

xkc4a37uw

February 3, 2025

##Objetivo - Predecir los mejores medios en los que gasta dinero en publicidad que maximicen las ventas; se utilizan la Regresión Lineal Simple y Múltiple.

```
[149]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
import plotly.express as px
import plotly.graph_objects as go
```

```
[150]: URL = 'https://raw.githubusercontent.com/Adrian-Cancino/DataScience/refs/heads/
↳main/Data/Advertising.csv'
```

```
[151]: datos = pd.read_csv(URL)
datos
```

```
[151]:
```

	TV	Radio	Newspaper	Sales
0	230.1	37.8	69.2	22.1
1	44.5	39.3	45.1	10.4
2	17.2	45.9	69.3	9.3
3	151.5	41.3	58.5	18.5
4	180.8	10.8	58.4	12.9
..
195	38.2	3.7	13.8	7.6
196	94.2	4.9	8.1	9.7
197	177.0	9.3	6.4	12.8
198	283.6	42.0	66.2	25.5
199	232.1	8.6	8.7	13.4

[200 rows x 4 columns]

```
[152]: dict_columnas = {
    'TV': 'TV',
    'Radio': 'Radio',
    'Newspaper': 'Periódico',
    'Sales': 'Ventas'
```

```
}

datos.rename(columns=dict_columnas, inplace=True)
datos
```

```
[152]:
```

	TV	Radio	Periódico	Ventas
0	230.1	37.8	69.2	22.1
1	44.5	39.3	45.1	10.4
2	17.2	45.9	69.3	9.3
3	151.5	41.3	58.5	18.5
4	180.8	10.8	58.4	12.9
..
195	38.2	3.7	13.8	7.6
196	94.2	4.9	8.1	9.7
197	177.0	9.3	6.4	12.8
198	283.6	42.0	66.2	25.5
199	232.1	8.6	8.7	13.4

[200 rows x 4 columns]

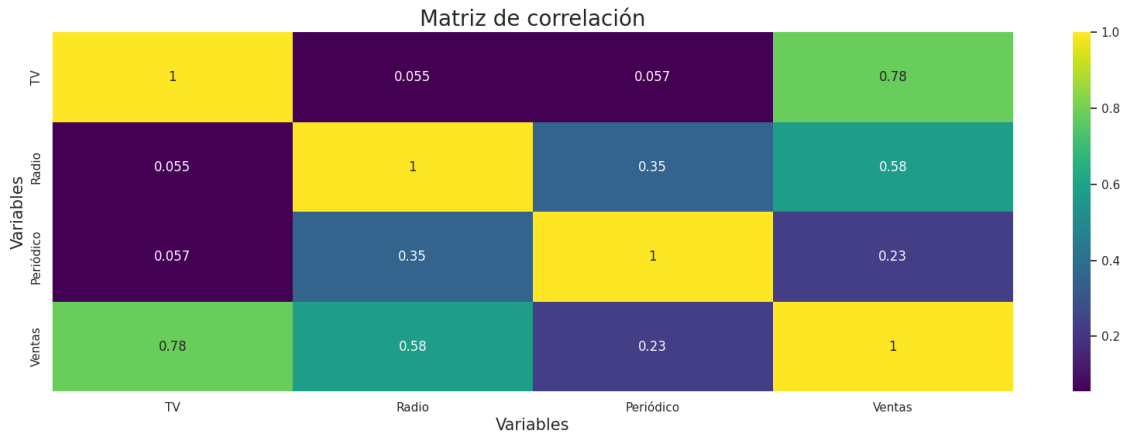
```
[153]: matrix_corr = datos.corr()
matrix_corr
```

```
[153]:
```

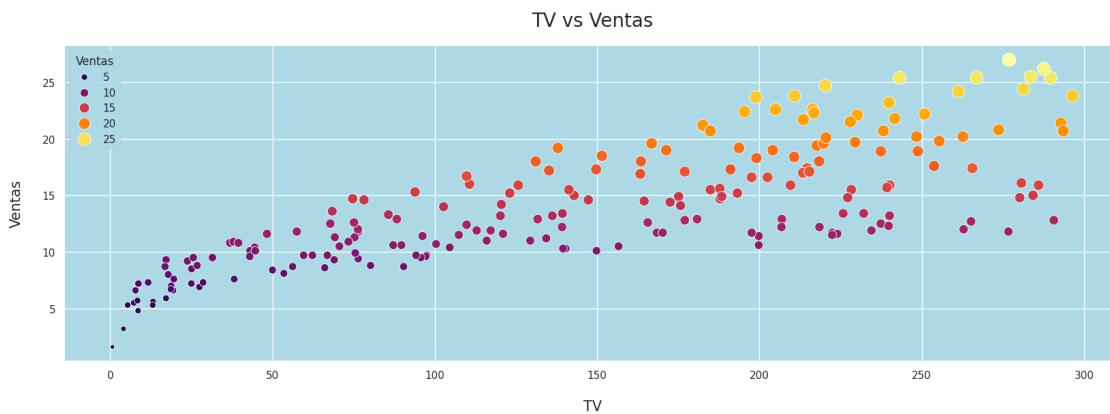
	TV	Radio	Periódico	Ventas
TV	1.000000	0.054809	0.056648	0.782224
Radio	0.054809	1.000000	0.354104	0.576223
Periódico	0.056648	0.354104	1.000000	0.228299
Ventas	0.782224	0.576223	0.228299	1.000000

```
[154]: fig = px.imshow(matrix_corr, text_auto=True, labels=dict(x='Variables',
↪y='Variables', color='Correlación'))
fig.update_layout(title='Matriz de correlación', height=800, width=800)
fig.show()
```

```
[155]: sns.set_style(style="darkgrid")
sns.set_theme(rc={'axes.facecolor':'lightblue', 'figure.facecolor':'white'})
plt.figure(figsize=(20, 6))
sns.heatmap(data=matrix_corr, annot=True, cmap='viridis')
plt.title('Matriz de correlación', fontsize=20)
plt.xlabel('Variables', fontsize=15)
plt.ylabel('Variables', fontsize=15)
plt.show()
```



```
[156]: sns.set_style(style="darkgrid")
sns.set_theme(rc={'axes.facecolor':'lightblue', 'figure.facecolor':'white'})
plt.figure(figsize=(20, 6))
sns.scatterplot(data=datos, x='TV', y='Ventas', palette='inferno',
                hue='Ventas', size='Ventas', sizes=(20, 200))
plt.title('TV vs Ventas', fontsize=20, pad=20)
plt.xlabel('TV', fontsize=15, labelpad=20)
plt.ylabel('Ventas', fontsize=15, labelpad=20)
plt.show()
```



```
[157]: fig = px.scatter(datos, x='TV', y='Ventas',
                        color='Ventas', size='Ventas',
                        color_continuous_scale='inferno',
                        size_max=20)
fig.update_layout(
    title='TV vs Ventas',
    title_font=dict(size=20),
```

```

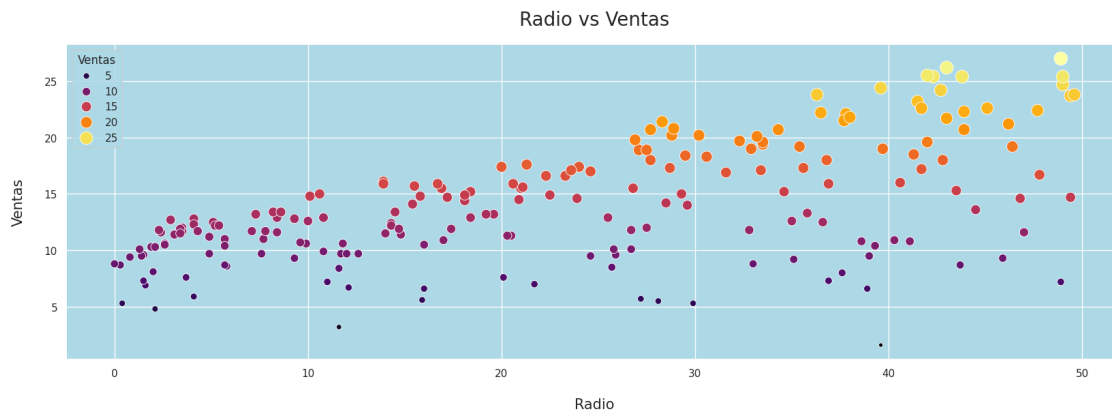
    xaxis_title='TV',
    yaxis_title='Ventas',
    plot_bgcolor='lightblue',
    paper_bgcolor='white',
    width=1400, height=500
)
fig.show()

```

```

[158]: sns.set_style(style="darkgrid")
sns.set_theme(rc={'axes.facecolor':'lightblue', 'figure.facecolor':'white'})
plt.figure(figsize=(20, 6))
sns.scatterplot(data=datos, x='Radio', y='Ventas', palette='inferno',
    hue='Ventas', size='Ventas', sizes=(20, 200))
plt.title('Radio vs Ventas', fontsize=20, pad=20)
plt.xlabel('Radio', fontsize=15, labelpad=20)
plt.ylabel('Ventas', fontsize=15, labelpad=20)
plt.show()

```

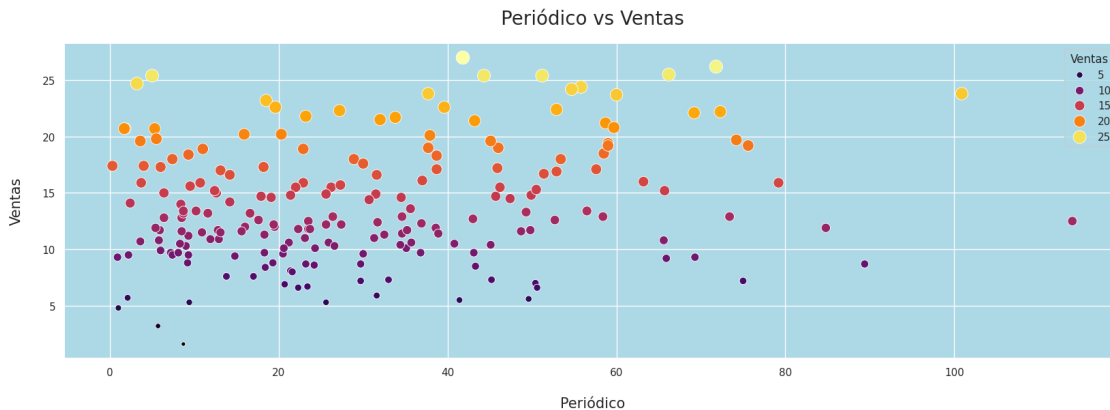


```

[159]: fig = px.scatter(datos, x='Radio', y='Ventas',
    color='Ventas', size='Ventas',
    color_continuous_scale='inferno',
    size_max=20)
fig.update_layout(
    title='Radio vs Ventas',
    title_font=dict(size=20),
    xaxis_title='Radio',
    yaxis_title='Ventas',
    plot_bgcolor='lightblue',
    paper_bgcolor='white',
    width=1400, height=500
)
fig.show()

```

```
[160]: sns.set_style(style="darkgrid")
sns.set_theme(rc={'axes.facecolor':'lightblue', 'figure.facecolor':'white'})
plt.figure(figsize=(20, 6))
sns.scatterplot(data=datos, x='Periódico', y='Ventas', palette='inferno',
               hue='Ventas', size='Ventas', sizes=(20, 200))
plt.title('Periódico vs Ventas', fontsize=20, pad=20)
plt.xlabel('Periódico', fontsize=15, labelpad=20)
plt.ylabel('Ventas', fontsize=15, labelpad=20)
plt.show()
```



```
[161]: fig = px.scatter(datos, x='Periódico', y='Ventas',
                        color='Ventas', size='Ventas',
                        color_continuous_scale='inferno',
                        size_max=20)
fig.update_layout(
    title='Periódico vs Ventas',
    title_font=dict(size=20),
    xaxis_title='Periódico',
    yaxis_title='Ventas',
    plot_bgcolor='lightblue',
    paper_bgcolor='white',
    width=1400, height=500
)
fig.show()
```

#Regresión simple - Modelos

```
[162]: from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
```

```
[163]: datos_tv = datos['TV']
datos_tv = datos_tv.values.reshape(-1, 1)
datos_tv
```

```
[163]: array([[230.1],
              [ 44.5],
              [ 17.2],
              [151.5],
              [180.8],
              [  8.7],
              [ 57.5],
              [120.2],
              [  8.6],
              [199.8],
              [ 66.1],
              [214.7],
              [ 23.8],
              [ 97.5],
              [204.1],
              [195.4],
              [ 67.8],
              [281.4],
              [ 69.2],
              [147.3],
              [218.4],
              [237.4],
              [ 13.2],
              [228.3],
              [ 62.3],
              [262.9],
              [142.9],
              [240.1],
              [248.8],
              [ 70.6],
              [292.9],
              [112.9],
              [ 97.2],
              [265.6],
              [ 95.7],
              [290.7],
              [266.9],
              [ 74.7],
              [ 43.1],
              [228. ],
              [202.5],
              [177. ],
              [293.6],
```

[206.9],
[25.1],
[175.1],
[89.7],
[239.9],
[227.2],
[66.9],
[199.8],
[100.4],
[216.4],
[182.6],
[262.7],
[198.9],
[7.3],
[136.2],
[210.8],
[210.7],
[53.5],
[261.3],
[239.3],
[102.7],
[131.1],
[69.],
[31.5],
[139.3],
[237.4],
[216.8],
[199.1],
[109.8],
[26.8],
[129.4],
[213.4],
[16.9],
[27.5],
[120.5],
[5.4],
[116.],
[76.4],
[239.8],
[75.3],
[68.4],
[213.5],
[193.2],
[76.3],
[110.7],
[88.3],
[109.8],

[134.3],
[28.6],
[217.7],
[250.9],
[107.4],
[163.3],
[197.6],
[184.9],
[289.7],
[135.2],
[222.4],
[296.4],
[280.2],
[187.9],
[238.2],
[137.9],
[25.],
[90.4],
[13.1],
[255.4],
[225.8],
[241.7],
[175.7],
[209.6],
[78.2],
[75.1],
[139.2],
[76.4],
[125.7],
[19.4],
[141.3],
[18.8],
[224.],
[123.1],
[229.5],
[87.2],
[7.8],
[80.2],
[220.3],
[59.6],
[0.7],
[265.2],
[8.4],
[219.8],
[36.9],
[48.3],
[25.6],

[273.7],
[43.],
[184.9],
[73.4],
[193.7],
[220.5],
[104.6],
[96.2],
[140.3],
[240.1],
[243.2],
[38.],
[44.7],
[280.7],
[121.],
[197.6],
[171.3],
[187.8],
[4.1],
[93.9],
[149.8],
[11.7],
[131.7],
[172.5],
[85.7],
[188.4],
[163.5],
[117.2],
[234.5],
[17.9],
[206.8],
[215.4],
[284.3],
[50.],
[164.5],
[19.6],
[168.4],
[222.4],
[276.9],
[248.4],
[170.2],
[276.7],
[165.6],
[156.6],
[218.5],
[56.2],
[287.6],

```
[253.8],  
[205. ],  
[139.5],  
[191.1],  
[286. ],  
[ 18.7],  
[ 39.5],  
[ 75.5],  
[ 17.2],  
[166.8],  
[149.7],  
[ 38.2],  
[ 94.2],  
[177. ],  
[283.6],  
[232.1]])
```

```
[164]: y = datos['Ventas'].values
```

```
[165]: X_train, X_test, y_train, y_test = train_test_split(datos_tv, y, test_size=0.3,  
↳ random_state=42)
```

```
[166]: reg = LinearRegression()  
reg.fit(X_train, y_train)
```

```
[166]: LinearRegression()
```

```
[167]: y_pred = reg.predict(X_test)  
print(f'Predicciones: {y_pred[:4]}, Valores reales: {y_test[:4]}')
```

```
Predicciones: [14.81785392 16.30754437 20.83230507  7.78243112], Valores reales:  
[16.9 22.4 21.4  7.3]
```

```
[168]: r_square = reg.score(X_test, y_test)  
print(f'Coeficiente de determinación: {r_square}')
```

```
rmse = np.sqrt(mean_squared_error(y_test, y_pred))  
print(f'Raíz del error cuadrático medio: {rmse}')
```

```
Coeficiente de determinación: 0.6714477229302764  
Raíz del error cuadrático medio: 2.9951613049072354
```

```
[169]: datos['Ventas'].min()
```

```
[169]: 1.6
```

```
[170]: datos['Ventas'].max()
```

[170]: 27.0

```
[171]: ventas_comp = pd.DataFrame({
        'TV': X_test.flatten(),
        'Ventas_reales': y_test.flatten(),
        'Ventas_predicción': y_pred.flatten()
    })
ventas_comp.head()
```

```
[171]:      TV  Ventas_reales  Ventas_predicción
0  163.3           16.9         14.817854
1  195.4           22.4         16.307544
2  292.9           21.4         20.832305
3   11.7            7.3          7.782431
4  220.3           24.7         17.463099
```

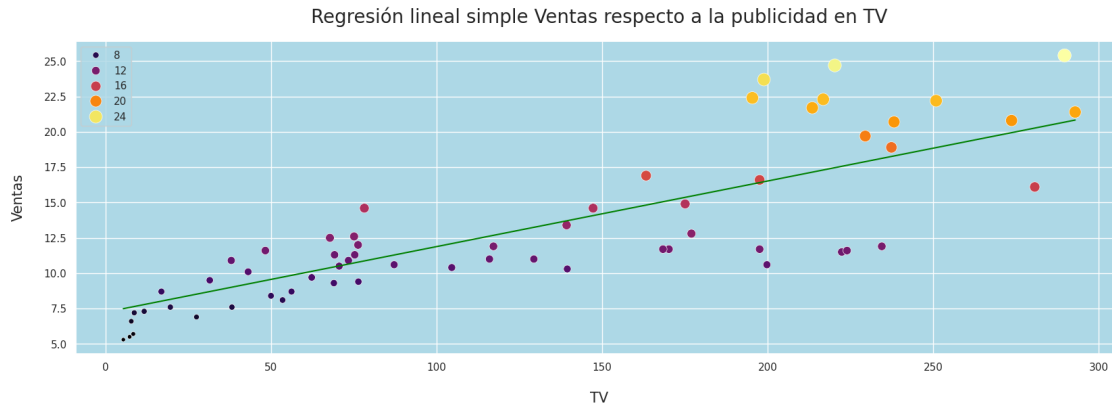
##Graficando modelo junto a datos de conjunto de testing.

```
[172]: from plotly.subplots import make_subplots
```

```
[173]: fig = px.scatter(datos, x='Periódico', y='Ventas',
                    color='Ventas', size='Ventas',
                    color_continuous_scale='inferno',
                    size_max=20)
```

```
[174]: fig = px.scatter(datos, x='Periódico', y='Ventas',
                    color='Ventas', size='Ventas',
                    color_continuous_scale='inferno',
                    size_max=20)
```

```
[175]: sns.set_style(style="darkgrid")
sns.set_theme(rc={'axes.facecolor':'lightblue', 'figure.facecolor':'white'})
plt.figure(figsize=(20, 6))
sns.scatterplot(data=ventas_comp, x='TV', y='Ventas_reales', palette='inferno',
    ↪hue='Ventas_reales', size='Ventas_reales', sizes=(20, 200))
sns.lineplot(data=ventas_comp, x='TV', y='Ventas_predicción', color='green')
plt.title('Regresión lineal simple Ventas respecto a la publicidad en TV',
    ↪fontsize=20, pad=20)
plt.xlabel('TV', fontsize=15, labelpad=20)
plt.ylabel('Ventas', fontsize=15, labelpad=20)
plt.show()
```



```
[176]: fig = make_subplots(rows=1, cols=1)
fig.add_trace(go.Scatter(x=X_train.flatten(), y=y_train, mode='markers',
    ↪name='Conjunto de entrenamiento', marker=dict(color=y_train, size=y_train)),
    ↪row=1, col=1)
fig.add_trace(go.Scatter(x=X_test.flatten(), y=reg.predict(X_test),
    ↪mode='lines', name='Regresión lineal', line=dict(color='green')), row=1,
    ↪col=1)
fig.update_layout(height=500, width=1500, title_text='Regresión lineal simple
    ↪Ventas respecto a la publicidad en TV')
fig.show()
```

```
[177]: dato_radio = datos['Radio']
dato_radio = dato_radio.values.reshape(-1, 1)
```

```
[178]: y = datos['Ventas'].values

X_train, X_test, y_train, y_test = train_test_split(dato_radio, y, test_size=0.
    ↪3, random_state=42)

reg = LinearRegression()
reg.fit(X_train, y_train)

y_pred = reg.predict(X_test)
print(f'Predicciones: {y_pred[:4]}, Valores reales: {y_test[:4]}')
```

Predicciones: [16.26772205 19.79791711 15.5441417 17.42983596], Valores reales: [16.9 22.4 21.4 7.3]

```
[179]: r_square = reg.score(X_test, y_test)
print(f'Coeficiente de determinación: {r_square}')
```

```
rmse = np.sqrt(mean_squared_error(y_test, y_pred))
```

```
print(f'Raíz del error cuadrático medio: {rmse}')
```

Coeficiente de determinación: 0.13116693684196756

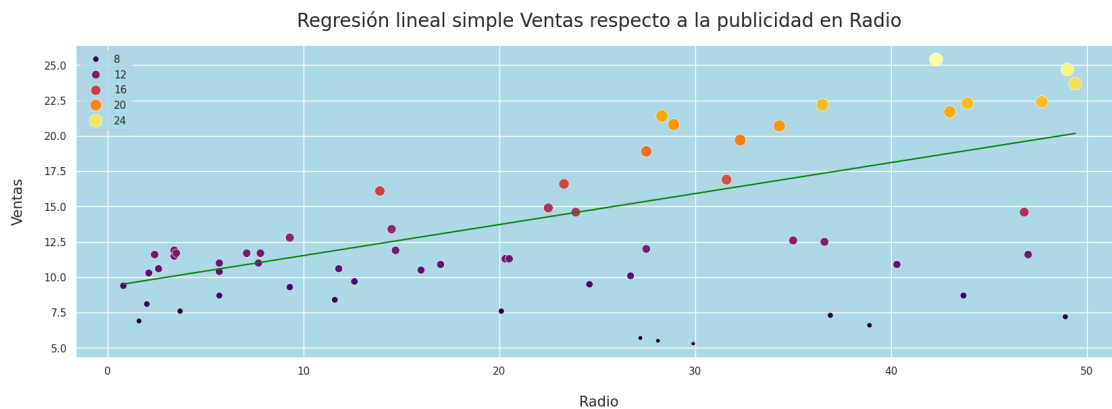
Raíz del error cuadrático medio: 4.870641281862208

```
[180]: ventas_comp = pd.DataFrame({
        'Radio': X_test.flatten(),
        'Ventas_reales': y_test.flatten(),
        'Ventas_predicción': y_pred.flatten()
    })
ventas_comp.head()
```

```
[180]:
```

	Radio	Ventas_reales	Ventas_predicción
0	31.6	16.9	16.267722
1	47.7	22.4	19.797917
2	28.3	21.4	15.544142
3	36.9	7.3	17.429836
4	49.0	24.7	20.082964

```
[181]: sns.set_style(style="darkgrid")
sns.set_theme(rc={'axes.facecolor': 'lightblue', 'figure.facecolor': 'white'})
plt.figure(figsize=(20, 6))
sns.scatterplot(data=ventas_comp, x='Radio', y='Ventas_reales',
                palette='inferno', hue='Ventas_reales', size='Ventas_reales', sizes=(20,
                200))
sns.lineplot(data=ventas_comp, x='Radio', y='Ventas_predicción', color='green')
plt.title('Regresión lineal simple Ventas respecto a la publicidad en Radio',
          fontsize=20, pad=20)
plt.xlabel('Radio', fontsize=15, labelpad=20)
plt.ylabel('Ventas', fontsize=15, labelpad=20)
plt.show()
```



```
[182]: fig = make_subplots(rows=1, cols=1)
fig.add_trace(go.Scatter(x=X_train.flatten(), y=y_train, mode='markers',
    ↪name='Conjunto de entrenamiento', marker=dict(color=y_train, size=y_train)),
    ↪row=1, col=1)
fig.add_trace(go.Scatter(x=X_test.flatten(), y=reg.predict(X_test),
    ↪mode='lines', name='Regresión lineal', line=dict(color='green')), row=1,
    ↪col=1)
fig.update_layout(height=500, width=1500, title_text='Regresión lineal simple
    ↪Ventas respecto a la publicidad en Radio')
fig.show()
```

```
[183]: datos_periodico = datos['Periódico']
datos_periodico = datos_periodico.values.reshape(-1, 1)
```

```
[184]: y = datos['Ventas'].values

X_train, X_test, y_train, y_test = train_test_split(datos_periodico, y,
    ↪test_size=0.3, random_state=42)

reg = LinearRegression()
reg.fit(X_train, y_train)

y_pred = reg.predict(X_test)
print(f'Predicciones: {y_pred[:4]}, Valores reales: {y_test[:4]}')
```

Predicciones: [15.86524548 15.86524548 15.27418406 15.39605239], Valores reales: [16.9 22.4 21.4 7.3]

```
[185]: r_square = reg.score(X_test, y_test)
print(f'Coeficiente de determinación: {r_square}')

rmse = np.sqrt(mean_squared_error(y_test, y_pred))
print(f'Raíz del error cuadrático medio: {rmse}')
```

Coeficiente de determinación: -0.057301173788661064
Raíz del error cuadrático medio: 5.373006089766974

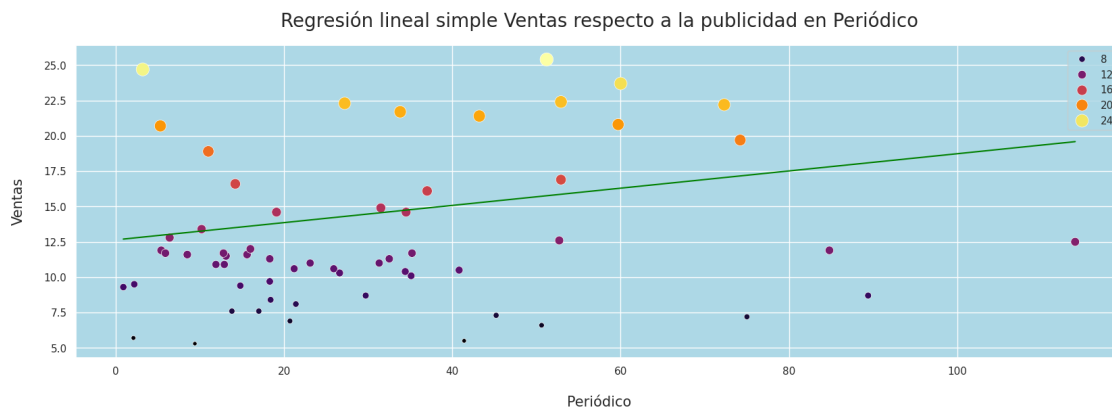
```
[186]: ventas_comp = pd.DataFrame({
    'Periódico': X_test.flatten(),
    'Ventas_reales': y_test.flatten(),
    'Ventas_predicción': y_pred.flatten()
})
ventas_comp.head()
```

```
[186]:
```

	Periódico	Ventas_reales	Ventas_predicción
0	52.9	16.9	15.865245
1	52.9	22.4	15.865245

2	43.2	21.4	15.274184
3	45.2	7.3	15.396052
4	3.2	24.7	12.836817

```
[187]: sns.set_style(style="darkgrid")
sns.set_theme(rc={'axes.facecolor':'lightblue', 'figure.facecolor':'white'})
plt.figure(figsize=(20, 6))
sns.scatterplot(data=ventas_comp, x='Periódico', y='Ventas_reales',
                palette='inferno', hue='Ventas_reales', size='Ventas_reales', sizes=(20,
                200))
sns.lineplot(data=ventas_comp, x='Periódico', y='Ventas_predicción',
             color='green')
plt.title('Regresión lineal simple Ventas respecto a la publicidad en
             Periódico', fontsize=20, pad=20)
plt.xlabel('Periódico', fontsize=15, labelpad=20)
plt.ylabel('Ventas', fontsize=15, labelpad=20)
plt.show()
```



```
[188]: fig = make_subplots(rows=1, cols=1)
fig.add_trace(go.Scatter(x=X_train.flatten(), y=y_train, mode='markers',
                        name='Conjunto de entrenamiento', marker=dict(color=y_train, size=y_train)),
            row=1, col=1)
fig.add_trace(go.Scatter(x=X_test.flatten(), y=reg.predict(X_test),
                        mode='lines', name='Regresión lineal', line=dict(color='green')), row=1,
            col=1)
fig.update_layout(height=500, width=1500, title_text='Regresión lineal simple
            Ventas respecto a la publicidad en Periódico')
fig.show()
```

Modelos de regresión múltiple

```
[189]: X = datos.drop(['Radio', 'Ventas'], axis=1).values
y = datos['Ventas'].values

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,
↳random_state=42)

reg = LinearRegression()
reg.fit(X_train, y_train)

y_pred = reg.predict(X_test)
print(f'Predicciones: {y_pred[:4]}, Valores reales: {y_test[:4]}')
```

Predicciones: [16.04429782 17.51962128 21.46954703 8.65505487], Valores reales:
[16.9 22.4 21.4 7.3]

```
[190]: r_square = reg.score(X_test, y_test)
print(f'Coeficiente de determinación: {r_square}')
```

```
rmse = np.sqrt(mean_squared_error(y_test, y_pred))
print(f'Raíz del error cuadrático medio: {rmse}')
```

Coeficiente de determinación: 0.668935545006016
Raíz del error cuadrático medio: 3.0065903074245406

TV-Radio

```
[191]: X = datos.drop(['Periódico', 'Ventas'], axis=1).values
y = datos['Ventas'].values

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,
↳random_state=42)

reg = LinearRegression()
reg.fit(X_train, y_train)

y_pred = reg.predict(X_test)
print(f'Predicciones: {y_pred[:4]}, Valores reales: {y_test[:4]}')
```

Predicciones: [16.43778963 21.11462413 21.48161733 10.82947123], Valores reales:
[16.9 22.4 21.4 7.3]

```
[192]: r_square = reg.score(X_test, y_test)
print(f'Coeficiente de determinación: {r_square}')
```

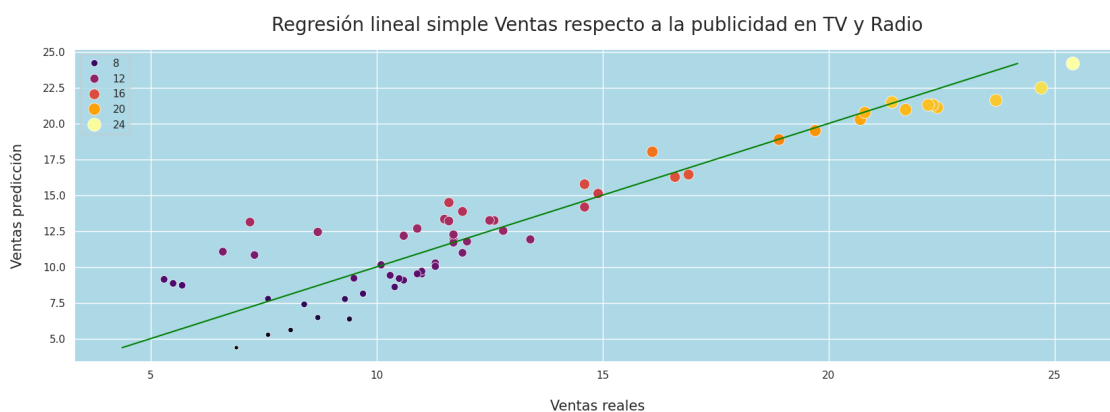
```
rmse = np.sqrt(mean_squared_error(y_test, y_pred))
print(f'Raíz del error cuadrático medio: {rmse}')
```

Coeficiente de determinación: 0.8656253548947075
Raíz del error cuadrático medio: 1.9154756731764253


```
[193]: ventas_comp = pd.DataFrame({
    'TV': X_test[:, 0].flatten(),
    'Radio': X_test[:, 1].flatten(),
    'Ventas_reales': y_test.flatten(),
    'Ventas_predicción': y_pred.flatten()
})
ventas_comp.head()
```

```
[193]:      TV  Radio  Ventas_reales  Ventas_predicción
0  163.3   31.6           16.9          16.437790
1  195.4   47.7           22.4          21.114624
2  292.9   28.3           21.4          21.481617
3   11.7   36.9            7.3          10.829471
4  220.3   49.0           24.7          22.475538
```

```
[194]: sns.set_style(style="darkgrid")
sns.set_theme(rc={'axes.facecolor': 'lightblue', 'figure.facecolor': 'white'})
plt.figure(figsize=(20, 6))
sns.scatterplot(data=ventas_comp, x=ventas_comp['Ventas_reales'],
    ↪y=ventas_comp['Ventas_predicción'], palette='inferno',
    ↪hue=ventas_comp['Ventas_predicción'], size=ventas_comp['Ventas_predicción'],
    ↪sizes=(20, 200))
sns.lineplot(data=ventas_comp, x=ventas_comp['Ventas_predicción'],
    ↪y=ventas_comp['Ventas_predicción'], color='green')
plt.title('Regresión lineal simple Ventas respecto a la publicidad en TV y
    ↪Radio', fontsize=20, pad=20)
plt.xlabel('Ventas reales', fontsize=15, labelpad=20)
plt.ylabel('Ventas predicción', fontsize=15, labelpad=20)
plt.show()
```



```
[195]: fig = make_subplots(rows=1, cols=1)
```

```
fig.add_trace(go.Scatter(x=ventas_comp['Ventas_reales'],  
    ↪y=ventas_comp['Ventas_predicción'], mode='markers', name='Conjunto de  
    ↪entrenamiento', marker=dict(color=ventas_comp['Ventas_predicción'],  
    ↪size=ventas_comp['Ventas_predicción'])), row=1, col=1)  
fig.add_trace(go.Scatter(x=ventas_comp['Ventas_predicción'], y=reg.  
    ↪predict(X_test), mode='lines', name='Regresión lineal',  
    ↪line=dict(color='green')), row=1, col=1)  
fig.update_layout(height=500, width=1500, title_text='Regresión lineal simple  
    ↪Ventas respecto a la publicidad en TV y Radio')  
fig.show()
```

1 Informe de Análisis de Publicidad

1.1 Introducción

El objetivo de este análisis fue determinar cuáles son los medios publicitarios más efectivos para maximizar las ventas. Para ello, utilicé modelos de regresión lineal simple y múltiple, analizando el impacto de la inversión en TV, radio y periódicos sobre las ventas.

1.2 Análisis Exploratorio de Datos

Para este estudio, trabajé con un conjunto de datos que contiene información sobre la inversión en publicidad en diferentes medios y las ventas obtenidas. Apliqué técnicas de exploración de datos para comprender mejor las relaciones entre las variables:

- Cálculo de la matriz de correlación para identificar la relación entre las variables.
- Visualización de dispersión entre la inversión en cada medio publicitario y las ventas.
- Mapas de calor para ilustrar la fuerza de las correlaciones.

Los resultados indicaron que la inversión en TV tiene la correlación más alta con las ventas, seguida de la radio, mientras que la inversión en periódico tiene una relación más débil.

1.3 Modelado Predictivo

Apliqué modelos de regresión lineal para cuantificar la relación entre la inversión publicitaria y las ventas:

1. Regresión Lineal Simple:

- Se entrenó un modelo con cada variable independiente por separado.
- Se observó que la TV tiene el coeficiente más alto, lo que indica que su impacto en las ventas es significativo.

2. Regresión Lineal Múltiple:

- Se entrenó un modelo considerando todas las variables.
- Se obtuvo un modelo con mejor capacidad predictiva que los modelos simples.

1.4 Conclusiones

Los hallazgos de este análisis confirman que la inversión en TV es el factor más determinante para aumentar las ventas. La radio también juega un papel relevante, mientras que la publicidad en periódicos tiene menor impacto.

Este análisis proporciona información clave para optimizar la asignación de presupuestos en publicidad, maximizando el retorno de inversión.

Simple-Poderoso.