Chapter 9: Data Visualization

Data visualization in python is perhaps one of the most utilized features for data science with python. The libraries in python come with lots of different features that enable users to make highly customized, elegant, and interactive plots.

In today's world, a lot of data is being generated on a daily basis. And sometimes to analyze this data for certain trends, patterns may become difficult if the data is in its raw format. To overcome this data visualization comes into play. Data visualization provides a good, organized pictorial representation of the data which makes it easier to understand, observe, analyze.

Packages for visualizations in python

Matplotlib

Matplotlib is a visualization library in Python for 2D plots of arrays. Matplotlib is written in Python and makes use of the NumPy library. It can be used in Python and IPython shells, Jupyter notebook, and web application servers. Matplotlib comes with a wide variety of plots like line, bar, scatter, histogram, etc. which can help us, deep-dive, into understanding trends, patterns, correlations. It was introduced by John Hunter in 2002.

Seaborn

Seaborn is a dataset-oriented library for making statistical representations in Python. It is developed atop matplotlib and to create different visualizations. It is integrated with pandas' data structures. The library internally performs the required mapping and aggregation to create informative visuals It is recommended to use a Jupyter/IPython interface in matplotlib mode.

Bokeh

Bokeh is an interactive visualization library for modern web browsers. It is suitable for large or streaming data assets and can be used to develop interactive plots and dashboards. There is a wide array of intuitive graphs in the library which can be leveraged to develop solutions. It works closely with PyData tools. The library is well-suited for creating customized visuals according to required use-cases. The visuals can also be made interactive to serve a what-if scenario model.

Importance of Data Visualization

- ***Data Visualization Discovers the Trends in Data***

  The most important thing that data visualization does is discover the trends in data. After all, it is much easier to observe data trends when all the data is laid out in front of you in a visual form as compared to data in a table.

- ***Data Visualization Provides a Perspective on the Data***

  Data Visualization provides a perspective on data by showing its meaning in the larger scheme of things. It demonstrates how particular data references stand with respect to the overall data picture.

- ***Data Visualization Puts the Data into the Correct Context***

  It is very difficult to understand the context of the data with data visualization. Since context provides the whole circumstances of the data, it is very difficult to grasp by just reading numbers in a table.

- ***Data Visualization Saves Time***

  It is definitely faster to gather some insights from the data using data visualization rather than just studying a chart.

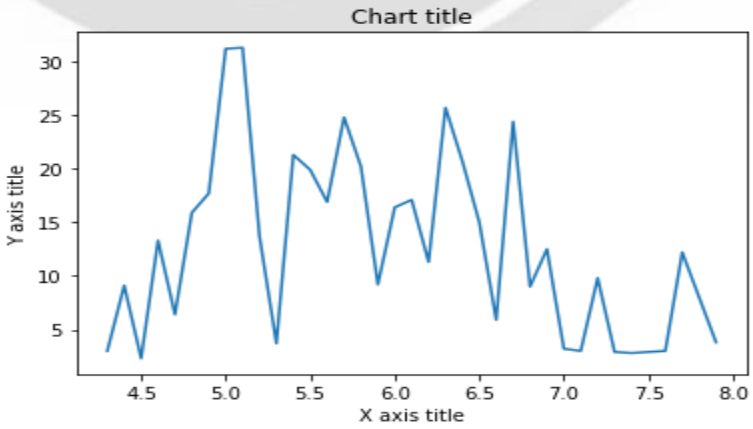
- ***Data Visualization Tells a Data Story***

  Data visualization is also a medium to tell a data story to the viewers. The visualization can be used to present the data facts in an easy-to-understand form while telling a story and leading the viewers to an inevitable conclusion. This data story, like any other type of story, should have a good beginning, a basic plot, and an ending that it is leading towards. For example, if a data analyst has to craft a data visualization for company executives detailing the profits on various products, then the data story can start with the profits and losses of various products and move on to recommendations on how to tackle the losses.

Types of Visualization Plots

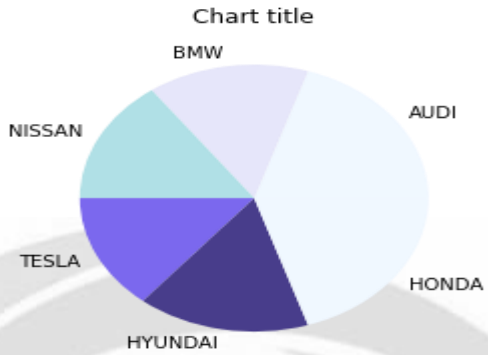
Line chart

A line chart is used for the representation of continuous data points. This visual can be effectively utilized when we want to understand the trend across time.



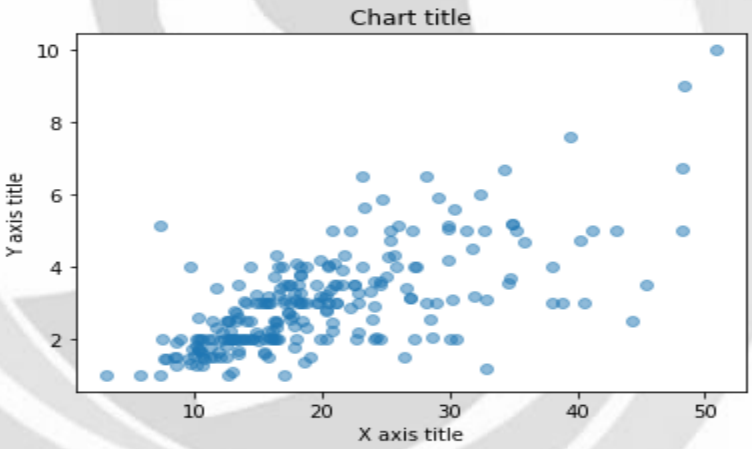
Pie Chart

Pie charts can be used to identify proportions of the different components in a given whole.



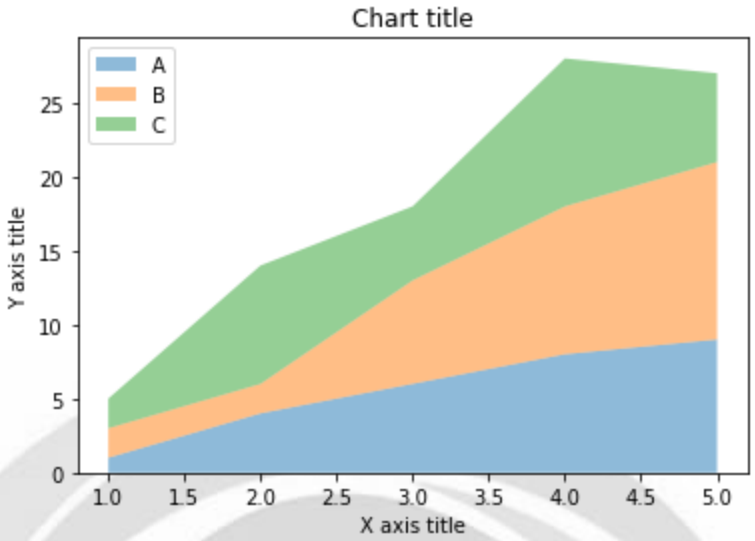
Scatter plot

Scatter plots can be leveraged to identify relationships between two variables. It can be effectively used in circumstances where the dependent variable can have multiple values for the independent variable.



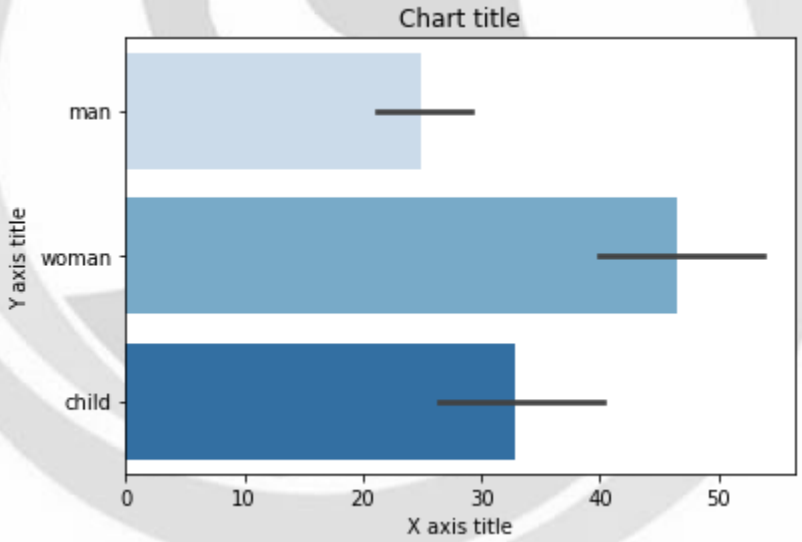
Area chart

Area charts are used to track changes over time for one or more groups. Area graphs are preferred over line charts when we want to capture the changes over time for more than 1 group.



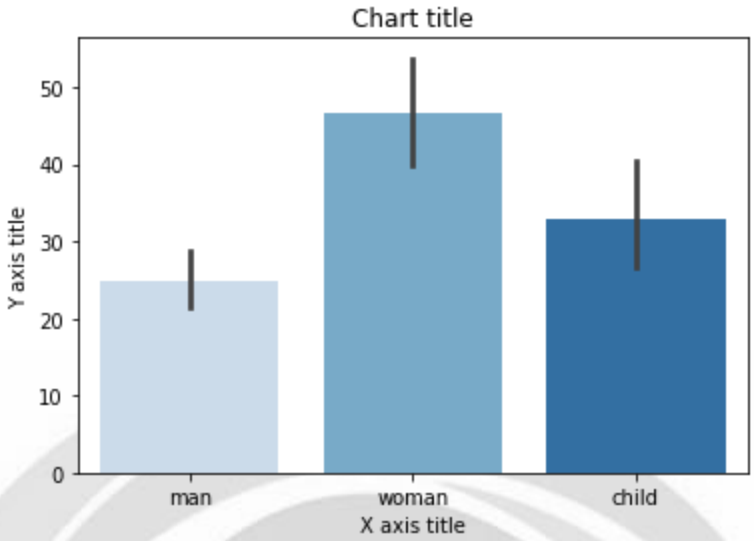
Bar chart

A bar chart is used when we want to compare metric values across different subgroups of the data. If we have a greater number of groups, a bar chart is preferred over a column chart.



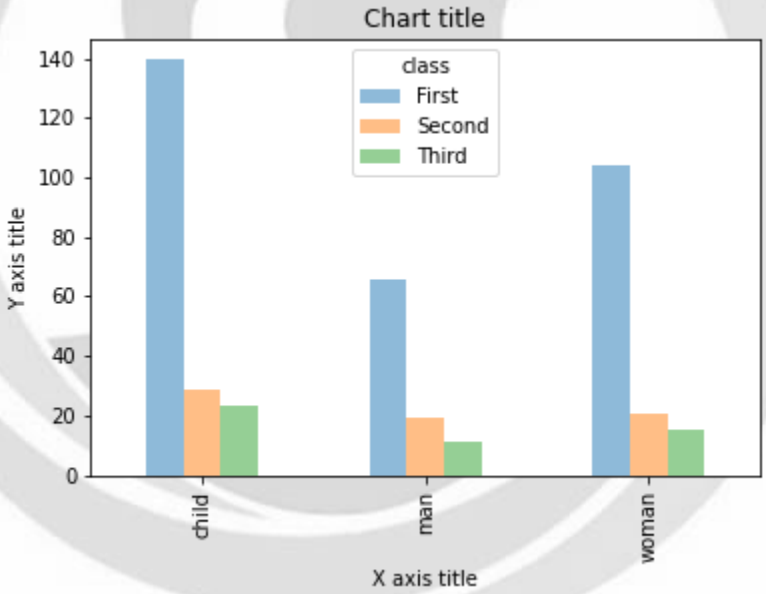
Column chart

Column charts are mostly used when we need to compare a single category of data between individual sub-items, for example, when comparing revenue between regions.



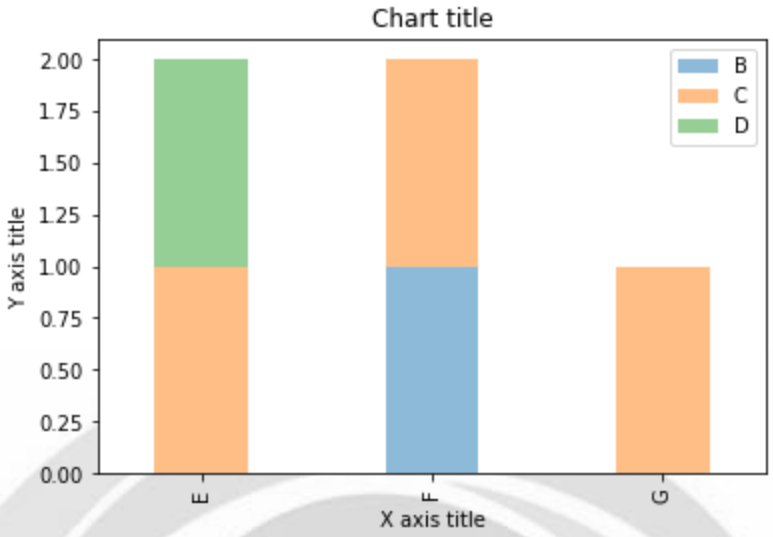
Grouped bar chart

A grouped bar chart is used when we want to compare the values in certain groups and sub-groups



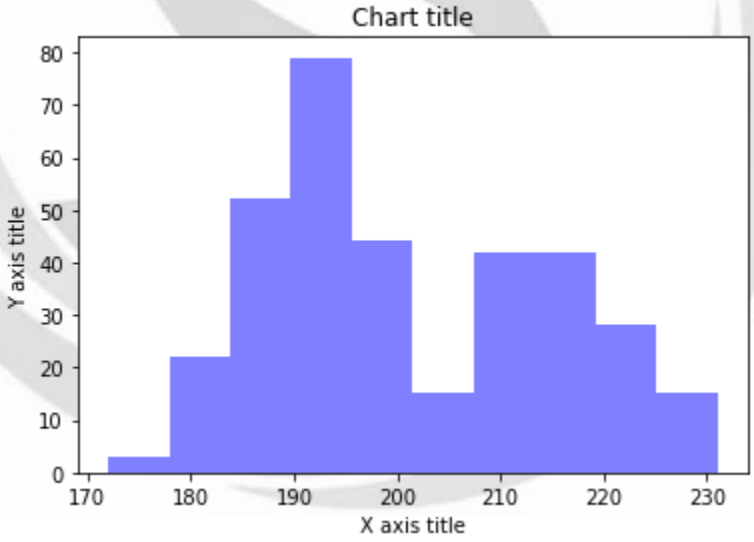
Stacked bar chart

A stacked bar chart is used when we want to compare the total sizes across the available groups and the composition of the different sub-groups



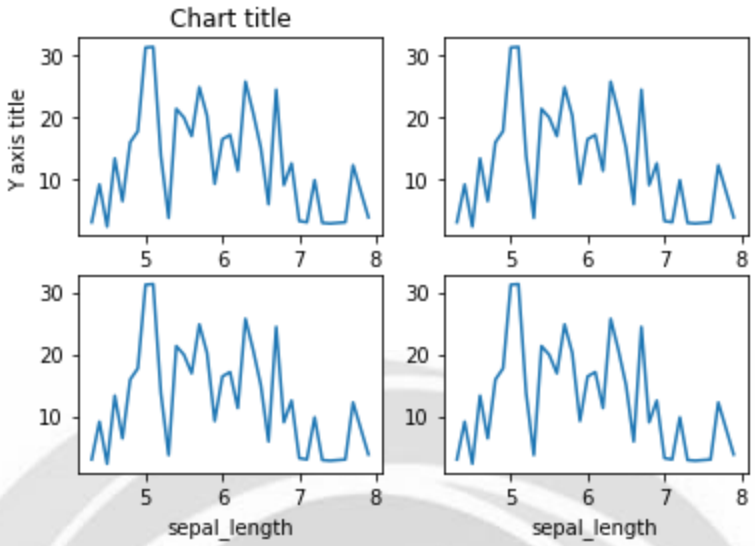
Column histogram

Column histograms are used to observe the distribution for a single variable with few data points.



Subplots

Subplots are powerful visualizations that help easy comparisons between plots



Introduction to Plotly

Plotly.py is an interactive, open-source, high-level, declarative, and browser-based visualization library for Python. It holds an array of useful visualization which includes scientific charts, 3D graphs, statistical charts, financial charts among others. Plotly graphs can be viewed in Jupyter notebooks, standalone HTML files, or hosted online. Plotly library provides options for interaction and editing. The robust API works perfectly in both local and web browser mode.

Plotly is a Montreal based technical computing company involved in development of data analytics and visualization tools such as Dash and Chart Studio. It has also developed open-source graphing Application Programming Interface (API) libraries for Python, R, MATLAB, Javascript and other computer programming languages.

Some of the important features of Plotly are as follows:

 It produces interactive graphs.

The graphs are stored in JavaScript Object Notation (JSON) data format so that they can be read using scripts of other programming languages such as R, Julia, MATLAB etc.

 Graphs can be exported in various raster as well as vector image formats

Installation of Plotly:

Now we can install plotly's Python package as given below using pip utility.

Syntax -! **pip install plotly**

Settings for online plotting:

Data and graph of online plot are saved in your plotly account. Online plots are generated by two methods both of which create a unique url for the plot and save it in your Plotly account.

 py.plot() : returns the unique url and optionally open the url.

 py.iplot() : when working in a Jupyter Notebook to display the plot in the

Setting for offline plotting:

Plotly allows you to generate graphs offline and save them in local machine. The plotly.offline.plot() function creates a standalone HTML that is saved locally and opened inside your web browser.

Use plotly.offline.iplot() when working offline in a Jupyter Notebook to display the plot in the notebook. In order to display the plot inside the notebook, you need to initiate plotly's notebook mode as follows:

from plotly.offline import init_notebook_mode

init_notebook_mode(connected=True)

Different kind of plots in plotly:

- Bar Graphs
- Hist plot
- Pie Chart
- Tables
- Scatter plot
- Scattergl plot
- Bubble Chats
- Violin plot
- Counter plot
- Box plot
- Plotly 3D Scatter plot and sub plots

Advantages of using Plotly:

- **Interactivity -** Plotly provides a feature that no other visualization library has — interactivity. This allows users to interact with graphs on display, allowing for a better storytelling experience. Zooming in and out, point value display, panning graphs, you name it, they have it.

- **Customization and Flexibility** - Since Plotly is plotted based on Pandas, you can easily perform complex transformations to your data before plotting it.

- **Range and Aesthetics:** Besides being able to plot **all Matplotlib and Seaborn charts**, Plotly offers a huge range of graphs and charts like —

  Statistical Charts which include but are not limited to Parallel Categories and Probability Tree Plots
  Scientific Charts you never thought of, ranging from Network Graphs to Radar Charts
  Financial Charts which are useful for Time-Series Analysis, examples include Candlesticks, Funnels and Bullet Chart.

Introduction to Cufflinks

Cufflinks connect Plotly with pandas to create graphs and charts of Dataframes directly. It's a Python library which is used to design graphs, especially interactive graphs. It can plot various graphs and charts like histogram, bar plot, boxplot, spread plot and many more. It is mainly used in data analysis as well as financial analysis. Cufflinks is an interactive visualization library which you can use to blow your audience away.

So, for our visualization we'll be using a wrapper on Plotly called Cufflinks designed to work with Pandas Dataframes. So, our entire stack is cufflinks > plotly > plotly.js > d3.js which means we get the efficiency of coding in Python with the incredible interactive graphics capabilities.

Advantages of Cufflinks over other plotting libraries:

Plotly and Cufflinks offer great benefits such as:

- Dynamic Plots
- Hosting Service
- Requires only a single line of code to make plots
- It works 100% offline

Installing Cuffflinks:

To use cufflinks into our Jupiter notebook first we have to install it. To install cufflinks, we have the below syntax:

pip install cufflinks

Then we have to import the required modules:

```
In [31]: #importing modules
         import plotly #importing the plotly library
         import cufflinks as cf #importing the cufflinks library
         cf.go_offline()  #This lets us make our plots offline
         import pandas as pd #importing the Data analysis tool
         %matplotlib inline
```

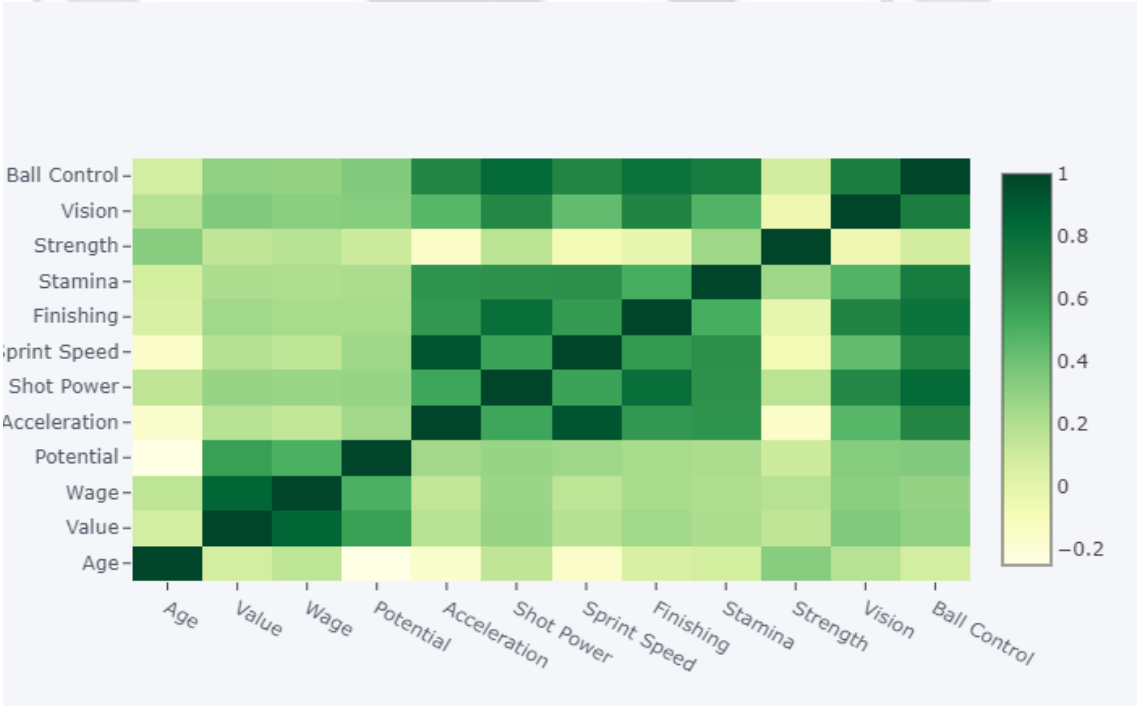
Note: The cf.go_offline() function allows you to make your plots offline which means that your plots are not saved online on your plotly account. The benefits of making the plots offline is that you can make changes to your plots in your notebook before you choose to save them. Also, to save your plots online you need to create a plotly account. You can create one here

To switch back to online mode you can use the syntax below:

```
cf.go_online() # switch back to online mode, where graphs are saved on your online plotly account
```

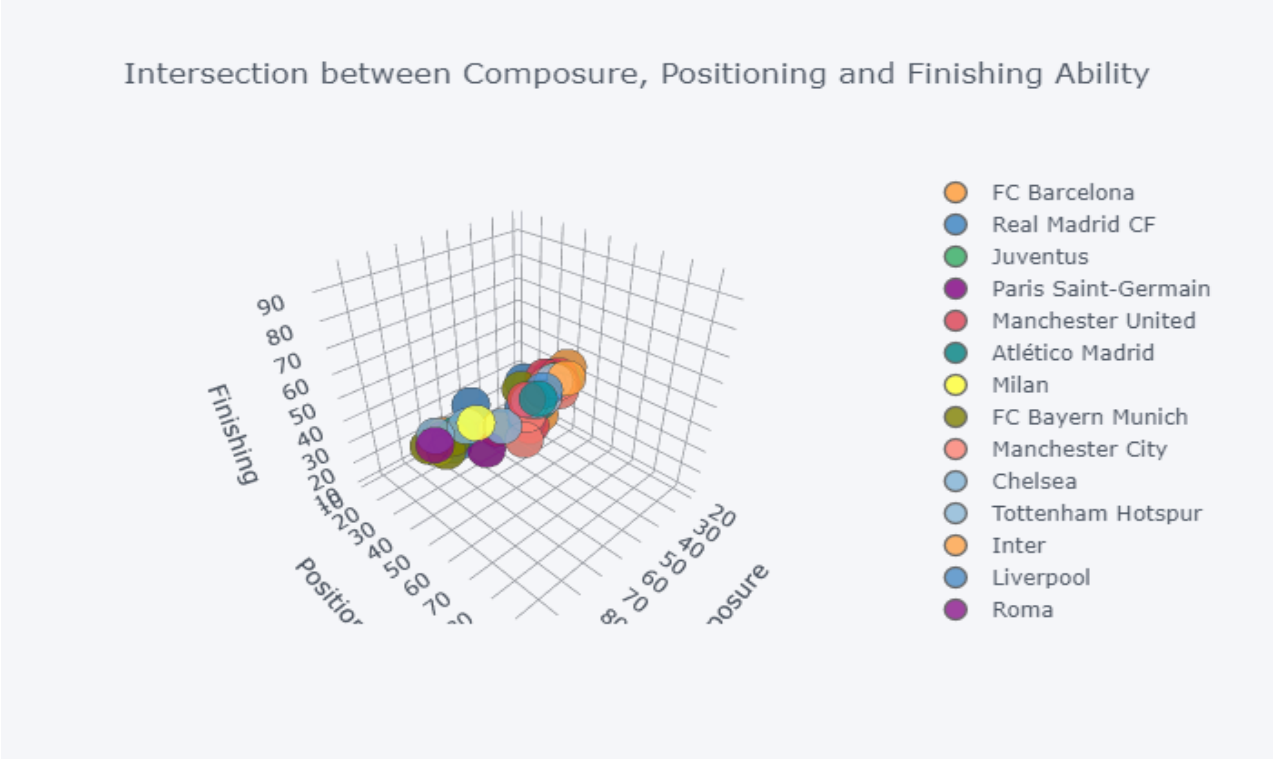
More advanced plots:

- **HeatMaps:** To visualize the correlations between numerical values.



- **Geographic Plottling – Chloropleth:** With plotly you can all plot geographical data. Now, while not as effective as geo-spatial plotting with folium a leaflet.js wrapper for python that generates html interactive maps, pandas & cufflinks works just fine for our purposes.

3D Plots



It is a pretty awesome tool for quick-fire EDA on any dataframe. I reckon it is the best plotting library if you're working with python, not only for its ease of use for you, but also in terms of bringing data to life for the reader.