Data Science with Python Programming

- Presentation By Uplatz
- Contact us: https://training.uplatz.com
- Email: info@uplatz.com
- Phone: +44 7836 212635

Uplatz

Data Visualisation with Seaborn

Learning outcomes:

- Introduction to seaborn
- Seaborn Functionalities
- Installing seaborn
- Different categories of plot in Seaborn
- Exploring Seaborn Plots

Introduction to seaborn

Seaborn is a library for making statistical graphics in Python. It is built on top of matplotlib and closely integrated with pandas data structures.
Seaborn is an open source, BSD-licensed Python library providing high level API for visualizing the data using Python programming language.
As Seaborn compliments and extends Matplotlib, the learning curve is quite gradual. If you know Matplotlib, you are already half way through Seaborn.

Uplatz

Introduction to seaborn

Important Features of Seaborn:

Seaborn is built on top of Python's core visualization library Matplotlib. It is meant to serve as a complement, and not a replacement. However, Seaborn comes with some very important features. Let us see a few of them here. The features help in

- Built in themes for styling matplotlib graphics
- Visualizing univariate and bivariate data
- Fitting in and visualizing linear regression models
- Plotting statistical time series data

Uplatz

Introduction to seaborn

Important Features of Seaborn:

- Seaborn works well with NumPy and Pandas data structures
- It comes with built in themes for styling Matplotlib graphics

In most cases, you will still use Matplotlib for simple plotting. The knowledge of Matplotlib is recommended to tweak Seaborn's default plots.

Seaborn Functionalities

Here is some of the functionality that seaborn offers:

- A dataset-oriented API for examining relationship between multiple variables.
- Specialized support for using categorical variables to show aggregate statistics
- Options for visualizing univariate or bivariate distributions and for comparing them between subsets of data
- Automatic estimation and plotting of linear regression models for different kinds dependent variables

Seaborn Functionalities

Here is some of the functionality that seaborn offers:

- Convenient views onto the overall structure of complex datasets
- High-level abstractions for structuring multi-plot grids that let you easily build complex visualizations
- Concise control over matplotlib figure styling with several built-in themes
- Tools for choosing color palettes that faithfully reveal patterns in your data

Installing *seaborn*

If you have Python and PIP already installed on a system, you can install it easily.
Open the command prompt and change the directory to your current working directory and then try this and see if works on your system:

pip3 install seaborn OR
python -m pip install seaborn OR
pip install --user seaborn

Different categories of plot in Seaborn

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

Different categories of plot in Seaborn

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 an useful approach is to draw multiple instances of the same plot on different subsets of the dataset.

Exploring Seaborn Plots

Distplots:

Distplot stands for distribution plot, it takes as input an array and plots a curve corresponding to the distribution of points in the array.

The seaborn.distplot() function is used to plot the **distplot**. The **distplot** represents the univariate distribution of data i.e. data distribution of a variable against the density distribution.

The seaborn.distplot() function accepts the data variable as an argument and returns the plot with the density distribution.

Exploring Seaborn Plots

Distplots:

Example_1: Basic Example

```
import matplotlib.pyplot as plt
import seaborn as sb
sb.distplot([0,1,2,3,4,5])
plt.show()
```

Exploring Seaborn Plots

Distplots: Example_2:

```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sb

#creates an array of specified shape and fills it with random values as per standard normal distribution

data = np.random.randn(200)
sb.distplot(data)
plt.show()
```

Uplatz

Exploring Seaborn Plots

Line Plots:

The line plot is one of the most basic plot in seaborn library. **Seaborn Line Plots** depict the relationship between continuous as well as categorical values in a continuous data point format. Syntax for creating line plot is

seaborn.lineplot(x, y, data)

x: Data variable for the x-axis

y: Data variable for the y-axis

data: The object pointing to the entire data set or data values

Exploring Seaborn Plots

Line Plots: Using random data to create a Seaborn Line Plot. Example_1:

```
import matplotlib.pyplot as plt
import seaborn as sb
import pandas as pd
x= [10,5,15,20,7,8]
y= [100,120,150,95,210,240]
data_plot = pd.DataFrame({"Product_no":x,
"Cost":y})
sb.lineplot(x='Product_no', y='Cost',
data=data_plot)
plt.show()
```

Exploring Seaborn Plots

Lmplot :

The lmplot is another most basic plot. It shows a line representing a linear regression model along with data points on the 2D-space and x and y can be set as the horizontal and vertical labels respectively.

seaborn.lmplot() method is used to draw a scatter plot onto a FacetGrid

Exploring Seaborn Plots

Lmplot : Example_1

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

df = sb.load_dataset("anscombe")
print(df)
# Show the results of a linear regression
sb.lmplot(x="x", y="y", data=df)
plt.show()
```

Exploring Seaborn Plots

Marginal plot :

A **marginal plot** allows to study the relationship between 2 numeric variables. The central chart display their **correlation**. It is usually a scatterplot, a hexbin plot, a 2D histogram or a 2D density plot. The marginal charts, usually at the top and at the right, show the **distribution** of the 2 variables using histogram or density plot. The seaborn library provides a joint plot function that is really handy to make this type of graphic. **jointplot()** allows you to basically match up two distplots for bivariate data. With your choice of what **kind** parameter to compare with: "scatter" "reg" "resid" "kde" "hex"

Exploring Seaborn Plots

stripplot and swarmplot:

The **stripplot** will draw a scatterplot where one variable is categorical. A strip plot can be drawn on its own, but it is also a good complement to a box or violin plot in cases where you want to show all observations along with some representation of the underlying distribution

The **swarmplot** is similar to **stripplot**(), but the points are adjusted (only along the categorical axis) so that they don’t overlap. This gives a better representation of the distribution of values, although it does not scale as well to large numbers of observations

Exploring Seaborn Plots

boxplot and violinplot:

boxplots and **violinplots** are used to shown the distribution of categorical data. A box plot (or box-and-whisker plot) shows the distribution of quantitative data in a way that facilitates comparisons between variables or across levels of a categorical variable. The box shows the quartiles of the dataset. A violin plot plays a similar role as a box and whisker plot. It shows the distribution of quantitative data across several levels of one (or more) categorical variables such that those distributions can be compared. Unlike a box plot, in which all of the plot components correspond to actual datapoints, the violin plot features a kernel density estimation of the underlying distribution.

Thank you

Uplatz