MODELO REGRESION LINEAL SIMPLE - LABORATORIO (JAUREGUI ROMERO EDUARDO) ¶

- X=INVERSION EN PUBLICIDAD DE RADIO
- Y=TOTAL DE VENTAS

IMPORTANDO LIBRERIAS

In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [3]:

```
datos = pd.read_csv('Advertising.csv',index_col=0)
```

In [32]:

```
datos.head()
```

Out[32]:

	TV	Radio	Newspaper	Sales
1	230.1	37.8	69.2	22.1
2	44.5	39.3	45.1	10.4
3	17.2	45.9	69.3	9.3
4	151.5	41.3	58.5	18.5
5	180.8	10.8	58.4	12.9

In [35]:

```
datos.tail()
```

Out[35]:

	TV	Radio	Newspaper	Sales
196	38.2	3.7	13.8	7.6
197	94.2	4.9	8.1	9.7
198	177.0	9.3	6.4	12.8
199	283.6	42.0	66.2	25.5
200	232.1	8.6	8.7	13.4

In [36]:

```
datos.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 200 entries, 1 to 200
Data columns (total 4 columns):
 #   Column     Non-Null Count  Dtype  
---  ------     --------------  -----  
 0   TV         200 non-null    float64
 1   Radio      200 non-null    float64
 2   Newspaper  200 non-null    float64
 3   Sales      200 non-null    float64
dtypes: float64(4)
memory usage: 7.8 KB
```

In [37]:

```
datos.describe()
```

Out[37]:

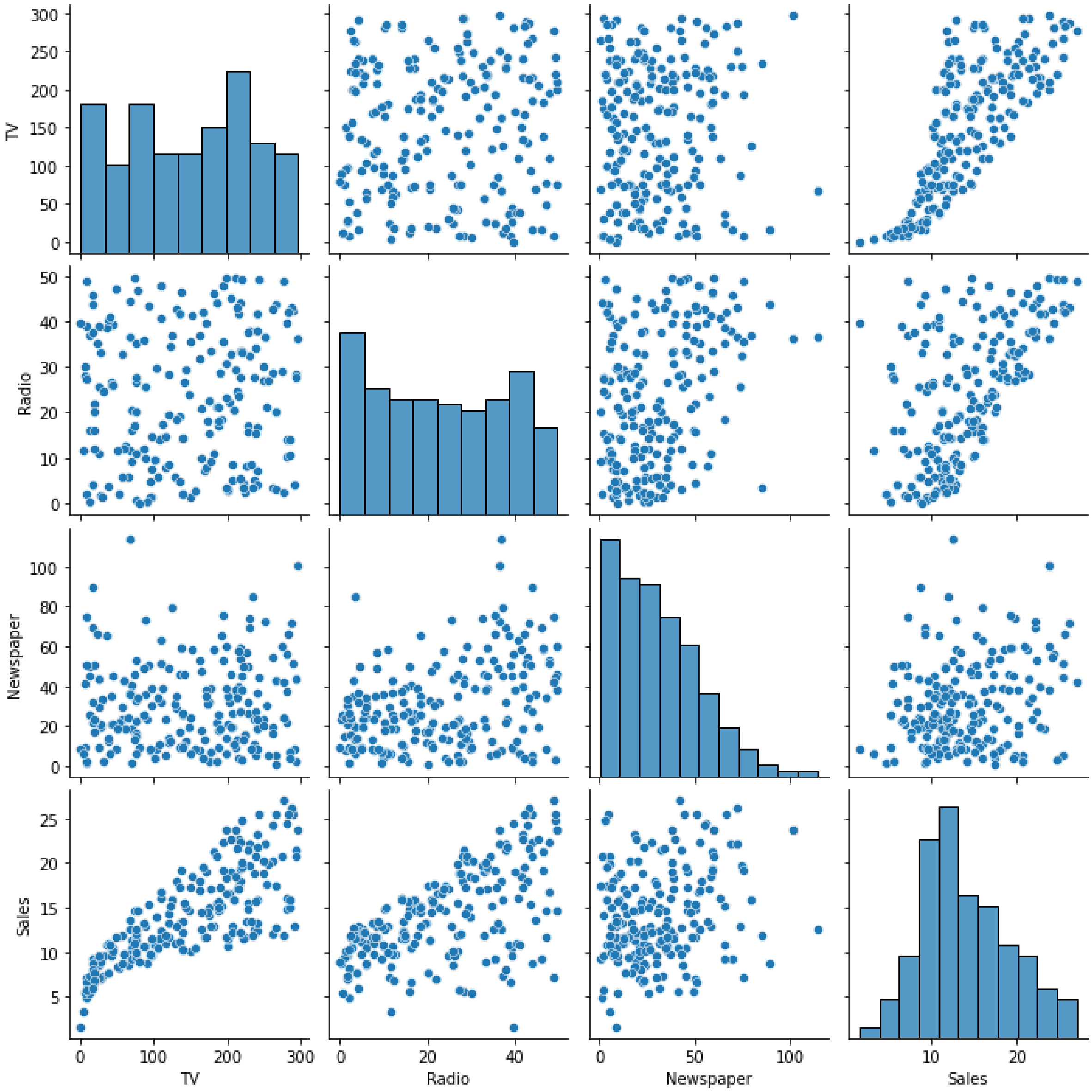
	TV	Radio	Newspaper	Sales
count	200.000000	200.000000	200.000000	200.000000
mean	147.042500	23.264000	30.554000	14.022500
std	85.854236	14.846809	21.778621	5.217457
min	0.700000	0.000000	0.300000	1.600000
25%	74.375000	9.975000	12.750000	10.375000
50%	149.750000	22.900000	25.750000	12.900000
75%	218.825000	36.525000	45.100000	17.400000
max	296.400000	49.600000	114.000000	27.000000

In [38]:

```
sns.pairplot(datos)
```

Out[38]:

```
<seaborn.axisgrid.PairGrid at 0x7f968b9204c0>
```

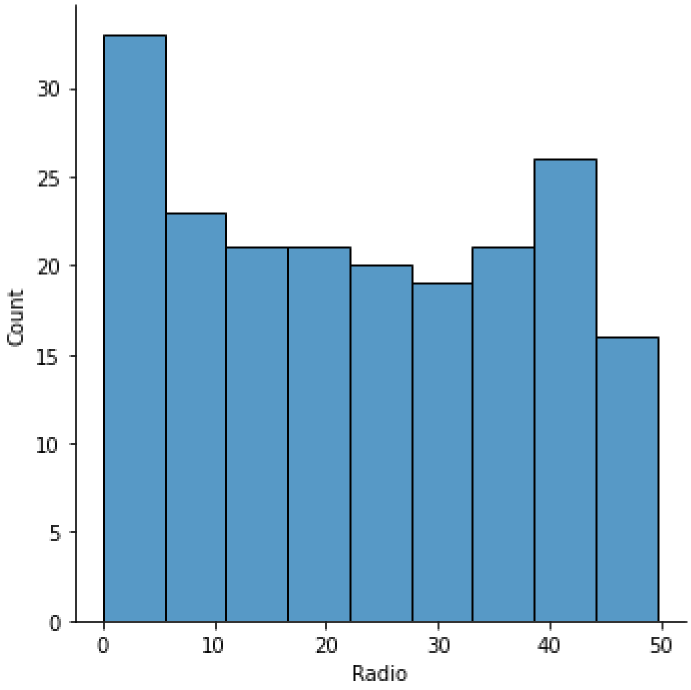


In [40]:

```
sns.displot(datos['Radio'])
```

Out[40]:

```
<seaborn.axisgrid.FacetGrid at 0x7f9698c76c70>
```

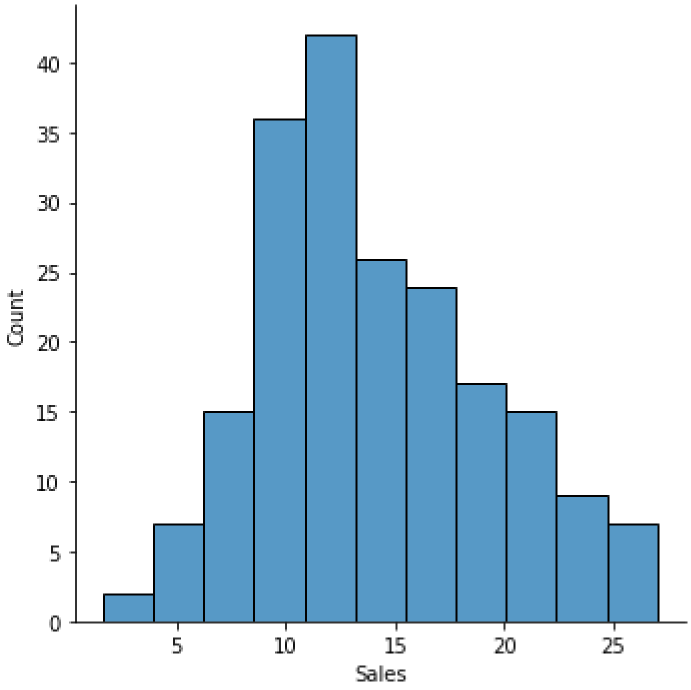


In [43]:

```
sns.displot(datos['Sales'])
```

Out[43]:

```
<seaborn.axisgrid.FacetGrid at 0x7f9698f9e640>
```



In [44]:

```
datos.corr()
```

Out[44]:

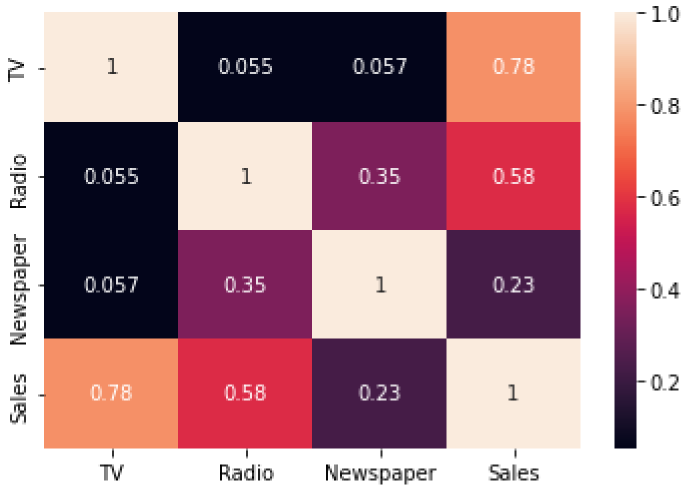
	TV	Radio	Newspaper	Sales
TV	1.000000	0.054809	0.056648	0.782224
Radio	0.054809	1.000000	0.354104	0.576223
Newspaper	0.056648	0.354104	1.000000	0.228299
Sales	0.782224	0.576223	0.228299	1.000000

In [45]:

```
sns.heatmap(datos.corr(),annot=True)
```

Out[45]:

```
<AxesSubplot:>
```



MODELO DE REGRESION LINEAL x=radio y=ventas

In [46]:

```
X=datos[["Radio"]]
```

In [8]:

```
y=datos["Sales"]
```

APLICACION DE MINIMOS CUADRADOS

In [9]:

```
from sklearn.linear_model import LinearRegression
```

In [10]:

```
modelo = LinearRegression()
```

In [13]:

```
modelo.fit(X,y)
```

Out[13]:

```
LinearRegression()
```

PARAMETROS OBTENIDOS PARA EL MODELO

In [16]:

```
print(f'B0={modelo.intercept_}')
```

```
B0=9.311638095158283
```

In [34]:

```
print(f'B0={modelo.coef_[0]}')
```

```
B0=0.2024957833924397
```

In [49]:

```
print(f'y`={modelo.coef_[0]} + {modelo.intercept_} x')
```

```
y`=0.2024957833924397 + 9.311638095158283 x
```

PREDICCIONES

In [21]:

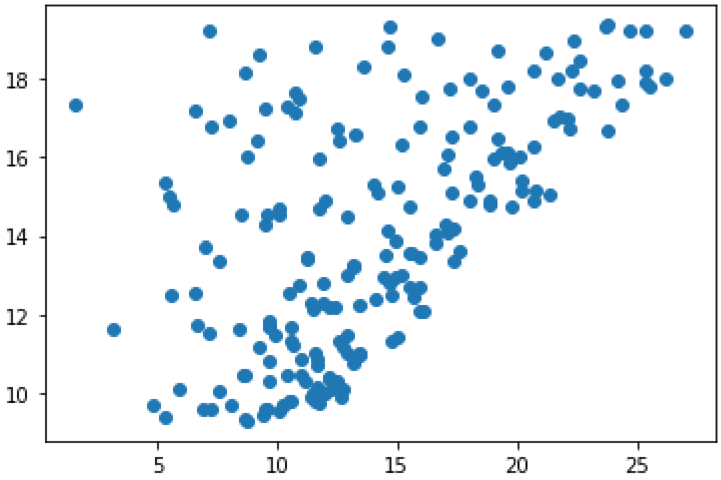
```
predicciones = modelo.predict(X)
```

In [23]:

```
plt.scatter(y,predicciones)
```

Out[23]:

```
<matplotlib.collections.PathCollection at 0x7f96a9114130>
```

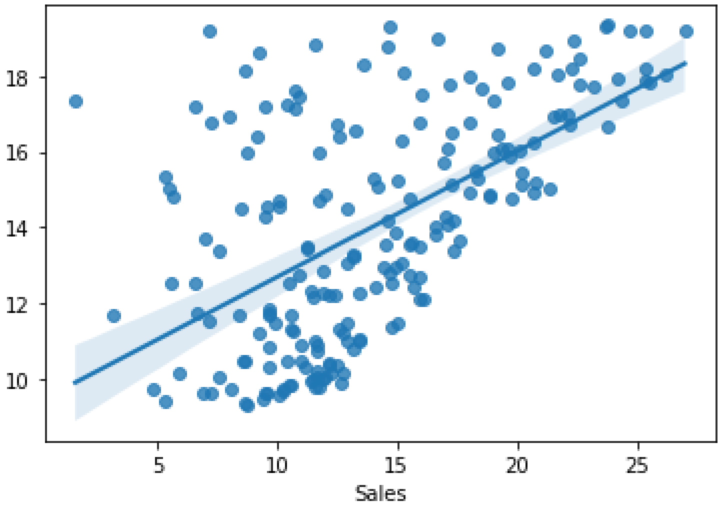


In [25]:

```
sns.regplot(x=y,y=predicciones,data=datos)
```

Out[25]:

```
<AxesSubplot:xlabel='Sales'>
```



METRICAS DE EVALUACION

In [27]:

```
from sklearn import metrics
```

In [54]:

```
RSS = print(f'RSS: {metrics.mean_squared_error(y,predicciones)*200}') #SUMA DE LOS E
```

```
RSS: 3618.4795490250876
```