```
import pandas as pd
df =
pd.read_csv('https://raw.githubusercontent.com/yessss28/Estadistica/
refs/heads/main/iris_synthetic_data.csv')
df.head()
```


```
{"summary":"{\n  \"name\": \"df\",\n  \"rows\": 3000,\n  \"fields\":
[\n    {\n      \"column\": \"sepal length\",\n      \"properties\":
{\n        \"dtype\": \"number\",\n        \"std\":
0.8050732740240121,\n        \"min\": 4.3,\n        \"max\": 7.9,\n
\"num_unique_values\": 37,\n        \"samples\": [\n          6.3,\n
5.6,\n          5.1\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"sepal width\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 0.4124722414970018,\n        \"min\":
2.0,\n        \"max\": 4.4,\n        \"num_unique_values\": 25,\n
\"samples\": [\n          3.2,\n          4.3,\n          3.8\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"petal length\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
1.7511826325162174,\n        \"min\": 0.9,\n        \"max\": 6.9,\n
\"num_unique_values\": 29,\n        \"samples\": [\n          6.9,\n
3.1,\n          4.1\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"petal width\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 0.7580218988039548,\n        \"min\":
0.1,\n        \"max\": 2.5,\n        \"num_unique_values\": 22,\n
\"samples\": [\n          0.3,\n          1.6,\n          1.4\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"label\",\n      \"properties\":
{\n        \"dtype\": \"category\",\n        \"num_unique_values\":
3,\n        \"samples\": [\n          \"Iris-setosa\",\n
\"Iris-versicolor\",\n          \"Iris-virginica\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    }\n  ]\n}","type":"dataframe","variable_name":"df"}
```


```
import pandas as pd
import numpy as np
df = pd.read_csv
('https://raw.githubusercontent.com/yessss28/Estadistica/refs/heads/
main/iris_synthetic_data.csv')
# eliminar registros cpn valores faltantes
df.dropna(inplace=True)
# 1. Establesca una variable dependiente ( Y ) y una variable
independiente ( X ).
X = df['petal length']
```

```
Y = df['petal width']

# 2. Realice un diagrama de dispersión para estos datos.
import matplotlib.pyplot as plt
plt.figure()
plt.scatter(X, Y, color = 'pink')
plt.xlabel('petal length')
plt.ylabel('petal width')
ax = plt.gca()
ax.spines['top'].set_visible(False)
ax.spines['right'].set_visible(False)

# 3. ¿Los datos soportan la suposición de linealidad?
# Sí

# 4. Calcule el coeficiente de correlación e interprete el resultado.
from scipy.stats import pearsonr
r, _ = pearsonr(X, Y)
print(f'Coeficiente de correlación: {r: 0.4f}\n')

# 5. Calcule el coeficiente de determinación e interprete el
resultado.
print(f'Coeficiente de determinación: {r ** 2: 0.4f}\n')


# 6. Obtenga la recta de regresión ajustada y grafíquelo sobre el
gráfico de
# dispersión.
import statsmodels.api as sm
x_constante = sm.add_constant(X)
modelo = sm.OLS(Y, x_constante).fit()

b0, b1 = modelo.params

fun = lambda x: b0 + b1 * x

Yc = fun(X)

plt.plot(X, Yc, color = 'black', linestyle = '--')

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

# 7. Obtenga un intervalo de confianza del 95% para la pendiente de la
recta de
# regresión ajustada ( b1 )
nivel_de_confianza = 0.95
intervalo_de_confianza = modelo.conf_int(alpha = 1 -
nivel_de_confianza)
```

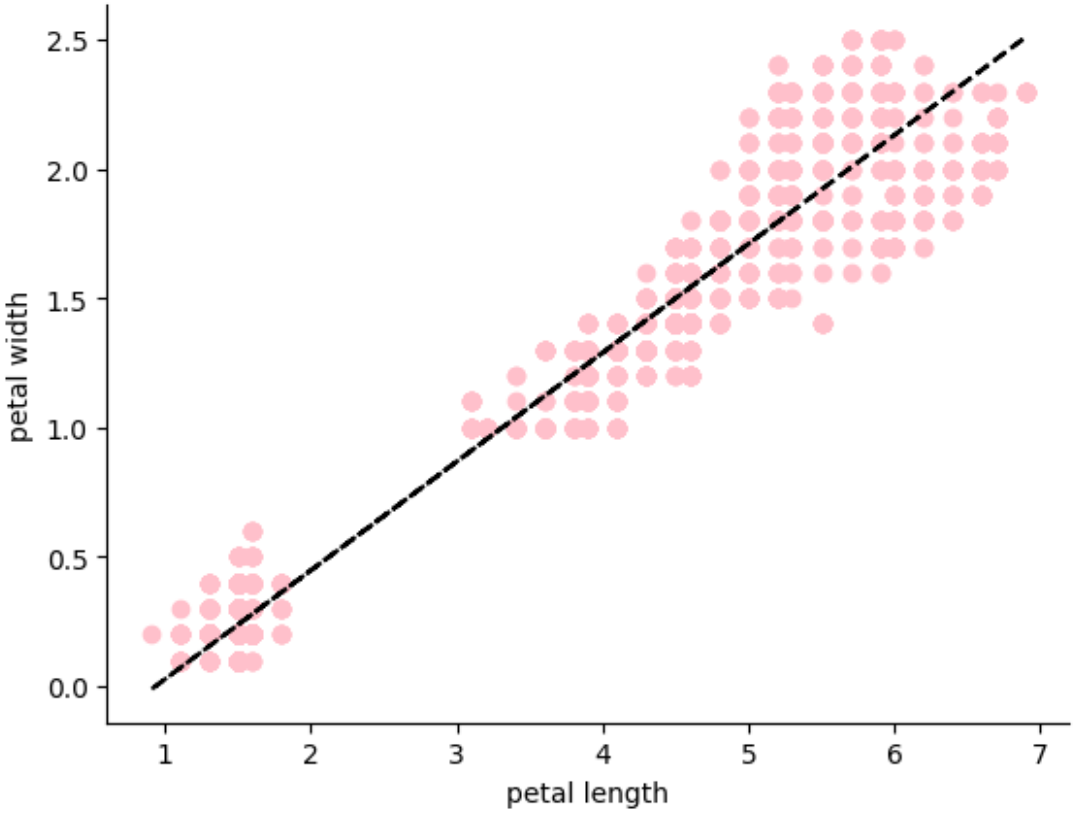
```
intervalo_de_confianza_b1 = intervalo_de_confianza.iloc[1]
print(f'Intervalo de confianza para b1 de {nivel_de_confianza: 0.0%}')
print(f'{intervalo_de_confianza_b1[0]: 0.4f} < b1 <
{intervalo_de_confianza_b1[1]: 0.4f}\n')


# 8. Calcule los residuales y trace un nuevo gráfico de dispersión.
Comente,
# ¿Parece que se verifican los supuestos?
residuales = modelo.resid
plt.figure()
plt.scatter(X, residuales, color = 'pink')
plt.xlabel('petal length')
plt.ylabel('petal width')
ax = plt.gca()
ax.spines['top'].set_visible(False)
ax.spines['right'].set_visible(False)
plt.axhline(y = 0, color = 'gray', linestyle = '--')

# 9. Realice la prueba de Shapiro para los residuales y comente el
resultado.
from scipy.stats import shapiro
_, valor_p_sh = shapiro(residuales)
print(f'valor-p de Shapiro: {valor_p_sh: 0.4f}\n')


# 10. Realice la prueba de Brausch-Pagan para los residuales y comente
el
# resultado.
from statsmodels.stats.api import het_breuschpagan
_, valor_p_bp, _, _ = het_breuschpagan(residuales, x_constante)
print(f'valor_p de Breusch-Pagan: {valor_p_bp: 0.4f}\n')


# 11. Tres estudiantes sacaron 70, 75 y 84 de calificación. Según la
recta de
# regresión ajustada, ¿cuáles son los resultados esperados para estos
tres alumnos?
print(f'para x = 70, y = {fun(70): 0.0f}')
print(f'para x = 75, y = {fun(75): 0.0f}')
print(f'para x = 84, y = {fun(84): 0.0f}\n')


# 12. Realice una tabla ANOVA e interprete el resultado.
from statsmodels.formula.api import ols
nuevo = df[['petal length', 'petal width']]
nuevo.columns = ['petal_length', 'petal_width']
# Y ~ X
modelo_2 = ols('petal_length ~ petal_width', data = nuevo).fit()
```

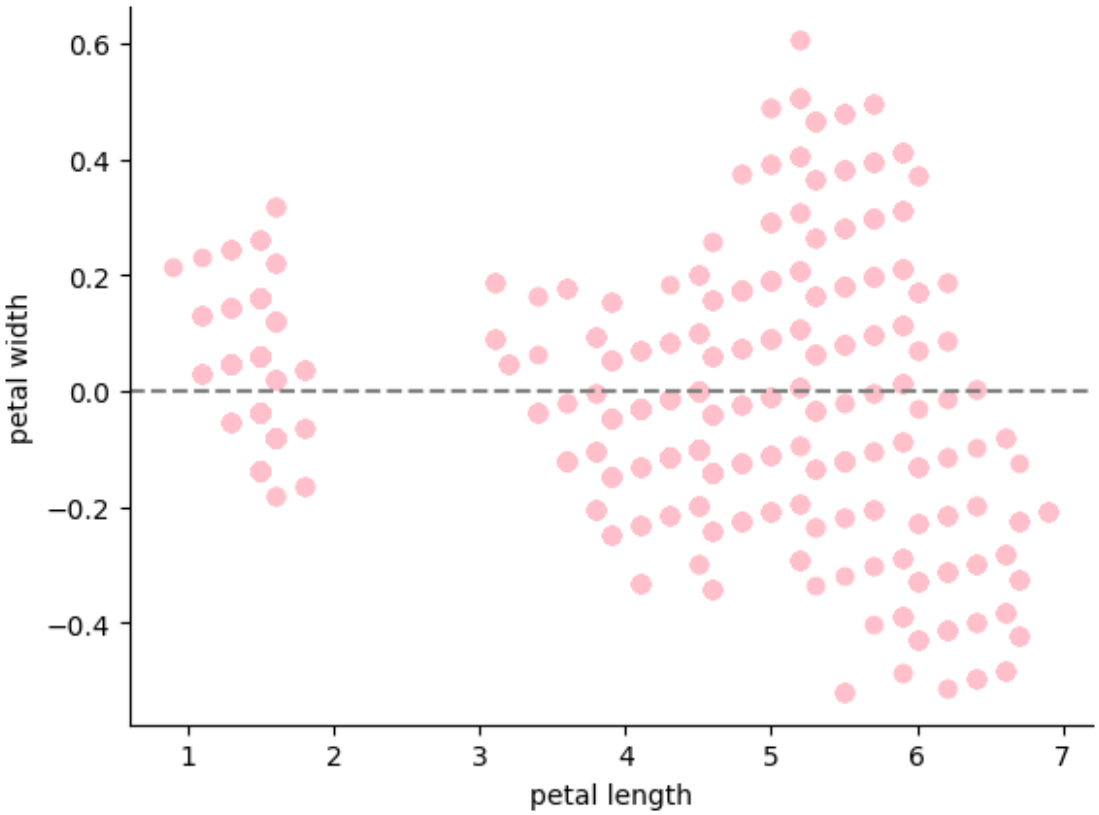
```
tabla_anova = sm.stats.anova_lm(modelo_2)
tabla_anova
```

```
Coeficiente de correlación:  0.9708

Coeficiente de determinación:  0.9424

Coeficiente de determinación:  0.9424

Intervalo de confianza para b1 de  95%
 0.4165 < b1 <  0.4239

valor-p de Shapiro:  0.0000

valor_p de Breusch-Pagan:  0.0000

para x = 70, y =  29
para x = 75, y =  31
para x = 84, y =  35
```


```
{"summary":"{\n  \"name\": \"tabla_anova\",\n  \"rows\": 2,\n
\"fields\": [\n    {\n      \"column\": \"df\",\n      \"properties\":
{\n        \"dtype\": \"number\",\n        \"std\":
2119.199023216083,\n        \"min\": 1.0,\n        \"max\": 2998.0,\n
\"num_unique_values\": 2,\n        \"samples\": [\n          2998.0,\n
1.0\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"sum_sq\",\n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": 5754.022962603994,\n        \"min\": 529.7189423729762,\n
\"max\": 8667.13625429376,\n        \"num_unique_values\": 2,\n
\"samples\": [\n          529.7189423729762,\n
8667.13625429376\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"mean_sq\",\n      \"properties\": {\n        \"dtype\": \"number\",\
n        \"std\": 6128.46587963397,\n        \"min\":
0.17669077464075258,\n        \"max\": 8667.13625429376,\n
\"num_unique_values\": 2,\n        \"samples\": [\n
0.17669077464075258,\n          8667.13625429376\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"F\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": null,\n        \"min\":
49052.568092000096,\n        \"max\": 49052.568092000096,\n
\"num_unique_values\": 1,\n        \"samples\": [\n
49052.568092000096\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"PR(>F)\",\n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": null,\n        \"min\": 0.0,\n        \"max\": 0.0,\n
\"num_unique_values\": 1,\n        \"samples\": [\n          0.0\n
```

```
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    }\n  ]\n}","type":"dataframe","variable_name":"tabla_anova"}
```


0.6
0.4
0.2
0.0
−0.2
−0.4
petal width
1
2
3
4
5
6
7
petal length