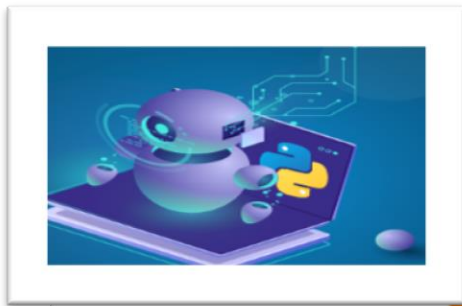


# Data science and Machine learning with Python



# Data Visualization with **Seaborn**

## Introduction to Seaborn

### What is Python Seaborn?

Seaborn is a library for making statistical graphics in Python. It is built on top of matplotlib and closely integrated with pandas data structures.

Here is some of the functionality that seaborn offers:

- A dataset-oriented API for examining relationships between multiple variables
- Specialized support for using categorical variables to show observations or aggregate statistics
- Options for visualizing univariate or bivariate distributions and for comparing them between subsets of data
- Automatic estimation and plotting of linear regression models for different kinds of dependent variables
- Convenient views onto the overall structure of complex datasets
- High-level abstractions for structuring multi-plot grids that let you easily build complex visualizations
- Concise control over matplotlib figure styling with several built-in themes
- Tools for choosing colour palettes that faithfully reveal patterns in your data

Seaborn aims to make visualization a central part of exploring and understanding data. Its dataset-oriented plotting functions operate on dataframes and arrays containing whole datasets and internally perform the necessary semantic mapping and statistical aggregation to produce informative plots

### How to install Seaborn?

- To install the Python Seaborn library, you can use the following commands based on the platform you use:

*pip install seaborn*

```

Administrator: Windows PowerShell
PS C:\> pip install seaborn
Collecting seaborn
  Downloading https://files.pythonhosted.org/packages/a8/76/220ba4420459d9c4c9c9587c6ce607bf56c25b3d3d2de62056efe482dad/seaborn-0.9.0-py3-none-any.whl (208kB)
    | 215kB 656kB/s
Requirement already satisfied: matplotlib>=1.4.3 in c:\program files\python37\lib\site-packages (from seaborn) (3.1.1)
Collecting scipy>=0.14.0
  Downloading https://files.pythonhosted.org/packages/50/eb/defa40367863304e1ef01c6572584c411446a5f29bdd9dc90f91509e9144/scipy-1.3.1-cp37-cp37m-win_amd64.whl (30.3MB)
    | 30.4MB 2.2MB/s
Requirement already satisfied: pandas>=0.15.2 in c:\program files\python37\lib\site-packages (from seaborn) (0.25.1)
Requirement already satisfied: numpy>=1.9.3 in c:\program files\python37\lib\site-packages (from seaborn) (1.17.2)
Requirement already satisfied: python-dateutil>=2.1 in c:\program files\python37\lib\site-packages (from matplotlib>=1.4.3->seaborn) (2.8.0)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\program files\python37\lib\site-packages (from matplotlib>=1.4.3->seaborn) (1.1.0)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in c:\program files\python37\lib\site-packages (from matplotlib>=1.4.3->seaborn) (2.4.2)
Requirement already satisfied: cycler>=0.10 in c:\program files\python37\lib\site-packages (from matplotlib>=1.4.3->seaborn) (0.10.0)
Requirement already satisfied: pytz>=2017.2 in c:\program files\python37\lib\site-packages (from pandas>=0.15.2->seaborn) (2019.3)
Requirement already satisfied: six>=1.5 in c:\program files\python37\lib\site-packages (from python-dateutil>=2.1->matplotlib>=1.4.3->seaborn) (1.12.0)
Requirement already satisfied: setuptools in c:\program files\python37\lib\site-packages (from kiwisolver>=1.0.1->matplotlib>=1.4.3->seaborn) (40.8.0)
Installing collected packages: scipy, seaborn
Successfully installed scipy-1.3.1 seaborn-0.9.0
PS C:\>

```

## Difference Between Matplotlib and Seaborn

	MatPlotLib	Seaborn
Functionality	Matplotlib is mainly deployed for basic plotting. Visualization using Matplotlib generally consists of bars, pies, lines, scatter plots and so on.	Seaborn, on the other hand, provides a variety of visualization patterns. It uses fewer syntax and has easily interesting default themes. It specializes in statistics visualization and is used if one has to summarize data in visualizations and also show the distribution in the data.
Handling Multiple Figures	Matplotlib has multiple figures can be opened, but need to be closed explicitly. plt.close() only closes the current figure. plt.close('all') would close em all.	Seaborn automates the creation of multiple figures. This sometimes leads to OOM (out of memory) issues.
Visualization	Matplotlib is a graphics package for data visualization in Python. It is well integrated with NumPy and Pandas. The pyplot module mirrors the MATLAB plotting commands closely. Hence, MATLAB users can easily transit to plotting with Python.	Seaborn is more integrated for working with Pandas data frames. It extends the Matplotlib library for creating beautiful graphics with Python using a more straightforward set of methods.
Data Frames and Arrays	Matplotlib works with data frames and arrays. It has different stateful APIs for plotting. The figures and axes are represented by the object and therefore plot() like calls without parameters suffices without having to manage parameters.	Seaborn works with the dataset as a whole and is much more intuitive than Matplotlib. For Seaborn, replot() is the entry API with 'kind' parameter to specify the type of plot which could be line, bar, or any of the other types. Seaborn is not stateful. Hence, plot() would require passing the object.

Designed by Abdur Rahman Joy - MCSD, MCPD, MCSE, MCTS, OCJP, Sr. Technical Trainer for VFX at IDB BISW (Scholarship program), and C#.net, R, Scala, Kotlin, JAVA, Android/iOS/Windows Mobile Apps, SQL server, Azure, Oracle, SharePoint Development, AWS , CEH, KALI Linux, Python, Data Science, Machine Learning ,Software Testing, Graphics, Multimedia and Game Developer at Joy Infosys and other premises like BITM, SkillsJob, PNTL, Leads Training and New Horizon inc , Cell #: +880-1712587348, email: [jspaonline@gmail.com](mailto:jspaonline@gmail.com). Web URL: <http://www.joyinfosys.com/me>.

Flexibility	Matplotlib is highly customizable and powerful.	Seaborn avoids a ton of boilerplate by providing default themes which are commonly used.
Use Cases	Pandas uses Matplotlib. It is a neat wrapper around Matplotlib.	Seaborn is for more specific use cases. Also, it is Matplotlib under the hood. It is specially meant for statistical plotting.

### Compare the following plots:

# Import the necessary libraries

```
import matplotlib.pyplot as plt
```

```
import pandas as pd
```

# Initialize Figure and Axes object

```
fig, ax = plt.subplots()
```

# Load in data

```
tips = pd.read_csv("https://raw.githubusercontent.com/mwaskom/seaborn-data/master/tips.csv")
```

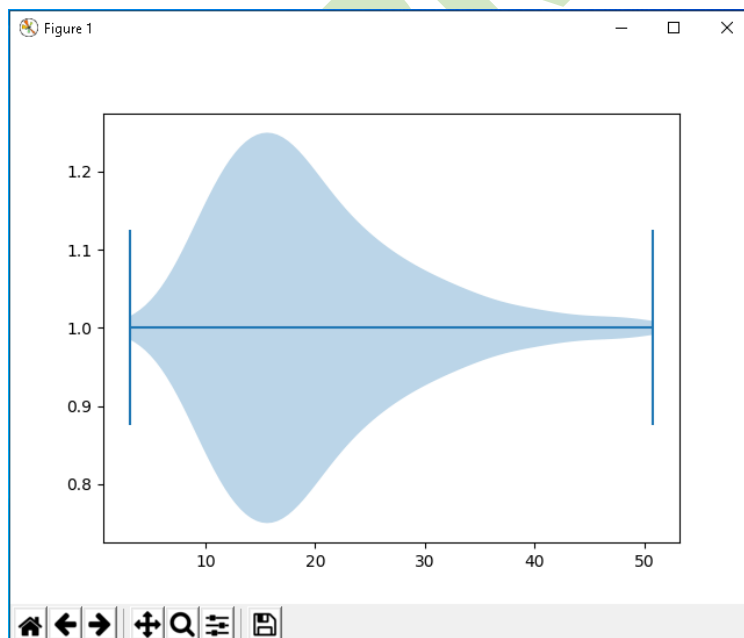
# Create violinplot

```
ax.violinplot(tips["total_bill"], vert=False)
```

# Show the plot

```
plt.show()
```

Result:



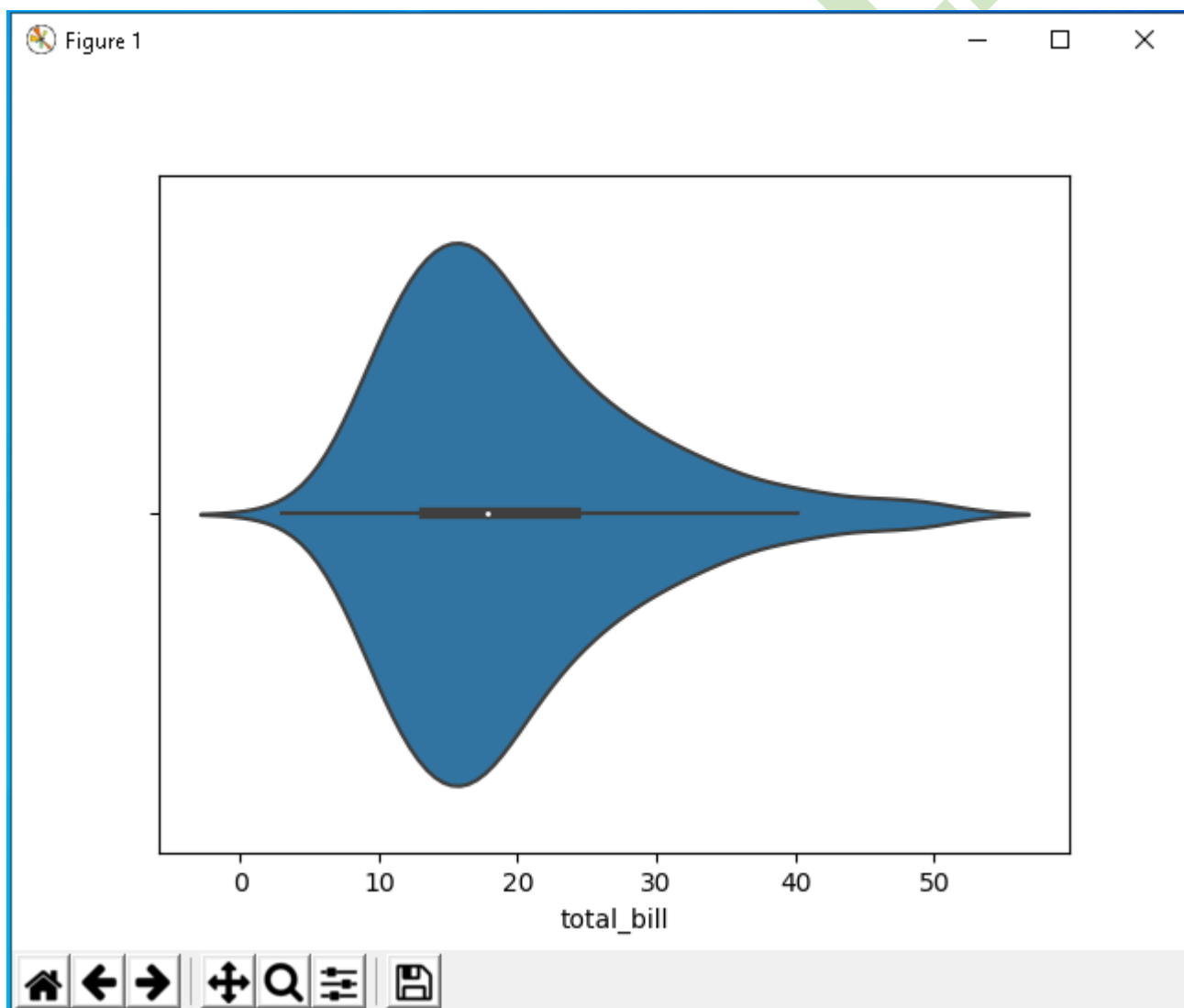
```
# Import the necessary libraries
import matplotlib.pyplot as plt
import seaborn as sns

# Load the data
tips = sns.load_dataset("tips")

# Create violinplot
sns.violinplot(x = "total_bill", data=tips)

# Show the plot
plt.show()
```

Res:



The Matplotlib defaults that usually don't speak to users are the colors, the tick marks on the upper and right axes, the style,...

The examples above also makes another frustration of users more apparent: the fact that working with DataFrames doesn't go quite as smoothly with Matplotlib, which can be annoying if you're doing exploratory analysis with Pandas. And that's exactly what Seaborn addresses: the plotting functions operate on DataFrames and arrays that contain a whole dataset.

As Seaborn compliments and extends Matplotlib, the learning curve is quite gradual: if you know Matplotlib, you'll already have most of Seaborn down.

## How To Load Data To Construct Seaborn Plots

When you're working with Seaborn, you can either use one of the built-in data sets that the library itself has to offer or you can load a Pandas DataFrame. In this section, you'll see how to do both.

### Loading A Built-in Seaborn Data Set

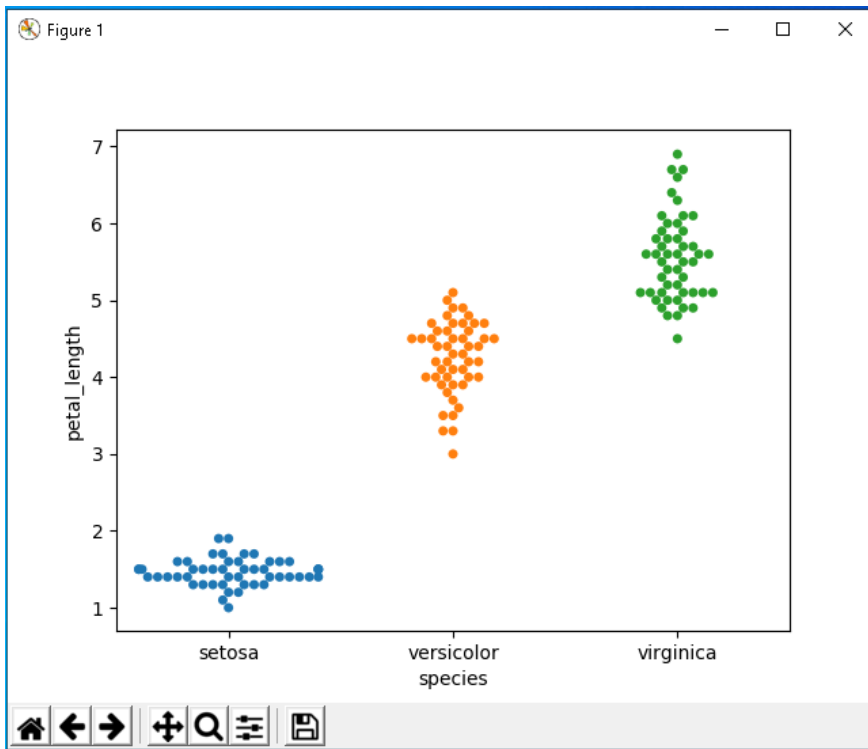
To start working with a built-in Seaborn data set, you can make use of the `load_dataset()` function. To get an overview or inspect all data sets that this function opens up to you, Check out the following example to see how the `load_dataset()` function works:

```
# Import necessary libraries
import seaborn as sns
import matplotlib.pyplot as plt

# Load iris data
iris = sns.load_dataset("iris")

# Construct iris plot
sns.swarmplot(x="species", y="petal_length", data=iris)

# Show plot
plt.show()
```



As an anecdote, it might be interesting for you to know that the import convention `sns` comes from the fictional character Samuel Norman “Sam” Seaborn on the television serial drama The West Wing. It’s an inside joke by the core developer of Seaborn, namely, Michael Waskom.

### Loading Your Pandas DataFrame Getting Your Data

Of course, most of the fun in visualizing data lies in the fact that you would be working with your own data and not the built-in data sets of the Seaborn library. Seaborn works best with Pandas DataFrames and arrays that contain a whole data set.

**Remember** that DataFrames are a way to store data in rectangular grids that can easily be overviewed. Each row of these grids corresponds to measurements or values of an instance, while each column is a vector containing data for a specific variable. This means that a DataFrame’s rows do not need to contain, but can contain, the same type of values: they can be numeric, character, logical, etc. Specifically for Python, DataFrames come with the Pandas library, and they are defined as a two-dimensional labeled data structures with columns of potentially different types.

The reason why Seaborn is so great with DataFrames is, for example, because labels from DataFrames are automatically propagated to plots or other data structures, as you saw in the first example of this tutorial, where you plotted a violinplot with Seaborn. There, you saw that the x-axis had a legend `total_bill`, while this was not the case with the Matplotlib plot. This already takes a lot of work away from you.



## How To Show Seaborn Plots

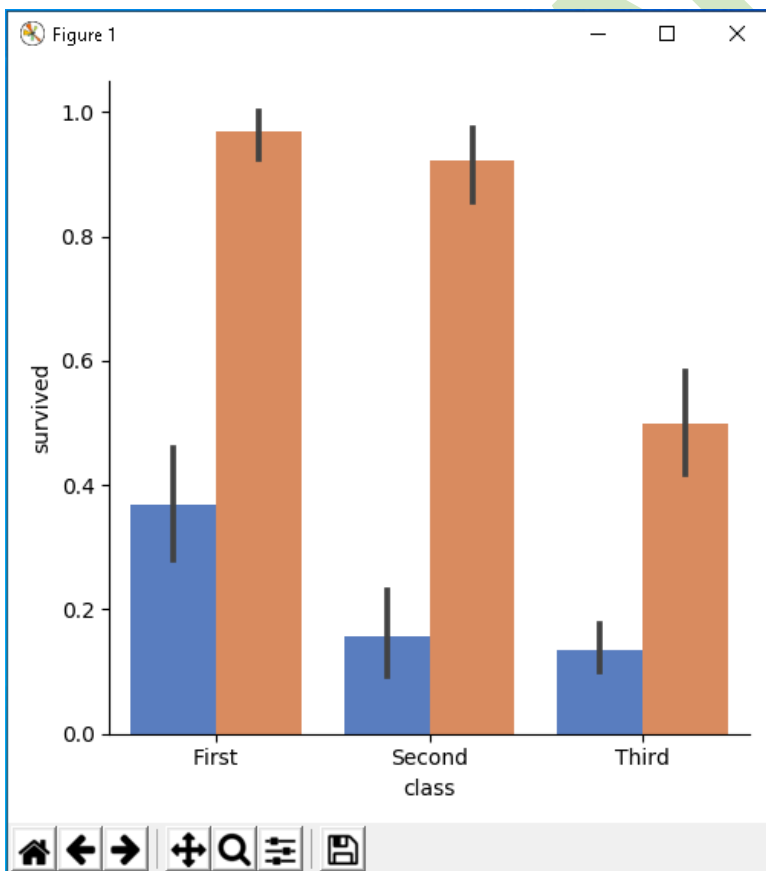
Matplotlib still underlies Seaborn, which means that the anatomy of the plot is still the same and that you'll need to use `plt.show()` to make the image appear to you. You might have already seen this from the previous example in this tutorial. In any case, here's another example where the `show()` function is used to show the plot:

```
# Import necessarily libraries
import matplotlib.pyplot as plt
import seaborn as sns

# Load data
titanic = sns.load_dataset("titanic")

# Set up a factorplot
g = sns.factorplot("class", "survived", "sex", data=titanic, kind="bar", palette="muted", legend=False)

# Show plot
plt.show()
```



Designed by Abdur Rahman Joy - MCSD, MCPD, MCSE, MCTS, OCPJ, Sr. Technical Trainer for VFX at IDB BISW (Scholarship program), and C#.net, R, Scala, Kotlin, JAVA, Android/IOS/Windows Mobile Apps, SQL server, Azure, Oracle, SharePoint Development, AWS , CEH, KALI Linux, Python, Data Science, Machine Learning ,Software Testing, Graphics, Multimedia and Game Developer at Joy Infosys and other premises like BITM, SkillsJob, PNTL, Leads Training and New Horizon inc , Cell #: +880-1712587348, email: [jspaonline@gmail.com](mailto:jspaonline@gmail.com). Web URL: <http://www.joyinfosys.com/me>.



**Note** that in the code chunk above you work with a built-in Seaborn data set and you create a factorplot with it. A factorplot is a categorical plot, which in this case is a bar plot. That's because you have set the `kind` argument to `"bar"`. Also, you set which colors should be displayed with the `palette` argument and that you set the `legend` to `False`.

## How To Use Seaborn's Colors As A colormap in Matplotlib?

Besides using Seaborn with Matplotlib defaults, there's also questions on how to bring in Seaborn colors into Matplotlib plots. You can make use of `color_palette()` to define a color map that you want to be using and the number of colors with the argument `n_colors`. In this case, the example will assume that there are 5 labels assigned to the data points that are defined in `data1` and `data2`, so that's why you pass `5` to this argument and you also make a list with length equal to `N` where 5 integers vary in the variable `colors`.

```
# Import the necessary libraries
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np
from matplotlib.colors import ListedColormap

# Define a variable N
N = 500

# Construct the colormap
current_palette = sns.color_palette("muted", n_colors=5)
cmap = ListedColormap(sns.color_palette(current_palette).as_hex())

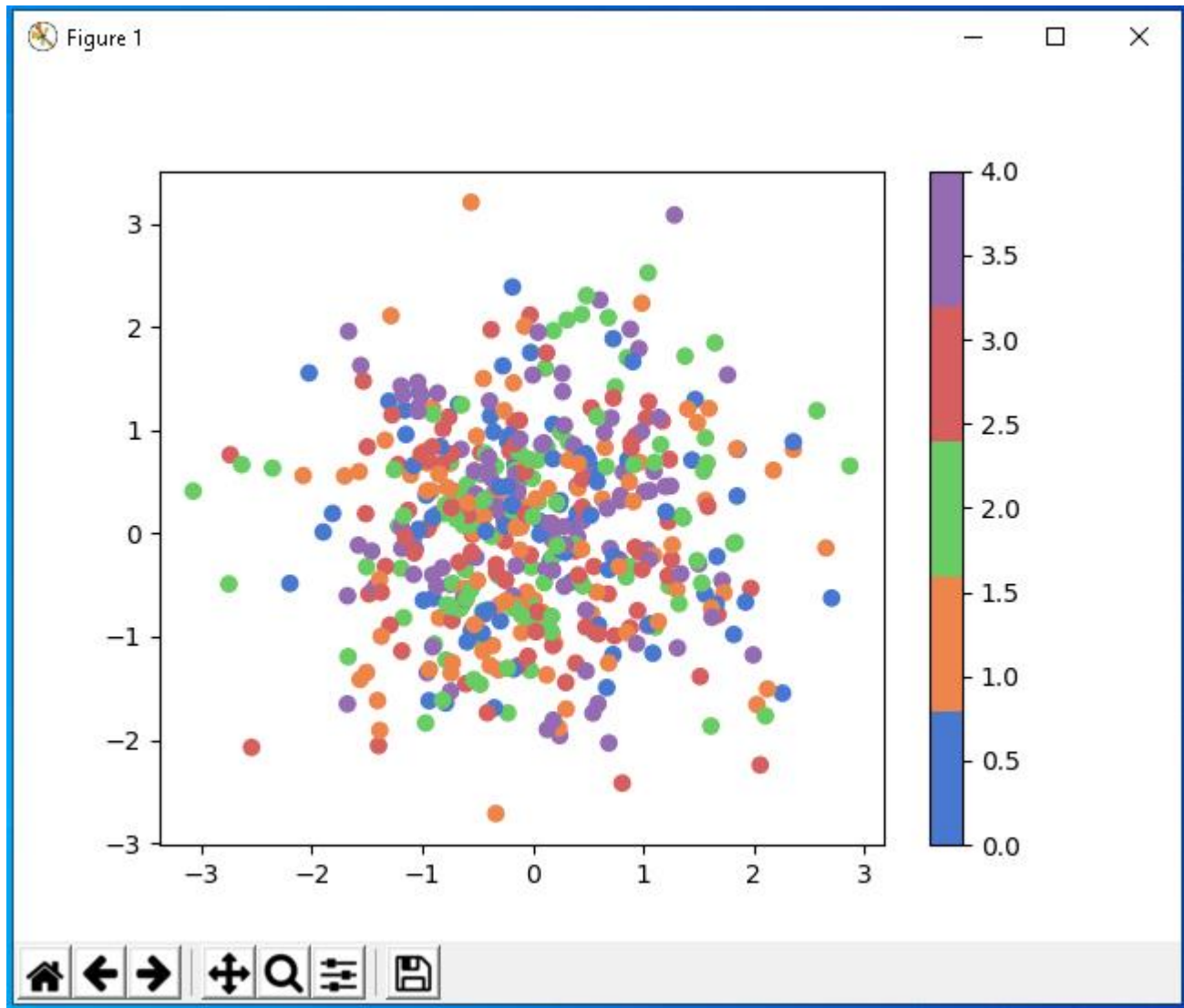
# Initialize the data
data1 = np.random.randn(N)
data2 = np.random.randn(N)
# Assume that there are 5 possible labels
colors = np.random.randint(0,5,N)

# Create a scatter plot
plt.scatter(data1, data2, c=colors, cmap=cmap)

# Add a color bar
plt.colorbar()

# Show the plot
```

plt.show()



## How To Temporarily Set The Plot Style

You can use `axes_style()` in a `with` statement to temporarily set the plot style. This, in addition to the use of `plt.subplot()`, will allow you to make figures that have differently-styled axes, like in the example below:

```
# Import necessary libraries
import matplotlib.pyplot as plt
import seaborn as sns

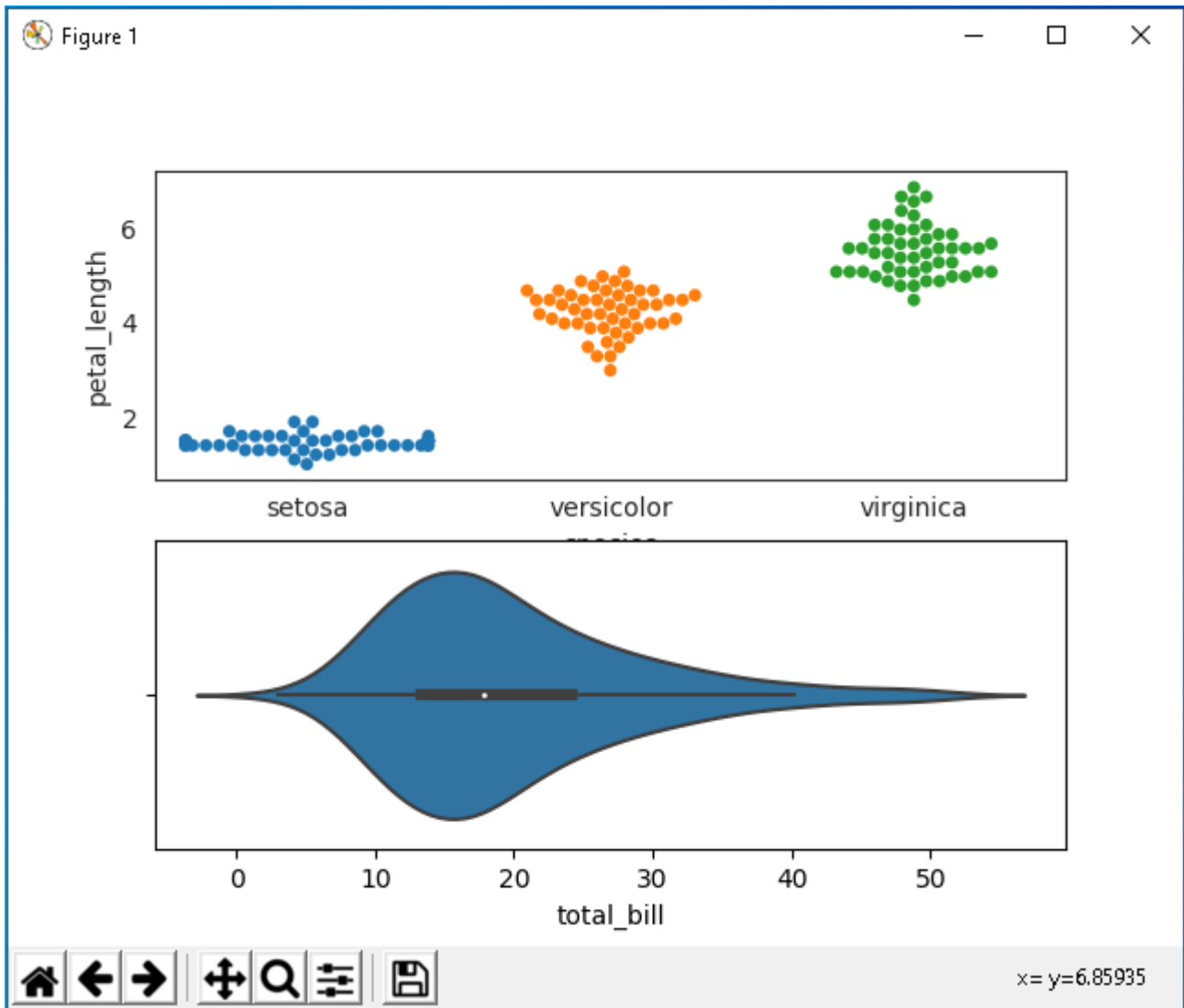
# Load data
iris = sns.load_dataset("iris")
tips = sns.load_dataset("tips")

# Set axes style to white for first subplot
with sns.axes_style("white"):
    plt.subplot(211)
    sns.swarmplot(x="species", y="petal_length", data=iris)

# Initialize second subplot
plt.subplot(212)

# Plot violinplot
sns.violinplot(x="total_bill", data=tips)

# Show the plot
plt.show()
```



## How To Add A Title

To add titles to your Seaborn plots, you basically follow the same procedure as you have done in the previous sections. For Axes-level functions, you'll adjust the title on the `Axes` level itself with the help of `set_title()`. Just pass in the title that you want to see appear:

# Import the libraries

```
import matplotlib.pyplot as plt
import seaborn as sns
```

```
tips = sns.load_dataset("tips")
```

# Create the boxplot

Designed by Abdur Rahman Joy - MCSD, MCPD, MCSE, MCTS, OCPJ, Sr. Technical Trainer for VFX at IDB BISW (Scholarship program), and C#.net, R, Scala, Kotlin, JAVA, Android/IOS/Windows Mobile Apps, SQL server, Azure, Oracle, SharePoint Development, AWS , CEH, KALI Linux, Python, Data Science, Machine Learning ,Software Testing, Graphics, Multimedia and Game Developer at Joy Infosys and other premises like BITM, SkillsJob, PNTL, Leads Training and New Horizon inc , Cell #: +880-1712587348, email: [jspaonline@gmail.com](mailto:jspaonline@gmail.com). Web URL: <http://www.joyinfosys.com/me>.

```
ax = sns.boxplot(x="total_bill", data=tips)
```

```
# Set title
```

```
ax.set_title("boxplot")
```

```
# Show the plot
```

```
plt.show()
```

