In [218…

```python
# Import libraries
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from statsmodels.formula.api import ols
import statsmodels.api as sm
```

In [258…

```python
data = pd.read_csv(r'C:\Users\HP\Desktop\Advance Data Analyst\5. Simplify Complex Data Relationships\2. Module 2\3. Evaluate linear regression model\Files\marketing_and_sales_data_evaluate_lr.csv')
data.head(10)
```

Out[258…

	TV	Radio	Social_Media	Sales
0	16.0	6.566231	2.907983	54.732757
1	13.0	9.237765	2.409567	46.677897
2	41.0	15.886446	2.913410	150.177829
3	83.0	30.020028	6.922304	298.246340
4	15.0	8.437408	1.405998	56.594181
5	29.0	9.614382	1.027163	105.889148
6	55.0	24.893811	4.273602	198.679825
7	31.0	17.355042	2.289855	108.733932
8	76.0	24.648898	7.130116	270.189400
9	13.0	0.431128	2.229423	48.280582

In [260…

```python
# Display the first 10 rows of the data

data.head(10)
```

Out[260…

	TV	Radio	Social_Media	Sales
0	16.0	6.566231	2.907983	54.732757
1	13.0	9.237765	2.409567	46.677897
2	41.0	15.886446	2.913410	150.177829
3	83.0	30.020028	6.922304	298.246340
4	15.0	8.437408	1.405998	56.594181
5	29.0	9.614382	1.027163	105.889148
6	55.0	24.893811	4.273602	198.679825
7	31.0	17.355042	2.289855	108.733932
8	76.0	24.648898	7.130116	270.189400
9	13.0	0.431128	2.229423	48.280582

In [264…

```python
# Display number of rows, number of columns

data.shape
```

Out[264… (4572, 4)

In [266…

```python
# Generate descriptive statistics about TV, Radio, and Social_Media.

data[['TV','Radio','Social_Media']].describe()
```

Out[266…

	TV	Radio	Social_Media
count	4562.000000	4568.000000	4566.000000
mean	54.066857	18.160356	3.323956
std	26.125054	9.676958	2.212670
min	10.000000	0.000684	0.000031
25%	32.000000	10.525957	1.527849
50%	53.000000	17.859513	3.055565
75%	77.000000	25.649730	4.807558
max	100.000000	48.871161	13.981662

In [268…

```python
# Calculate the average missing rate in the sales column.
missing_sales = data.Sales.isna().mean()

# Convert the missing_sales from a decimal to a percentage and round to 2 decimal places.
missing_sales = round(missing_sales*100, 2)

# Display the results (missing_sales must be converted to a string to be concatenated in the print statement).
print('Percentage of promotions missing Sales: ' +  str(missing_sales) + '%')
```

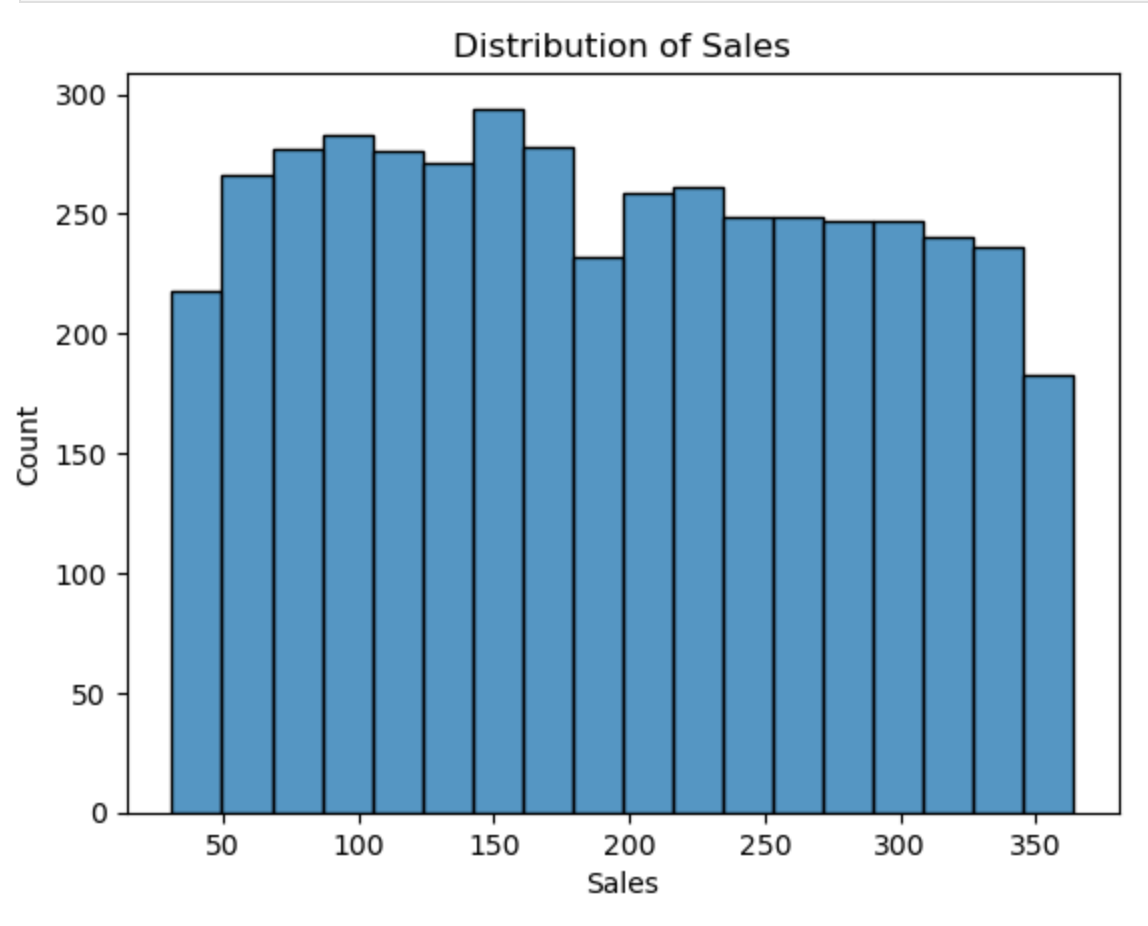
```
Percentage of promotions missing Sales: 0.13%
```

In [270…

```python
# Subset the data to include rows where Sales is present.

data = data.dropna(subset = ['Sales'], axis = 0)
```

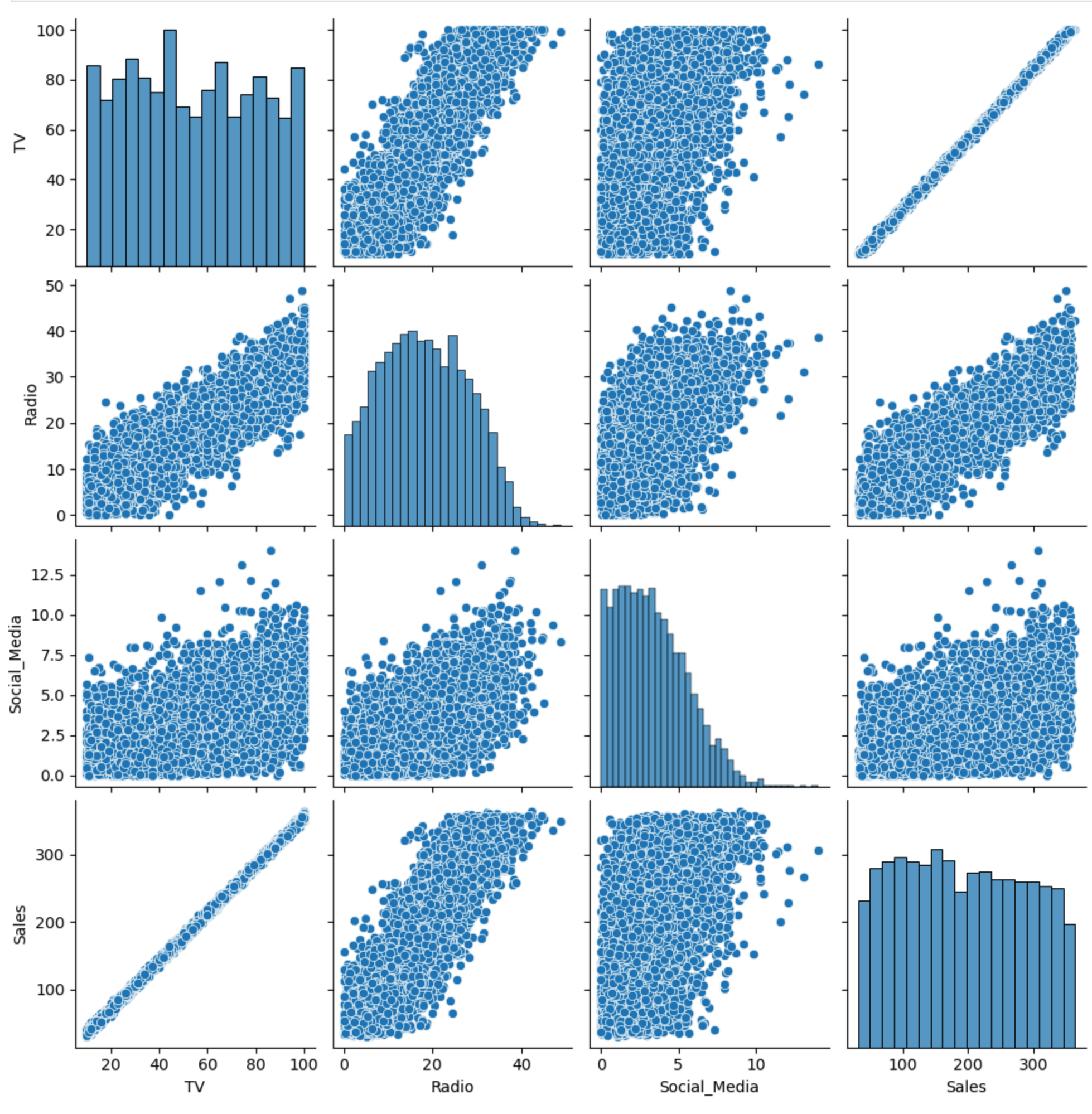
In [272…

```python
# Create a histogram of the Sales.

fig = sns.histplot(data['Sales'])

# Add a title
fig.set_title('Distribution of Sales');
```



In [274…

```python
# Create a pairplot of the data.

sns.pairplot(data);
```



In [276…

```python
# Define the OLS formula.
ols_formula = 'Sales ~ TV'

# Create an OLS model.
OLS = ols(formula = ols_formula, data = data)

# Fit the model.
model = OLS.fit()

# Save the results summary.
model_results = model.summary()

# Display the model results.
model_results
```

Out[276…

OLS Regression Results

Dep. Variable:	Sales	R-squared:	0.999
Model:	OLS	Adj. R-squared:	0.999
Method:	Least Squares	F-statistic:	4.527e+06
Date:	Fri, 27 Sep 2024	Prob (F-statistic):	0.00
Time:	13:22:19	Log-Likelihood:	-11393.
No. Observations:	4556	AIC:	2.279e+04
Df Residuals:	4554	BIC:	2.280e+04
Df Model:	1		
Covariance Type:	nonrobust		

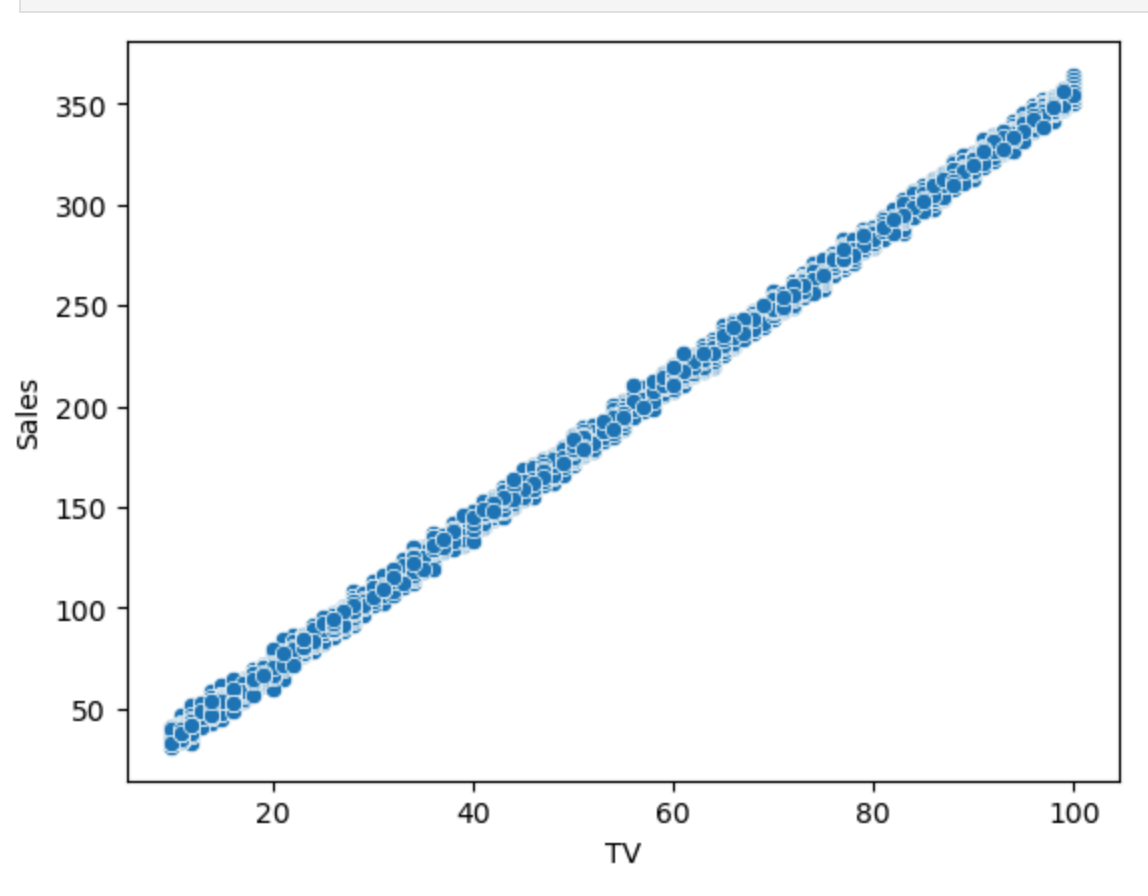
	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	-0.1263	0.101	-1.257	0.209	-0.323	0.071
TV	3.5614	0.002	2127.776	0.000	3.558	3.565

Omnibus:	0.051	Durbin-Watson:	2.002
Prob(Omnibus):	0.975	Jarque-Bera (JB):	0.030
Skew:	0.001	Prob(JB):	0.985
Kurtosis:	3.012	Cond. No.	138.

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

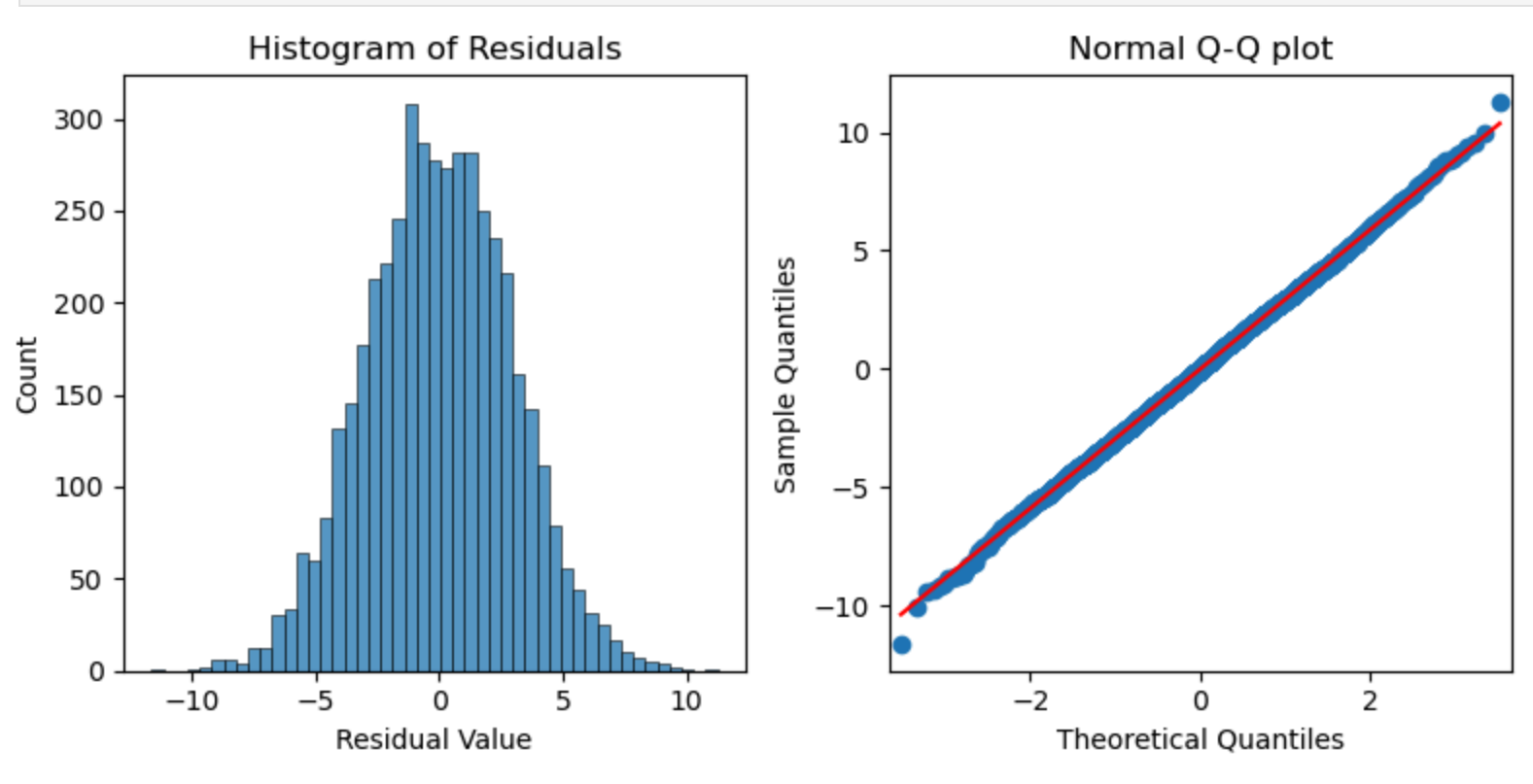
In [278…

```python
# Create a scatterplot comparing X and Sales (Y).

sns.scatterplot(x = data['TV'], y = data['Sales']);
```



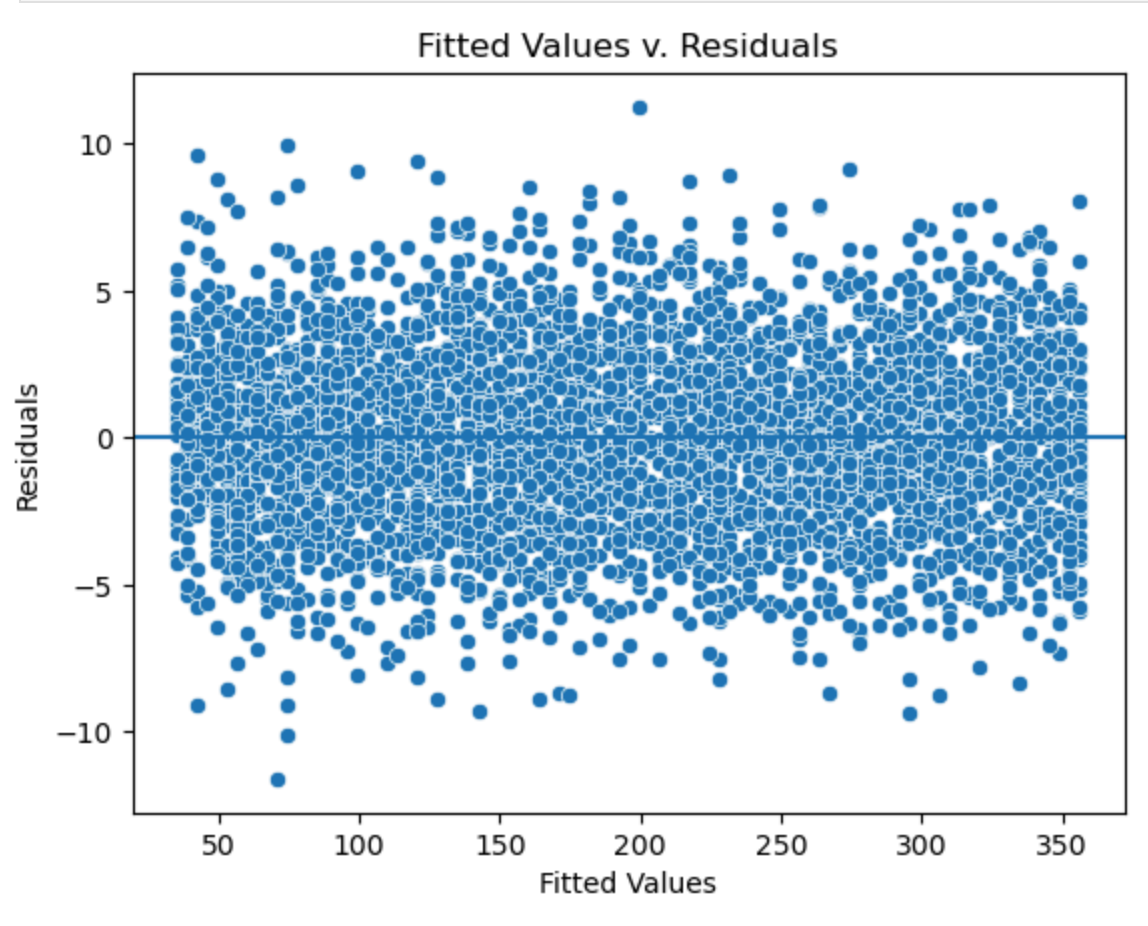
In [280…

```python
# Calculate the residuals.

residuals = model.resid

# Create a 1x2 plot figure.
fig, axes = plt.subplots(1, 2, figsize = (8,4))

# Create a histogram with the residuals .

sns.histplot(residuals, ax=axes[0])

# Set the x label of the residual plot.
axes[0].set_xlabel("Residual Value")

# Set the title of the residual plot.
axes[0].set_title("Histogram of Residuals")

# Create a Q-Q plot of the residuals.

sm.qqplot(residuals, line='s',ax = axes[1])

# Set the title of the Q-Q plot.
axes[1].set_title("Normal Q-Q plot")

# Use matplotlib's tight_layout() function to add space between plots for a cleaner appearance.
plt.tight_layout()

# Show the plot.
plt.show()
```



In [282…

```python
# Create a scatterplot with the fitted values from the model and the residuals.

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")

# Add a line at y = 0 to visualize the variance of residuals above and below 0.

fig.axhline(0)

# Show the plot.
plt.show()
```



In [284…

```python
# Display the model_results defined previously.

model_results
```

Out[284…

OLS Regression Results

Dep. Variable:	Sales	R-squared:	0.999
Model:	OLS	Adj. R-squared:	0.999
Method:	Least Squares	F-statistic:	4.527e+06
Date:	Fri, 27 Sep 2024	Prob (F-statistic):	0.00
Time:	13:22:19	Log-Likelihood:	-11393.
No. Observations:	4556	AIC:	2.279e+04
Df Residuals:	4554	BIC:	2.280e+04
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	-0.1263	0.101	-1.257	0.209	-0.323	0.071
TV	3.5614	0.002	2127.776	0.000	3.558	3.565

Omnibus:	0.051	Durbin-Watson:	2.002
Prob(Omnibus):	0.975	Jarque-Bera (JB):	0.030
Skew:	0.001	Prob(JB):	0.985
Kurtosis:	3.012	Cond. No.	138.

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

In [ ]:

```python
# Key takeaways

# Exploratory data analysis can be used to find a suitable X variable for a simple linear regression model.
# It's important to check that simple linear regression assumptions are met before interpreting model results.
# R-squared is a useful measure of prediction error.
# It's important to provide measures of uncertainty (e.g., p-values, confidence intervals) with any coefficient estimates.

# What findings would you share with others?

# Sales is relatively equally distributed between $25 and $350 million across all promotions.
# 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. Social media and sales appear to have a weak linear relationship.
# When TV is used as the independent variable, the R-squared for the simple linear regression model estimating sales is 0.999.
# When TV is used as the independent variable, the coefficient for the intercept is -0.1263 and the coefficient for TV is 3.5614.
# When TV is used as the independent variable, the slope of the coefficient for TV has a p-value of 0.000 and a 95% confidence interval of [3.558, 3.565].

# How would you frame your findings to stakeholders?

# Among TV, social media, and radio, TV had the strongest positive linear relationship with sales.
# The linear regression model estimates that 99.9% of the variation in sales is explained by the TV promotional budget.
```

```
# In other words, nearly all of the variation in sales can be explained by the TV promotional budget alone, making TV an excellent predictor of sales.
# According to the model, when TV is used as the independent variable X, an increase of one million dollars for the TV promotional budget would result in an estimated 3.5614 million more dollars in sales.
# The interval (3.518 million, 3.765 million) has a 95% probability of containing the true estimate of the increase in sales for a one million dollar increase in the TV promotional budget.
# Therefore, the estimate provided in the previous bullet is very confident.
```