data_visualization

January 12, 2023

1 *Day-11*

1.1 Data Visualization

Author: Kaleem Ullah
Date: 2023-01-11
Email: kaleemrao417@outlook.com

1.2 Contents

1.3 Part-1

1.3.1 Most important elements for Data-Visualization

1. Data-type: (Categorical, Numerical, Time-series, Text, Image, Audio, Video, etc.)
2. How to Plot the data (Purpose): (scatter, line, bar, box, hist, scatter + line, pair, etc.)

- **Q. What is a plot/graph and how it formed?**
- **Ans:**
  - A plot is a graphical representation of data.
  - It is formed by plotting points on a graph.
  - A graph has 2 axis, x-axis and y-axis.
  - Each point on the graph is plotted by taking the x-coordinate and y-coordinate of the point.

* The x-axis is the horizontal axis and y-axis is the vertical axis.
* The x-axis is the independent variable and y-axis is the dependent variable.
* The x-axis is the input and y-axis is the output.
* Exmple:

```
If x = 3
   y = 4
```

Then the point (3,4) is plotted on the graph.
The points are connected by a line to form a graph.
The graph is plotted by taking the x-coordinates and y-coordinates of the points.

- **Example-1**

[ ]:
```
import pandas as pd
import matplotlib.pyplot as plt
```

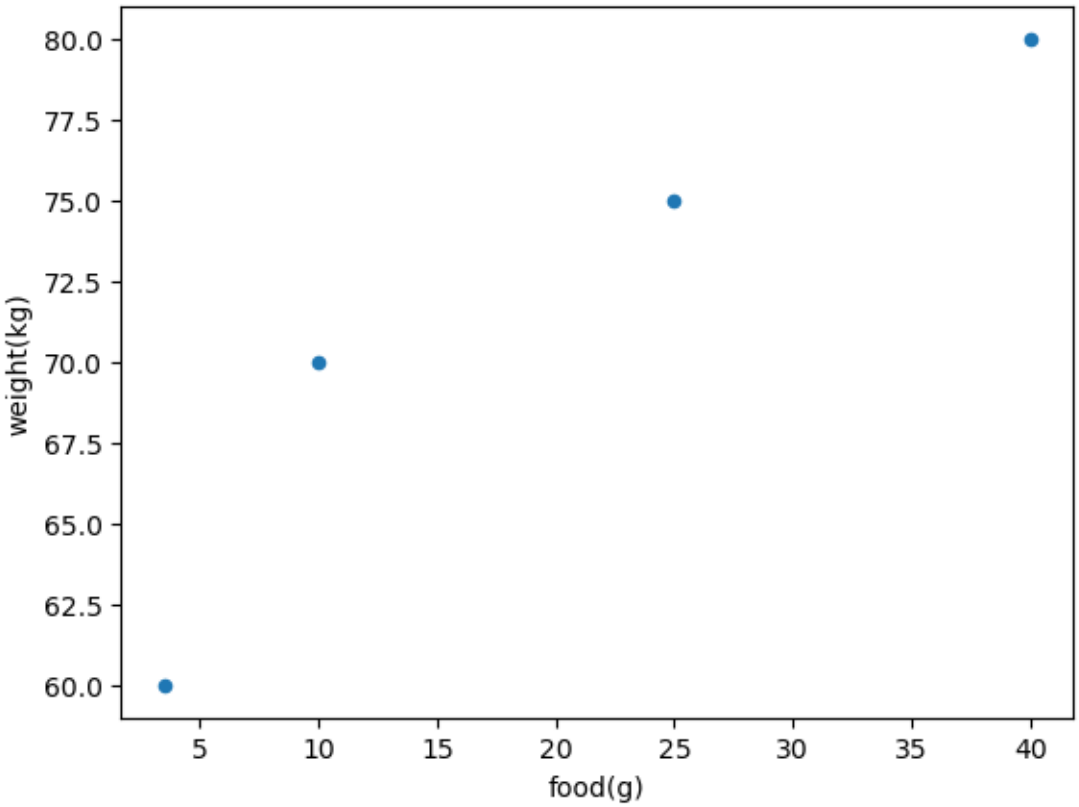
[ ]:
```
df = pd.DataFrame({'food(g)':[3.5, 10, 25, 40] , 'weight(kg)':[60, 70, 75, 80]})
df
```

[ ]:

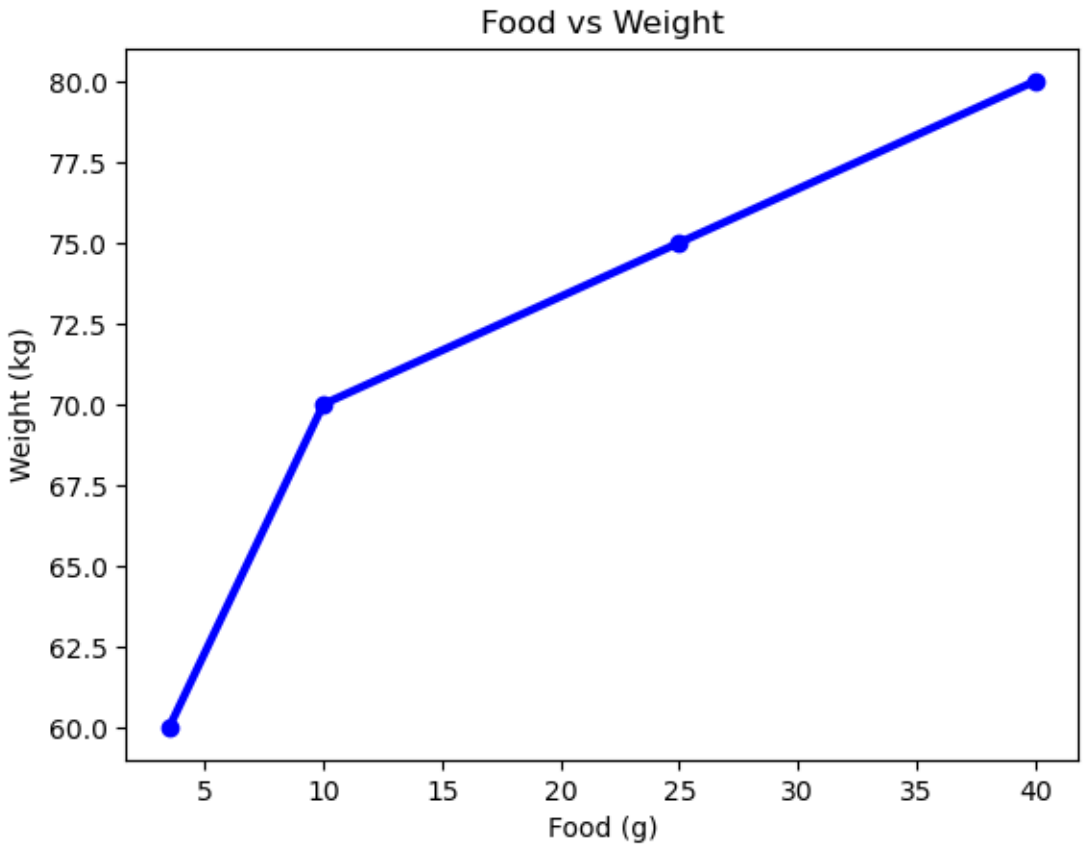
	food(g)	weight(kg)
0	3.5	60
1	10.0	70
2	25.0	75
3	40.0	80

[ ]:
```
df.plot.scatter(x='food(g)', y='weight(kg)')
```

[ ]:
```
<AxesSubplot:xlabel='food(g)', ylabel='weight(kg)'>
```



[ ]:
```
x = df['food(g)'].values
y = df['weight(kg)'].values
plt.plot(x, y, '-o', color='blue', linewidth=3)
plt.xlabel('Food (g)')
plt.ylabel('Weight (kg)')
plt.title('Food vs Weight')
plt.show()
```



- **Explaination:**
  - X-axis has tick marks and values on it (These values are called scale of X-axis) and same of Y-axis.
  - Every point on the graph is plotted by taking the x-coordinate and y-coordinate of the point.
  - First point is plotted at (3.5, 60), second point is plotted at (10, 70), third point is plotted at (25, 75) and the last point is plotted at (40, 80).

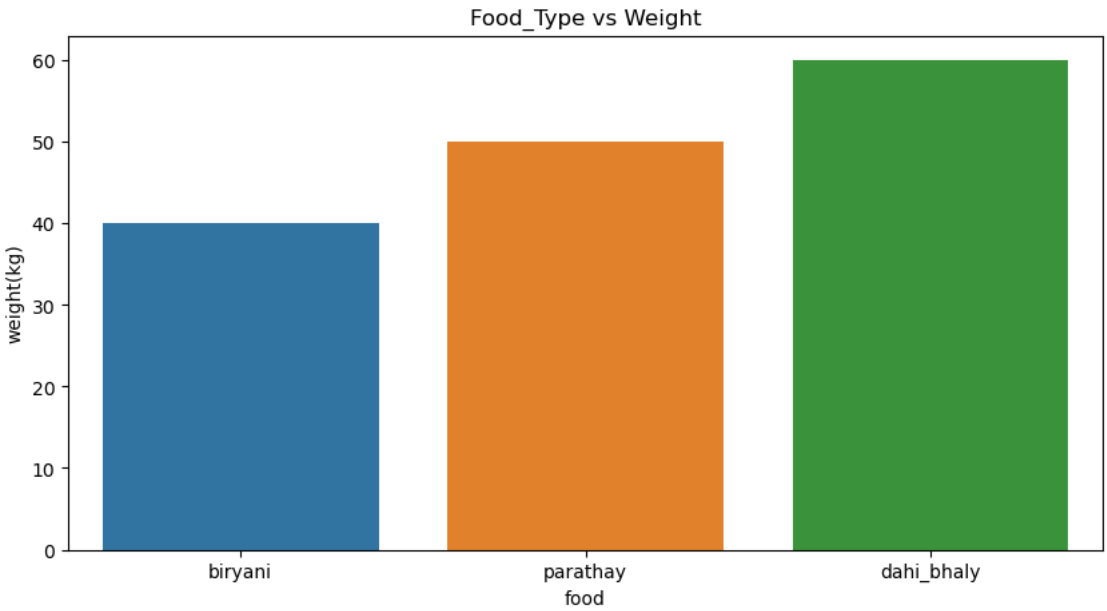
- **Example-2**

[ ]:
```
df1 = pd.DataFrame({'food':['biryani' , 'parathay' , 'dahi_bhaly'] ,
↪'weight(kg)':[40, 50, 60]})
df1
```

[ ]:

	food	weight(kg)
0	biryani	40
1	parathay	50
2	dahi_bhaly	60

[ ]:
```
import seaborn as sns
plt.figure(figsize=(10, 5))
sns.barplot(x='food', y='weight(kg)', data=df1)
plt.title('Food_Type vs Weight')
plt.show()
```



1.3.2 Important Elements of a plot:

1. **Data-Points**
2. **Canvas**
3. **Axis**
4. **Text**
5. **Information**

1.4 Part-2

1.4.1 Plotting in Python

Importing the libraries

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

Import dataset

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

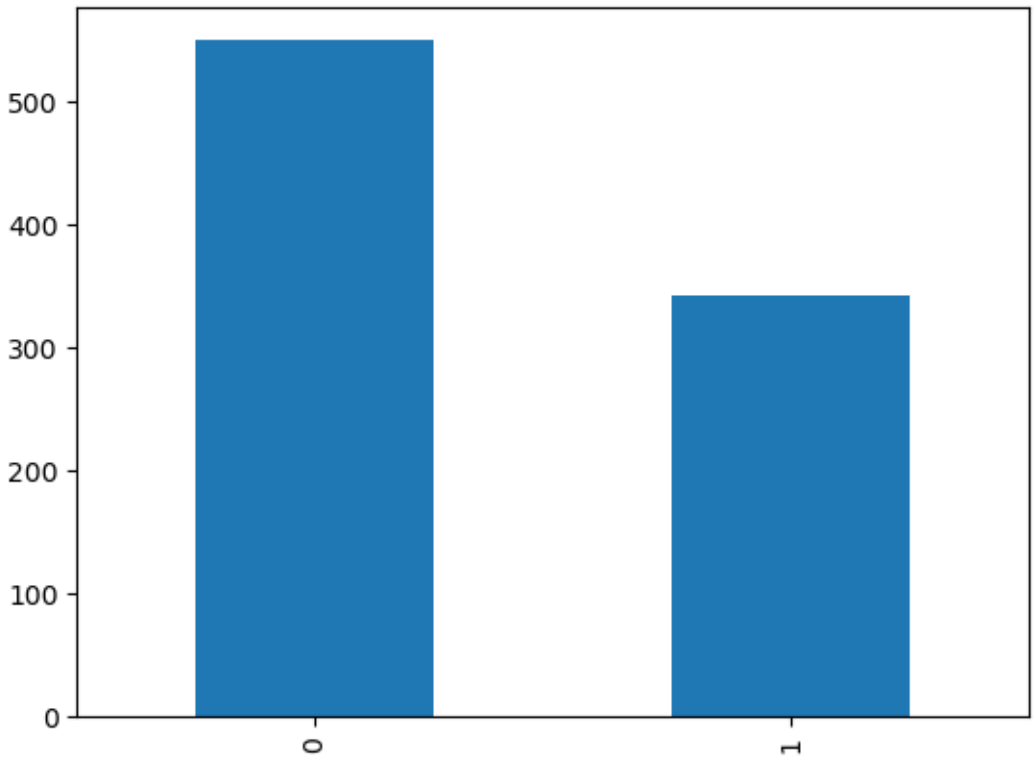
[ ]:
```
kashti.isnull().sum()
```

[ ]:
```
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
```

Bar-Plot

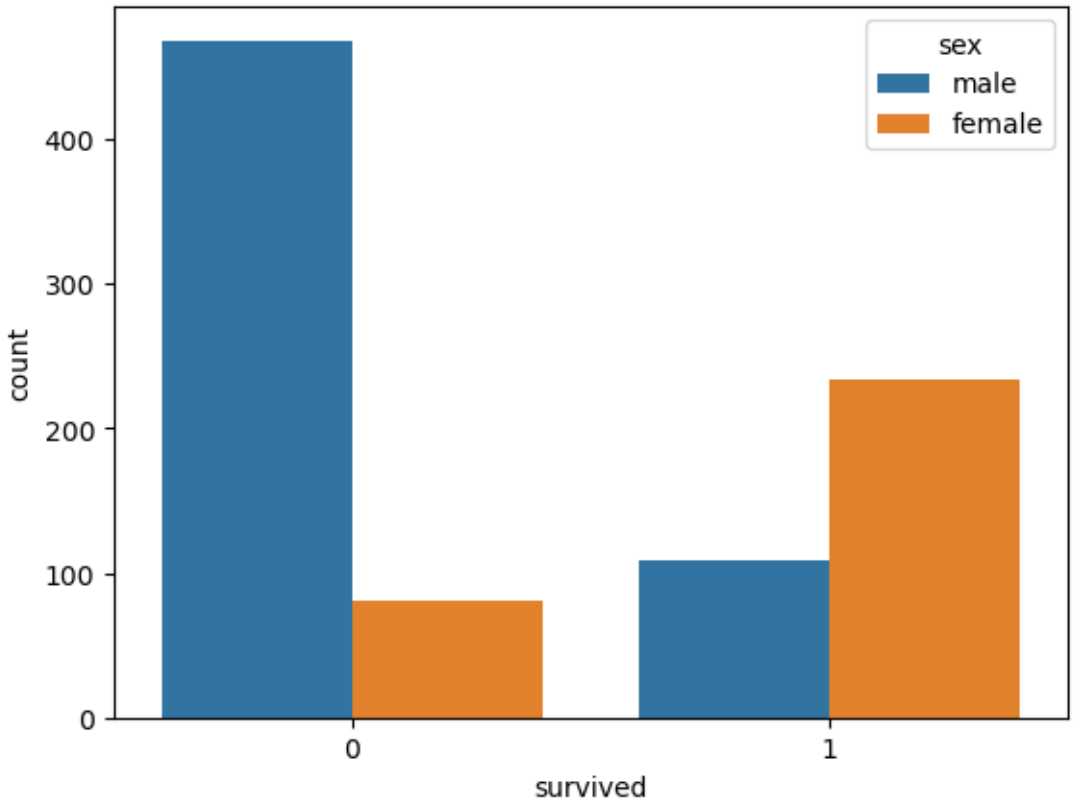
[ ]:
```
kashti.survived.value_counts().plot(kind='bar')
```

[ ]: <AxesSubplot:>



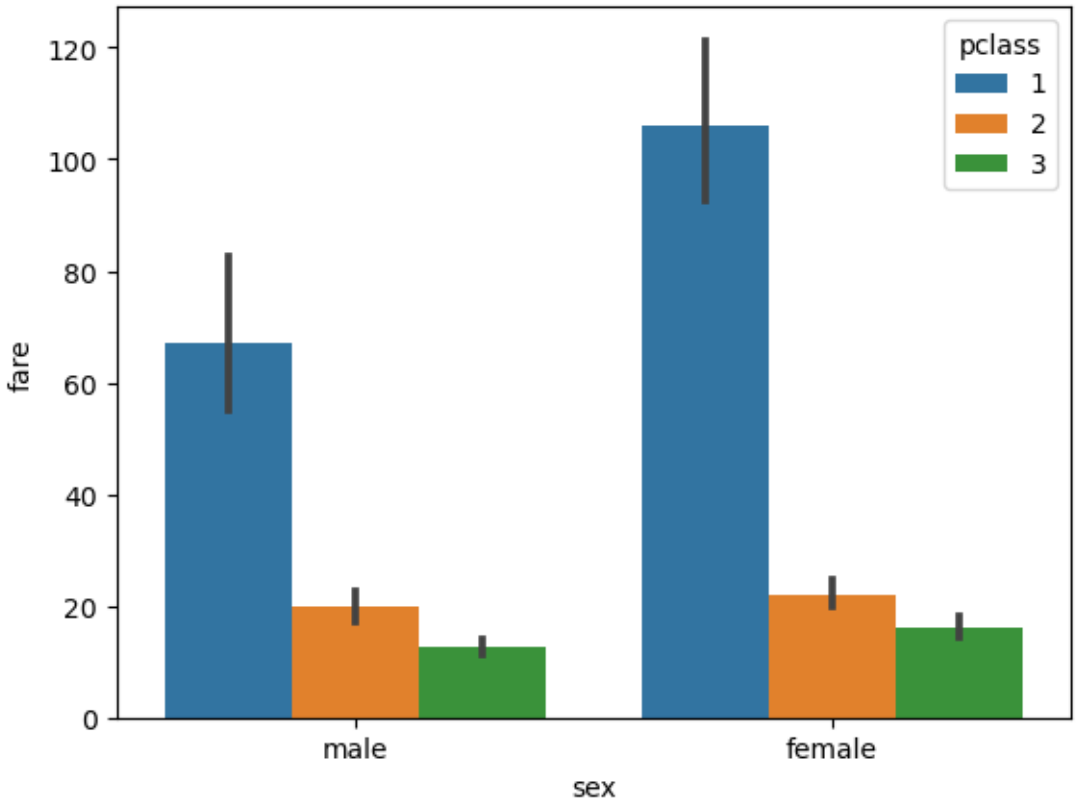
[ ]:
```
sns.countplot(x='survived', hue='sex', data=kashti)
```

[ ]: <AxesSubplot:xlabel='survived', ylabel='count'>



[ ]:
```
sns.barplot(x='sex', y='fare', hue='pclass', data=kashti)
```

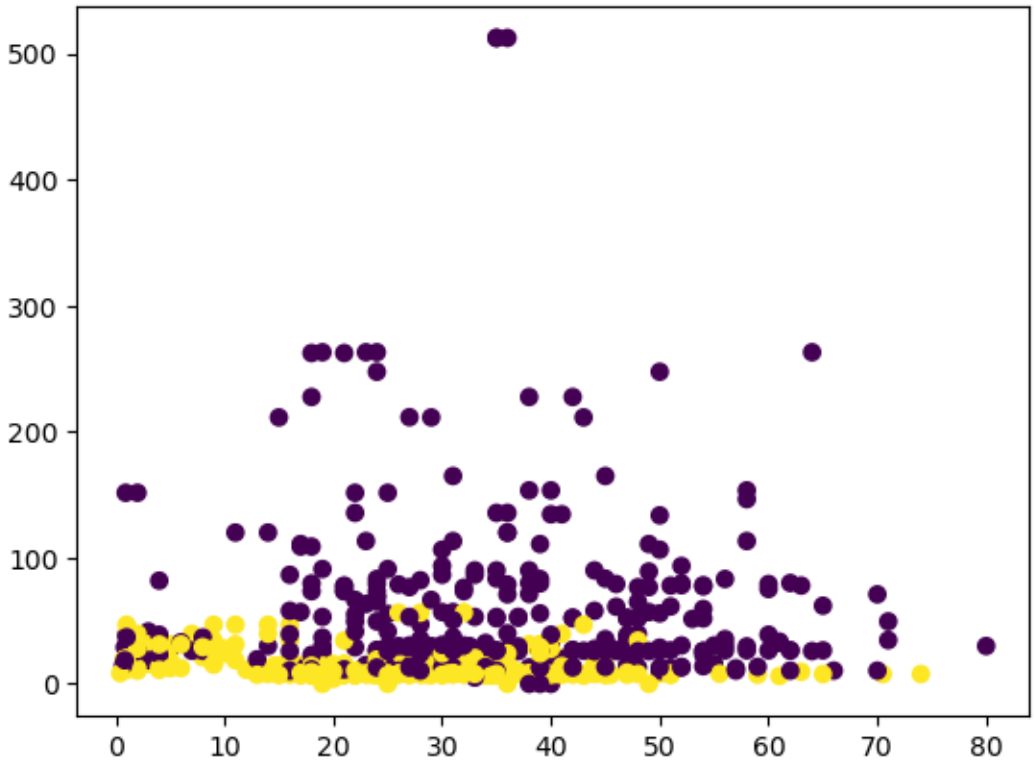
[ ]:
```
<AxesSubplot:xlabel='sex', ylabel='fare'>
```



Scatter-Plot

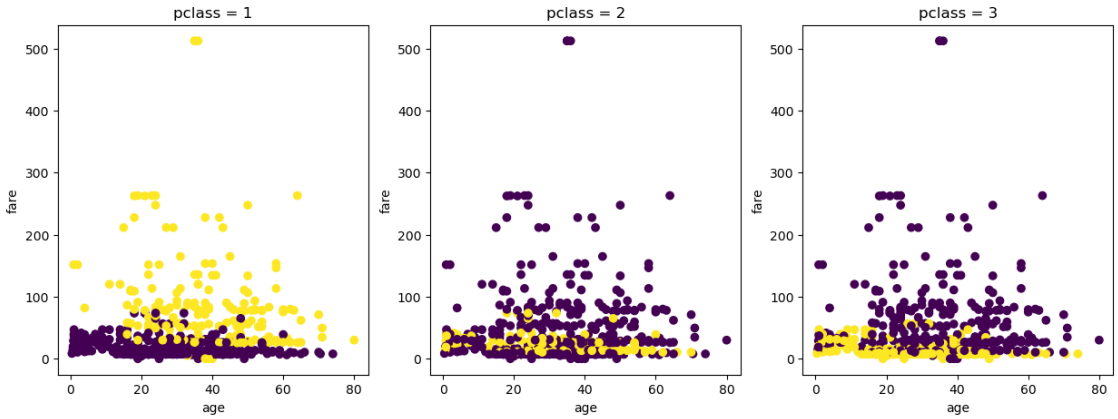
```
[ ]: X=kashti[['age', 'fare']]
     Y=kashti['pclass'] == 3                    # This can be changed to 2 or 3(unique␣
      ↪values)
     plt.scatter(X['age'], X['fare'], c=Y)
                                                # yellow = 1, blue = 0 (as per the␣
      ↪above line)
```

```
[ ]: <matplotlib.collections.PathCollection at 0x294b100d280>
```



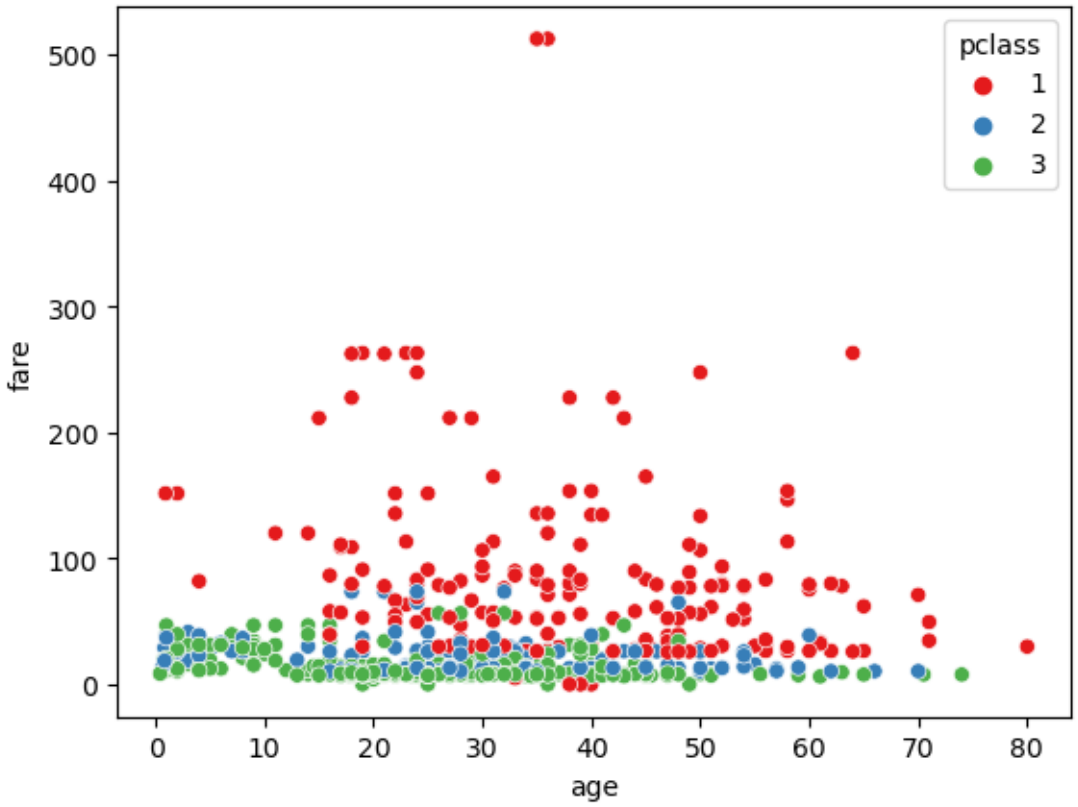
[ ]:
```
fig, ax = plt.subplots(1,3, figsize=(15,5))
class_list=[1,2,3]

for i in range(3):
    Y=kashti['pclass'] == class_list[i]
    ax[i].scatter(X['age'], X['fare'], c=Y)
    ax[i].set_title('pclass = ' + str(class_list[i]))
    ax[i].set_xlabel('age')
    ax[i].set_ylabel('fare')
plt.show()
```



[ ]:
```
sns.scatterplot(x='age', y='fare', hue='pclass', data=kashti, palette='Set1')
```

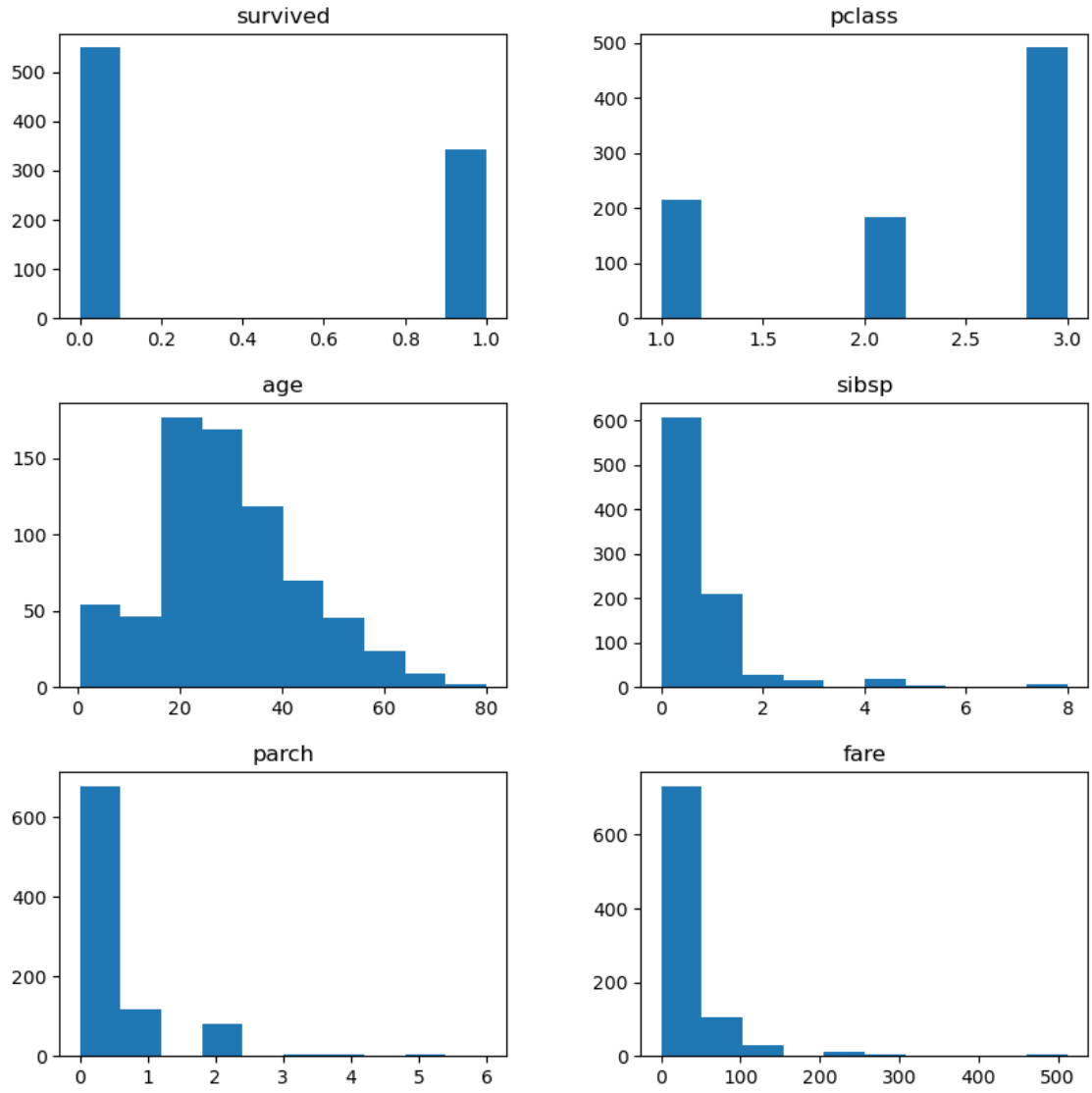
[ ]: <AxesSubplot:xlabel='age', ylabel='fare'>



Hist-Plot

[ ]:
```
kashti.hist(figsize=(10,10), grid=False)
```

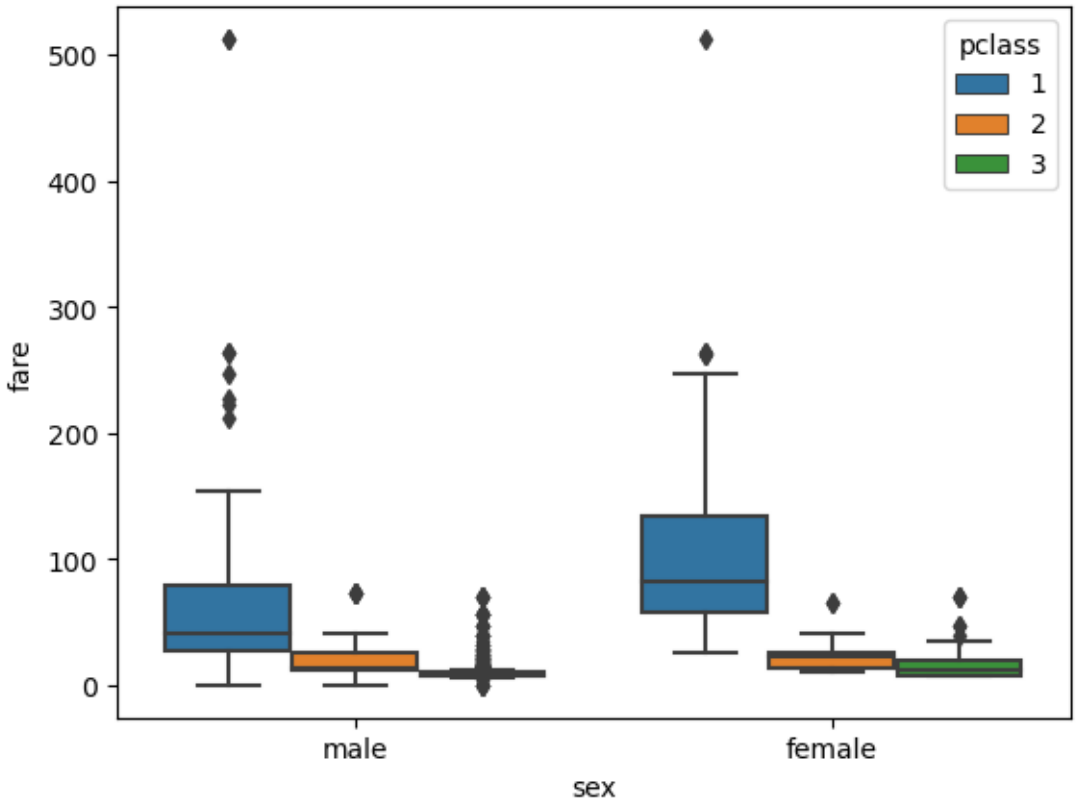
[ ]:
```
array([[<AxesSubplot:title={'center':'survived'}>,
        <AxesSubplot:title={'center':'pclass'}>],
       [<AxesSubplot:title={'center':'age'}>,
        <AxesSubplot:title={'center':'sibsp'}>],
       [<AxesSubplot:title={'center':'parch'}>,
        <AxesSubplot:title={'center':'fare'}>]], dtype=object)
```



Box-Plot

[ ]:
```
sns.boxplot(x='sex', y='fare', hue='pclass', data=kashti)
```

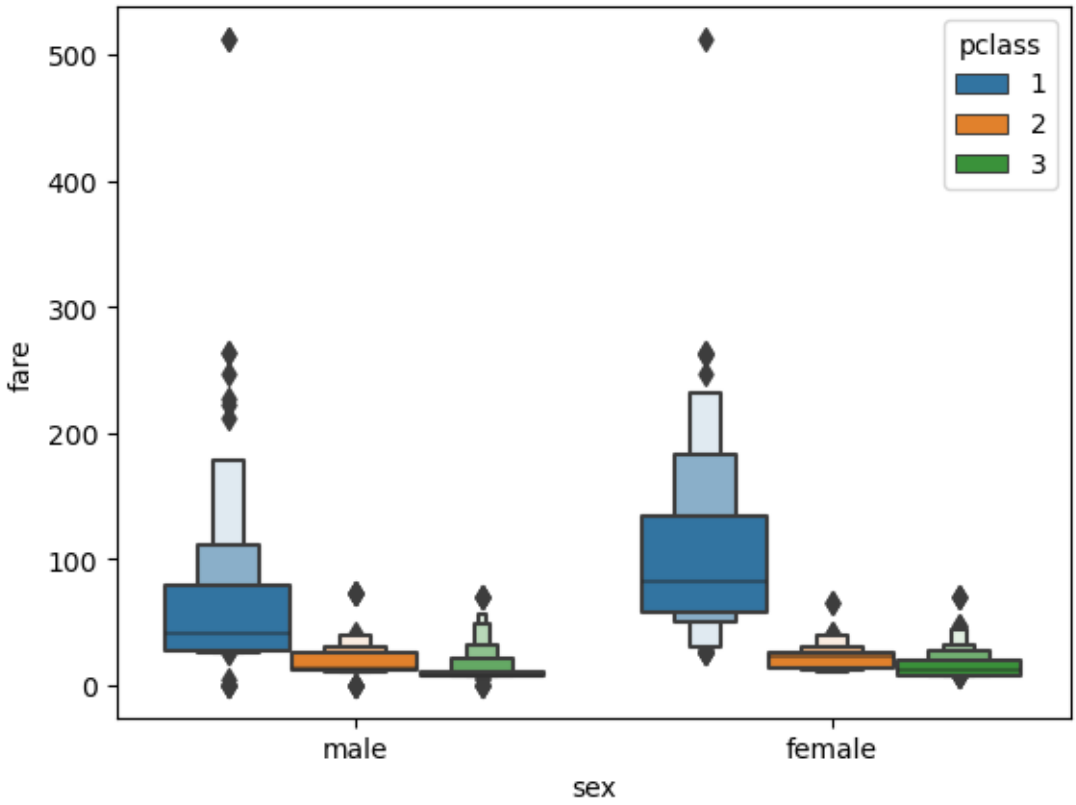
[ ]:
```
<AxesSubplot:xlabel='sex', ylabel='fare'>
```



Boxen-Plot

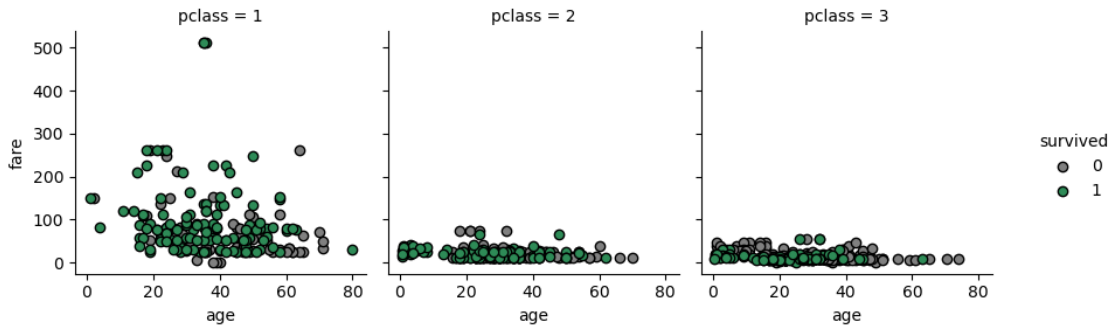
[ ]:
```
sns.boxenplot(x='sex', y='fare', hue='pclass', data=kashti)
```

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



Sub-Plots

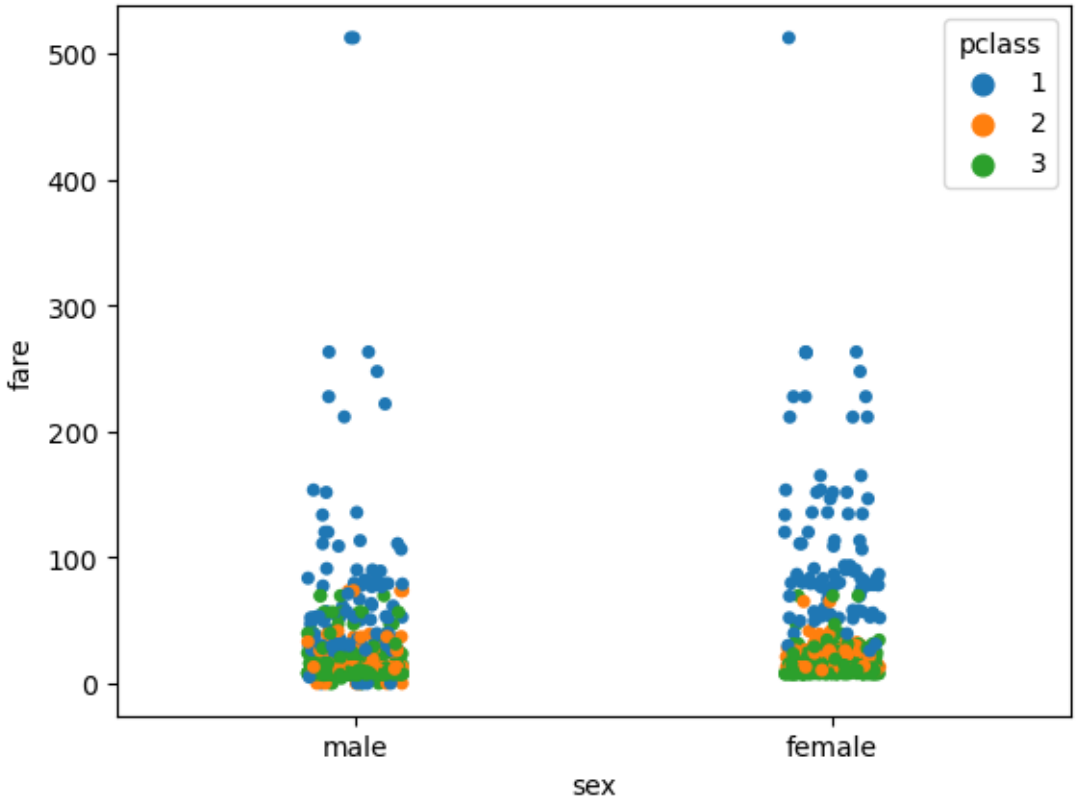
```
[ ]: g=sns.FacetGrid(kashti, hue = 'survived', col='pclass', margin_titles=True,
                     palette = {1:'seagreen', 0:'gray'})
     g=g.map(plt.scatter, 'age', 'fare', edgecolor='k').add_legend()
```



Strip_Plot

```
[ ]: sns.stripplot(x='sex', y='fare', hue='pclass', data=kashti)
```

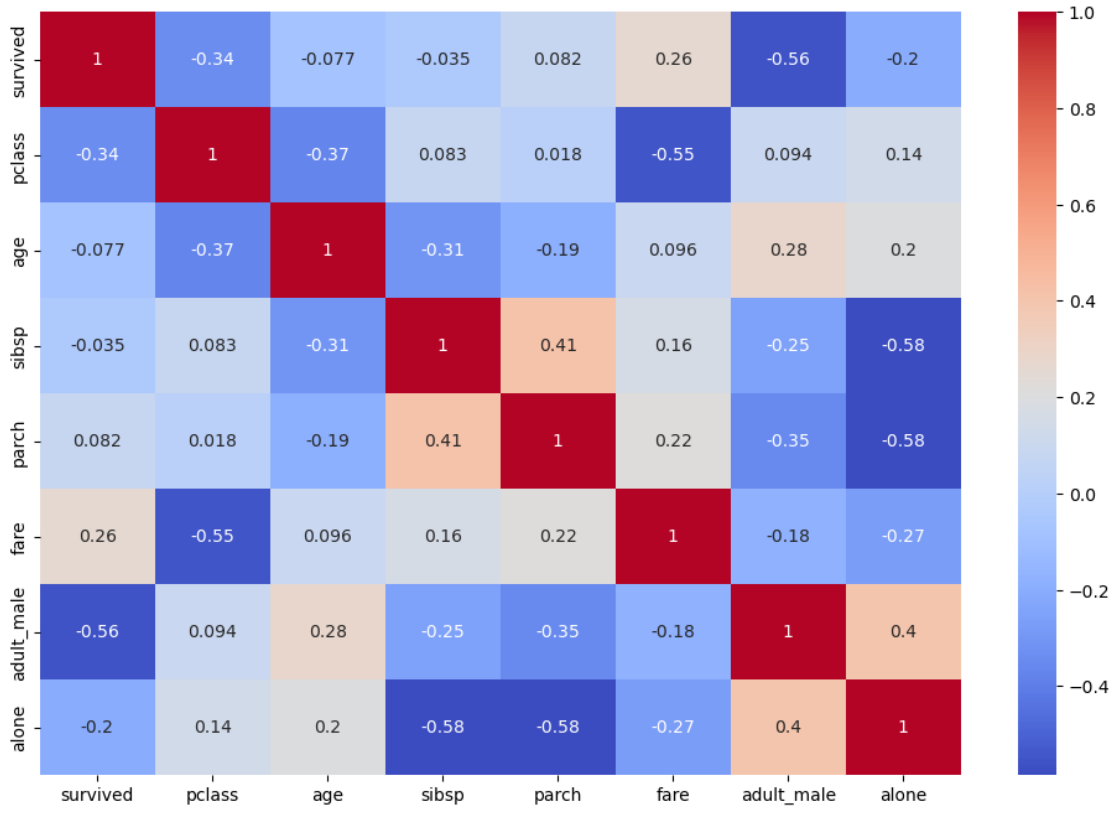
[ ]: <AxesSubplot:xlabel='sex', ylabel='fare'>



Heatmap

[ ]:
```
corr = kashti.corr()
plt.figure(figsize=(12,8))
sns.heatmap(corr, annot=True, cmap='coolwarm')
```

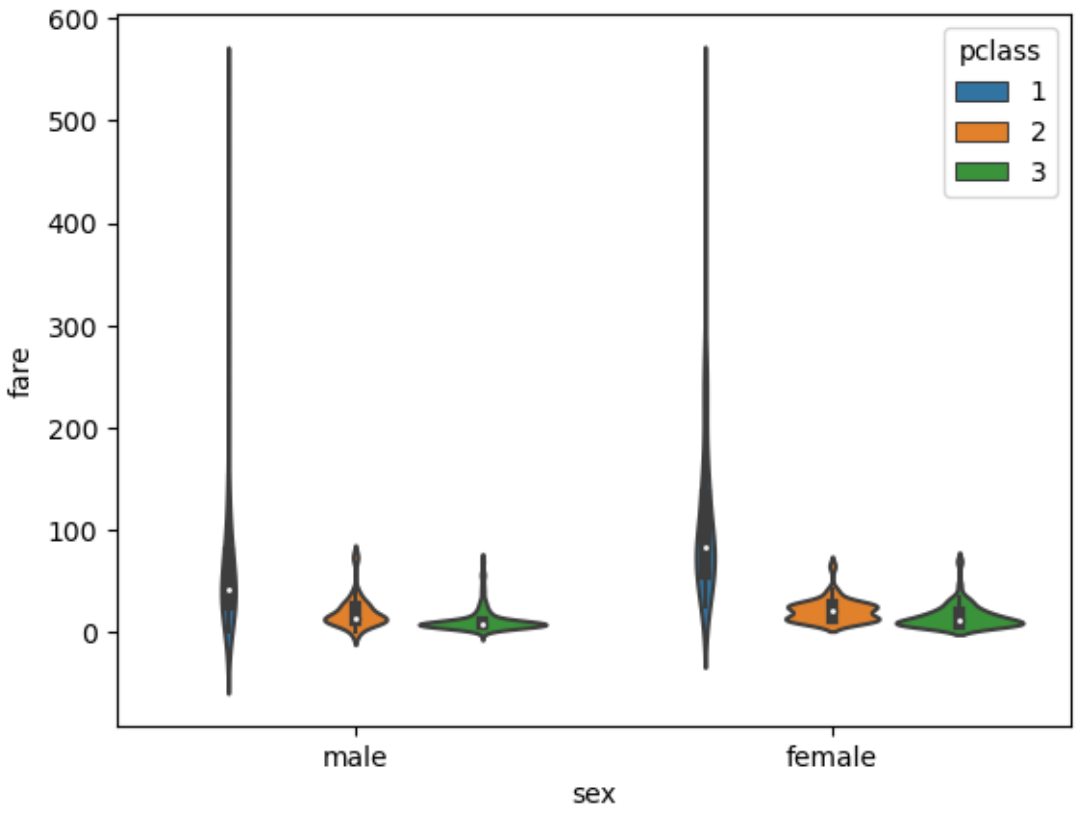
[ ]: <AxesSubplot:>



Violin-Plot

[ ]:
```
sns.violinplot(x='sex', y='fare', hue='pclass', data=kashti)
```

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


600
500
400
300
200
100
0
fare
pclass
1
2
3
male
female
sex