**Ten Must-Know Seaborn Plots**

[[Sneha Bajaj](https://medium.com/@snehabajaj108?source=post_page-----1f3a82dc99c5--------------------------------)](https://medium.com/@snehabajaj108?source=post_page-----1f3a82dc99c5--------------------------------)

[Sneha Bajaj](https://medium.com/@snehabajaj108?source=post_page-----1f3a82dc99c5--------------------------------)

·

[Follow](https://medium.com/m/signin?actionUrl=https%3A%2F%2Fmedium.com%2F_%2Fsubscribe%2Fuser%2Fb235560f9346&operation=register&redirect=https%3A%2F%2Fmedium.com%2F%40snehabajaj108%2Ften-must-know-seaborn-plots-1f3a82dc99c5&user=Sneha+Bajaj&userId=b235560f9346&source=post_page-b235560f9346----1f3a82dc99c5---------------------post_header-----------)

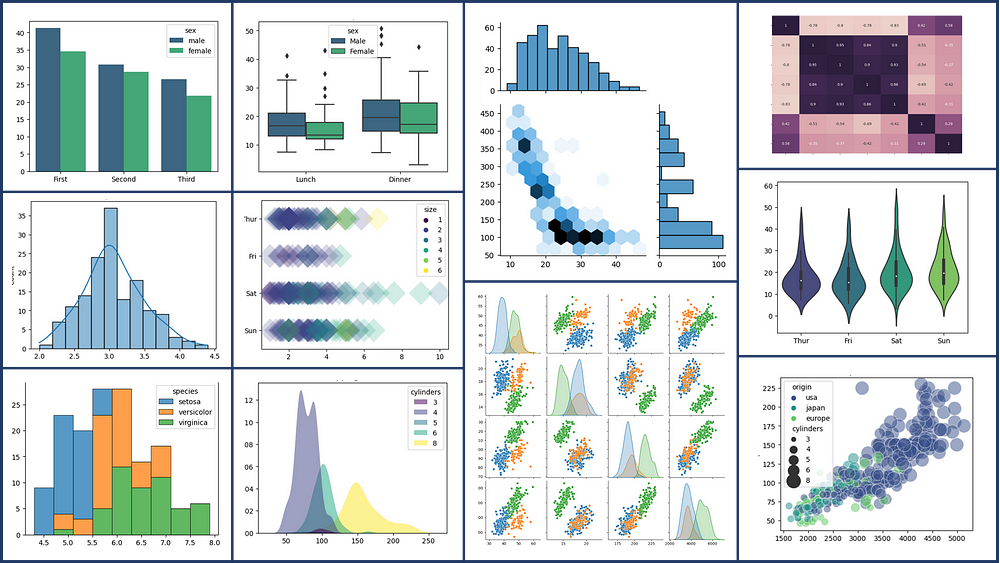
9 min read

·

Dec 29, 2023

20

For anyone who wants to ace visualisation using Python, here are some commonly used plots with explanation of use-cases and code examples



Plots by Author using Seaborn

Before we begin, lets import the following python packages required to process tabular data and create visualisations. We will also be importing a few datasets from the seaborn library:

#Import Libraries  
import pandas as pd  
import numpy as np   
import seaborn as sns  
import matplotlib  
import matplotlib.pyplot as plt  
%matplotlib inline  
  
#Import Datasets  
exercise = sns.load\_dataset('exercise')  
iris = sns.load\_dataset('iris')  
penguins = sns.load\_dataset('penguins')  
mpg = sns.load\_dataset('mpg')  
titanic = sns.load\_dataset('titanic')  
tips = sns.load\_dataset('tips')

The ten visualisations discussed in this article are indexed below for your convenience:

[1. Bar Plots](https://medium.com/@snehabajaj108/ten-must-know-seaborn-plots-1f3a82dc99c5#0333)  
[2. Count Plots](https://medium.com/@snehabajaj108/ten-must-know-seaborn-plots-1f3a82dc99c5#65a6)  
[3. Histograms](https://medium.com/@snehabajaj108/ten-must-know-seaborn-plots-1f3a82dc99c5#406f)  
[4. Cat Plots (Box, Violin, Swarm, Boxen)](https://medium.com/@snehabajaj108/ten-must-know-seaborn-plots-1f3a82dc99c5#7220)  
[5. Multiple Plots using FacetGrid](https://medium.com/@snehabajaj108/ten-must-know-seaborn-plots-1f3a82dc99c5#d490)  
[6. Joint Plots](https://medium.com/@snehabajaj108/ten-must-know-seaborn-plots-1f3a82dc99c5#1faf)  
[7. KDE Plots](https://medium.com/@snehabajaj108/ten-must-know-seaborn-plots-1f3a82dc99c5#6669)  
[8. Pairplots](https://medium.com/@snehabajaj108/ten-must-know-seaborn-plots-1f3a82dc99c5#3cda)  
[9. Heatmaps](https://medium.com/@snehabajaj108/ten-must-know-seaborn-plots-1f3a82dc99c5#c733)  
[10. Scatter Plots](https://medium.com/@snehabajaj108/ten-must-know-seaborn-plots-1f3a82dc99c5#1a1f)

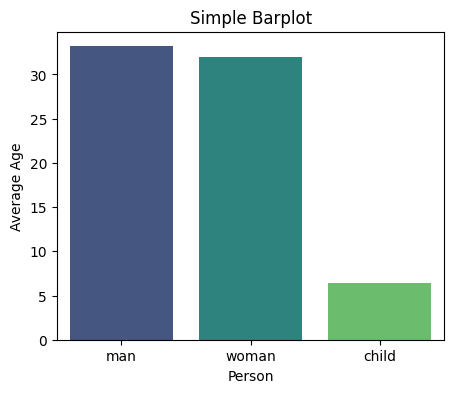
**1. Bar Plots**

Bar plots can be used to visualize various types of data, such as counts, frequencies, percentages, or averages. They are particularly useful for displaying and comparing data from different categories.

**Use Cases:**

1. *Categorical Comparison:*Each bar represents a distinct category, and the height of the bar indicates the aggregated value associated with that category (count, sum or mean).  
   For example, *average age of titanic passengers by gender:*

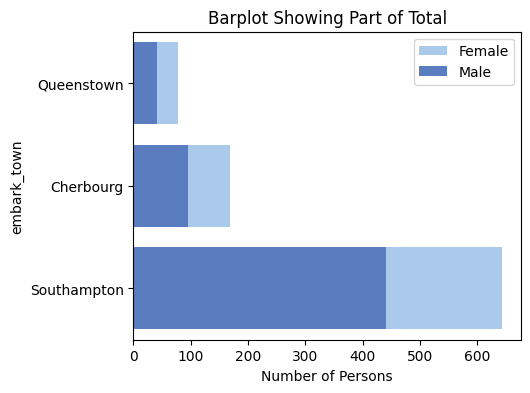
# Simple barplot  
sns.barplot(data=titanic, x="who", y="age", estimator='mean',   
 errorbar=None, palette='viridis')  
plt.title('Simple Barplot')  
plt.xlabel('Person')  
plt.ylabel('Average Age')  
plt.show();



Plot by Author using Seaborn

*2. Proportional Representation through Stacked Bar Charts:* Bar plots can also represent proportions or percentages. By scaling the height of each bar to represent the proportion of observations in a category, it is possible to compare the relative distribution of different categories.  
For example, *proportion of males from different towns on the Titanic:*

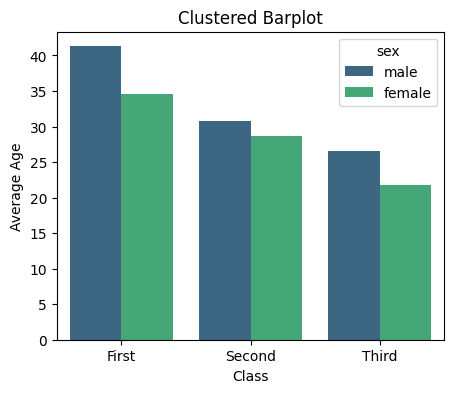
#Prepare data for next plot  
data = titanic.groupby('embark\_town').agg({'who':'count','sex': lambda x: (x=='male').sum()}).reset\_index()  
data.rename(columns={'who':'total', 'sex':'male'}, inplace=True)  
data.sort\_values('total', inplace=True)  
  
# Barplot Showing Part of Total  
sns.set\_color\_codes("pastel")  
sns.barplot(x="total", y="embark\_town", data=data,  
 label="Female", color="b")   
sns.set\_color\_codes("muted")  
sns.barplot(x="male", y="embark\_town", data=data,  
 label="Male", color="b")   
plt.title('Barplot Showing Part of Total')  
plt.xlabel('Number of Persons')  
plt.legend(loc='upper right')  
plt.show()



Plot by Author using Seaborn

*3. Comparison of Subcategories within each category through Clustered Bar Plots:* Multiple bars can be grouped within each category to represent different subcategories, allowing for comparison and analysis.  
For example, *what is the average age of males and females within each class?*

# Clustered barplot  
sns.barplot(data=titanic, x='class', y='age', hue='sex',   
 estimator='mean', errorbar=None, palette='viridis')  
plt.title('Clustered Barplot')  
plt.xlabel('Class')  
plt.ylabel('Average Age')  
plt.show();



Plot by Author using Seaborn

**For more on bar plots, Seaborn Documentation**: [https://seaborn.pydata.org/generated/seaborn.barplot.html](https://jovian.com/outlink?url=https%3A%2F%2Fseaborn.pydata.org%2Fgenerated%2Fseaborn.barplot.html)

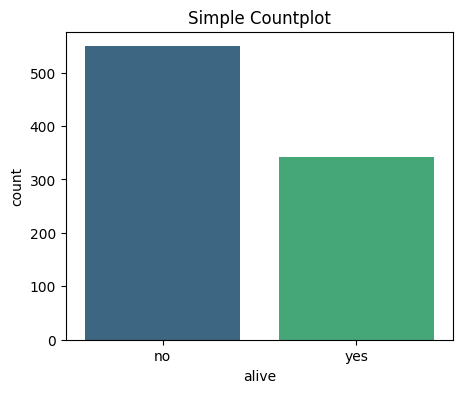
**2. Count Plots**

A count plot displays the number of occurrences of each category in a categorical variable. The x-axis represents the categories of the variable, while the y-axis represents the count or frequency of each category.

**Use Cases:**

* *Frequency Distribution of categorical variables:* Each bar represents a category, and the height of the bar represents the frequency or count of observations in that category. This helps to identify the most common or least common categories.  
  For example, *the status of passengers on the titanic*

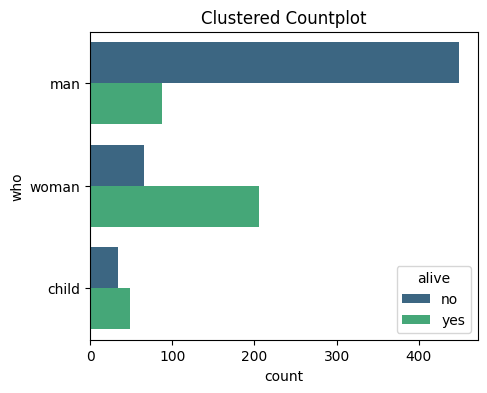
# Simple Countplot  
sns.countplot(data=titanic, x='alive', palette='viridis')  
plt.title('Simple Countplot')  
plt.show();



Plot by Author using Seaborn

* Relationship between different categorical variables  
  For example, *the status of passengers by gender on the titanic*

# Clustered Countplot  
sns.countplot(data=titanic, y="who",   
 hue="alive", palette='viridis')  
plt.title('Clustered Countplot')  
plt.show();



Plot by Author using Seaborn

**Seaborn Documentation**: [https://seaborn.pydata.org/generated/seaborn.countplot.html](https://jovian.com/outlink?url=https%3A%2F%2Fseaborn.pydata.org%2Fgenerated%2Fseaborn.countplot.html)

**3. Histograms**

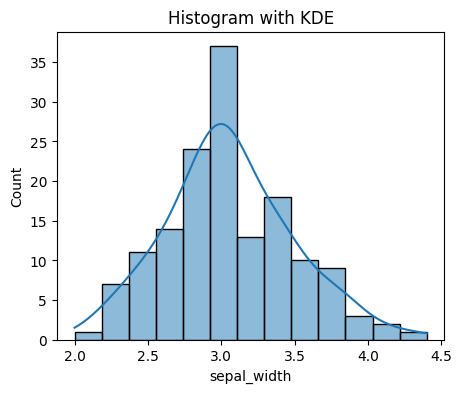
Histograms are graphical representations of the distribution of a dataset. They can reveal important characteristics of the data, such as whether it follows a normal distribution, is skewed to one side, or has multiple peaks. They display the frequency or count of observations within different intervals or “bins” of the data.

The x-axis of a histogram represents the range of values in the dataset, divided into equally spaced intervals or bins. The y-axis represents the frequency or count of observations falling within each bin. The height of each bar in the histogram corresponds to the number of observations in that bin.

**Use Cases**:

1. Visualize the shape, centre, range and spread of a continuous/numeric variable and to identify any patterns or outliers.  
   For example, *distribution of the width of flower petals*

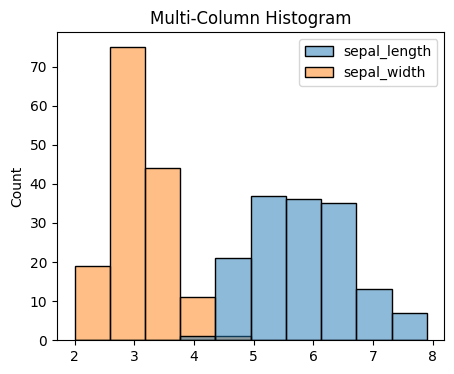
# Histogram with KDE  
sns.histplot(data=iris, x='sepal\_width', kde=True)  
plt.title('Histogram with KDE')  
plt.show();



Plot by Author using Seaborn

2. Compare the distribution of many continuous variables  
For example, *Distribution of petal length and sepal length for flowers*

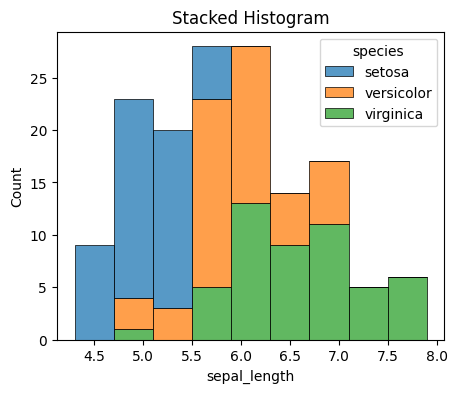
# Histogram with multiple features  
sns.histplot(data=iris[['sepal\_length','sepal\_width']])  
plt.title('Multi-Column Histogram')  
plt.show();



Plot by Author using Seaborn

3. Compare the distribution of a continuous variable for different categories  
For example, *distribution of petal length for different species of flowers*

#Stacked Histogram  
sns.histplot(iris, x='sepal\_length', hue='species', multiple='stack',  
 linewidth=0.5)  
plt.title('Stacked Histogram')  
plt.show()



Plot by Author using Seaborn

**Seaborn Documentation**: [https://seaborn.pydata.org/generated/seaborn.histplot.html](https://jovian.com/outlink?url=https%3A%2F%2Fseaborn.pydata.org%2Fgenerated%2Fseaborn.histplot.html)

**4. Cat Plots (Box, Violin, Swarm, Boxen)**

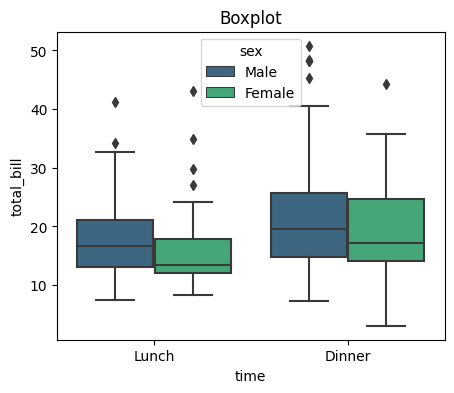
Catplot is a higher-level versatile function that combines several underlying categorical seaborn plots like boxplots, violinplots, swarmplots, pointplots, barplots and countplots.

**Use Cases**:

* Explore relationship between categorical and a continuous variables
* Get the statistical summary of a continuous variable

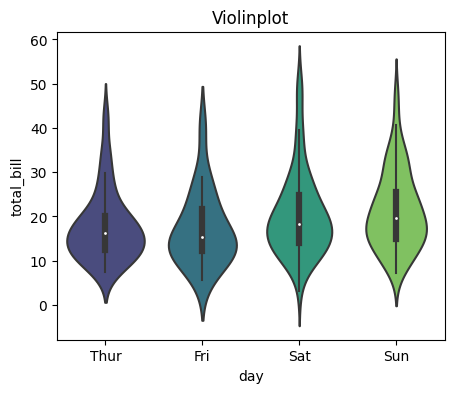
Examples:

# Boxplot  
sns.boxplot(data=tips, x='time', y='total\_bill', hue='sex', palette='viridis')  
plt.title('Boxplot')  
plt.show()



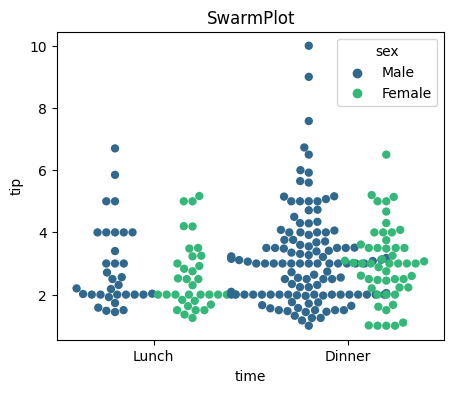
Plot by Author using Seaborn

# Violinplot  
sns.violinplot(data=tips, x='day', y='total\_bill', palette='viridis')  
plt.title('Violinplot')  
plt.show()



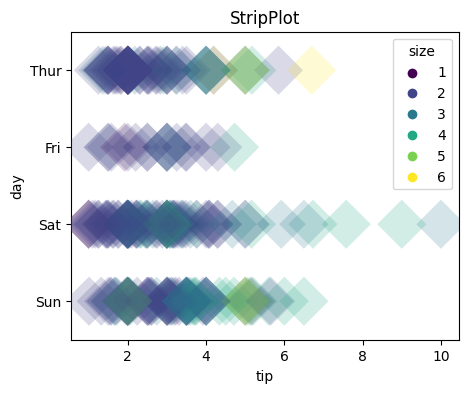
Plot by Author using Seaborn

#Swarmplot  
sns.swarmplot(data=tips, x='time', y='tip', dodge=True, palette='viridis', hue='sex', s=6)  
plt.title('SwarmPlot')  
plt.show()



Plot by Author using Seaborn

#StripPlot  
sns.stripplot(data=tips, x='tip', hue='size', y='day', s=25, alpha=0.2,  
 jitter=False, marker='D',palette='viridis')  
plt.title('StripPlot')  
plt.show()



Plot by Author using Seaborn

**Seaborn Documentation**: [https://seaborn.pydata.org/generated/seaborn.catplot.html](https://jovian.com/outlink?url=https%3A%2F%2Fseaborn.pydata.org%2Fgenerated%2Fseaborn.catplot.html)

**5. Multiple Plots using FacetGrid**

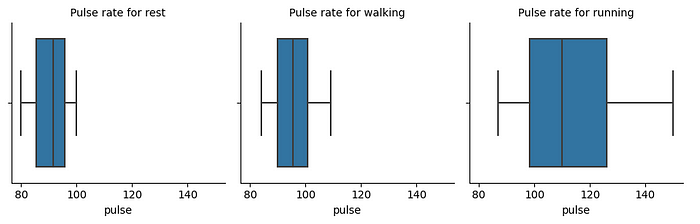
FacetGrid is a feature in the seaborn library that allows you to create multiple subsets of your data in a grid-like arrangement. You can create a grid of plots where each plot represents a category. The subsets are determined by the column names given in the ‘col’ and ‘row’ attribute of FacetGrid(). The individual plots within the grid can be any type of plot supported by seaborn, such as scatter plots, line plots, bar plots, or histograms.

**Use Cases:**

* Compare and analyse different groups or categories within your dataset
* Create subplots seamlessly

For example, *Boxplots for pulse rate during different activities:*

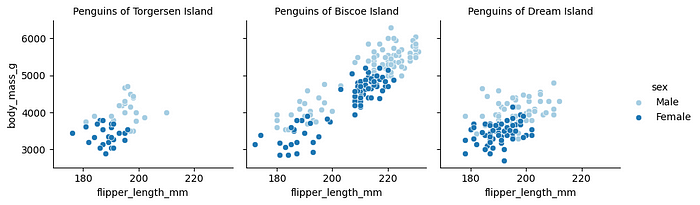
# Creating subplots using FacetGrid  
g = sns.FacetGrid(exercise, col='kind', palette='Paired')  
  
# Drawing a plot on every facet  
g.map(sns.boxplot, 'pulse')  
g.set\_titles(col\_template="Pulse rate for {col\_name}")  
g.add\_legend();



Plot by Author using Seaborn

*Scatter plots for flipper length and body mass of Penguins from different islands*

# Creating subplots using FacetGrid  
g = sns.FacetGrid(penguins, col='island',hue='sex', palette='Paired')  
  
# Drawing a plot on every facet  
g.map(sns.scatterplot, 'flipper\_length\_mm', 'body\_mass\_g')  
g.set\_titles(template="Penguins of {col\_name} Island")  
g.add\_legend();



Plot by Author using Seaborn

**Seaborn Documentation**: [https://seaborn.pydata.org/generated/seaborn.FacetGrid.html](https://jovian.com/outlink?url=https%3A%2F%2Fseaborn.pydata.org%2Fgenerated%2Fseaborn.FacetGrid.html)

**6. Joint Plots**

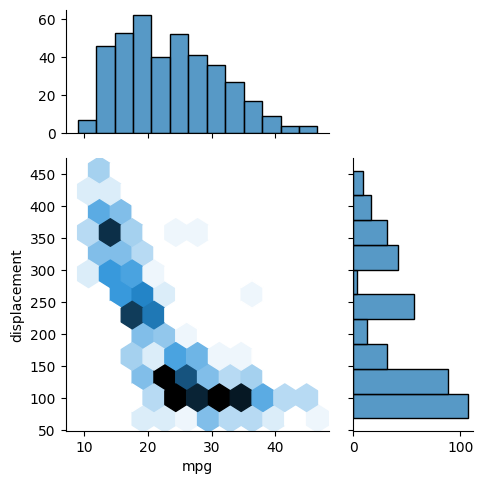
A joint plot combines multiple univariate and bivariate plots in a single figure. The central plot typically displays a scatter plot or a hexbin plot, representing the joint distribution of the two variables. This main plot is accompanied by additional plots along the axes (histograms or KDEs) that show the distributions of each variable individually.

**Use Cases:**

* Finding the relationship between 2 variables
* Comparing the individual distributions of 2 different variables

For example, *comparison of the displacement and mpg for cars*

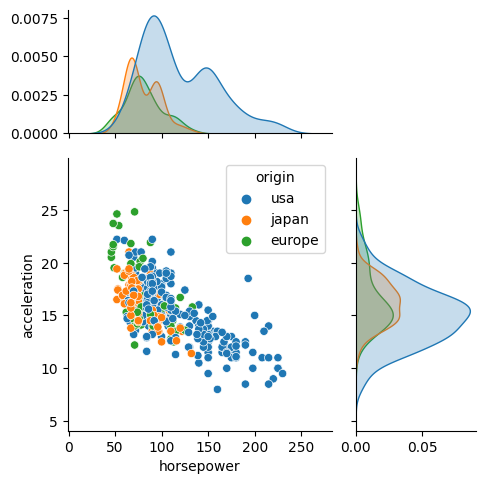
# Hex Plot with Histogram margins  
sns.jointplot(x="mpg", y="displacement", data=mpg,  
 height=5, kind='hex', ratio=2, marginal\_ticks=True);



Plot by Author using Seaborn

*Comparison of acceleration and horsepower for cars from different countries*

# Scatter Plot with KDE Margins  
sns.jointplot(x="horsepower", y="acceleration", data=mpg,  
 hue="origin", height=5, ratio=2, marginal\_ticks=True);



Plot by Author using Seaborn

**Seaborn Documentation:** [https://seaborn.pydata.org/generated/seaborn.jointplot.html](https://jovian.com/outlink?url=https%3A%2F%2Fseaborn.pydata.org%2Fgenerated%2Fseaborn.jointplot.html)

**7. KDE Plots**

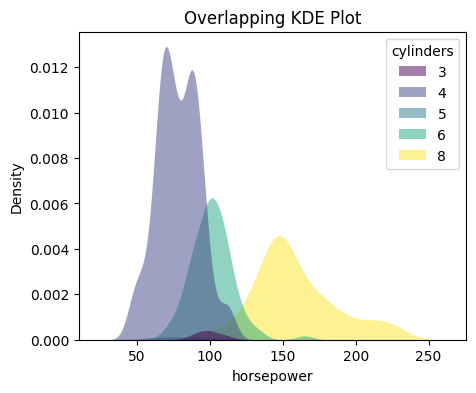
A KDE (Kernel Density Estimate) plot is a smoothed version of a histogram representing the probability density function of a continuous random variable. The y-axis represents the density or likelihood of observing a particular value of the variable, and the x-axis represents the values of the variable itself.

**Use cases:**

* visualization of the distribution of a single variable (univariate analysis)
* Insights into the shape, peaks, and skewness of the distribution.

For example, *comparing the horsepower of cars with respect to number of cylinders*

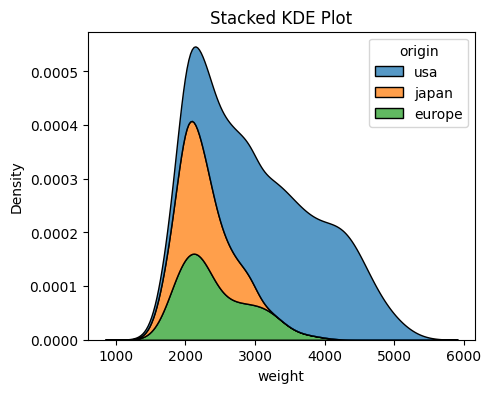
#Overlapping KDE Plots  
sns.kdeplot(data=mpg, x='horsepower', hue='cylinders', fill=True,  
 palette='viridis', alpha=.5, linewidth=0)  
plt.title('Overlapping KDE Plot')  
plt.show();



Plot by Author using Seaborn

*Comparing the weight of cars across different countries:*

#Stacked KDE Plots  
sns.kdeplot(data=mpg, x="weight", hue="origin", multiple="stack")  
plt.title('Stacked KDE Plot')  
plt.show();



Plot by Author using Seaborn

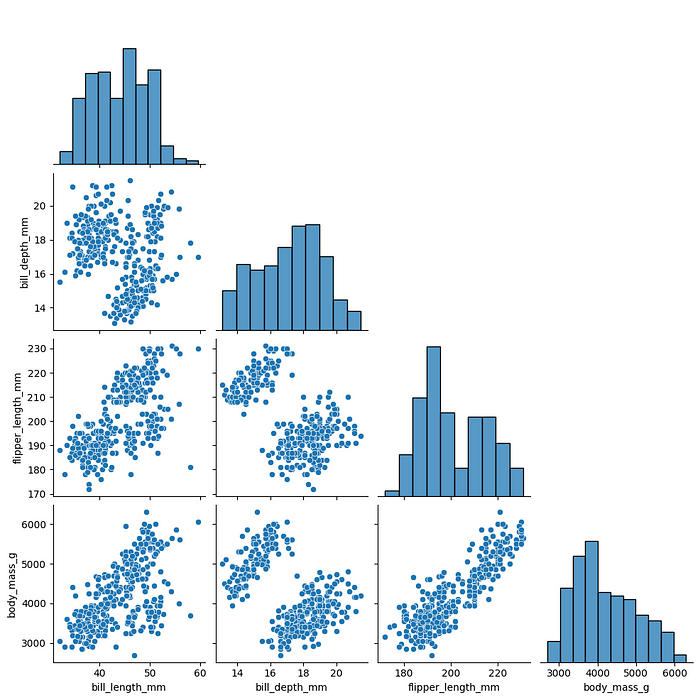
**Seaborn Documentation:** [https://seaborn.pydata.org/generated/seaborn.kdeplot.html](https://jovian.com/outlink?url=https%3A%2F%2Fseaborn.pydata.org%2Fgenerated%2Fseaborn.kdeplot.html)

**8. Pairplots**

A pair plot is a type of visualization that allows you to explore the relationships between multiple variables in a dataset. It is a grid of scatter plots, where each variable is plotted against every other variable. In a pair plot, the diagonal entries are histograms or density plots for each variable, showing the distribution of values.

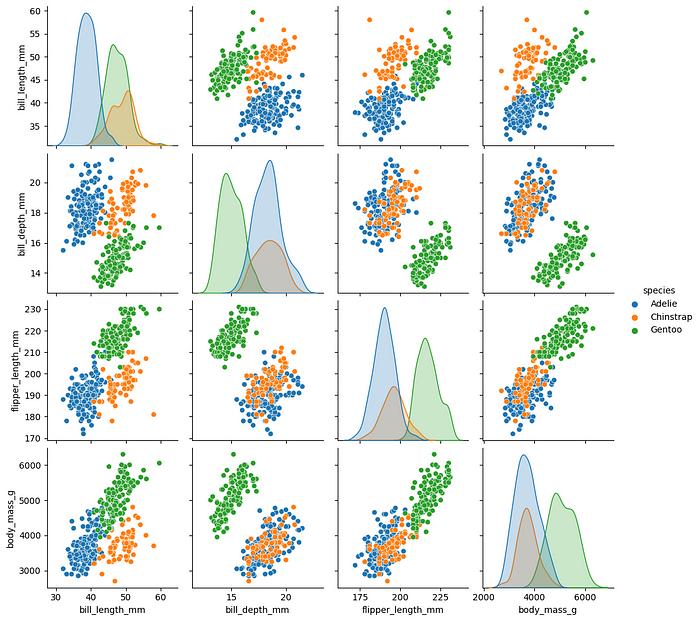
**Use Cases:**  
Identification of correlations or patterns between variables, such as linear or non-linear relationships, clusters, or outliers.  
For example, *visualisation of relationship between different features of penguins*

#Simple Pairplot  
sns.pairplot(data=penguins, corner=True);



Plot by Author using Seaborn

# Pairplot with hues  
sns.pairplot(data=penguins, hue='species');



Plot by Author using Seaborn

It can be observed that introducing hue in the plot helps us identify the crucial differences between different species of penguins.

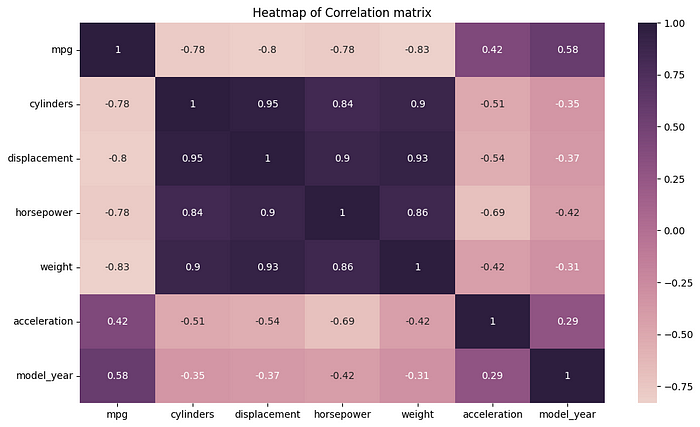
**Seaborn Documentation**: [https://seaborn.pydata.org/generated/seaborn.pairplot.html](https://jovian.com/outlink?url=https%3A%2F%2Fseaborn.pydata.org%2Fgenerated%2Fseaborn.pairplot.html)

**9. Heatmaps**

Heatmaps are a type of visual representation that use color-coded cells to display the values of a matrix or a table of data. In a heatmap, the rows and columns of the matrix represent two different variables, and the color intensity of each cell represents the value or magnitude of the data point at the intersection of those variables.

**Use Cases:**  
Correlation analysis, visualisation of pivot tables which aggragate data by rows and columns  
For example, *visualisation of the correlation between all the numerical columns of the mpg dataset:*

#Selection of numeric columns from the dataset  
num\_cols = list(mpg.select\_dtypes(include='number'))  
fig = plt.figure(figsize=(12,7))  
  
#Correlation Heatmap  
sns.heatmap(data=mpg[num\_cols].corr(),   
 annot=True, cmap=sns.cubehelix\_palette(as\_cmap=True))  
plt.title('Heatmap of Correlation matrix')  
plt.show()



Plot by Author using Seaborn

**Seaborn Documentation**: [https://seaborn.pydata.org/generated/seaborn.heatmap.html](https://jovian.com/outlink?url=https%3A%2F%2Fseaborn.pydata.org%2Fgenerated%2Fseaborn.heatmap.html)

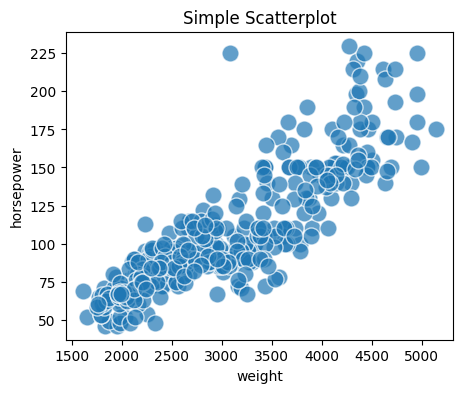
**10. Scatter Plots**

A scatterplot displays the relationship between two continuous variables. It is constructed by plotting individual data points on a graph with one variable represented on the x-axis and the other variable represented on the y-axis. The resulting plot consists of multiple points scattered across the graph, hence the name “scatterplot.”

**Use Cases:**

1. *Relationship Assessment:*Scatterplots help determine the nature of the relationship between two continuous variables. It can reveal if there is a positive correlation (both variables increase or decrease together), negative correlation (one variable increases while the other decreases), or no correlation (no apparent relationship).  
   For example, *it can be observed from the scatterplot below that the horsepower and weight of cars is positively correlated*

# Simple Scatterplot  
sns.scatterplot(data=mpg, x='weight', y='horsepower', s=150, alpha=0.7)  
plt.title('Simple Scatterplot')  
plt.show();

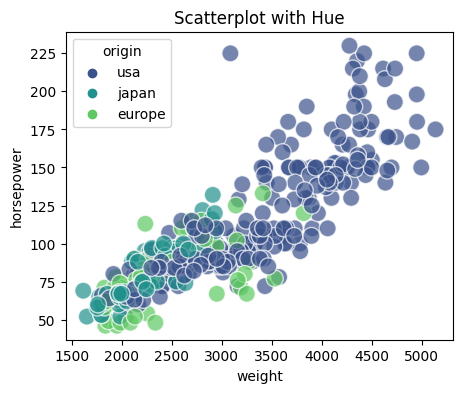


Plot by Author using Seaborn

2. *Outlier Identification:*Scatterplots can highlight outliers, which are data points that deviate significantly from the overall pattern.

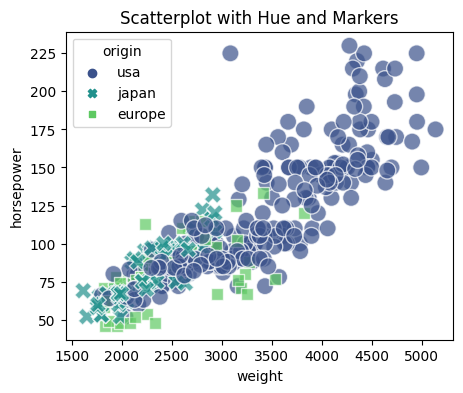
3. *Clustering and Grouping:* By visually examining the distribution of points, you can identify if there are natural groupings or patterns among the variables.  
For example, *comparing the horsepower and weight of cars from different countries*

# Scatterplot with Hue  
sns.scatterplot(data=mpg, x='weight', y='horsepower', s=150, alpha=0.7,  
 hue='origin', palette='viridis')  
plt.title('Scatterplot with Hue')  
plt.show()



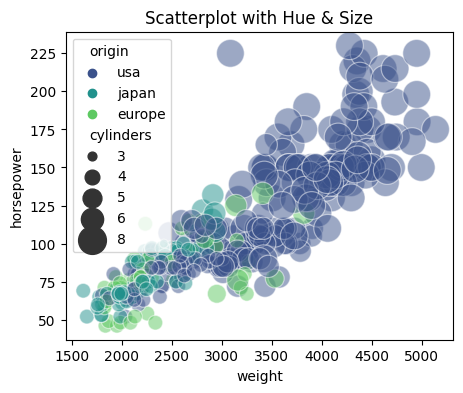
Plot by Author using Seaborn

# Scatterplot with Hue and Markers  
sns.scatterplot(data=mpg, x='weight', y='horsepower', s=150, alpha=0.7,  
 style='origin',palette='viridis', hue='origin')  
plt.title('Scatterplot with Hue and Markers')  
plt.show()



Plot by Author using Seaborn

# Scatterplot with Hue & Size  
sns.scatterplot(data=mpg, x='weight', y='horsepower', sizes=(40, 400), alpha=.5,  
 palette='viridis', hue='origin', size='cylinders')  
plt.title('Scatterplot with Hue & Size')  
plt.show()



Plot by Author using Seaborn

4. *Trend Analysis:*By plotting data points chronologically, scatterplots can depict the evolution or progression of variables, helping to identify trends or changes in behaviour.

5. *Model Validation:* By comparing the predicted values of a model to the actual values, scatterplots can visualize the accuracy or deviation of the model’s predictions.

**To know more about scatter plots, refer Seaborn Documentation:** [https://seaborn.pydata.org/generated/seaborn.scatterplot.html](https://jovian.com/outlink?url=https%3A%2F%2Fseaborn.pydata.org%2Fgenerated%2Fseaborn.scatterplot.html)

**Conclusion**

* This article summarises most of the commonly used visualisations in the Seaborn Library of Python.
* Refer Seaborn Gallery for more unique plots: [https://seaborn.pydata.org/examples/index.html](https://jovian.com/outlink?url=https%3A%2F%2Fseaborn.pydata.org%2Fexamples%2Findex.html)

***If you found this helpful, do add some claps and follow me for more on analytics, statistics and data science!***