



 main ▾

...

CLG\_CHUTİYAP / DSBDA / 31112\_ASSIGNMENT\_08.ipynb

 mean-understanding45 \* 

History

 1 contributor

1865 lines (1865 sloc) | 188 KB

...

# Data Visualization I

1. Use the inbuilt dataset 'titanic'. The dataset contains 891 rows and contains information about the passengers who boarded the unfortunate Titanic ship. Use the Seaborn library to see if we can find any patterns in the data.
2. Write a code to check how the price of the ticket (column name: 'fare') for each passenger is distributed by plotting a histogram.

```
In [1]: import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

```
In [2]: titanic = sns.load_dataset('titanic')
titanic_copy = titanic.copy()
titanic.head()
```

```
Out[2]:
```

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

```
In [3]: titanic
```

```
Out[3]:
```

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	deceased
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	NaN
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	NaN
4	0	3	male	35.0	0	0	8.0500	S	Third	man	True	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...
886	0	2	male	27.0	0	0	13.0000	S	Second	man	True	NaN
887	1	1	female	19.0	0	0	30.0000	S	First	woman	False	NaN
888	0	3	female	NaN	1	2	23.4500	S	Third	woman	False	NaN
889	1	1	male	26.0	0	0	30.0000	C	First	man	True	NaN
890	0	3	male	32.0	0	0	7.7500	Q	Third	man	True	NaN

891 rows × 15 columns

```
In [4]: titanic.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 [5]: `titanic.describe()`

	survived	pclass	age	sibsp	parch	fare
<b>count</b>	891.000000	891.000000	714.000000	891.000000	891.000000	891.000000
<b>mean</b>	0.383838	2.308642	29.699118	0.523008	0.381594	32.204208
<b>std</b>	0.486592	0.836071	14.526497	1.102743	0.806057	49.693429
<b>min</b>	0.000000	1.000000	0.420000	0.000000	0.000000	0.000000
<b>25%</b>	0.000000	2.000000	20.125000	0.000000	0.000000	7.910400
<b>50%</b>	0.000000	3.000000	28.000000	0.000000	0.000000	14.454200
<b>75%</b>	1.000000	3.000000	38.000000	1.000000	0.000000	31.000000
<b>max</b>	1.000000	3.000000	80.000000	8.000000	6.000000	512.329200

In [6]: `titanic.isna().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 [7]: `titanic.columns`

Out[7]: Index(['survived', 'pclass', 'sex', 'age', 'sibsp', 'parch', 'fare', 'embarked', 'class', 'who', 'adult\_male', 'deck', 'embark\_town', 'alive', 'alone'], dtype='object')

In [8]: `titanic.drop(['deck'], axis = 1, inplace = True)`  
`titanic`

```
Out[8]:
```

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male
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
...	...	...	...	...	...	...	...	...	...	...	...
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
888	0	3	female	NaN	1	2	23.4500	S	Third	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

891 rows × 14 columns



titanic.columns

```
In [9]: titanic.columns
```

```
Out[9]: Index(['survived', 'pclass', 'sex', 'age', 'sibsp', 'parch', 'fare',
            'embarked', 'class', 'who', 'adult_male', 'embark_town', 'alive',
            'alone'],
            dtype='object')
```

```
In [10]: titanic['embarked']
```

```
Out[10]: 0      S
         1      C
         2      S
         3      S
         4      S
         ..
        886     S
        887     S
        888     S
        889     C
        890     Q
        Name: embarked, Length: 891, dtype: object
```

```
In [ ]:
```

```
In [11]: titanic['embarked'].value_counts()
```

```
Out[11]: S      644
         C      168
         Q       77
        Name: embarked, dtype: int64
```

```
In [12]: titanic['embarked'].fillna(titanic['embarked'].mode()[0], inplace = True)
         titanic['embarked'].value_counts()
```

```
Out[12]: S      646
         C      168
         Q       77
        Name: embarked, dtype: int64
```

```
In [13]: titanic['embarked'].isna().sum()
```

```
Out[13]: 0
```

```
In [14]: titanic['embark_town'].value_counts()
```

```
Out[14]: Southampton    644  
Cherbourg             168  
Queenstown            77  
Name: embark_town, dtype: int64
```

```
In [15]: titanic['embark_town'].fillna(titanic['embark_town'].mode()[0], inplace = True)  
titanic['embark_town'].value_counts()
```

```
Out[15]: Southampton    646  
Cherbourg             168  
Queenstown            77  
Name: embark_town, dtype: int64
```

```
In [16]: titanic['embark_town'].isna().sum()
```

```
Out[16]: 0
```

```
In [17]: titanic_copy['age'].describe()
```

```
Out[17]: count      714.000000  
mean        29.699118  
std         14.526497  
min          0.420000  
25%         20.125000  
50%         28.000000  
75%         38.000000  
max         80.000000  
Name: age, dtype: float64
```

```
In [18]: titanic['age'].fillna(titanic['age'].mean(), inplace = True)  
titanic['age'].value_counts()
```

```
Out[18]: 29.699118    177  
24.000000     30  
22.000000     27  
18.000000     26  
28.000000     25  
...  
36.500000      1  
55.500000      1  
0.920000       1  
23.500000      1  
74.000000      1  
Name: age, Length: 89, dtype: int64
```

```
In [19]: titanic['age'].isna().sum()
```

```
Out[19]: 0
```

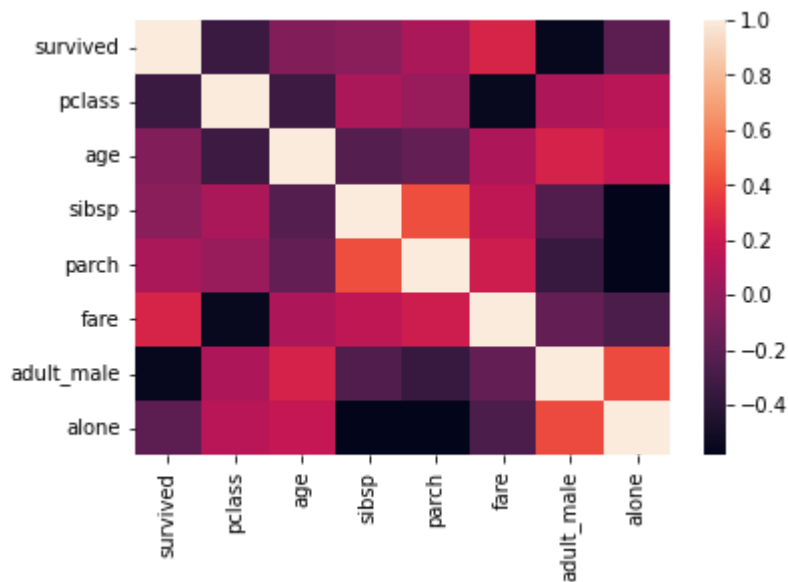
```
In [20]: titanic.corr()
```

```
Out[20]:
```

	survived	pclass	age	sibsp	parch	fare	adult_male	alone
survived	1.000000	-0.338481	-0.069809	-0.035322	0.081629	0.257307	-0.557080	-0.203367

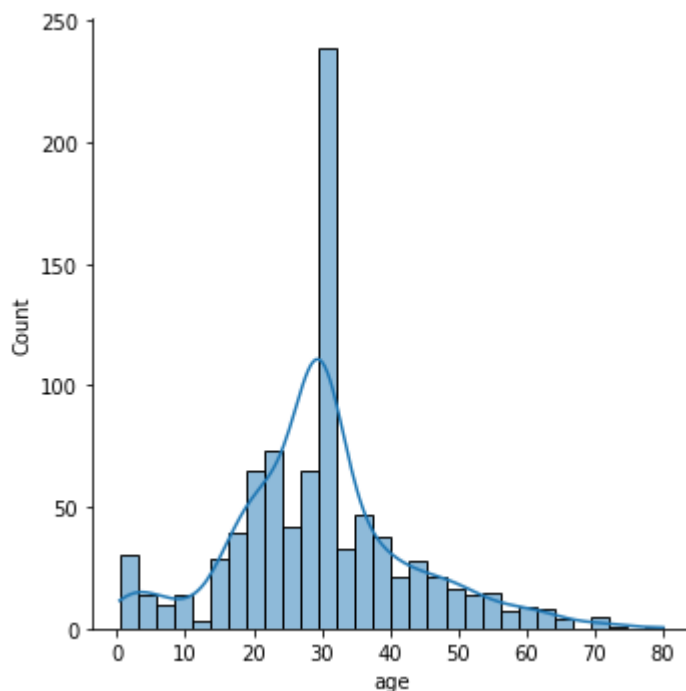
<b>pclass</b>	-0.338481	1.000000	-0.331339	0.083081	0.018443	-0.549500	0.094035	0.135207
<b>age</b>	-0.069809	-0.331339	1.000000	-0.232625	-0.179191	0.091566	0.253236	0.179775
<b>sibsp</b>	-0.035322	0.083081	-0.232625	1.000000	0.414838	0.159651	-0.253586	-0.584471
<b>parch</b>	0.081629	0.018443	-0.179191	0.414838	1.000000	0.216225	-0.349943	-0.583398
<b>fare</b>	0.257307	-0.549500	0.091566	0.159651	0.216225	1.000000	-0.182024	-0.271832
<b>adult_male</b>	-0.557080	0.094035	0.253236	-0.253586	-0.349943	-0.182024	1.000000	0.404744
<b>alone</b>	-0.203367	0.135207	0.179775	-0.584471	-0.583398	-0.271832	0.404744	1.000000

```
In [21]: ax = sns.heatmap(titanic.corr())
```



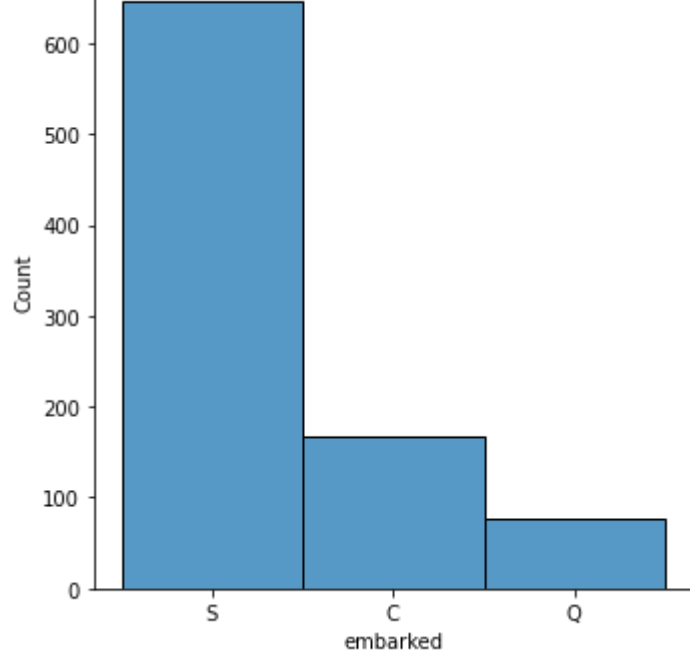
```
In [22]: sns.displot(titanic, x = 'age', kde = True)
```

```
Out[22]: <seaborn.axisgrid.FacetGrid at 0x7f250c0cb490>
```



```
In [23]: sns.displot(titanic, x = 'embarked')
```

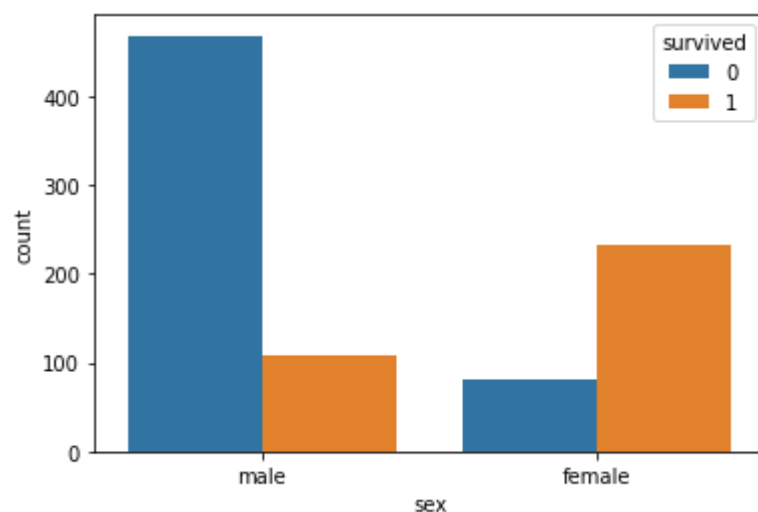
```
Out[23]: <seaborn.axisgrid.FacetGrid at 0x7f250c0da280>
```



In [ ]:

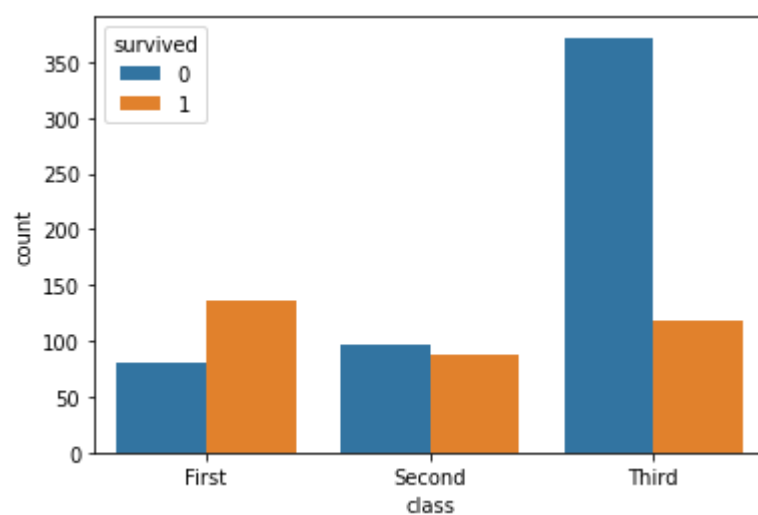
In [24]: `sns.countplot(x='sex', hue='survived', data = titanic)`

Out[24]: `<AxesSubplot:xlabel='sex', ylabel='count'>`



In [25]: `sns.countplot(x='class', hue='survived', data = titanic)`

Out[25]: `<AxesSubplot:xlabel='class', ylabel='count'>`



In [26]:

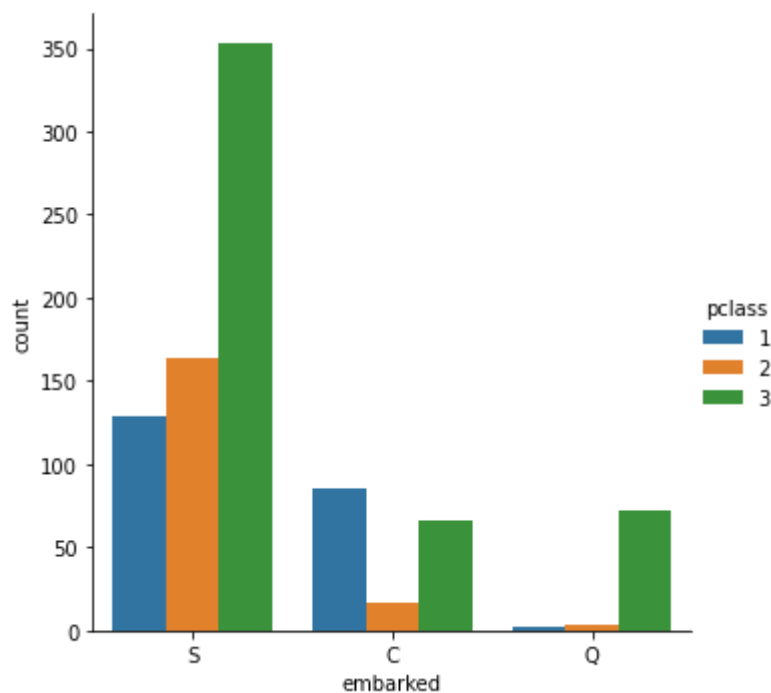
```
In [26]: print('Mean age of passengers who are from Southampton:',
          titanic[titanic['embarked'] == 'S']['age'].mean())
print('Mean age of passengers who are from Cherbourg:',
      titanic[titanic['embarked'] == 'C']['age'].mean())
print('Mean age of passengers who are from Queenstown:',
      titanic[titanic['embarked'] == 'Q']['age'].mean())
```

Mean age of passengers who are from Southampton: 29.544381715534513  
Mean age of passengers who are from Cherbourg: 30.562419467787112  
Mean age of passengers who are from Queenstown: 29.113724216959504

```
In [27]: sns.catplot('embarked',data=titanic,hue='pclass',kind='count')
```

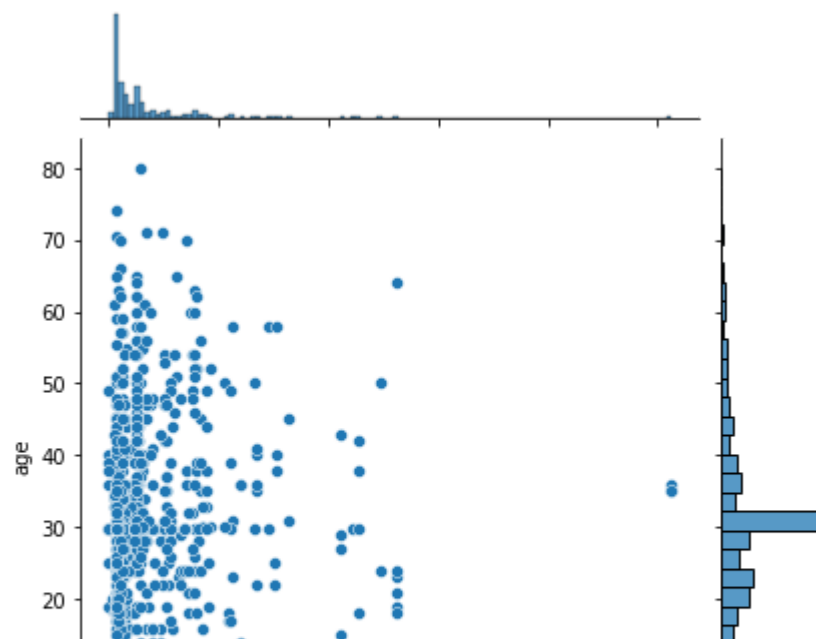
/home/pict/.local/lib/python3.8/site-packages/seaborn/\_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

```
warnings.warn(
Out[27]: <seaborn.axisgrid.FacetGrid at 0x7f2509a11a00>
```

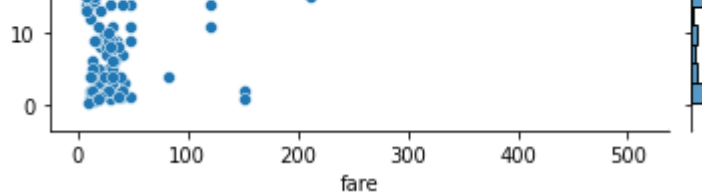


```
In [28]: sns.jointplot(x='fare',y='age',data=titanic)
```

```
Out[28]: <seaborn.axisgrid.JointGrid at 0x7f250c02d070>
```

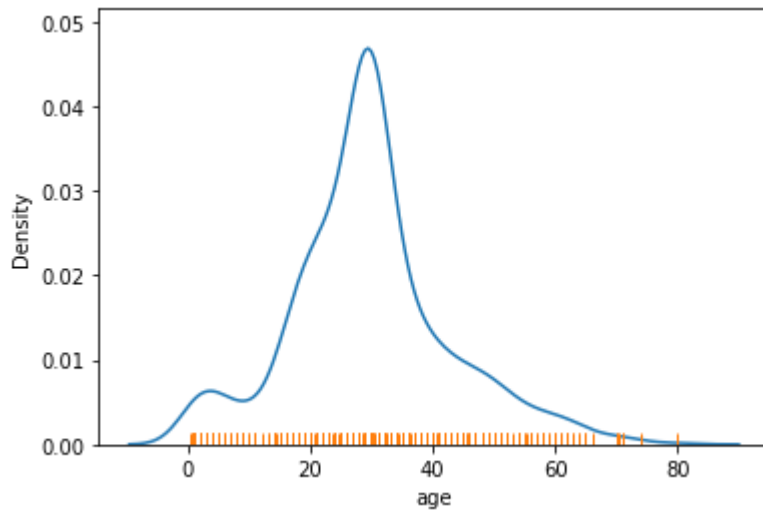






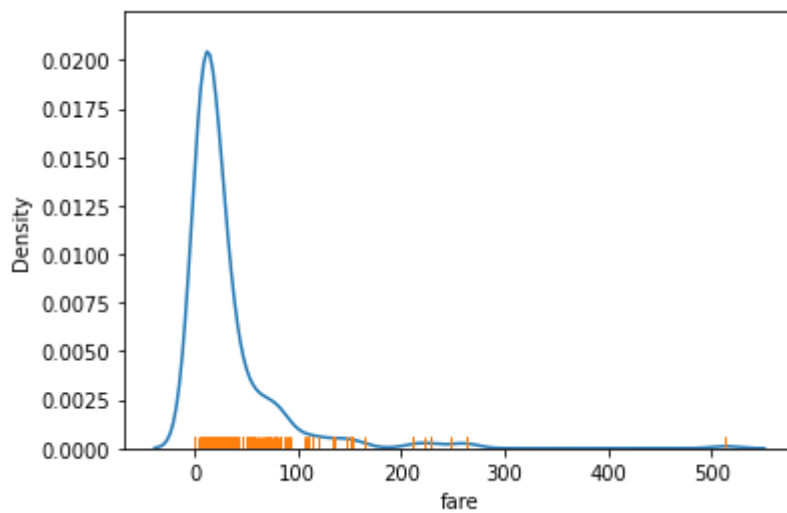
```
In [29]: sns.kdeplot(data = titanic, x = 'age')
sns.rugplot(data = titanic, x = 'age')
```

Out[29]: <AxesSubplot:xlabel='age', ylabel='Density'>



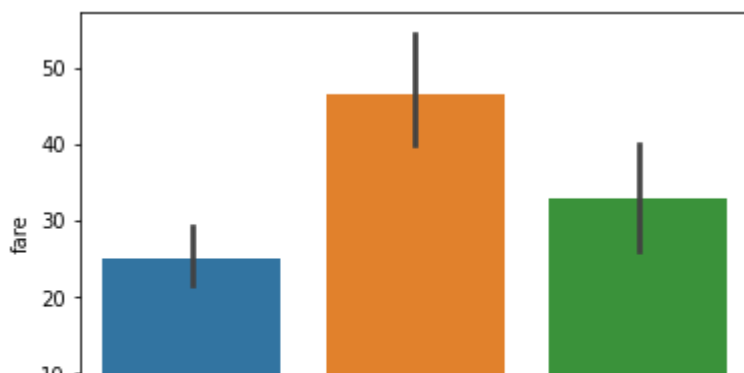
```
In [30]: sns.kdeplot(data = titanic, x = 'fare')
sns.rugplot(data = titanic, x = 'fare')
```

Out[30]: <AxesSubplot:xlabel='fare', ylabel='Density'>



```
In [31]: sns.barplot(data = titanic, y = 'fare', x = 'who')
```

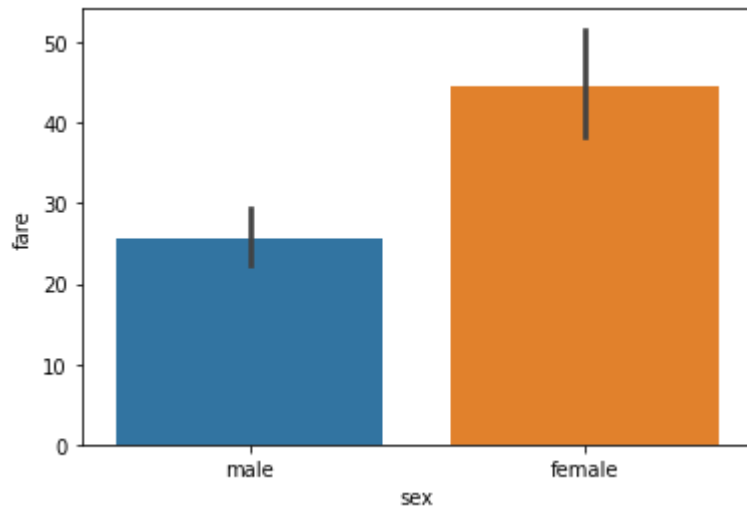
Out[31]: <AxesSubplot:xlabel='who', ylabel='fare'>





```
In [32]: sns.barplot(data = titanic, y = 'fare', x = 'sex')
```

```
Out[32]: <AxesSubplot:xlabel='sex', ylabel='fare'>
```



```
In [33]: sns.barplot(data = titanic, y = 'fare', x = 'alone')
```

```
Out[33]: <AxesSubplot:xlabel='alone', ylabel='fare'>
```

