

Introduction

The beauty of an art lies in the message it conveys. At times, reality is not what we see or perceive. The endless efforts from the likes of Vinci and Picasso have tried to bring people closer to the reality using their exceptional artworks on a certain topic/matter.

Data scientists are no less than artists. They make paintings in form of digital visualization (of data) with a motive of manifesting the hidden patterns / insights in it. It is even more interesting to know that, the tendency of human perception, cognition and communication increases when he / she gets exposed to visualized form of any content/data.

There are multiple tools for performing [visualization](https://www.analyticsvidhya.com/blog/2015/05/data-visualization-resource/) in data science. In this article, I have demonstrated various visualization charts using [Python](https://www.analyticsvidhya.com/learning-paths-data-science-business-analytics-business-intelligence-big-data/learning-path-data-science-python/).

What does it take to make visualization in Python?

Not much ! Python has already made it easy for you – with two exclusive libraries for visualization, commonly known as *matplotlib and seaborn.* Heard of them?

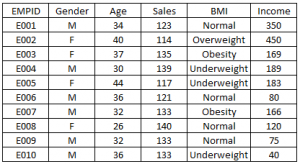
***Matplotlib*:** Python based plotting library offers *matplotlib* with a complete 2D support along with limited 3D graphic support. It is useful in producing publication quality figures in interactive environment across platforms. It can also be used for animations as well. To know more about this library, check this [link](http://matplotlib.org/).

***Seaborn*:** Seaborn is a library for creating informative and attractive statistical graphics in python. This library is based on matplotlib. Seaborn offers various features such as built in themes, color palettes, functions and tools to visualize univariate, bivariate, linear regression, matrices of data, statistical time series etc which lets us to build complex visualizations. To know more about this library, check this [link](http://stanford.edu/~mwaskom/software/seaborn/).

What are the different visualizations I can make?

Last week, [A comprehensive guide on Data Visualization](https://www.analyticsvidhya.com/blog/2015/05/data-visualization-resource/) was published to introduce you to the most commonly used visualizations techniques. We recommend you to refer that before proceeding further, in case you haven’t.

Below are the python codes with their output. I have used following data set to create these visualization:

[](https://www.analyticsvidhya.com/wp-content/uploads/2015/05/Data_Set.png)

Import Data Set:

import matplotlib.pyplot as plt

import pandas as pd

df=pd.read\_excel("E:/First.xlsx", "Sheet1")

Histogram :

fig=plt.figure() #Plots in matplotlib reside within a figure object, use plt.figure to create new figure

#Create one or more subplots using add\_subplot, because you can't create blank figure

ax = fig.add\_subplot(1,1,1)

#Variable

ax.hist(df['Age'],bins = 7) # Here you can play with number of bins

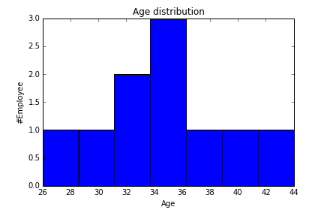
Labels and Tit

plt.title('Age distribution')

plt.xlabel('Age')

plt.ylabel('#Employee')

plt.show()

[](https://www.analyticsvidhya.com/wp-content/uploads/2015/05/Histogram1.png)

Box Plot

import matplotlib.pyplot as plt

import pandas as pd

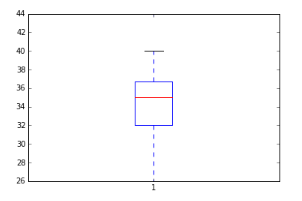
fig=plt.figure()

ax = fig.add\_subplot(1,1,1)

#Variable

ax.boxplot(df['Age'])

plt.show()

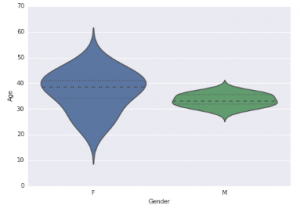
[](https://www.analyticsvidhya.com/wp-content/uploads/2015/05/Box_Plot_Violin_Plot1.png)

Violin Plot

import seaborn as sns

sns.violinplot(df['Age'], df['Gender']) #Variable Plot

sns.despine()

[](https://www.analyticsvidhya.com/wp-content/uploads/2015/05/Box_Plot_Violin_Plot.png)

Bar Chart

var = df.groupby('Gender').Sales.sum() #grouped sum of sales at Gender level

fig = plt.figure()

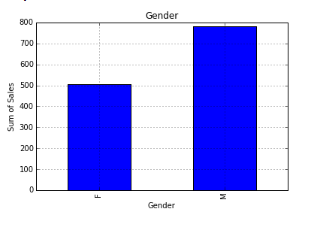
ax1 = fig.add\_subplot(1,1,1)

ax1.set\_xlabel('Gender')

ax1.set\_ylabel('Sum of Sales')

ax1.set\_title("Gender wise Sum of Sales")

var.plot(kind='bar')

[](https://www.analyticsvidhya.com/wp-content/uploads/2015/05/box-plot.png)

You can read more about pandas **groupby** [here](http://pandas.pydata.org/pandas-docs/stable/groupby.html) and for dataframe. For plot refer this [link](http://pandas.pydata.org/pandas-docs/dev/generated/pandas.DataFrame.plot.html).

Line Chart

var = df.groupby('BMI').Sales.sum()

fig = plt.figure()

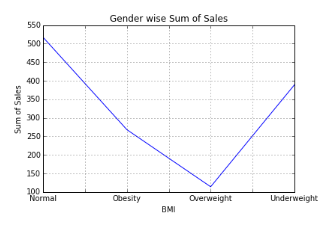
ax1 = fig.add\_subplot(1,1,1)

ax1.set\_xlabel('BMI')

ax1.set\_ylabel('Sum of Sales')

ax1.set\_title("BMI wise Sum of Sales")

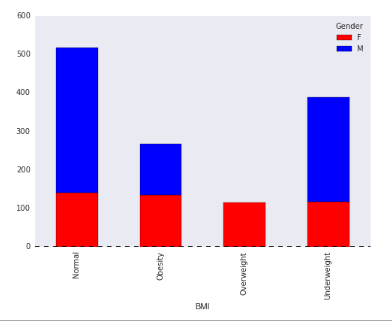
var.plot(kind='line')

[](https://www.analyticsvidhya.com/wp-content/uploads/2015/05/Line.png)

Stacked Column Chart

var = df.groupby(['BMI','Gender']).Sales.sum()

var.unstack().plot(kind='bar',stacked=True,  color=['red','blue'], grid=False)



Dataframe.unstack() returns a Data Frame having a new level of column labels whose inner-most level consists of the pivoted index labels. Read more about dataframe.unstack [here](http://pandas.pydata.org/pandas-docs/dev/generated/pandas.DataFrame.unstack.html).

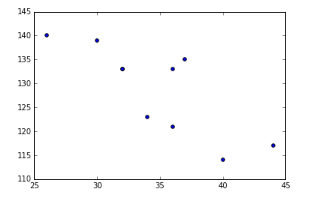
Scatter Plot

fig = plt.figure()

ax = fig.add\_subplot(1,1,1)

ax.scatter(df['Age'],df['Sales']) #You can also add more variables here to represent color and size.

plt.show()

[](https://www.analyticsvidhya.com/wp-content/uploads/2015/05/Scatter_Chart.png)

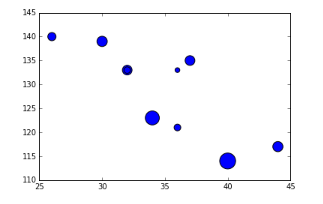
Bubble Plot

fig = plt.figure()

ax = fig.add\_subplot(1,1,1)

ax.scatter(df['Age'],df['Sales'], s=df['Income']) # Added third variable income as size of the bubble

plt.show()



Pie chart

var=df.groupby(['Gender']).sum().stack()

temp=var.unstack()

type(temp)

x\_list = temp['Sales']

label\_list = temp.index

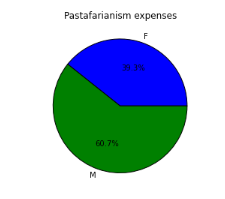
pyplot.axis("equal") #The pie chart is oval by default. To make it a circle use pyplot.axis("equal")

#To show the percentage of each pie slice, pass an output format to the autopctparameter

plt.pie(x\_list,labels=label\_list,autopct="%1.1f%%")

plt.title("Pastafarianism expenses")

plt.show()

[](https://www.analyticsvidhya.com/wp-content/uploads/2015/05/Pie.png)

Heat Map

import numpy as np

#Generate a random number, you can refer your data values also

data = np.random.rand(4,2)

rows = list('1234') #rows categories

columns = list('MF') #column categories

fig,ax=plt.subplots()

#Advance color controls

ax.pcolor(data,cmap=plt.cm.Reds,edgecolors='k')

ax.set\_xticks(np.arange(0,2)+0.5)

ax.set\_yticks(np.arange(0,4)+0.5)

# Here we position the tick labels for x and y axis

ax.xaxis.tick\_bottom()

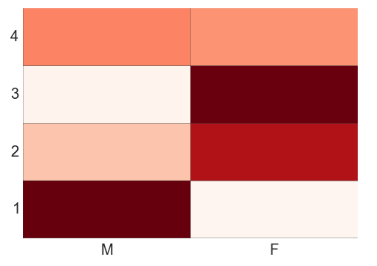
ax.yaxis.tick\_left()

#Values against each labels

ax.set\_xticklabels(columns,minor=False,fontsize=20)

ax.set\_yticklabels(rows,minor=False,fontsize=20)

plt.show()

[[](https://www.analyticsvidhya.com/wp-content/uploads/2015/05/Heat.png)](https://www.analyticsvidhya.com/wp-content/uploads/2015/05/Heat.png)

You can attempt to plot a heat based on two variables like Gender on x-axis, BMI on Y-axis and Sales values as data points. Please share your code and output in comment section.

End Notes

By now, you must have realized, how beautifully data can be presented using visualization. I find performing visualization in Python much easier as compared to R. In this article, we discussed about deriving various visualizations in Python. In this process, we made use of matplotlib and seaborn in python.  In the subsequent articles we will explore map visualization and word cloud in python.

Do you have any questions on visualization methods / techniques / selection? Feel free to [discuss](http://discuss.analyticsvidhya.com/) it with us.

If you like what you just read & want to continue your analytics learning, [subscribe to our emails](http://feedburner.google.com/fb/a/mailverify?uri=analyticsvidhya), [follow us on twitter](http://twitter.com/analyticsvidhya) or like our [facebook page](http://facebook.com/analyticsvidhya" \t "_blank).

Share this:

* [362Click to share on LinkedIn (Opens in new window)362](https://www.analyticsvidhya.com/blog/2015/05/data-visualization-python/?share=linkedin&nb=1)
* [17Click to share on Facebook (Opens in new window)17](https://www.analyticsvidhya.com/blog/2015/05/data-visualization-python/?share=facebook&nb=1)
* [Click to share on Google+ (Opens in new window)](https://www.analyticsvidhya.com/blog/2015/05/data-visualization-python/?share=google-plus-1&nb=1)
* [Click to share on Twitter (Opens in new window)](https://www.analyticsvidhya.com/blog/2015/05/data-visualization-python/?share=twitter&nb=1)
* [Click to share on Pocket (Opens in new window)](https://www.analyticsvidhya.com/blog/2015/05/data-visualization-python/?share=pocket&nb=1)
* [Click to share on Reddit (Opens in new window)](https://www.analyticsvidhya.com/blog/2015/05/data-visualization-python/?share=reddit&nb=1)