**Python Data Visualization Libraries**

Top of Form

We are all aware that visualization is an important part of a data analysis and there are many well-known widely used Python visualization packages available like Matplotlib, Seaborn etc. In this post I will be discussion about top Python packages available and their features.

**Matplotlib**

Matplotlib is incredibly exceptional Python data visualization library. It is most widely used even though it is over a decade old. Its design resembles to 1980's MATLAB programming language. As it was first Python data visualization library, many other libraries build on top of it. Other libraries like Pandas and Seaborn allows many features of Matplotlib to use will few lines of code. More often it is criticized for not creating publication-quality graphs easily and quickly. Also, it has some default features, which gives 90s feel. They tried to overcome this with Matplotlib 2.0 release.

Created By: John D. Hunter

Learn More here : [Matplotlib](https://matplotlib.org/)

**Seaborn**

Seaborn uses the features of Matplotlib to create large charts and graphs using few lines of code. It has advantage over Matplotlib on its default style and color schemes, which gives more modern feel. Important features of seaborn are -

* A dataset-oriented API for examining relationships between multiple variables
* Specialized support for using categorical variables to show observations or aggregate statistics
* Options for visualizing univariate or bivariate distributions and for comparing them between subsets of data
* Convenient views onto the overall structure of complex datasets

Created By: Michael Waskom

Learn More here: [Seaborn](http://seaborn.pydata.org/index.html" \t "_blank)

**ggplot**

ggplot is based on ggplot2, and R based package. It functions differently than Matplotlib, it lest you create multilayer component plot to represent a data completely. Analysis generally requires plotting multiple times, ggplot overcome this representation of code by implementing them using high-level API. This results in less time spend in plot creation and more time analyzing the graph. As per the creator, ggplot is not good fit for people needing to make highly customized visualization. It is highly integrated with Pandas, so it is advised to use DataFrame when using ggplot.

Created By: ŷhat

learn More here: [ggplot](http://ggplot.yhathq.com/how-it-works.html" \t "_blank)

**Bokeh**

Like ggplot it is also based on *Grammar of Graphics*. But unlike ggplot, it is more related to Python instead of R. It has great advantages over abilities to build very interactive, web-friendly plots, which we can easily export as HTML docs or JSON objects. It also supports real-time feed and streaming of data. Bokeh provides 3 level of interface according to user's need. The top level is to create quick charts, which includes functions for plotting bar, box plots and histograms. The center level has similar features as Matplotlib, which allows to control the building blocks of charts. The lowest level is oriented towards programmer and developers.

Created by: Continuum Analytics

Learn More here: [Bokeh](https://docs.bokeh.org/en/latest/)

**plotly**

It is an online platform for data visualization, but you can access its features from python notebook as well. Like Bokeh, it also has advantages of creating interactive plots, but it offers some other charts which you won't find in other libraries, like, contour plots, dendograms and 3D charts.

Created by: Plotly

Learn More here: [plotly](https://plotly.com/python/)

***References:***

<https://analyticsindiamag.com/top-5-python-libraries-for-data-visualization/>

<https://www.fusioncharts.com/blog/best-python-data-visualization-libraries/>

<https://mode.com/blog/python-data-visualization-libraries/>

<https://www.analyticsvidhya.com/blog/2020/03/6-data-visualization-python-libraries/>

Bottom of Form