2024-1

Multivariate Data Analysis

Youjin Shin
(yj.shinn@catholic.ac.kr)

Understanding Data Characteristics

◆ Understanding Data Characteristics

➢ The Foundation of Data Analysis

- Before diving into any data analysis, it's crucial to thoroughly understand the data you're working with. This understanding forms the bedrock upon which analysis methods are chosen and applied.

➢ Characteristics to Consider

- **Type of Data:** Knowing whether your data is nominal, ordinal, interval, or ratio is fundamental in selecting the right statistical tools.
- **Distribution of Data:** The spread and shape of your data can significantly affect which analyses are appropriate.
- **Potential Relationships:** Being aware of possible connections between variables can guide you in choosing the right analysis method.

Data Types

◆ Understanding Data Types

- Understanding the types of data is crucial in both univariate and multivariate analysis, as it guides the selection of appropriate statistical methods and data visualization techniques.

• Univariate Data

- Univariate data involves the analysis of a single variable. Its main purpose is to describe the data and find patterns that occur within it. Here are the main types:

1. **Nominal (Categorical) Data**: Represents categories or groups that do not have a natural order. Examples include gender (male, female), types of cuisine (Italian, Mexican, Japanese), and blood type (A, B, AB, O).
    - **Dataset Example:** Survey responses on preferred type of pet.
    - **Data Points:** {Dog, Cat, Bird, Fish, Reptile}
    - **Analysis:** Count or percentage of each pet preference in the survey.

Data Types

2. **Ordinal Data:** Like nominal data, but the categories have a logical or natural order. However, the intervals between the categories are not necessarily equal. Examples include education level (high school, bachelor's, master's, Ph.D.), satisfaction rating (very unsatisfied, unsatisfied, neutral, satisfied, very satisfied), and socio-economic status (low, middle, high).

   - **Dataset Example**: Feedback on a new product rated from 1 to 5.
   - **Data Points:** {1 (Very Unsatisfied), 2 (Unsatisfied), 3 (Neutral), 4 (Satisfied), 5 (Very Satisfied)}
   - **Analysis:** Median rating or distribution of satisfaction levels.

Data Types

3. **Interval Data:** Numeric data where the distance between two values is meaningful. However, there is no true zero point. Examples include temperature in Celsius or Fahrenheit, where the difference between 20°C and 30°C is the same as between 30°C and 40°C, but 0°C does not mean the absence of temperature.

    - **Dataset Example:** Temperatures of different cities on a specific day.
    - **Data Points:** {New York: 20°C, Los Angeles: 25°C, Chicago: 15°C, Miami: 30°C}
    - **Analysis:** Average temperature, temperature range, or comparison of temperature differences between cities.

Data Types

4. **Ratio Data**: Also numeric and similar to interval data, but with a meaningful zero value, allowing for the calculation of ratios. Examples include height, weight, age, and income. For instance, someone who earns $100,000 makes twice as much as someone who earns $50,000, and 0 means no income.

    - **Dataset Example**: Heights of students in a class.
    - **Data Points:** {170 cm, 160 cm, 165 cm, 175 cm, 180 cm}
    - **Analysis:** Average height, height distribution, or identifying the tallest/shortest student.

Data Types

◆ Multivariate Data

- Multivariate data involves the analysis of more than one variable at a time, allowing for the examination of relationships and patterns between variables. The types of data involved can be the same as univariate data (nominal, ordinal, interval, ratio), but they are analyzed in combination. Examples of multivariate data include:

1. **Survey Data:** A survey might collect various types of data from respondents, such as age (ratio), gender (nominal), income level (ratio), satisfaction with a service (ordinal), and frequency of service use (ratio). Analyzing these variables together can reveal insights into customer behavior and preferences.

   - **Dataset Example:** Survey responses including gender, preferred type of pet, and satisfaction rating.
   - **Data Points:** Each respondent's data might look like {Gender: Female, Pet Preference: Cat, Satisfaction: 4 (Satisfied)}
   - **Analysis:** Explore relationships between gender and pet preference, or satisfaction levels among different pet preferences.

Data Types

2. **Health Data:** Medical research often collects multivariate data, such as patient age (ratio), gender (nominal), blood pressure (ratio), cholesterol levels (ratio), and disease classification (nominal). Researchers can explore relationships between lifestyle factors and health outcomes.

   - **Dataset Example:** Patients' age, gender, blood pressure, and disease classification.
   - **Data Points:** A patient's record could be {Age: 45, Gender: Male, Blood Pressure: 120/80 mmHg, Disease: Hypertension}
   - **Analysis:** Investigate correlations between age, blood pressure, and the prevalence of hypertension; analyze differences by gender.

Data Types

3. **Financial Data:** Financial analysts might examine multiple variables such as company revenue (ratio), profit margin (ratio), market share (ratio), and industry category (nominal) to identify factors driving financial performance.

   - **Dataset Example:** Companies' annual revenue, profit margin, and industry category.
   - **Data Points:** A company's data might look like {Revenue: $10M, Profit Margin: 15%, Industry: Technology}
   - **Analysis:** Analyze the financial health across industries; compare revenue and profit margins between technology and non-technology sectors.

Data Types

4. **Environmental Data:** Studies on environmental change might include variables like temperature (interval), precipitation levels (ratio), pollution levels (ratio), and types of vegetation cover (nominal) across different regions and times.

   - **Dataset Example:** Monthly average temperature, precipitation levels, and types of vegetation cover in different regions.
   - **Data Points:** Region data could be {Temperature: 22°C, Precipitation: 120 mm, Vegetation: Forest}
   - **Analysis:** Study climate patterns and their impact on vegetation types; compare environmental factors across regions.

Data Visualization

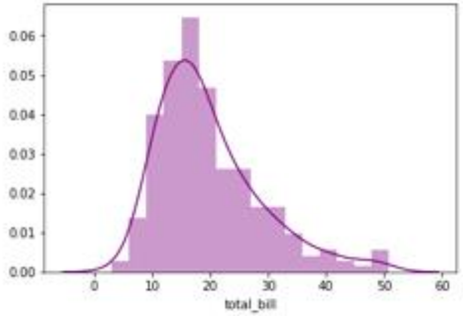
◆ Data Visualization

- Understanding the characteristics of your data and employing visualization techniques are indispensable steps in the data analysis process. They ensure that the chosen methods are suitable and that key features of the data are not overlooked. This foundation enables more accurate, insightful, and actionable findings.

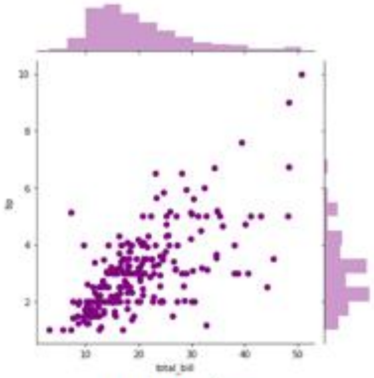
◆ Python Data Visualization Libraries

Seaborn

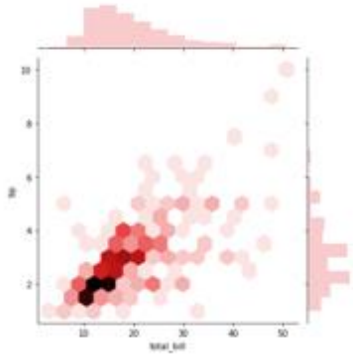
◆ Seaborn Plots



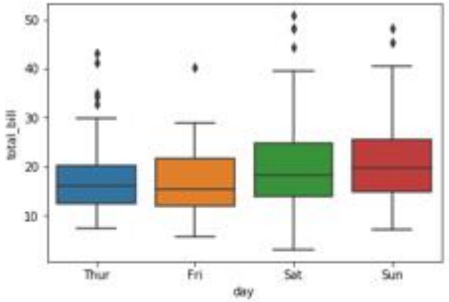
distplot



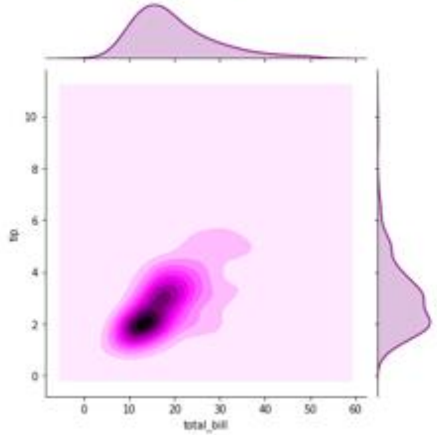
Jointplot



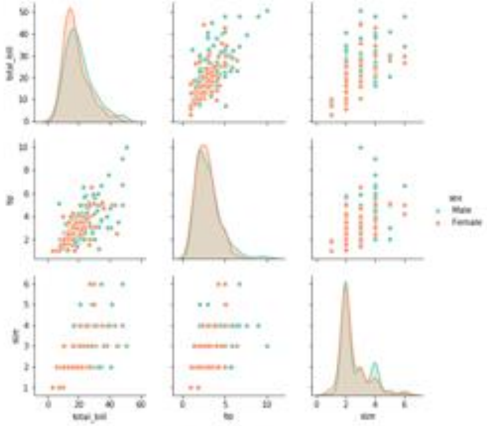
Hexplots



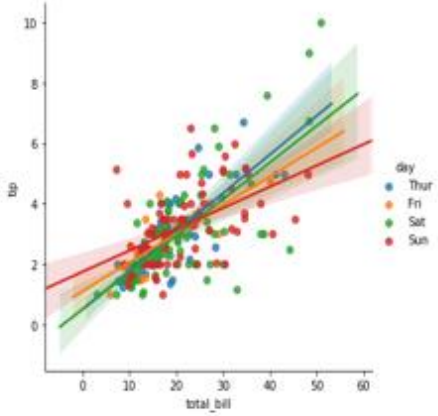
Boxplots



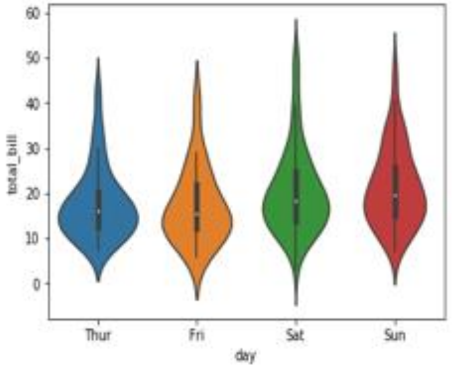
KDE Plot



Pair Plots



LM Plots



Violin Plots

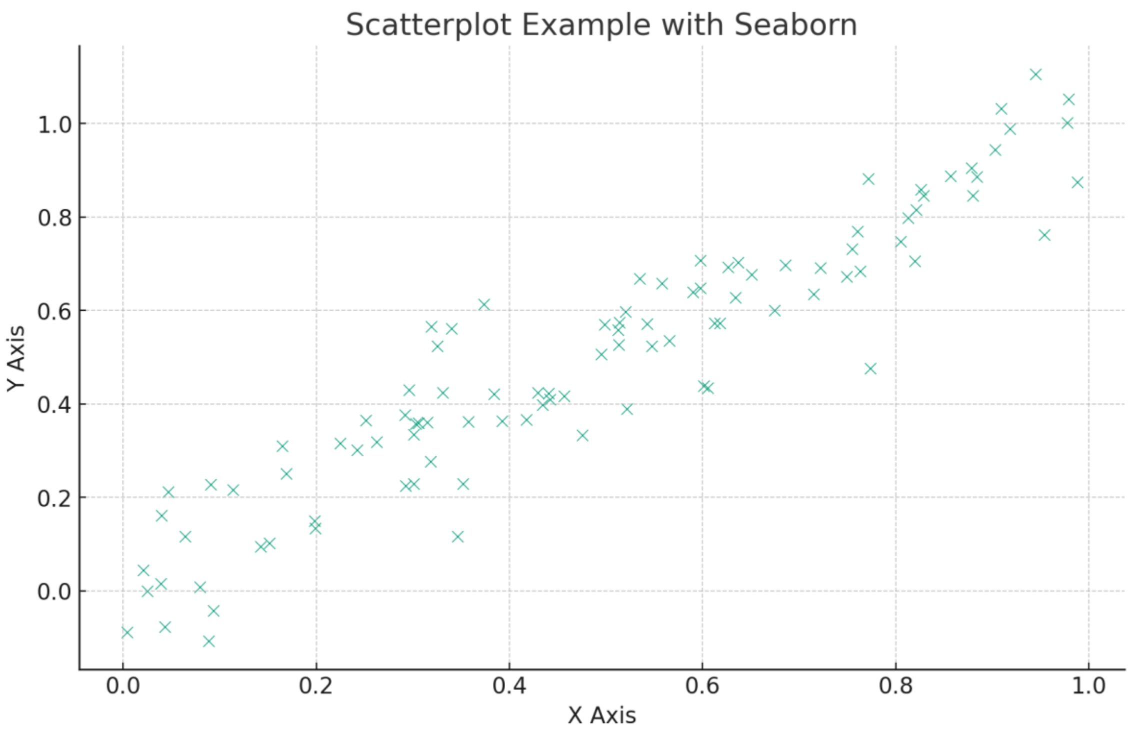
Seaborn

1. Relational Plots

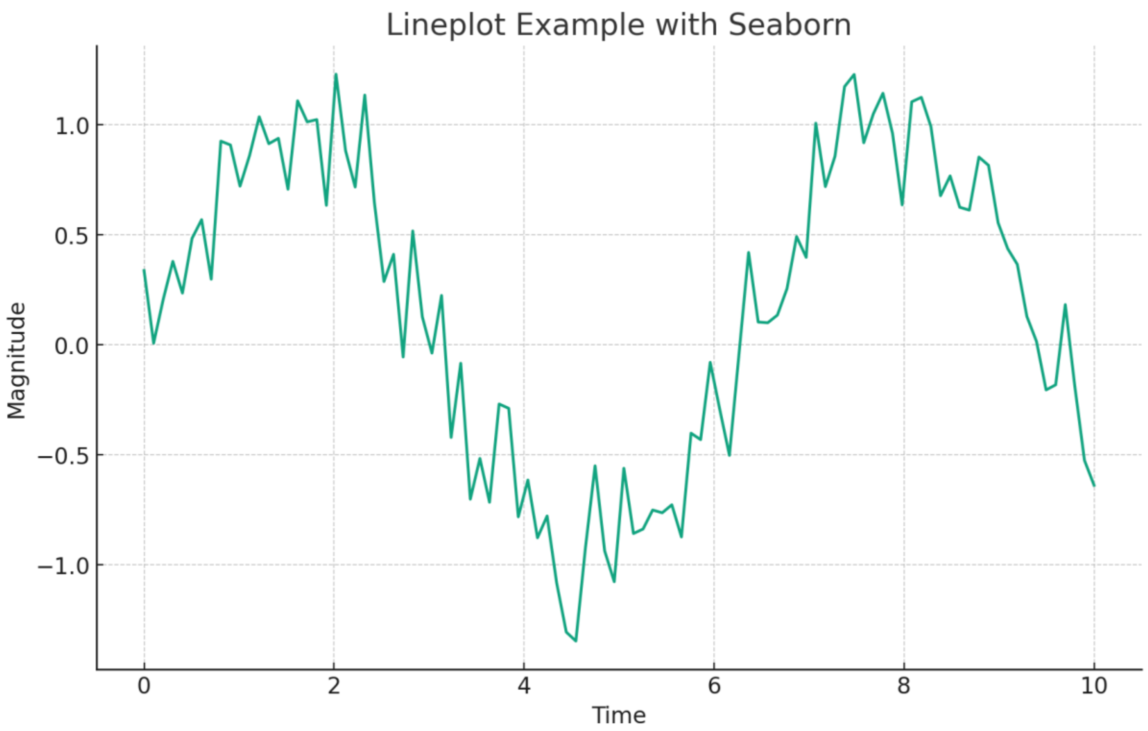
Relational plots are used to understand the relationship between two or more variables.

- **scatterplot():** Draws a scatter plot to show the relationship between two variables. It's useful for observing the distribution and relationship between two numerical variables.
- **lineplot():** Draws a line plot that is useful for visualizing data over time or ordered categories, showing trends over a continuous or categorical variable.

Seaborn



Scatterplot Example with Seaborn



Lineplot Example with Seaborn

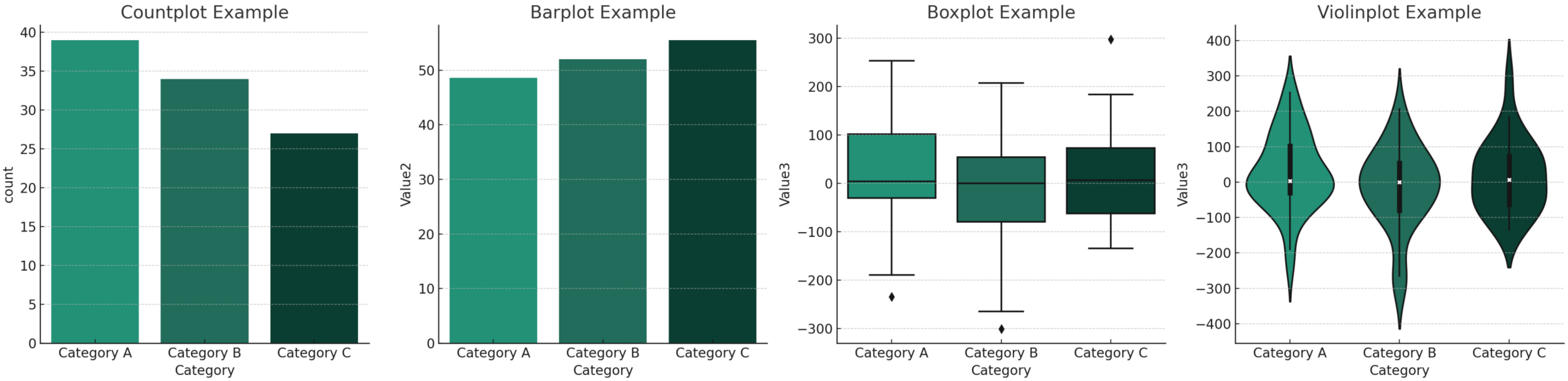
Seaborn

2. **Categorical Plots**

These plots are used for visualizing data where at least one variable is categorical.

- **catplot():** A figure-level function that provides access to several axes-level functions for plotting categorical data, including countplot, barplot, boxplot, violinplot, and more. It can create plots of different types with a unified interface.
    - **countplot():** Shows the counts of observations in each categorical bin using bars.
    - **barplot():** Displays the mean (or other estimators) of a numerical variable for each category of a categorical variable, with error bars.
    - **boxplot():** Shows the distribution of quantitative data across different categories using the quartiles of the dataset while highlighting outliers.
    - **violinplot():** Combines aspects of boxplots and kernel density plots, showing the distribution of the data across different categories and its probability density.

Seaborn


Countplot Example
count
0
5
10
15
20
25
30
35
40
Category A
Category B
Category C
Category
Barplot Example
Value2
0
10
20
30
40
50
Category A
Category B
Category C
Category
Boxplot Example
Value3
−300
−200
−100
0
100
200
300
Category A
Category B
Category C
Category
Violinplot Example
Value3
−400
−300
−200
−100
0
100
200
300
400
Category A
Category B
Category C
Category

Seaborn

◆ Barplot vs. countplot

① barplot() is used when you want to observe the relationship between a categorical variable and a continuous variable. The barplot function will display the mean (by default) of the continuous variable for each category and also includes error bars to show the uncertainty around the mean estimate.

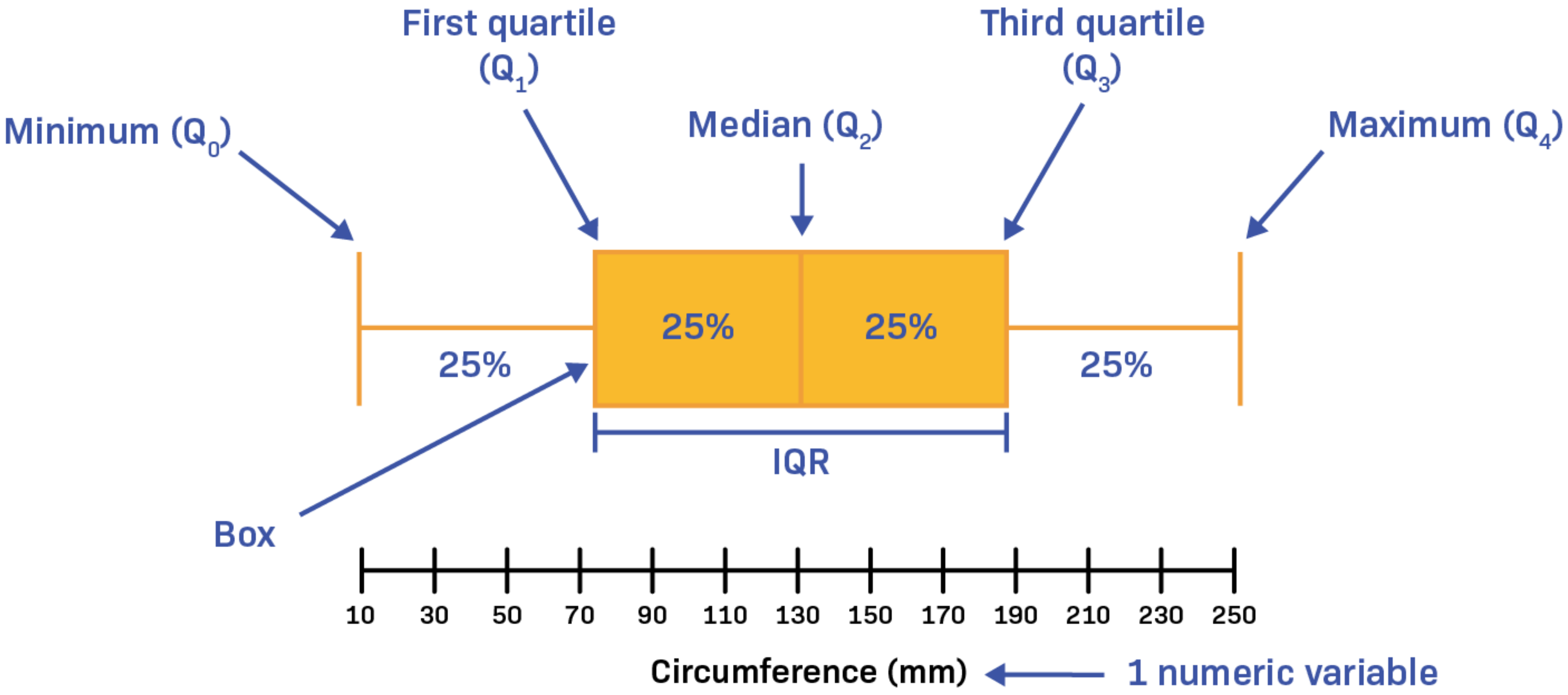
- ➢ Example: If you're interested in understanding the average test scores (a continuous variable) for different study groups (a categorical variable), a barplot would be suitable.

② countplot() is essentially a special case of a barplot where the dataset is categorical, and you're simply counting the number of occurrences in each category.

- ➢ Example: If you simply want to count the number of students in each study group (a categorical variable), you would use a countplot.

Seaborn

◆ Boxplot



Seaborn

◆ **Violinplot**

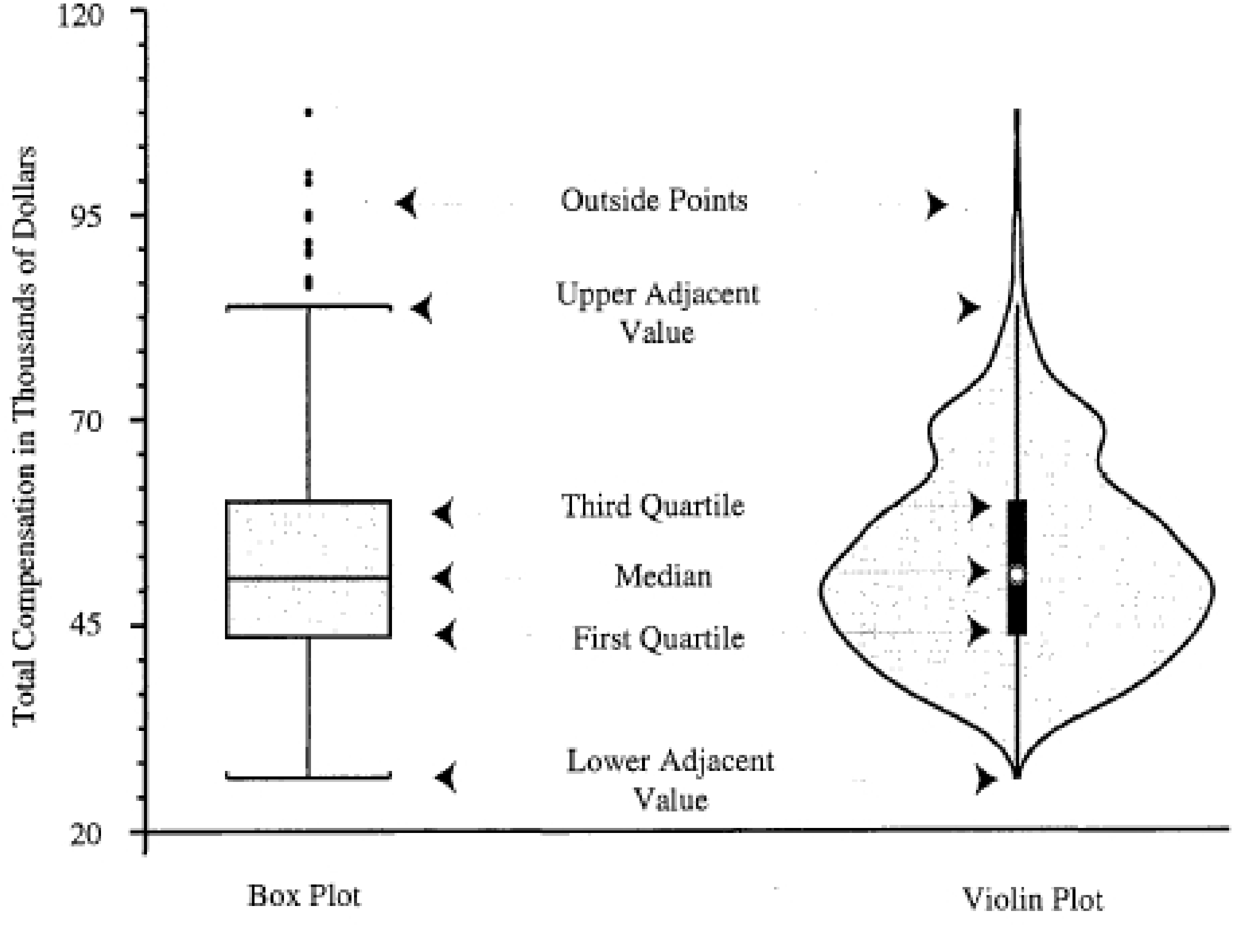


Figure 1. Common Components of Box Plot and Violin Plot. Total compensation for all academic ranks.

Seaborn

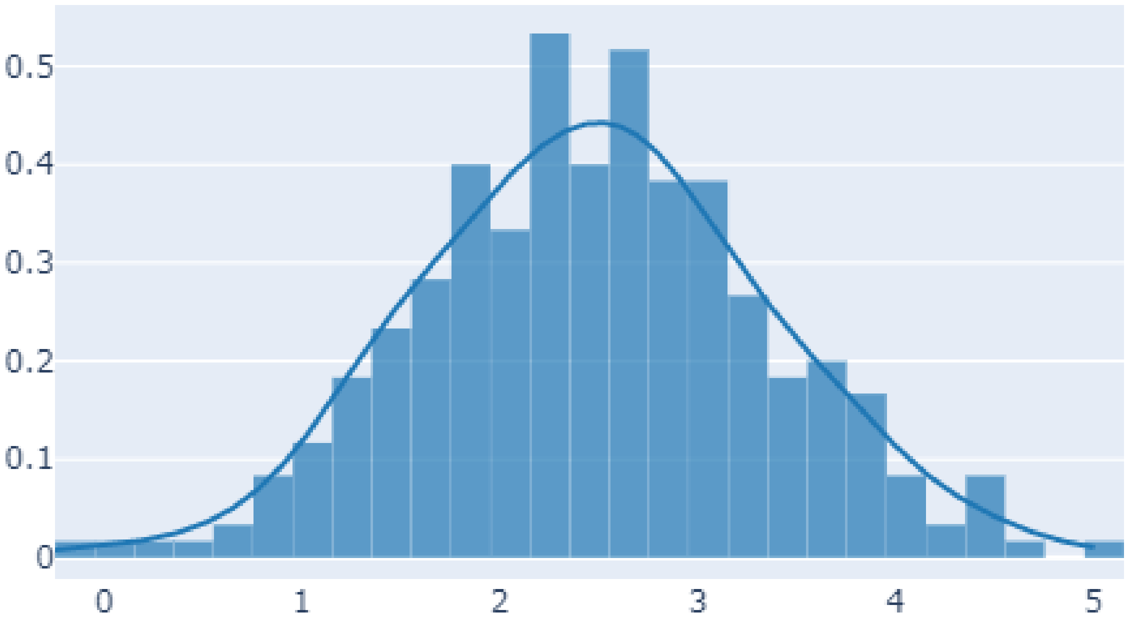
3. Distribution Plots

Distribution plots are designed to visualize the distribution of a dataset.

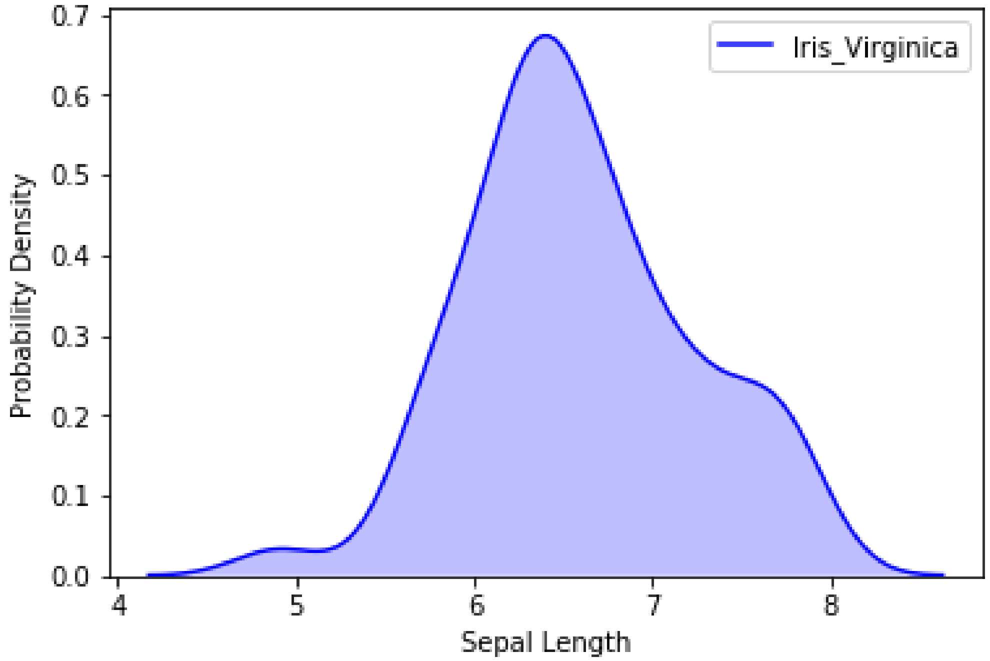
- **distplot()/histplot():** Visualizes the distribution of a univariate dataset as a histogram and can also fit a kernel density estimate (KDE) to the data.
- **kdeplot():** Plots the kernel density estimate, which is a way to estimate the probability density function of a continuous random variable.
- **jointplot():** Creates a multi-panel figure that shows both the bivariate (or joint) relationship between two variables along with the univariate (or marginal) distribution of each on separate axes.

Seaborn

distplot()/histplot()

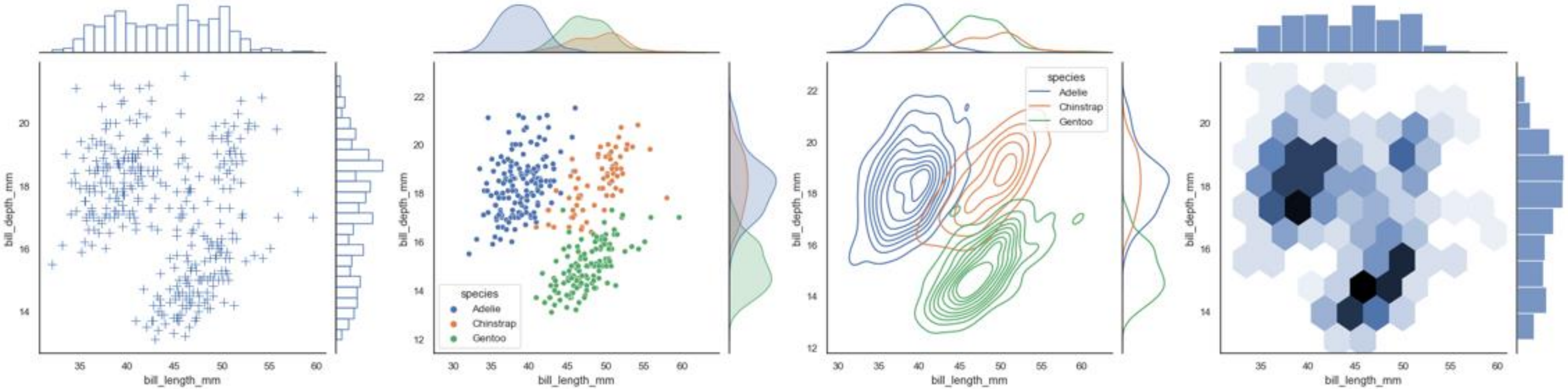


Kdeplot()



Seaborn

jointplot()



Seaborn

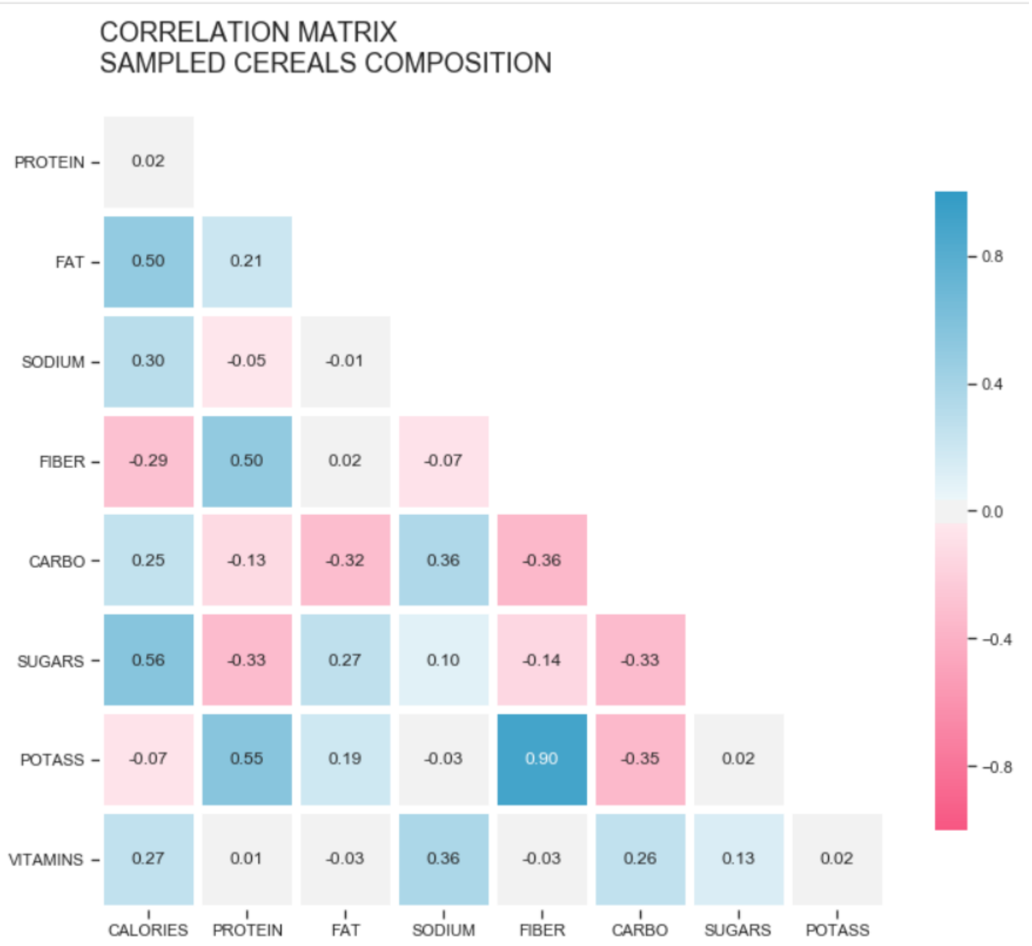
4. Matrix Plots

Matrix plots allow for plotting data as color-encoded matrices and are useful for visualizing the relationships between multiple variables.

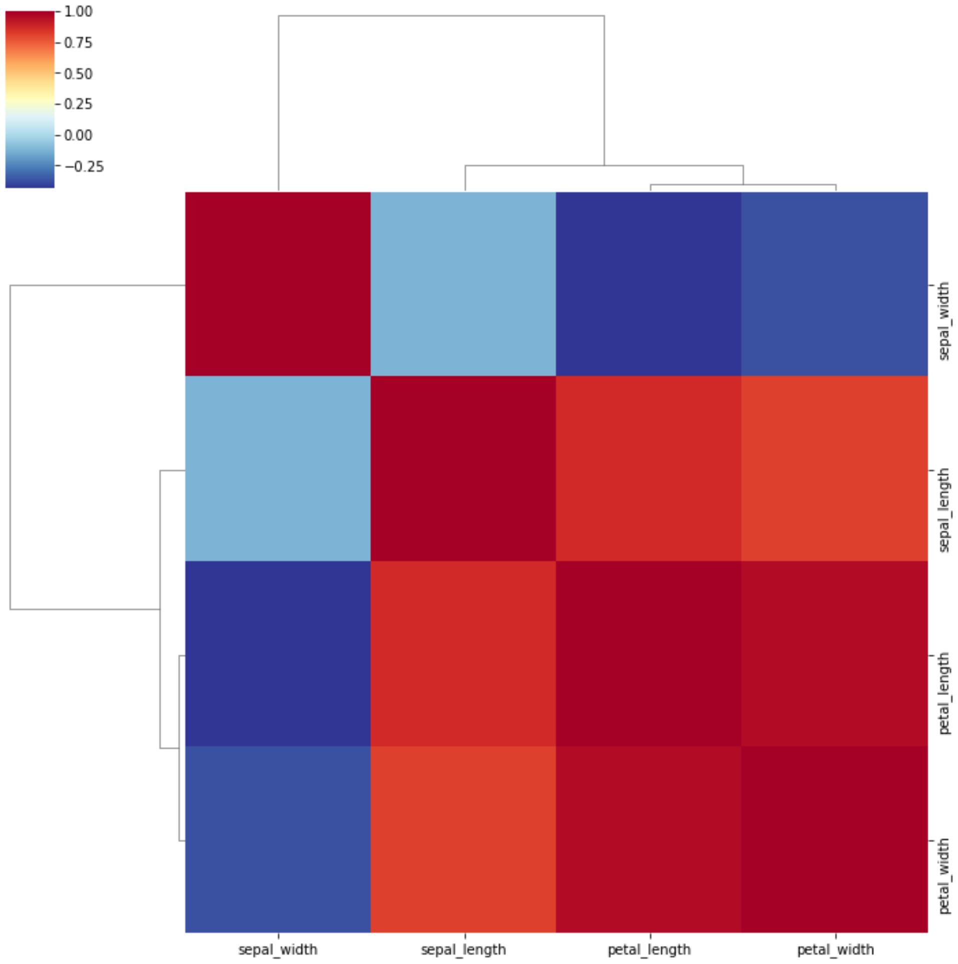
- **heatmap():** Visualizes data as a color-encoded matrix, which can be helpful for showing correlations between variables or for displaying a confusion matrix.
- **clustermap():** Performs hierarchical clustering and displays a heatmap of the clustered data, providing insights into the structure of the data.

Seaborn

heatmap()



clustermap()



Seaborn

5. Regression Plots

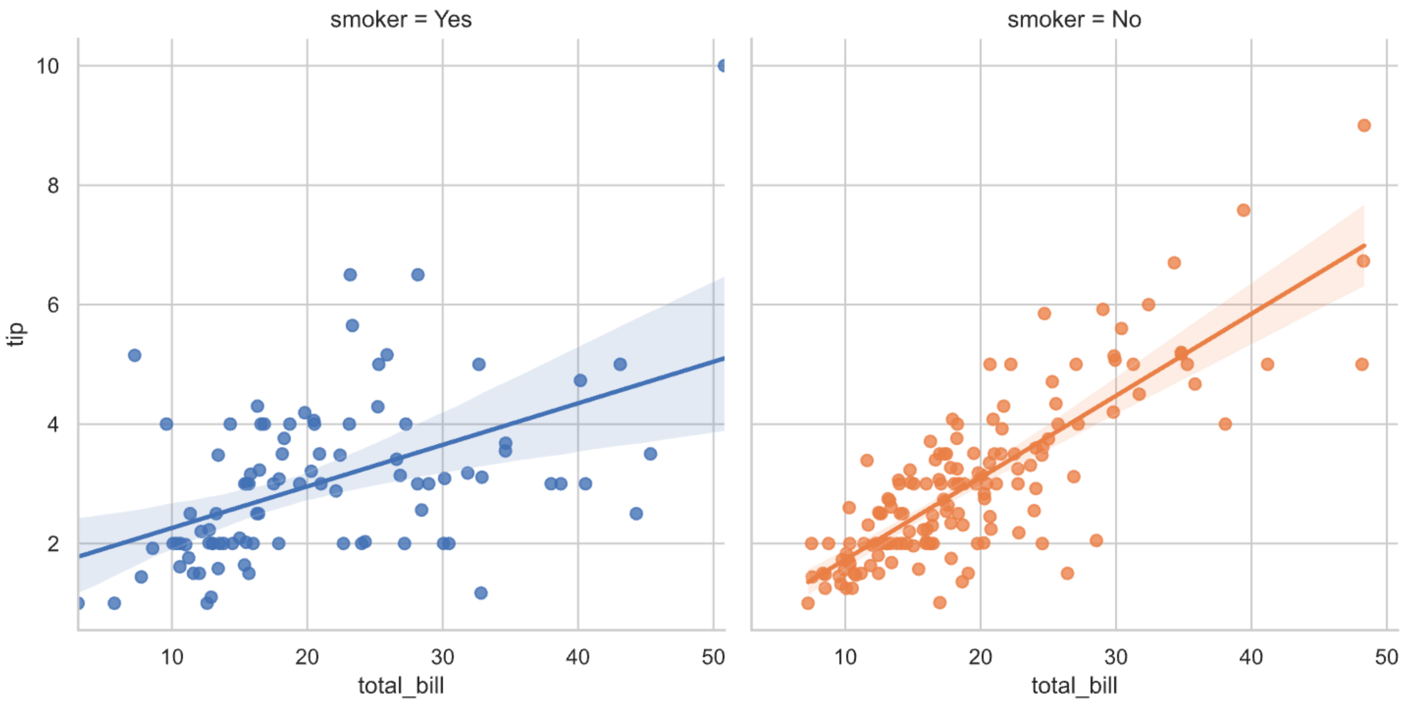
These plots are used to visualize the linear relationships between variables.

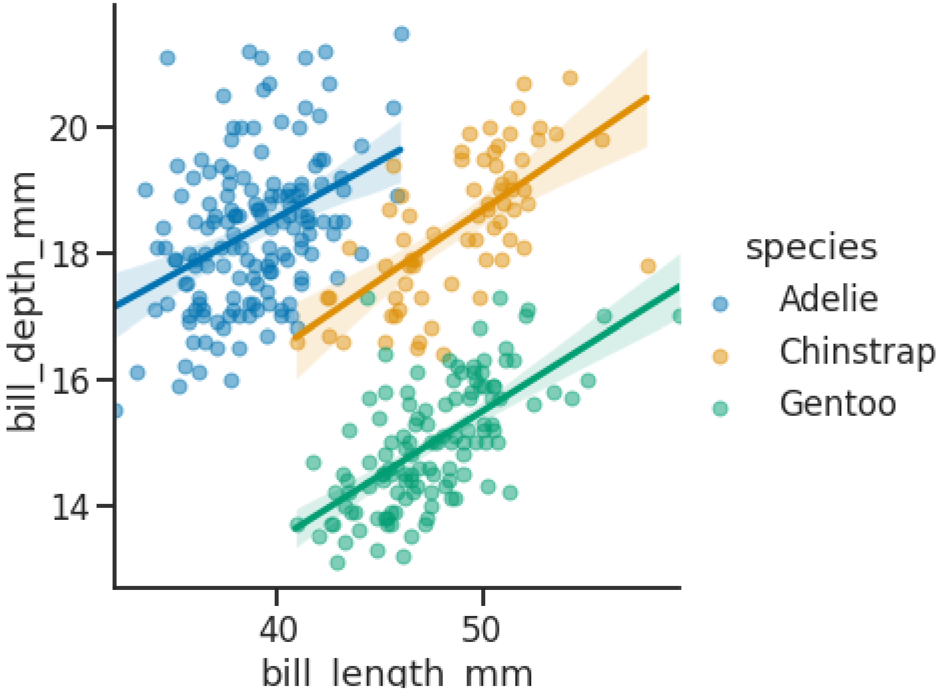
- **regplot():** Plots the scatter plot and fits a linear regression model to it.
- **lmplot():** A figure-level function that combines regplot() and FacetGrid, allowing you to plot linear regression models across different subsets of the dataset.

Seaborn

regplot()

lmplot()





Seaborn

6. Multi-Plot Grids

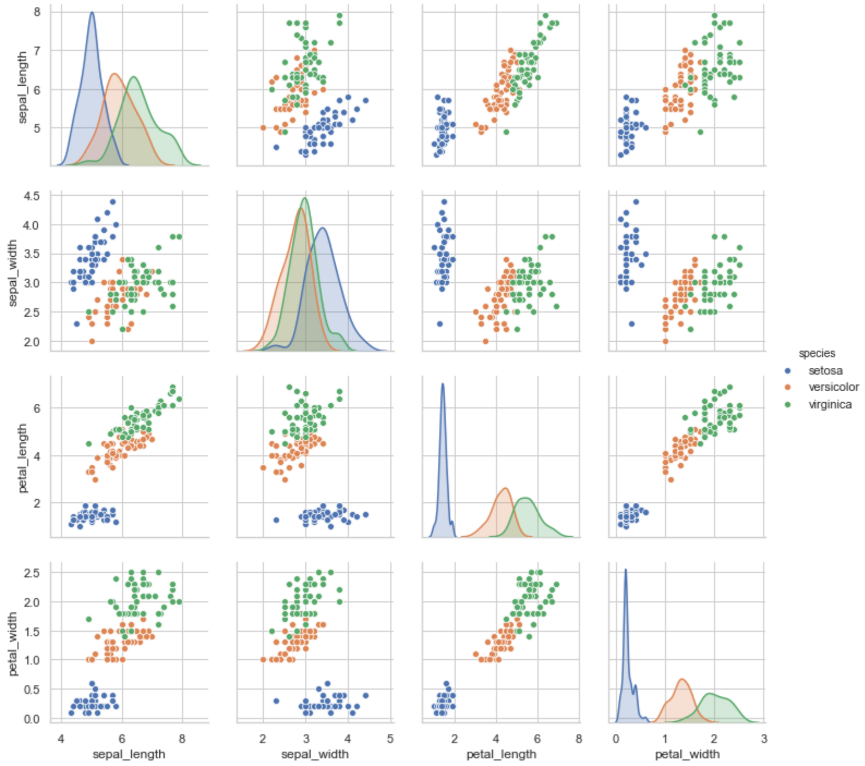
Seaborn provides functions to create figures that contain multiple subplots, which are useful for comparing different subsets of data.

- **pairplot():** Plots pairwise relationships in a dataset. By default, this function will create a grid of Axes such that each variable in the data will be shared across the y-axes across a single row and the x-axes across a single column.
- **FacetGrid, PairGrid, JointGrid:** These are classes that allow for the creation of multi-plot grids for conditioning on different variables.

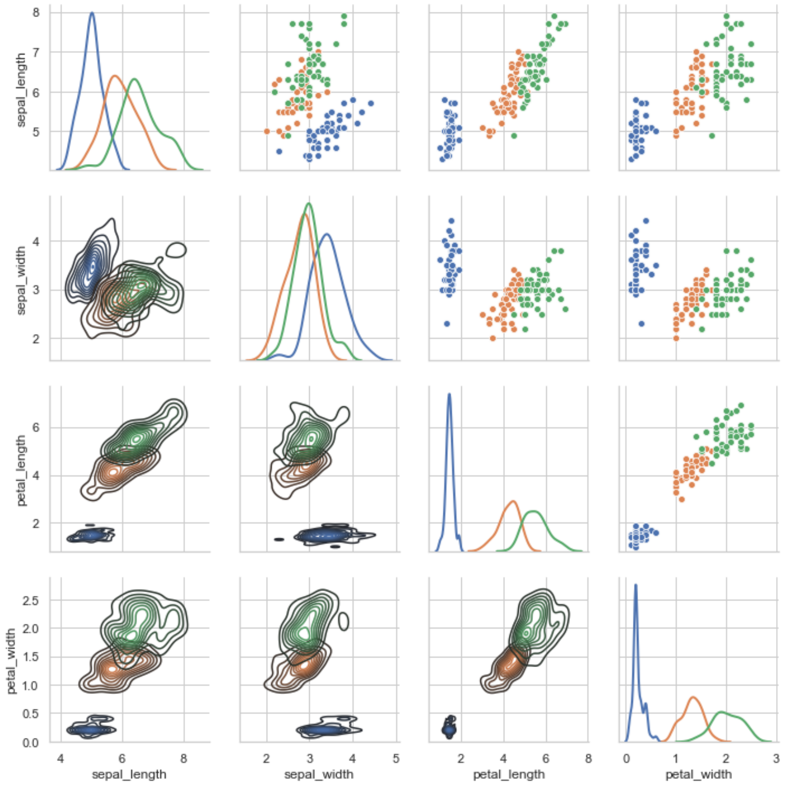
Seaborn

- PairGrid was created to eliminate the inconveniences of PairPlot, and the biggest difference is that each visualization tool can be applied to the top and bottom of the diagonal line.

pairplot()



pairgrid()



Lab: Titanic Dataset

```
# import packages
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from matplotlib import rcParams
import seaborn as sns
import warnings



# ignore warnings
warnings.filterwarnings('ignore')
```

Lab: Titanic Dataset

```
### Data Load (titanic) ###
df_titanic = sns.load_dataset('titanic')


### Understanding data ###


print(df_titanic.head())
print(df_titanic.info())
print(df_titanic.describe())
```

```
C:\Users\user\anaconda3\envs\Class_MDA\python.exe
C:\Users\user\PycharmProjects\MDA\MDA_02.py
  survived  pclass    sex   age  ... deck  embark_town  alive  alone
0        0       3    male  22.0  ...  NaN  Southampton    no  False
1        1       1  female  38.0  ...    C    Cherbourg   yes  False
2        1       3  female  26.0  ...  NaN  Southampton   yes   True
3        1       1  female  35.0  ...    C  Southampton   yes  False
4    0       3    male  35.0  ...  NaN  Southampton    no   True
[5 rows x 15 columns]
```

Lab: Titanic Dataset

```
<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)
```

Lab: Titanic Dataset

```
<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)
```

1. **survived** Survival or not? 0 means death, 1 means survival.
2. **pclass** Room class 1 is grade 1, 2 is grade 2, 3 is grade 3
3. **sex** gender Male is male, female is female.
4. **age** age
5. **sibsp** Number of siblings and spouses traveling together
6. **parch** Number of children and parents traveling together
7. **Fare** fare
8. **embarked** First letter of boarding destination name C is for Cherbourg, Q is Queenstown, S is Southampton
9. **class** Room class First grade 1st grade, Second grade 2nd grade, Third grade 3rd grade
10. **who** man, woman, child
11. **adult_male** Whether you are an adult male If True, it is an adult male, if False, it is anyone else.
12. **deck** Cabin number first alphabet A, B, C, D, E, F, G
13. **embark_town** boarding point name Cherbourg, Queenstown, Southampton
14. **alive** Survival If no, you die, if yes, you survive.
15. **alone** Whether you boarded alone If True, board alone, if False, board with family.

Lab: Titanic Dataset

```
### Data Load (titanic) ###
df_titanic = sns.load_dataset('titanic')


### Understanding data ###


print(df_titanic.head())
print(df_titanic.info())
print(df_titanic.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
```

Lab: Titanic Dataset

```
# DataFrame[column name].value_counts()
print(df_titanic['survived'].value_counts())          # 0:Dead, 1:survived
print(df_titanic['who'].value_counts())               # man, woman, child
print(df_titanic['class'].value_counts())             # First, second, third
print(df_titanic['adult_male'].value_counts())        # True, False
```

```
survived
0    549
1    342
Name: count, dtype: int64

who
man      537
woman    271
child     83
Name: count, dtype: int64
```

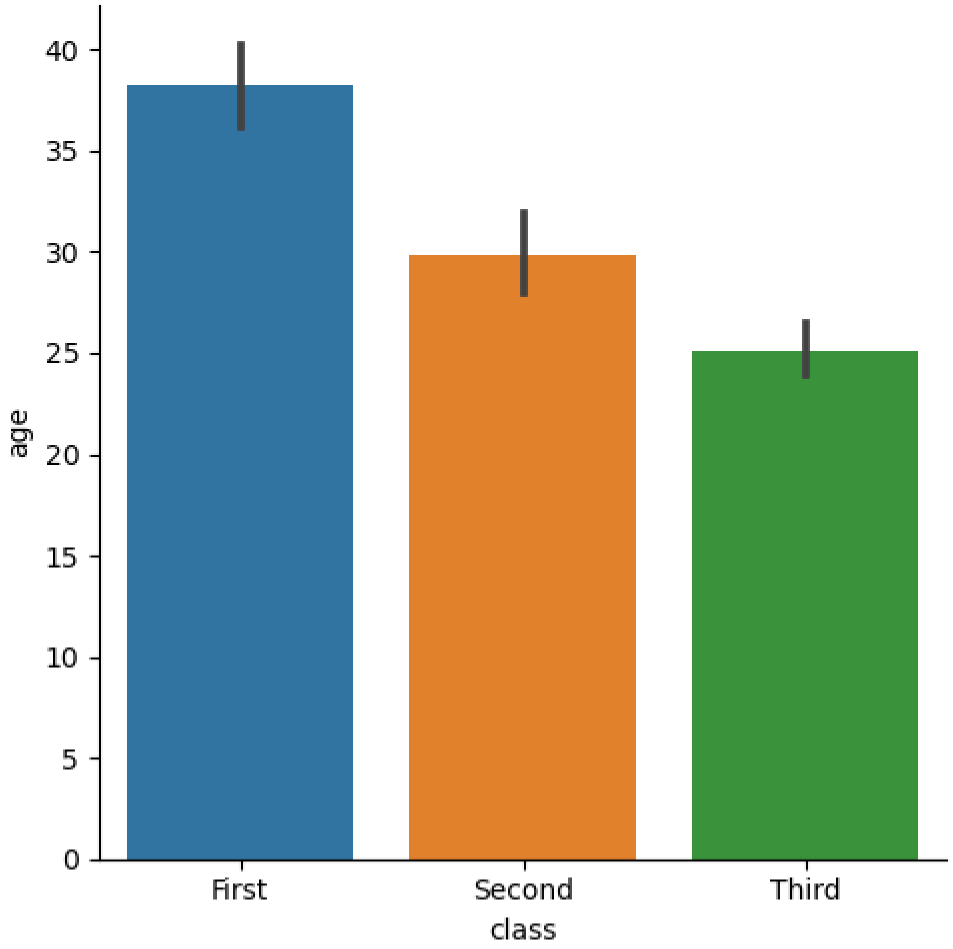
```
class
Third     491
First     216
Second    184
Name: count, dtype: int64

adult_male
True     537
False    354
Name: count, dtype: int64
```

Lab: Titanic Dataset

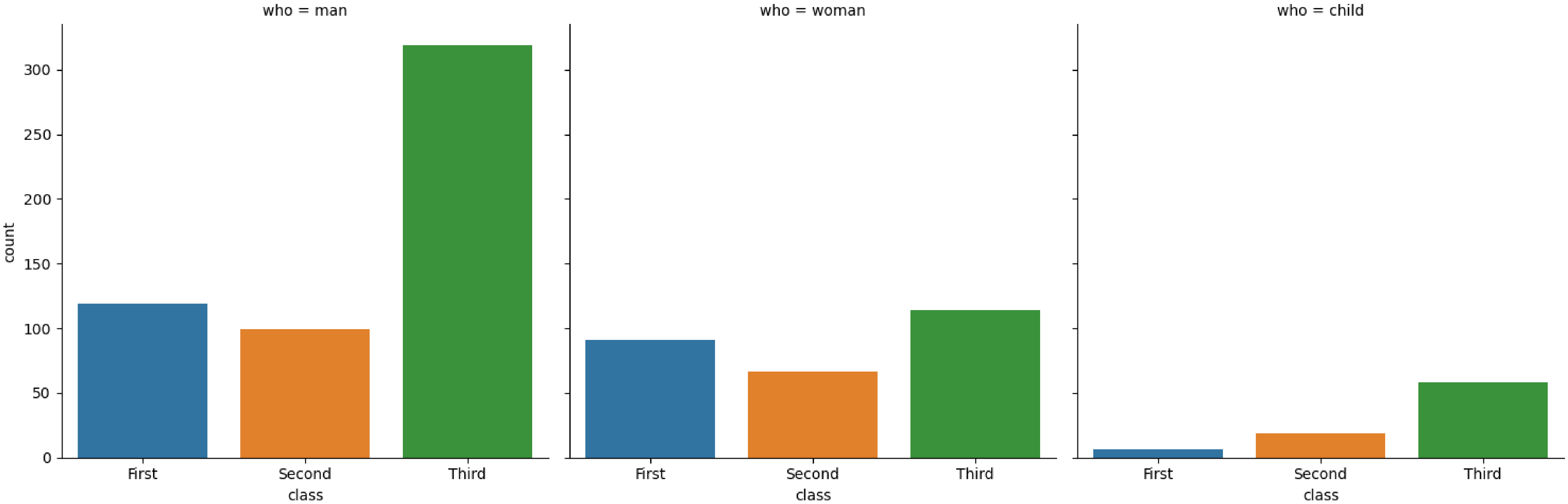
```
########## Visualization ##########
# barplot
sns.catplot(x='class',
            y='age',
            kind='bar',
            data=df_titanic)
```

The average is expressed as the height of the bar, and the deviation is expressed as an error bar.



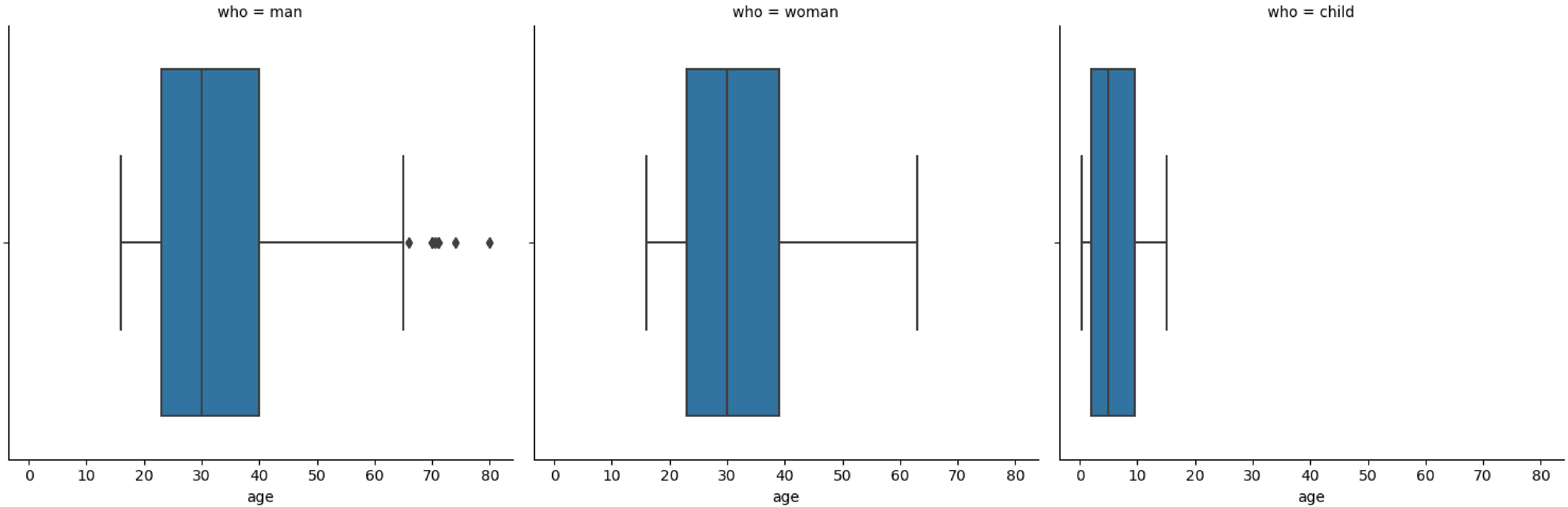
Lab: Titanic Dataset

```
########## Visualization ##########
# countplot
sns.catplot(x='class', col='who', kind='count', data=df_titanic)
```



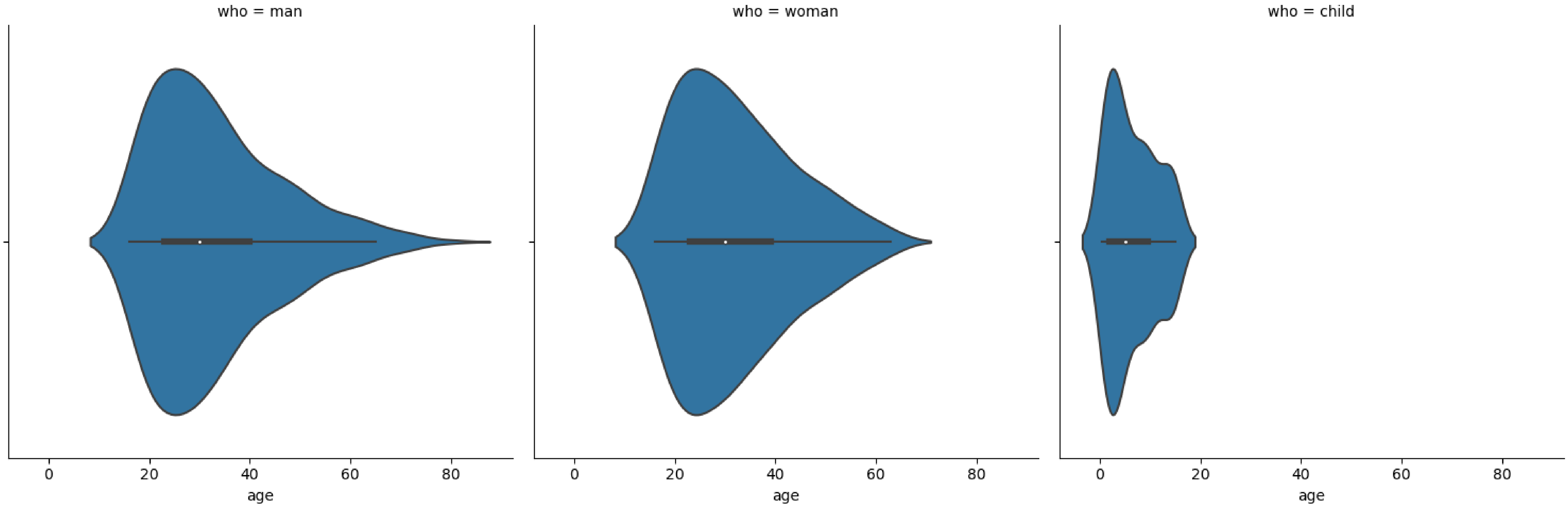
Lab: Titanic Dataset

```
########## Visualization ##########
# boxplot
sns.catplot(x='age’, col='who', kind='box', data=df_titanic)
```



Lab: Titanic Dataset

```
########## Visualization ##########
# violinplot
sns.catplot(x='age', col='who', kind='violin', data=df_titanic)
```



Bye Bye