# DS7- Exploratory Data Analysis

March 16, 2022

## 1 Exploratory Data Analysis (EDA)

Date: 16-03-2022

Title: Engr.

Name: Muhammad Ammar Mohsin

Email: ammarmohsin104@gmail

Whatsapp: +923030777088

#### 1.0.1 Exploratory Data Analysis (EDA)

In statistics, exploratory data analysis is an approach to analyzing data sets to summarize their main characteristics, often with visual methods. A statistical model can be used or not, but primarily EDA is for seeing what the data can tell us beyond the formal modeling or hypothesis testing task.

#### 1.0.2 Exploratory Data Analysis in Python

EDA in Python uses data visualization to draw meaningful patterns and insights. It also involves the preparation of data sets for analysis by removing irregularities in the data.

#### 1.0.3 Univariate analysis

Univariate analysis is the simplest form of data analysis, where the data being analyzed consists of only one variable. Since it's a single variable, it doesn't deal with causes or relationships. The main purpose of univariate analysis is to describe the data and find patterns that exist within it.

**Visualization for Univariate Analysis** - Box Plots - Histograms

#### 1.0.4 Multivariate analysis

Multivariate data analysis refers to any statistical technique used to analyze data that arises from more than one variable. This models more realistic applications, where each situation, product, or decision involves more than a single variable.

**Visualization for Univariate Analysis** - Scatter Plots - Bar Chart - Correlation Matrix - Pair Plot - Heatmap

## 1.1 Hands-on Practice

The import steps to keep in mind are: 1. Understand the data 2. Clean the data 3. Find a realtionship between data

[ ]:
```python
# Import Libraries
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

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

```
     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
..        …       …       …     …      …      …        …        …       …
886         0       2    male  27.0      0      0  13.0000        S  Second
887         1       1  female  19.0      0      0  30.0000        S   First
888         0       3  female   NaN      1      2  23.4500        S   Third
889         1       1    male  26.0      0      0  30.0000        C   First
890         0       3    male  32.0      0      0   7.7500        Q   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
..     …           …   …          …     …     …
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

[891 rows x 15 columns]
```

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

[ ]:
```python
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
```

[ ]:
```
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
```

[ ]:
```
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
```

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

```
(891, 15)
```

[ ]:
```python
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 |

[ ]:
```python
# 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
```

[ ]:
```python
# Columns name
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["who"].unique()
```

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

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

```
array([ True, False])
```

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

### 1.2 Section 7.1: Cleaning and Filtering 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
```

[ ]:
```
# removing missing value column (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
```

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

```
(891, 14)
```

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

```
     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
..        …       …       …   …      …      …        …       …       …
885         0       3  female  39.0      0      5  29.1250        Q   Third
886         0       2    male  27.0      0      0  13.0000        S  Second
887         1       1  female  19.0      0      0  30.0000        S   First
889         1       1    male  26.0      0      0  30.0000        C   First
890         0       3    male  32.0      0      0   7.7500        Q   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
..     ...         ...          ...   ...    ...
885  woman       False   Queenstown    no  False
886    man        True  Southampton    no   True
887  woman       False  Southampton   yes   True
889    man        True    Cherbourg   yes   True
890    man        True   Queenstown    no   True

[712 rows x 14 columns]
```

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

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

```
(712, 14)
```

[ ]:
```python
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
```

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

```
(712, 14)
```

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

```
(891, 15)
```

[ ]:
```python
ks_clean["sex"].value_counts()
```

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

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

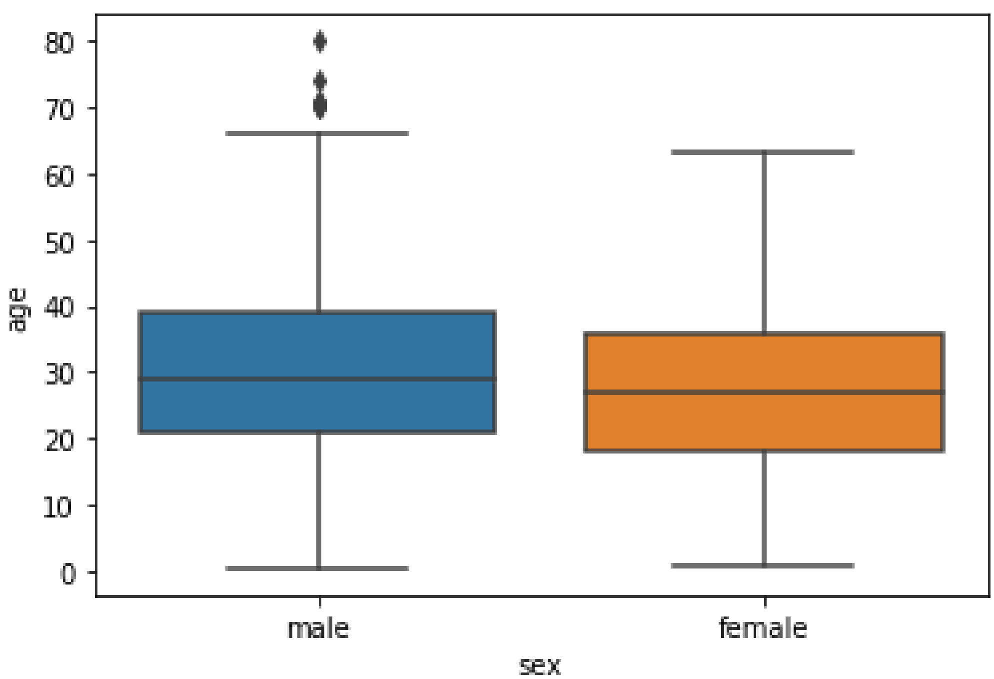
| | survived | pclass | age | sibsp | parch | fare |
|---|---|---|---|---|---|---|
| count | 891.000000 | 891.000000 | 714.000000 | 891.000000 | 891.000000 | 891.000000 |
| mean | 0.383838 | 2.308642 | 29.699118 | 0.523008 | 0.381594 | 32.204208 |
| std | 0.486592 | 0.836071 | 14.526497 | 1.102743 | 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_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 |

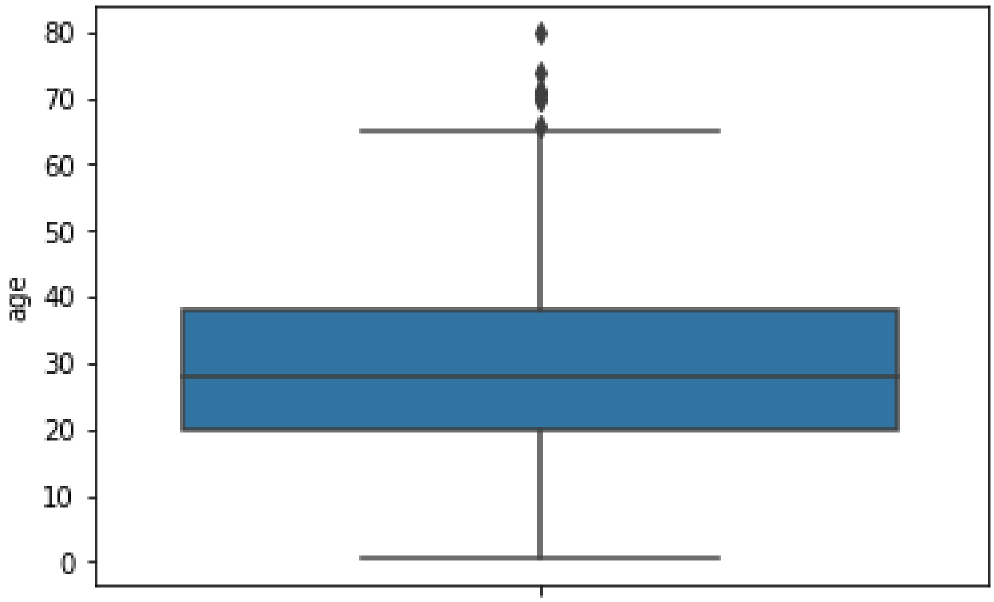
[ ]:
```
sns.boxplot(x = 'sex', y = 'age', data = ks_clean)
```

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



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

<AxesSubplot:ylabel='age'>

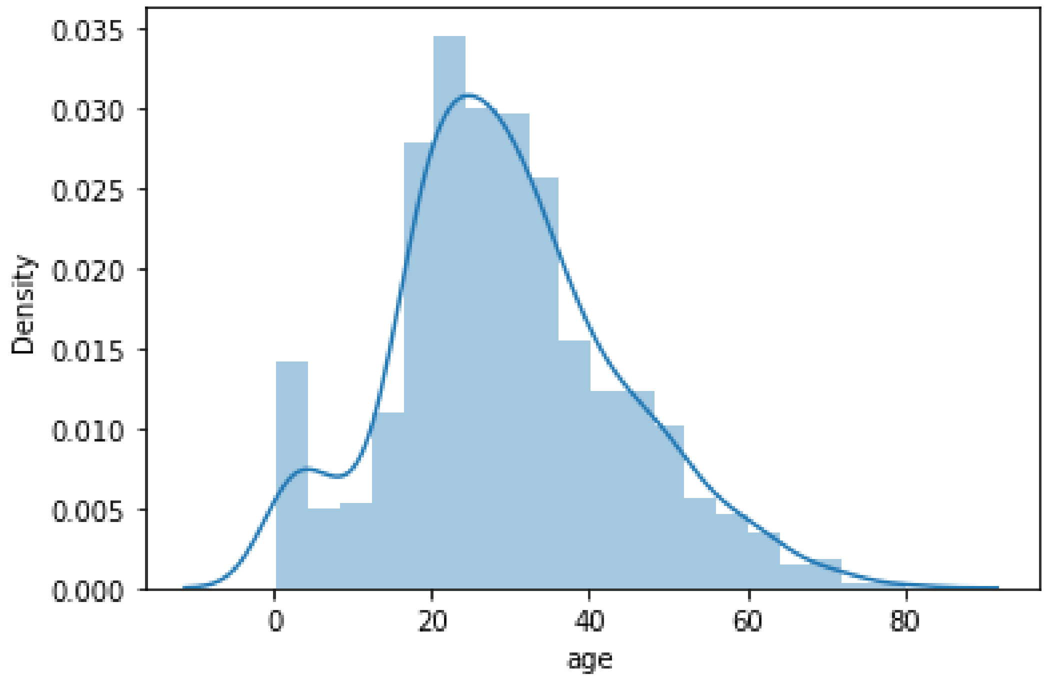



[ ]:
```
sns.distplot(ks_clean['age'])
```

C:\Users\ammar\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).

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




```
[ ]: # Out liers removal
     ks_clean['age'].mean()
```

```
29.64209269662921
```

```
[ ]: ks_clean = ks_clean[ks_clean['age'] < 68]
     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.shape
```

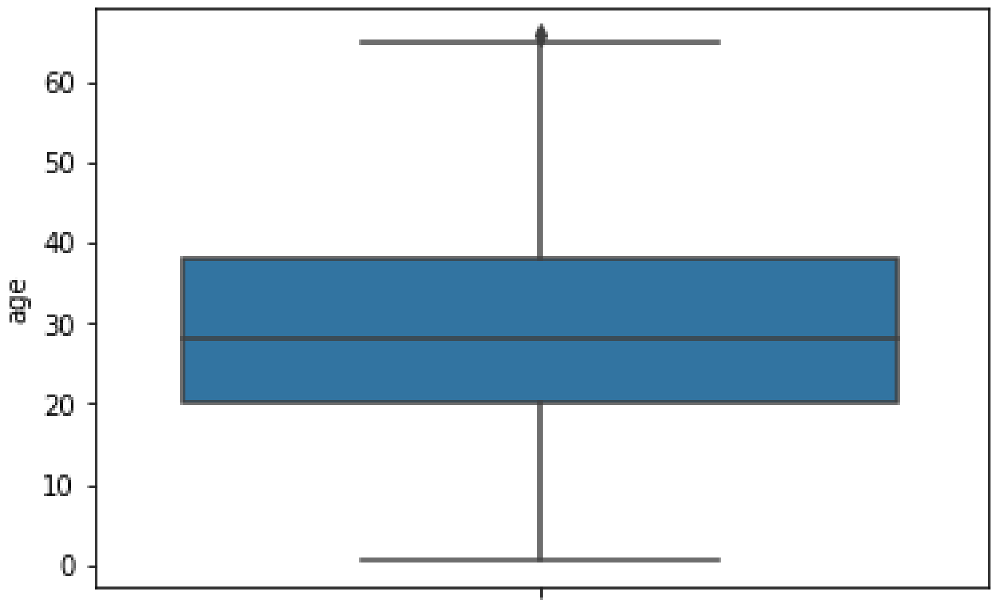
```
(705, 14)
```

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

29.21797163120567

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

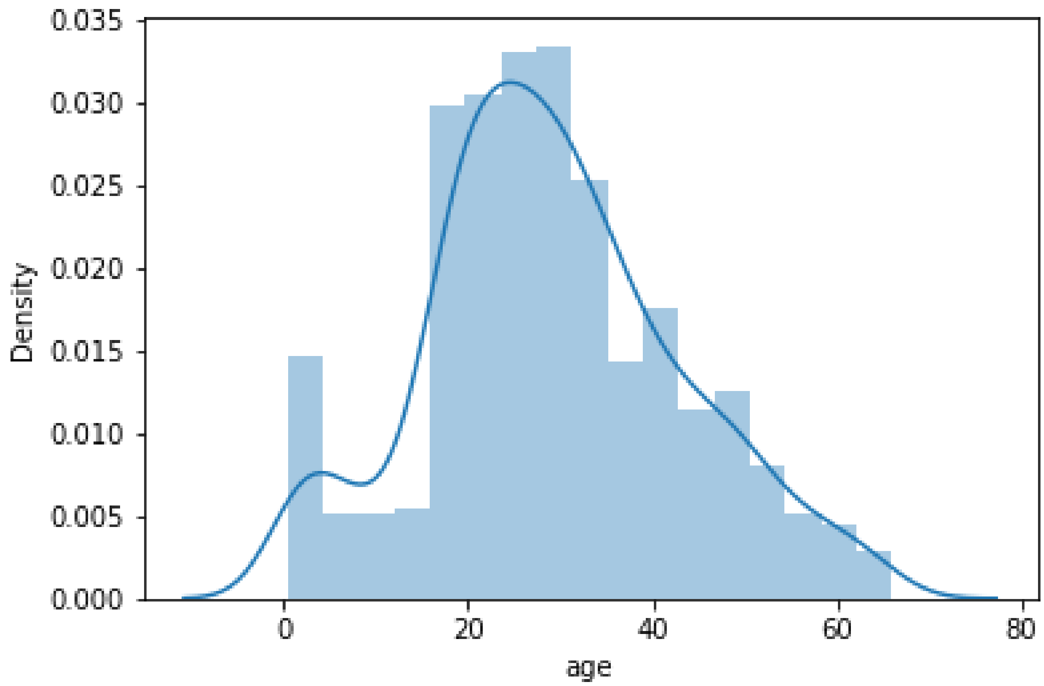
<AxesSubplot:ylabel='age'>




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

```
C:\Users\ammar\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).
```

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

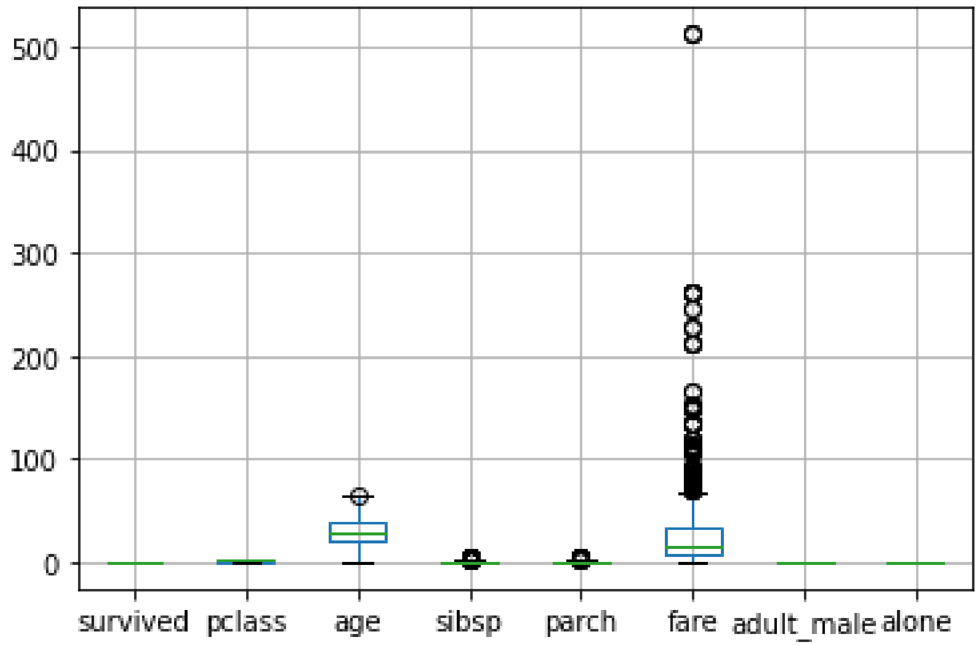



[ ]:
```
ks_clean.head()
```

```
   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.boxplot()
```

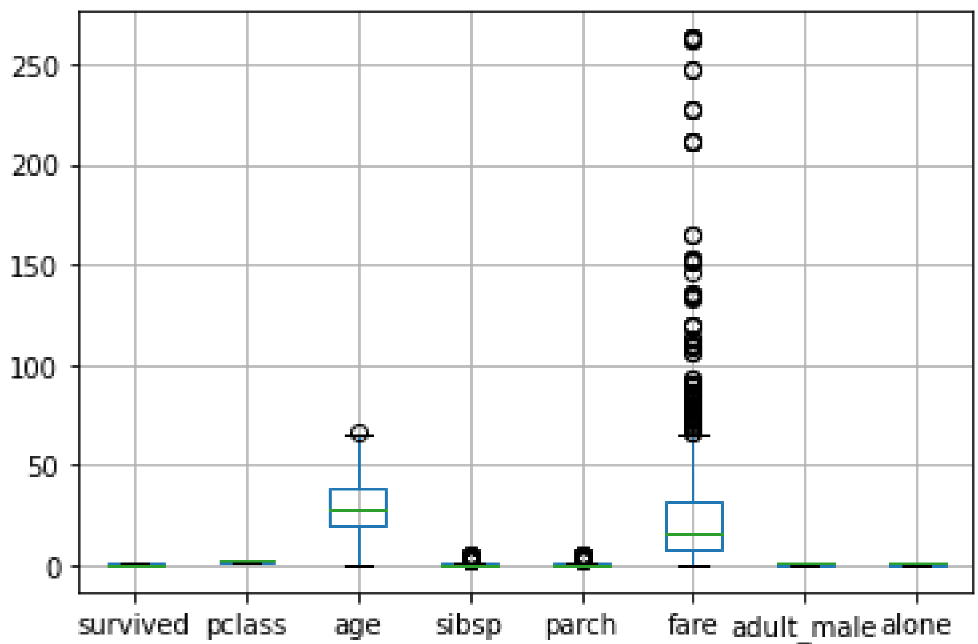
```
<AxesSubplot:>
```




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

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

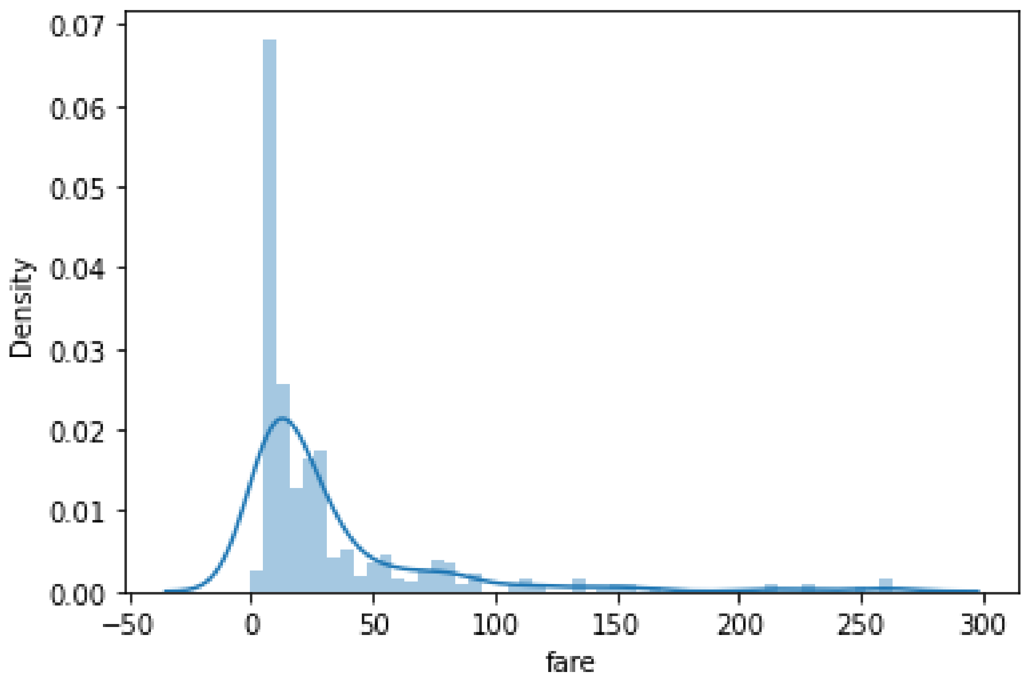
<AxesSubplot:>



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

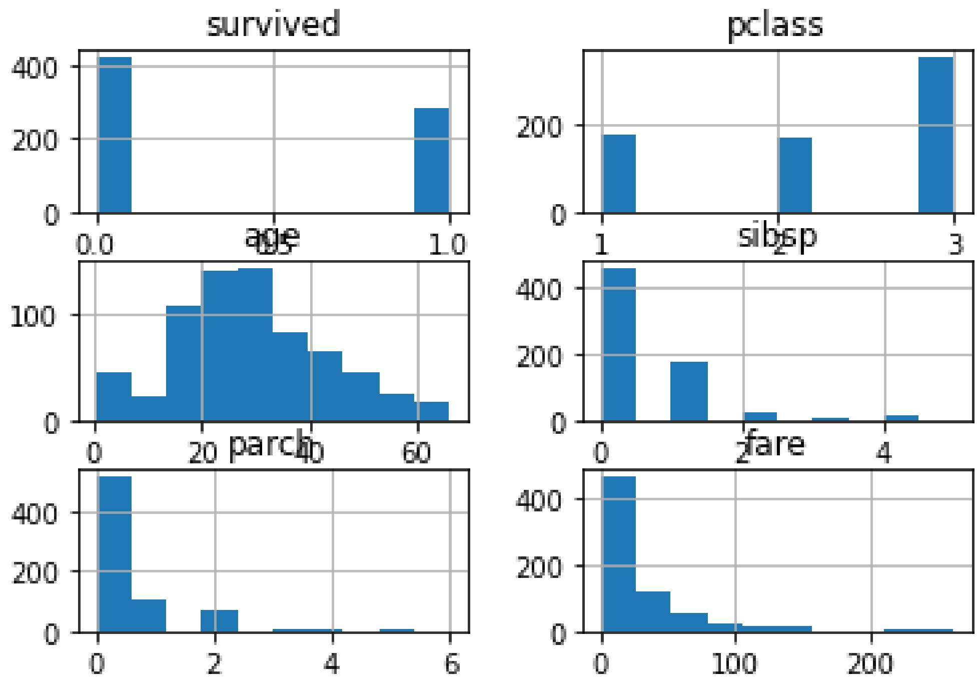
```
C:\Users\ammar\AppData\Local\Programs\Python\Python310\lib\site-
packages\seaborn\distributions.py:2619: FutureWarning:

`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).
```

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



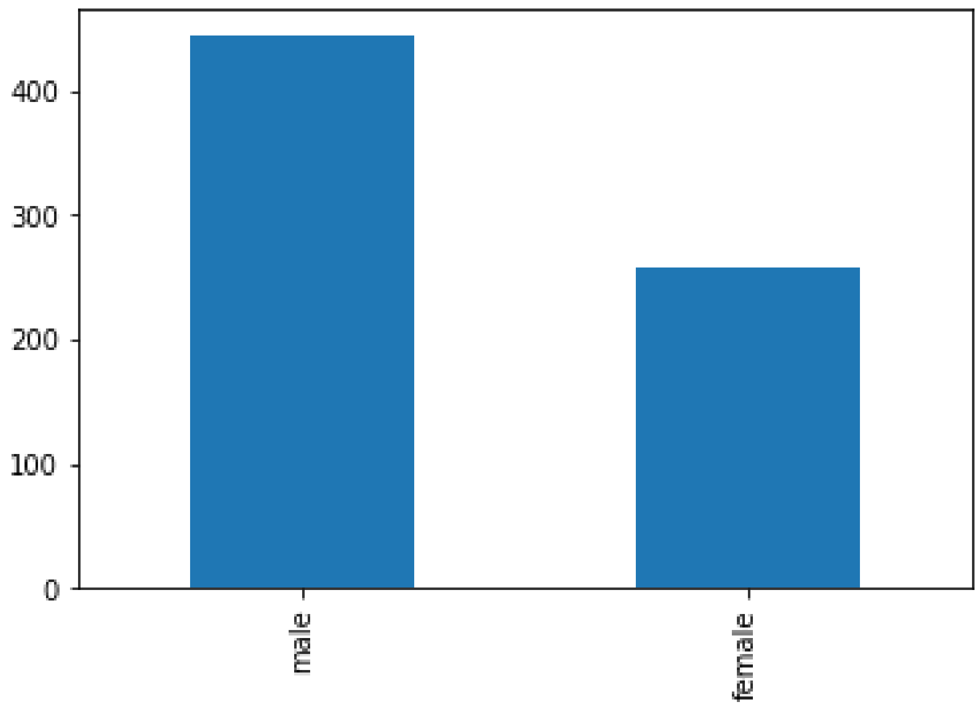

[ ]:
```python
ks_clean.hist()
```

```
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)
```




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

```
<AxesSubplot:>
```



[ ]:
```
ks_clean.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' ,'who']).mean()
```

```
                     survived  pclass        age     sibsp     parch  \
sex    class  who
female First  child  0.666667     1.0  10.333333  0.666667  1.666667
              man         NaN     NaN        NaN       NaN       NaN
              woman  0.978022     1.0  35.500000  0.549451  0.417582
       Second child  1.000000     2.0   6.600000  0.700000  1.300000
              man         NaN     NaN        NaN       NaN       NaN
              woman  0.909091     2.0  32.179688  0.454545  0.500000
       Third  child  0.533333     3.0   7.100000  1.533333  1.100000
              man         NaN     NaN        NaN       NaN       NaN
              woman  0.491228     3.0  27.854167  0.728070  0.719298
male   First  child  1.000000     1.0   5.306667  0.666667  2.000000
              man    0.352941     1.0  42.382653  0.302521  0.235294
              woman       NaN     NaN        NaN       NaN       NaN
       Second child  1.000000     2.0   2.258889  0.888889  1.222222
              man    0.080808     2.0  33.588889  0.292929  0.131313
              woman       NaN     NaN        NaN       NaN       NaN
       Third  child  0.321429     3.0   6.515000  2.821429  1.321429
              man    0.119122     3.0  28.995556  0.294671  0.128527
              woman       NaN     NaN        NaN       NaN       NaN

                           fare  adult_male     alone
sex    class  who
```

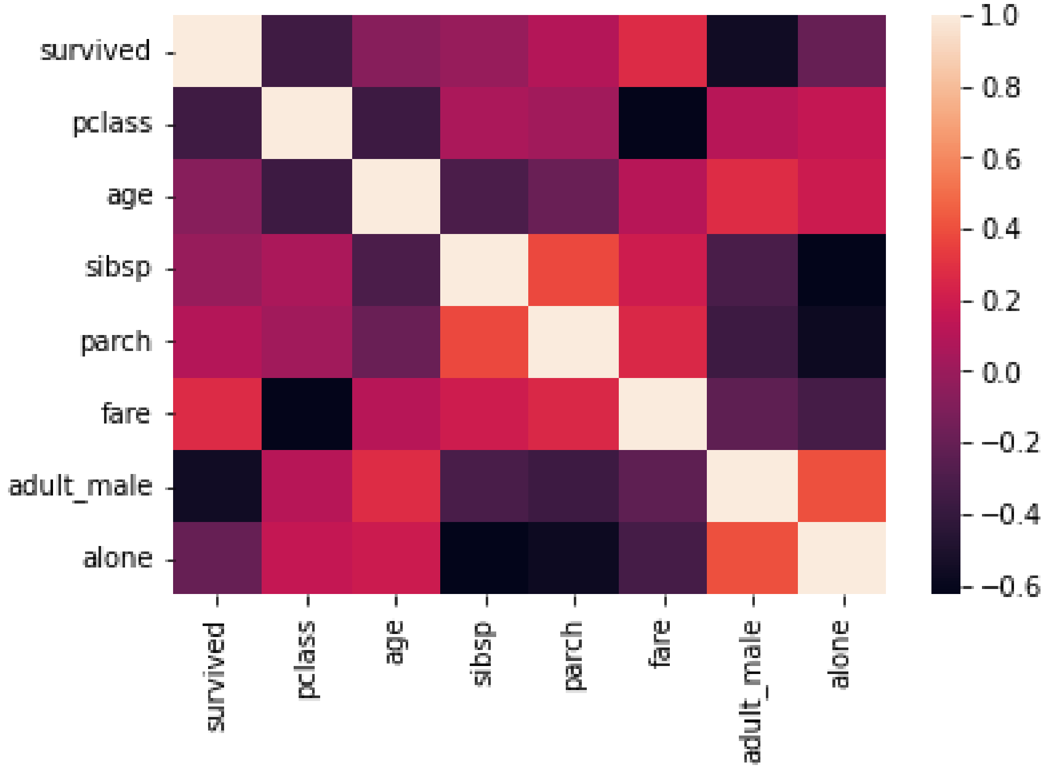
```
female First  child  160.962500         0.0  0.000000
              man           NaN         NaN       NaN
              woman  104.317995         0.0  0.373626
       Second child   29.240000         0.0  0.000000
              man           NaN         NaN       NaN
              woman   20.868624         0.0  0.484848
       Third  child   19.023753         0.0  0.166667
              man           NaN         NaN       NaN
              woman   15.354351         0.0  0.482456
male   First  child  117.802767         0.0  0.000000
              man     65.951086         1.0  0.630252
              woman         NaN         NaN       NaN
       Second child   27.306022         0.0  0.000000
              man     19.054124         1.0  0.727273
              woman         NaN         NaN       NaN
       Third  child   27.716371         0.0  0.035714
              man     11.340213         1.0  0.824451
              woman         NaN         NaN       NaN
```

### 1.3 Section 7.2: Relationship (Correlation)

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

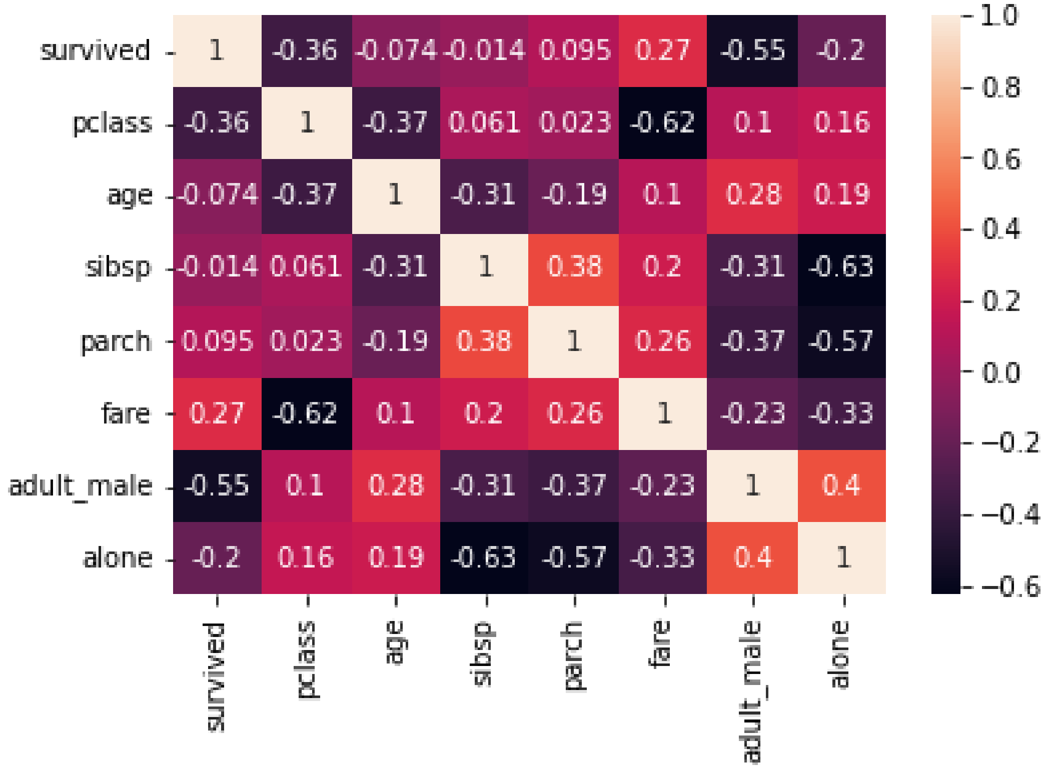
[ ]:
```
sns.heatmap(corr_ks_clean)
```

```
<AxesSubplot:>
```



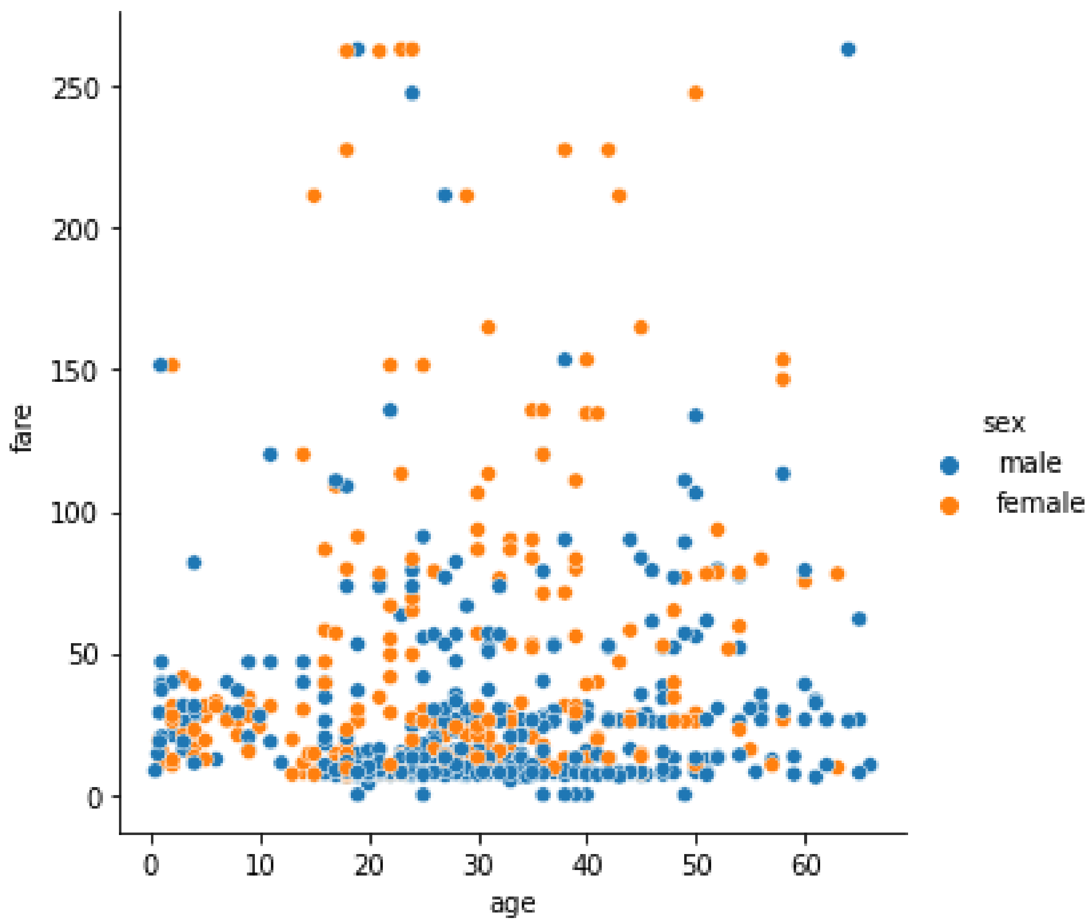

[ ]:
```
sns.heatmap(corr_ks_clean, annot = True)
```

<AxesSubplot:>



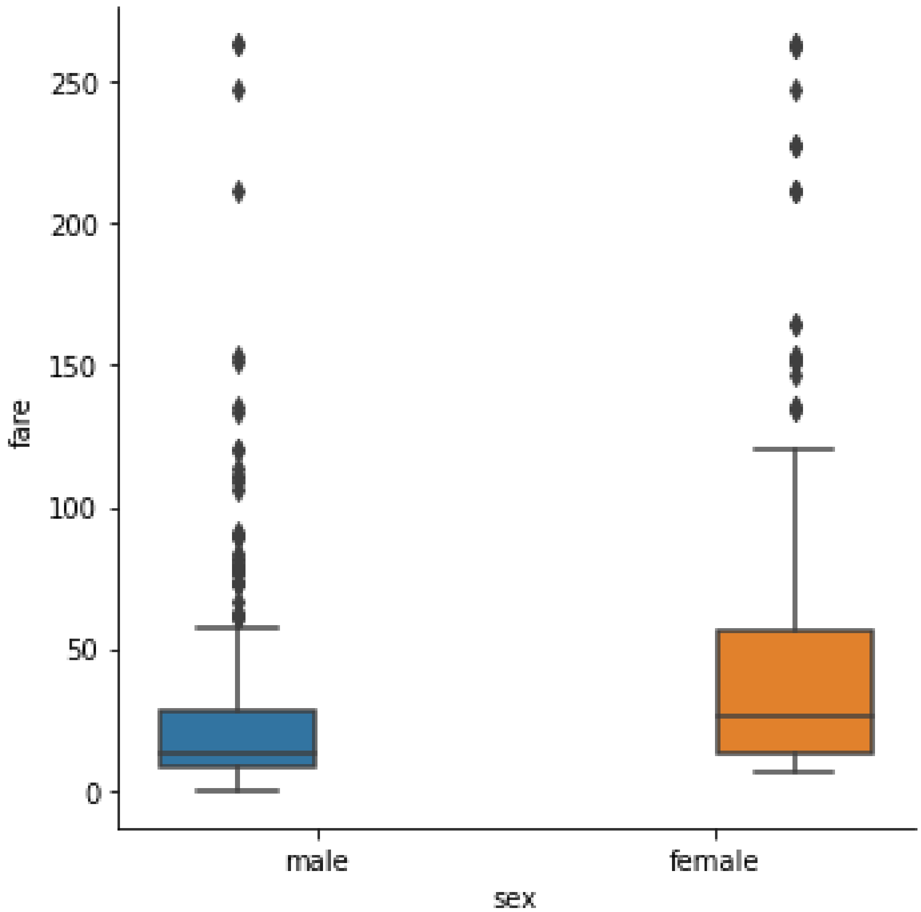

[ ]:
```
sns.relplot(x= 'age', y = 'fare', hue = 'sex' , data = ks_clean)
```

```
<seaborn.axisgrid.FacetGrid at 0x21e3ac04eb0>
```




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

```
<seaborn.axisgrid.FacetGrid at 0x21e3cde34c0>
```




Log Transformation

```
[ ]: ks_clean['fare_log'] = np.log(ks_clean['fare'])
```

```
[ ]: 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 | 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_clean, kind=
    "box")
```

<seaborn.axisgrid.FacetGrid at 0x21e38d027d0>