# Linear regression

As you're learning, simple linear regression is a way to model the relationship between two variables. By assessing the direction and magnitude of a relationship, data professionals are able to uncover patterns and transform large amounts of data into valuable knowledge. This enables them to make better predictions and decisions.

The dataset provided includes information about marketing campaigns across TV, radio, and social media, as well as how much revenue in sales was generated from these campaigns. Based on this information, company leaders will make decisions about where to focus future marketing resources. Therefore, it is critical to provide them with a clear understanding of the relationship between types of marketing campaigns and the revenue generated as a result of this investment.

In [1]:
```
# Import relevant Python libraries and modules



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

# Data Exploration

In [3]:
```
# Load the dataset into a DataFrame and save in a variable


df = pd.read_csv("Fk8EtlRPRyWPBLZUT-cl2Q_83304da42a7a433aa14111ee7a7c79f1_
```

In [4]:
```
df.head()
```

Out[4]:

| | TV | Radio | Social Media | Influencer | Sales |
|---|---|---|---|---|---|
| **0** | Low | 1.218354 | 1.270444 | Micro | 90.054222 |
| **1** | Medium | 14.949791 | 0.274451 | Macro | 222.741668 |
| **2** | Low | 10.377258 | 0.061984 | Mega | 102.774790 |
| **3** | High | 26.469274 | 7.070945 | Micro | 328.239378 |
| **4** | High | 36.876302 | 7.618605 | Mega | 351.807328 |

In [6]:
```
# Display number of rows, number of columns


df.shape
```

Out[6]: (572, 5)

In [9]:
```
df.isna().sum()   #    check for missing value
```

Out[9]:
```
TV              1
Radio           1
Social Media    0
Influencer      0
Sales           1
dtype: int64
```

In [12]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 572 entries, 0 to 571
Data columns (total 5 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   TV            571 non-null    object 
 1   Radio         571 non-null    float64
 2   Social Media  572 non-null    float64
 3   Influencer    572 non-null    object 
 4   Sales         571 non-null    float64
dtypes: float64(3), object(2)
memory usage: 22.5+ KB
```

In [14]:
```
df.isna().any(axis=1).sum()
```

Out[14]: 3

There are three rows that contains missing values, drop them.

In [16]:
```
df = df.dropna(axis=0)
```

In [17]:
```
df.isna().any(axis=1).sum()
```

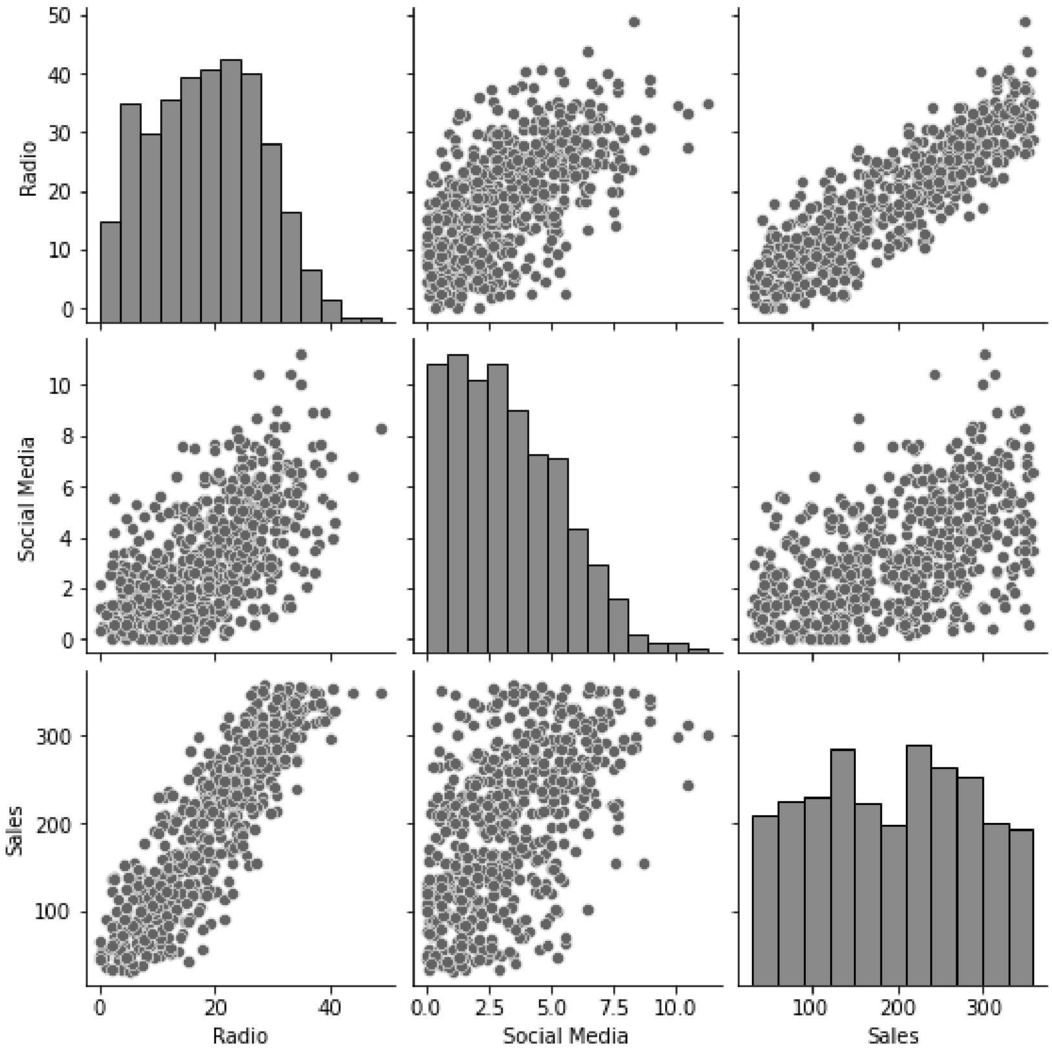
Out[17]: 0

# Check model assumptions.

# 1 linearity assumption

In [18]:
```
# Create plot of pairwise relationships


sns.pairplot(df)
```

Out[18]: <seaborn.axisgrid.PairGrid at 0x1548cf2f640>




Since the points cluster around a line, it seems the assumption of linearity is met.

# Model building

In [20]:
```
# Select relevant columns
# Save resulting DataFrame in a separate variable to prepare for regressio


ols_data = df[["Radio", "Sales"]]
```

In [23]:
```
# Display first 10 rows of the new DataFrame


ols_data.head(10)
```

Out[23]:

| | Radio | Sales |
|---|---|---|
| **0** | 1.218354 | 90.054222 |
| **1** | 14.949791 | 222.741668 |
| **2** | 10.377258 | 102.774790 |
| **3** | 26.469274 | 328.239378 |
| **4** | 36.876302 | 351.807328 |
| **5** | 25.561910 | 261.966812 |
| **6** | 37.263819 | 349.861575 |
| **7** | 13.187256 | 140.415286 |
| **8** | 29.520170 | 264.592233 |
| **9** | 3.773287 | 55.674214 |

In [24]:
```
# Write the linear regression formula
# Save it in a variable


ols_formula = "Sales ~ Radio"
```

In [25]:
```
# Implement OLS


OLS = ols(formula = ols_formula, data = ols_data)
```

In [26]:
```
# Fit the model to the data
# Save the fitted model in a variable



model = OLS.fit()
```

# Results and Evaluation

In [28]:
```
# Get summary of results



model.summary()
```

Out[28]:

OLS Regression Results

| | | | |
|---|---|---|---|
| **Dep. Variable:** | Sales | **R-squared:** | 0.757 |
| **Model:** | OLS | **Adj. R-squared:** | 0.757 |
| **Method:** | Least Squares | **F-statistic:** | 1768. |
| **Date:** | Wed, 31 May 2023 | **Prob (F-statistic):** | 2.07e-176 |
| **Time:** | 22:06:47 | **Log-Likelihood:** | -2966.7 |
| **No. Observations:** | 569 | **AIC:** | 5937. |
| **Df Residuals:** | 567 | **BIC:** | 5946. |
| **Df Model:** | 1 | | |
| **Covariance Type:** | nonrobust | | |

| | coef | std err | t | P>\|t\| | [0.025 | 0.975] |
|---|---|---|---|---|---|---|
| **Intercept** | 41.5326 | 4.067 | 10.211 | 0.000 | 33.544 | 49.521 |
| **Radio** | 8.1733 | 0.194 | 42.048 | 0.000 | 7.791 | 8.555 |

| | | | |
|---|---|---|---|
| **Omnibus:** | 2.267 | **Durbin-Watson:** | 1.880 |
| **Prob(Omnibus):** | 0.322 | **Jarque-Bera (JB):** | 2.221 |
| **Skew:** | -0.102 | **Prob(JB):** | 0.329 |
| **Kurtosis:** | 2.772 | **Cond. No.** | 45.7 |

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

The y-intercept is 41.5326.

The slope is 8.1733.

sales = 8.1733 * radio promotion budget + 41.5326

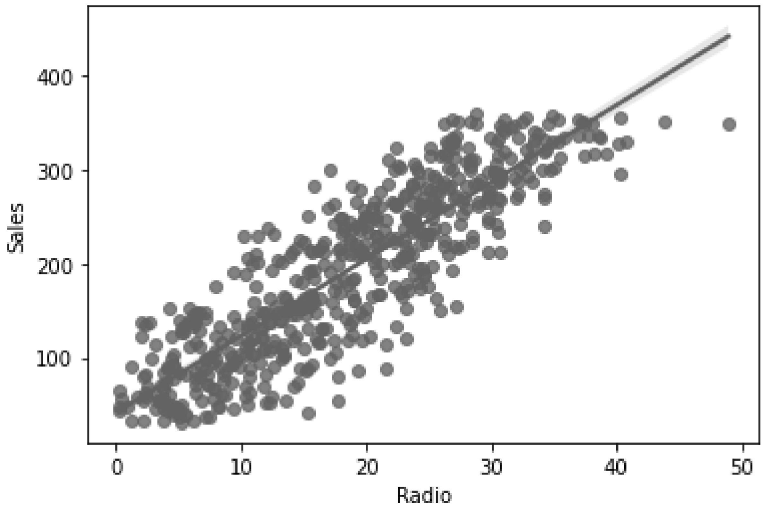
Companies with 1 million dollars more in their radio promotion budget accrue 8.1733 million dollars more in sales on average.

In [30]:

```
# Plot the OLS data with the best fit regression line



sns.regplot(x = "Radio", y = "Sales", data = ols_data)
```

Out[30]:

```
<AxesSubplot:xlabel='Radio', ylabel='Sales'>
```



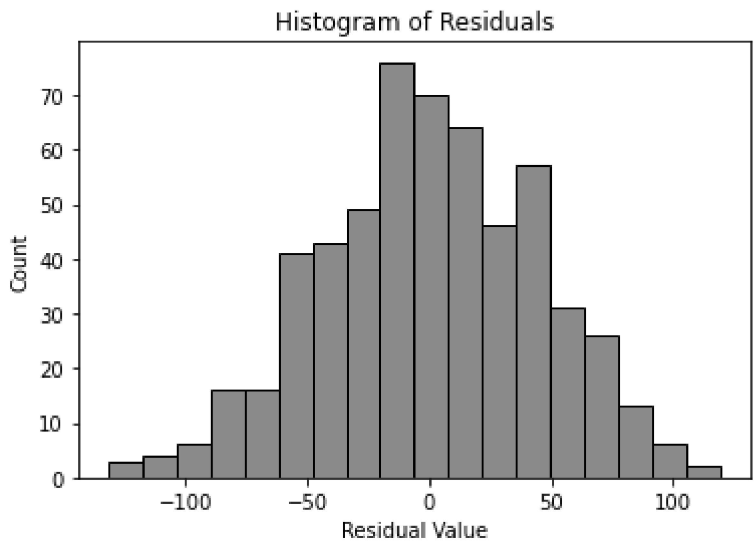

# Check the normality assumption.

In [31]:

```
# Get the residuals from the model


residuals = model.resid
```

In [34]:

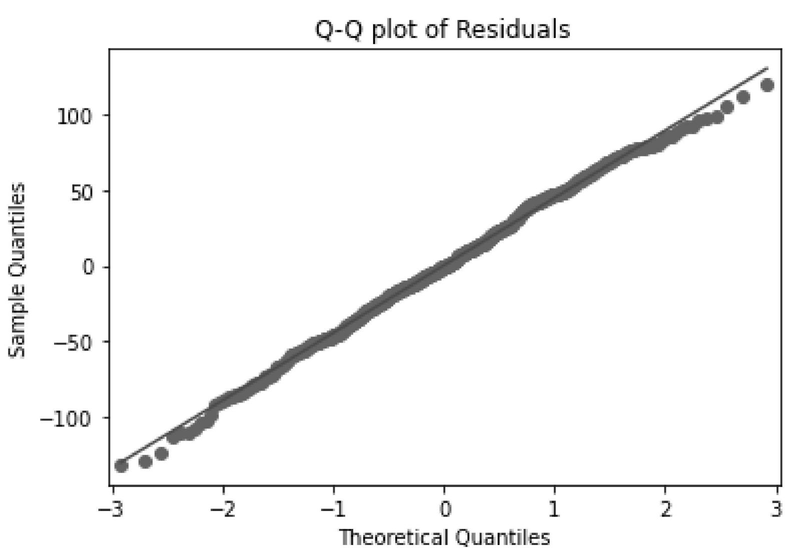
```
# Visualize the distribution of the residuals



fig = sns.histplot(residuals)
fig.set_xlabel("Residual Value")
fig.set_title("Histogram of Residuals")
plt.show()
```




The distribution of the residuals is approximately normal. This indicates that the assumption of normality is likely met.

In [35]:

```
# Create a Q-Q plot


sm.qqplot(residuals, line='s')
plt.title("Q-Q plot of Residuals")
plt.show()
```

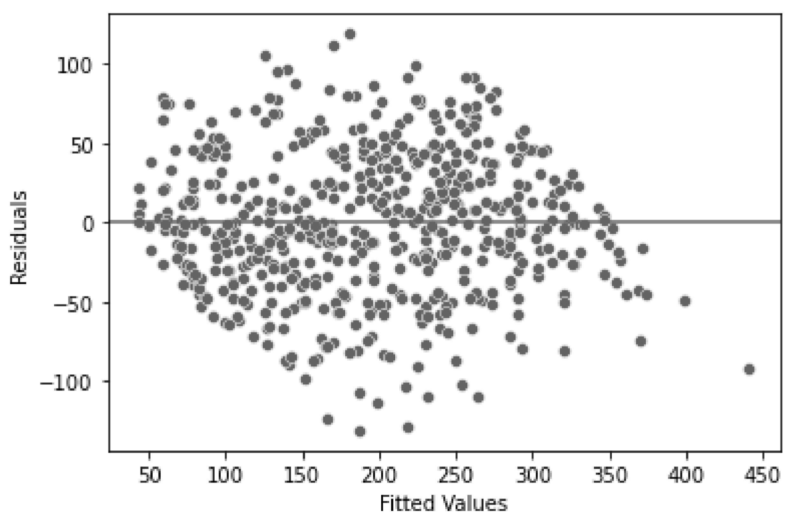


The points closely follow a straight diagonal line trending upward. This confirms that the normality assumption is met

# Check the assumptions of independent observation and homoscedasticity.

In [37]:
```
# Get fitted values


fitted_values = model.predict(ols_data["Radio"])
```

In [38]:
```
# Create a scatterplot of residuals against fitted values


fig = sns.scatterplot(x=fitted_values, y=residuals)
fig.axhline(0)
fig.set_xlabel("Fitted Values")
fig.set_ylabel("Residuals")
plt.show()
```




In the preceding scatterplot, the data points have a cloud-like resemblance and do not follow an explicit pattern. So it appears that the independent observation assumption has not been violated. Given that the residuals appear to be randomly spaced, the homoscedasticity assumption seems to be met.

# Findings

In the simple linear regression model, the y-intercept is 41.5326 and the slope is 8.1733. One interpretation: If a company has a budget of 1 million dollars more for promoting their products/services on the radio, the company's sales would increase by 8.1733 million dollars on average. Another interpretation: Companies with 1 million dollars more in their radio promotion budget accrue 8.1733 million dollars more in sales on average.

The results are statistically significant with a p-value of 0.000, which is a very small value (and smaller than the common significance level of 0.05). This indicates that there is a very low probability of observing data as extreme or more extreme than this dataset when the null hypothesis is true. In this context, the null hypothesis is that there is no relationship between radio promotion budget and sales i.e. the slope is zero, and the alternative hypothesis is that there is a relationship between radio promotion budget and sales i.e. the slope is not zero. So, you could reject the null hypothesis and state that there is a relationship between radio promotion budget and sales for companies in this data.

The slope of the line of best fit that resulted from the regression model is approximate and subject to uncertainty (not the exact value). The 95% confidence interval for the slope is from 7.791 to 8.555. This indicates that there is a 95% probability that the interval [7.791, 8.555] contains the true value for the slope.

# Recommendations

Based on the dataset at hand and the regression analysis conducted here, there is a notable relationship between radio promotion budget and sales for companies in this data, with a p-value of 0.000 and standard error of 0.194. For companies represented by this data, a 1 million dollar increase in radio promotion budget could be accociated with a 8.1733 million dollar increase in sales. It would be worth continuing to promote products/services on the radio. Also, it is recommended to consider further examining the relationship between the two variables (radio promotion budget and sales) in different contexts. For example, it would help to gather more data to understand whether this relationship is different in certain industries or when promoting certain types of products/services.

In [ ]:

```
1
```