



**Campus Querétaro**

**Herramientas computacionales: el arte de la analítica (Gpo 101)**

Semestre: Febrero-Junio de 2024

**Actividad 7 - Regresión**

**Docente**

*Prof. Pedro Oscar Pérez Murueta*

**Equipo**

Edgar Roann Santillán Bernal | A00572737



```
+ Código + Texto Se guardaron todos los cambios
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.

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

X = df[['Age']]
y = df['Average of ap_hi']

model = LinearRegression()
model.fit(X, y)

a = model.coef_[0]
b = model.intercept_

print("El valor de a (pendiente) para la presión alta es:", a)
print("El valor de b (intersección) para la presión alta es:", b)

El valor de a (pendiente) para la presión alta es: 0.47769702977669154
El valor de b (intersección) para la presión alta es: 103.3969740964366

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

X = df[['Age']]
y = df['Average of ap_lo']

model = LinearRegression()
model.fit(X, y)

a = model.coef_[0]
b = model.intercept_

print("El valor de a (pendiente) para la presión baja es:", a)
print("El valor de b (intersección) para la presión baja es:", b)

El valor de a (pendiente) para la presión baja es: 0.6089810580238237
El valor de b (intersección) para la presión baja es: 63.726200489422745

Gráfica los datos reales contra los obtenidos con el modelo. Se debe visualizar los datos reales (azul), recta del modelo (negro) y distancias entre ambos. (verde)

[9] # Presión alta

X = df[['Age']]
y = df['Average of ap_hi']

model = LinearRegression()
model.fit(X, y)

a = model.coef_[0]
b = model.intercept_

y_pred = model.predict(X)

plt.scatter(X, y, color='blue', label='Datos reales')
plt.plot(X, y_pred, color='black', label='Recta del modelo')

for i in range(len(X)):
    plt.plot([X.iloc[i], X.iloc[i]], [y.iloc[i], y_pred[i]], color='green', linestyle='--')

plt.xlabel('Edad')
plt.ylabel('Presión arterial alta (ap_hi)')
plt.title('Datos reales vs. Modelo de regresión lineal')
plt.legend()

plt.show()
```

```
+ Código + Texto Se guardaron todos los cambios
[7] El valor de a (pendiente) para la presión alta es: 0.47769702977669154
El valor de b (intersección) para la presión alta es: 103.3969740964366

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

X = df[['Age']]
y = df['Average of ap_lo']

model = LinearRegression()
model.fit(X, y)

a = model.coef_[0]
b = model.intercept_

print("El valor de a (pendiente) para la presión baja es:", a)
print("El valor de b (intersección) para la presión baja es:", b)

El valor de a (pendiente) para la presión baja es: 0.6089810580238237
El valor de b (intersección) para la presión baja es: 63.726200489422745

Gráfica los datos reales contra los obtenidos con el modelo. Se debe visualizar los datos reales (azul), recta del modelo (negro) y distancias entre ambos. (verde)

[9] # Presión alta

X = df[['Age']]
y = df['Average of ap_hi']

model = LinearRegression()
model.fit(X, y)

a = model.coef_[0]
b = model.intercept_

y_pred = model.predict(X)

plt.scatter(X, y, color='blue', label='Datos reales')
plt.plot(X, y_pred, color='black', label='Recta del modelo')

for i in range(len(X)):
    plt.plot([X.iloc[i], X.iloc[i]], [y.iloc[i], y_pred[i]], color='green', linestyle='--')

plt.xlabel('Edad')
plt.ylabel('Presión arterial alta (ap_hi)')
plt.title('Datos reales vs. Modelo de regresión lineal')
plt.legend()

plt.show()
```

```
+ Código + Texto Se guardaron todos los cambios
Gráfica los datos reales contra los obtenidos con el modelo. Se debe visualizar los datos reales (azul), recta del modelo (negro) y distancias entre ambos. (verde)

[9] # Presión alta

X = df[['Age']]
y = df['Average of ap_hi']

model = LinearRegression()
model.fit(X, y)

a = model.coef_[0]
b = model.intercept_

y_pred = model.predict(X)

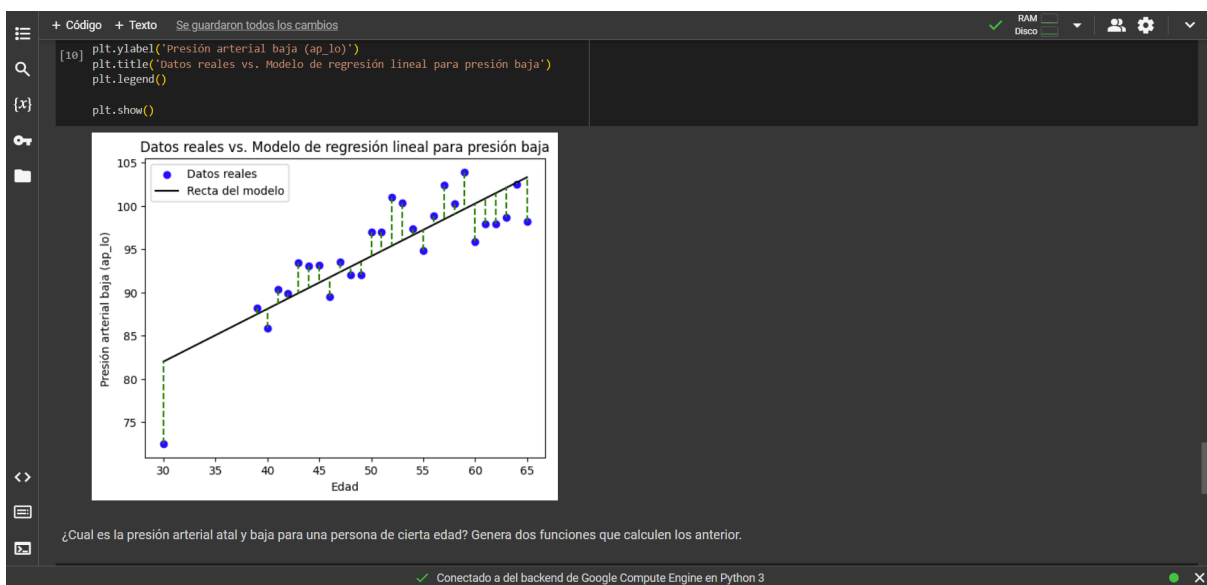
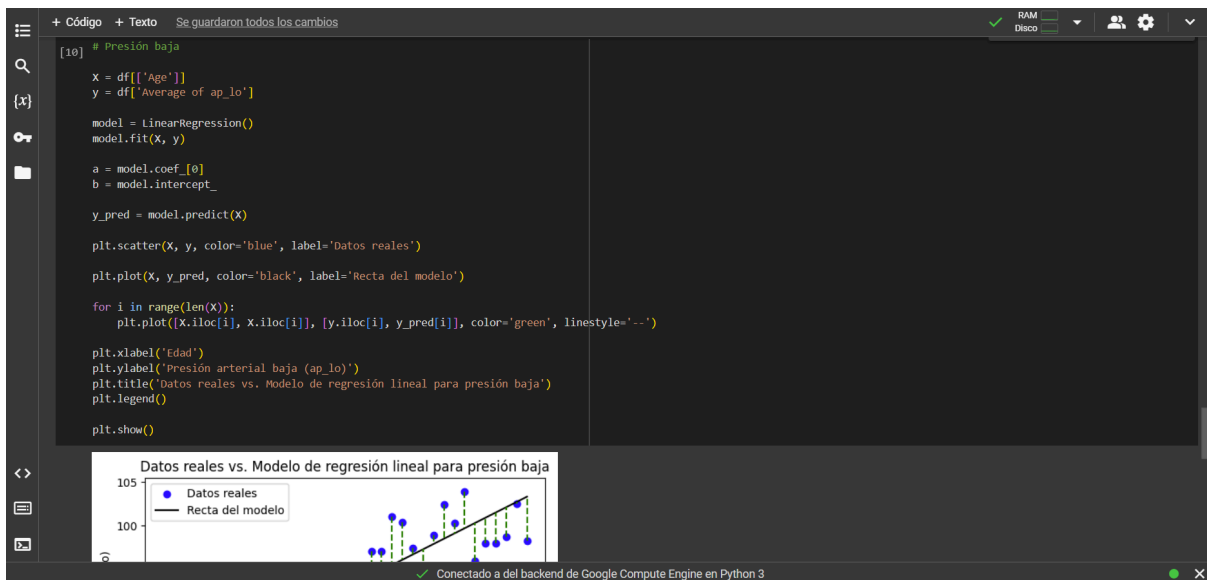
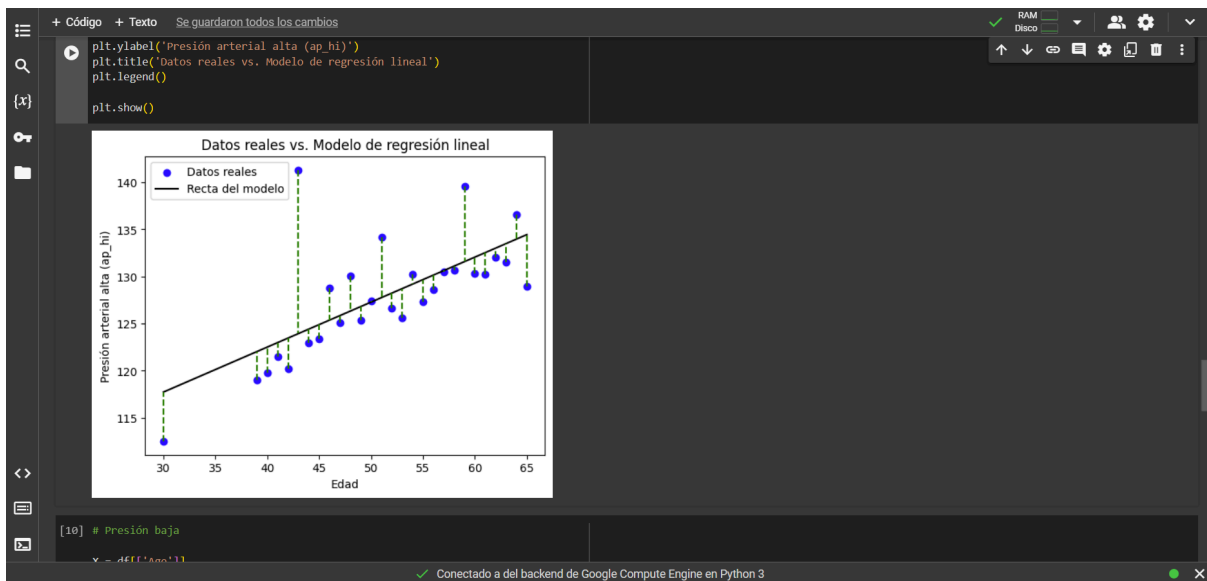
plt.scatter(X, y, color='blue', label='Datos reales')
plt.plot(X, y_pred, color='black', label='Recta del modelo')

for i in range(len(X)):
    plt.plot([X.iloc[i], X.iloc[i]], [y.iloc[i], y_pred[i]], color='green', linestyle='--')

plt.xlabel('Edad')
plt.ylabel('Presión arterial alta (ap_hi)')
plt.title('Datos reales vs. Modelo de regresión lineal')
plt.legend()




plt.show()
```

Datos reales vs. Modelo de regresión lineal

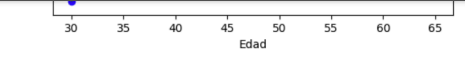


+ Código + Texto Se guardaron todos los cambios

RAM  
Disco



[10]



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

[13]

```
def pressure_low(age):  
    a_low = -0.02158996 # Pendiente  
    b_low = 83.19465161 # Intersección  
  
    return a_low * age + b_low  
  
query_age = 76  
print("Presión arterial baja para la edad {}: {:.2f}".format(query_age, pressure_low(query_age)))
```

Presión arterial baja para la edad 76: 81.55

[14]

```
def pressure_high(age):  
    a_high = 0.357971 # Pendiente  
    b_high = 95.6508411 # Intersección  
  
    return a_high * age + b_high  
  
query_age = 76  
print("Presión arterial alta para la edad {}: {:.2f}".format(query_age, pressure_high(query_age)))
```

Presión arterial alta para la edad 76: 122.86

Conectado a del backend de Google Compute Engine en Python 3

