1. Style and colour:

**set\_style()**method is used to set the aesthetic of the plot. It means it affects things like the color of the axes, whether the grid is active or not, or other aesthetic elements. There are five themes available in Seaborn.

* darkgrid
* whitegrid
* dark
* white
* ticks

Example:

1. Using set\_style():

**import** seaborn as sns

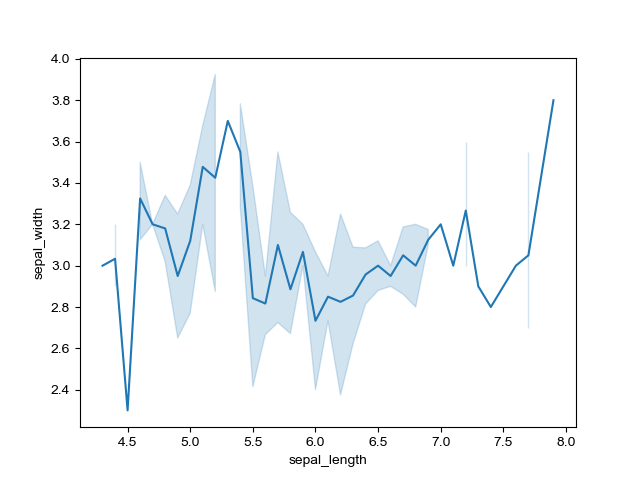
**import** matplotlib.pyplot as plt

data **=** sns.load\_dataset("iris")

sns.lineplot(x**=**"sepal\_length", y**=**"sepal\_width", data**=**data)

sns.set\_style("dark")

plt.show()



1. Using figure():

**import** seaborn as sns

**import** matplotlib.pyplot as plt

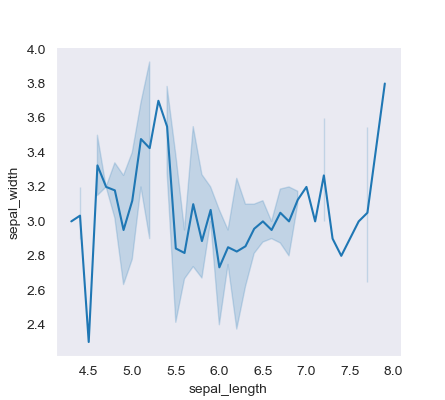
data **=** sns.load\_dataset("iris")

plt.figure(figsize **=** (2, 4))

sns.lineplot(x**=**"sepal\_length", y**=**"sepal\_width", data**=**data)

sns.despine()

plt.show()



1. Using set\_context():

**import** seaborn as sns

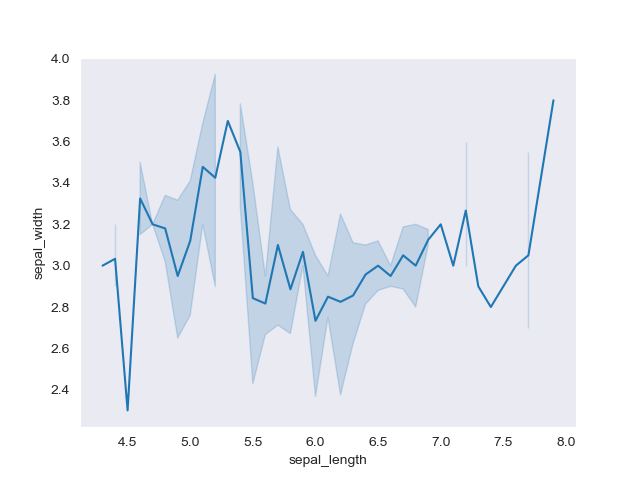
**import** matplotlib.pyplot as plt

data **=** sns.load\_dataset("iris")

sns.lineplot(x**=**"sepal\_length", y**=**"sepal\_width", data**=**data)

sns.set\_context("paper")

plt.show()



Colour: **color\_palette()**method is used to give colors to the plot. Another function **palplot()**is used to deal with the color palettes and plots the color palette as a horizontal array.

Example:

A)

**import** seaborn as sns

**import** matplotlib.pyplot as plt

palette **=** sns.color\_palette()

sns.palplot(palette)

plt.show()



B)

**import** seaborn as sns

**import** matplotlib.pyplot as plt

palette **=** sns.color\_palette('PiYG', 11)

sns.palplot(palette)

plt.show()



1. Distribution plot: Used for examining univariate and bivariate distributions.4 types of distribution plots namely:

A)Displot: It is used basically for univariant set of observations and visualizes it through a histogram

Example:

**import** seaborn as sns

**import** matplotlib.pyplot as plt

**from** warnings **import** filterwarnings

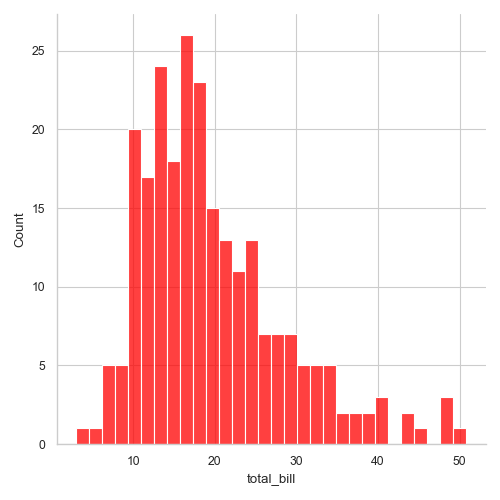
df **=** sns.load\_dataset('tips')

df.head()

sns.set\_style('whitegrid')

sns.displot(df['total\_bill'], kde **=** False, color **=**'red', bins **=** 30)

plt.show()



1. Joinplot:  It basically combines two different plots.

Example:

**import** seaborn as sns

**import** matplotlib.pyplot as plt **%** matplotlib inline

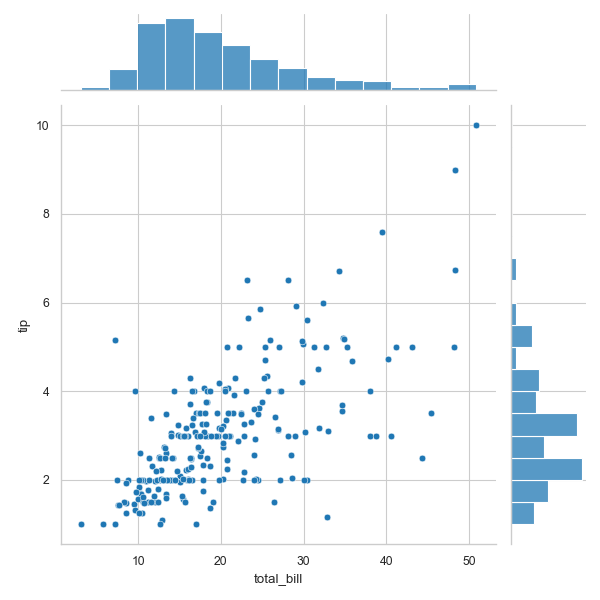
**from** warnings **import** filterwarnings

df **=** sns.load\_dataset('tips')

df.head()

sns.jointplot(x ='total\_bill', y ='tip', data = df)

plt.show()



1. Pairplot:It represents pairwise relation across the entire dataframe and supports an additional argument called **hue** for categorical separation.

Example:

**import** seaborn as sns

**import** matplotlib.pyplot as plt

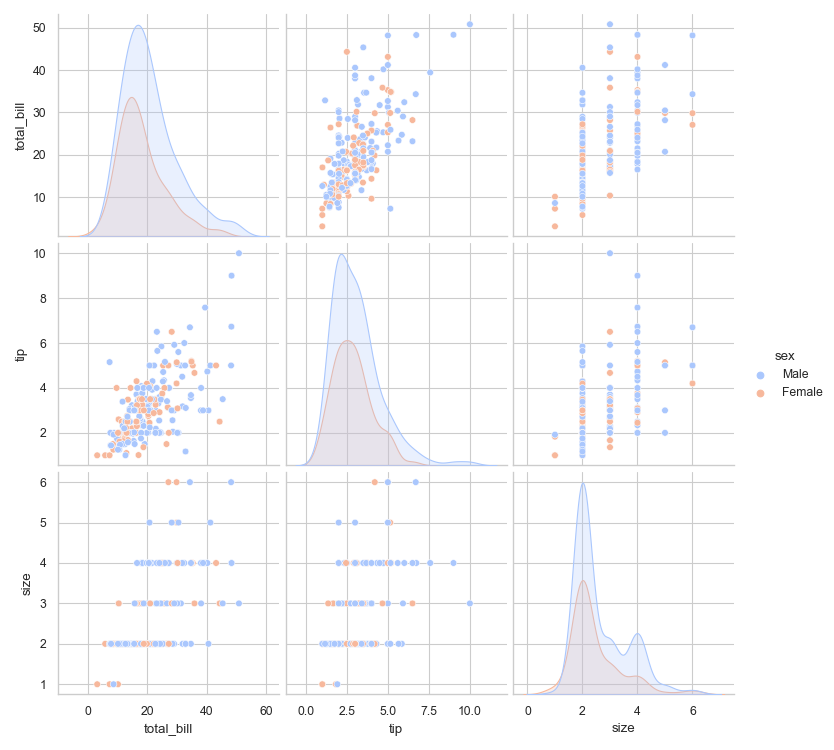
**from** warnings **import** filterwarnings

df **=** sns.load\_dataset('tips')

df.head()

sns.pairplot(df, hue ="sex", palette ='coolwarm')

plt.show()



1. Rugplot: It plots datapoints in an array as sticks on an axis.

Example:

**import** seaborn as sns

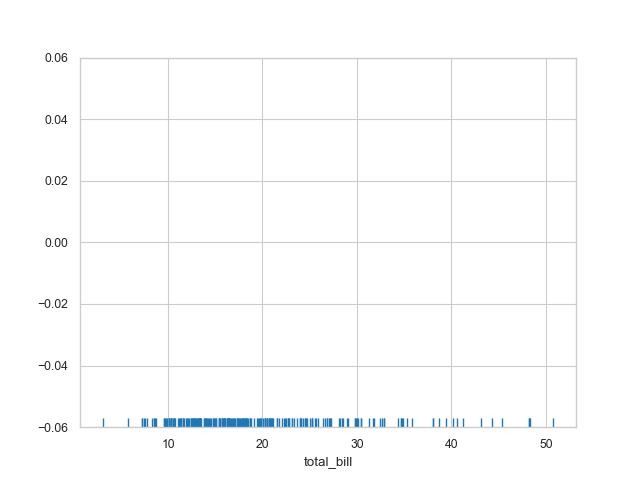
**import** matplotlib.pyplot as plt

**from** warnings **import** filterwarnings

df **=** sns.load\_dataset('tips')

sns.rugplot(df['total\_bill'])

plt.show()



1. Categorical plots:
2. Barplot:A **barplot** is basically used to aggregate the categorical data according to some methods and by default its the mean.

Example:

import seaborn as sns

from warnings import filterwarnings

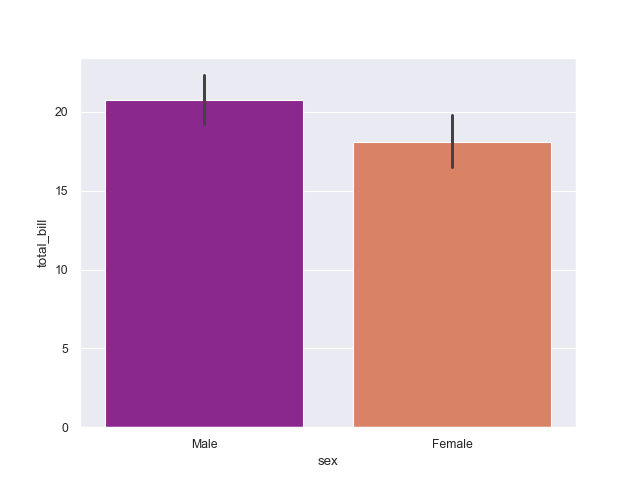
df = sns.load\_dataset('tips')

df.head()

sns.set\_style('darkgrid')

sns.barplot(x **=**'sex', y **=**'total\_bill', data **=** df, palette **=**'plasma')

plt.show()



1. Countplot: A countplot basically counts the categories and returns a count of their occurrences.

Example:

import seaborn as sns

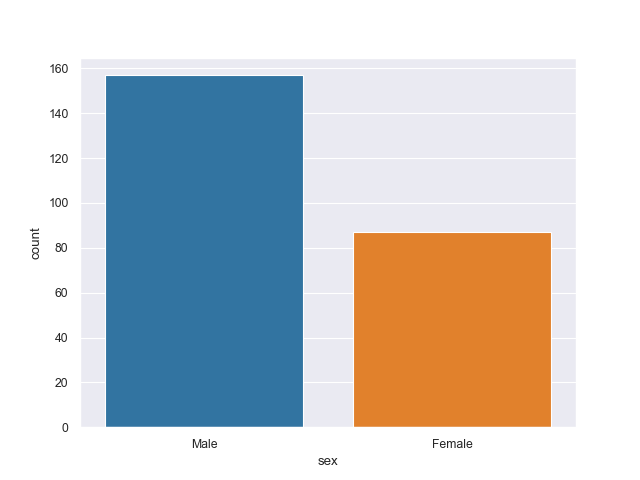
from warnings import filterwarnings

df = sns.load\_dataset('tips')

df.head()

sns.countplot(x ='sex', data = df)

plt.show()



1. Boxplot: It shows the distribution of the quantitative data that represents the comparisons between variables.

Example:

import seaborn as sns

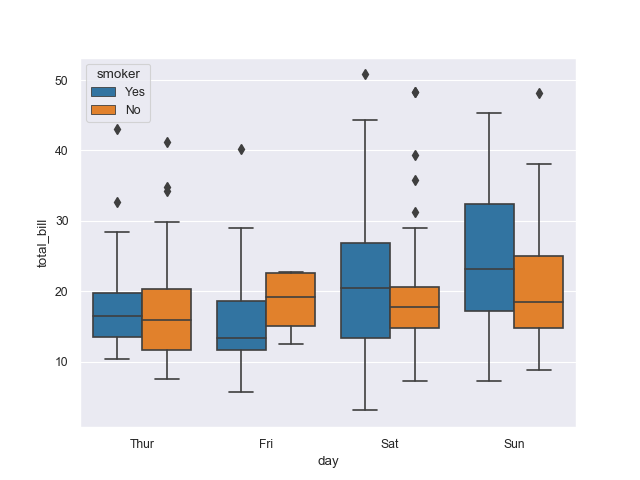
from warnings import filterwarnings

df = sns.load\_dataset('tips')

df.head()

sns.boxplot(x ='day', y ='total\_bill', data = df, hue ='smoker')

plt.show()



1. Stripplot: It basically creates a scatter plot based on the category.

Example:

import seaborn as sns

from warnings import filterwarnings

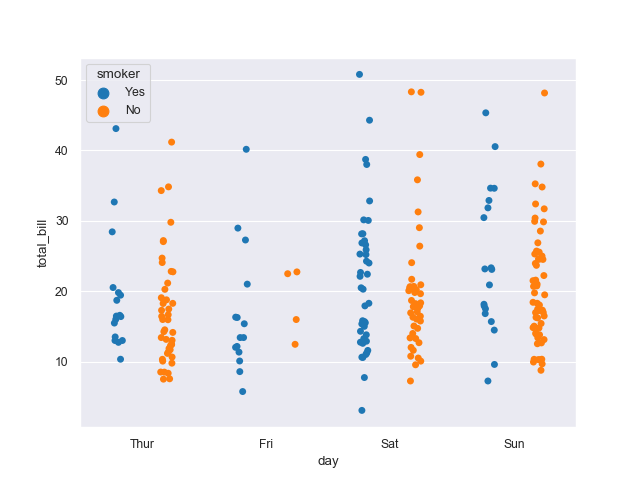
df = sns.load\_dataset('tips')

df.head()

sns.stripplot(x ='day', y ='total\_bill', data = df,

jitter = True, hue ='smoker', dodge = True)

plt.show()



1. Matrix plots:
2. HeatMaps: To use a heatmap the data should be in a matrix form. By matrix we mean that the index name and the column name must match in some way so that the data that we fill inside the cells are relevant.

Example:

**import** seaborn as sns

**import** matplotlib.pyplot as plt **%** matplotlib inline

dataset **=** sns.load\_dataset('tips')

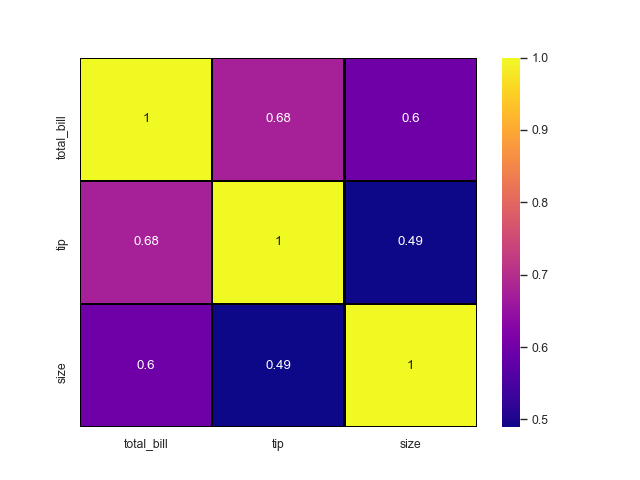
dataset.head()

tc **=** dataset.corr()

sns.heatmap(tc, annot **=** True, cmap **=**'plasma',

            linecolor **=**'black', linewidths **=** 1)

plt.show()



B)Cluster maps: They are used in hierarchical clustering. It performs the clustering based on the similarity of the rows and columns.

Example:

**import** pandas as pd

**import** seaborn as sns

**import** matplotlib.pyplot as plt **%** matplotlib inline

fd **=** sns.load\_dataset('flights')

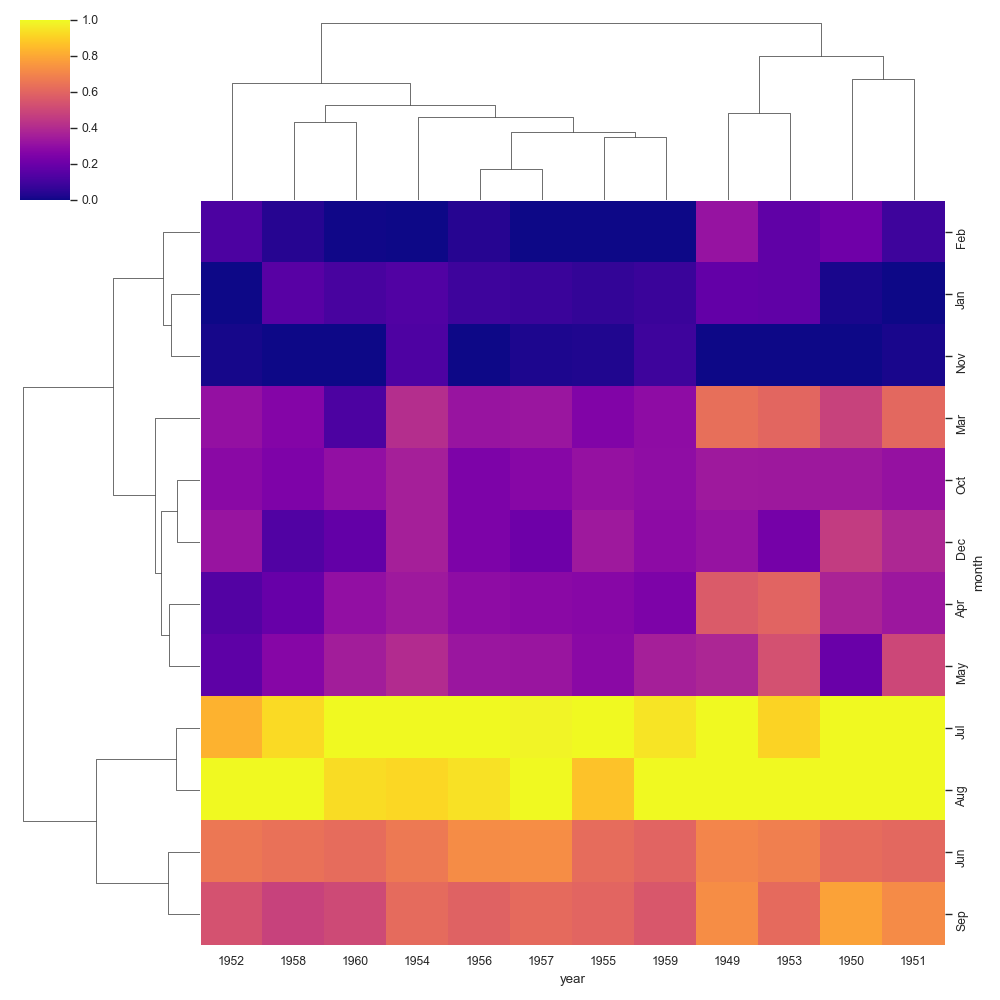
df **=** pd.pivot\_table(values **=**'passengers',

                    index **=**'month', columns **=**'year', data **=** fd)

df.head()

sns.clustermap(df, cmap **=**'plasma', standard\_scale **=** 1)

plt.show()



1. Grids: Grids are general types of plots that allow you to map plot types to grid rows and columns, which helps you to create similar character-separated plots.

A)PairGrid takes all the numerical columns and grids them up making subplots .

Example:

import seaborn as sns

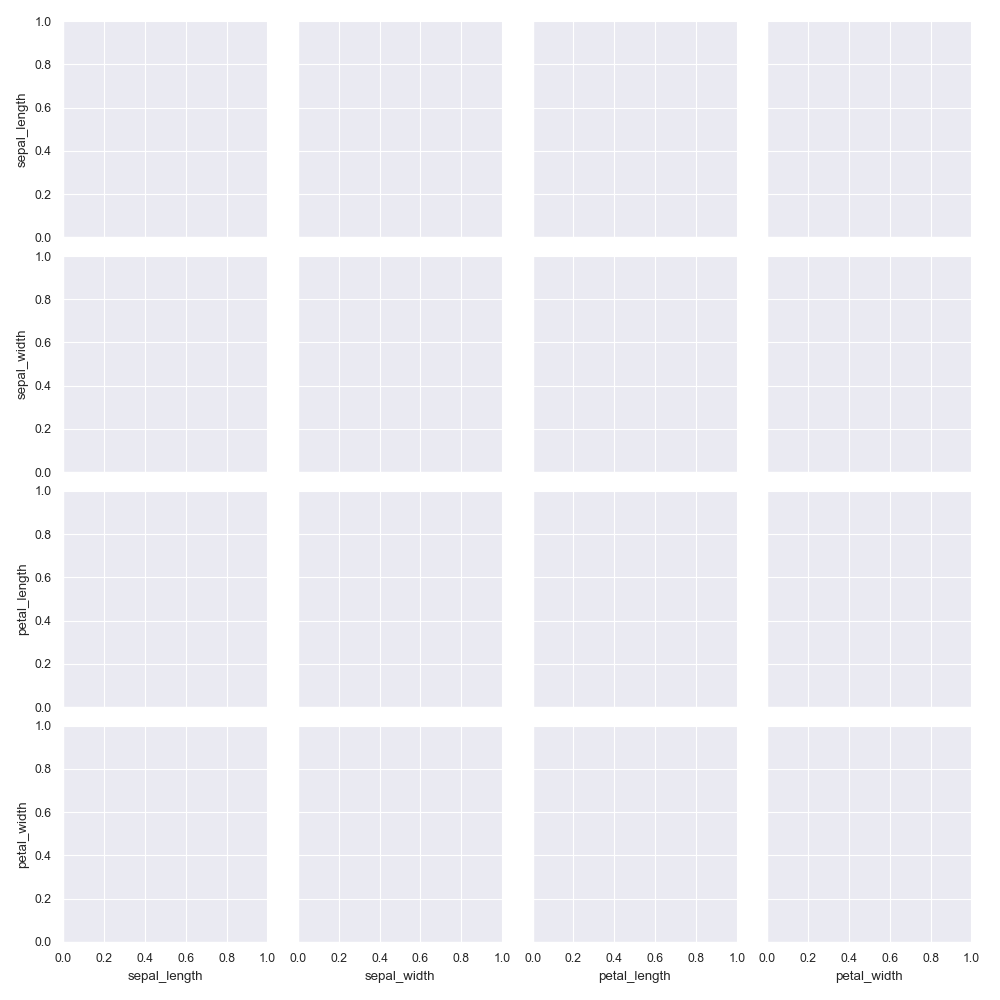
import matplotlib.pyplot as plt

iris = sns.load\_dataset("iris")

iris.head()

sns.PairGrid(data = iris)

plt.show()



1. FacetGrid:FacetGrid is a general way of creating plot grids based on a function. Its object uses the dataframe as input and the names of the variables that shape the row, column, or color dimensions of the grid.

Example:

import seaborn as sns

import matplotlib.pyplot as plt

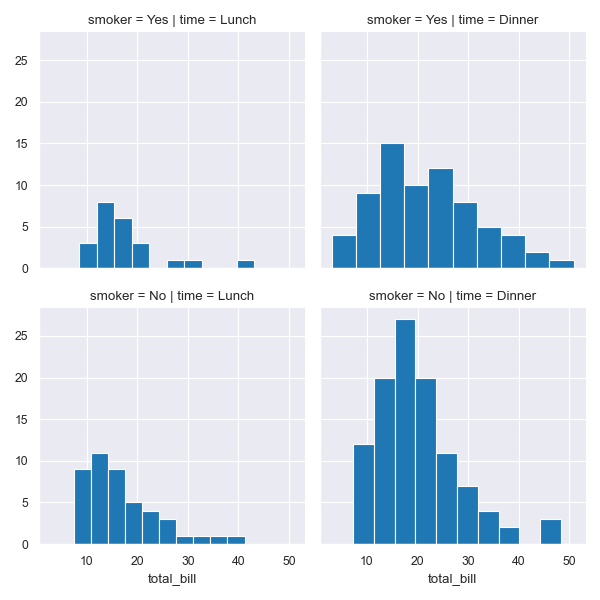
tips = sns.load\_dataset("tips")

tips.head()

g = sns.FacetGrid(tips, col="time", row="smoker")

g = g.map(plt.hist, "total\_bill")

plt.show()



1. JointGrid: Jointplot by Seaborn shows a relationship in the margins between 2 variables (bivariate) and 1D profiles (univariate). This plot is a product form that wraps up the JointGrid.

Example:

import seaborn as sns

import matplotlib.pyplot as plt

tips = sns.load\_dataset("tips")

tips.head()

g = sns.JointGrid(x="total\_bill", y="tip", data=tips)

g = g.plot(sns.regplot, sns.displot)

plt.show()

1. Regression plots: Regression plots as the name suggests creates a regression line between 2 parameters and helps to visualize their linear relationships.

Example:

import seaborn as sns

dataset = sns.load\_dataset('tips')

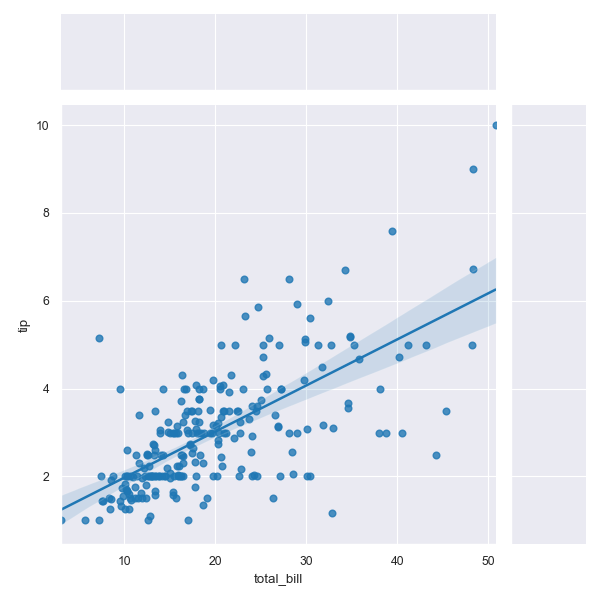
dataset.head()

1. Linear Plot:

sns.set\_style('whitegrid')

sns.lmplot(x ='total\_bill', y ='tip', data = dataset)

plt.show()



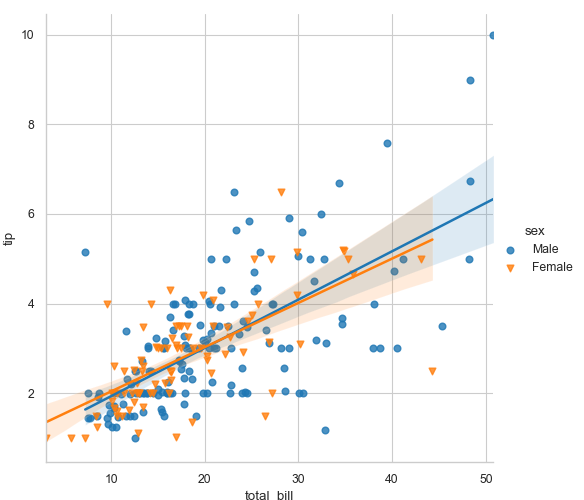
1. Linear Plots with additional parameters:

sns.set\_style('whitegrid')

sns.lmplot(x ='total\_bill', y ='tip', data = dataset,

hue ='sex', markers =['o', 'v'])

plt.show()



1. Setting size and colour:

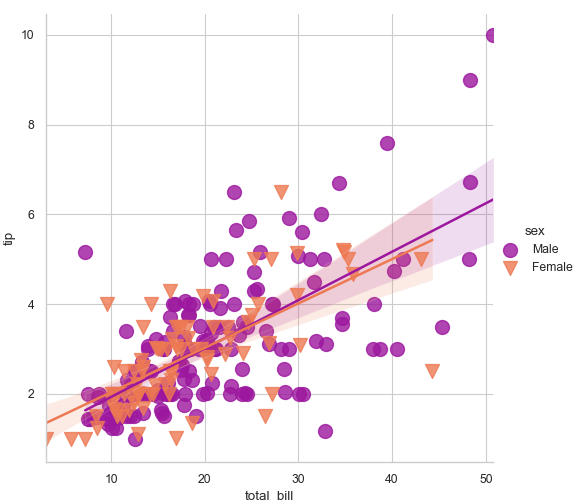
sns.set\_style('whitegrid')

sns.lmplot(x ='total\_bill', y ='tip', data = dataset, hue ='sex',

markers =['o', 'v'], scatter\_kws ={'s':100},

palette ='plasma')

plt.show()

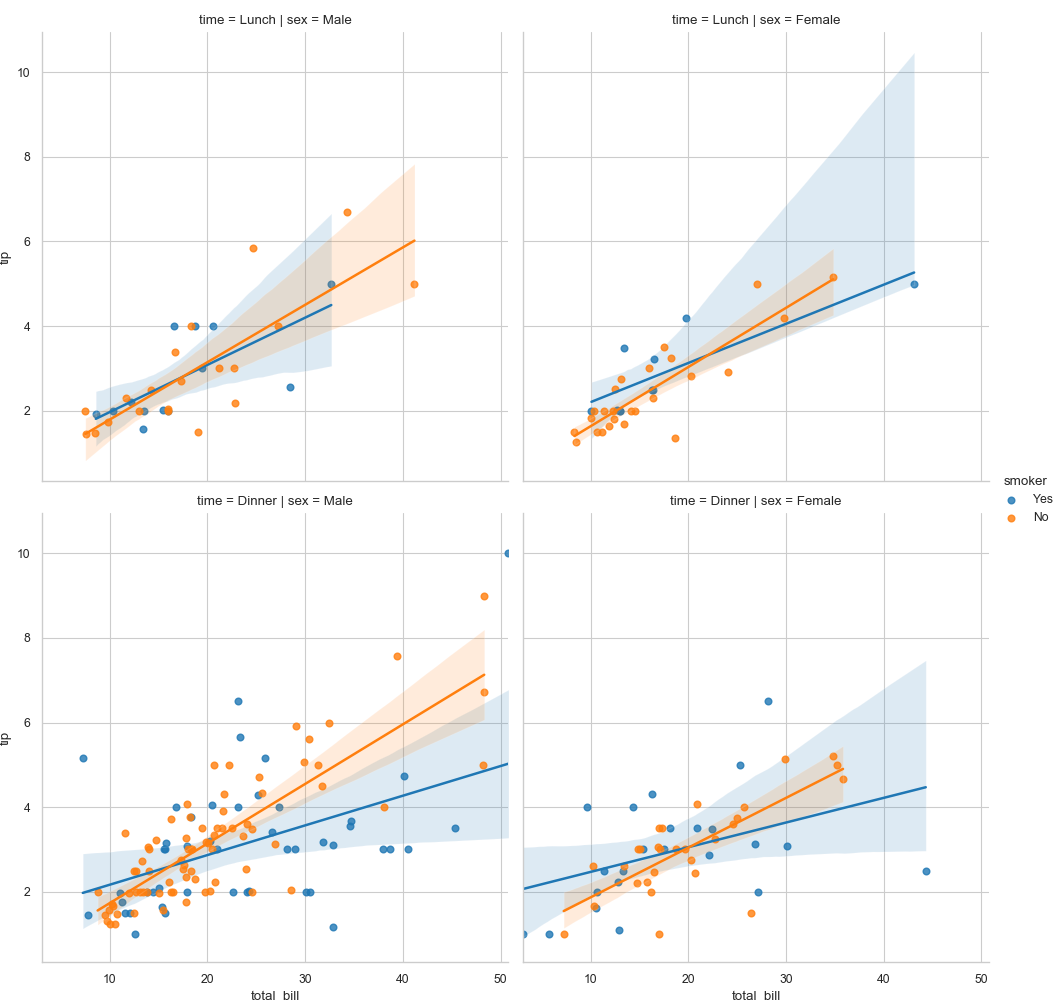


1. Display multiple plots:

sns.lmplot(x ='total\_bill', y ='tip', data = dataset,

col ='sex', row ='time', hue ='smoker')

plt.show()



1. Size and aspect ratio:

sns.lmplot(x ='total\_bill', y ='tip', data = dataset, col ='sex',

row ='time', hue ='smoker', aspect = 0.6,

size = 4, palette ='coolwarm')

plt.show()

