

#a) Revisa si el dataset tiene valores nulos mediante df.info(), en dado caso, aplica df.dropna (inplace = True).

```
import pandas as pd
import numpy as np

df =
pd.read_csv("https://raw.githubusercontent.com/Karlamch/karlaMCH/refs/heads/main/data.csv")
df.dropna(inplace = True)
df

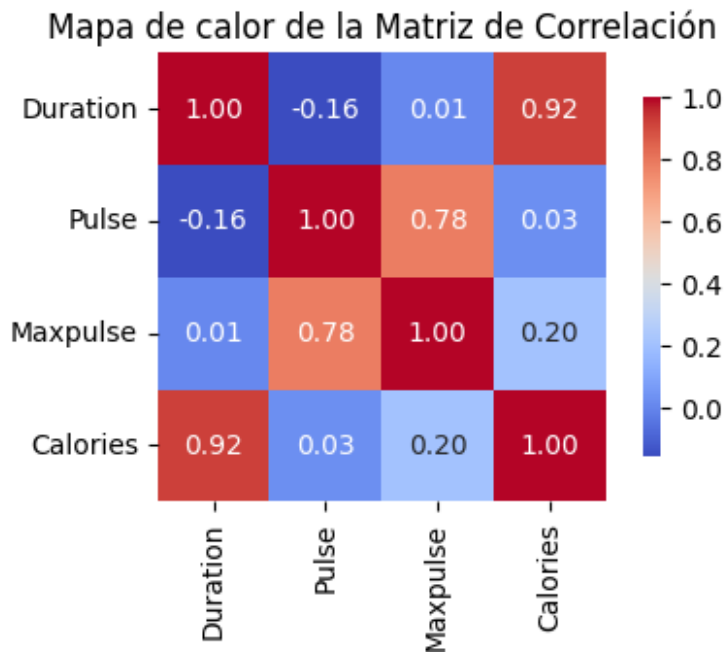
{"summary":{"\n  \"name\": \"df\",\n  \"rows\": 164,\n  \"fields\": [\n    {\n      \"column\": \"Duration\",\n      \"properties\": {\n        \"dtype\": \"number\",\n        \"std\": 42,\n        \"min\": 15,\n        \"max\": 300,\n        \"num_unique_values\": 16,\n        \"samples\": [\n          60,\n          45,\n          210\n        ],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n      },\n      \"column\": \"Pulse\",\n      \"properties\": {\n        \"dtype\": \"number\",\n        \"std\": 14,\n        \"min\": 80,\n        \"max\": 159,\n        \"num_unique_values\": 47,\n        \"samples\": [\n          159,\n          112,\n          153\n        ],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n      },\n      \"column\": \"Maxpulse\",\n      \"properties\": {\n        \"dtype\": \"number\",\n        \"std\": 16,\n        \"min\": 100,\n        \"max\": 184,\n        \"num_unique_values\": 57,\n        \"samples\": [\n          130,\n          127,\n          146\n        ],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n      },\n      \"column\": \"Calories\",\n      \"properties\": {\n        \"dtype\": \"number\",\n        \"std\": 266.37991924435164,\n        \"min\": 50.3,\n        \"max\": 1860.4,\n        \"num_unique_values\": 142,\n        \"samples\": [\n          328.0,\n          282.0,\n          229.4\n        ],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n      }\n    ]\n  },\n  \"type\": \"dataframe\",\n  \"variable_name\": \"df\"}
```

#b) Obtén un mapa de calor con la matriz de correlación. Determina las variabls independientes y variables dependientes y redacta una hipótesis

```
import seaborn as sns
import matplotlib.pyplot as plt

correlation_matrix = df.corr()
plt.figure(figsize=(5,3))
sns.heatmap(correlation_matrix, annot=True, cmap= "coolwarm",
fmt=".2f", square=True, cbar_kws={"shrink": .8})
plt.title("Mapa de calor de la Matriz de Correlación")
```

```
Text(0.5, 1.0, 'Mapa de calor de la Matriz de Correlación')
```



#c) Obtén el modelo que permite relacionar las variables independientes y la variable dependiente.

```
x= df [ ["Duration","Maxpulse"]]
y= df ["Calories"]

import statsmodels.api as sm
x_constante =sm.add_constant(x)
modelo = sm.OLS(y, x_constante).fit()

b0, b1, b2 =modelo.params

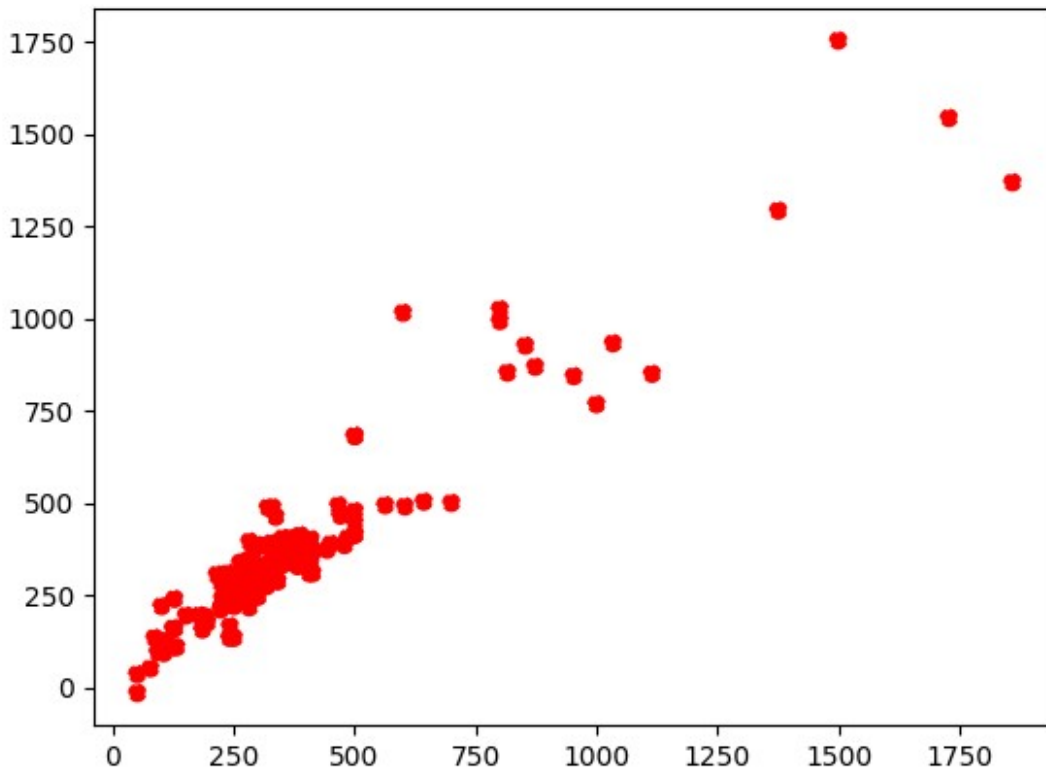
fun= lambda x1,x2: b0 + b1*x1 + b2*x2

Yc = modelo.predict(x_constante)

plt.scatter(y, Yc, color ="red", linestyle = "--")

from sklearn.metrics import r2_score #recomendada
r2 = r2_score(y, Yc)
print(f"Coeficiente de determinacion. {r2: 0.4}\n")

Coeficiente de determinacion. 0.8908
```



#d) Obtén el coeficiente de correlación e interpreta el resultado.

```
from sklearn.metrics import r2_score
r2= r2_score (y,Yc)
print(f'Coeficiente de correlacion:{r2: 0.4}/n ')
residuales = y - Yc
```

#e) Verifica los supuestos mediante un gráfico "y" vs "y_pred", el test de Shapiro-Wilk y el test de Breusch-Pagan. Interpreta los resultados.

```
from scipy.stats import shapiro
_, valor_p_sh = shapiro(residuales)
print(f'Valor p del test de Shapiro-Wilk: {valor_p_sh: 0.4f}')
from statsmodels.stats.api import het_breuschpagan
_, valor_p_bp, _, _ = het_breuschpagan(modelo.resid,
modelo.model.exog)
print(f'Valor p del test de Breusch-Pagan: {valor_p_bp: 0.4f}')
```

#Se rechaza Ho

*#f) Tomando en cuenta la información obtenida, redacta una conclusión.
#CONCLUSIÓN:El mapa de calor reveló una correlación moderada entre las variables y las calorías quemadas.*

#El modelo de regresión mostró un coeficiente de determinación de 0,8908/n, lo que indica el porcentaje de variabilidad de las calorías

*explicada por Duration y Maxpulse.
#Se rechaza Ho por que el valor es muy bajo*

Coeficiente de correlacion: 0.8908/n
Valor p del test de Shapiro-Wilk: 0.0000
Valor p del test de Breusch-Pagan: 0.0000