

Instituto Tecnológico y de Estudios Superiores de Monterrey

Campus, Querétaro



***Herramientas computacionales: el arte de la analítica
(Gpo 201)***

Actividad: Regresión lineal

Estudiantes:

Karen Cebreros López

A01704254

Profesor:

Pedro Pérez

Fecha de entrega:

Jueves 12 de mayo del 2022

Importa librerías necesarias y carga del archivo 'presion.csv'

```
[1] import matplotlib.pyplot as plt
import seaborn as sns; sns.set()
import pandas as pd
import numpy as np
from sklearn.linear_model import LinearRegression
```

```
[2] from google.colab import files

uploaded = files.upload()

for fn in uploaded.keys():
    print('User uploaded file "{name}" with length {length} bytes'.format(
        name=fn, length=len(uploaded[fn])))
```

Choose Files presion.csv

- presion.csv(text/csv) - 772 bytes, last modified: 5/12/2022 - 100% done

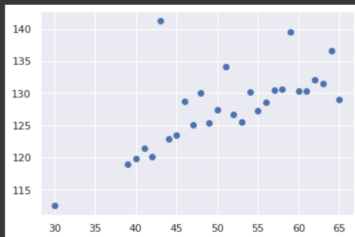
Saving presion.csv to presion.csv
User uploaded file "presion.csv" with length 772 bytes

```
[3] df_p = pd.read_csv('presion.csv')
df_p.head(6)
```

	Age	Average of ap_hi	Average of ap_lo
0	30	112.500000	72.500000
1	39	119.029340	88.229829
2	40	119.789630	85.858889
3	41	121.490862	90.344648
4	42	120.163872	89.887957
5	43	141.294203	93.388406

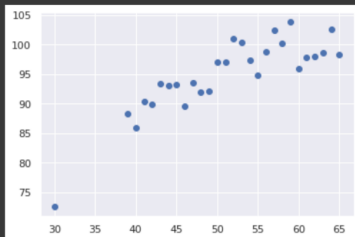
Graficar la información de la edad y presión alta

```
[4] plot_a_aphi = plt.scatter(df_p['Age'], df_p['Average of ap_hi']);
```



Graficar la información de la edad y presión baja

```
[5] plot_a_aplo = plt.scatter(df_p['Age'], df_p['Average of ap_lo']);
```



PRESIÓN ALTA: Genera una regresión lineal para la gráfica "edad-presión alta" Gráfica los datos reales contra los obtenidos con el modelo. Se debe visualizar los datos reales (azúl), recta del modelo (negro) y distancias entre ambos. (verde)

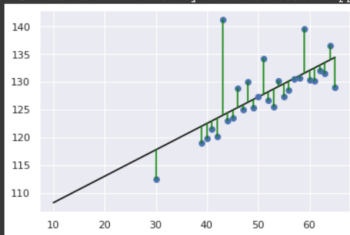
```
[6] del_hi = LinearRegression(fit_intercept=True)

del_hi.fit(df_p['Age'][:, np.newaxis], df_p['Average of ap_hi'])

it = np.linspace(65, 10, 1000)
it = model_hi.predict(xfit[:, np.newaxis])

ot_a_aphi
t.plot(xfit, yfit, color="black");
t.plot(df_p['Age'], df_p['Average of ap_hi'], 'o')
t.plot(np.vstack((df_p['Age'],df_p['Age'])), np.vstack((df_p['Average of ap_hi'], model_hi.predict(df_p['Age'][:, np.newaxis]))), color="green");

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:3: FutureWarning: Support for multi-dimensional indexing (e.g. `obj[:, None]`) is deprecated. This is separate from the ipykernel package so we can avoid doing imports until
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:11: FutureWarning: Support for multi-dimensional indexing (e.g. `obj[:, None]`) is deprecated. This is added back by InteractiveShellApp.init_path()
```



¿Cuál es el valor de a y cuál es el valor de b para la presión alta?

```
[7] a_hi = model_hi.coef_[0]
b_hi = model_hi.intercept_
print("a:", a_hi)
print("b:", b_hi)
```

PRESIÓN BAJA: Genera una regresión lineal para la gráfica "edad-presión baja" Gráfica los datos reales contra los obtenidos con el modelo. Se debe visualizar los datos reales (azúl), recta del modelo (negro) y distancias entre ambos. (verde)

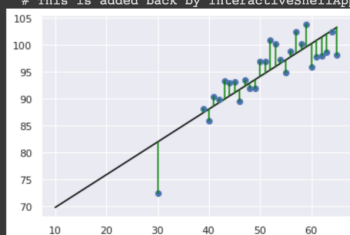
```
[8] del_lo = LinearRegression(fit_intercept=True)

del_lo.fit(df_p['Age'][:, np.newaxis], df_p['Average of ap_lo'])

it = np.linspace(65, 10, 1000)
it = model_lo.predict(xfit[:, np.newaxis])

ot_a_aplo
t.plot(xfit, yfit, color="black");
t.plot(df_p['Age'], df_p['Average of ap_lo'], 'o')
t.plot(np.vstack((df_p['Age'],df_p['Age'])), np.vstack((df_p['Average of ap_lo'], model_lo.predict(df_p['Age'][:, np.newaxis]))), color="green");

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:3: FutureWarning: Support for multi-dimensional indexing (e.g. `obj[:, None]`) is deprecated. This is separate from the ipykernel package so we can avoid doing imports until
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:11: FutureWarning: Support for multi-dimensional indexing (e.g. `obj[:, None]`) is deprecated. This is added back by InteractiveShellApp.init_path()
```



¿Cuál es el valor de a y cuál es el valor de b para la presión baja?

```
[9] a_lo = model_lo.coef_[0]
b_lo = model_lo.intercept_
print("a:", a_lo)
print("b:", b_lo)

a: 0.6089810580238237
b: 63.726200409422745
```

¿Cual es la presión arterial atal y baja para una persona de cierta edad? Genera dos funciones que calculen los anterior.

```
[10] def pressure_high(age):  
      return (a_hi*age) + b_hi  
  
      query_age= 76  
      pressure_high(query_age)
```

139.70194835946515

```
def pressure_low(age):  
      return (a_lo*age) + b_lo  
  
      query_age= 76  
      pressure_low(query_age)
```

110.00876081923334