Actividad - Regresión Lineal

- **Nombre:**Diego Arturo Padilla Domínguez
- **Matrícula:**A01552594

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.

```
# Carga las librerías necesarias.
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns; sns.set()
```

```
# Carga el conjunto de datos al ambiente de Google Colab y muestra los primeros
# 6 renglones.
from google.colab import drive
drive.mount('/gdrive')
%cd /gdrive/MyDrive/SemanaTec/Repositorio1/
```

```
Drive already mounted at /gdrive; to attempt to forcibly remount, call drive.mount("/gdr
/gdrive/MyDrive/SemanaTec/Repositorio1
```

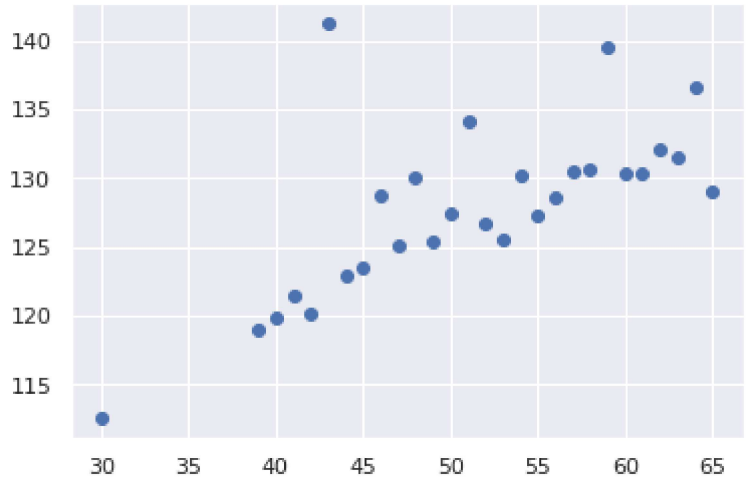
```
df = pd.read_csv('presion.csv')
df.head(6)
```

	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

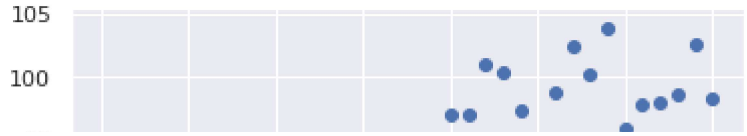
El conjunto de datos contiene información demográfica sobre los asegurados en una compañía de seguros:

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

```
# Grafica la información de la edad y presión alta
y = df["Average of ap_hi"]
x = df["Age"]
plt.scatter(x, y);
```




```
# Grafica la información de la edad y presión baja
y2 = df["Average of ap_lo"]
x2 = df["Age"]
plt.scatter(x2, y2);
```



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.


```
x.shape
x2.shape
```

```
(28,)
```

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

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: FutureWarning: Support t
  """Entry point for launching an IPython kernel.
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:3: FutureWarning: Support t
  This is separate from the ipykernel package so we can avoid doing imports until
(28, 1)
```

```
np.linspace(0, 10, 100)
```

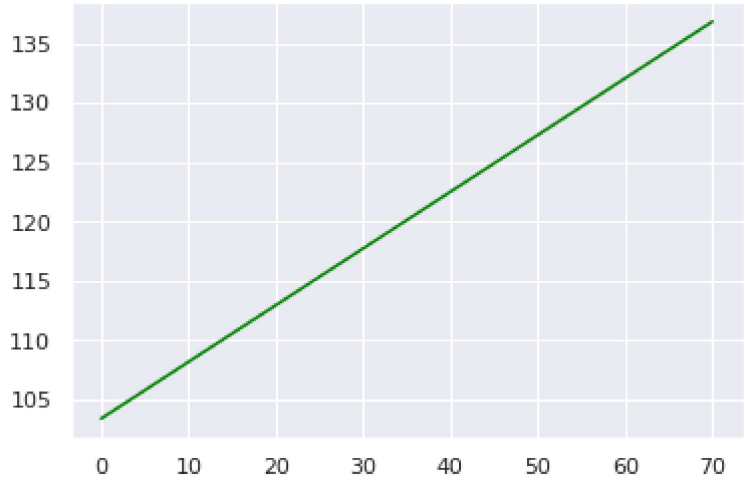
```
array([ 0.        ,  0.1010101 ,  0.2020202 ,  0.3030303 ,  0.4040404 ,
        0.50505051,  0.60606061,  0.70707071,  0.80808081,  0.90909091,
        1.01010101,  1.11111111,  1.21212121,  1.31313131,  1.41414141,
        1.51515152,  1.61616162,  1.71717172,  1.81818182,  1.91919192,
        2.02020202,  2.12121212,  2.22222222,  2.32323232,  2.42424242,
        2.52525253,  2.62626263,  2.72727273,  2.82828283,  2.92929293,
        3.03030303,  3.13131313,  3.23232323,  3.33333333,  3.43434343,
        3.53535354,  3.63636364,  3.73737374,  3.83838384,  3.93939394,
        4.04040404,  4.14141414,  4.24242424,  4.34343434,  4.44444444,
        4.54545455,  4.64646465,  4.74747475,  4.84848485,  4.94949495,
        5.05050505,  5.15151515,  5.25252525,  5.35353535,  5.45454545,
        5.55555556,  5.65656566,  5.75757576,  5.85858586,  5.95959596,
        6.06060606,  6.16161616,  6.26262626,  6.36363636,  6.46464646,
        6.56565657,  6.66666667,  6.76767677,  6.86868687,  6.96969697,
        7.07070707,  7.17171717,  7.27272727,  7.37373737,  7.47474747,
        7.57575758,  7.67676768,  7.77777778,  7.87878788,  7.97979798,
        8.08080808,  8.18181818,  8.28282828,  8.38383838,  8.48484848,
        8.58585859,  8.68686869,  8.78787879,  8.88888889,  8.98989899,
        9.09090909,  9.19191919,  9.29292929,  9.39393939,  9.49494949,
        9.5959596 ,  9.6969697 ,  9.7979798 ,  9.8989899 , 10.        ])
```

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

```
from sklearn.linear_model import LinearRegression

model = LinearRegression(fit_intercept=True)

model.fit(x[:, np.newaxis], y)

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

plt.plot(xfit, yfit, color="green");
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:5: FutureWarning: Support t
  """
```



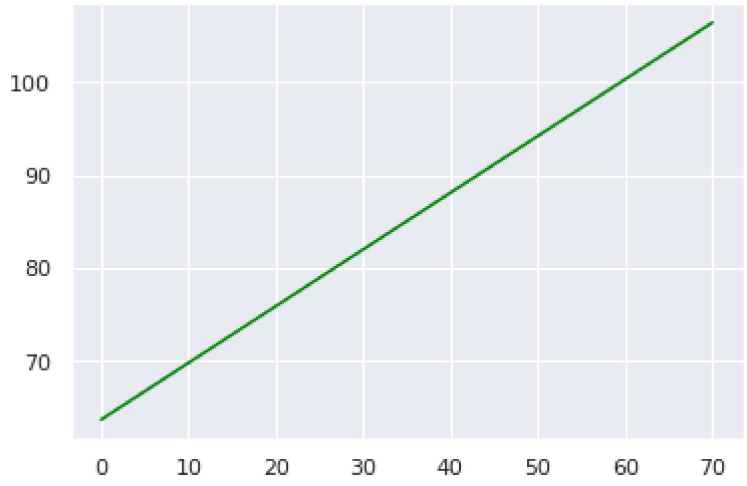

```
print("Model slope:    ", model.coef_[0])
print("Model intercept:", model.intercept_)
```

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

```
from sklearn.linear_model import LinearRegression

model = LinearRegression(fit_intercept=True)

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

```
xfit2 = np.linspace(0, 70, 1000)
yfit2 = model.predict(xfit2[:, np.newaxis])

plt.plot(xfit2, yfit2, color="green");
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:5: FutureWarning: Support t
  """
```




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

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

```
# ¿Cuál es el valor de a y cuál es el valor de b para la presión alta?
a=0.47769702977669154
B=103.3969740964366
```

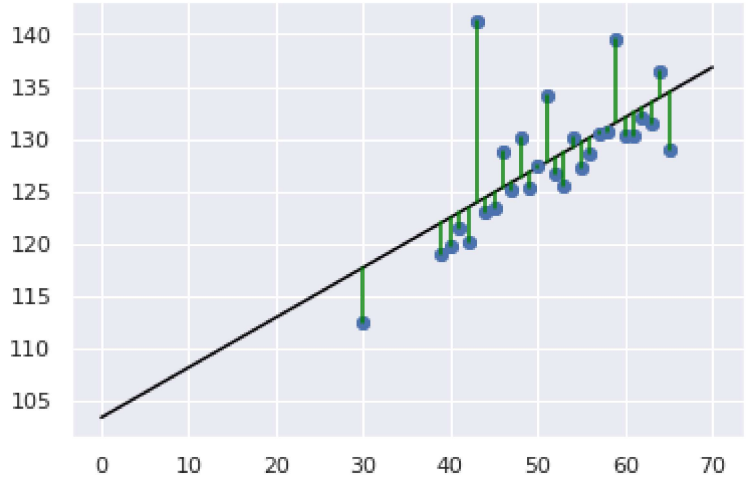
```
# ¿Cuál es el valor de a y cuál es el valor de b para la presión baja?
a: 0.6089810580238237
b: 63.726200409422745
```

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)

```
# Presión alta
from sklearn.linear_model import LinearRegression

model = LinearRegression(fit_intercept=True)

model.fit(x[:, np.newaxis], y)

xfit = np.linspace(0, 70, 1000)
```

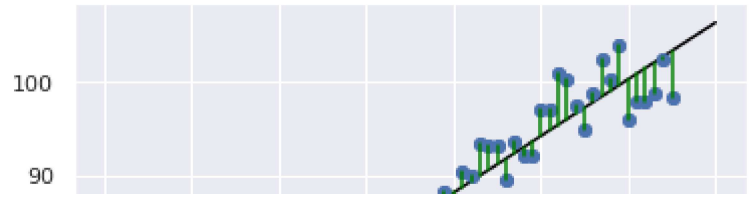
```
yfit = model.predict(xfit[:, np.newaxis])

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

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:6: FutureWarning: Support t

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:14: FutureWarning: Support
```




```
# Presión baja
from sklearn.linear_model import LinearRegression

model = LinearRegression(fit_intercept=True)

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

xfit2 = np.linspace(0, 70, 1000)
yfit2 = model.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, model.predict(x2[:, np.newaxis])]), color="green"
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:6: FutureWarning: Support f

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:14: FutureWarning: Support
```



¿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):
  y=0.47769702977669154*age+103.3969740964366
  return y

query_age= 76
pressure_low(query_age)
```

```
139.70194835946515
```

```
def pressure_high(age):
  y=0.6089810580238237*age+63.726200409422745
  return y

query_age= 76
pressure_high(query_age)
```

```
110.00876081923334
```

✓ 0 s se ejecutó 16:38