandas\core\ar
raylike.py:397: RuntimeWarning: divide by zero encountered in log
  result = getattr(ufunc, method)(*inputs, **kwargs)
```

Out[ ]:

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	emba
0	0	3	male	22.0	1	0	7.2500	S	Third	man	True	Soutl
1	1	1	female	38.0	1	0	71.2833	C	First	woman	False	Cl
2	1	3	female	26.0	0	0	7.9250	S	Third	woman	False	Soutl
3	1	1	female	35.0	1	0	53.1000	S	First	woman	False	Soutl
4	0	3	male	35.0	0	0	8.0500	S	Third	man	True	Soutl

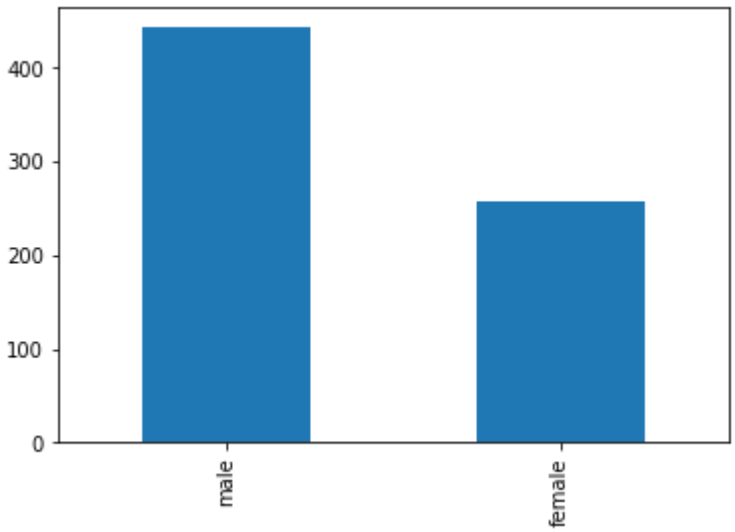
In [ ]:
```
sh_clean.hist()
```

Out[ ]:
```
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)
```



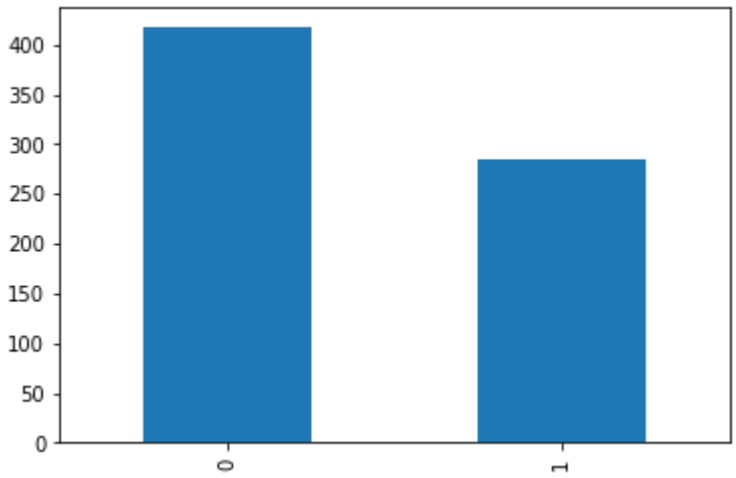
In [ ]:
```
pd.value_counts(sh_clean['sex']).plot.bar()
```

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



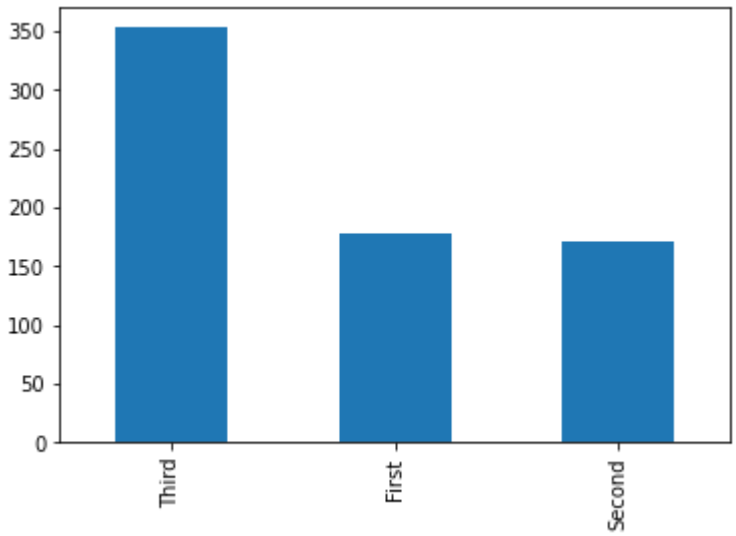
In [ ]:
```
pd.value_counts(sh_clean['survived']).plot.bar()
```

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



In [ ]:
```
pd.value_counts(sh_clean['class']).plot.bar()
```

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



In [ ]:
```
sh_clean.groupby(['sex']).mean()
```

Out[ ]:

sex	survived	pclass	age	sibsp	parch	fare	adult_male	alone
female	0.751938	2.077519	27.717054	0.647287	0.717054	45.530120	0.000000	0.375969
male	0.203160	2.352144	29.967652	0.446953	0.273138	25.070973	0.909707	0.668172

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

Out[ ]:

sex	class	survived	pclass	age	sibsp	parch	fare	adult_male	alone
female	First	0.963415	1.0	34.231707	0.560976	0.512195	103.696393	0.000000	0.353659
	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	0.389474	1.0	40.067579	0.389474	0.336842	62.901096	0.968421	0.526316
	Second	0.154639	2.0	29.972474	0.381443	0.247423	21.331959	0.907216	0.628866
	Third	0.151394	3.0	26.143108	0.494024	0.258964	12.197757	0.888446	0.737052

In [ ]:
```
sh1.groupby(['sex', 'class']).mean()
```

Out[ ]:

sex	class	survived	pclass	sibsp	parch	fare	adult_male	alone
female	First	0.968085	1.0	0.553191	0.457447	106.125798	0.000000	0.361702
	Second	0.921053	2.0	0.486842	0.605263	21.970121	0.000000	0.421053
	Third	0.500000	3.0	0.895833	0.798611	16.118810	0.000000	0.416667
male	First	0.368852	1.0	0.311475	0.278689	67.226127	0.975410	0.614754
	Second	0.157407	2.0	0.342593	0.222222	19.741782	0.916667	0.666667
	Third	0.135447	3.0	0.498559	0.224784	12.661633	0.919308	0.760807

In [ ]:
```
sh_clean.groupby(['sex', 'class','who']).mean()
```

Out[ ]:

sex	class	who	survived	pclass	age	sibsp	parch	fare	adult_male	
female	First	child	0.666667	1.0	10.333333	0.666667	1.666667	160.962500	0.0	0.(
		man	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
		woman	0.975000	1.0	35.137500	0.550000	0.462500	106.656824	0.0	0.3
	Second	child	1.000000	2.0	6.600000	0.700000	1.300000	29.240000	0.0	0.(
		man	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
		woman	0.906250	2.0	32.179688	0.468750	0.515625	20.812175	0.0	0.4
	Third	child	0.533333	3.0	7.100000	1.533333	1.100000	19.023753	0.0	0.1
		man	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
		woman	0.430556	3.0	27.854167	0.527778	0.888889	14.563542	0.0	0.4
male	First	child	1.000000	1.0	5.306667	0.666667	2.000000	117.802767	0.0	0.(
		man	0.382979	1.0	41.079787	0.372340	0.287234	70.711215	1.0	0.5
		woman	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
	Second	child	1.000000	2.0	2.258889	0.888889	1.222222	27.306022	0.0	0.(
		man	0.068182	2.0	32.806818	0.329545	0.147727	20.720975	1.0	0.6
		woman	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
	Third	child	0.321429	3.0	6.515000	2.821429	1.321429	27.716371	0.0	0.(
		man	0.130045	3.0	28.607623	0.201794	0.125561	10.249231	1.0	0.8
		woman	NaN	NaN	NaN	NaN	NaN	NaN	NaN	

Relationship

In [ ]:
```
sh_clean.corr()
```

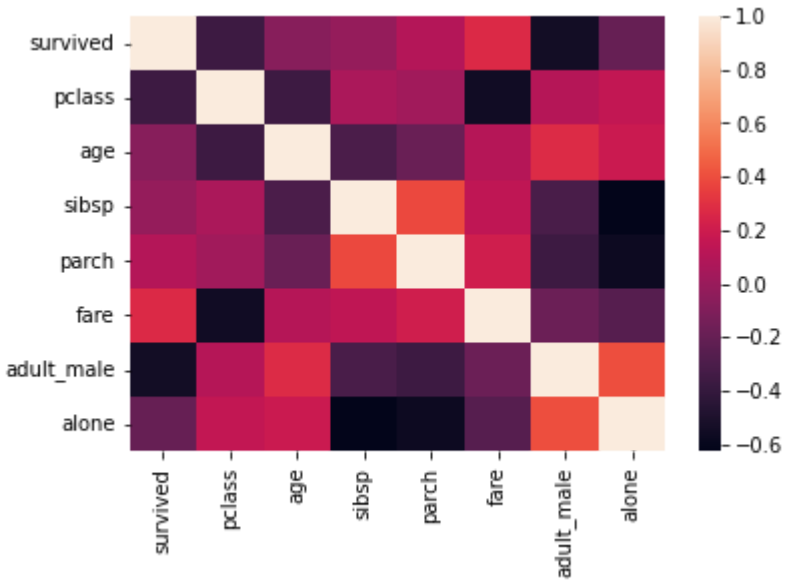
Out[ ]:

	survived	pclass	age	sibsp	parch	fare	adult_male	alone
survived	1.000000	-0.361986	-0.069069	-0.017955	0.093914	0.266589	-0.550326	-0.198219
pclass	-0.361986	1.000000	-0.366776	0.064347	0.022950	-0.554871	0.101479	0.154086
age	-0.069069	-0.366776	1.000000	-0.309139	-0.185259	0.102494	0.273060	0.184792
sibsp	-0.017955	0.064347	-0.309139	1.000000	0.381330	0.138389	-0.310780	-0.627808
parch	0.093914	0.022950	-0.185259	0.381330	1.000000	0.205286	-0.364170	-0.575273
fare	0.266589	-0.554871	0.102494	0.138389	0.205286	1.000000	-0.177108	-0.261067
adult_male	-0.550326	0.101479	0.273060	-0.310780	-0.364170	-0.177108	1.000000	0.398185
alone	-0.198219	0.154086	0.184792	-0.627808	-0.575273	-0.261067	0.398185	1.000000

In [ ]:
```
corr_sh_clean=sh_clean.corr()
```

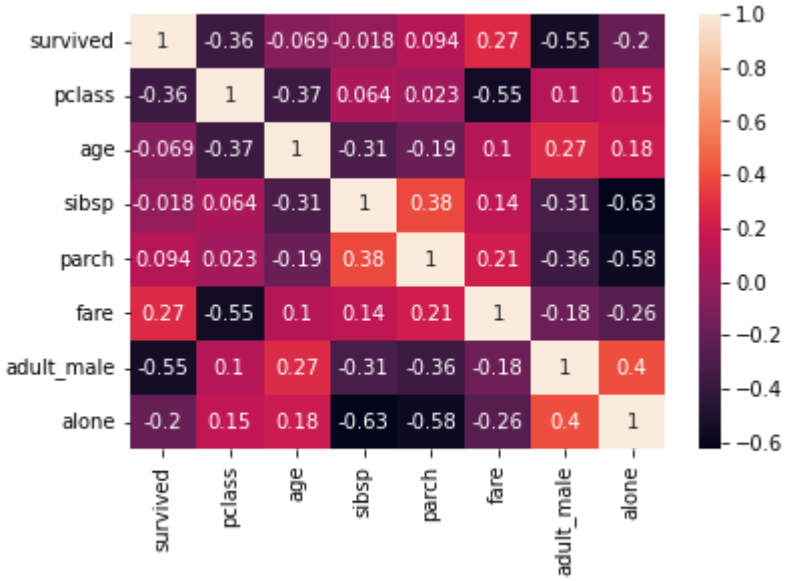
In [ ]:
```
sns.heatmap(corr_sh_clean)
```

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



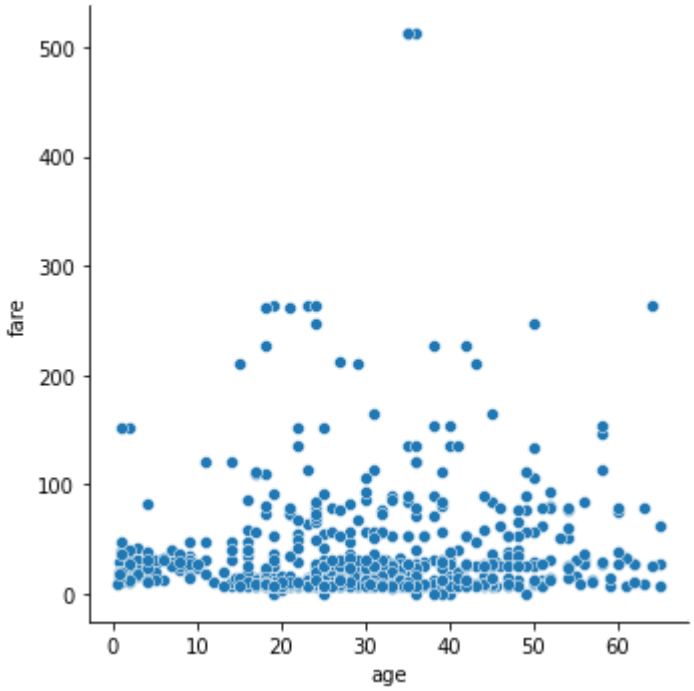
In [ ]:
```
sns.heatmap(corr_sh_clean, annot=True )
```

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



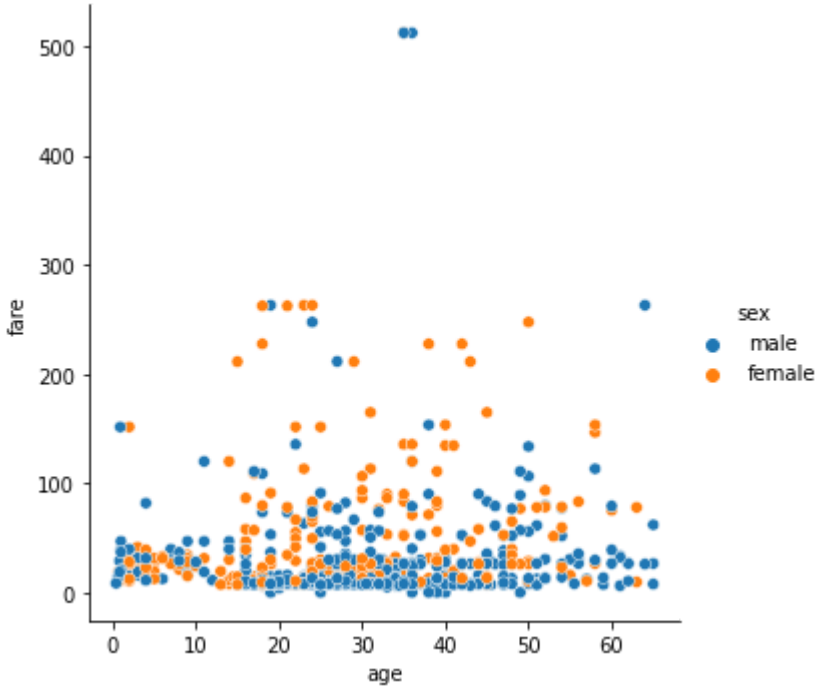
In [ ]:
```
sns.relplot(x="age", y="fare", data=sh_clean)
```

Out[ ]:
```
<seaborn.axisgrid.FacetGrid at 0x1cd79f64d90>
```



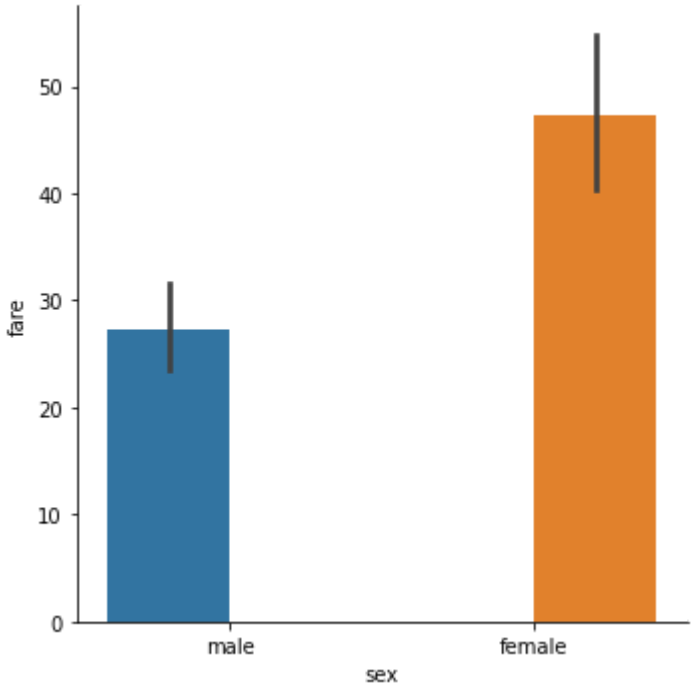
In [ ]:
```
sns.relplot(x="age", y="fare",hue="sex", data=sh_clean)
```

Out[ ]:
```
<seaborn.axisgrid.FacetGrid at 0x1cd7a003e50>
```



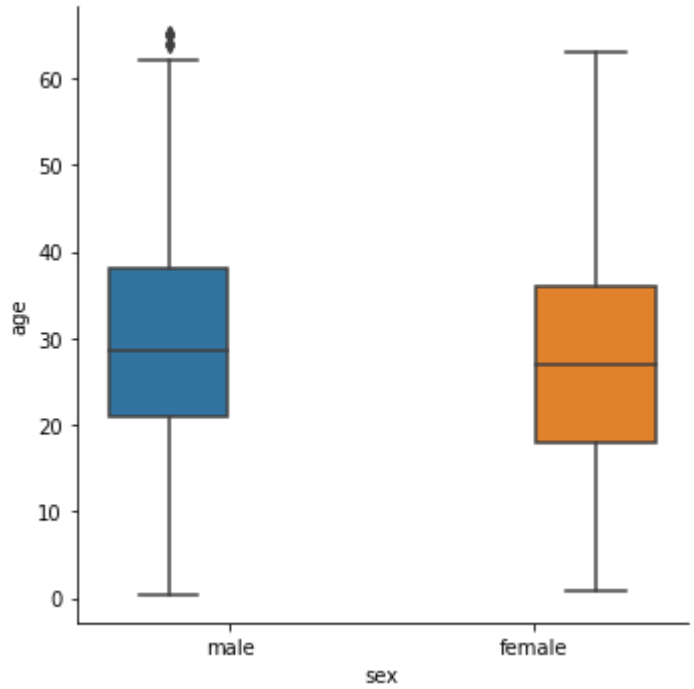
In [ ]:
```
sns.catplot(x="sex", y="fare",hue="sex", data=sh_clean, kind="bar")
```

Out[ ]:
```
<seaborn.axisgrid.FacetGrid at 0x1cd7a30c7f0>
```



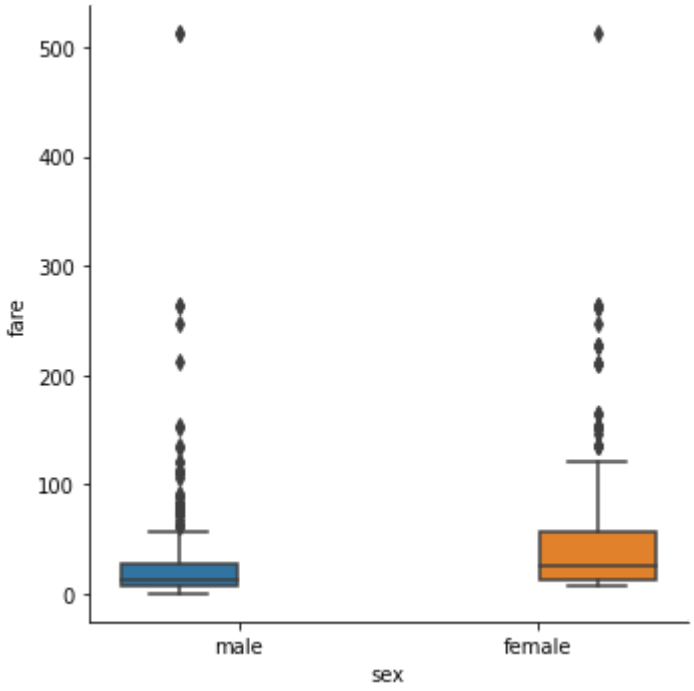
In [ ]:
```
sns.catplot(x="sex", y="age",hue="sex", data=sh_clean, kind="box")
```

Out[ ]:
```
<seaborn.axisgrid.FacetGrid at 0x1cd7a2cffd0>
```



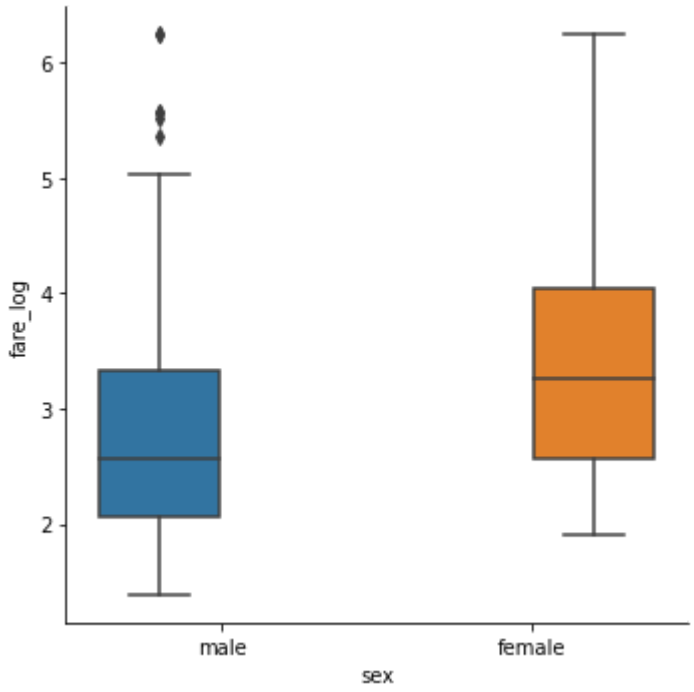
In [ ]:
```
sns.catplot(x="sex", y="fare",hue="sex", data=sh_clean, kind="box")
```

Out[ ]:
```
<seaborn.axisgrid.FacetGrid at 0x1cd7a1cb460>
```



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

Out[ ]:
```
<seaborn.axisgrid.FacetGrid at 0x1cd7a2476a0>
```



In [ ]: