**5.Seaborn:**

**Introduction:**

* It is a Python data visualization library based on matplotlib.
* It provides a high-level interface for drawing attractive and informative statistical graphics.
* Default datatypes in seaborn- tips-dowjones-fmri-dots-healthexp

To load we use: load\_dataset("ds\_name")

* hue=different color for different category
* style=different shapes
* color palette- pastal, bright, dark, muted, colorblind, deep
* kind=line

**Ex: (loading tips dataset)**

import seaborn as sns

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

tips.head()

**Output:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| S.No | Total\_Bill | Tip | Sex | Smoker | Day | Time | Size |
| 0 | 16.99 | 1.01 | Female | No | Sun | Dinner | 2 |
| 1 | 10.34 | 1.66 | Male | No | Sun | Dinner | 3 |
| 2 | 21.01 | 3.50 | Male | No | Sun | Dinner | 3 |
| 3 | 23.68 | 3.31 | Male | No | Sun | Dinner | 2 |
| 4 | 24.59 | 3.61 | Female | No | Sun | Dinner | 4 |

**Ex: (loading dowjones dataset)**

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

dowjones.head()

**Output:**

|  |  |  |
| --- | --- | --- |
| S.No | Date | Price |
| 0 | 1914-12-01 | 55.00 |
| 1 | 1915-01-01 | 56.55 |
| 2 | 1915-02-01 | 56.00 |
| 3 | 1915-03-01 | 58.30 |
| 4 | 1915-04-01 | 66.45 |

**Ex: (loading fmri dataset)**

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

fmri.head()

**Output:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| S.No | Subject | Timepoint | Event | Region | Signal |
| 0 | s13 | 18 | stim | parietal | -0.017552 |
| 1 | s5 | 14 | stim | parietal | -0.080883 |
| 2 | s12 | 18 | stim | parietal | -0.081033 |
| 3 | s11 | 18 | stim | parietal | -0.046134 |
| 4 | s10 | 18 | stim | parietal | -0.037970 |

**Ex: (loading dots dataset)**

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

dots.head()

**Output:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| S.No | Allign | Choice | Time | Coherence | Firing\_Rate |
| 0 | dots | T1 | -80 | 0.0 | 33.189967 |
| 1 | dots | T1 | -80 | 3.2 | 31.691726 |
| 2 | dots | T1 | -80 | 6.4 | 34.279840 |
| 3 | dots | T1 | -80 | 12.8 | 32.631874 |
| 4 | dots | T1 | -80 | 25.6 | 35.060487 |

**Ex: (loading healthexp dataset)**

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

healthexp.head()

**Output:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S.No | Year | Country | Spending\_USD | Life\_Expectancy |
| 0 | 1970 | Germany | 252.311 | 70.6 |
| 1 | 1970 | France | 192.143 | 72.2 |
| 2 | 1970 | Great Britain | 123.993 | 71.9 |
| 3 | 1970 | Japan | 150.437 | 72.0 |
| 4 | 1970 | USA | 326.961 | 70.9 |

**5.1.Relplot()** :

This function provides us the access to some other different axes-level functions which shows the relationships between two variables with semantic mappings of subsets.

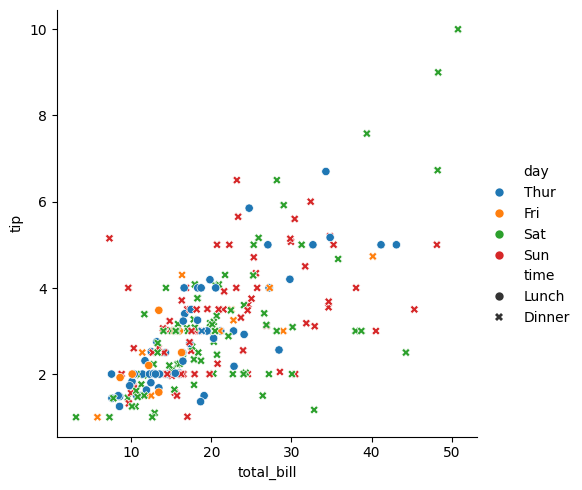
* seaborn.relplot(x=" ", y=" ", data=)

**Ex-1:**

sns.relplot(data=tips,x="total\_bill",y="tip",hue="day",style="time")

**#hue-creating difference based on a column via colors**

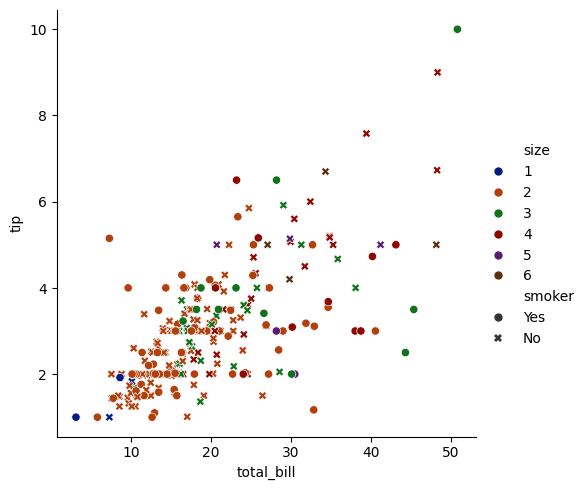
**Output:**



**Ex-2:**

sns.relplot(data=tips,x="total\_bill",y="tip",hue="size",style="smoker",palette="dark")

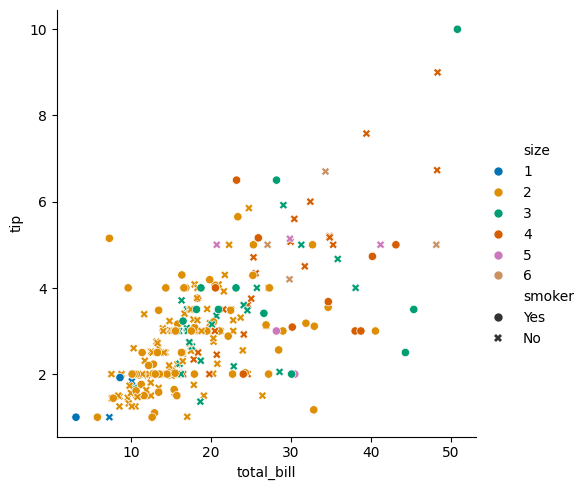
**Output:**



**Ex-3:**

sns.relplot(data=tips,x="total\_bill",y="tip",hue="size",style="smoker",palette="colorblind")

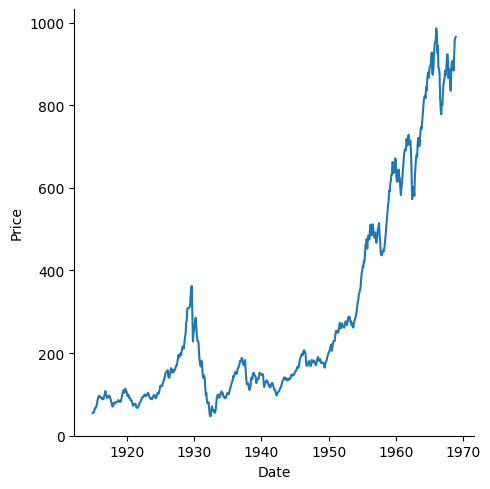
**Output:**



**Ex-4:**

sns.relplot(data=dowjones,x="Date",y="Price",kind="line")

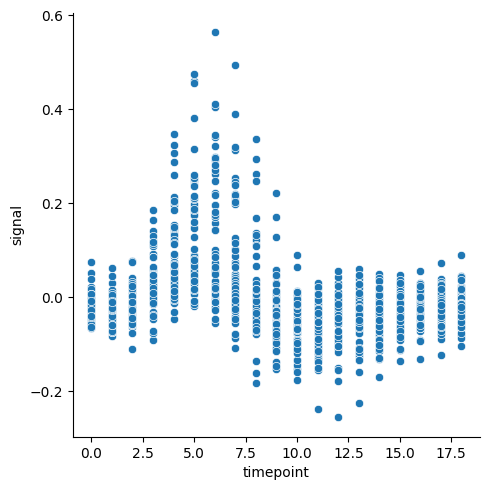
**Output:**



**Ex-5:**

sns.relplot(data=fmri,x="timepoint",y="signal")

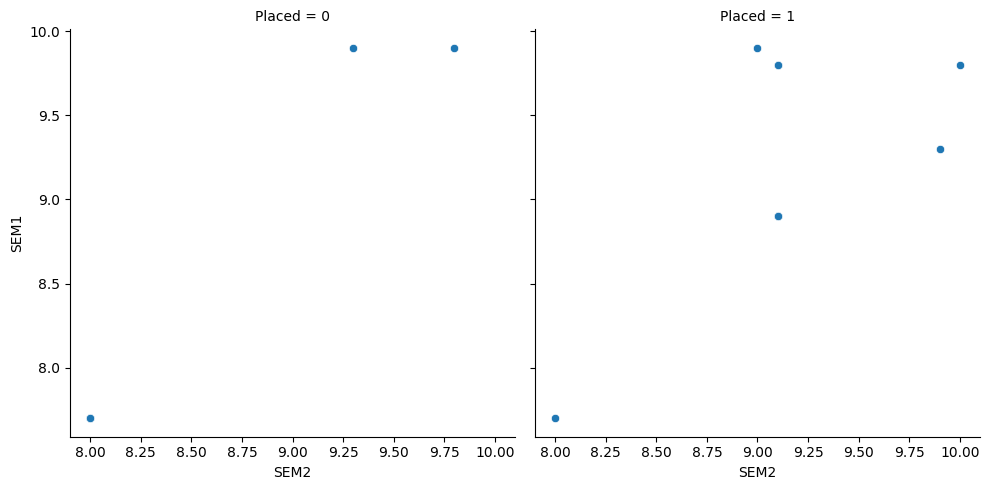
**Output:**



**Ex-6:**

sns.relplot(data=dfn,x="SEM2",y="SEM1",col="Placed")

**Output:**



**5.2.Catplot() :**

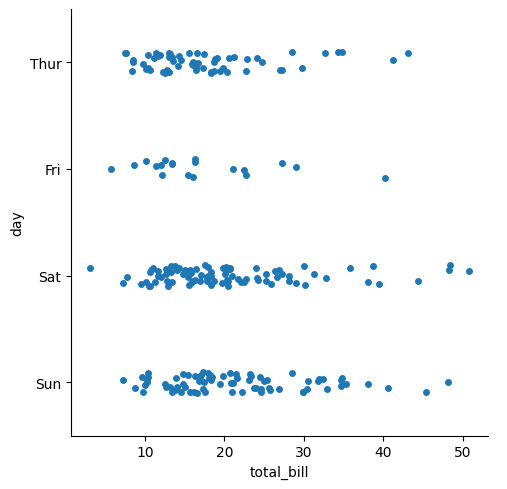
* catplot() method is used to plot categorical plots.
* This function gives users access to a number of axes-level functions that illustrate the connection between numerical data and one or more category variables.

Syntax: seaborn.catplot(x=" ", y=" ", hue=" ", data=" ", row=" ", col=" ",kind=" ", color=" ", palette=" ")

**Ex-1:**

sns.catplot(data=tips,x="total\_bill",y="day")

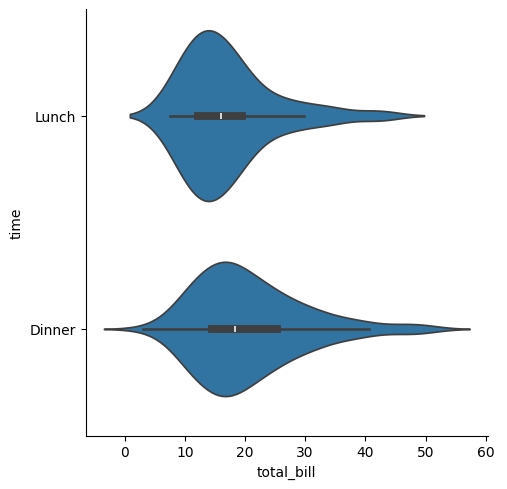
**Output:**



**Ex-2:**

sns.catplot(data=tips,x="total\_bill",y="time",kind="violin")

**Output:**



**5.3.lineplot() :**

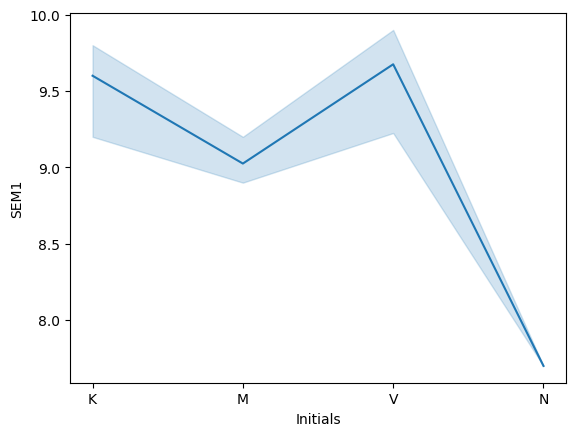
* Line Plot is one of the easiest and most basic graphical analysis techniques that play an important role in data analysis when working on machine learning or data science projects.
* They are used to express a relationship between two variables.
* The article focuses on plotting a line chart in Python using Matplotlib.

**Ex:**

import seaborn as sns

pl=sns.lineplot(x='Initials',y='SEM1',data=dropped)

**Output:**



**Problem:**

**-** load diabetes.csv

- create relplot

- with index numbersas x-axis age in y-axis and class as columns

**Code:**

df1=pd.read\_csv("/content/diabetcsv.csv")

df1

df1['index\_no.s']=range(1,769) #df['newcol']=values

df1.head()

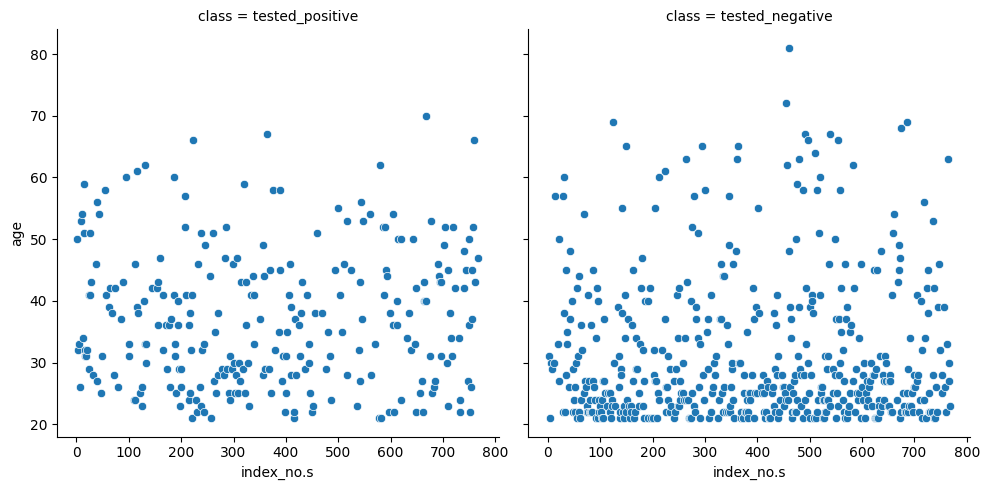
sns.relplot(data=df1,x="index\_no.s",y="age",col="class")

**Output:**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | preg | plas | pres | skin | insu | mass | pedi | age | class |
| 0 | 6 | 148 | 72 | 35 | 0 | 33.6 | 0.627 | 50 | tested\_positive |
| 1 | 1 | 85 | 66 | 29 | 0 | 26.6 | 0.351 | 31 | tested\_negative |
| 2 | 8 | 183 | 64 | 0 | 0 | 23.3 | 0.672 | 32 | tested\_positive |
| 3 | 1 | 89 | 66 | 23 | 94 | 28.1 | 0.167 | 21 | tested\_negative |
| 4 | 0 | 137 | 40 | 35 | 168 | 43.1 | 2.288 | 33 | tested\_positive |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 763 | 10 | 101 | 76 | 48 | 180 | 32.9 | 0.171 | 63 | tested\_negative |
| 764 | 2 | 122 | 70 | 27 | 0 | 36.8 | 0.340 | 27 | tested\_negative |
| 765 | 5 | 121 | 72 | 23 | 112 | 26.2 | 0.245 | 30 | tested\_negative |
| 766 | 1 | 126 | 60 | 0 | 0 | 30.1 | 0.349 | 47 | tested\_positive |
| 767 | 1 | 93 | 70 | 31 | 0 | 30.4 | 0.315 | 23 | tested\_negative |

768 rows × 9 columns

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | preg | plas | pres | skin | insu | mass | pedi | age | class | index\_no.s |
| 0 | 6 | 148 | 72 | 35 | 0 | 33.6 | 0.627 | 50 | tested\_positive | 1 |
| 1 | 1 | 85 | 66 | 29 | 0 | 26.6 | 0.351 | 31 | tested\_negative | 2 |
| 2 | 8 | 183 | 64 | 0 | 0 | 23.3 | 0.672 | 32 | tested\_positive | 3 |
| 3 | 1 | 89 | 66 | 23 | 94 | 28.1 | 0.167 | 21 | tested\_negative | 4 |
| 4 | 0 | 137 | 40 | 35 | 168 | 43.1 | 2.288 | 33 | tested\_positive | 5 |



**linear fit:**

* A linear line is rotated to a particular angle until the distance b/w individual points in a graph that are plotte and the titled line is minimum in all cases
* lmplot()

**linear regression(algorithm):**

- plot the data set

- performing linear fit

- predict the value based on graph

**Ex:**

sns.lmplot(data=tips,x="size",y="tip")

**Output:**

