2024년도 1학기
인공지능
Artificial Intelligence

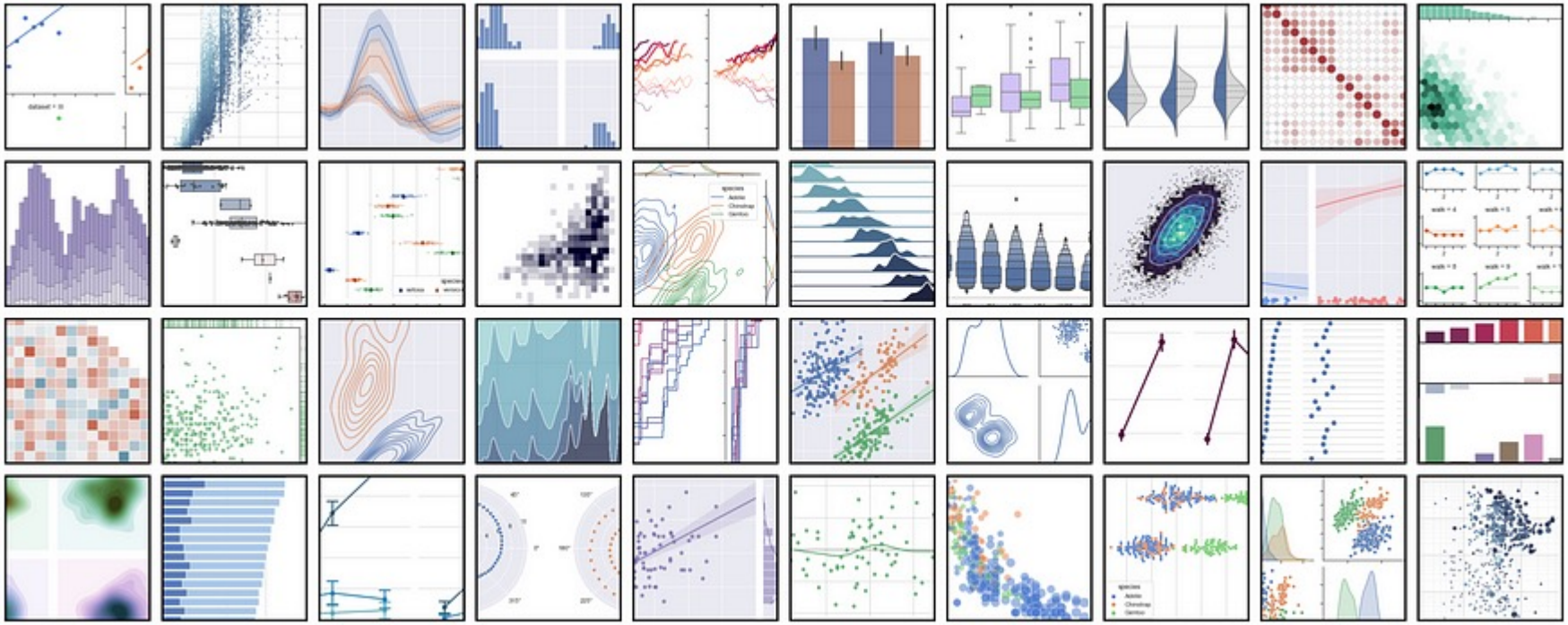
Day
06

데이터시각화 - Seaborn

CONTENTS

Seaborn는 무엇인가?

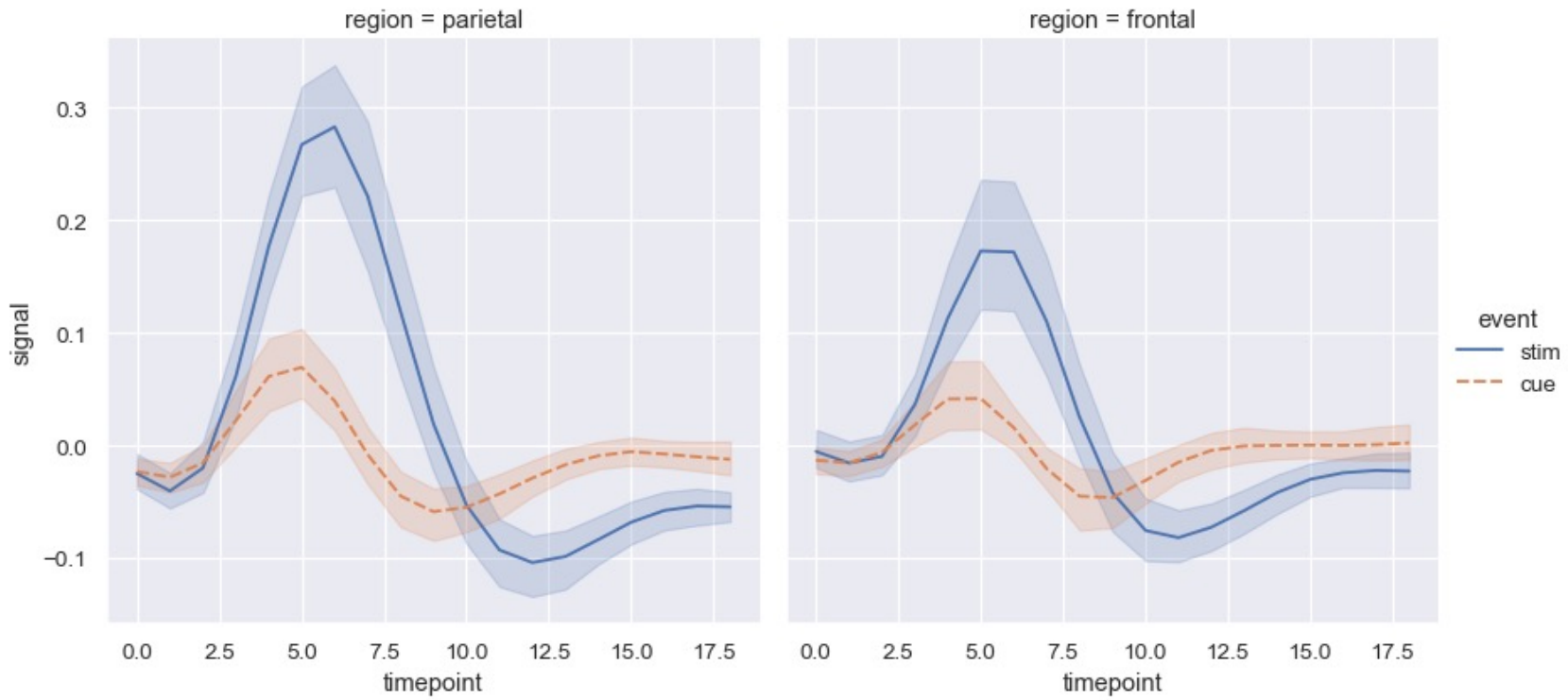
- Seaborn is a Python data visualization library based on matplotlib
- It provides a high-level interface for drawing attractive and info rmative statistical graphics



Seaborn Features

❖ Statistical Estimation

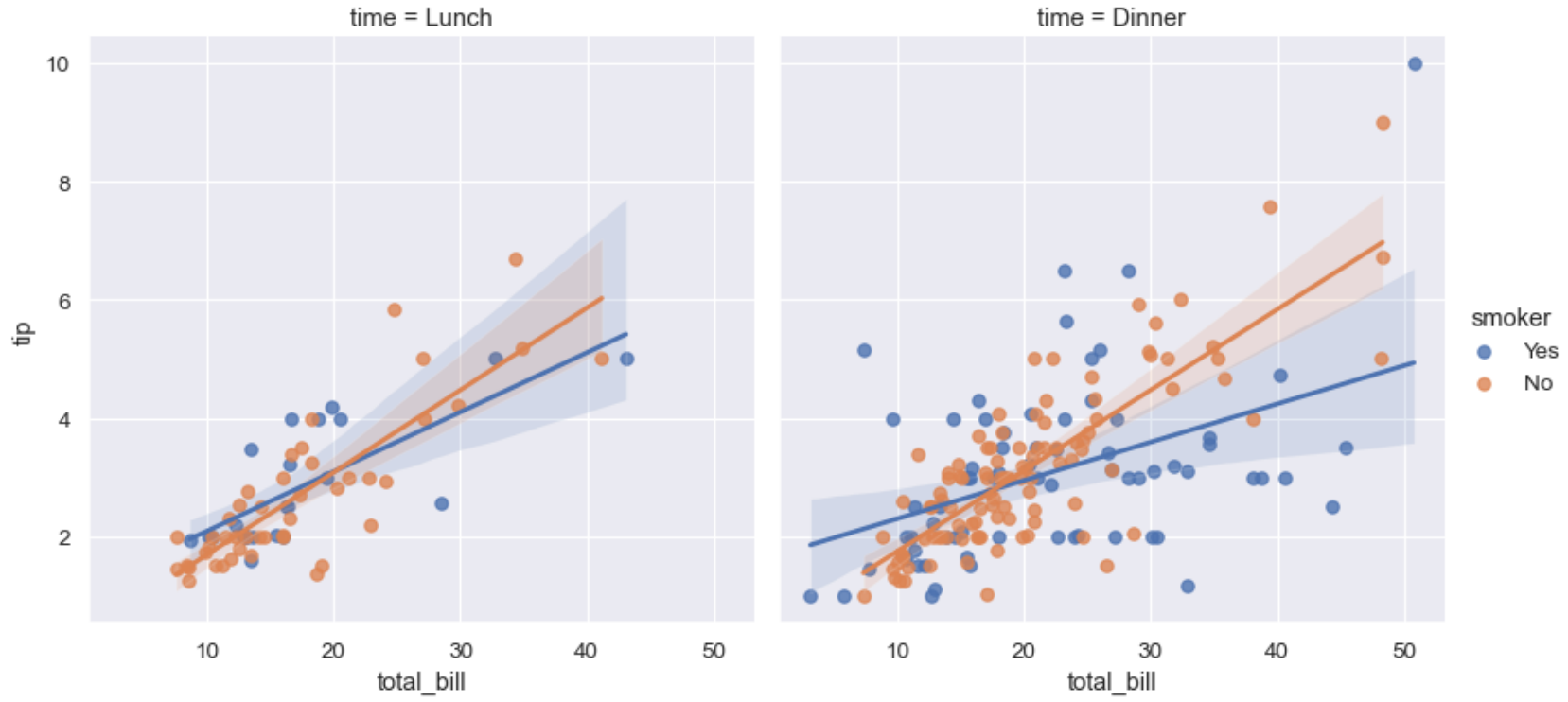
- Average of values



Seaborn Features

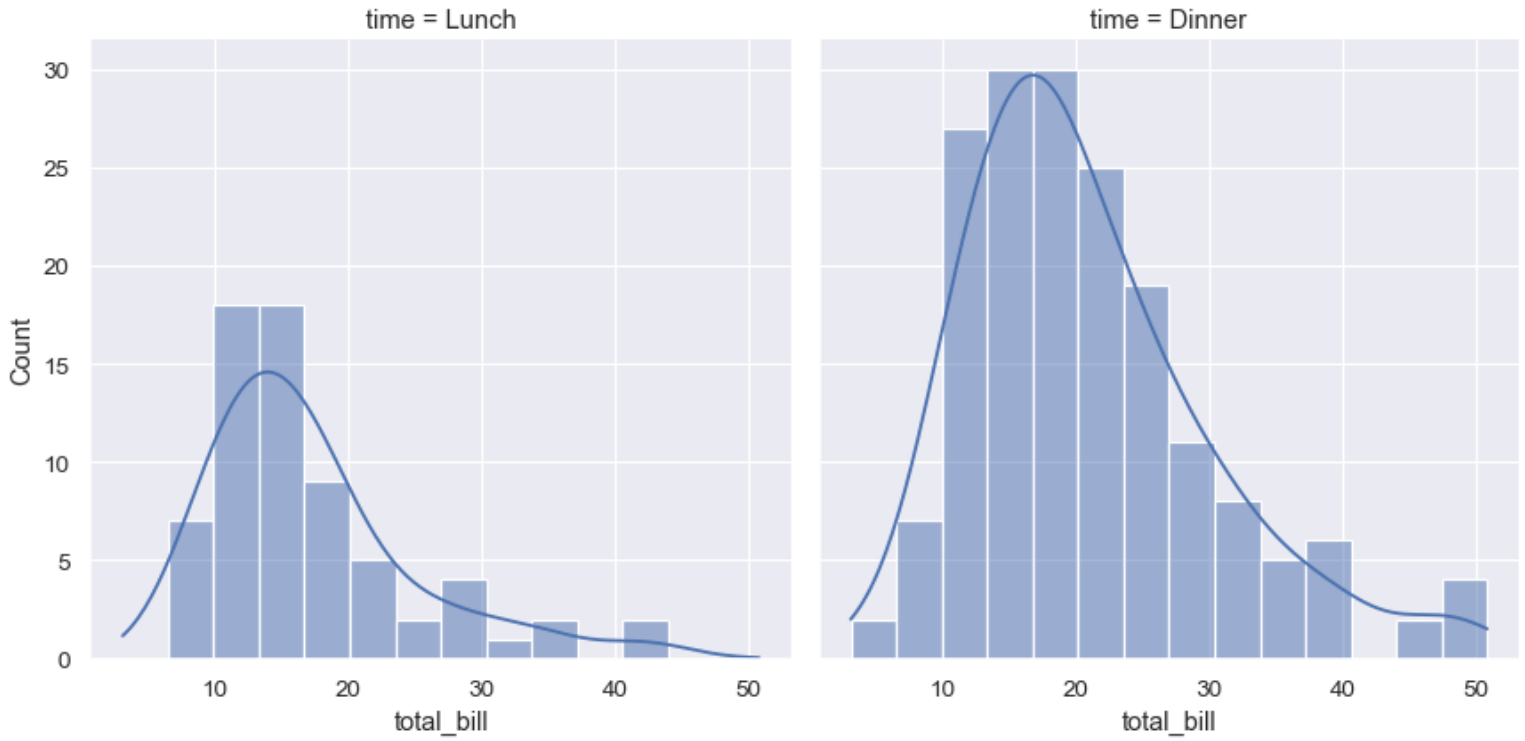
❖ Statistical Estimation

- Linear Regression Model



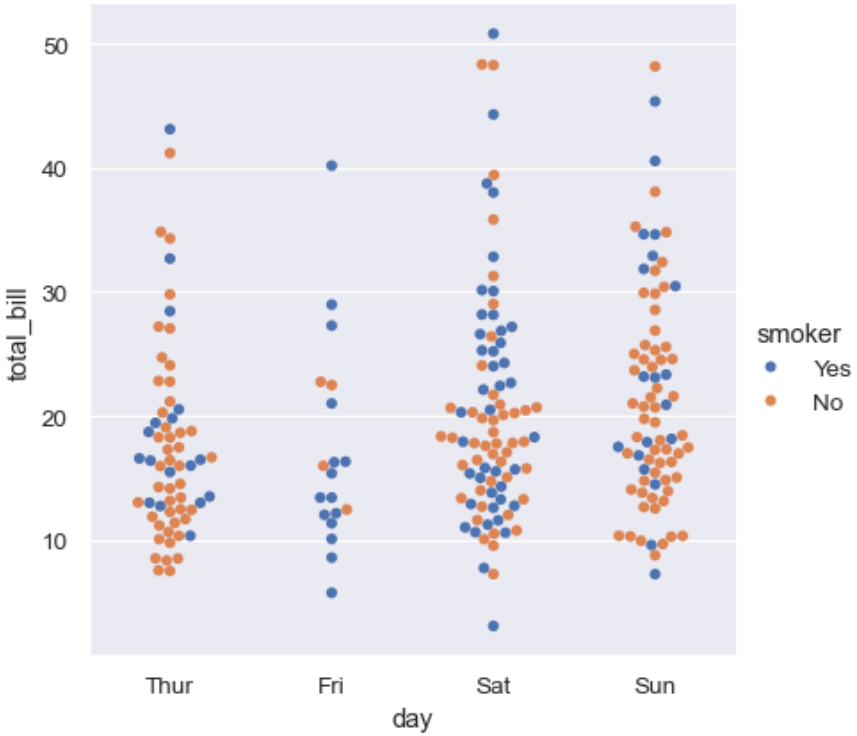
Seaborn Features

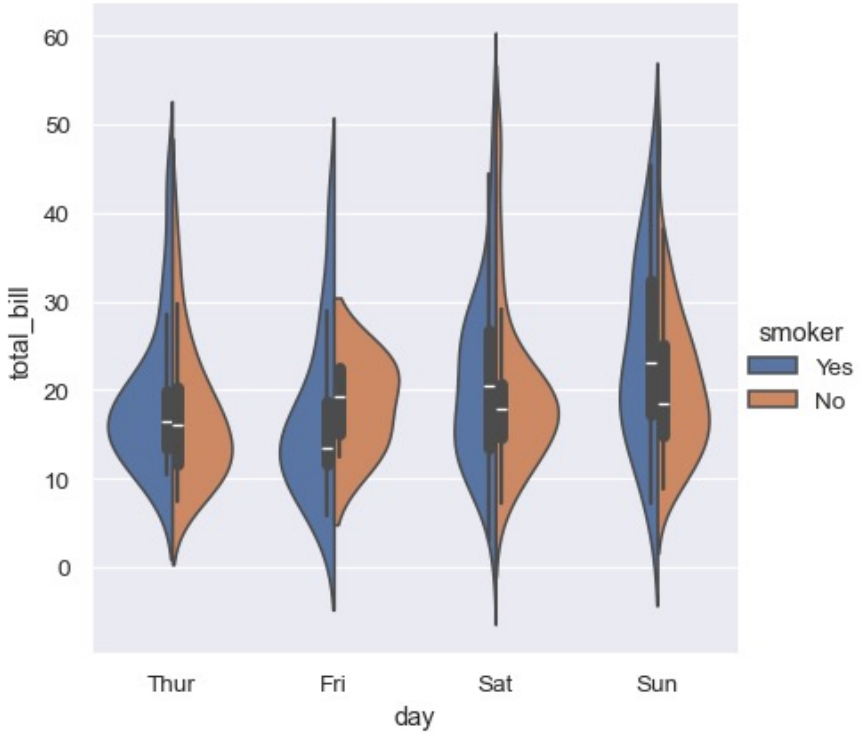
- **Distributional representations**
  - Histograms or kernel density estimation



Seaborn Features

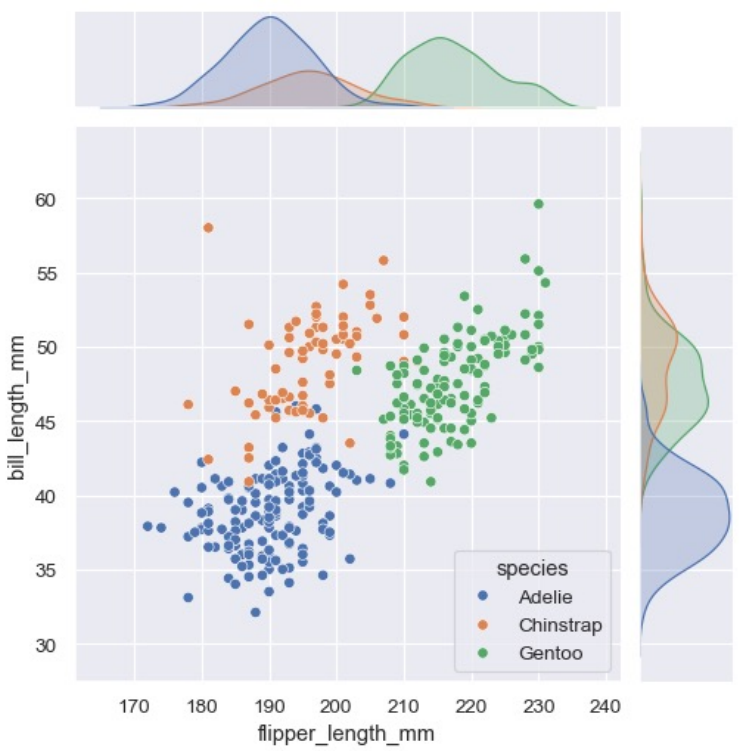
- **Plots for categorical data**
  - Swarm plot, kernel density estimation for underlying distribution

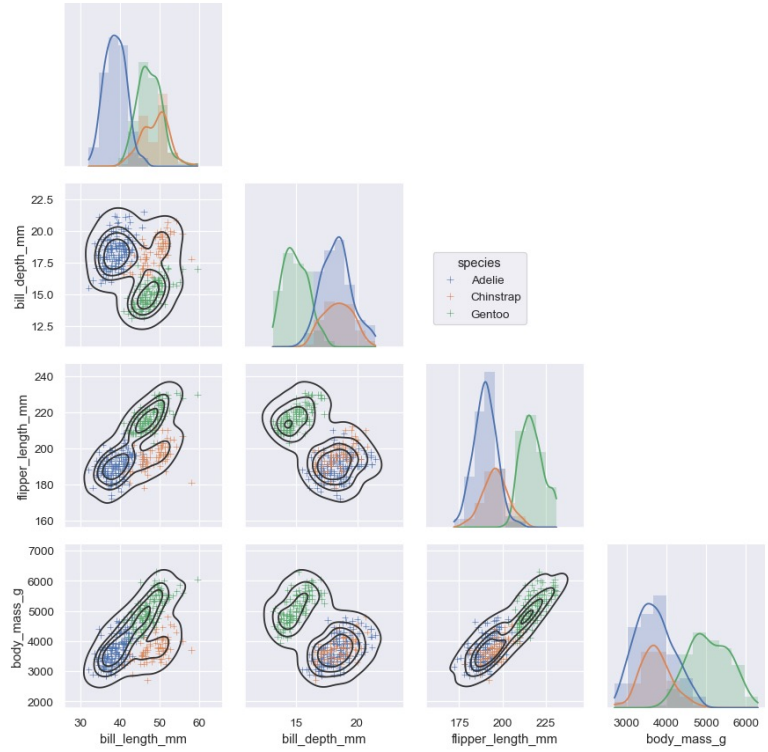




Seaborn Features

- **Multivariate views on complex datasets**
  - Easy coding for complex graphs





Plotly 설치

❖ Jupyter Notebook을 실행합니다.

❖ 명령 프롬프트(CMD)를 실행하고 다음 명령어를 사용합니다.

- pip install seaborn

❖ 다른 라이브러리 또한 설치해야 합니다.

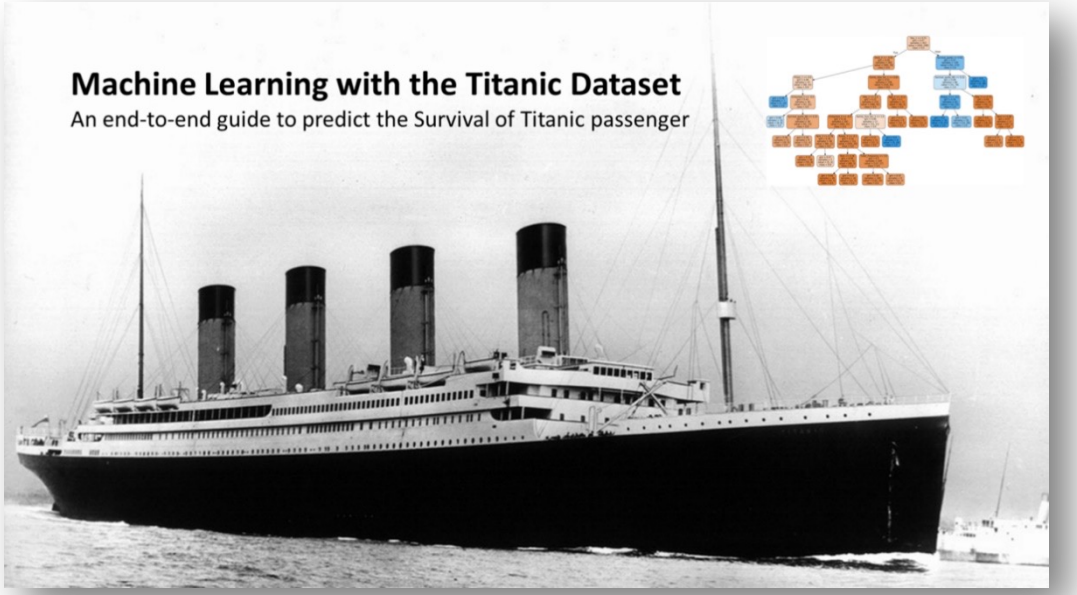
- Numpy, scipy, matplotly, pandas

❖ plotly 라이브러리 버전을 확인합니다.

```
import seaborn                    //Importing seaborn library
print(seaborn.__version__)        //Printing library version
```

Data

❖ Dataset



- titanic.csv

```
import pandas as pd

titanic_df = pd.read_csv("D:/titanic.csv")
```

Data

- **info()**
  - Provides essential details about your dataset
    - titanic_df.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 895 entries, 0 to 894
Data columns (total 12 columns):
PassengerId    895 non-null int64
Survived       895 non-null int64
Pclass         895 non-null int64
Name           895 non-null object
sex          895 non-null object
Age            718 non-null float64
SibSp          895 non-null int64
Parch          895 non-null int64
Ticket         895 non-null object
Fare           884 non-null float64
Cabin          204 non-null object
Embarked       893 non-null object
dtypes: float64(2), int64(5), object(5)
memory usage: 66.5+ KB
```

C

Seaborn Charts

Count Plot

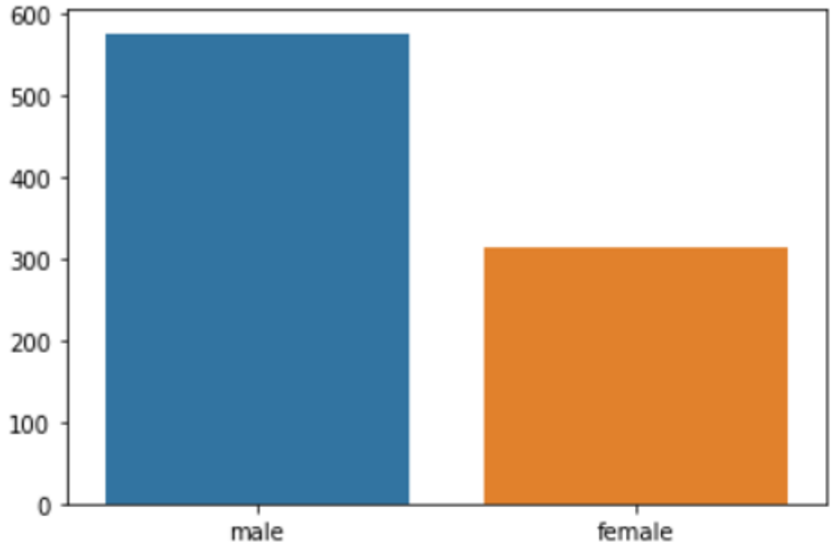
- **countplot()**
  - Show the counts of observations in each category using bars
    - data – dataset for plotting
    - x, y – inputs for plotting data

```
#importing library
import seaborn as sns
import matplotlib.pyplot as plt

#importing dataset
titanic = sns.load_dataset("titanic")

#plotting chart
sns.countplot(x='sex', data=titanic)
plt.show()
```

Count Plot

❖ countplot()

- Show the counts of observations in each category using bars
  - data – dataset for plotting
  - x, y – inputs for plotting data



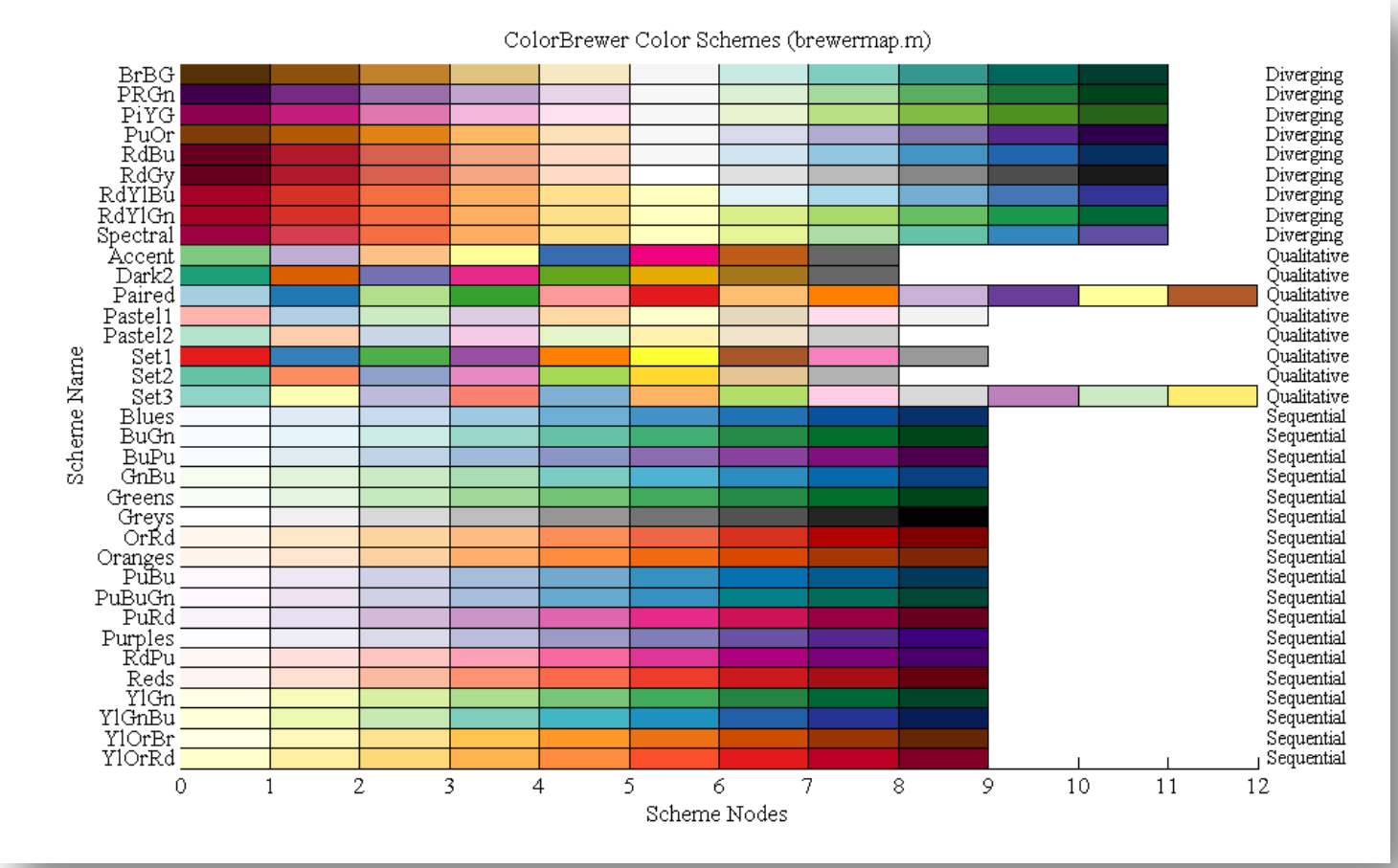
Count Plot

- **countplot()**
  - hue – visualize the data of different categories in one plot
  - palette – colors to use for the different levels of the hue variable

```
#importing library
import seaborn as sns
import matplotlib.pyplot as plt

#importing dataset
titanic = sns.load_dataset("titanic")

#plotting chart
sns.countplot(x='sex',
              data=titanic,
              hue="embark_town",
              palette="Blues")
plt.show()
```

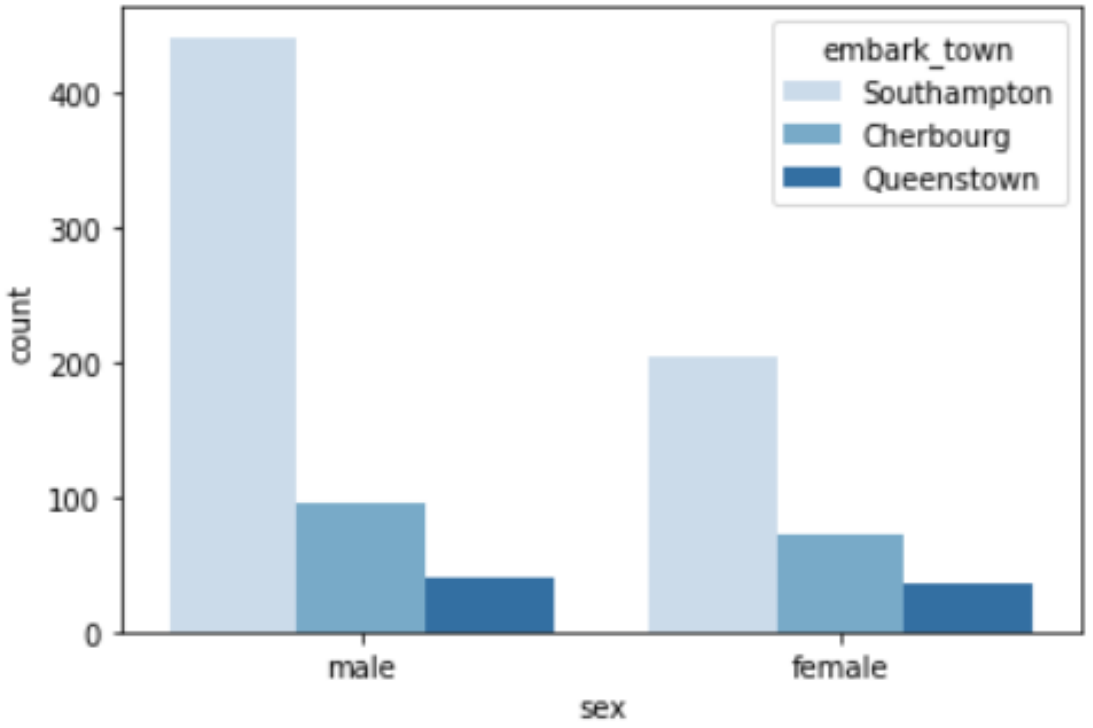
Count Plot

- Seaborn palette colors



Count Plot

- **countplot()**
  - hue – visualize the data of different categories in one plot
  - palette – colors to use for the different levels of the hue variable



Count Plot

Passenger	Survived	Pclass	Name
1	0	3	Braund, M
2	1	1	Cumings,
3	1	3	Heikkinen
4	1	1	Futrelle, M
5	0	3	Allen, Mr.
6	0	3	Moran, Mı
7	0	1	McCarthy,

❖ countplot()

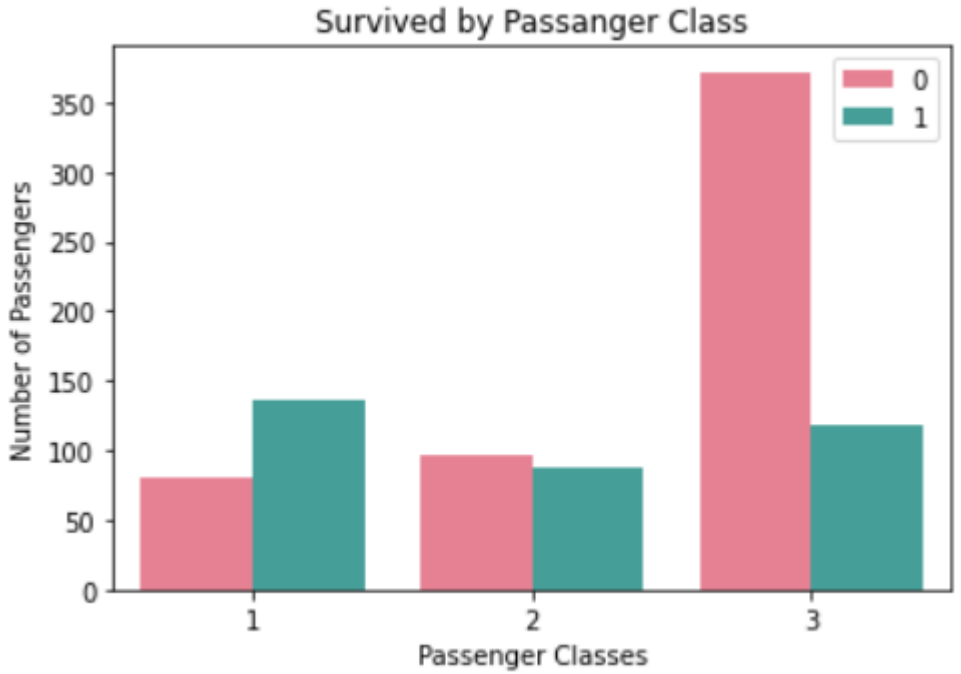
- You can add options like title, label names and legend location

```
#importing library
import seaborn as sns
import matplotlib.pyplot as plt

#importing dataset
titanic = sns.load_dataset("titanic")

#plotting chart x축이 plcass이고 hue는 x축 안에서 survived (생존자)를 카운팅하는 것
sns.countplot(x='pclass', data=titanic, hue='survived', palette="husl")
plt.title("Survived by Passanger Class")
plt.xlabel("Passenger Classes")
plt.ylabel("Number of Passengers")
plt.legend(loc='best')
plt.show()
```

Palette의 속성값을 15페이지에 있는 값으로 변경해보자

Count Plot

- **countplot()**
  - *plt.legend(loc='best')*
    - upper left, upper right, lower left, lower right, etc (Default value 'best')



loc의 값을 upper right, upper left, lower left, lower right로 변경해보자

Bar Plot

❖ barplot()

- Unlike count plot, bar plots must include x and y axis variables

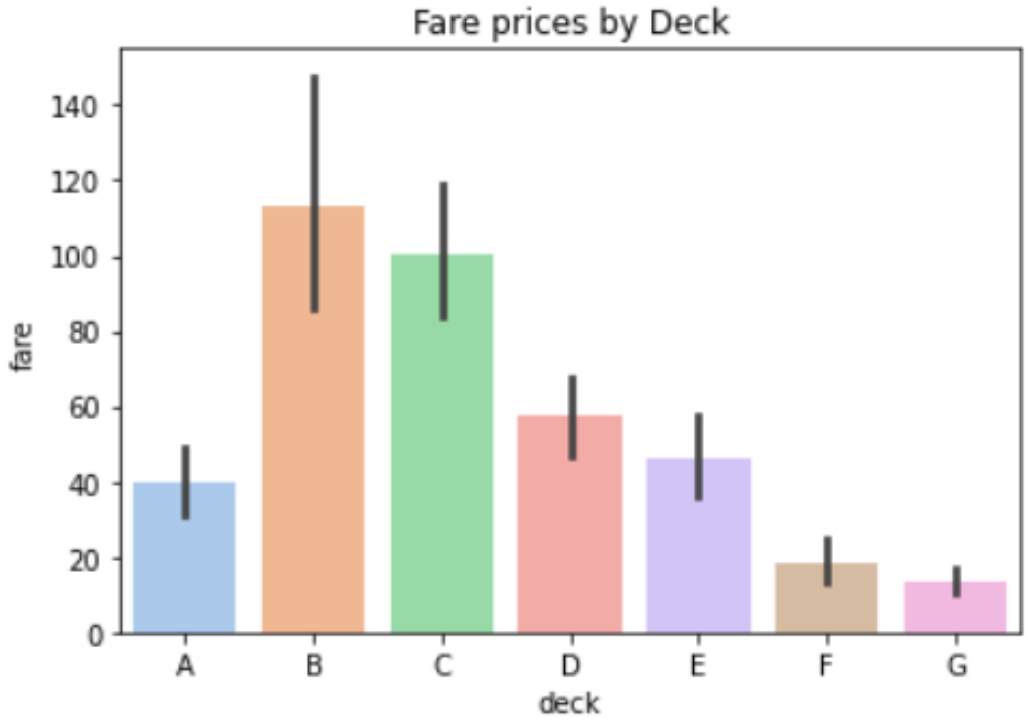
```
#importing library
import seaborn as sns
import matplotlib.pyplot as plt

#importing dataset
titanic = sns.load_dataset("titanic")

#plotting chart
sns.barplot(x='deck', y='fare', data=titanic, palette="pastel")
plt.title("Fare prices by Deck")
plt.show()
```

Deck 별 요금

Bar Plot

❖ barplot()

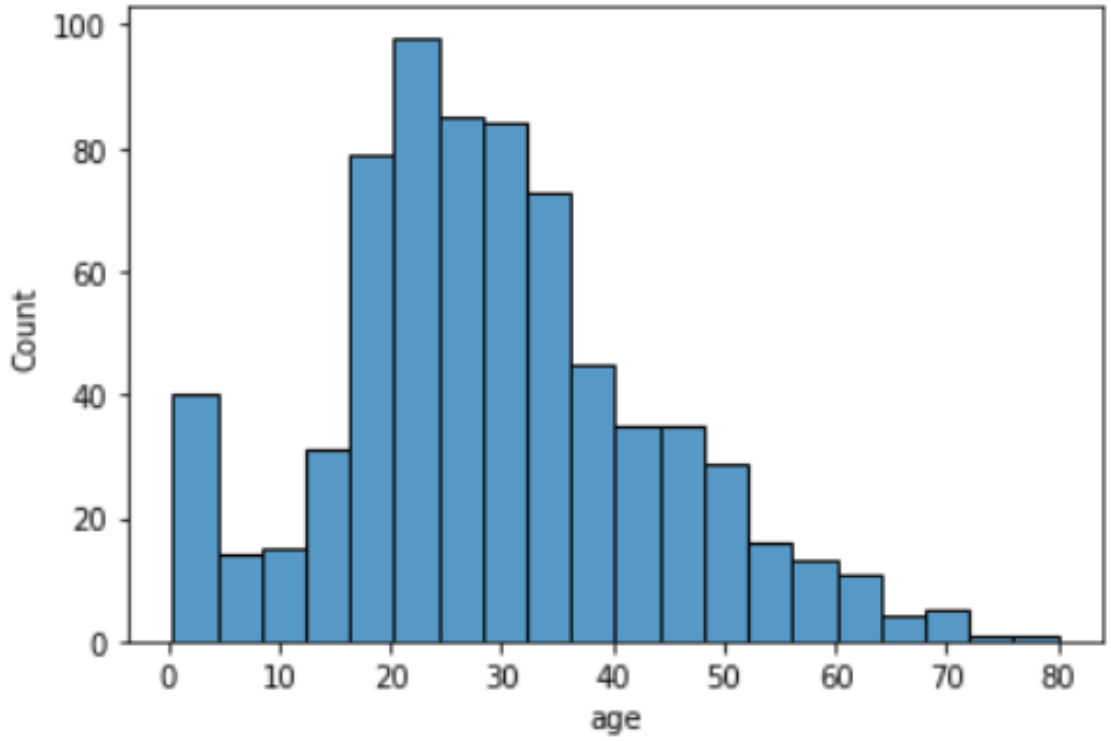
- Show point estimates and errors as rectangular bars
  - You can disable error bars with the option: errorbar=None
  - You can add bar labels with option: bar_label()



Errorbar의 옵션을 None으로 바꿔보고 bar_label을 추가해보자

Histogram Plot

- **histplot()**
  - Univariate or bivariate histograms to show distributions of datasets
    - *sns.histplot(x='age', data = titanic)*



Histogram Plot

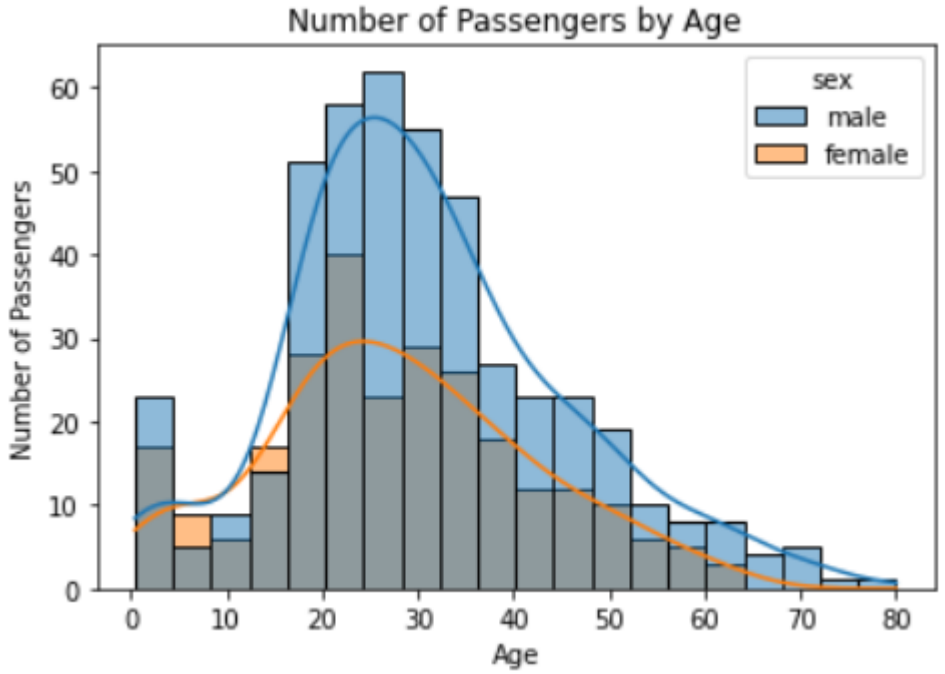
❖ histplot()

- Example of bivariate histograms with title and label names
- kde – kernel density estimation

```
#importing library
import seaborn as sns
import matplotlib.pyplot as plt

#importing dataset
titanic = sns.load_dataset("titanic")

#plotting chart    나이대별 성별 분포
sns.histplot(x='age', data = titanic, hue='sex', kde = True)
plt.title("Number of Passengers by Age")
plt.xlabel("Age")
plt.ylabel("Number of Passengers")
plt.show()
```

Histogram Plot

- **histplot()**
  - Example of bivariate histograms with title and label names
  - kde – kernel density estimation



Histogram Plot

histplot()

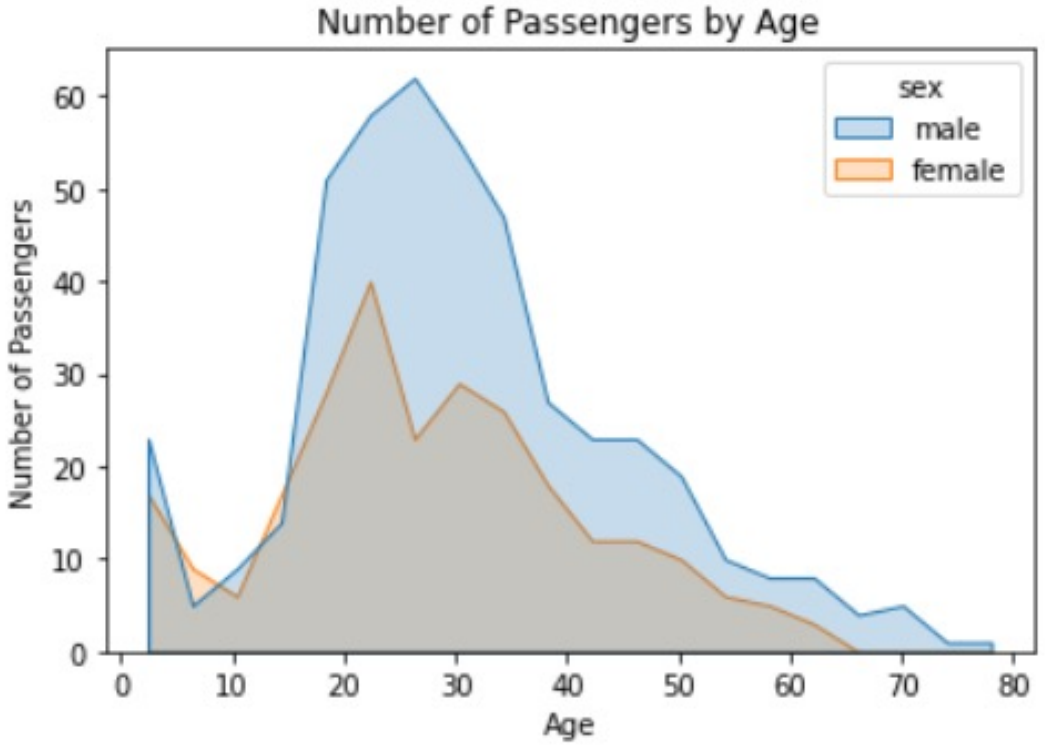
- Overlapping bars can be hard to visually resolve. A different approach would be to draw polygon

```
sns.histplot(x='age',
             data = titanic,
             hue='sex',
             element = 'poly')
```

- Other options include
  - element{"bars", "step", "poly"}

Histogram Plot

- **histplot()**
  - *element = 'poly'*



Element 속성 값을 다른 값으로 변경해보자

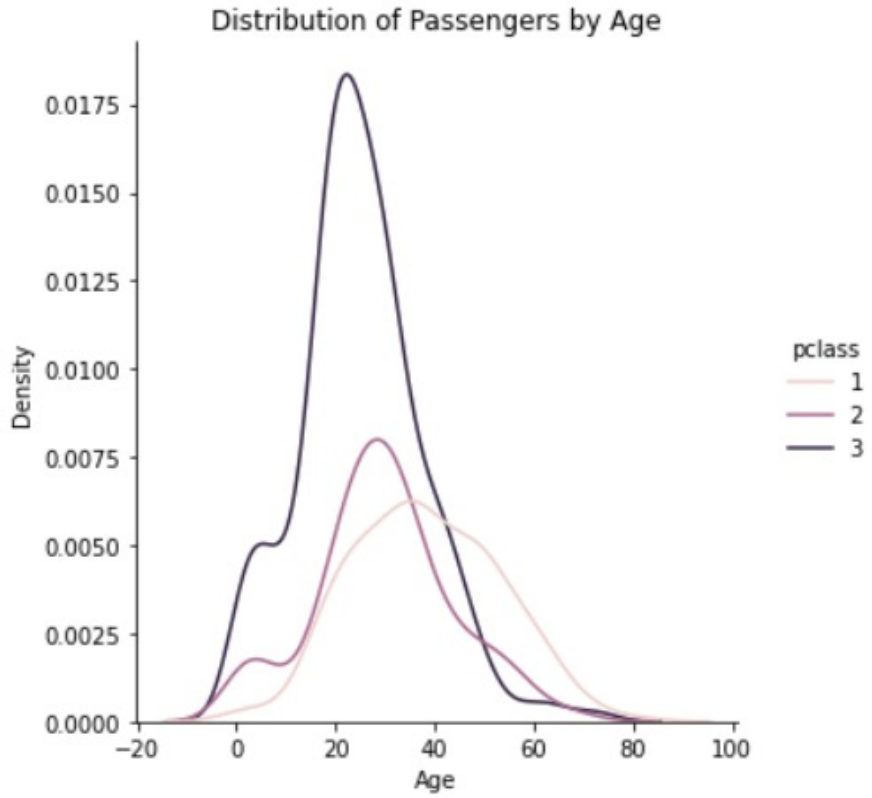
Distribution Plot

- **displot()**
  - Shows the univariate or bivariate distribution of data

```
#importing library
import seaborn as sns
import matplotlib.pyplot as plt

#importing dataset
titanic = sns.load_dataset("titanic")

#plotting chart
sns.displot(titanic, x='age', kind='kde', hue='pclass')

plt.title("Distribution of Passengers by Age")
plt.xlabel("Age")
plt.ylabel("Density")
plt.show()
```

Distribution Plot

- displot()
  - kind includes hist, kde, ecdf

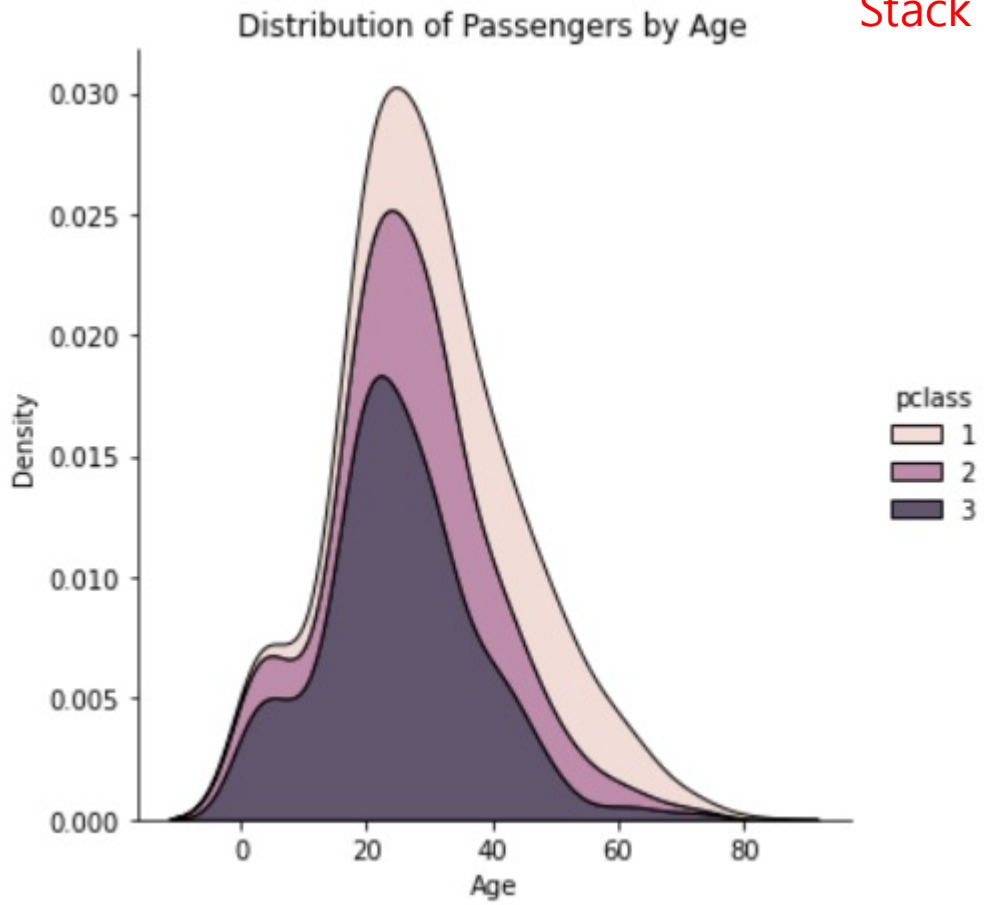


Kind를 다른 값으로 변경해보자

Distribution Plot

- displot()
  - sns.displot(titanic, x='age', kind='kde', hue='pclass', multiple='stack')

Stack 형태로 변경



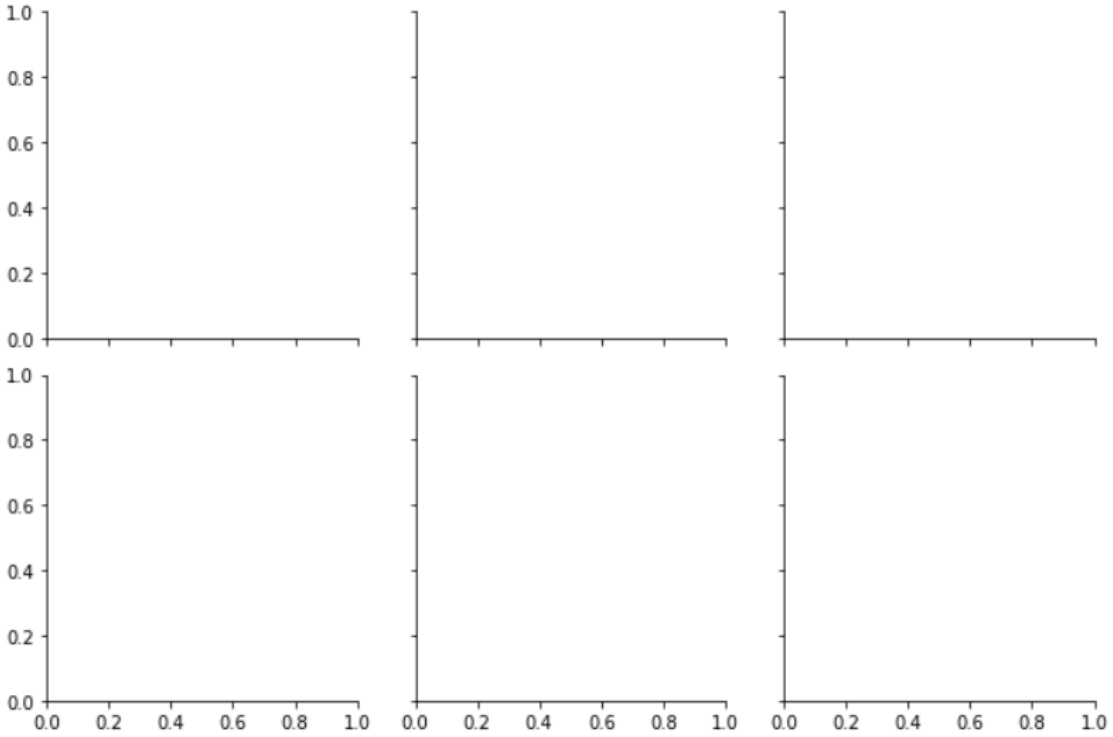
Facet Grids

- **FacetGrid()**
  - Multi-plot grid for plotting conditional relationships

```
#importing library
import seaborn as sns
import matplotlib.pyplot as plt

#importing dataset
titanic = sns.load_dataset("titanic")

#plotting chart
sns.displot(titanic, x='age', kind='kde', hue='pclass')

g = sns.FacetGrid(titanic,
                  col="pclass",
                  row="survived",
                  hue='sex')
g.map(sns.kdeplot, "age", shade = True)
```

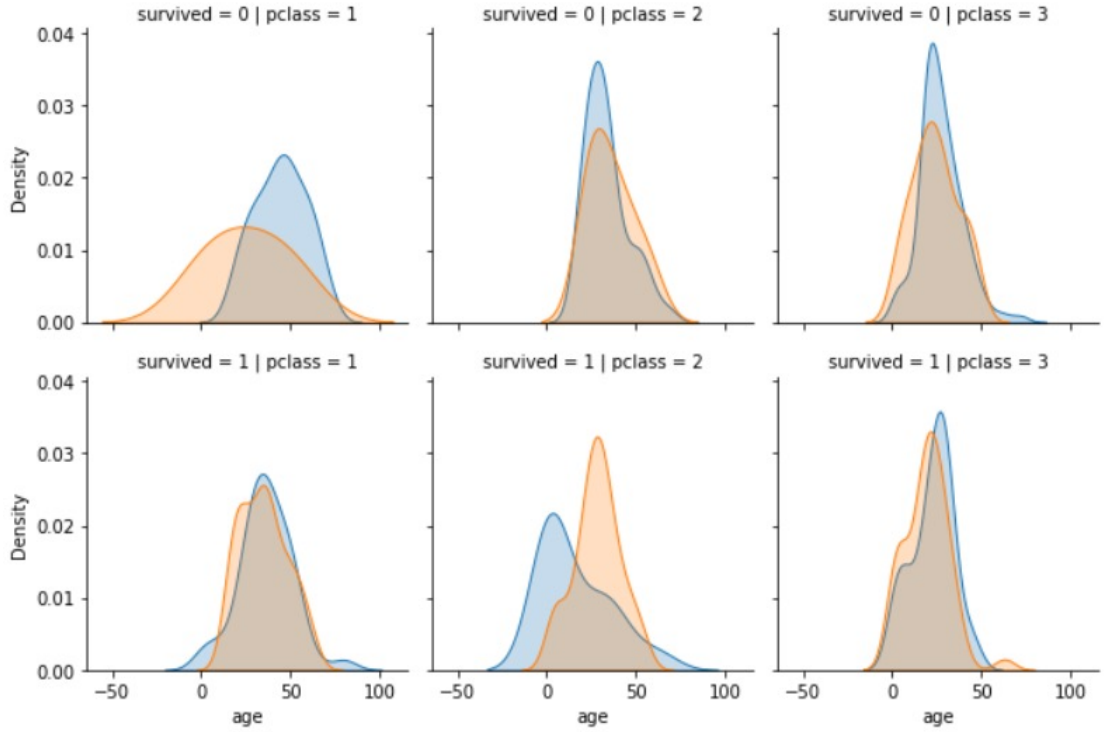
Facet Grids

- **FacetGrid()**
  - Two steps
    - Step 1: Create a basic plot with the layout



Facet Grids

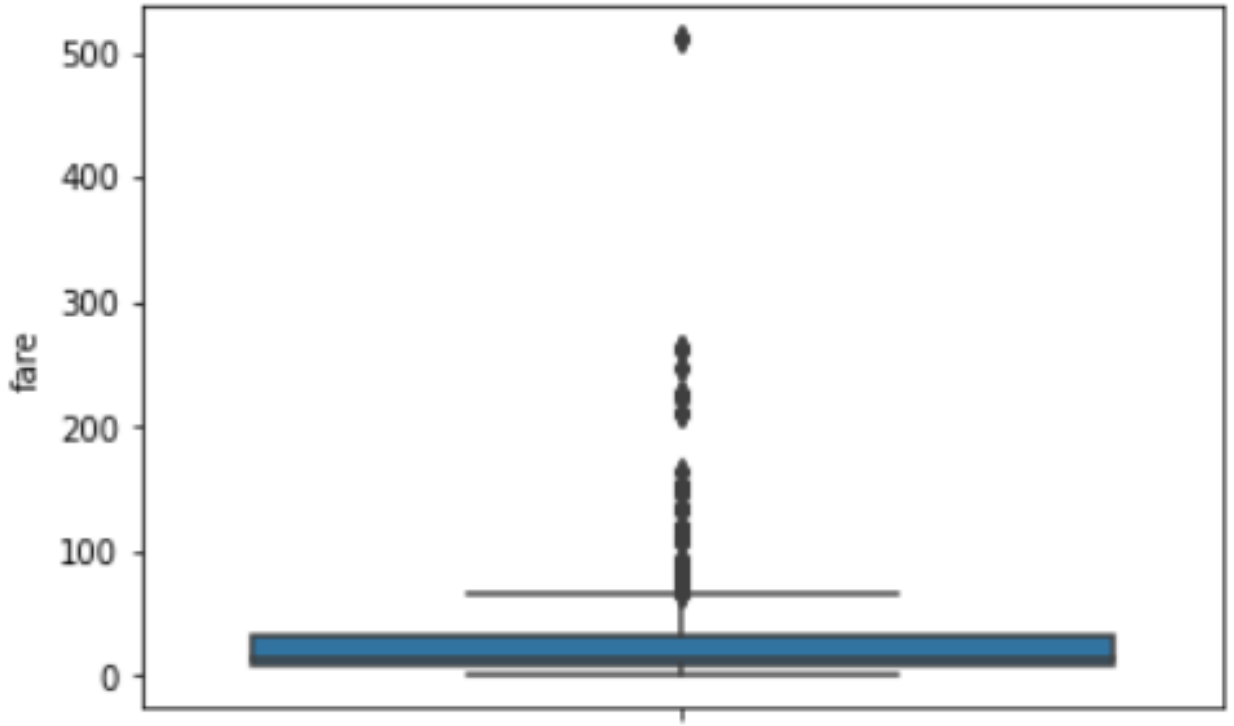
- **FacetGrid()**
  - Two steps
    - Step 2: Use map() function to apply plotting functions



Box Plot

❖ boxplot()

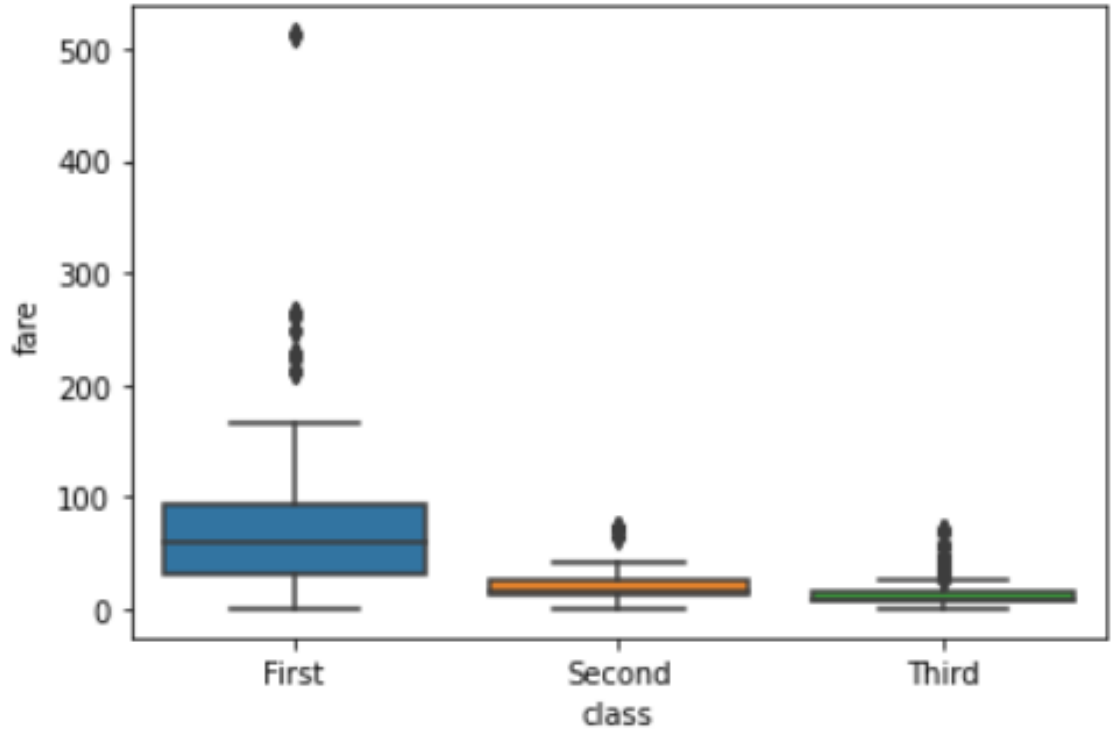
- Draw a box plot to show distributions with respect to categories
  - *sns.boxplot(y='fare', data=titanic)*



Box Plot

❖ boxplot()

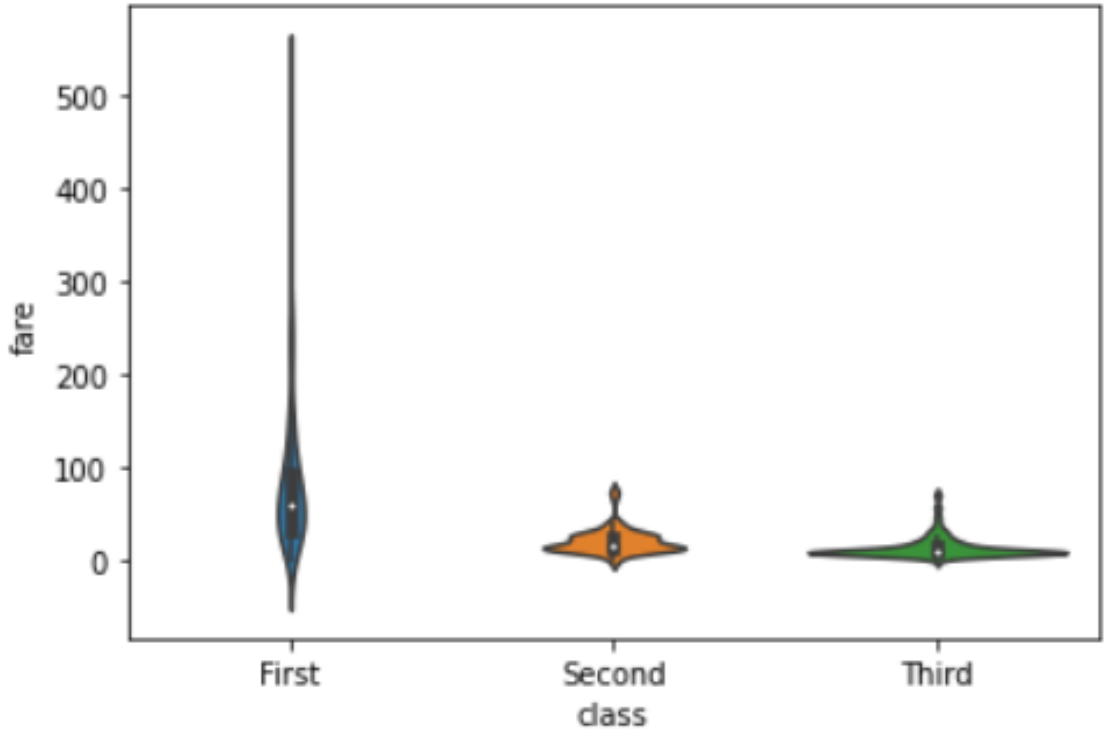
- You can create multiple boxes in a single plot
  - *sns.boxplot(x = 'class', y='fare', data=titanic)*



Box Plot

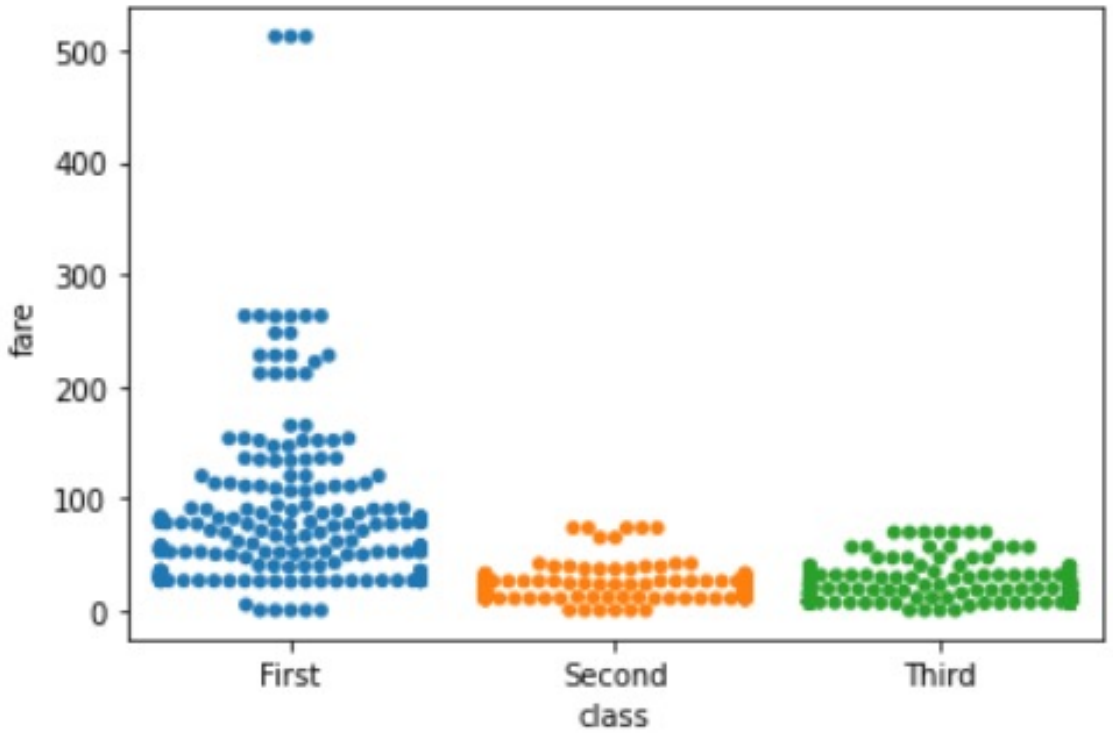
❖ violinplot()

- Draw a patch representing a KDE
  - *sns.violinplot(y='fare', x = 'class', data=titanic)*



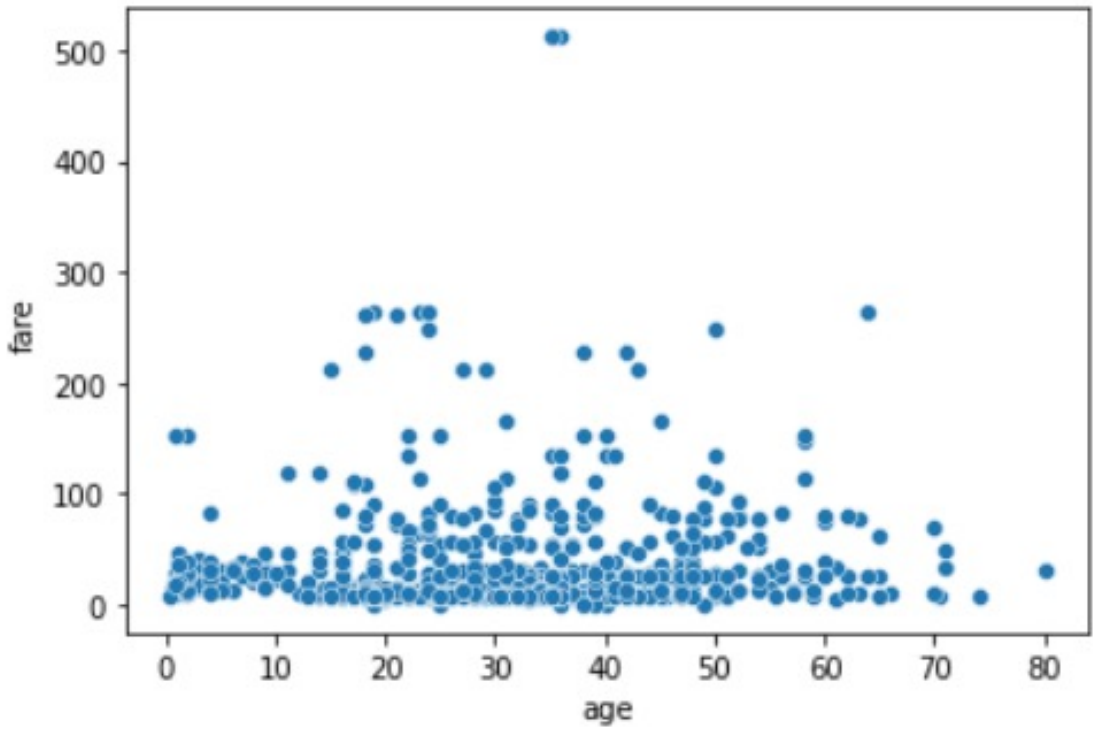
Box Plot

- swarmplot()
  - Draw a categorical scatterplot with points adjusted to be non-overlapping
    - *sns.swarmplot(y='fare', x = 'class', data=titanic)*



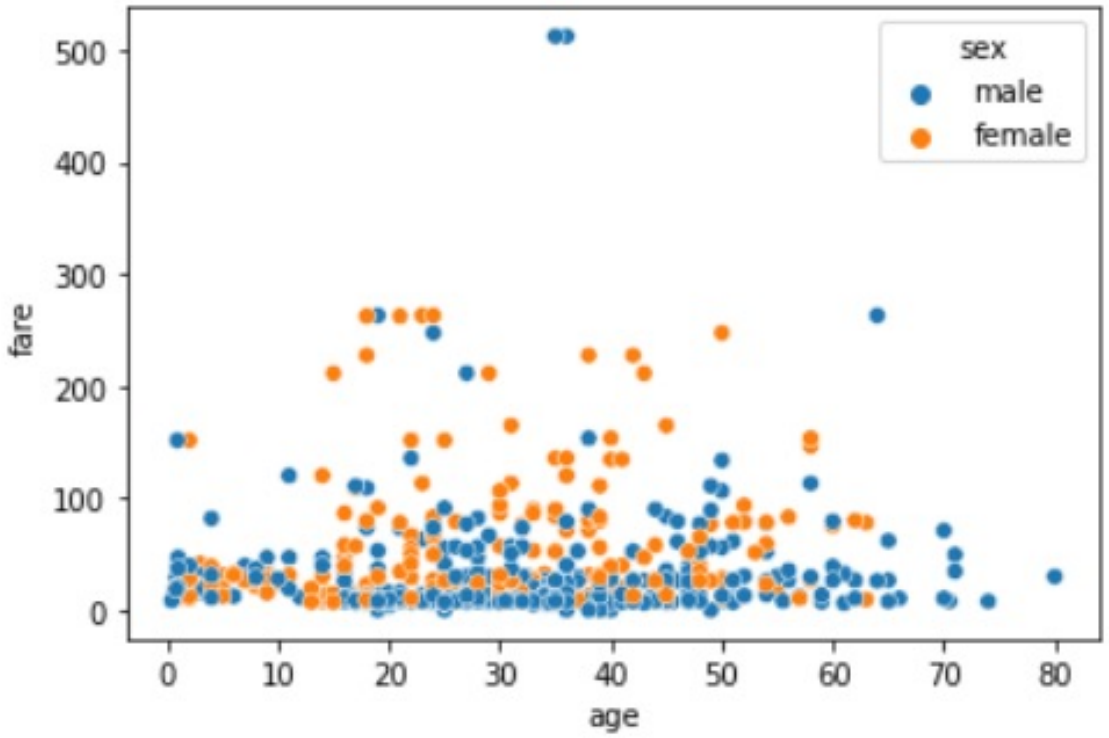
Scatter Plot

- scatterplot()
  - Draw a scatter plot with possibility of several semantic groupings
    - *sns.scatterplot(x='age', y = 'fare', data = titanic)*



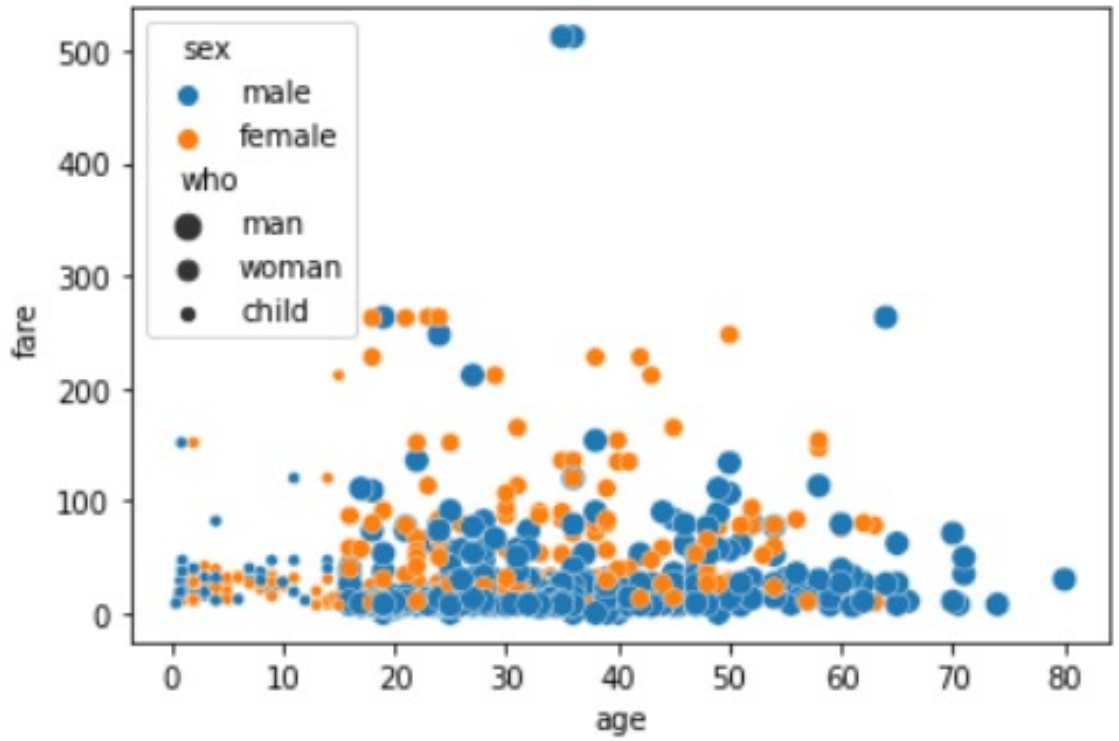
Scatter Plot

- scatterplot()
  - You can include hue option to show different categories in one plot
    - *sns.scatterplot(x='age', y = 'fare', data = titanic, hue = 'sex')*



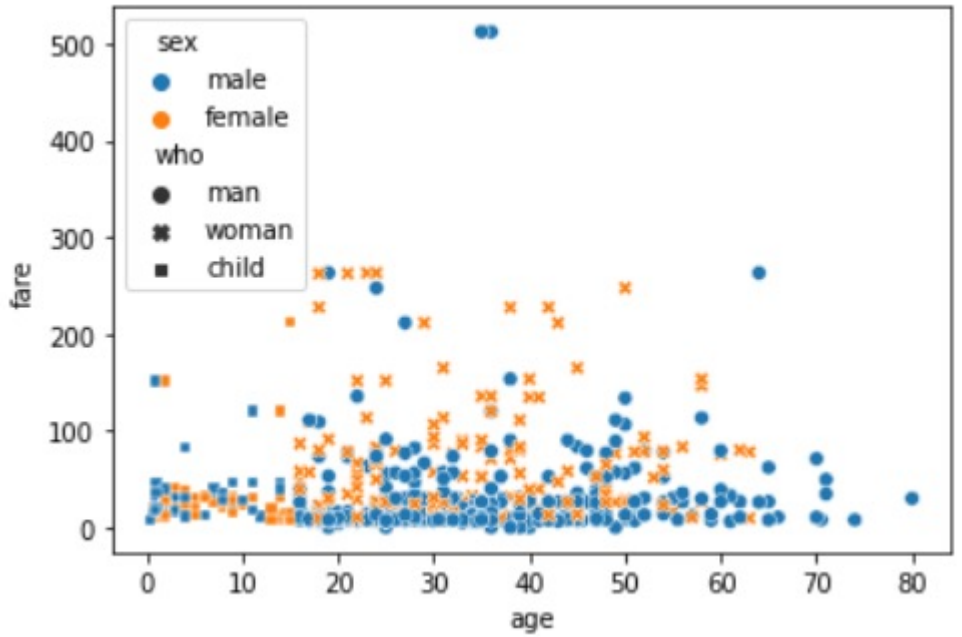
Scatter Plot

- **scatterplot()**
  - size – variable that will produce points with different sizes
    - *sns.scatterplot(x='age', y = 'fare', data = titanic, hue = 'sex', size = 'who')*



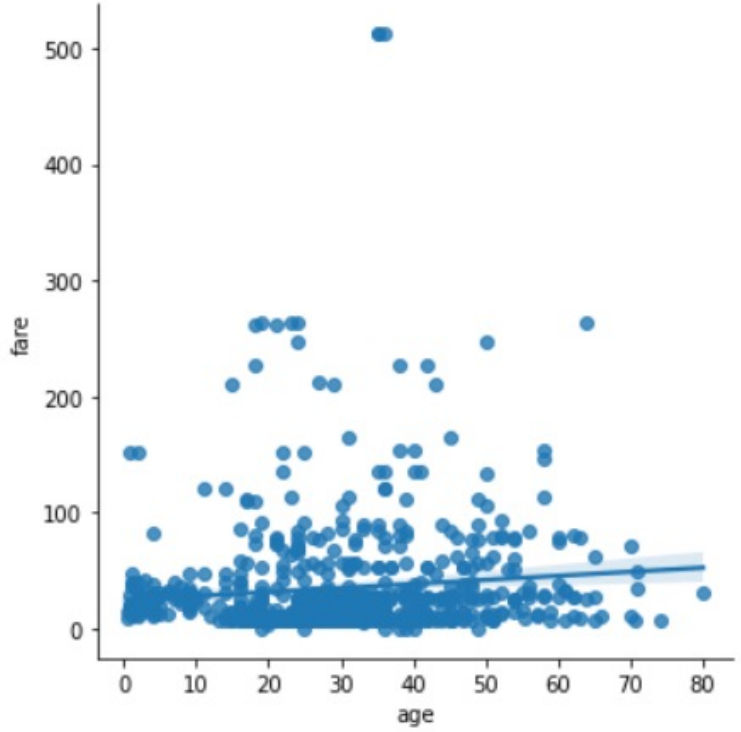
Scatter Plot

- **scatterplot()**
  - You can change the style of markers
    - *sns.scatterplot(x='age', y = 'fare', data = titanic, hue = 'sex',*
      style = 'who')



Scatter Plot

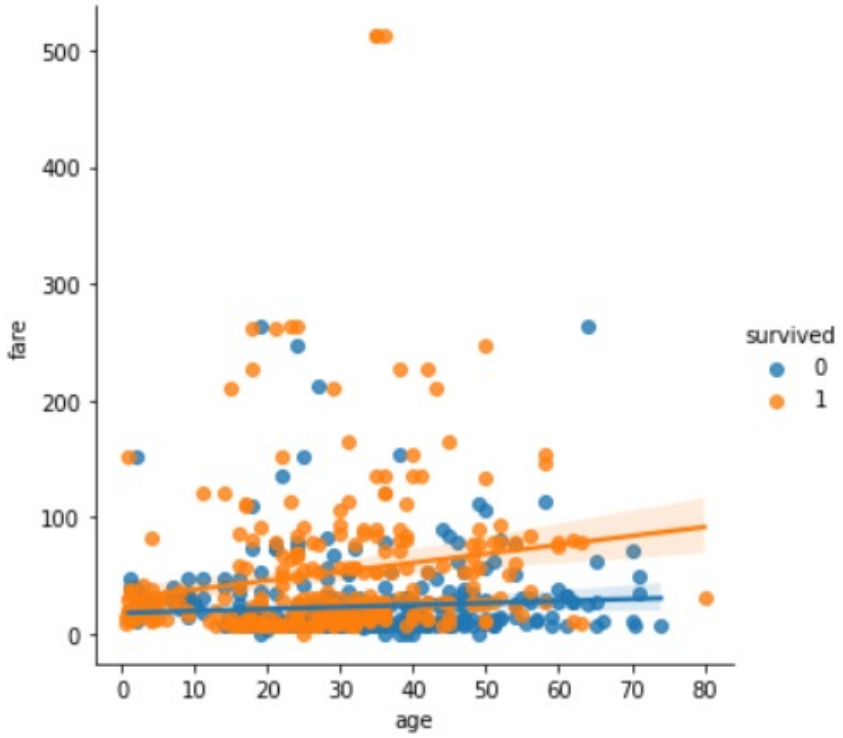
- **lmplot()**
  - Plot data and regression model
    - *sns.lmplot(x='age', y = 'fare', data = titanic)*



Scatter Plot

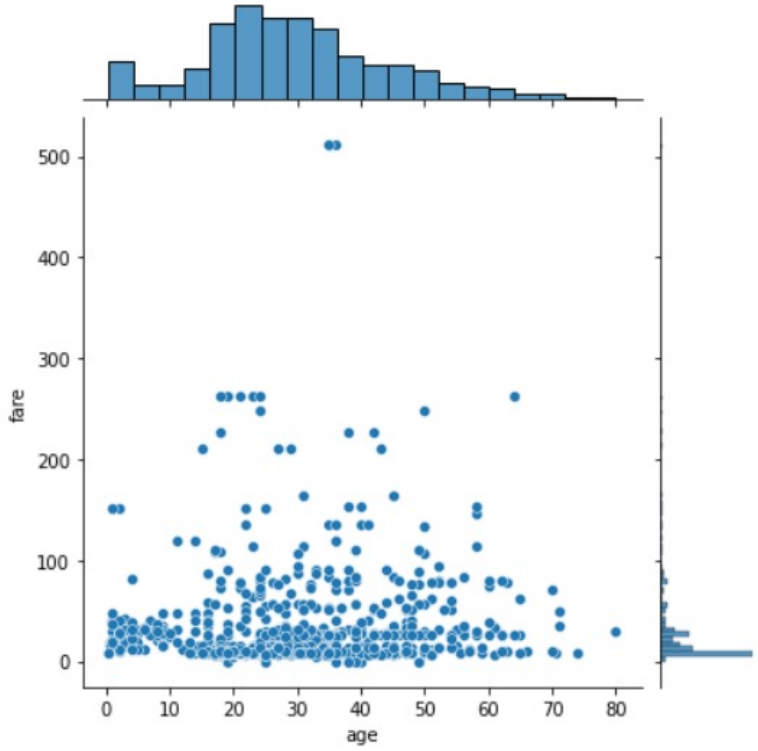
❖ lmplot()

- You can include hue option to show different categories in one plot
  - *sns.lmplot(x='age', y = 'fare', data = titanic, hue = 'survived')*



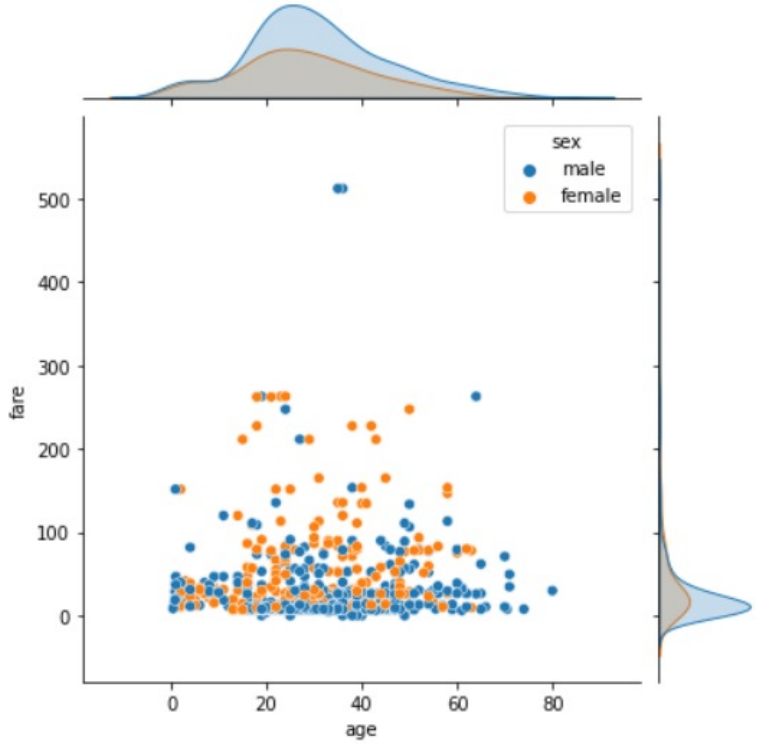
Joint Plot

- **jointplot()**
  - Draw a plot of two variables with bivariate and univariate graphs
    - *sns.jointplot(x='age', y='fare', data=titanic)*



Joint Plot

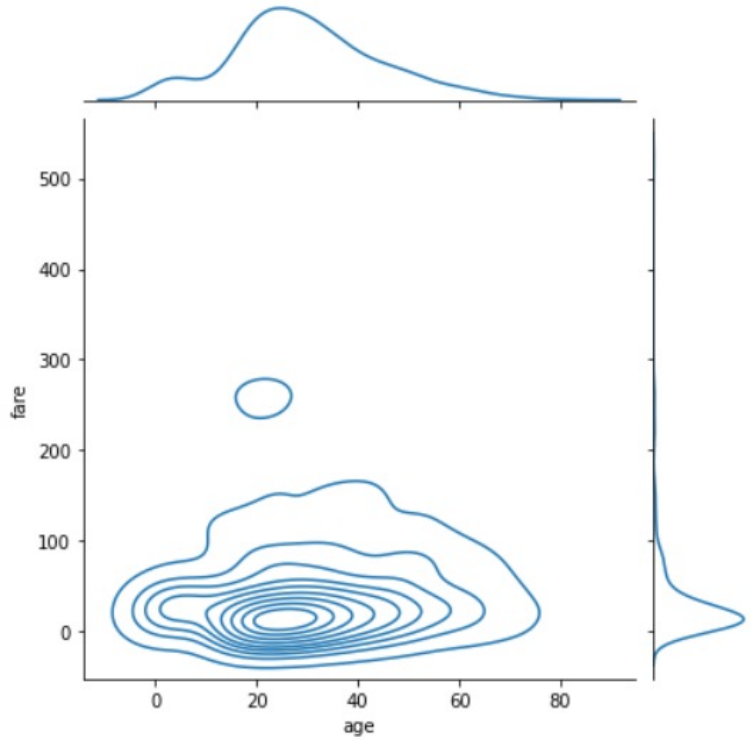
- **jointplot()**
  - Draw a plot of two variables with bivariate and univariate graphs
    - *sns.jointplot(x='age', y='fare', data=titanic, hue='sex')*



Joint Plot

❖ jointplot()

- You can also draw the distribution graph
  - *sns.jointplot(x='age', y='fare', data=titanic, kind="kde")*

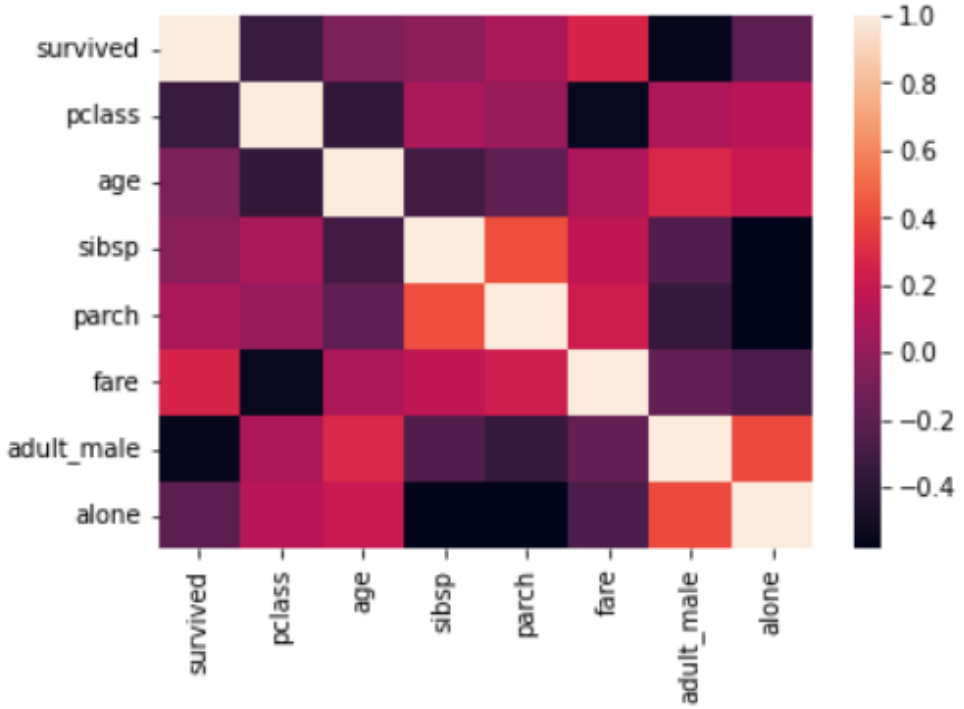


다른 kind를 적용해보자

Heat Plot

❖ heatmap()

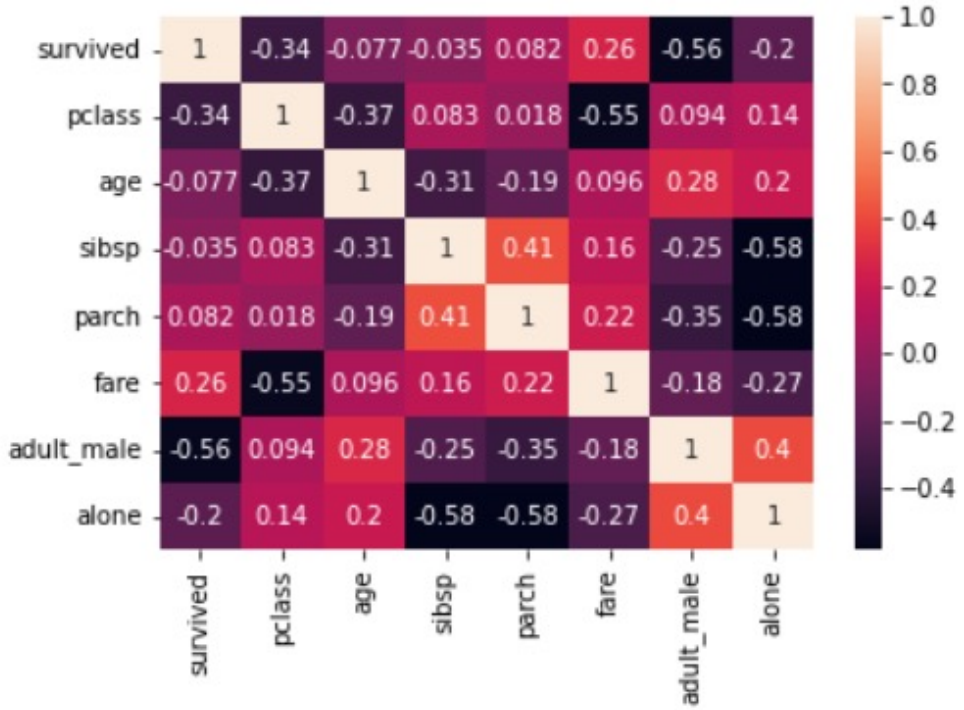
- Plot rectangular data as a color-encoded matrix
  - *titanic_corr = titanic.corr()*
  - *sns.heatmap(data = titanic_corr)*



Heat Plot

❖ heatmap()

- annot=True to show the labels on the heatmap
  - *titanic_corr = titanic.corr()*
  - *sns.heatmap(data = titanic_corr, annot=True)*



Homework for Lecture 6

- **Perform the following**
  - Use on of the dataframes provided by Seaborn
  - Apply basic charts
    - Bar and scatter charts
  - Apply statistical charts
    - Boxplot, violin and joint plots
  - Study at least two more by yourself and apply it to the dataset
  - Add a detailed explanation for each graph

감사합니다!