Data Wrangling Notebook

> **Steps**
>
> - Data collection
> - handling missing val
> - data formating
> - data normalization (scaling, centring)
> - Data binnin (for group of data)
> - making dummies of catagorical data nurmerical data
> - Clean the Data
> - Find a Relationship between data
> - analayize data
> -

In [ ]:
```
import plotly.express as px
import pandas as pd
import numpy as np
import os
import matplotlib.pyplot as plt
import seaborn as sns
```

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

Out[ ]:

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

In [ ]:
```
# ere we will convert the age into days instrad of year
df['age']= df['age']*365
# assignment to remove the zeros
# df['age'] = df['age'].astype('int64')
df.dtypes
```

Out[ ]:
```
survived          int64
pclass            int64
sex               int64
age             float64
sibsp             int64
parch             int64
```

```
fare            float64
embarked         object
class          category
who              object
adult_male         bool
deck           category
embark_town      object
alive            object
alone              bool
dtype: object
```

In [ ]:
```
# two ways
# df_gender = pd.get_dummies(df['sex'])
# df_new = pd.concat([df, df_gender], axis=1)
df['sex'] = df['sex'].map({'male': 1, 'female': 0})

df.head()
```

Out[ ]:

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	deck	emb
0	0	3	1	22.0	1	0	7.2500	S	Third	man	True	NaN	Sou
1	1	1	0	38.0	1	0	71.2833	C	First	woman	False	C	C
2	1	3	0	26.0	0	0	7.9250	S	Third	woman	False	NaN	Sou
3	1	1	0	35.0	1	0	53.1000	S	First	woman	False	C	Sou
4	0	3	1	35.0	0	0	8.0500	S	Third	man	True	NaN	Sou

Binning

> grouping of value into smaller no of val\ convert numeric into categories (1-15)(15-30) etc\ to have better understaing\

In [ ]:
```
pd.qcut(
    df.age,                     # Column to bin
    3,                      # Number of quantiles
    labels=None,            # List of labels to include
    retbins=False,          # Whether to return the bins/labels or not
    precision=3,            # The precision to store and display the bins labels
    duplicates='raise'      # If bin edges are not unique, raise a ValueError
)
```

Out[ ]:
```
0      (0.419, 23.0]
1       (34.0, 80.0]
2       (23.0, 34.0]
3       (34.0, 80.0]
4       (34.0, 80.0]
           ...
886     (23.0, 34.0]
887    (0.419, 23.0]
888              NaN
889     (23.0, 34.0]
890     (23.0, 34.0]
```

```
Name: age, Length: 891, dtype: category
Categories (3, interval[float64]): [(0.419, 23.0] < (23.0, 34.0] < (34.0, 80.0]]
```

In [ ]:
```
df['Age Groups'] = pd.qcut(df['age'], 4)
df.head()
```

Out[ ]:

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

In [ ]:
```
df['Age Groups'] = pd.qcut(
   df['age'],
   [0, 0.25, 0.5, 0.75, 1],
   labels=['0-25%', '26-49%', '51-75%', '76-100%']
)
df.head()
```

Out[ ]:

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

In [ ]:
```

```

EDA in Python

> **Steps**
>
> - Understand the Data
> - Clean the Data

- Fimd a Relationship between data

In [ ]:
```
import plotly.express as px
import pandas as pd
import numpy as np
import os
import matplotlib.pyplot as plt
import seaborn as sns
```

In [ ]:
```
df = sns.load_dataset('titanic')
# df = pd.read_csv('/asdf/asdf/titanic.csv')
df.head(5)
```

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

In [ ]:
```
df.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

In [ ]:
```
df.shape
```

```
(891, 15)
```

In [ ]:
```
# unique values checking in data
df.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
```

In [ ]:
```
# col names
df.columns
```

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

In [ ]:
```
df['sex'].unique()
```

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

In [ ]:
```
df['age'].unique()
```

```
array([22.  , 38.  , 26.  , 35.  ,   nan, 54.  ,  2.  , 27.  , 14.  ,
        4.  , 58.  , 20.  , 39.  , 55.  , 31.  , 34.  , 15.  , 28.  ,
        8.  , 19.  , 40.  , 66.  , 42.  , 21.  , 18.  ,  3.  ,  7.  ,
       49.  , 29.  , 65.  , 28.5 ,  5.  , 11.  , 45.  , 17.  , 32.  ,
       16.  , 25.  ,  0.83, 30.  , 33.  , 23.  , 24.  , 46.  , 59.  ,
       71.  , 37.  , 47.  , 14.5 , 70.5 , 32.5 , 12.  ,  9.  , 36.5 ,
       51.  , 55.5 , 40.5 , 44.  ,  1.  , 61.  , 56.  , 50.  , 36.  ,
       45.5 , 20.5 , 62.  , 41.  , 52.  , 63.  , 23.5 ,  0.92, 43.  ,
       60.  , 10.  , 64.  , 13.  , 48.  ,  0.75, 53.  , 57.  , 80.  ,
       70.  , 24.5 ,  6.  ,  0.67, 30.5 ,  0.42, 34.5 , 74.  ])
```

In [ ]:
```
df['who'].unique()
```

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

In [ ]:
```
# Assignment
pd.unique(df[['sex', 'who' , 'age', 'fare']].values.ravel('K'))
```

```
array(['male', 'female', 'man', 'woman', 'child', 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, 1.0,
       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, 7.25, 71.2833,
       7.925, 53.1, 8.05, 8.4583, 51.8625, 21.075, 11.1333, 30.0708, 16.7,
       26.55, 31.275, 7.8542, 29.125, 7.225, 8.0292, 35.5, 31.3875, 263.0,
       7.8792, 7.8958, 27.7208, 146.5208, 7.75, 10.5, 82.1708, 7.2292,
       11.2417, 9.475, 41.5792, 15.5, 21.6792, 17.8, 39.6875, 7.8,
       76.7292, 61.9792, 27.75, 46.9, 83.475, 27.9, 15.2458, 8.1583,
```

```
8.6625, 73.5, 14.4542, 56.4958, 7.65, 12.475, 9.5, 7.7875, 47.1,
15.85, 34.375, 61.175, 20.575, 34.6542, 63.3583, 77.2875, 8.6542,
7.775, 24.15, 9.825, 14.4583, 247.5208, 7.1417, 22.3583, 6.975,
7.05, 15.0458, 26.2833, 9.2167, 79.2, 6.75, 11.5, 36.75, 7.7958,
12.525, 66.6, 7.3125, 61.3792, 7.7333, 69.55, 16.1, 15.75, 20.525,
25.925, 33.5, 30.6958, 25.4667, 28.7125, 0.0, 15.05, 22.025,
8.4042, 6.4958, 10.4625, 18.7875, 113.275, 76.2917, 90.0, 9.35,
13.5, 7.55, 26.25, 12.275, 7.125, 52.5542, 20.2125, 86.5, 512.3292,
79.65, 153.4625, 135.6333, 19.5, 29.7, 77.9583, 20.25, 78.85,
91.0792, 12.875, 8.85, 151.55, 23.25, 12.35, 110.8833, 108.9,
56.9292, 83.1583, 262.375, 164.8667, 134.5, 6.2375, 57.9792,
133.65, 15.9, 9.225, 75.25, 69.3, 55.4417, 211.5, 4.0125, 227.525,
15.7417, 7.7292, 120.0, 12.65, 18.75, 6.8583, 7.875, 14.4, 55.9,
8.1125, 81.8583, 19.2583, 19.9667, 89.1042, 38.5, 7.725, 13.7917,
9.8375, 7.0458, 7.5208, 12.2875, 9.5875, 49.5042, 78.2667, 15.1,
7.6292, 22.525, 26.2875, 59.4, 7.4958, 34.0208, 93.5, 221.7792,
106.425, 49.5, 13.8625, 7.8292, 39.6, 17.4, 51.4792, 26.3875,
40.125, 8.7125, 42.4, 15.55, 32.3208, 7.0542, 8.4333, 25.5875,
9.8417, 8.1375, 10.1708, 211.3375, 13.4167, 7.7417, 9.4833, 7.7375,
8.3625, 23.45, 25.9292, 8.6833, 8.5167, 7.8875, 37.0042, 6.45,
6.95, 8.3, 6.4375, 39.4, 14.1083, 13.8583, 50.4958, 9.8458,
10.5167], dtype=object)
```

Cleaning and Filtering the Data

> Finding missing value Findnig

In [ ]:
```
df.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
```

In [ ]:
```
# droping the col
dff = df.drop(['deck'], axis= 1)
dff.head()
```

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

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	embark_
4	0	3	male	35.0	0	0	8.0500	S	Third	man	True	Southar

In [ ]:
```
dff = dff.dropna()
dff.head(2)
```

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	embark_
0	0	3	male	22.0	1	0	7.2500	S	Third	man	True	Southar
1	1	1	female	38.0	1	0	71.2833	C	First	woman	False	Cherl

In [ ]:
```
dff.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
```

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

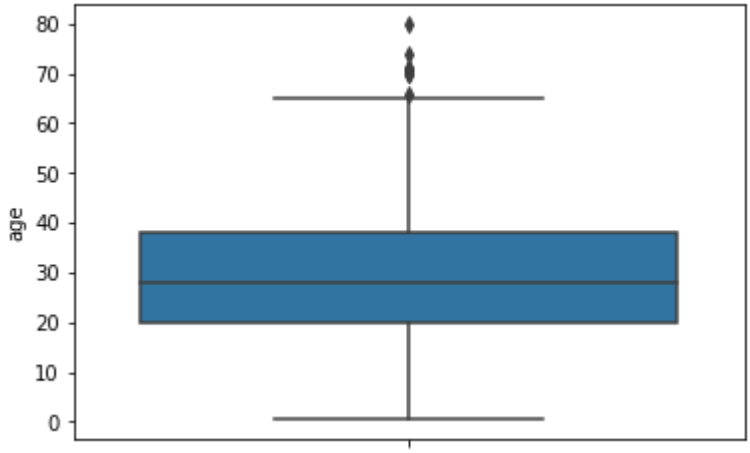
```
male      453
female    259
Name: sex, dtype: int64
```

In [ ]:
```
dff.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

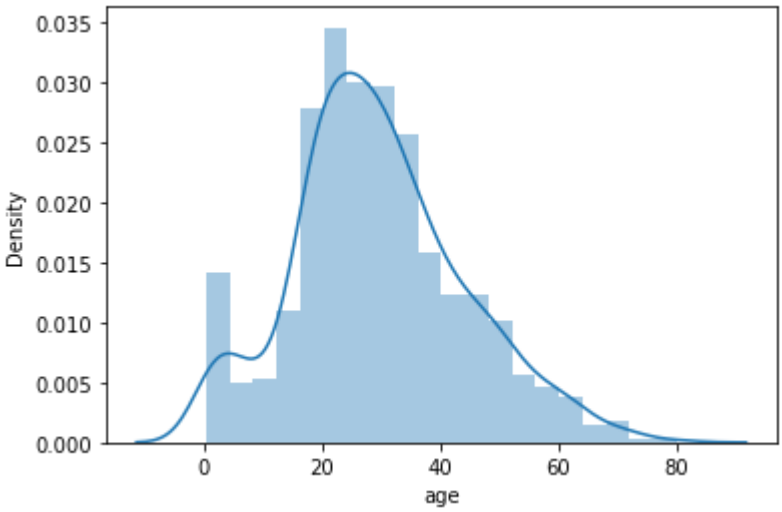
In [ ]:
```
# out lier finding
sns.boxplot( y = 'age', data = dff)#x = 'sex',
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x20bd2890ac8>
```



In [ ]:
```
sns.distplot(df['age'])# normality check or disperstion zaida hy so for ferfactly data
```

```
C:\Users\Ali\anaconda3\envs\python-chilla\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).

<matplotlib.axes._subplots.AxesSubplot at 0x20bd3006400>
```



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

```
29.64209269662921
```

In [ ]:
```
dff =  dff[dff['age']< 68]
dff.head()
```

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	embark_

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

In [ ]:
```
print(dff.shape)
dff.head(2)
```

```
(705, 14)
```

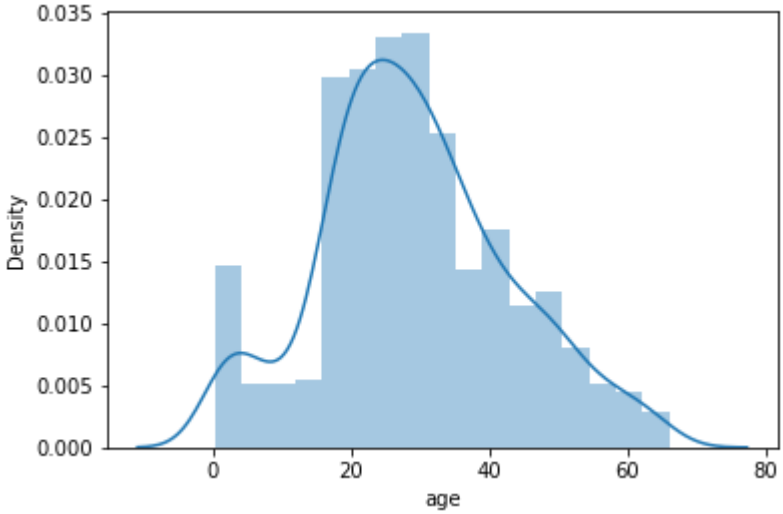
	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	embark_
0	0	3	male	22.0	1	0	7.2500	S	Third	man	True	Southan
1	1	1	female	38.0	1	0	71.2833	C	First	woman	False	Cherl

In [ ]:
```
dff.age.value_counts()
```

```
24.00    30
22.00    27
18.00    26
19.00    25
28.00    25
         ..
55.50     1
36.50     1
12.00     1
14.50     1
0.42      1
Name: age, Length: 83, dtype: int64
```

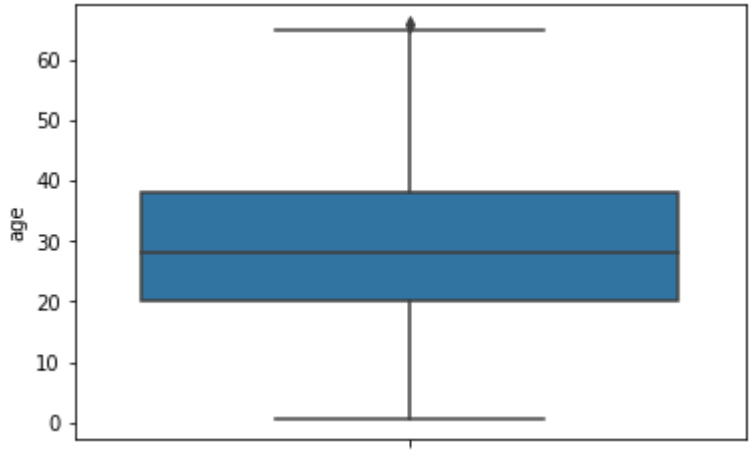
In [ ]:
```
sns.distplot( dff['age'])
```

```
C:\Users\Ali\anaconda3\envs\python-chilla\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).

<matplotlib.axes._subplots.AxesSubplot at 0x20bd30c6278>
```



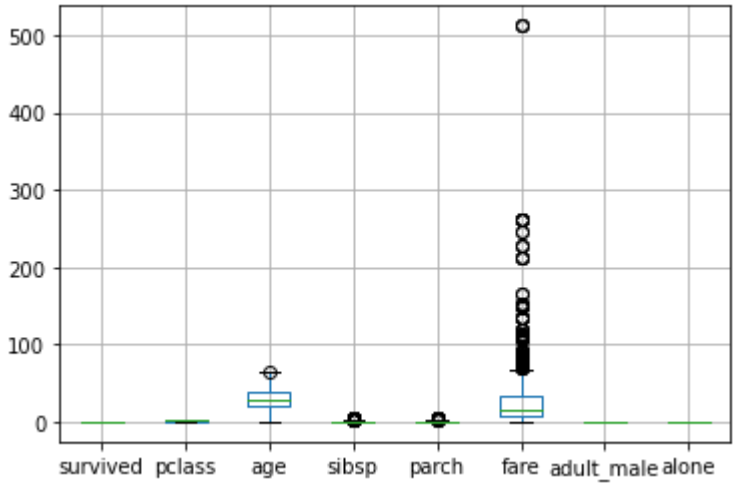
In [ ]:
```
sns.boxplot(y= 'age', data= dff)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x20bd31694e0>
```



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

```
<matplotlib.axes._subplots.AxesSubplot at 0x20bd31f5e48>
```

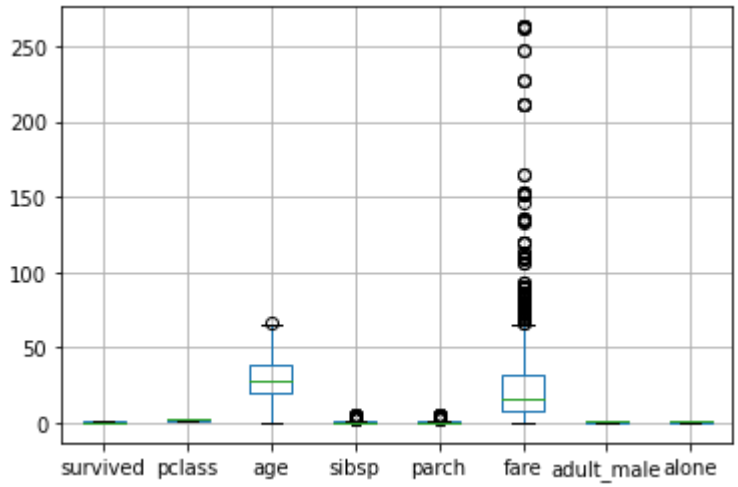


In [ ]:
```
dff = dff[dff['fare']< 300]
dff.head()
```

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

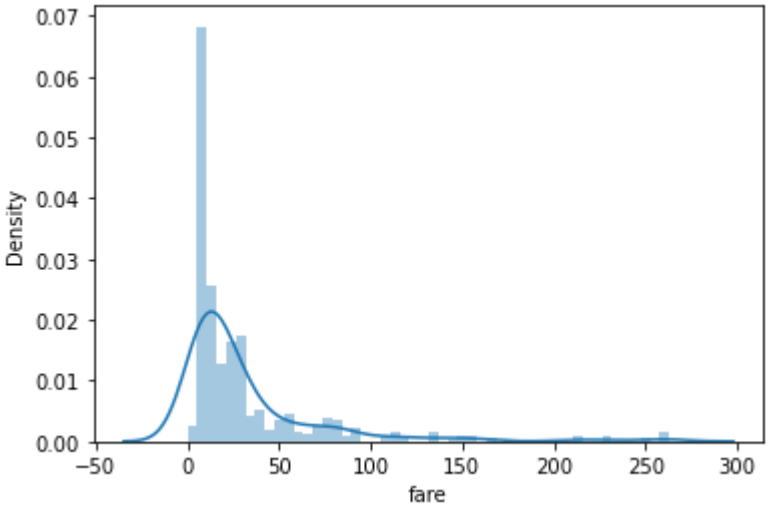
In [ ]:
```
dff.boxplot()
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x20bd32de128>
```



In [ ]:
```
sns.distplot(dff['fare'])
```

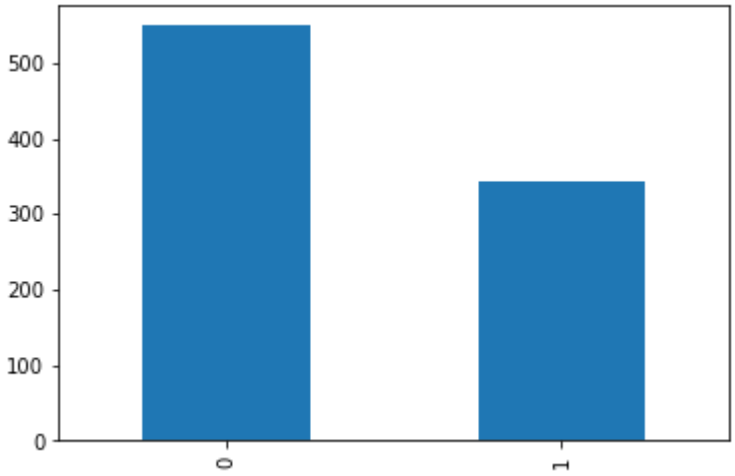
```
C:\Users\Ali\anaconda3\envs\python-chilla\lib\site-packages\seaborn\distributions.py:2619: FutureWarning:

`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).

<matplotlib.axes._subplots.AxesSubplot at 0x20bd342c240>
```



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

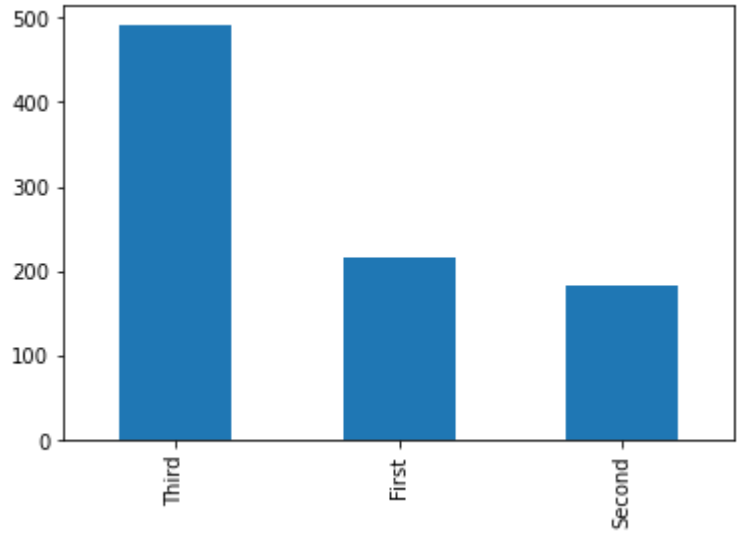
In [ ]:
```
pd.value_counts(df['survived']).plot.bar()
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x20bd59c66a0>
```



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

```
<matplotlib.axes._subplots.AxesSubplot at 0x20bd5a3a588>
```



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

sex	survived	pclass	age	sibsp	parch	fare	adult_male	alone
female	0.751938	2.077519	27.717054	0.647287	0.717054	45.530120	0.00000	0.375969
male	0.202703	2.351351	30.048806	0.445946	0.272523	25.038155	0.90991	0.668919

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

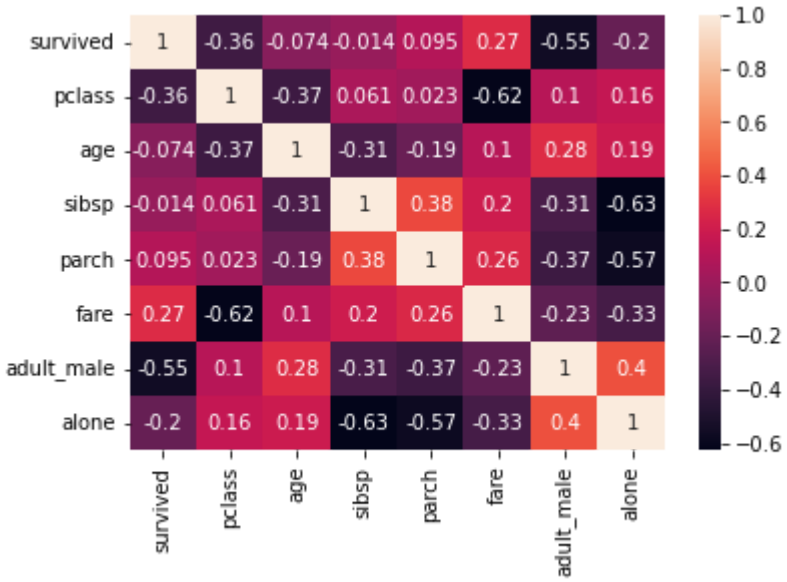
sex	class	survived	pclass	age	sibsp	parch	fare	adult_male	alone
female	**First**	0.963415	1.0	34.231707	0.560976	0.512195	103.696393	0.000000	0.353659
	Second	0.918919	2.0	28.722973	0.500000	0.621622	21.951070	0.000000	0.405405
	Third	0.460784	3.0	21.750000	0.823529	0.950980	15.875369	0.000000	0.372549
male	**First**	0.389474	1.0	40.067579	0.389474	0.336842	62.901096	0.968421	0.526316
	Second	0.153061	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 or Correlation

In [ ]:
```
cor = dff.corr() #do variable ka relation k interactioon ak k bharny say dosra bhar rah
```

In [ ]:
```
sns.heatmap(cor, annot = True)#only numerical data corelation can be find
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x20bd5d1e7b8>
```



In [ ]:
```
sns.relplot(x = 'age', y= 'sex', hue= 'sex', date = dff)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-51-0267b8f67305> in <module>
----> 1 sns.relplot(x = 'age', y= 'sex', hue= 'sex', date = dff)


~\anaconda3\envs\python-chilla\lib\site-packages\seaborn\_decorators.py in inner_f(*arg
s, **kwargs)
     44             )
     45         kwargs.update({k: arg for k, arg in zip(sig.parameters, args)})
---> 46         return f(**kwargs)
     47     return inner_f
     48


~\anaconda3\envs\python-chilla\lib\site-packages\seaborn\relational.py in relplot(x, y,
 hue, size, style, data, row, col, col_wrap, row_order, col_order, palette, hue_order, h
ue_norm, sizes, size_order, size_norm, markers, dashes, style_order, legend, kind, heigh
t, aspect, facet_kws, units, **kwargs)
    948         data=data,
    949         variables=plotter.get_semantics(locals()),
--> 950         legend=legend,
    951     )
    952     p.map_hue(palette=palette, order=hue_order, norm=hue_norm)
```

```
~\anaconda3\envs\python-chilla\lib\site-packages\seaborn\relational.py in __init__(self, data, variables, x_bins, y_bins, estimator, ci, n_boot, alpha, x_jitter, y_jitter, legend)

    585         )
    586
--> 587         super().__init__(data=data, variables=variables)
    588
    589         self.alpha = alpha


~\anaconda3\envs\python-chilla\lib\site-packages\seaborn\_core.py in __init__(self, data, variables)

    603     def __init__(self, data=None, variables={}):
    604
--> 605         self.assign_variables(data, variables)
    606
    607         for var, cls in self._semantic_mappings.items():


~\anaconda3\envs\python-chilla\lib\site-packages\seaborn\_core.py in assign_variables(self, data, variables)

    667             self.input_format = "long"
    668             plot_data, variables = self._assign_variables_longform(
--> 669                 data, **variables,
    670             )
    671


~\anaconda3\envs\python-chilla\lib\site-packages\seaborn\_core.py in _assign_variables_longform(self, data, **kwargs)

    901
    902                 err = f"Could not interpret value `{val}` for parameter `{key}`"
--> 903                 raise ValueError(err)
    904
    905             else:


ValueError: Could not interpret value `sex` for parameter `hue`
```

In [ ]:
```
dff
```

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	eml
0	0	3	male	22.0	1	0	7.2500	S	Third	man	True	Sou
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	Sou
3	1	1	female	35.0	1	0	53.1000	S	First	woman	False	Sou
4	0	3	male	35.0	0	0	8.0500	S	Third	man	True	Sou
...	...	...	...	...	...	...	...	...	...	...	...	
885	0	3	female	39.0	0	5	29.1250	Q	Third	woman	False	Q
886	0	2	male	27.0	0	0	13.0000	S	Second	man	True	Sou
887	1	1	female	19.0	0	0	30.0000	S	First	woman	False	Sou
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	Q

702 rows × 14 columns

In [ ]:
```
sns.catplot(x = 'sex', y= 'fare', hue= 'sex', date = dff, kind = 'box')
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-59-919ca77c5291> in <module>
----> 1 sns.catplot(x = 'sex', y= 'fare', hue= 'sex', date = dff, kind = 'box')


~\anaconda3\envs\python-chilla\lib\site-packages\seaborn\_decorators.py in inner_f(*args, **kwargs)
     44             )
     45         kwargs.update({k: arg for k, arg in zip(sig.parameters, args)})
---> 46         return f(**kwargs)
     47     return inner_f
     48


~\anaconda3\envs\python-chilla\lib\site-packages\seaborn\categorical.py in catplot(x, y, hue, data, row, col, col_wrap, estimator, ci, n_boot, units, seed, order, hue_order, row_order, col_order, kind, height, aspect, orient, color, palette, legend, legend_out, sharex, sharey, margin_titles, facet_kws, **kwargs)
```

```
   3790     p = _CategoricalPlotter()
   3791     p.require_numeric = plotter_class.require_numeric
-> 3792     p.establish_variables(x_, y_, hue, data, orient, order, hue_order)
   3793     if (
   3794         order is not None
```

~\anaconda3\envs\python-chilla\lib\site-packages\seaborn\categorical.py in establish_variables(self, x, y, hue, data, orient, order, hue_order, units)

```
    151                 if isinstance(var, str):
    152                     err = "Could not interpret input '{}'".format(var)
--> 153                     raise ValueError(err)
    154
    155             # Figure out the plotting orientation
```

ValueError: Could not interpret input 'sex'

In [ ]: