SALES PREDICTION USING PYTHON

In [3]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import statsmodels.api as sm
from statsmodels.formula.api import ols
```

In [4]:
```
data = pd.read_csv("advertising.csv")
```

Out[4]:

	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	12.0
3	151.5	41.3	58.5	16.5
4	180.8	10.8	58.4	17.9
...	...	...	...	...
195	38.2	3.7	13.8	7.6
196	94.2	4.9	8.1	14.0
197	177.0	9.3	6.4	14.8
198	283.6	42.0	66.2	25.5
199	232.1	8.6	8.7	18.4

200 rows × 4 columns

In [5]:
```
data.head()
```

Out[5]:

	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	12.0
3	151.5	41.3	58.5	16.5
4	180.8	10.8	58.4	17.9

In [6]:
```
data.shape
```

Out[6]:
```
(200, 4)
```

In [9]:
```
data[['TV','Radio','Newspaper']].describe()
```

Out[9]:

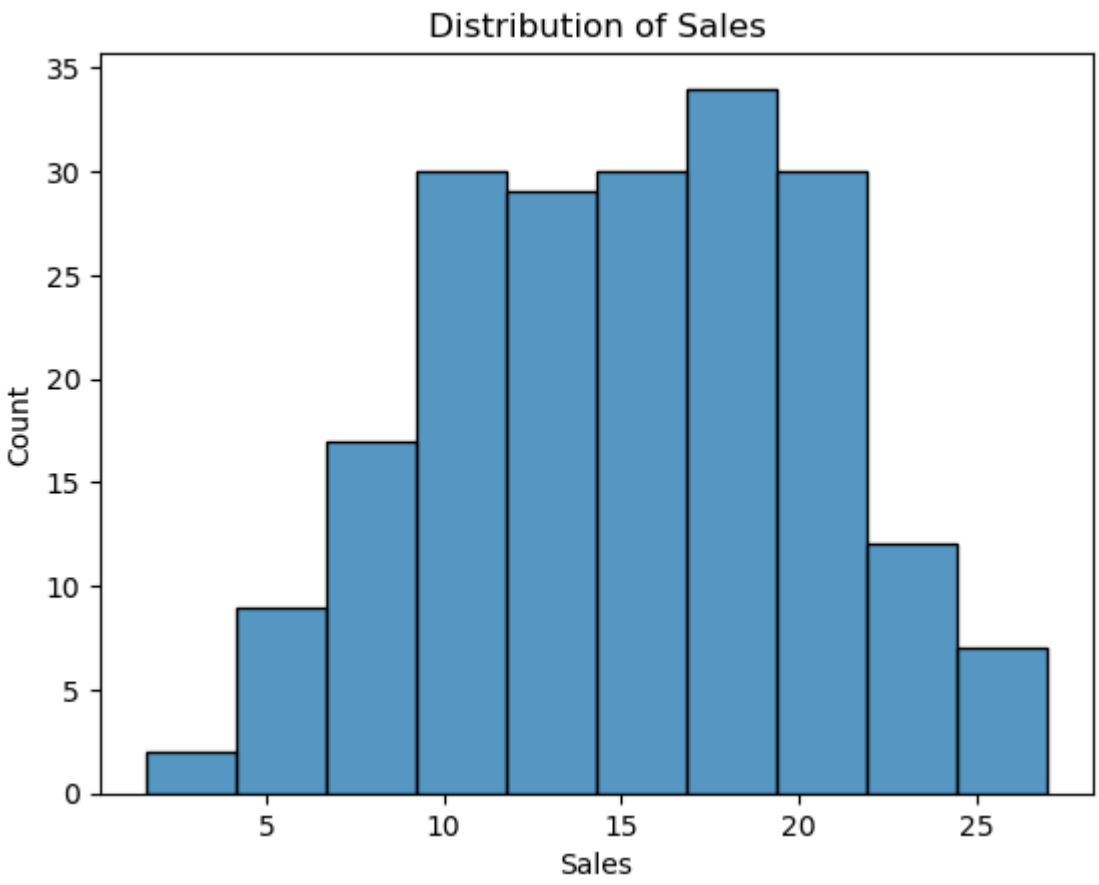
	TV	Radio	Newspaper
count	200.000000	200.000000	200.000000
mean	147.042500	23.264000	30.554000
std	85.854236	14.846809	21.778621
min	0.700000	0.000000	0.300000
25%	74.375000	9.975000	12.750000
50%	149.750000	22.900000	25.750000
75%	218.825000	36.525000	45.100000
max	296.400000	49.600000	114.000000

In [10]:
```
# Calculate the average missing rate in the sales column.
missing_sales = data.Sales.isna().mean()
missing_sales = round(missing_sales*100, 2)
```

```
Percentage of missing Sales: 0.0%
```

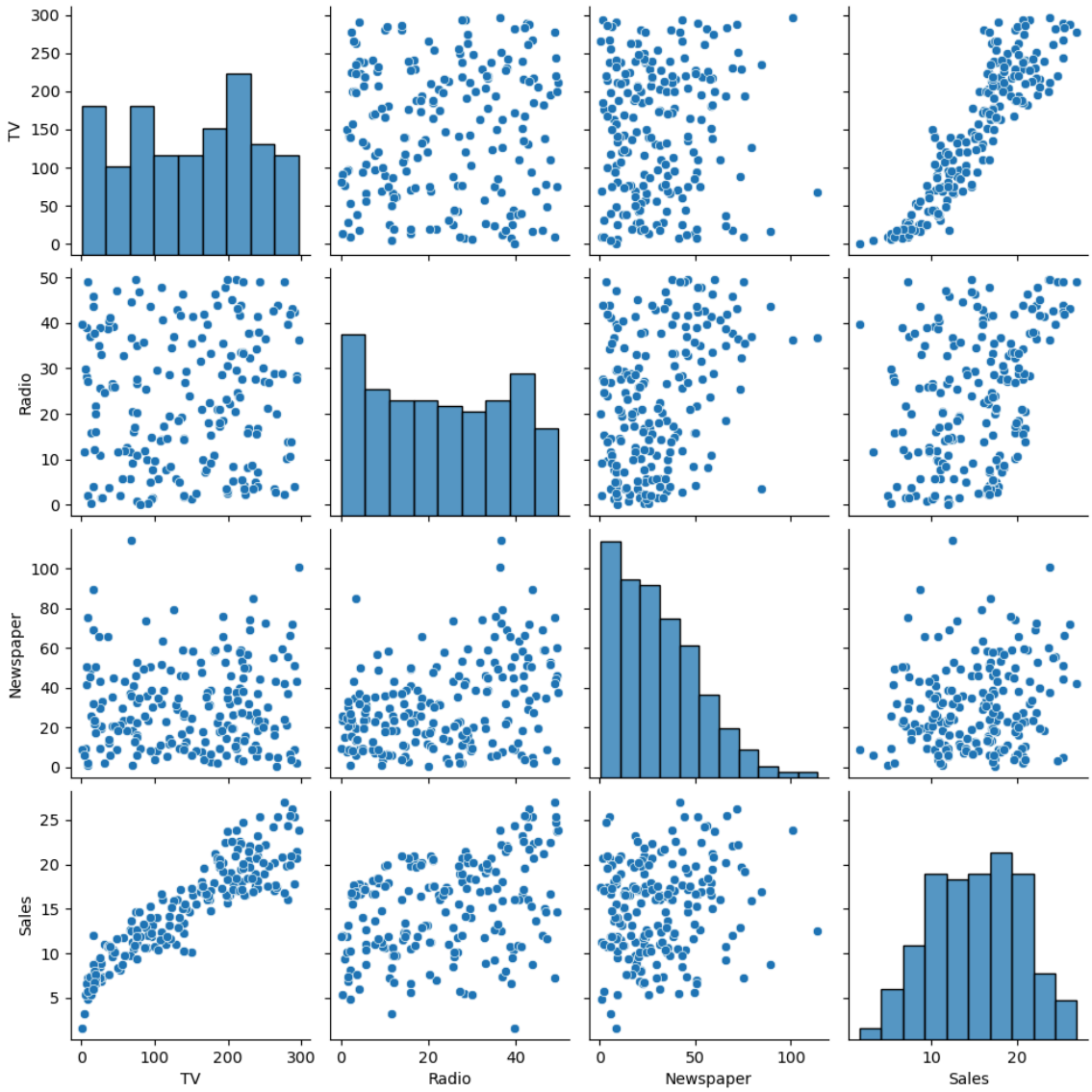
In [11]:
```
# from the above missing sales percentage we are sure that there is no miss
```

In [12]:
```
fig = sns.histplot(data['Sales'])
```



In [13]:
```
sns.pairplot(data)
```

```
C:\Users\HP\anaconda3\Lib\site-packages\seaborn\axisgrid.py:118: UserWarni
ng: The figure layout has changed to tight
  self._figure.tight_layout(*args, **kwargs)
```



#here I have selected TV AS AN (X) VARIABLE AS IT HAS STRONGEST RELATIONSHIP WITH SALES.

In [14]:
```
# Define the OLS formula.
f = 'Sales ~ TV'
# Create an OLS model.
OLS = ols(formula = f, data = data)
# Fit the model.
model = OLS.fit()
# Save the results summary.
model_results = model.summary()
```

Out[14]:

OLS Regression Results

Dep. Variable:	Sales	**R-squared:**	0.812
Model:	OLS	**Adj. R-squared:**	0.811
Method:	Least Squares	**F-statistic:**	856.2
Date:	Mon, 07 Oct 2024	**Prob (F-statistic):**	7.93e-74
Time:	21:23:12	**Log-Likelihood:**	-448.99
No. Observations:	200	**AIC:**	902.0
Df Residuals:	198	**BIC:**	908.6
Df Model:	1		
Covariance Type:	nonrobust		

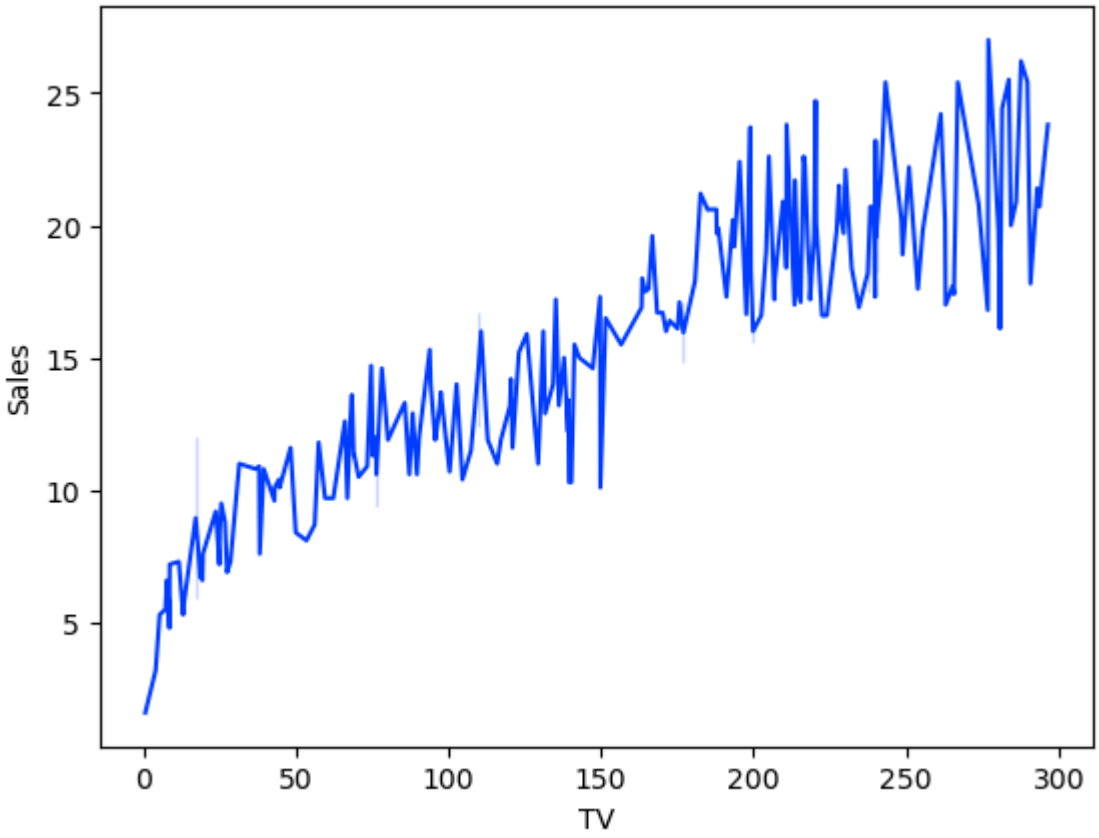
	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	6.9748	0.323	21.624	0.000	6.339	7.611
TV	0.0555	0.002	29.260	0.000	0.052	0.059

Omnibus:	0.013	**Durbin-Watson:**	2.029
Prob(Omnibus):	0.993	**Jarque-Bera (JB):**	0.043
Skew:	-0.018	**Prob(JB):**	0.979
Kurtosis:	2.938	**Cond. No.**	338.

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

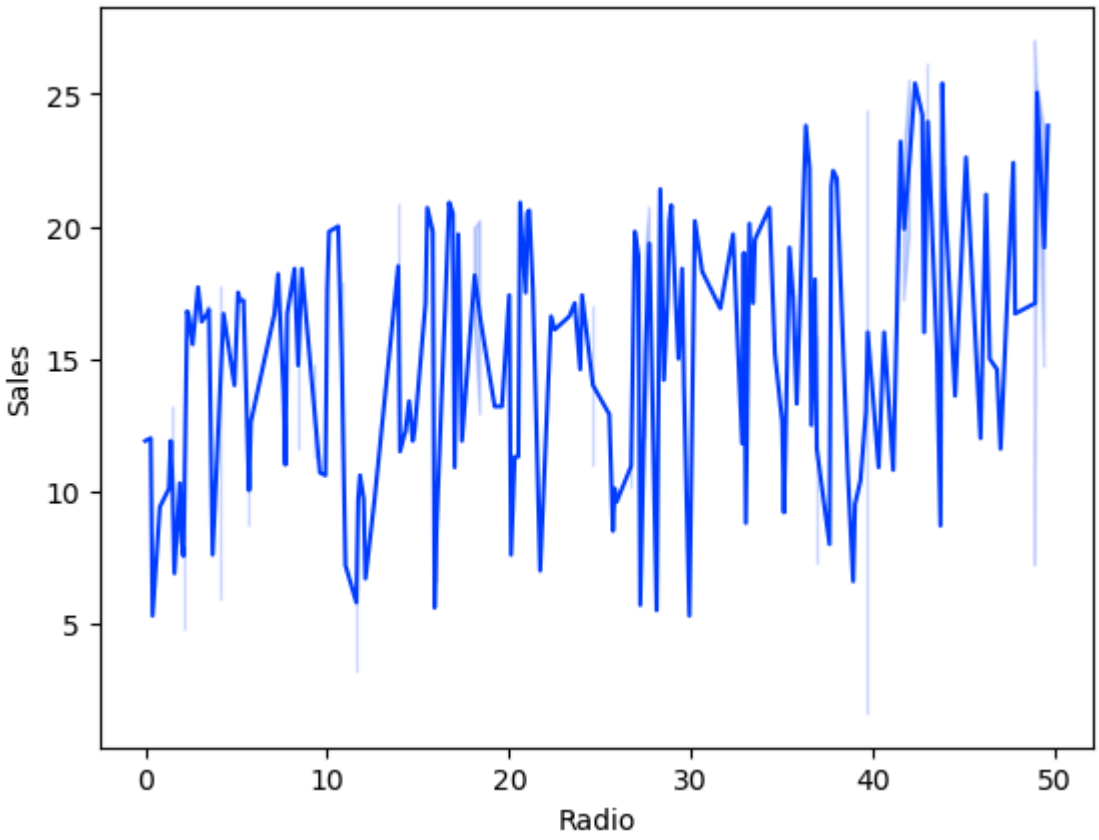
In [ ]:

In [15]:
```
#Linearity
```



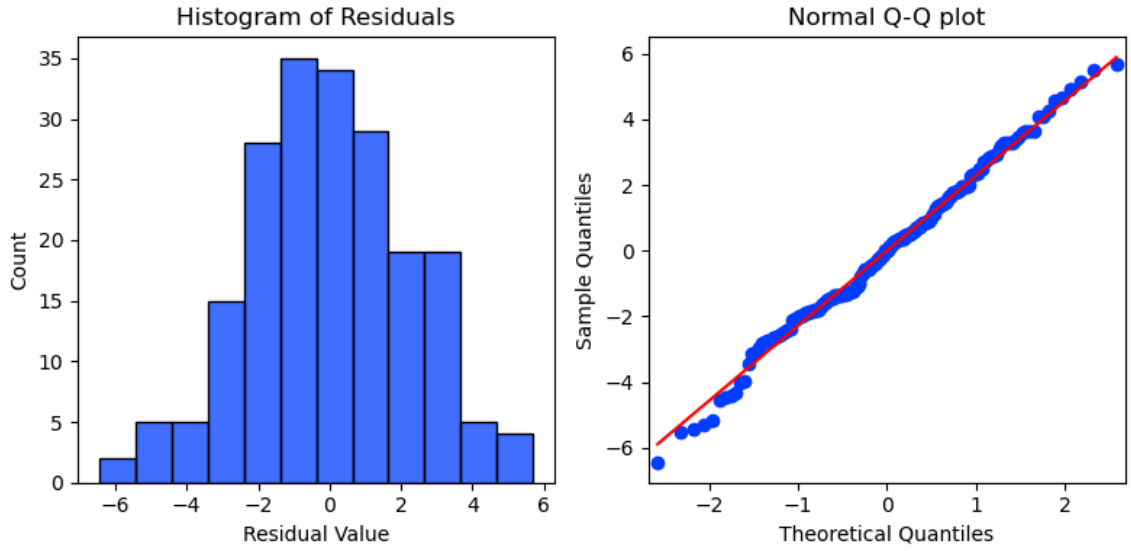
In [16]:
```
sns.lineplot(x=data['Radio'], y=data['Sales'])
```

Out[16]:
```
<Axes: xlabel='Radio', ylabel='Sales'>
```



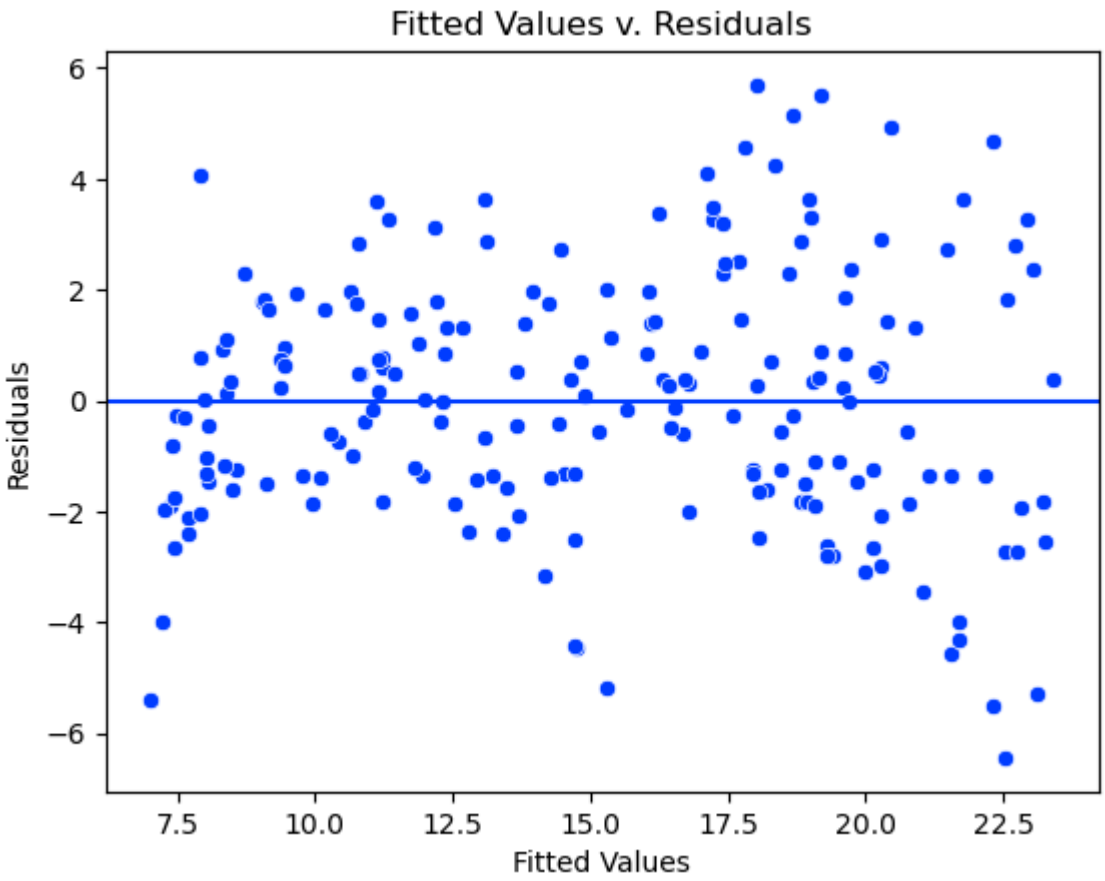
In [17]:
```
#Normality
residuals = model.resid
fig, axes = plt.subplots(1, 2, figsize = (8,4))
sns.histplot(residuals, ax=axes[0])
axes[0].set_xlabel("Residual Value")
axes[0].set_title("Histogram of Residuals")

sm.qqplot(residuals, line='s',ax = axes[1])
axes[1].set_title("Normal Q-Q plot")
plt.tight_layout()
plt.show()
```



In [18]:

```
###  homoscedasticity
fig = sns.scatterplot(x = model.fittedvalues, y = model.resid)
# Set the x-axis label.
fig.set_xlabel("Fitted Values")
# Set the y-axis label.
fig.set_ylabel("Residuals")
# Set the title.
fig.set_title("Fitted Values v. Residuals")
fig.axhline(0)
```



In the data visualization, TV has the strongest linear relationship with sales. Radio and sales appear to have a moderate linear relationship, but there is larger variance than between TV and sales. Newspaper and sales appear to have a weak linear relationship.

Y=Intercept+Slope∗X Sales (in millions)=Intercept+Slope∗TV (in millions)

In [ ]:

In [ ]: