

Business Intelligence

Plotly

Hrant Davtyan

Content

- What are the Python visualization libraries?
- Introduction to Plotly
 - Structure
 - Basic plots
 - Pandas
 - Figure Factory
 - Subplots
 - Maps

Visualization libraries

Frontend

Python is abundant with visualization libraries. Here are some:

- matplotlib – the default choice, based on MatLab plots, low level
- seaborn – for statisticians, based on matplotlib, more user friendly
- ggplot – for R users,
- bokeh – for interactivity and interfaces
- plotly – default choice for interactive plotting, available also in R

Plotly

Plotting

Plotly has several options for plotting:

- Online
 - Inside Jupyter Notebook (`from plotly.plotly import iplot`)
 - Not in Jupyter Notebook (`from plotly.plotly import plot`)
- **Offline**
 - **Inside Jupyter Notebook** (`from plotly.offline import iplot`)
 - Not in Jupyter Notebook (`from plotly.offline import plot`)

In case of plotting inside the Notebook (using iplot), the following line of code must be executed, to show the plot inline:

- `from plotly.offline import init_notebook_mode`
- `init_notebook_mode(connected=True)`

Objects

Most of the graphing functions (such as Scatter, Histogram etc) in Plotly are stored in **Graph objects**, which is usually imported as **go**:

- **import** plotly.graph_objs **as** go

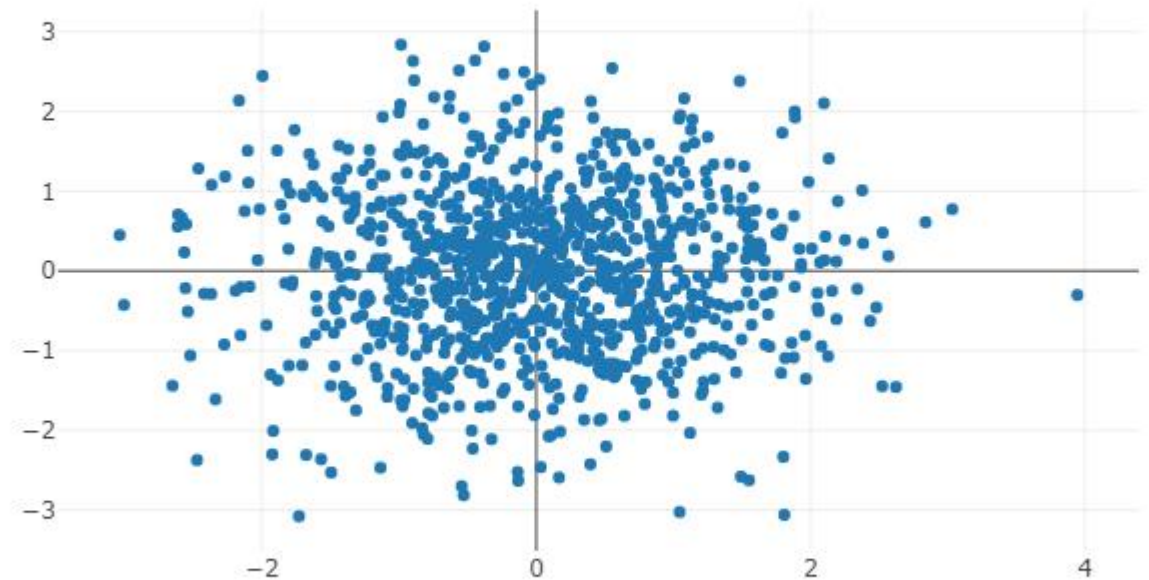
Yet, there are still some in other locations, such as **distribution plot**, which is located in **figure factory**:

- **import** plotly.figure_factory **as** ff

Basic plots

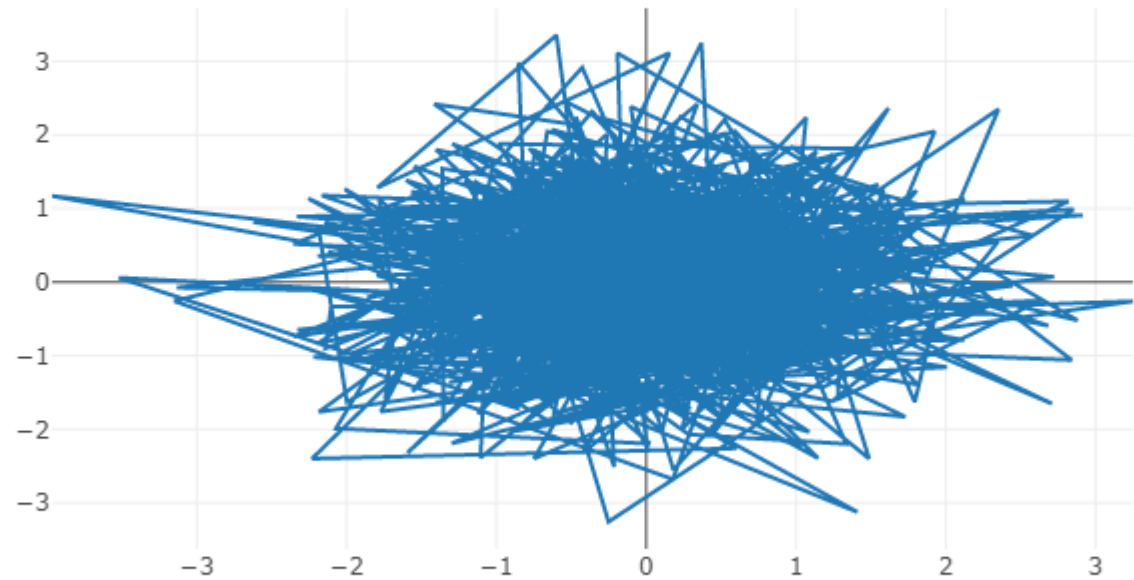
Scatterplot

```
1 from plotly.offline import plot, iplot
2 import plotly.graph_objs as go
3
4 import numpy as np
5
6 N = 1000
7 random_x = np.random.randn(N)
8 random_y = np.random.randn(N)
9
10 trace = go.Scatter(
11     x = random_x,
12     y = random_y,
13     mode = 'markers',
14     name = 'markers'
15 )
16
17 data = [trace]
18 iplot(data)
```



Line plot

```
1 from plotly.offline import plot, iplot
2 import plotly.graph_objs as go
3
4 import numpy as np
5
6 N = 1000
7 random_x = np.random.randn(N)
8 random_y = np.random.randn(N)
9
10 trace = go.Scatter(
11     x = random_x,
12     y = random_y,
13     mode = 'lines',
14     name = 'lines'
15 )
16
17 data = [trace]
18 iplot(data)
```



Layout

```
1 from plotly.offline import plot, iplot
2 import plotly.graph_objs as go
3
4 import numpy as np
5
6 N = 1000
7 random_x = np.random.randn(N)
8 random_y = np.random.randn(N)
9
10 trace = go.Scatter(
11     x = random_x,
12     y = random_y,
13     mode = 'lines',
14     name = 'lines'
15 )
16
17 data = [trace]
18
19 layout = dict(title = "Scatter",
20               xaxis = dict(zeroline=False),
21               yaxis = dict(zeroline=False)
22               )
23 figure = dict(data=data,layout=layout)
24
25 iplot(figure)
```

