Introduction to Seaborn – Python

What is Seaborn

Seaborn is an amazing visualization library for statistical graphics plotting in Python. It provides beautiful default styles and color palettes to make statistical plots more attractive. It is built on top matplotlib library and is also closely integrated with the data structures from pandas.

Seaborn aims to make visualization the central part of exploring and understanding data. It provides dataset-oriented APIs so that we can switch between different visual representations for the same variables for a better understanding of the dataset.

Different categories of plot in Seaborn

Plots are basically used for visualizing the relationship between variables. Those variables can be either completely numerical or a category like a group, class, or division. Seaborn divides the plot into the below categories –

Relational plots: This plot is used to understand the relation between two variables.

Categorical plots: This plot deals with categorical variables and how they can be visualized.

Distribution plots: This plot is used for examining univariate and bivariate distributions

Regression plots: The regression plots in Seaborn are primarily intended to add a visual guide that helps to emphasize patterns in a dataset during exploratory data analyses.

Matrix plots: A matrix plot is an array of scatterplots.

Multi-plot grids: It is a useful approach to draw multiple instances of the same plot on different subsets of the dataset.

Some basic plots using seaborn

Histplot: Seaborn Histplot is used to visualize the univariate set of distributions(single variable). It plots a histogram, with some other variations like kdeplot and rugplot. The Histplot function takes several arguments but the important ones are

data: This is the array, series, or dataframe that you want to visualize. It is a required parameter.

x: This specifies the column in the data to use for the histogram. If your data is a dataframe, you can specify the column by name.

y: This specifies the column in the data to use for the histogram when you want to create a bivariate histogram. By default, it is set to None, meaning that a univariate histogram will be plotted.

bins: This specifies the number of bins to use when dividing the data into intervals for plotting. By default, it is set to “auto”, which uses an algorithm to determine the optimal number of bins.

kde: This parameter controls whether to display a kernel density estimate (KDE) of the data in additionn to the histogram. By default, it is set to False, meaning that a KDE will not be plotted.

```
import numpy as np

import seaborn as sns

sns.set(style="white")

# Generate a random univariate dataset

rs = np.random.RandomState(10)

d = rs.normal(size=100)

# Plot a simple histogram and kde

sns.histplot(d, kde=True, color="m")
```

```
import numpy as np

import seaborn as sns

sns.set(style="white")

# Generate a random univariate dataset

rs = np.random.RandomState(10)

d = rs.normal(size=100)

# Plot a simple histogram and kde

sns.histplot(d, kde=True, color="m")
```

# Python program to illustrate Plotting categorical scatter plots with Seaborn

```
# importing the required module

import matplotlib.pyplot as plt

import seaborn as sns

# x axis values

x =['sun', 'mon', 'fri', 'sat', 'tue', 'wed', 'thu']

# y axis values

y =[5, 6.7, 4, 6, 2, 4.9, 1.8]

# plotting strip plot with seaborn

ax = sns.stripplot(x, y);

# giving labels to x-axis and y-axis

ax.set(xlabel ='Days', ylabel ='Amount_spend')

# giving title to the plot
```

```
plt.title('My first graph');

# function to show plot

plt.show()
```

Plotting Chart Using seaborn Library

1. Line plot:

The seaborn line plot is one of the most basic plots presents in the seaborn library. We use the seaborn line plot mainly to visualize the given data in some time-series form, i.e., in a continuous manner with respect to time.

Example -

```
# Importing seaborn library in program
import seaborn as sns
# Importing mataplotlib library to show graph in output
import matplotlib.pyplot as plt
# Setting style with set() function
sns.set(style="dark")
# Using dataset() function to declare data type
FMR = sns.load_dataset("fmri")
# Plotting various responses for different\
# Regions and events
sns.lineplot(x="timepoint",
          y="signal",
          hue="region",
          style="event",
          data=FMR) # using lineplot() function to create line plot
plt.show() # using show() function
```

2. Dist plot:

We use the seaborn dist plots to plot histograms with the given variables and data as a result. We can plot histograms with some other variations such as rugplot and kdeplot using a dist plot.

Example -

```
# importing numpy as np library module
import numpy as np
# Importing seaborn library in program
import seaborn as sns
# Importing mataplotlib library to show graph in output
import matplotlib.pyplot as plt
# Selecting style for boxplot with set() function
sns.set(style="white")
# Generate a random univariate type distribution
ru = np.random.RandomState(10)
d = ru.normal(size=100)
# Plotting a simple histogram with kdeplot variation
sns.histplot(d, kde=True, color="m")
plot = sns.histplot(d, kde=True, color="m")
print(plot)
plt.show() # using show() function
```

3. Lmplot:

The Lmplot is another one of the basic plots in the seaborn library. The Lmplot shows a line that represents a linear regression model with the data points on the given two-dimensional (2-D) space. In this 2-D space, we can set x and y variables as the vertical and horizontal labels, respectively.

Example -

```
# Importing seaborn library in program
import seaborn as sns
# Importing matplotlib library to show graph in output
import matplotlib.pyplot as plt
# Using set() function to set style
sns.set(style="ticks")
# Using dataset() function
ds = sns.load_dataset("anscombe")
```

```
# Showing results in the form of linear regression

sns.lmplot(x="x", y="y", data=ds)



plot = sns.lmplot(x="x", y="y", data=ds)

print(plot)

plt.show() # using show() function
```

Countplot in Python

In this article, we will discuss how we can create a countplot using the seaborn library and how the different parameters can be used to infer results from the features of our dataset.

Seaborn library

The seaborn library is widely used among data analysts, the galaxy of plots it contains provides the best possible representation of our data.

The seaborn library can be imported into our working environment using-

```
import seaborn as sns
```

Let us now discuss why do we use countplot and what is the significance of its parameters.

Countplot

The countplot is used to represent the occurrence(counts) of the observation present in the categorical variable.

It uses the concept of a bar chart for the visual depiction.

Parameters-

The following parameters are specified when we create a countplot, let us get a brief idea of them-

x and y- This parameter specifies the data we refer to for representation and then observes the highlighted patterns.

color- This parameter specifies the color that can give a good appearance to our plot.

palette- It takes the value of the palette. It is mostly used to show the hue variable.

hue- This parameter specifies the column name.

data- This parameter specifies the data frame we would like to take for the representation. For instance, data can be an array.

dodge- This parameter is an optional one and it accepts a Boolean value as input.

saturation- This parameter accepts a float value. A variation in the intensity of colors can be observed when we specify this.

hue_order- The parameter hue_order takes strings as an input.

kwargs- The parameter kwargs specifies the key and value mappings.

ax- The parameter ax is an optional one and is used to take axes on which plots are created.

orient- The parameter orient is an optional one and tells the orientation of the plot that we need, horizontal or vertical.

Now let us see what are the different ways of representing our attributes.

In the first example, we will create a countplot for a single variable. We have taken the dataset 'tips' to implement the same.

1. Value counts for a single variable

Example -

```
import matplotlib.pyplot as plt

import seaborn as sns

import pandas as pd

#loading the dataset 'tips'

df=pd.read_csv("/content/tips.csv")

#plotting the graph

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

plt.show()
```

2. Representing two categorical variables using hue parameter

Example -

```
import matplotlib.pyplot as plt

import seaborn as sns

import pandas as pd

#loading the dataset 'tips'

df=pd.read_csv("/content/tips.csv")

#plotting the graph

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

plt.show()
```

3. Creating Horizontal Plots

Example -

```
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
#loading the dataset 'tips'
df=pd.read_csv("/content/tips.csv")
#plotting the graph
sns.countplot(y='sex',hue='smoker',data=df)
plt.show()
```

4. Using color palettes

Input-

```
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
#loading the dataset 'tips'
df=pd.read_csv("/content/tips.csv")
#plotting the graph
sns.countplot(x='sex', data=df, palette='Set1')
plt.show()
```

5. Using a parameter 'color'

Example -

```
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
#loading the dataset 'train'
df=pd.read_csv("/content/train.csv")
#plotting the graph
sns.countplot(x='Pclass',hue='Sex', data=df, color='green')
plt.show()
```

6. Using the parameter 'saturation'

Example -

```
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
#loading the dataset 'train'
df=pd.read_csv("/content/train.csv")
#plotting the graph
sns.countplot(x='Pclass',data=df, color='green', saturation=0.1)
plt.show()
```

Example -

```
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
#loading the dataset 'train'
df=pd.read_csv("/content/train.csv")
sns.countplot(x='Sex', data=df, color="green", facecolor=(0,0,0,0), linewidth=5, edgecolor=sns.color_palette("BrBG",2))
plt.show()
```