

```

import pandas as pd
df =
pd.read_csv('https://raw.githubusercontent.com/yesssss28/Estadistica/
refs/heads/main/mental_health.csv')
df
df.head(50)

{"summary":{"\n  \"name\": \"df\", \n  \"rows\": 6420, \n  \"fields\":
[\n    {\n      \"column\": \"Entity\", \n      \"properties\": {\n
\"dtype\": \"category\", \n      \"num_unique_values\": 214, \n
\"samples\": [\n        \"Argentina\", \n        \"Tuvalu\", \n
\"European Union (27)\", \n        ], \n      \"semantic_type\": \"\", \n
      \"description\": \"\", \n      }, \n    }, \n    {\n
\"column\": \"Code\", \n      \"properties\": {\n
\"dtype\":
\"category\", \n      \"num_unique_values\": 205, \n
\"samples\": [\n        \"BRB\", \n        \"AUS\", \n
\"LSO\", \n        ], \n      \"semantic_type\": \"\", \n
      \"description\": \"\", \n      }, \n    }, \n    {\n
\"column\":
\"Year\", \n      \"properties\": {\n
\"dtype\": \"number\", \n
\"std\": 8, \n      \"min\": 1990, \n      \"max\": 2019, \n
\"num_unique_values\": 30, \n      \"samples\": [\n        2017, \n
2005, \n        2013, \n        ], \n      \"semantic_type\": \"\", \n
      \"description\": \"\", \n      }, \n    }, \n    {\n
\"column\":
\"Schizophrenia disorders (share of population) - Sex: Both - Age:
Age-standardized\", \n      \"properties\": {\n
\"dtype\":
\"number\", \n      \"std\": 0.0393828304804131, \n      \"min\":
0.18841599, \n      \"max\": 0.4620453, \n
\"num_unique_values\": 6406, \n      \"samples\": [\n
0.24450433, \n        0.28247505, \n        0.2089815, \n        ], \n
      \"semantic_type\": \"\", \n      \"description\": \"\", \n
      }, \n    }, \n    {\n
\"column\":
\"Depressive disorders (share of population) - Sex: Both - Age: Age-standardized\", \n
      \"properties\": {\n
\"dtype\": \"number\", \n      \"std\":
0.9252858338210148, \n      \"min\": 1.5223331, \n      \"max\":
7.6458993, \n      \"num_unique_values\": 6416, \n      \"samples\":
[\n        3.9951913, \n        4.8294444, \n        4.957605, \n
        ], \n      \"semantic_type\": \"\", \n      \"description\": \"\", \n
      }, \n    }, \n    {\n
\"column\":
\"Anxiety disorders (share of population) - Sex: Both - Age: Age-standardized\", \n
      \"properties\": {\n
\"dtype\": \"number\", \n      \"std\":
1.0505430995124705, \n      \"min\": 1.8799964, \n      \"max\":
8.624634, \n      \"num_unique_values\": 6417, \n      \"samples\":
[\n        6.2759886, \n        3.7112138, \n        4.057681, \n
        ], \n      \"semantic_type\": \"\", \n      \"description\": \"\", \n
      }, \n    }, \n    {\n
\"column\":
\"Bipolar disorders (share of population) - Sex: Both - Age: Age-standardized\", \n
      \"properties\": {\n
\"dtype\": \"number\", \n      \"std\":
0.23339076361266692, \n      \"min\": 0.18166696, \n      \"max\":
1.5067295, \n      \"num_unique_values\": 6385, \n      \"samples\":
[\n        0.53775173, \n        0.70583045, \n        0.32984126, \n

```

```

n        ],\n        \n        \"semantic_type\": \"\", \n        \"description\": \"\", \n        }, \n        { \n        \"column\": \n        \"Eating disorders (share of population) - Sex: Both - Age: Age- \n        standardized\", \n        \"properties\": { \n        \"dtype\": \n        \"number\", \n        \"std\": 0.1383802210484159, \n        \"min\": \n        0.044780303, \n        \"max\": 1.0316882, \n        \"num_unique_values\": 6417, \n        \"samples\": [ \n        0.3780618, \n        0.073114336, \n        0.09078071 \n        ], \n        \"semantic_type\": \"\", \n        \"description\": \"\" \n        } \n        } \n        ] \n    }\", \"type\": \"dataframe\", \"variable_name\": \"df\"}

```

#. a) hipótesis de causalidad: Redacta una hipótesis sobre la causalidad entre las dos variables, a su vez, establece la variable dependiente y la variable independiente.

Hipotesis de casualidad: "Los trastornos depresivos han ido aumentando mientras que los trastornos alimentarios han bajado".

```

import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
df =
pd.read_csv('https://raw.githubusercontent.com/yesssss28/Estadistica/
refs/heads/main/mental_health.csv')
df
df.head(50)

```

eliminar registros con valores faltantes

```

df.dropna(inplace=True)
X = df['Eating disorders (share of population) - Sex: Both - Age: Age-
standardized'] # variable independiente
Y = df['Depressive disorders (share of population) - Sex: Both - Age:
Age-standardized'] # variable dependiente

```

#. b) recta de regresión ajustada: Realiza los cálculos necesarios para encontrar la recta de regresión ajustada. Incluye los procedimientos que te llevaron a la recta de regresión (Colab)

```

import matplotlib.pyplot as plt
plt.scatter(X, Y, color = 'pink')
plt.xlabel('Depressive disorders (share of population) - Sex: Both -
Age: Age-standardized')
plt.ylabel('Eating disorders (share of population) - Sex: Both - Age:
Age-standardized')

```

```

ax = plt.gca()
ax.spines['top'].set_visible(False)
ax.spines['right'].set_visible(False)

```

#. c) grafica los puntos y la recta de regresión (Colab)

```

import statsmodels.api as sm
X = sm.add_constant(X)
model = sm.OLS(Y, X).fit()
b0, b1 = model.params
Fun = lambda X: b0 + b1*X
Yc = Fun(X)
plt.plot(X, Yc, color = 'pink', linestyle = '--')

```

#. d) Coeficiente de correlación y determinación: Realiza los cálculos necesarios para obtener la r de Pearson y el coeficiente de determinación. Incluye dichos cálculos (Colab).

```
from scipy.stats import pearsonr
# Use the original X values (before adding the constant) for
correlation calculation
X_original = df['Eating disorders (share of population) - Sex: Both -
Age: Age-standardized']
r,_ = pearsonr(X_original, Y)
print(f'coeficiente de correlacion:{r:0.4}/n')
print(f'coeficiente de determinacion{r ** 2: 0.4f}/n')
#. e) Interpreta los resultados del coeficiente de correlación y el
coeficiente de determinación.
# Coeficiente de correlacion: Si el coeficiente de correlación es
negativo y alto, esto respaldaría la hipótesis de que a medida que los
trastornos alimentarios disminuyen, los trastornos depresivos
aumentan.
# Coeficiente de determinacion: Si el coeficiente de determinación es
bajo, significaría que la disminución de los trastornos alimentarios
no es un buen predictor del aumento de los trastornos depresivos.
```

#. f) Calcula el intervalo de confianza del 95% para β_1 y β_0 ¿Qué dice el intervalo de confianza a la posibilidad de que β_1 sea igual a cero? (Colab)

```
nivel_de_confianza = 0.95
intervalo_de_confianza = model.conf_int(alpha = 1 -
nivel_de_confianza)
intervalo_de_confianza_b0 = intervalo_de_confianza.loc['const']
intervalo_de_confianza_b1 = intervalo_de_confianza.loc['Eating
disorders (share of population) - Sex: Both - Age: Age-standardized']
print(f'intervalo de confianza para b0 {nivel_de_confianza:0.0%}')
print(f'intervalo de confianza para b1: {intervalo_de_confianza_b1[0]:
0.4f} <= b1 <= {intervalo_de_confianza_b1[1]: 0.4f}')
```

```
intervalo_de_confianza = model.conf_int(alpha = 1 - nivel_de_confianza)
intervalo_de_confianza_b0 = intervalo_de_confianza.loc['const']
intervalo_de_confianza_b1 = intervalo_de_confianza.loc['Eating
disorders (share of population) - Sex: Both - Age: Age-standardized']
print(f'intervalo de confianza para b0 {nivel_de_confianza:0.0%}')
print(f'intervalo de confianza para b1: {intervalo_de_confianza_b1[0]:
0.4f} <= b1 <= {intervalo_de_confianza_b1[1]: 0.4f}')
```

#. la variable independiente (trastornos alimenticios) podría no tener un efecto estadísticamente significativo en la variable dependiente

#. g) Realiza el gráfico de los residuales: A partir de este gráfico, menciona si los datos cumplen con los supuestos para la regresión: linealidad, normalidad en torno a la recta, homoscedasticidad (hay

```

más, pero con estas nos bastan) (Colab)
residuales = model.resid
plt.figure()
# Use X_original instead of X for the scatter plot
plt.scatter(X_original, residuales, color='yellow')
plt.xlabel('Depressive disorders (share of population) - Sex: Both - Age: Age-standardized')
plt.ylabel('Residuales')
plt.title('Grafica de residuales')
ax = plt.gca()
ax.spines['top'].set_visible(False)
ax.spines['right'].set_visible(False)
plt.axhline(y=0, color='black', linestyle='--')

from scipy.stats import shapiro
_, valor_p_sh = shapiro(residuales)
print(f'valor p de shapiro: {valor_p_sh: 0.4f}')

from statsmodels.stats.diagnostic import het_breuschpagan
_, valor_p_bp, _, _ = het_breuschpagan(model.resid, X)
print(f'valor p de breuschpagan: {valor_p_bp: 0.4f}')
#. Linealidad: el gráfico de dispersión de los residuales muestra un patrón aleatorio alrededor de la línea horizontal, el supuesto de linealidad se cumple.
#. Normalidad: Los residuos no siguen una distribución normal, lo que sugiere que el supuesto de normalidad no se cumple.
#. Homoscedasticidad: Los residuos muestran heterocedasticidad, lo que indica que el supuesto de homoscedasticidad no se cumple.

coeficiente de correlacion:-0.0461/n
coeficiente de determinacion 0.0021/n
intervalo de confianza para b0 95%
intervalo de confianza para b1: -0.4803 <= b1 <= -0.1427
intervalo de confianza para b0 95%
intervalo de confianza para b1: -0.4803 <= b1 <= -0.1427
valor p de shapiro: 0.0000
valor p de breuschpagan: 0.0000

/usr/local/lib/python3.11/dist-packages/scipy/stats/_axis_nan_policy.py:531: UserWarning: scipy.stats.shapiro: For N > 5000, computed p-value may not be accurate. Current N is 6150.
  res = hypotest_fun_out(*samples, **kws)

```



