

# DIPLOMADO DE PROGRAMACIÓN CON PYTHON V3

TEMA 3 MATPLOTLIB

**UNIBE**

EDUCACIÓN  
CONTINUA



Eliezer Figueroa  
MCT, MOS, MCSA, MCSE, ITIL, SCF  
3 Septiembre, 2019

## COMPONENTES DEL TEMA

---

- ✓ INSTALACIÓN DE MATPLOTLIB
- ✓ PLOT()
- ✓ MÚLTIPLES CONJUNTOS DE DATOS
- ✓ GRÁFICOS DE BARRA
- ✓ GRÁFICOS DE PASTEL
- ✓ HISTOGRAMA
- ✓ SCATTER PLOTS
- ✓ 3D PLOTS



- Para hacer las inferencias estadísticas necesarias, se hace necesario visualizar los datos y Matplotlib es una de esas soluciones para los usuarios de Python.
- Es una biblioteca de trazado muy potente útil para aquellos que trabajan con Python y NumPy.
- El módulo más utilizado de Matplotlib es Pyplot que proporciona una interfaz como MATLAB, pero en su lugar, utiliza Python y es de código abierto.

```
python -m pip install -U pip  
python -m pip install -U matplotlib
```

- Se sugiere que descargue el paquete anaconda.
- Instala Python, Jupyter y otras bibliotecas importantes de Python incluyendo Matplotlib, Numpy, Pandas, scikit-learn.
- Para comenzar rápidamente con Matplotlib sin instalar nada en su equipo local, puede usar Google Colab.

```
import matplotlib.pyplot as plt  
  
plt.plot([1,2,3,4],[1,4,9,16])  
plt.show()
```

- Aquí tenga en cuenta que la primera matriz aparece en el eje X y la segunda matriz aparece en el eje Y de la gráfica.

```
import matplotlib.pyplot as plt

plt.plot([1,2,3,4],[1,4,9,16])

plt.title("FIRST PLOT")
plt.xlabel("X LABEL")
plt.ylabel("Y LABEL")

plt.show()
```

- Ahora que nuestra primera gráfica está lista, agreguemos el título, y nombre a los ejes usando los métodos `title()`, `xlabel()` y `ylabel()` respectivamente.

```
import matplotlib.pyplot as plt

plt.figure(figsize=(15,5))
plt.plot([1,2,3,4],[1,4,9,16])

plt.title("FIRST PLOT")
plt.xlabel("X LABEL")
plt.ylabel("Y LABEL")

plt.show()
```

- También podemos especificar el tamaño de la figura usando el método figure() y pasando los valores como una tupla de la longitud de filas y columnas al argumento figize.

```
import matplotlib.pyplot as plt

plt.plot([1,2,3,4],[1,4,9,16],"go")

plt.title("FIRST PLOT")
plt.xlabel("X LABEL")
plt.ylabel("Y LABEL")

plt.show()
```

- Con cada argumento de X,Y también puede pasar un tercer argumento opcional en forma de una cadena que indica el color y el tipo de línea del trazado. El formato predeterminado es b- lo que significa una línea azul sólida. En la figura de abajo usamos go que significa círculos verdes.



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

x=np.arange(1,5)
y=x**3

plt.plot([1,2,3,4],[1,4,9,16], 'go', x, y, 'r^')

plt.title("FIRST PLOT")
plt.xlabel("X LABEL")
plt.ylabel("Y LABEL")

plt.show()
```

- También podemos trazar varios conjuntos de datos pasando varios conjuntos de argumentos de los ejes X e Y en el método plot() como se muestra.

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

x=np.arange(1,5)
y=x**3

plt.subplot(1,2,1)
plt.plot([1,2,3,4],[1,4,9,16], 'go')
plt.title("1st subplot")

plt.subplot(1,2,2)
plt.plot(x,y,"r^")
plt.title("2nd subplot")

plt.suptitle("My sub-plots")
plt.show()
```

- El método subplot() toma tres argumentos: son nrow, ncol e index.
- Indican el número de filas, el número de columnas y el número de índice de la subparcela.

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

x=np.arange(1,5)
y=x**3

plt.subplot(1,2,1)
plt.plot([1,2,3,4],[1,4,9,16], 'go')
plt.title("1st subplot")

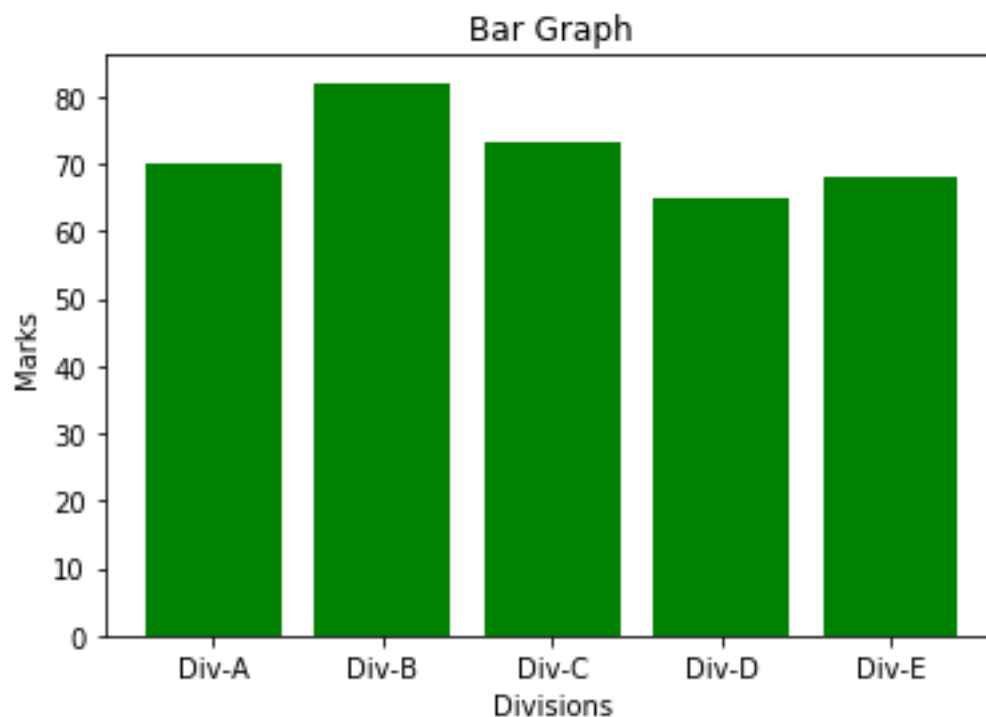
plt.subplot(1,2,2)
plt.plot(x,y,"r^")
plt.title("2nd subplot")

plt.suptitle("My sub-plots")
plt.show()
```

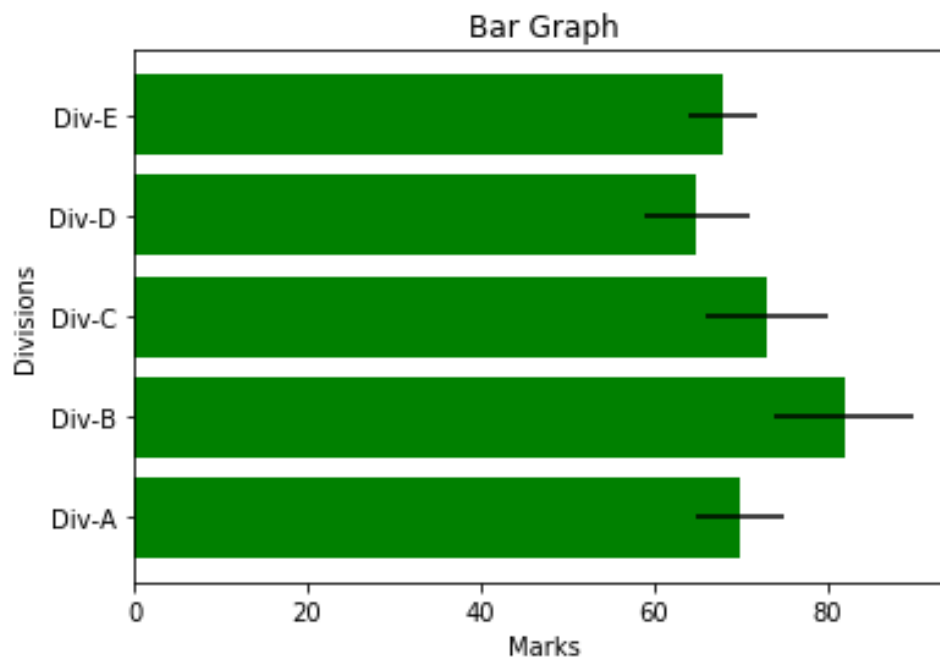
- El método subplot() toma tres argumentos: son nrow, ncol e index.
- Indican el número de filas, el número de columnas y el número de índice de la subparcela.



```
divisions = ["Div-A", "Div-B", "Div-C", "Div-D", "Div-E"]  
division_average_marks = [70, 82, 73, 65, 68]  
  
plt.bar(divisions, division_average_marks, color='green')  
plt.title("Bar Graph")  
plt.xlabel("Divisions")  
plt.ylabel("Marks")  
plt.show()
```



```
divisions = ["Div-A", "Div-B", "Div-C", "Div-D", "Div-E"]  
division_average_marks = [70, 82, 73, 65, 68]  
variance = [5, 8, 7, 6, 4]  
  
plt.barh(divisions, division_average_marks, xerr=variance, color='green')  
plt.title("Bar Graph")  
plt.xlabel("Marks")  
plt.ylabel("Divisions")  
plt.show()
```



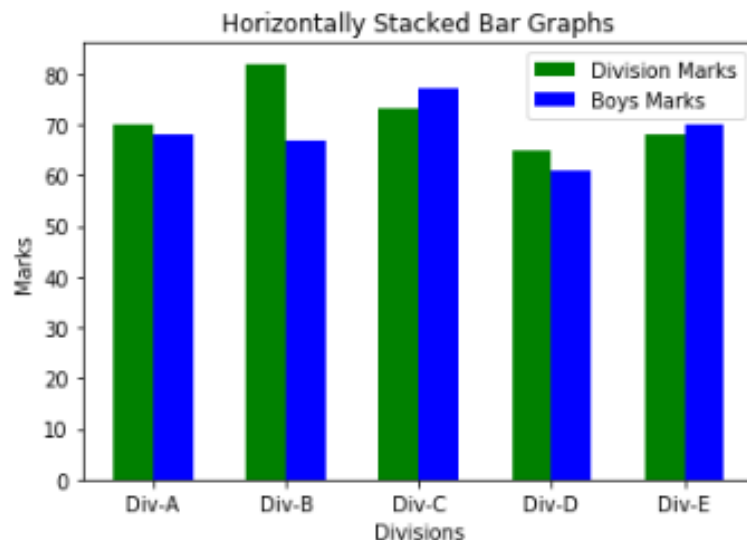
```
divisions = ["Div-A", "Div-B", "Div-C", "Div-D", "Div-E"]
division_average_marks = [70, 82, 73, 65, 68]
boys_average_marks = [68, 67, 77, 61, 70]

index = np.arange(5)
width = 0.30

plt.bar(index, division_average_marks, width, color='green',label='Division Marks')
plt.bar(index+width, boys_average_marks, width, color='blue',label='Boys Marks')
plt.title("Horizontally Stacked Bar Graphs")

plt.ylabel("Marks")
plt.xlabel("Divisions")
plt.xticks(index+ width/2, divisions)

plt.legend(loc='best')
plt.show()
```



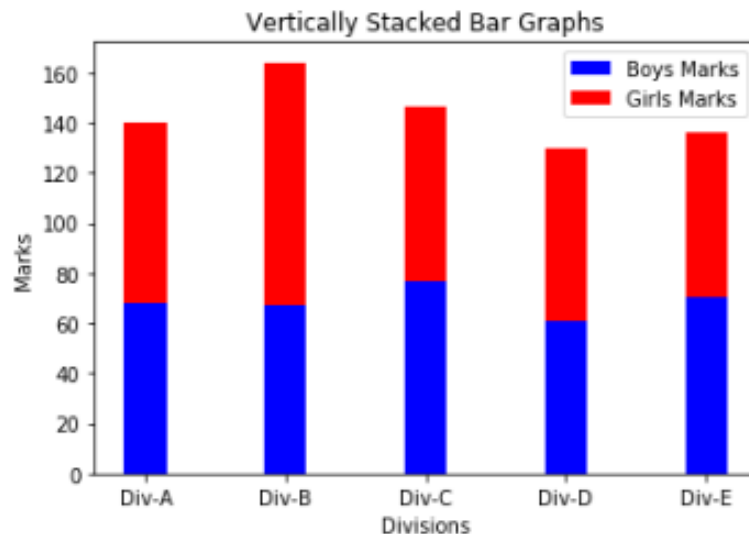
```
divisions = ["Div-A", "Div-B", "Div-C", "Div-D", "Div-E"]
boys_average_marks = [68, 67, 77, 61, 70]
girls_average_marks = [72, 97, 69, 69, 66]

index = np.arange(5)
width = 0.30

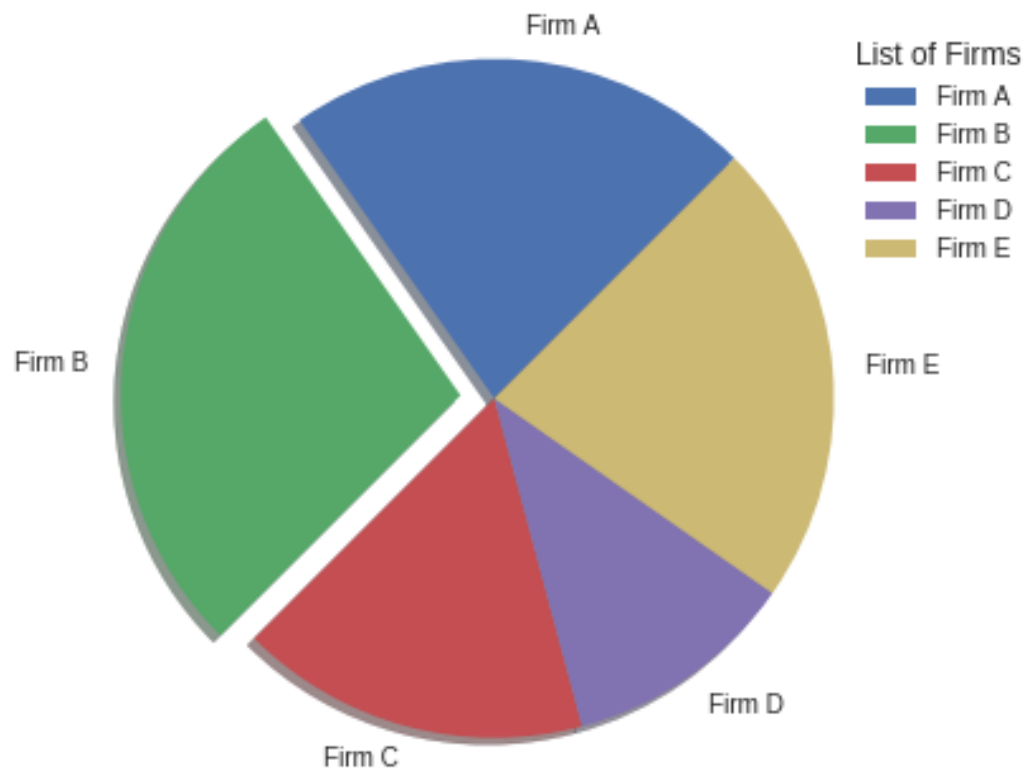
plt.bar(index, boys_average_marks, width, color="blue", label="Boys Marks")
plt.bar(index, girls_average_marks, width, color="red", label="Girls Marks", bottom=boys_average_marks)

plt.title("Vertically Stacked Bar Graphs")
plt.xlabel("Divisions")
plt.ylabel("Marks")
plt.xticks(index, divisions)

plt.legend(loc='best')
plt.show()
```



```
firms = ["Firm A", "Firm B", "Firm C", "Firm D", "Firm E"]  
market_share = [20, 25, 15, 10, 20]  
Explode = [0, 0.1, 0, 0, 0]  
plt.pie(market_share, explode=Explode, labels=firms, shadow=True, startangle=45)  
plt.axis('equal')  
plt.legend(title="List of Firms")  
plt.show()
```

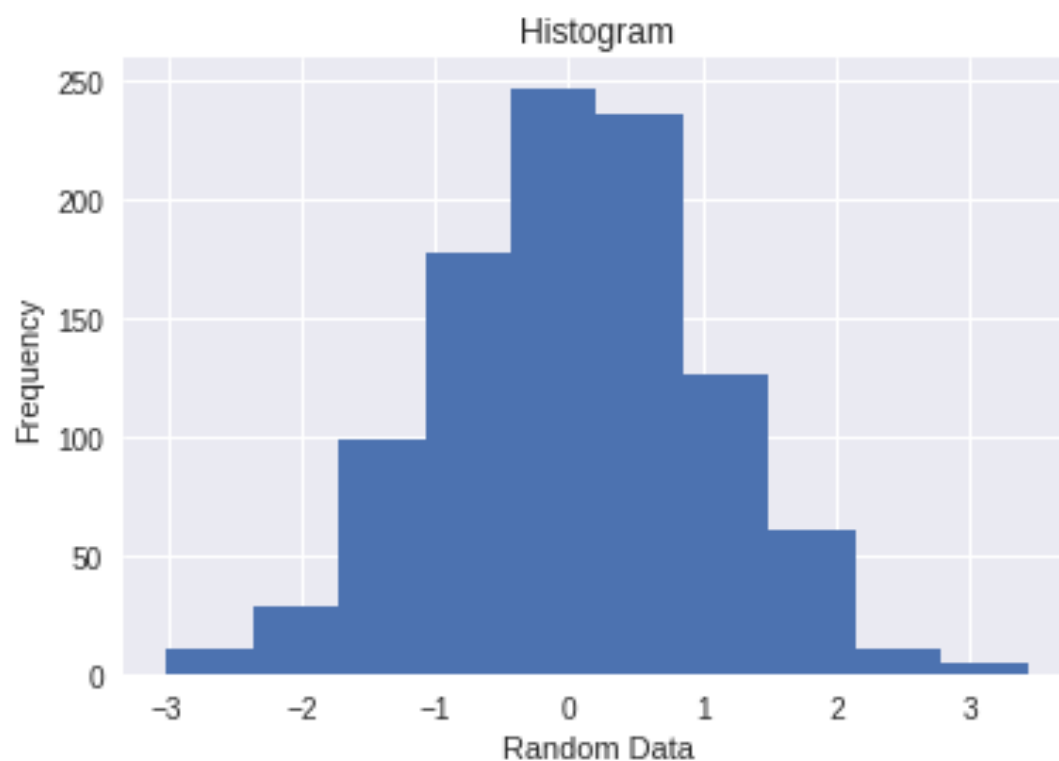






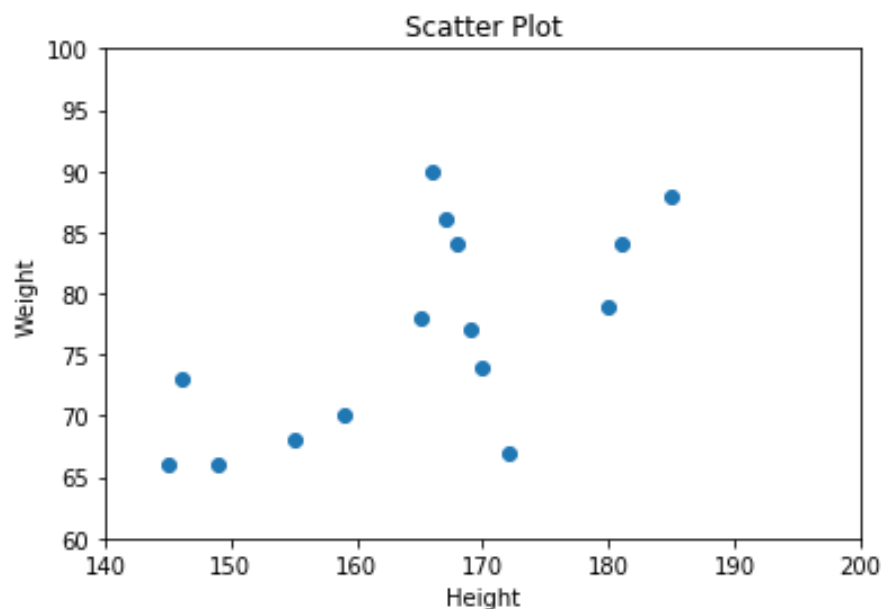
```
x = np.random.randn(1000)

plt.title("Histogram")
plt.xlabel("Random Data")
plt.ylabel("Frequency")
plt.hist(x,10)
plt.show()
```



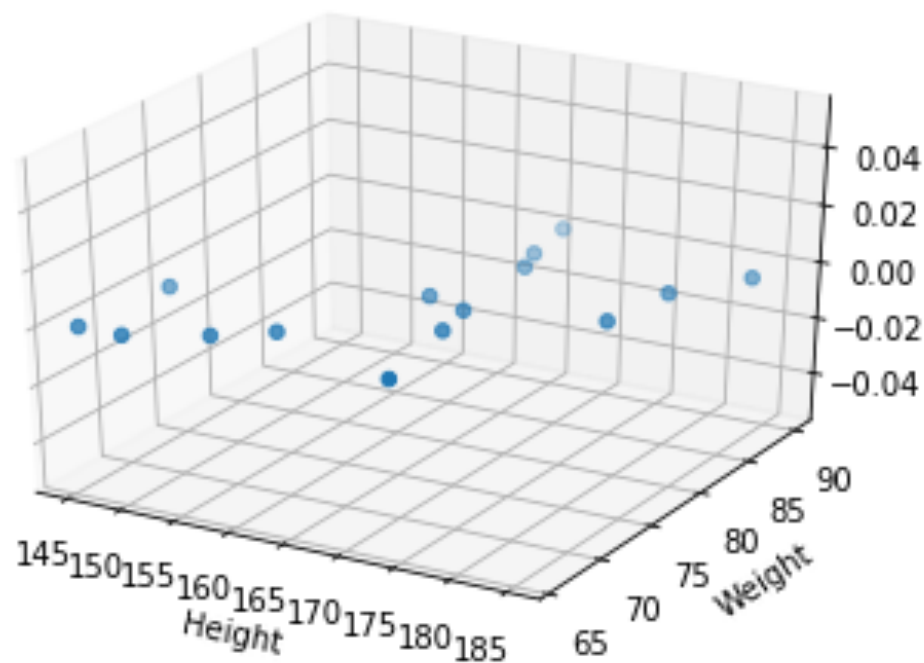


```
height = np.array([167,170,149,165,155,180,166,146,  
                  159,185,145,168,172,181,169])  
weight = np.array([86,74,66,78,68,79,90,73,  
                  70,88,66,84,67,84,77])  
  
plt.xlim(140,200)  
plt.ylim(60,100)  
plt.scatter(height,weight)  
plt.title("Scatter Plot")  
plt.xlabel("Height")  
plt.ylabel("Weight")  
plt.show()
```



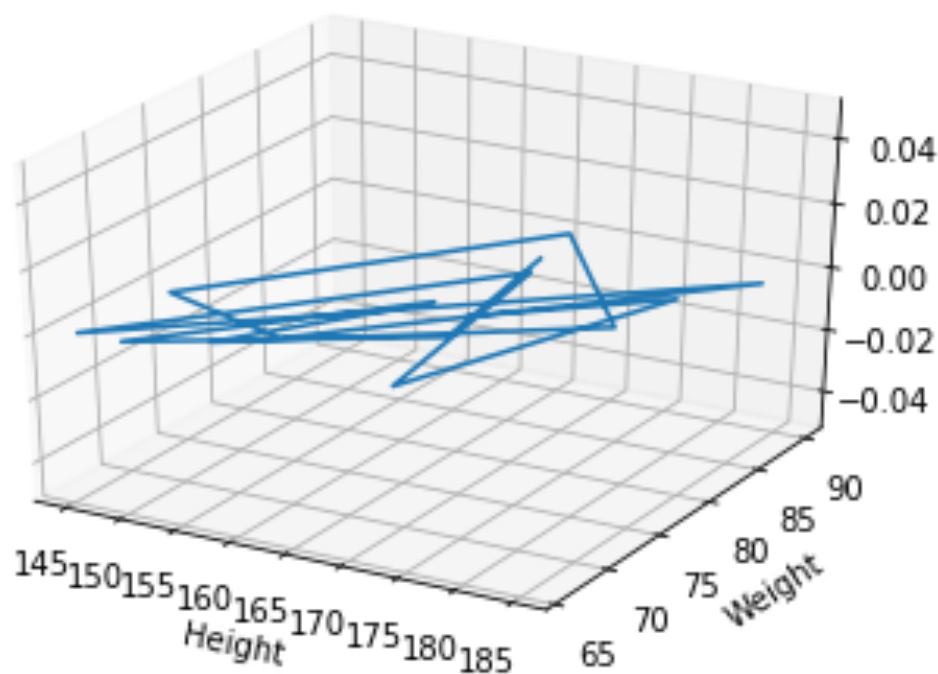


```
ax = plt.axes(projection='3d')  
ax.scatter3D(height, weight)  
ax.set_xlabel("Height")  
ax.set_ylabel("Weight")  
plt.show()
```





```
ax = plt.axes(projection='3d')  
ax.plot3D(height,weight)  
ax.set_xlabel("Height")  
ax.set_ylabel("Weight")  
plt.show()
```



- Killol govani. (2003). Learn basics of Python's powerful Plotting library. Retrieved 15 September, 2019, from <https://towardsdatascience.com/matplotlib-tutorial-learn-basics-of-pythons-powerful-plotting-library-b5d1b8f67596>

# MUCHAS GRACIAS

Eliezer Figueroa  
MCT,MOS,MCSA,MCSE,ITIL,SCF

**UNIBE**

EDUCACIÓN  
CONTINUA

