# Pandas In-built Visualization

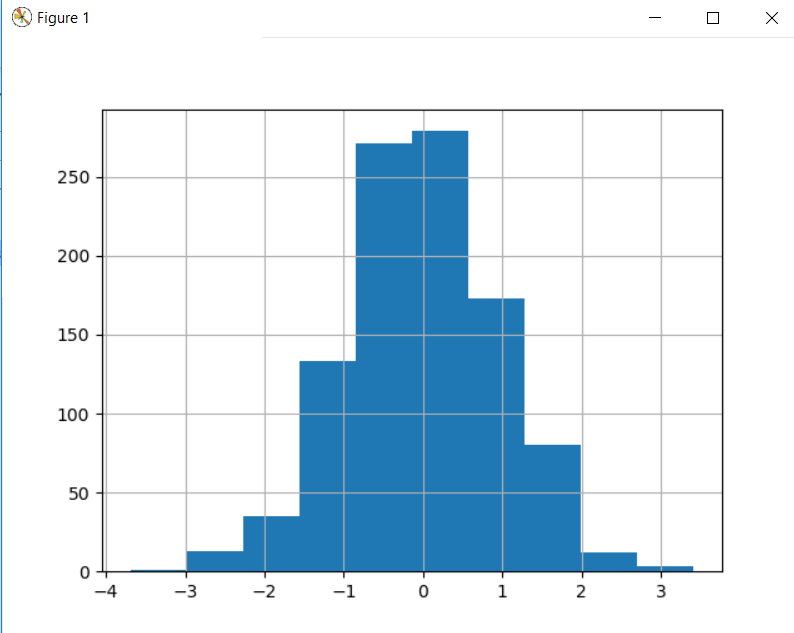
we will learn about pandas built-in capabilities for data visualization! It's built-off of matplotlib, but it baked into pandas for easier usage!

## Style Sheets

Matplotlib has [style sheets](http://matplotlib.org/gallery.html#style_sheets) you can use to make your plots look a little nicer. These style sheets include plot\_bmh,plot\_fivethirtyeight,plot\_ggplot and more. They basically create a set of style rules that your plots follow. I recommend using them, they make all your plots have the same look and feel more professional. You can even create your own if you want your company's plots to all have the same look (it is a bit tedious to create on though).

We can directly plot histograms using pandas as –

import numpy as np  
import pandas as pd  
import matplotlib.pyplot as plt  
df1 = pd.read\_csv("E:/Py-DS-ML-Bootcamp-master/Refactored\_Py\_DS\_ML\_Bootcamp-master/07-Pandas-Built-in-Data-Viz/df1.csv",index\_col=0)  
df2 = pd.read\_csv("E:/Py-DS-ML-Bootcamp-master/Refactored\_Py\_DS\_ML\_Bootcamp-master/07-Pandas-Built-in-Data-Viz/df2.csv")  
df1["A"].hist()  
plt.show()

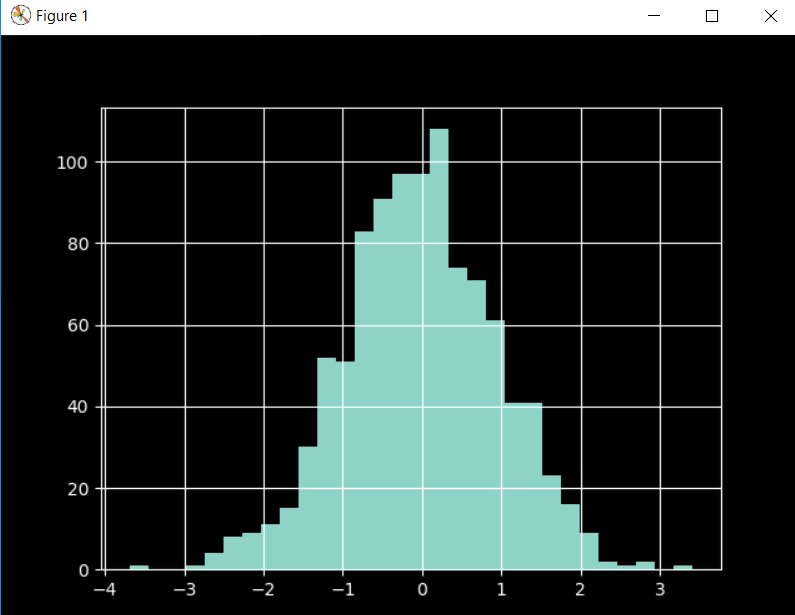


We can add matplotlib’s style arguments “bins” since its calling matplotlib behind the scenes-

df1["A"].hist(bins=30)

We can also use plt.style.use() to specify styles

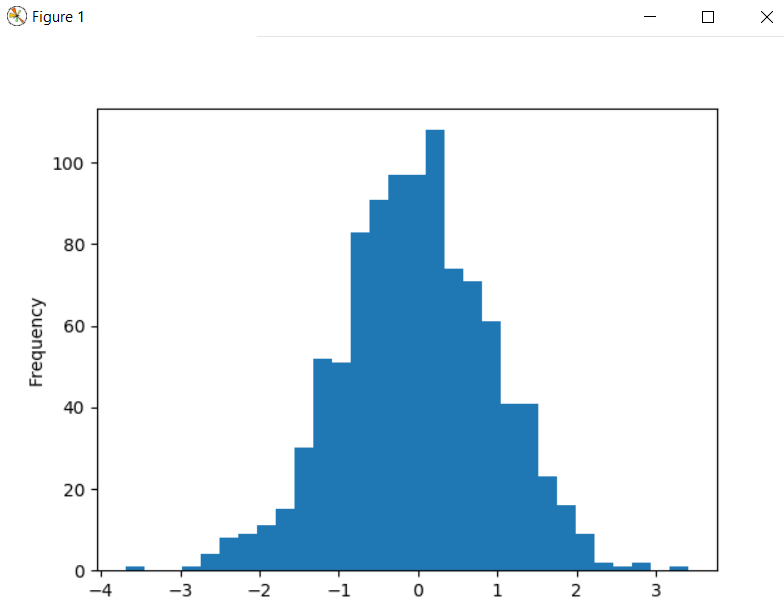
plt.style.use("dark\_background")



## **df.plot(kind=””) way**

One way of drawing plots using pandas is to use plot method and specify the kind argument about the type of plot which is required.

import seaborn as sns  
import numpy as np  
import pandas as pd  
import matplotlib.pyplot as plt  
df1 = pd.read\_csv("E:/Py-DS-ML-Bootcamp-master/Refactored\_Py\_DS\_ML\_Bootcamp-master/07-Pandas-Built-in-Data-Viz/df1.csv",index\_col=0)  
df2 = pd.read\_csv("E:/Py-DS-ML-Bootcamp-master/Refactored\_Py\_DS\_ML\_Bootcamp-master/07-Pandas-Built-in-Data-Viz/df2.csv")  
df1["A"].plot(kind="hist",bins=30)  
plt.show()



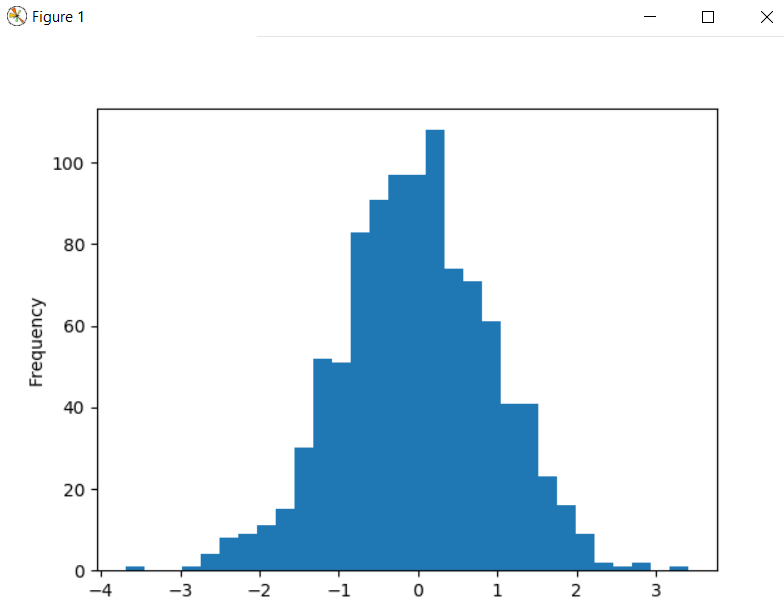
## Df.plot.type

There are several plot types built-in to pandas, most of them statistical plots by nature:

* df.plot.area
* df.plot.barh
* df.plot.density
* df.plot.hist
* df.plot.line
* df.plot.scatter
* df.plot.bar
* df.plot.box
* df.plot.hexbin
* df.plot.kde
* df.plot.pie

### Hist

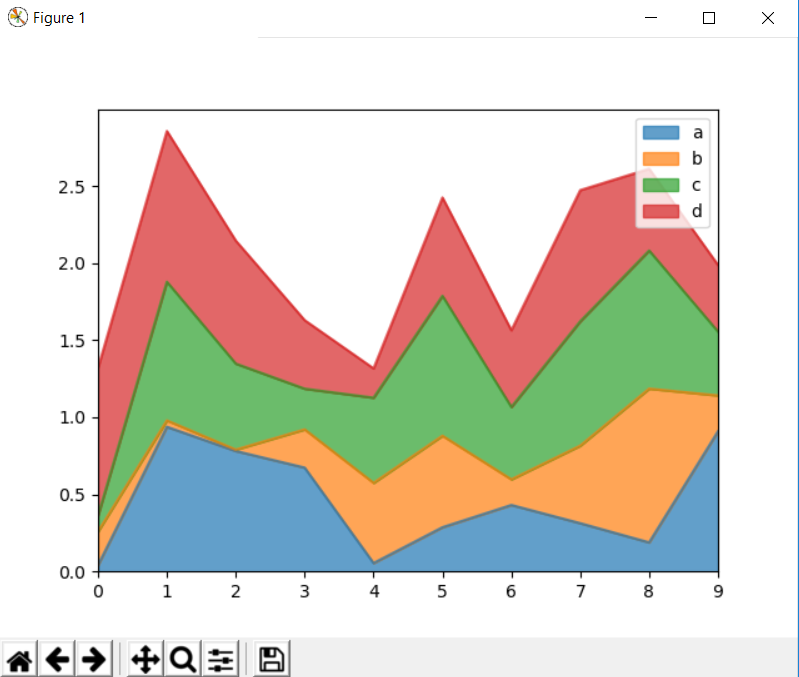
import seaborn as sns  
import numpy as np  
import pandas as pd  
import matplotlib.pyplot as plt  
df1 = pd.read\_csv("E:/Py-DS-ML-Bootcamp-master/Refactored\_Py\_DS\_ML\_Bootcamp-master/07-Pandas-Built-in-Data-Viz/df1.csv",index\_col=0)  
df2 = pd.read\_csv("E:/Py-DS-ML-Bootcamp-master/Refactored\_Py\_DS\_ML\_Bootcamp-master/07-Pandas-Built-in-Data-Viz/df2.csv")  
df1["A"].plot.hist(bins=30)  
plt.show()



### Area

Alpha argument is for transparency. It signifies the area under the line connected by various data points.

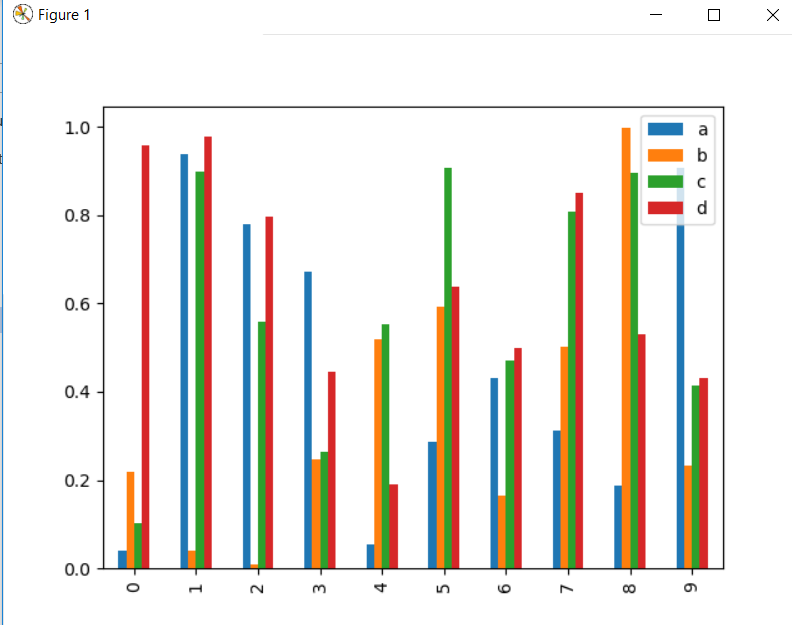
import seaborn as sns  
import numpy as np  
import pandas as pd  
import matplotlib.pyplot as plt  
df1 = pd.read\_csv("E:/Py-DS-ML-Bootcamp-master/Refactored\_Py\_DS\_ML\_Bootcamp-master/07-Pandas-Built-in-Data-Viz/df1.csv",index\_col=0)  
df2 = pd.read\_csv("E:/Py-DS-ML-Bootcamp-master/Refactored\_Py\_DS\_ML\_Bootcamp-master/07-Pandas-Built-in-Data-Viz/df2.csv")  
df2.plot.area(alpha=0.7)  
plt.show()



### Bar

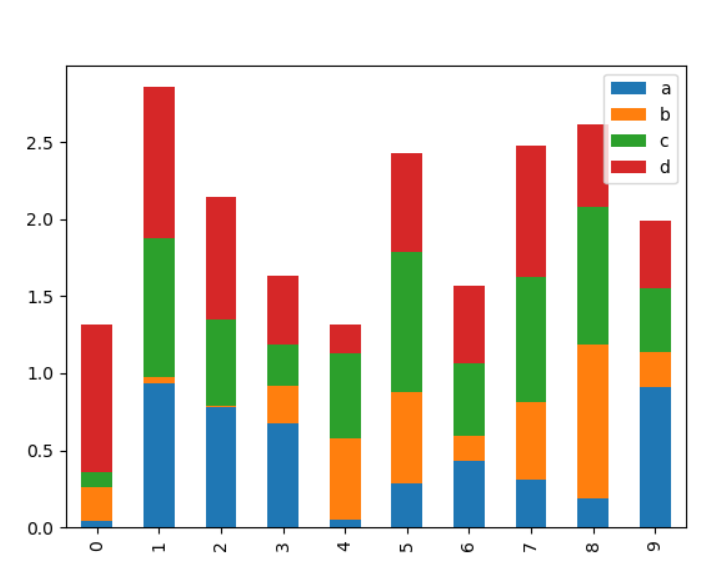
It will plot a barplot based on index column if your index is categorical.

import seaborn as sns  
import numpy as np  
import pandas as pd  
import matplotlib.pyplot as plt  
df1 = pd.read\_csv("E:/Py-DS-ML-Bootcamp-master/Refactored\_Py\_DS\_ML\_Bootcamp-master/07-Pandas-Built-in-Data-Viz/df1.csv",index\_col=0)  
df2 = pd.read\_csv("E:/Py-DS-ML-Bootcamp-master/Refactored\_Py\_DS\_ML\_Bootcamp-master/07-Pandas-Built-in-Data-Viz/df2.csv")  
df2.plot.bar()  
plt.show()



We can specify stacking to be true also.

import seaborn as sns  
import numpy as np  
import pandas as pd  
import matplotlib.pyplot as plt  
df1 = pd.read\_csv("E:/Py-DS-ML-Bootcamp-master/Refactored\_Py\_DS\_ML\_Bootcamp-master/07-Pandas-Built-in-Data-Viz/df1.csv",index\_col=0)  
df2 = pd.read\_csv("E:/Py-DS-ML-Bootcamp-master/Refactored\_Py\_DS\_ML\_Bootcamp-master/07-Pandas-Built-in-Data-Viz/df2.csv")  
df2.plot.bar(stacked=True)  
plt.show()

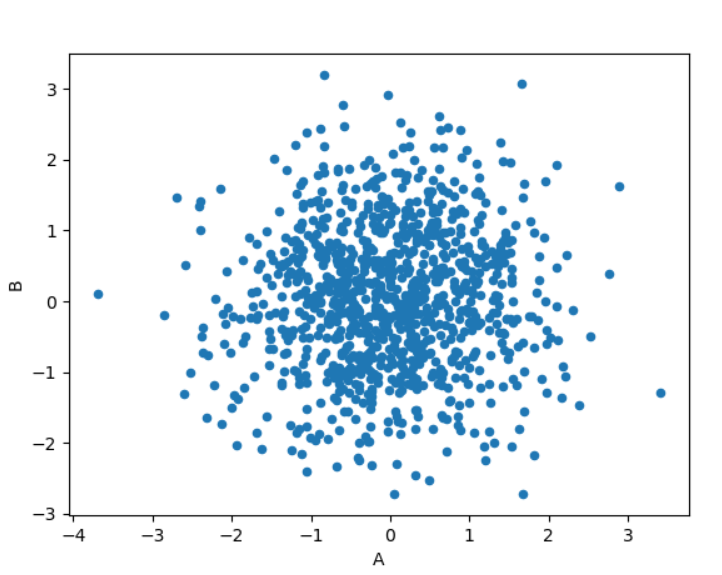


### Line

import seaborn as sns  
import numpy as np  
import pandas as pd  
import matplotlib.pyplot as plt  
df1 = pd.read\_csv("E:/Py-DS-ML-Bootcamp-master/Refactored\_Py\_DS\_ML\_Bootcamp-master/07-Pandas-Built-in-Data-Viz/df1.csv",index\_col=0)  
df2 = pd.read\_csv("E:/Py-DS-ML-Bootcamp-master/Refactored\_Py\_DS\_ML\_Bootcamp-master/07-Pandas-Built-in-Data-Viz/df2.csv")  
df1.plot.line(x=df1.index,y='B',figsize=(12,3),lw=1)  
plt.show()

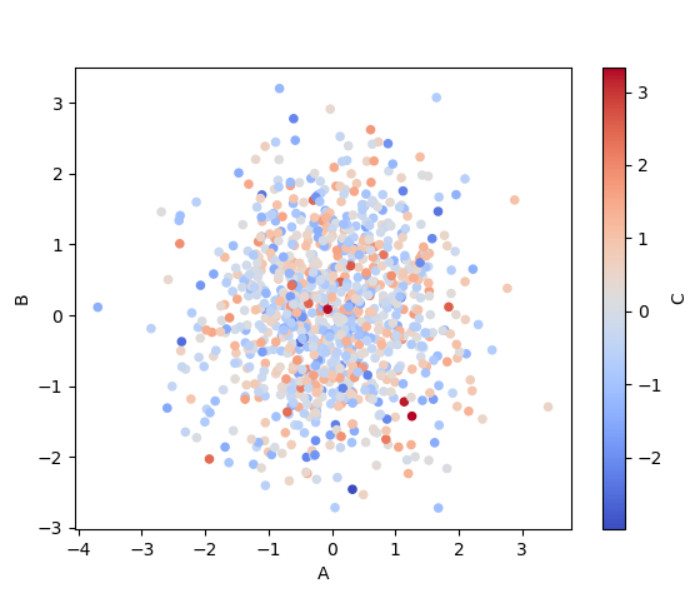
### Scatter

import seaborn as sns  
import numpy as np  
import pandas as pd  
import matplotlib.pyplot as plt  
df1 = pd.read\_csv("E:/Py-DS-ML-Bootcamp-master/Refactored\_Py\_DS\_ML\_Bootcamp-master/07-Pandas-Built-in-Data-Viz/df1.csv",index\_col=0)  
df2 = pd.read\_csv("E:/Py-DS-ML-Bootcamp-master/Refactored\_Py\_DS\_ML\_Bootcamp-master/07-Pandas-Built-in-Data-Viz/df2.csv")  
df1.plot.scatter(x='A',y='B')  
plt.show()



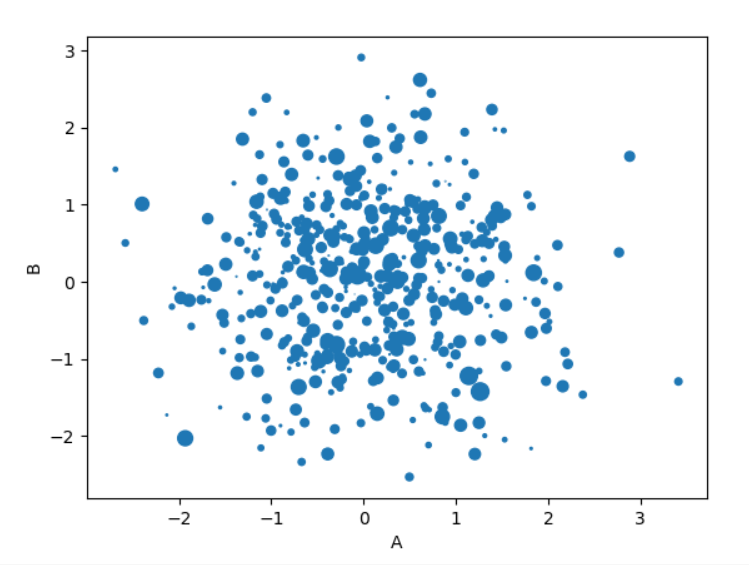
We can also show another column either by colour or by size. First we will see by colour

import seaborn as sns  
import numpy as np  
import pandas as pd  
import matplotlib.pyplot as plt  
df1 = pd.read\_csv("E:/Py-DS-ML-Bootcamp-master/Refactored\_Py\_DS\_ML\_Bootcamp-master/07-Pandas-Built-in-Data-Viz/df1.csv",index\_col=0)  
df2 = pd.read\_csv("E:/Py-DS-ML-Bootcamp-master/Refactored\_Py\_DS\_ML\_Bootcamp-master/07-Pandas-Built-in-Data-Viz/df2.csv")  
df1.plot.scatter(x='A',y='B',c="C",cmap="coolwarm")  
plt.show()



If we specify another column by size, then we need to pass it as a dataframe column, the bigger dots are depicting values from C column.

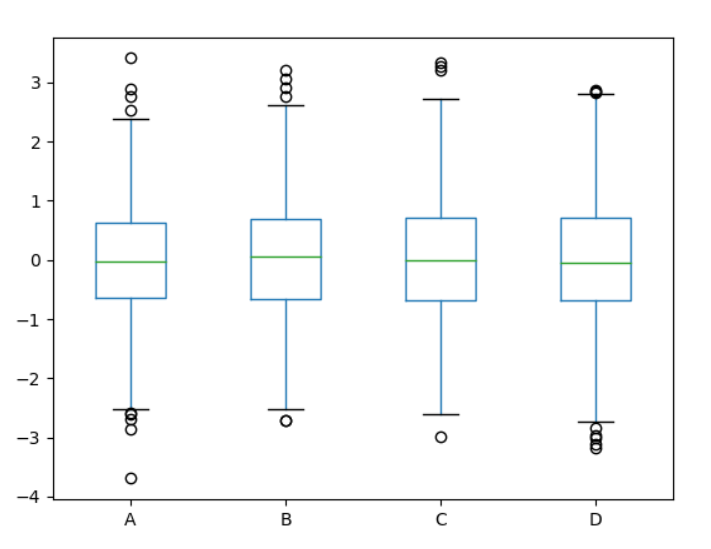
import seaborn as sns  
import numpy as np  
import pandas as pd  
import matplotlib.pyplot as plt  
df1 = pd.read\_csv("E:/Py-DS-ML-Bootcamp-master/Refactored\_Py\_DS\_ML\_Bootcamp-master/07-Pandas-Built-in-Data-Viz/df1.csv",index\_col=0)  
df2 = pd.read\_csv("E:/Py-DS-ML-Bootcamp-master/Refactored\_Py\_DS\_ML\_Bootcamp-master/07-Pandas-Built-in-Data-Viz/df2.csv")  
df1.plot.scatter(x='A',y='B',s=df1["C"]\*30)  
plt.show()



### BoxPlots

# Can also pass a by= argument for groupby

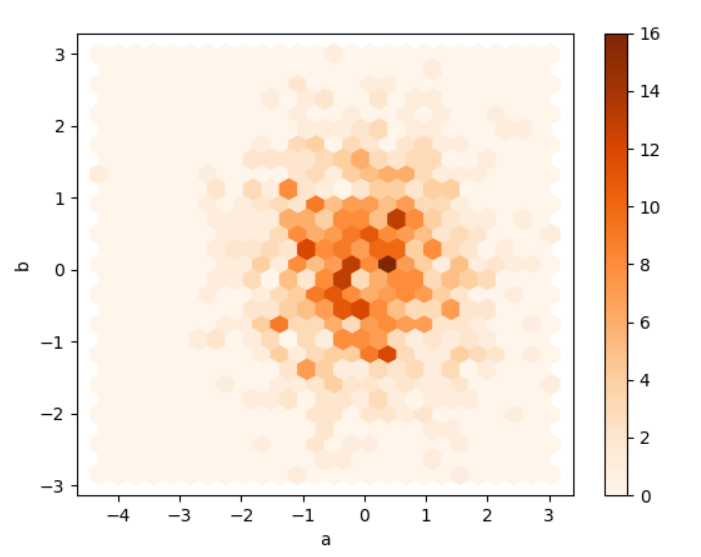
import seaborn as sns  
import numpy as np  
import pandas as pd  
import matplotlib.pyplot as plt  
df1 = pd.read\_csv("E:/Py-DS-ML-Bootcamp-master/Refactored\_Py\_DS\_ML\_Bootcamp-master/07-Pandas-Built-in-Data-Viz/df1.csv",index\_col=0)  
df2 = pd.read\_csv("E:/Py-DS-ML-Bootcamp-master/Refactored\_Py\_DS\_ML\_Bootcamp-master/07-Pandas-Built-in-Data-Viz/df2.csv")  
df1.plot.box()  
plt.show()



### Hexagonal Bin

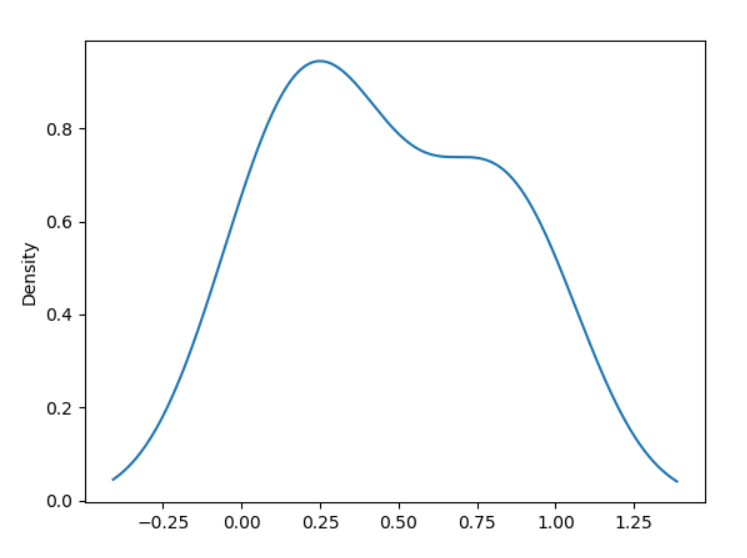
Useful for Bivariate Data, alternative to scatterplot. Its similar to scatterplot except it uses hexagons to show the values.

import seaborn as sns  
import numpy as np  
import pandas as pd  
import matplotlib.pyplot as plt  
df1 = pd.read\_csv("E:/Py-DS-ML-Bootcamp-master/Refactored\_Py\_DS\_ML\_Bootcamp-master/07-Pandas-Built-in-Data-Viz/df1.csv",index\_col=0)  
df2 = pd.read\_csv("E:/Py-DS-ML-Bootcamp-master/Refactored\_Py\_DS\_ML\_Bootcamp-master/07-Pandas-Built-in-Data-Viz/df2.csv")  
df = pd.DataFrame(np.random.randn(1000, 2), columns=['a', 'b'])  
df.plot.hexbin(x='a',y='b',gridsize=25,cmap='Oranges')  
plt.show()



### KDE

import seaborn as sns  
import numpy as np  
import pandas as pd  
import matplotlib.pyplot as plt  
df1 = pd.read\_csv("E:/Py-DS-ML-Bootcamp-master/Refactored\_Py\_DS\_ML\_Bootcamp-master/07-Pandas-Built-in-Data-Viz/df1.csv",index\_col=0)  
df2 = pd.read\_csv("E:/Py-DS-ML-Bootcamp-master/Refactored\_Py\_DS\_ML\_Bootcamp-master/07-Pandas-Built-in-Data-Viz/df2.csv")  
df2["a"].plot.kde()  
plt.show()



Can also use .density() to plot kde and can be used on entire dataframe as well

df2.plot.density()

