

▼ Actividad - Regresión Lineal

- **Nombre:** Victor Noel Madrid Castillo
- **Matrícula:** A01562528

Entregar: Archivo PDF de la actividad, así como el archivo .ipynb en tu repositorio.

Nota: Recuerda habrá una penalización de **50** puntos si la actividad fue entregada fuera de la fecha límite.

Importante:

- Colocar nombres de ejes en gráficas.
- Títulos en las gráficas.
- Contestar cada pregunta.

Carga el conjunto de datos `presion.csv` (se encuentra en el repositorio de la clase) y realiza un análisis estadístico de las variables.

```
1 # Carga las librerías necesarias.
2 import pandas as pd
3 import matplotlib.pyplot as plt
4 import seaborn as sns; sns.set()
5 import numpy as np
6 from sklearn.linear_model import LinearRegression

1 # Carga el conjunto de datos al ambiente de Google Colab y muestra los primeros
2 # 6 renglones.
3 from google.colab import files
4
5 uploaded = files.upload()
6
7 for fn in uploaded.keys():
8     print('User uploaded file "{name}" with length {length} bytes'.format(
9         name=fn, length=len(uploaded[fn])))
10
11 df = pd.read_csv('presion.csv')
12 df.head(6)
```

Choose Files presion.csv

- **presion.csv**(text/csv) - 801 bytes, last modified: 3/21/2023 - 100% done

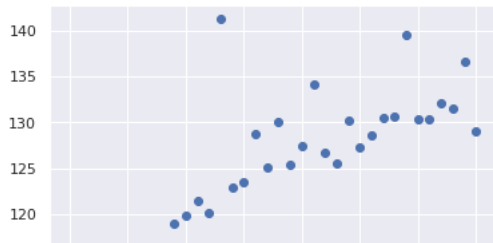
Saving presion.csv to presion (2).csv
User uploaded file "presion.csv" with length 801 bytes

	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	

El conjunto de datos contiene información demográfica sobre los asegurados en una compañía de seguros:

- **Age:** Edad de la persona.
- **Average of ap_hi:** Promedio de presión alta.
- **Average of ap_lo:** Promedio de presión baja.

```
1 # Grafica la información de la edad y presión alta
2 x = df['Age'].values
3 y = df['Average of ap_hi'].values
4 plt.scatter(x, y);
```



Genera una regresión lineal para obtener una aproximación de la ecuación

$$y = ax + b$$

donde a se conoce comúnmente como **pendiente**, y b se conoce comúnmente como **intersección**, tanto para presión alta como la presión baja.

```
1 # ¿Cuál es el valor de a y cuál es el valor de b para la presión alta?
2 x = df['Age'].values
3 y = df['Average of ap_hi'].values
4
5 modela = LinearRegression(fit_intercept=True)
6 modela.fit(x[:, np.newaxis], df['Average of ap_hi'].values)
7 print("Model slope:      ", modela.coef_[0])
8 print("Model intercept:", modela.intercept_)
```

```
Model slope:      0.47769702977669154
Model intercept: 103.3969740964366
```

```
1 # ¿Cuál es el valor de a y cuál es el valor de b para la presión baja?
2 x = df['Age'].values
3 y = df['Average of ap_lo'].values
4
5 modelb = LinearRegression(fit_intercept=True)
6 modelb.fit(x[:, np.newaxis], df['Average of ap_lo'].values)
7 print("Model slope:      ", modelb.coef_[0])
8 print("Model intercept:", modelb.intercept_)
```

```
Model slope:      0.6089810580238237
Model intercept: 63.726200409422745
```

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)

```
1 # Presión alta
2 x = df['Age'].values
3 y = df['Average of ap_hi'].values
4 plt.plot(x,y, color='green', )
5 plt.scatter(x, y);
6
7 modela = LinearRegression(fit_intercept=True)
8 modela.fit(x[:, np.newaxis], df['Average of ap_hi'].values)
9
10 yfit = modela.predict(x[:, np.newaxis])
11 plt.plot(x, yfit, color='black')
12
13 print("Model slope:      ", modela.coef_[0])
14 print("Model intercept:", modela.intercept_)
```

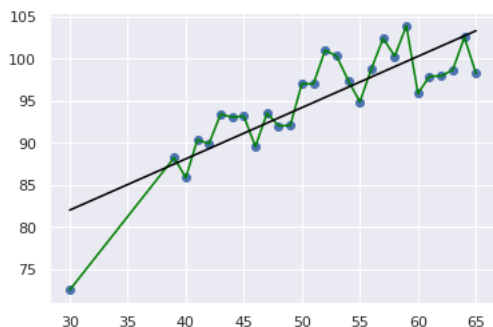
```

Model slope: 0.6089810580238237
Model intercept: 63.726200409422745

1 # Presión baja
2 x = df['Age'].values
3 y = df['Average of ap_lo'].values
4 plt.plot(x,y, color='green', )
5 plt.scatter(x, y);
6
7 modelb = LinearRegression(fit_intercept=True)
8 modelb.fit(x[:, np.newaxis], df['Average of ap_lo'].values)
9
10 yfit = modelb.predict(x[:, np.newaxis])
11 plt.plot(x, yfit, color='black')
12
13 print("Model slope: ", modelb.coef_[0])
14 print("Model intercept:", modelb.intercept_)

```

Model slope: 0.6089810580238237
Model intercept: 63.726200409422745



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

```

1 def pressure_high(age):
2     return modela.predict([[age]])
3
4 query_age= 76
5 pressure_high(query_age)
6
array([139.70194836])

1 def pressure_low(age):
2     return modelb.predict([[age]])
3
4 query_age= 76
5 pressure_low(query_age)
6
array([110.00876082])

```

✓ 0 s completado a las 21:50

No se ha podido establecer conexión con el servicio reCAPTCHA. Comprueba tu conexión a Internet y vuelve a cargar la página para ver otro reCAPTCHA.

● ✕