The following list provides a few plotting libraries for you to get started based on their use case(s).  This list is focused on providing a few solid options for each case rather than overwhelming you with the variety of options available.

The foundation: Matplotlib, most used plotting library, best for two-dimensional non-interactive plots. A possible replacement is pygal, it provides similar functionality but generates vector graphics SVG output and has a more user-friendly interface.

Specific use cases:

* Specialized **statistical plots**, like automatically fitting a linear regression with confidence interval or like scatter plots color-coded by category.
  + [seaborn](https://seaborn.pydata.org/tutorial): it builds on top of Matplotlib and it can also be used as a replacement for matplotlib just for an easier way to specify color palettes and **plotting aestetics**
* **Grammar of graphics plotting**, if you find the interface of Matplotlib too verbose, Python provides packages based on a different paradigm of plot syntax based on R's ggplot2
  + [ggplot](http://ggplot.yhathq.com/): it provides similar functionality to Matplotlib and is also based on Matplotlib but provides a different interface.
  + [altair](https://altair-viz.github.io/): it has a simpler interface compared to ggplot and generates Javascript based plots easily embeddable into the Jupyter Notebook or exported as PNG.
* **Interactive plots**, i.e. pan, zoom that work in the Jupyter Notebooks but also can be exported as Javascript to work standalone on a webpage.
  + [bokeh](http://bokeh.pydata.org/en/latest/docs/user_guide/quickstart.html): maintained by Continuum Analytics, the company behind Anaconda
  + [plotly](https://plot.ly/python/line-charts/#simple-line-plot): is both a library and a cloud service where you can store and share your visualizations (it has free/paid accounts)
* **Interactive map visualization**

\*[folium](https://github.com/python-visualization/folium): Creates HTML pages that include the Leaflet.js javascript plotting library to display data on top of maps. \*[plotly](https://plot.ly/python/choropleth-maps/): it supports color-coded country/world maps embedded in the Jupyter Notebook.

* **Realtime plots** that update with streaming data, even integrated in a dashboard with user interaction.
  + [bokeh plot server](http://bokeh.pydata.org/en/latest/docs/user_guide/server.html#userguide-server): it is part of Bokeh but requires to launch a separate Python process that takes care of responding to events from User Interface or from streaming data updates.
* **3D plots** are not easy to interpret, it is worth first consider if a combination of 2D plots could provide a better insight into the data
  + [mplot3d](https://matplotlib.org/mpl_toolkits/mplot3d/tutorial.html#mplot3d-tutorial): Matplotlib tookit for 3D visualization