

ACTIVIDAD 08 - REGRESIÓN LINEAL

```
[ ] # Carga las librerías necesarias.
```

```
import matplotlib.pyplot as plt
import seaborn as sns; sns.set()
import numpy as np
import pandas as pd
```

```
[ ] # 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])))
```

```
df = pd.read_csv('presion.csv')
df.head(6)
```

Elegir archivos Sin archivos seleccionados Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

Saving presion.csv to presion.csv

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

	Age	Average of ap_hi	Average of ap_lo
0	30	112.500000	72.500000
1	39	119.029340	88.229829

```
[ ] x = df["Age"]
    y1 = df["Average of ap_hi"]
    y2 = df["Average of ap_lo"]
```

```
[ ] # Grafica la información de la edad y presión alta
```

```
fig = plt.figure(figsize = (6, 4))
```

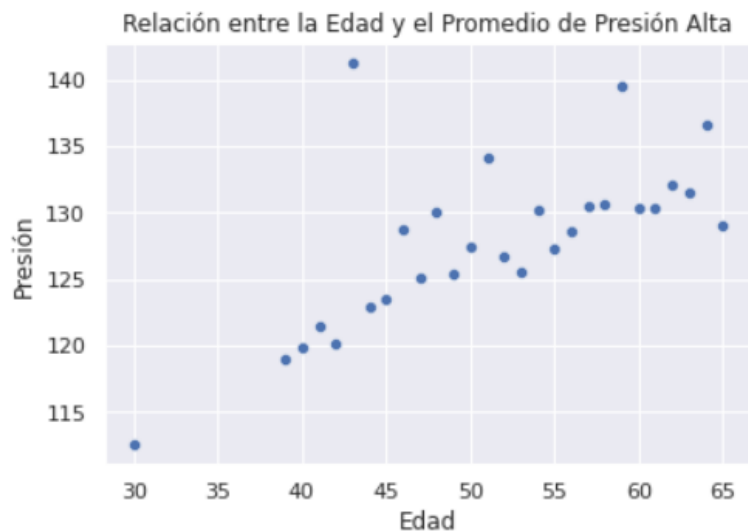
```
sns.scatterplot(data = df, x = "Age", y = "Average of ap_hi");
```

```
plt.title('Relación entre la Edad y el Promedio de Presión Alta')
```

```
plt.xlabel('Edad')
```

```
plt.ylabel('Presión')
```

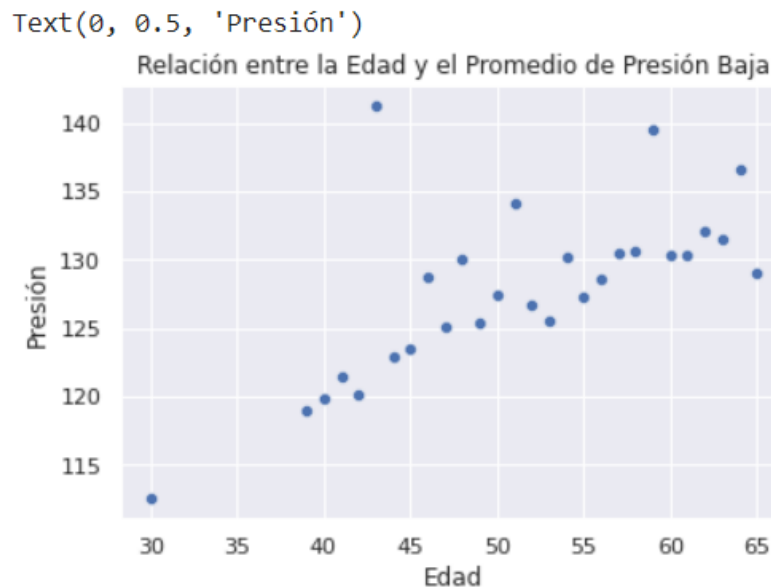
```
Text(0, 0.5, 'Presión')
```



```
[ ] # Grafica la información de la edad y presión baja

fig = plt.figure(figsize = (6, 4))
sns.scatterplot(data = df, x = "Age", y = "Average of ap_hi");

plt.title('Relación entre la Edad y el Promedio de Presión Baja')
plt.xlabel('Edad')
plt.ylabel('Presión')
```



```
[ ] # ¿Cuál es el valor de a y cuál es el valor de b para la presión alta?

from sklearn.linear_model import LinearRegression

model1 = LinearRegression(fit_intercept = True)
model1.fit(x[:, np.newaxis], y1)

print("Model slope: ", model1.coef_[0])
print("Model intercept: ", model1.intercept_)

Model slope:  0.47769702977669154
Model intercept:  103.3969740964366
<ipython-input-6-b4b6a858388e>:6: FutureWarning: Support for multi-dimensional indexing (e.g. `obj[:, None]`) is deprecated and will be removed in a f
model1.fit(x[:, np.newaxis], y1)
```

Por lo tanto, $a = 0.47769702977669154$ y $b = 103.3969740964366$

```
[ ] # ¿Cuál es el valor de a y cuál es el valor de b para la presión baja?

model2 = LinearRegression(fit_intercept = True)

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

print("Model slope: ", model2.coef_[0])
print("Model intercept: ", model2.intercept_)

Model slope:  0.6089810580238237
Model intercept:  63.726200409422745
```

```
[ ] # Presión alta

xfit = np.linspace(0, 70, 1000)
yfit = model1.predict(xfit[:, np.newaxis])

plt.scatter(x, y1)
plt.plot(xfit, yfit, color = "black");
plt.plot(x, y1, 'o')
plt.plot(np.vstack([x,x]),
         np.vstack([y1, model1.predict(x[:, np.newaxis])]), color = "green");

plt.title('Relación entre la Edad y el Promedio de Presión Alta')
plt.xlabel('Edad')
plt.ylabel('Presión')
```

<ipython-input-10-067a30789ada>:10: FutureWarning: Support for multi-dimensional indexing (e.g. `obj[:, None]`) is deprecated in favor of `obj[:, np.newaxis]`
 np.vstack([y1, model1.predict(x[:, np.newaxis])]), color = "green");
 Text(0, 0.5, 'Presión')



```
[ ] # Presión baja

xfit = np.linspace(0, 70, 1000)
yfit = model2.predict(xfit[:, np.newaxis])

plt.scatter(x, y2)
plt.plot(xfit, yfit, color = "black");
plt.plot(x, y2, 'o')
plt.plot(np.vstack([x, x]), np.vstack([y2, model2.predict(x[:, np.newaxis])]), color = "green");

plt.title('Relación entre la Edad y el Promedio de Presión Baja')
plt.xlabel('Edad')
plt.ylabel('Presión')
```

<ipython-input-12-7ea534fdb057>:9: FutureWarning: Support for multi-dimensional indexing (e.g. `obj[:, None]`) is deprecated in favor of `obj[:, np.newaxis]`
 plt.plot(np.vstack([x, x]), np.vstack([y2, model2.predict(x[:, np.newaxis])]), color = "green");
 Text(0, 0.5, 'Presión')



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

```
[ ] def pressure_low(age):  
    low = (model2.coef_[0]*age) + (model2.intercept_)  
    return low
```

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

110.00876081923334

```
[ ] def pressure_high(age):  
    high = (model1.coef_[0]*age) + (model1.intercept_)  
    return high
```

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

139.70194835946515