

▼ Seaborn Module

Python Seaborn module serves the purpose of Data Visualization at an ease with higher efficiency. In order to represent the variations in a huge data set, data visualization is considered as the best way to depict and analyze the data.

Seaborn stands out to have a better set of functions to carry out data visualization than Matplotlib in an optimized and efficient manner. It supports NumPy and Pandas data structure to represent the data sets.

But, in order to get started with the Seaborn module, I would strongly recommend the readers to understand the Python Matplotlib module.

▼ Python Seaborn For Statistical Analysis

Statistical Analysis is the basic estimation out of some parameters of the data-set to a large extent. Data Visualization can be considered as the best way to perform statistical analysis i.e. predict the outcome or the cause based on diagrammatic values.

Either of the following ways can be taken into consideration during the statistical analysis:

seaborn.scatterplot() seaborn.lineplot()

▼ 1. seaborn.scatterplot()

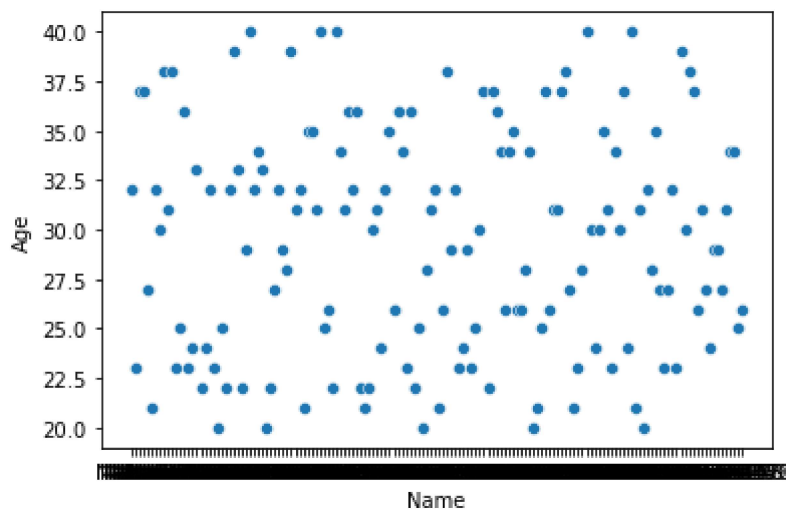
The seaborn.scatterplot() function is basically used to depict the relationship between the parameters on the given axes respectively. Every point on the graph depicts a value corresponding to it.

```
import seaborn
import pandas as pd
import matplotlib.pyplot as plt

#csv = pd.read_csv('iris.csv')
csv = pd.read_csv('Book1.csv')

#res = seaborn.scatterplot(x="petal.length", y="variety", data=csv)
res = seaborn.scatterplot(x="Name", y="Age", data=csv)
plt.show()

csv
```



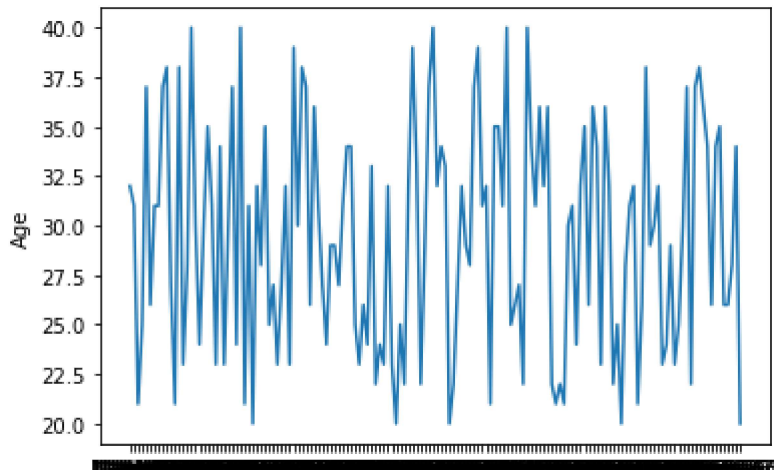
	Name	Age
0	Name1	32
1	Name2	23
2	Name3	37
3	Name4	37
4	Name5	27
...
145	Name146	31
146	Name147	34
147	Name148	34
148	Name149	25
149	Name150	26

150 rows × 2 columns

▼ 2. seaborn.lineplot()

The `seaborn.lineplot()` function can be extensively used in situations wherein we feel the need to check the dependency of a parameter on the other in a continuous manner relative to time.

```
import seaborn
import pandas
import matplotlib.pyplot as plt
csv = pandas.read_csv('Book1.csv')
res = seaborn.lineplot(x="Name", y="Age", data=csv)
plt.show()
```



▼ Categorical Scatter Plot

Categorical data divides and represents itself in the form of discrete groups i.e. a subset of the original data.

Python Seaborn module contains the following methods to represent and visualize categorical data:

`seaborn.catplot()` `seaborn.stripplot()` `seaborn.swarmplot()`

▼ `seaborn.catplot()`

The `seaborn.catplot()` function, as mentioned above, is one of the techniques to analyze the relationship between a numeric value and a categorical group of values together.

```
import seaborn
import pandas
import matplotlib.pyplot as plt
```

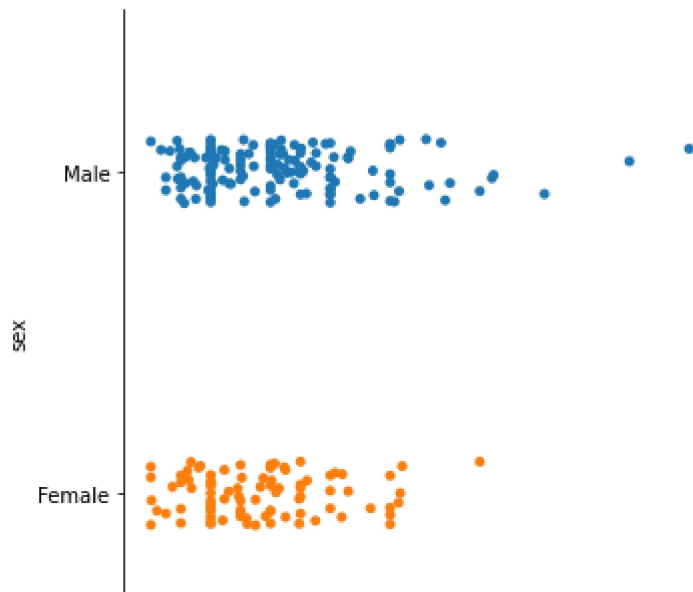
```
csv = seaborn.load_dataset("tips")
print(csv)
res = seaborn.catplot(x="tip", y="sex", jitter=True, data=csv)
```

```
# csv = pandas.read_csv('iris.csv')
# res = seaborn.catplot(x="petal.length", y="variety", jitter=False, data=csv)
# # print(csv.head())
# plt.show()
```

#The parameter `jitter` is useful when the data set consists of data points that overlap.
#In such cases, setting a `jitter` value can help them get uniformly distributed.

	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
..
239	29.03	5.92	Male	No	Sat	Dinner	3
240	27.18	2.00	Female	Yes	Sat	Dinner	2
241	22.67	2.00	Male	Yes	Sat	Dinner	2
242	17.82	1.75	Male	No	Sat	Dinner	2
243	18.78	3.00	Female	No	Thur	Dinner	2

[244 rows x 7 columns]



▼ Categorical Distribution Plots

Categorical Distribution data basically refers to the type of data wherein the result describes the certain possibility of the random/chosen variable to belong to one of the given possible categories.

Python Seaborn has the following functions to represent the categorical distributed data efficiently:

1. `seaborn.violinplot()`
2. `seaborn.boxplot()`
3. `seaborn.boxenplot()`

▼ `seaborn.boxplot()`

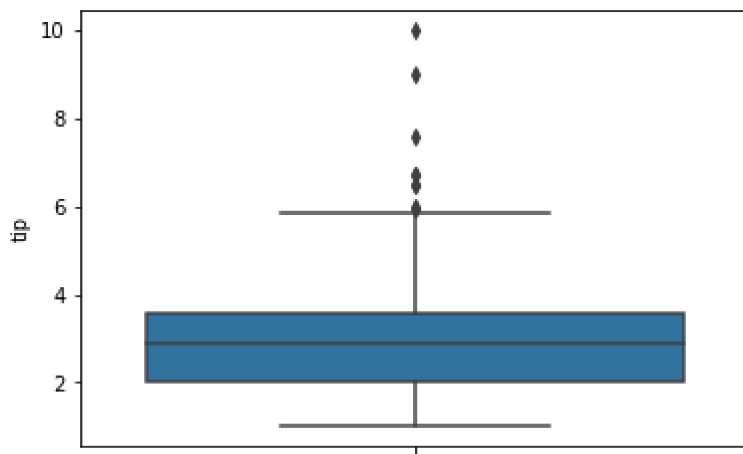
The `seaborn.boxplot()` function represents the categorical distribution of data and sets comparison among the different categorical data inputs.

The 'box' structure represents the main quartile of the data input while the 'line' structure represents the rest of the distribution of data. The outliers are represented by points using an inter-quartile

function.

```
import seaborn
import pandas
import matplotlib.pyplot as plt
csv = seaborn.load_dataset("tips")
res = seaborn.boxplot(y=csv['tip'])

#csv = pandas.read_csv("C:\\Book1.csv")
#res = seaborn.boxplot(x=csv['Age'])
plt.show()
```



▼ Categorical estimate plots

The estimation of categorical data basically refers to the representation of certain estimation or prediction of the categorical data values to the corresponding data variable.

Python Seaborn has the following functions to be used for the estimation of categorical data:

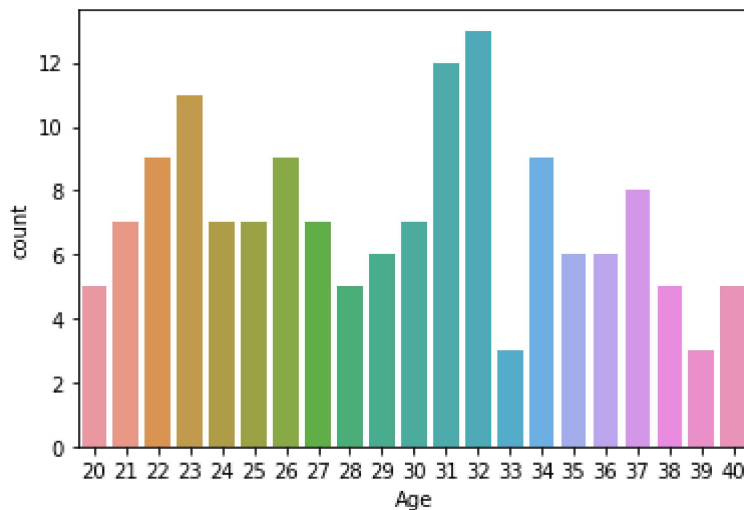
1. `seaborn.countplot()`
2. `seaborn.barplot()`
3. `seaborn.pointplot()`

▼ `seaborn.countplot()`

The `countplot()` function basically counts the frequency of the input data field and represented it along the y-axis while the data field – 'Age' being represented along the x-axis.

```
import seaborn
import pandas
import matplotlib.pyplot as plt
```

```
import matplotlib.pyplot as plt
csv = pandas.read_csv("Book1.csv")
res = seaborn.countplot(x=csv['Age'])
plt.show()
```



▼ Plotting univariate distributions with Seaborn

Univariate distribution basically refers to the distribution of the data with respect to a single random variable/data item.

Python Seaborn module's `seaborn.distplot()` function can be used to represent the univariate distribution of data set.

```
import seaborn
import pandas
import matplotlib.pyplot as plt
seaborn.set_style("darkgrid")
csv = pandas.read_csv("Book1.csv")
res=seaborn.distplot(csv['Age'])
plt.show()
```

```
C:\Users\Shyamala\anaconda3\lib\site-packages\seaborn\distributions.py:2551: FutureWarning:
warnings.warn(msg, FutureWarning)
```

0.07

▼ Depicting bivariate distributions with Seaborn

Bivariate distribution refers to the visualization of data with respect to two data columns or items of the data set.

The `seaborn.jointplot()` can be used to depict the relationship between the two data variables.

```
import seaborn
import pandas
import matplotlib.pyplot as plt
# seaborn.set_style("darkgrid") #whitegrid is also available
# csv = pandas.read_csv("Book1.csv")
# res=seaborn.jointplot(x=csv['Age'], y=csv['Age'])
# plt.show()

csv = seaborn.load_dataset("tips")
res=seaborn.jointplot(csv['total_bill'], csv['tip'])
plt.show()
```

```
C:\Users\Shyamala\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning:
warnings.warn(
```

