

Nombre: Aldo Tena García

Matrícula: A01275222

```
import matplotlib.pyplot as plt
import seaborn as sns; sns.set()

import numpy as np
import pandas as pd

from sklearn.linear_model import LinearRegression

# Carga el conjunto de datos al ambiente de Google Colab y muestra los primeros
# 6 renglones.
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])))
```

Elegir archivos presion.csv

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

Saving presion.csv to presion (7).csv
User uploaded file "presion.csv" with length 772 bytes

```
df = pd.read_csv('presion.csv')
df.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	

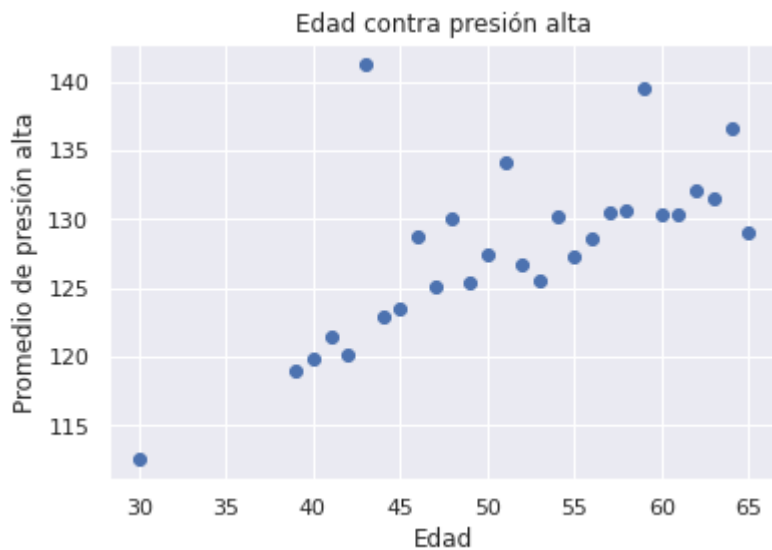
Age: Edad de la persona.

Average of ap_hi: Promedio de presión alta.

Average of ap_lo: Promedio de presión baja.

```
# Grafica la información de la edad y presión alta
x1 = df['Age']
y1 = df['Average of ap_hi']
plt.scatter(x1, y1);
plt.title('Edad contra presión alta')
plt.xlabel('Edad')
plt.ylabel('Promedio de presión alta')
```

```
Text(0, 0.5, 'Promedio de presión alta')
```



```
# Grafica la información de la edad y presión baja
x2 = df['Age']
y2 = df['Average of ap_lo']
plt.scatter(x2, y2);
plt.title('Edad contra presión baja')
plt.xlabel('Edad')
plt.ylabel('Promedio de presión baja')
```

```
Text(0, 0.5, 'Promedio de presión baja')
```



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

```
105
```

```
x1.shape
```

```
(28,)
```

```
30 35 40 45 50 55 60 65
```

```
x1_new = x1[:, np.newaxis]
```

```
x1_new.shape
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: FutureWarning: Support 1
    """Entry point for launching an IPython kernel.
(28, 1)
```

```
np.linspace(0, 10, 10)
```

```
array([ 0.          ,  1.11111111,  2.22222222,  3.33333333,  4.44444444,
        5.55555556,  6.66666667,  7.77777778,  8.88888889, 10.          ])
```

```
np.vstack([x1,x1])
```

```
array([[30, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53,
        54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65],
       [30, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53,
        54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65]])
```

```
model1 = LinearRegression(fit_intercept=True)
```

```
model1.fit(x1[:, np.newaxis], y1)
```

```
xfit1 = np.linspace(0, 10, 1000)
```

```
yfit1 = model1.predict(xfit1[:, np.newaxis])
```

```
plt.scatter(x1, y1)
```

```
plt.plot(xfit1, yfit1, color="black");
```

```
plt.plot(x1, y1, 'o')
```

```
plt.plot(np.vstack([x1,x1]), np.vstack([y1, model1.predict(x1[:, np.newaxis])])), color="green"
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:3: FutureWarning: Sup
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: Su
# This is added back by InteractiveShellApp.init_path()
```



```
print("Pendiente de la presión alta:      ", model1.coef_[0])
print("Intersección de la presión alta:", model1.intercept_)
```

```
Pendiente de la presión alta:      0.47769702977669154
Intersección de la presión alta: 103.3969740964366
```

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

```
x2.shape
```

```
(28,)
```

```
x2_new = x2[:, np.newaxis]
x2_new.shape
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: FutureWarning: Support 1
"""Entry point for launching an IPython kernel.
(28, 1)
```

```
np.linspace(0, 10, 10)
```

```
array([ 0.          ,  1.11111111,  2.22222222,  3.33333333,  4.44444444,
        5.55555556,  6.66666667,  7.77777778,  8.88888889, 10.          ])
```

```
np.vstack([x2,x2])
```

```
array([[30, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53,
        54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65],
       [30, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53,
        54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65]])
```

```
[30, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53,
54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65]])
```

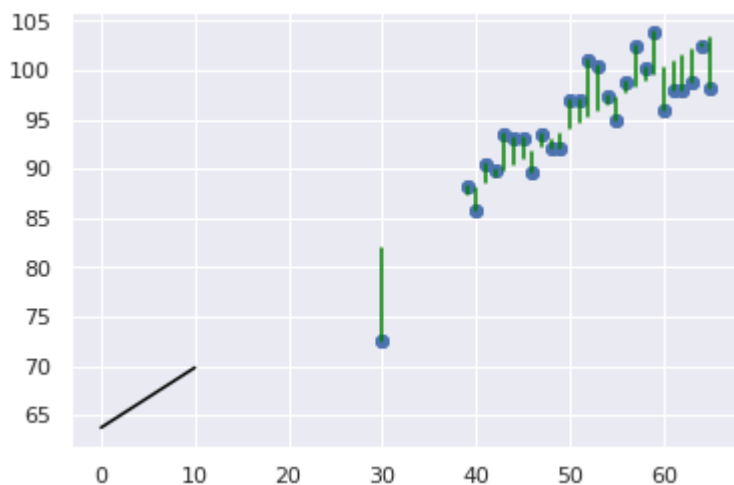
```
model2 = LinearRegression(fit_intercept=True)

model2.fit(x2[:, np.newaxis], y2)

xfit2 = np.linspace(0, 10, 1000)
yfit2 = model2.predict(xfit2[:, np.newaxis])

plt.scatter(x2, y2)
plt.plot(xfit2, yfit2, color="black");
plt.plot(x2, y2, 'o')
plt.plot(np.vstack([x2,x2]), np.vstack([y2, model2.predict(x2[:, np.newaxis])])), color="green"
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:3: FutureWarning: Sup
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: Su
# This is added back by InteractiveShellApp.init_path()
```



```
print("Pendiente de la presión baja:    ", model2.coef_[0])
print("Intersección de la presión baja:", model2.intercept_)
```

```
Pendiente de la presión baja:    0.6089810580238237
Intersección de la presión baja: 63.726200409422745
```

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

▼ Genera dos funciones que calculen los anterior.

```
#Función para la presión alta
def pressure_high(age):
    ph = (model1.coef_[0] * age) + model1.intercept_
    return ph
```

```
query_age= 76
pressure_high(query_age)
```

139.70194835946515

```
#Función para la presión baja
def pressure_low(age):
    pl = (model2.coef_[0] * age) + model2.intercept_
    return pl
```

```
query_age= 76
pressure_low(query_age)
```

110.00876081923334

✓ 0 s completado a las 18:01

