Hyperiondev

TASK

Data Visualisation - Simple

Visit our website

Introduction

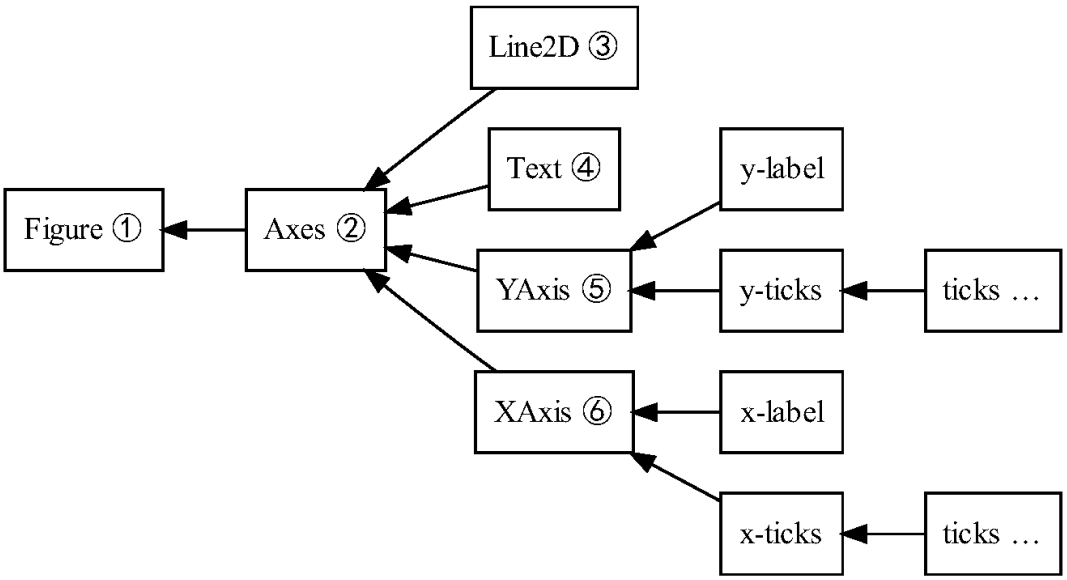
WELCOME TO THE DATA VISUALISATION - SIMPLE TASK!

In this task, you will practise using two data visualisation libraries, matplotlib and seaborn.

MATPLOTLIB

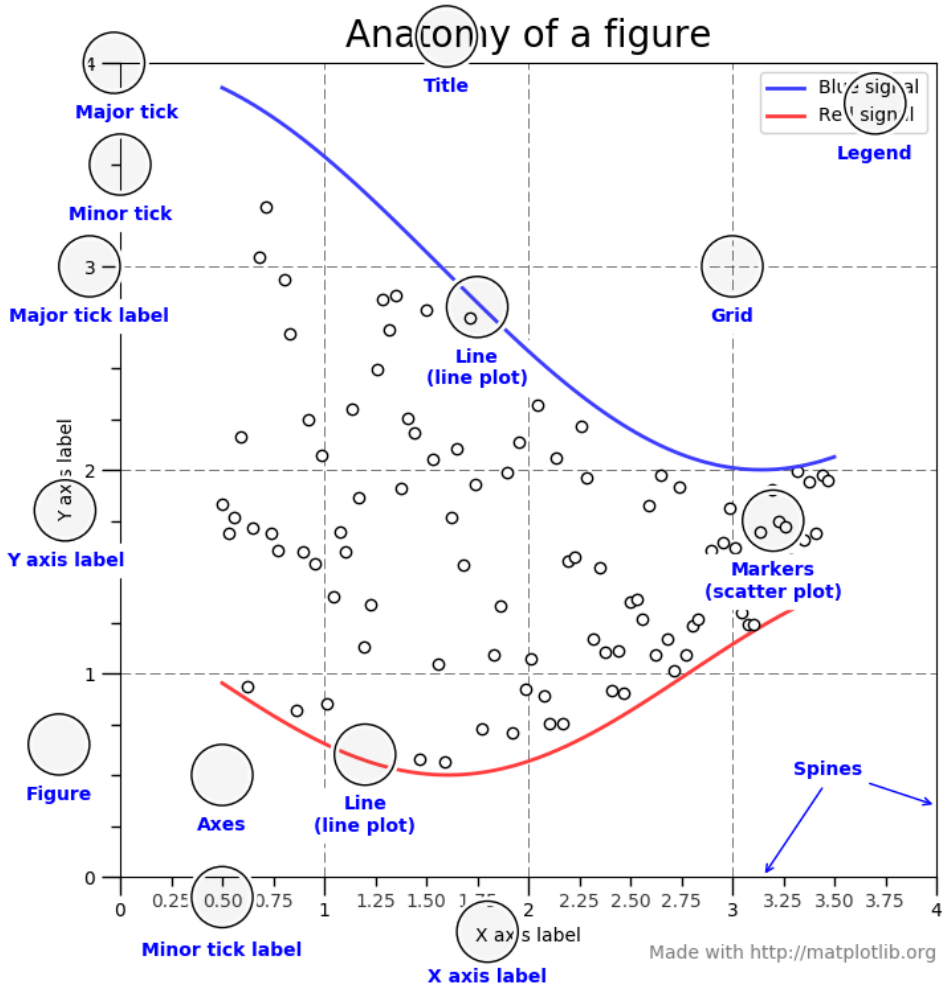
Many packages in Python allow one to perform powerful data analysis. **Matplotlib** is "... a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and IPython shells, Jupyter Notebook, web application servers, and four graphical user interface toolkits" (Hunter, 2007). With matplotlib, we are able to draw many different graphs in Python. Explore the **matplotlib documentation** to learn more.

Matplotlib is organised into a hierarchy. At the top of the hierarchy, there is a library called pyplot. We create a pyplot to create a figure. The figure keeps track of all the child axes, artists (titles, figure legends, etc.), and the canvas.



(Hunter, 2007)

Artists are defined as follows: "Basically everything you can see on the figure is an artist (even the Figure, Axes, and Axis objects). This includes Text objects, Line2D objects, collection objects, Patch objects ... (you get the idea). When the figure is rendered, all of the artists are drawn to the canvas. Most Artists are tied to an Axes; such an Artist cannot be shared by multiple Axes or moved from one to another" (Hunter, 2007).



(Hunter, 2007)

All of the plotting functions expect `np.array` or `np.ma.masked_array` as input so it is best to convert any pandas (or similar array-like data structures) to arrays before using them with matplotlib.

Install matplotlib

First let's check if you have matplotlib installed. Open up your terminal, input the following, and then hit enter:

```
pip3 show matplotlib
```

If it is not installed you will need pip (a package manager) to install it. Use the following command to check that you have the latest version of pip:

```
pip3 install --upgrade pip
```

Then use the following command to install matplotlib:

```
python -m pip3 install -U matplotlib
```

For further guidelines for installing matplotlib, see the official documentation **here**.

Creating a figure

To use packages or libraries in Python, you will need to import them at the top of your Python file:

```
import matplotlib.pyplot as plt
import numpy as np
```

We can create a quick dataset using Numpy and visualise the data using matplotlib:

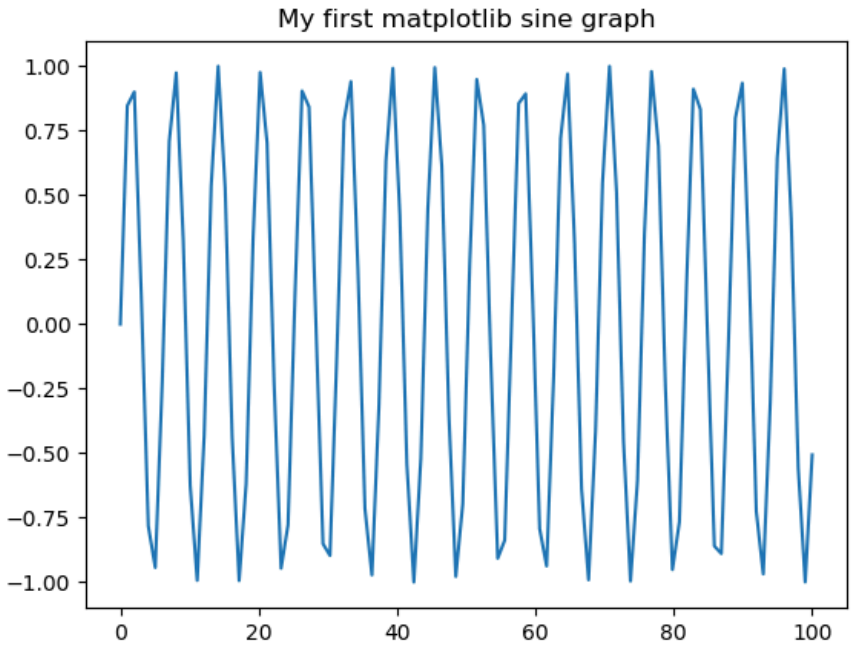
```
# Prepare the data
x = np.linspace(0, 100, 100) # x axis
y = np.sin(x) # y values

# Plot the data
plt.plot(x, y, label="sine")

# Create a title
plt.title("My first matplotlib sine graph")

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

If you run the program, you will get something like this:



Try to play around with the sine graph code provided in an example file to get a better understanding.

Extra resource

For more information about working with matplotlib, please consult the fourth chapter ("**Visualization with Matplotlib**") in the book entitled, "**Python Data Science Handbook**" by Jake VanderPlas. You can also explore the **matplotlib webpage**.

SEABORN

Seaborn is a data visualisation library that has been built on top of matplotlib. While matplotlib provides basic graphs, such as line and bar charts, seaborn can provide a bit more in terms of graphing. In addition, it integrates quite well with pandas.

Some commonly-used seaborn plots include:

- `histplot()`
- `barplot()`
- `boxplot()`

And there are many others! You will get accustomed to many of these methods during the course of this bootcamp.

Let's say that we are reading insurance data that contains a column for age and a column for the insurance charge. We would like to understand the relationship existing between these two columns:

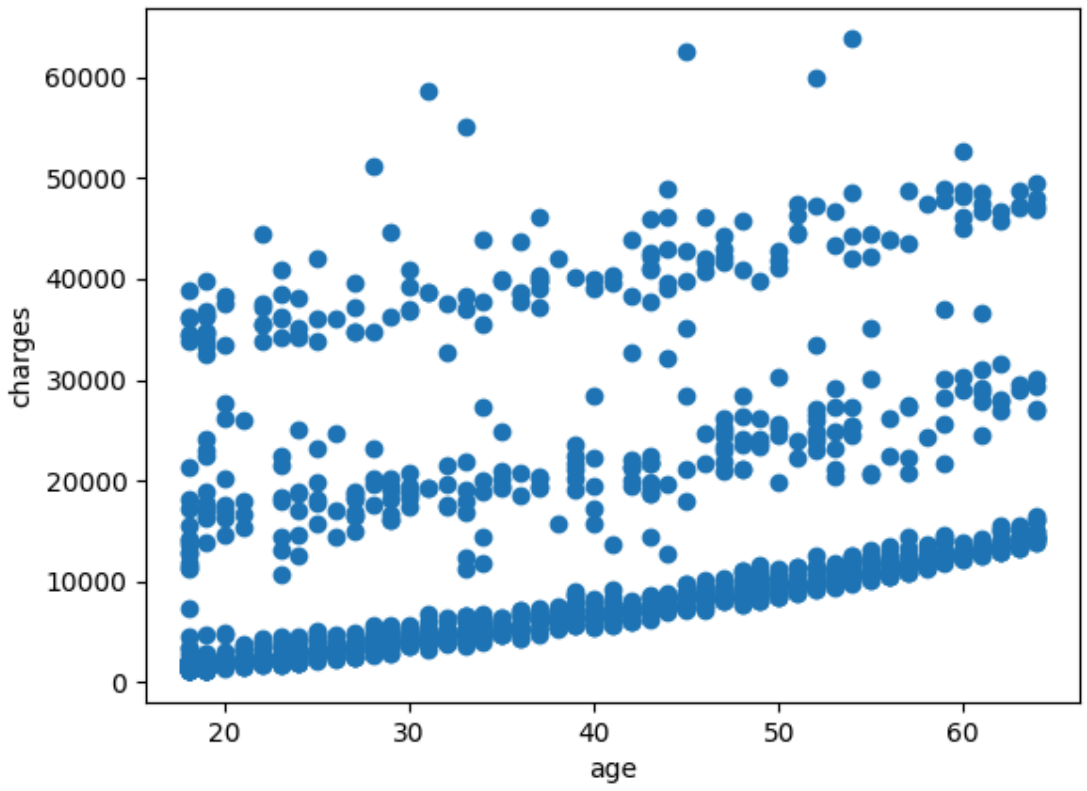
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Load data
ins_df = pd.read_csv('insurance.csv')
```

In matplotlib, the **`scatter()`** method would be most appropriate. This can be achieved as follows:

```
# Plot scatterplot
plt.figure()
plt.scatter(ins_df['age'], ins_df['charges'])
plt.xlabel("age")
plt.ylabel('charges')
plt.show()
plt.close()
```

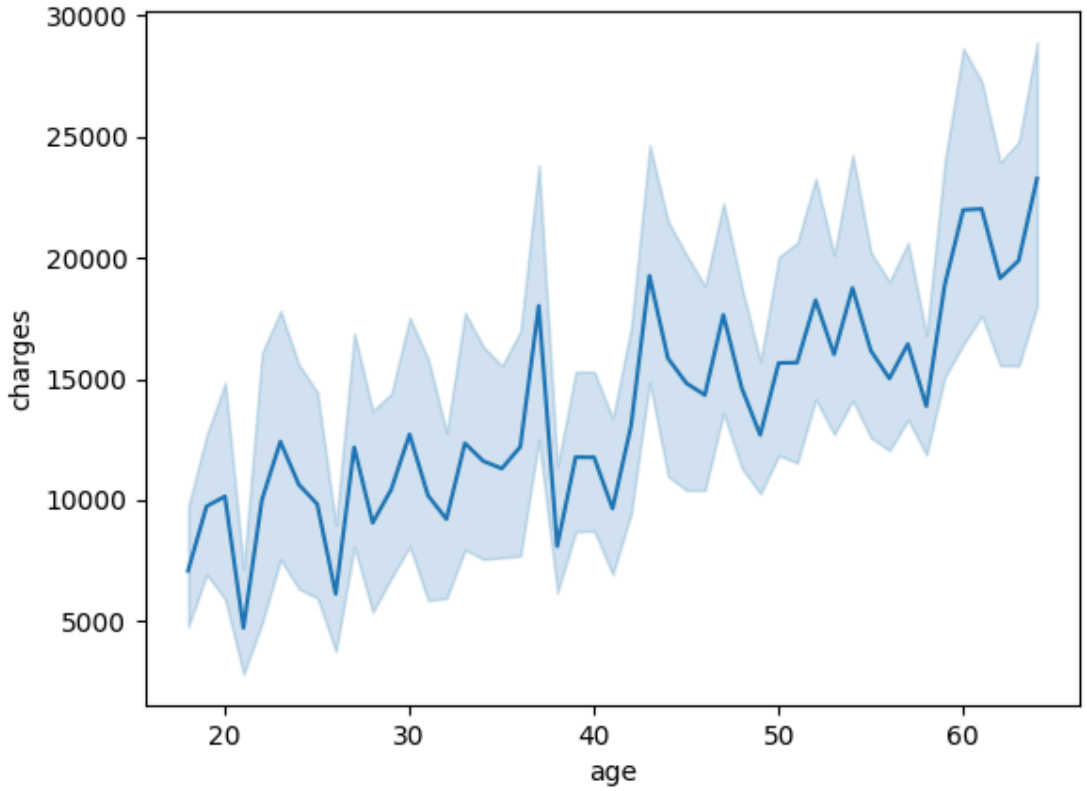
And would look something like this:



You can get a general sense of this data, but it may take time to understand the overall trend. In Seaborn, there is a **`lineplot()`** method that automatically plots averages and standard deviations for ease of reading. To do this, see the following:

```
# Plot lineplot
plt.figure()
sns.lineplot(x='age', y='charges', data=ins_df)
plt.show()
plt.savefig('sns_lineplot.png') # save as a png image file
plt.close()
```

And will end up with something that looks like this:



This makes it a lot easier to see the overall trend existing in the data: the higher your age, the larger your insurance charges.

Instructions

First, read and run the **example files** provided. Feel free to write and run your own example code before doing the practical task to become more comfortable with the concepts covered in this task.

Practical Task

Follow these steps:

- Open the Jupyter notebook named **data_viz_task.ipynb**.
- Generate the following graphs from the `Cars93.csv` dataset. Then, answer the accompanying questions in the markdown cells in the notebook:
    - A box plot for the revs per mile for the Audi, Hyundai, Suzuki, and Toyota car manufacturers. Which of these manufacturers has the car with the highest revs per mile?
    - A histogram of MPG in the city. On the same axis, show a histogram of MPG on the highway. Is it generally more fuel efficient to drive in the city or on the highway?
    - A lineplot showing the relationship between the 'Wheelbase' and 'turning circle'. What is this relationship? What happens when the wheelbase gets larger?
    - A bar plot showing the mean horsepower for each car Type (Small, Midsize, etc.). Does a larger car mean more horsepower?

Rate us

Share your thoughts

HyperionDev strives to provide internationally-excellent course content that helps you achieve your learning outcomes.

Think that the content of this task, or this course as a whole, can be improved? Do you think we've done a good job?

Click here to share your thoughts anonymously.

REFERENCES

Hunter, J.D., "Matplotlib: A 2D Graphics Environment", Computing in Science & Engineering, vol. 9, no. 3, pp. 90-95, 2007.