```
from google.colab import drive
drive.mount('/gdrive')
```

```
Mounted at /gdrive
```

```
%cd /gdrive/MyDrive/TC1002S/
#Importar el Modulo para leer JSON
import json

# Lectura del archivo
with open('credentials.json', 'r') as myfile:
    data = myfile.read()

# Leer el formato del archivo
obj = json.loads(data)

# Vamos a guardar los datos en estas variables
GIT_USERNAME = obj['user']

# token
GIT_TOKEN = obj['token']

# Repo
GIT_REPO = obj['repo']

# Creamos la ruta al repositorio de nuestra cuenta
GIT_PATH = "https://" + GIT_USERNAME + ":" + GIT_TOKEN + "@github.com/" +\
            GIT_USERNAME + "/" + GIT_REPO + ".git"

print(GIT_PATH)
```

```
/gdrive/MyDrive/TC1002S
https://DanyGuti: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@github.com/DanyGuti/SemanaTecTC100
```

```
%cd SemanaTecTC1002S/
```

```
/gdrive/MyDrive/TC1002S/SemanaTecTC1002S
```

```
!git remote -v
```

```
cursoFuente     https://github.com/DanyGuti/SemanaTecTC1002S.git (fetch)
cursoFuente     https://github.com/DanyGuti/SemanaTecTC1002S.git (push)
origin  https://DanyGuti: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@github.com/DanyGuti/Semana
origin  https://DanyGuti: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@github.com/DanyGuti/Semana
```

```
!git pull
```

```
remote: Enumerating objects: 7, done.
remote: Counting objects: 100% (7/7), done.
remote: Compressing objects: 100% (5/5), done.
remote: Total 6 (delta 2), reused 3 (delta 1), pack-reused 0
Unpacking objects: 100% (6/6), 560.79 KiB | 1.56 MiB/s, done.
From https://github.com/DanyGuti/SemanaTecTC1002S
   4668d8f..c46a893  main       -> cursoFuente/main
```

```
Updating 4668d8f..c46a893
Fast-forward
 Act7Collab.ipynb                  | 1220 ++++++++++++++++++++++++++++++++++++
 Act7Collab.ipynb_Colaboratory.pdf |  Bin 0 -> 460369 bytes
 2 files changed, 1220 insertions(+)
 create mode 100644 Act7Collab.ipynb
 create mode 100644 Act7Collab.ipynb_Colaboratory.pdf
```

```
!git status
```

```
On branch main
Your branch is up to date with 'cursoFuente/main'.

nothing to commit, working tree clean
```

Actividad 8 - Regresión Lineal

- **Nombre: Daniel Gutiérrez Gómez**
- **Matrícula: A01068056**
- 03/23/23

Importando las librerías necesarias para realizar la actividad de la regresión lineal:

```
import pandas as pd
import numpy as np
from scipy import stats
import matplotlib.pyplot as plt
import seaborn as sns; sns.set_theme()
import numpy as np
from sklearn.linear_model import LinearRegression
```

Cargar el conjunto de datos de presion.csv para poder realizar el análisis de regresión lineal

```
humans_pressure = pd.read_csv('./datasets/presion.csv')
```

Cargar el conjunto de datos de los primeros 6 renglones

```
display(humans_pressure.iloc[:6])
```

	Age	Average of ap_hi	Average of ap_lo
0	30	112.500000	72.500000

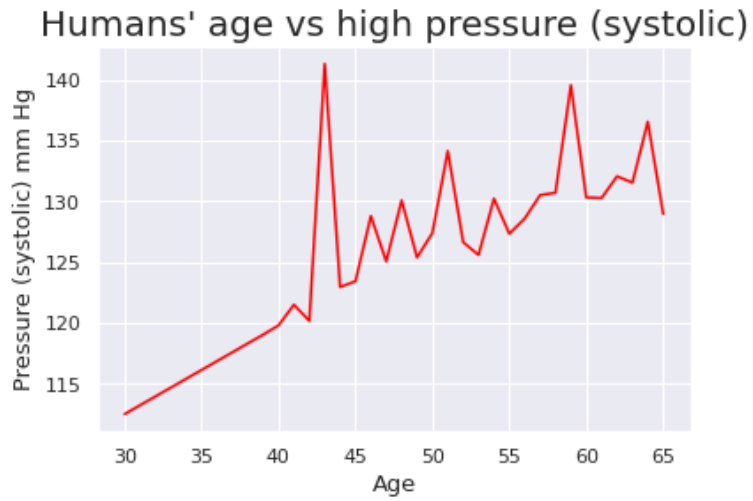
El dataset contiene lo siguiente:

- **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

```
plt.plot(humans_pressure["Age"], humans_pressure["Average of ap_hi"], color='red')
plt.title("Humans' age vs high pressure (systolic)", fontsize=20)
plt.xlabel('Age', fontsize=13)
plt.ylabel('Pressure (systolic) mm Hg', fontsize=13)
```

```
Text(0, 0.5, 'Pressure (systolic) mm Hg')
```



Grafica la información de la edad y presión baja

```
plt.plot(humans_pressure["Age"], humans_pressure["Average of ap_lo"], color='blue')
plt.title("Humans' age vs low pressure (systolic)", fontsize=20)
plt.xlabel('Age', fontsize=13)
plt.ylabel('Pressure (systolic) mm Hg', fontsize=13)
```

```
Text(0, 0.5, 'Pressure (systolic) mm Hg')
```

Generar una regresión lineal

Con la fórmula de: 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.

Gráfico de la presión alta

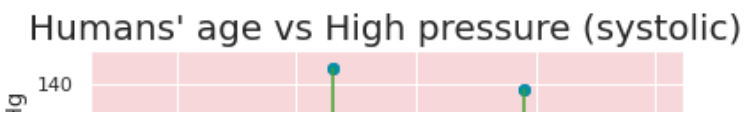
```
x_high = humans_pressure["Age"].values
y_high = humans_pressure["Average of ap_hi"].values


# Set newaxis for x, where there's just one column and the number of rows
# (matricial modeling)
x_new_high_pressure = x_high[:, np.newaxis]
x_new_high_pressure.shape
```

```
(28, 1)
```

```
model_high = LinearRegression(fit_intercept=True)
model_high.fit(x_new_high_pressure, y_high)
xfit_high = np.linspace(25, 70, 1000)
yfit_high = model_high.predict(xfit_high[:, np.newaxis])

ax = plt.axes()
ax.set_facecolor('#F7D7DA')
plt.title("Humans' age vs High pressure (systolic)", fontsize=20)
plt.xlabel('Age', fontsize=13)
plt.ylabel('Pressure (systolic) mm Hg', fontsize=13)
plt.scatter(x_high, y_high)
plt.plot(xfit_high, yfit_high, color='black')
plt.plot(x_high, y_high, 'o', color='#0091A9')
plt.plot(np.vstack([x_high,x_high]), np.vstack([y_high, model_high.predict(x_high[:, np.newaxis])
```




```
print("Model slope (a value):    ", model_high.coef_[0])
print("Model intercept with pressure of (b value):", model_high.intercept_)
```

```
Model slope (a value):     0.47769702977669154
Model intercept with pressure of (b value): 103.3969740964366
```

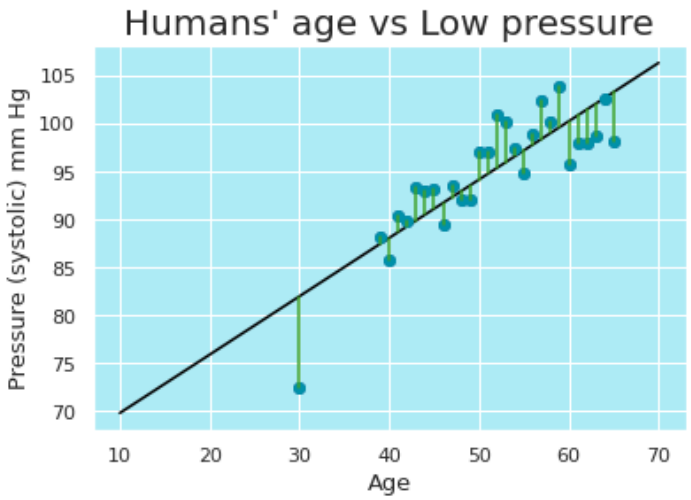
Gráfico de la presión baja


```
x_low = humans_pressure["Age"].values
y_low = humans_pressure["Average of ap_lo"].values


# Set newaxis for x, where there's just one column and the number of rows
# (matricial modeling)
x_new_low_pressure = x_low[:, np.newaxis]
x_new_low_pressure.shape
```

```
(28, 1)
```

```
model_low = LinearRegression(fit_intercept=True)
model_low.fit(x_new_low_pressure, y_low)
xfit_low = np.linspace(10, 70, 1000)
yfit_low = model_low.predict(xfit_low[:, np.newaxis])
ax = plt.axes()
ax.set_facecolor('#ADEBF5')
plt.title("Humans' age vs Low pressure", fontsize=20)
plt.xlabel('Age', fontsize=13)
plt.ylabel('Pressure (systolic) mm Hg', fontsize=13)
plt.scatter(x_low, y_low)
plt.plot(xfit_low, yfit_low, color='black')
plt.plot(x_low, y_low, 'o', color='#0091A9')
plt.plot(np.vstack([x_low,x_low]), np.vstack([y_low, model_low.predict(x_low[:, np.newaxis])]), c
```



```
print("Model slope (a value):    ", model_low.coef_[0])
print("Model intercept with pressure of (b value):", model_low.intercept_)
```

```
Model slope (a value):     0.6089810580238237
Model intercept with pressure of (b value): 63.726200409422745
```

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

Función de la presión arterial alta

```
def pressure_low(age):
  return 0.6089810580238237 * age + 63.726200409422745

query_age = 76
print("El promedio de presión sistólica baja a los 76 es: ", pressure_low(query_age))
```

```
El promedio de presión sistólica baja a los 76 es:  110.00876081923334
```

```
def pressure_high(age):
  return 0.47769702977669154 * age + 103.3969740964366

query_age= 76
print("El promedio de presión sistólica alta a los 76 es: ", pressure_high(query_age))
```

```
El promedio de presión sistólica alta a los 76 es:  139.70194835946515
```

✓ 0 s se ejecutó 13:58