Activity_Perform multiple linear regression

August 20, 2024

1 Performing multiple linear regression

1.1 Introduction

For this project, i will be analyzing a small business' historical marketing promotion data. Each row corresponds to an independent marketing promotion where their business uses TV, social media, radio, and influencer promotions to increase sales.

To address the business' request, i will conduct a multiple linear regression analysis to estimate sales from a combination of independent variables. This will include:

- Exploring and cleaning data
- Using plots and descriptive statistics to select the independent variables
- Creating a fitting multiple linear regression model
- Checking model assumptions
- Interpreting model outputs and communicating the results to non-technical stakeholders

1.2 Step 1: Imports

1.2.1 Import packages

Import relevant Python libraries and modules.

[1]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import statsmodels.api as sm
from statsmodels.formula.api import ols
```

1.2.2 Load dataset

[2]:
```
data = pd.read_csv('marketing_sales_data.csv')

# The first five rows.
data.head()
```

[2]:

	TV	Radio	Social Media	Influencer	Sales
0	Low	3.518070	2.293790	Micro	55.261284
1	Low	7.756876	2.572287	Mega	67.574904
2	High	20.348988	1.227180	Micro	272.250108
3	Medium	20.108487	2.728374	Mega	195.102176
4	High	31.653200	7.776978	Nano	273.960377

1.3 Step 2: Data exploration

The features in the data are:

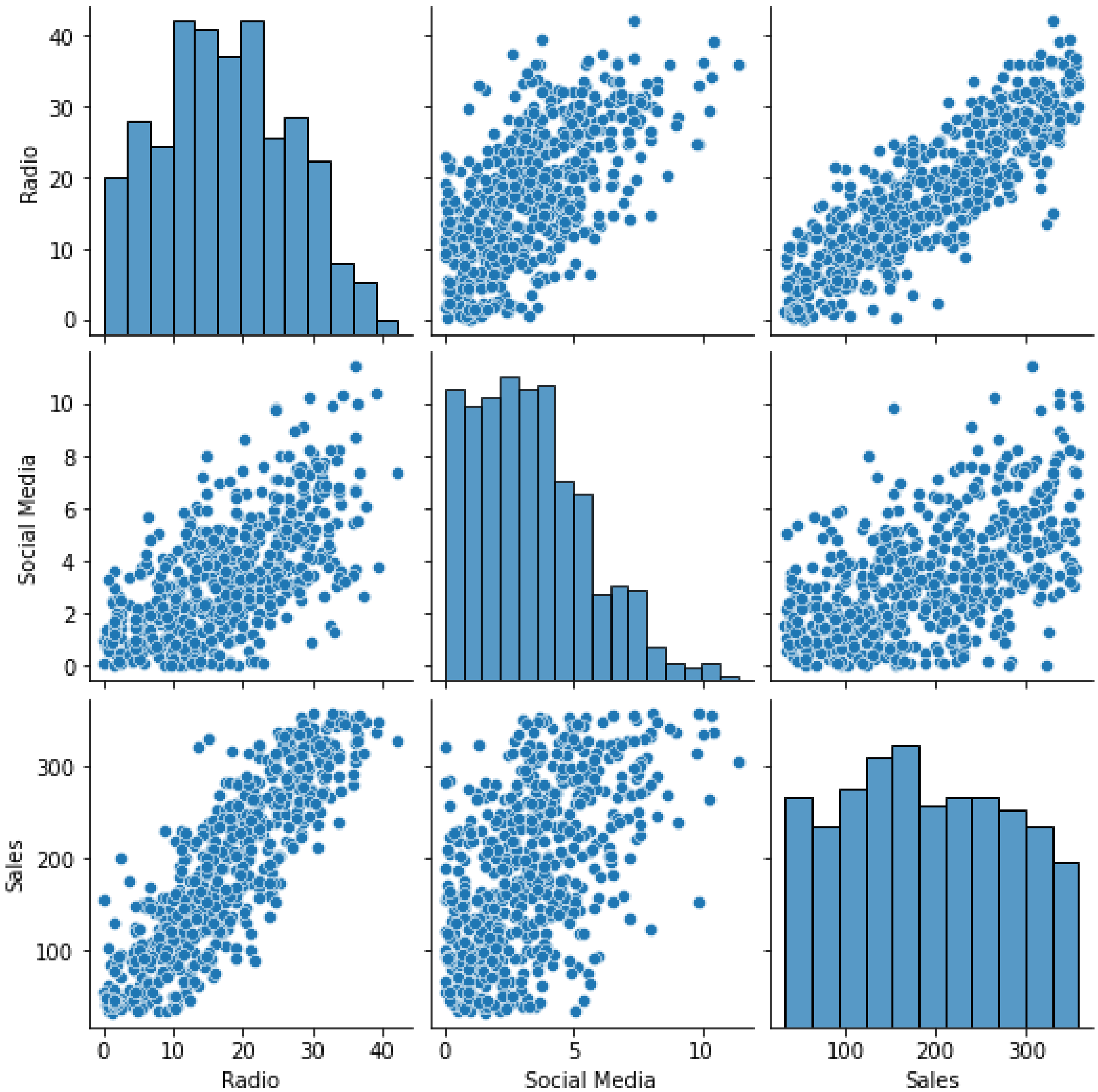
- TV promotional budget (in "Low," "Medium," and "High" categories)
- Social media promotional budget (in millions of dollars)
- Radio promotional budget (in millions of dollars)
- Sales (in millions of dollars)
- Influencer size (in "Mega," "Macro," "Micro," and "Nano" categories)

1.3.1 pairplot of the data

pairplot to visualize the relationship between the continous variables in `data`.

[3]:
```
sns.pairplot(data)
```

[3]: <seaborn.axisgrid.PairGrid at 0x71a15e53b2d0>



`Radio` and Social `Media` both appear to have linear relationships with Sales.

`TV` and `Influencer` are excluded from the pairplot because they are not numeric.

1.3.2 The mean sales for each categorical variable

[4]:
```
# the mean sales for each TV category.

print(data.groupby('TV')['Sales'].mean())


# the mean sales for each Influencer category.
```

```
print(data.groupby('Influencer')['Sales'].mean())
```

```
TV
High      300.853195
Low        90.984101
Medium    195.358032
Name: Sales, dtype: float64
Influencer
Macro    181.670070
Mega     194.487941
Micro    188.321846
Nano     191.874432
Name: Sales, dtype: float64
```

The average Sales for High `TV` promotions is considerably higher than for Medium and Low `TV` promotions. `TV` may be a strong predictor of Sales.

The categories for `Influencer` have different average `Sales`, but the variation is not substantial. `Influencer` may be a weak predictor of `Sales`.

[5]:
```
data = data.dropna(axis=0)
```

1.3.3 Clean column names

The `ols()` function doesn't run when variable names contain a space.

[6]:
```
# Rename all columns in data that contain a space.

data = data.rename(columns={'Social Media': 'Social_Media'})
```

1.4 Step 3: Model building

1.4.1 Fit a multiple linear regression model that predicts sales

- `TV` was selected as an independent variable, as the preceding analysis showed a strong relationship between the `TV` promotional budget and the average `Sales`.
- `Radio` was selected because the pairplot showed a strong linear relationship between `Radio` and `Sales`.
- `Social Media` was not selected because it did not increase model performance and it was later determined to be correlated with another independent variable: `Radio`.
- `Influencer` was not selected because it did not show a strong relationship to `Sales` in the preceding analysis.

[7]:
```
# Define the OLS formula.
ols_formula = 'Sales ~ C(TV) + Radio'

# Create an OLS model.
```

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

# Fit the model.

model = OLS.fit()

# Save the results summary.

results = model.summary()

# Display the model results.

results
```

[7]:
```
<class 'statsmodels.iolib.summary.Summary'>
"""
                            OLS Regression Results
==============================================================================
Dep. Variable:                  Sales   R-squared:                       0.904
Model:                            OLS   Adj. R-squared:                  0.904
Method:                 Least Squares   F-statistic:                     1783.
Date:                Tue, 20 Aug 2024   Prob (F-statistic):          1.63e-288
Time:                        21:00:48   Log-Likelihood:                -2714.0
No. Observations:                 572   AIC:                             5436.
Df Residuals:                     568   BIC:                             5453.
Df Model:                           3
Covariance Type:            nonrobust
================================================================================
===
                      coef    std err          t      P>|t|      [0.025
0.975]
--------------------------------------------------------------------------------
---
Intercept         218.5261      6.261     34.902      0.000     206.228
230.824
C(TV)[T.Low]     -154.2971      4.929    -31.303      0.000    -163.979
-144.616
C(TV)[T.Medium]   -75.3120      3.624    -20.780      0.000     -82.431
-68.193
Radio               2.9669      0.212     14.015      0.000       2.551
3.383
==============================================================================
Omnibus:                       61.244   Durbin-Watson:                   1.870
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               18.077
Skew:                           0.046   Prob(JB):                     0.000119
Kurtosis:                       2.134   Cond. No.                         142.
```

```
==============================================================================

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

1.4.2 Checking model assumptions

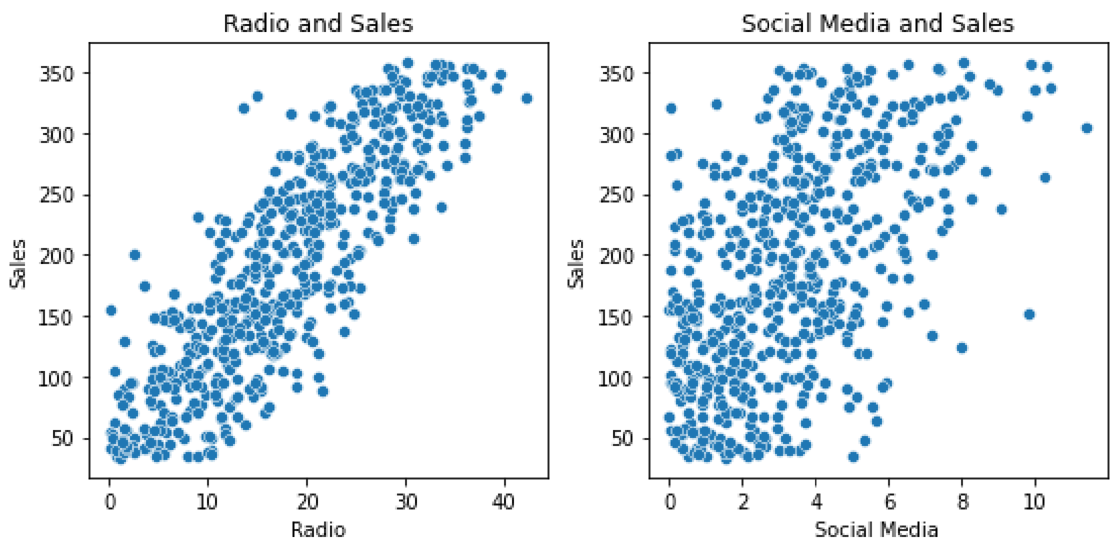
1.4.3 Model assumption: Linearity

scatterplots comparing the continuous independent variable(s) we selected previously with `Sales` to check the linearity assumption.

[8]:
```
fig, axes = plt.subplots(1,2, figsize=(8,4))
sns.scatterplot(x=data['Radio'], y=data['Sales'], ax=axes[0])
axes[0].set_title("Radio and Sales")

sns.scatterplot(x = data['Social_Media'], y = data['Sales'],ax=axes[1])
axes[1].set_title("Social Media and Sales")
axes[1].set_xlabel("Social Media")

plt.tight_layout()
```



The linearity assumption holds for `Radio`. `Social Media` was not included in the preceding multiple linear regression model, but it does appear to have a linear relationship with Sales.

1.4.4 Model assumption: Independence

The **independent observation assumption** states that each observation in the dataset is independent. As each marketing promotion (i.e., row) is independent from one another, the independence assumption is not violated.

1.4.5 Model assumption: Normality

- **Plot 1**: Histogram of the residuals
- **Plot 2**: Q-Q plot of the residuals

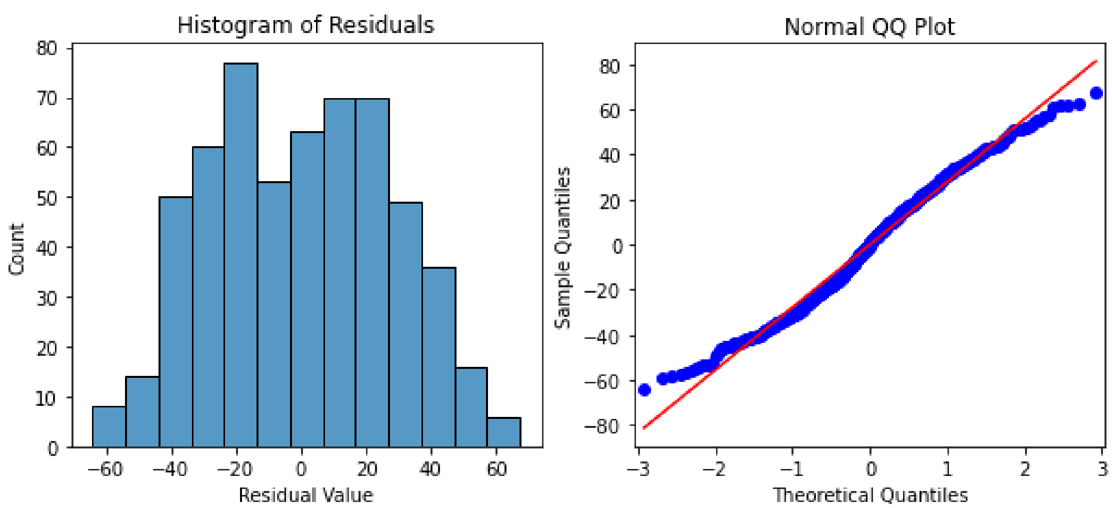
[9]:
```
# Calculate the residuals.
residuals = model.resid

fig, axes = plt.subplots(1, 2, figsize=(10,4))

# histogram with the residuals.
sns.histplot(residuals, ax=axes[0])
axes[0].set_xlabel("Residual Value")
axes[0].set_title("Histogram of Residuals")

# Create a Q-Q plot of the residuals.
sm.qqplot(residuals, line='s',ax = axes[1])
axes[1].set_title("Normal QQ Plot")

plt.show()
```



The histogram of the residuals are approximately normally distributed, which supports that the normality assumption is met for this model. The residuals in the Q-Q plot form a straight line, further supporting that this assumption is met.

1.4.6 Model assumption: Constant variance

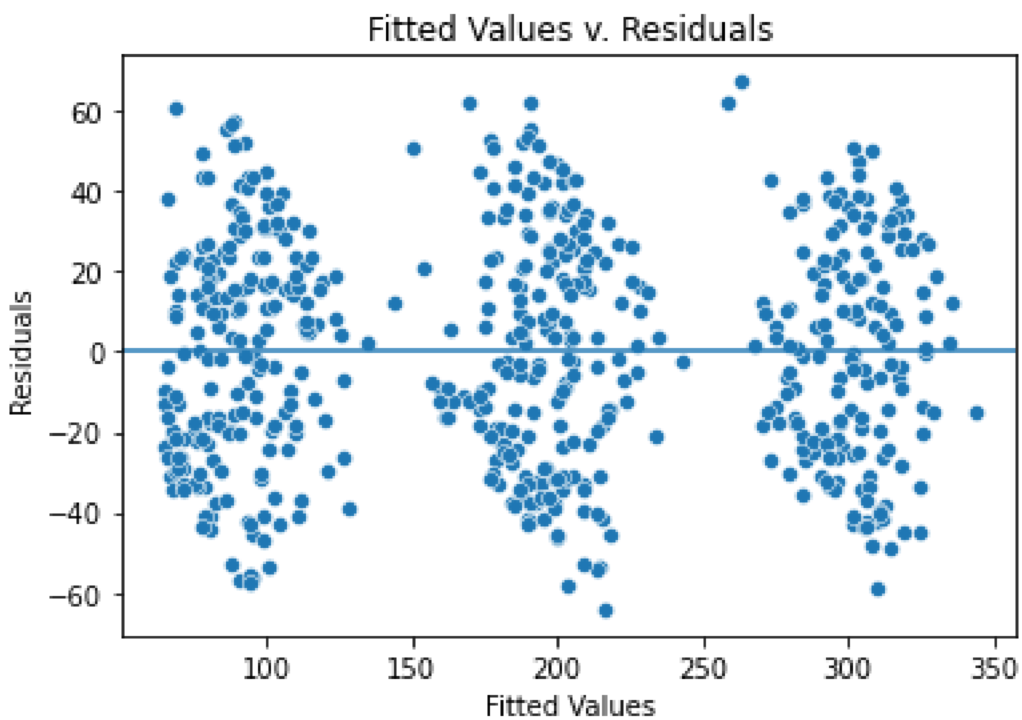
Checking that the **constant variance assumption** is not violated by creating a scatterplot with the fitted values and residuals. Add a line at $y = 0$ to visualize the variance of residuals above and below $y = 0$.

[10]:
```
# Create a scatterplot with the fitted values from the model and the residuals.
fig = sns.scatterplot(x=model.fittedvalues, y=residuals)

fig.set_xlabel("Fitted Values")
fig.set_ylabel("Residuals")
fig.set_title("Fitted Values v. Residuals")

fig.axhline(0)

plt.show()
```



The fitted values are in three groups because the categorical variable is dominating in this model, meaning that TV is the biggest factor that decides the sales.

However, the variance where there are fitted values is similarly distributed, validating that the assumption is met.

1.4.7 Model assumption: No multicollinearity

The **no multicollinearity assumption** states that no two independent variables (X_i and X_j) can be highly correlated with each other.

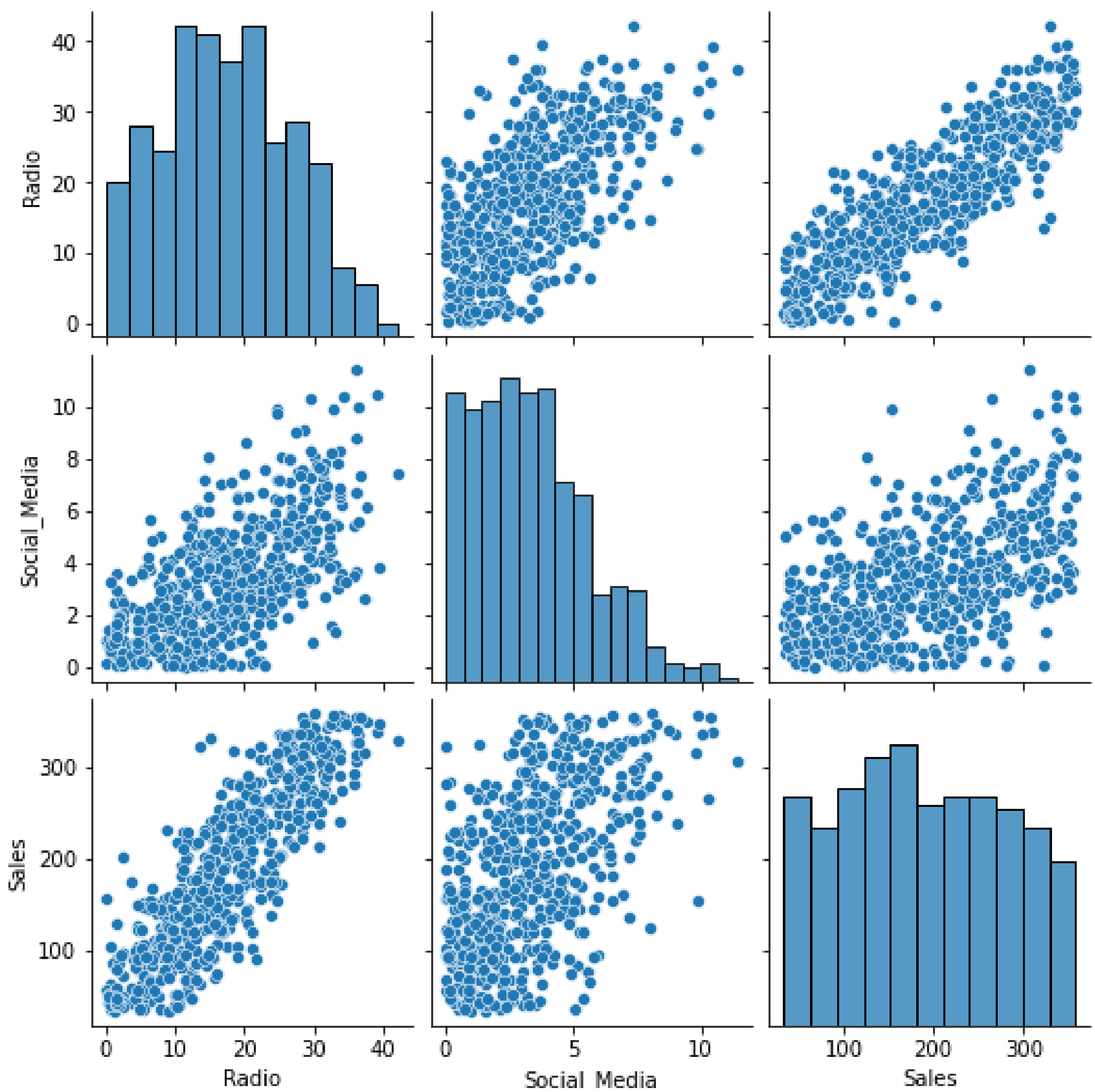
Two common ways to check for multicollinearity are to:

- Scatterplots to show the relationship between pairs of independent variables
- The variance inflation factor to detect multicollinearity

[11]:
```
# pairplot of the data.

sns.pairplot(data)
```

[11]: <seaborn.axisgrid.PairGrid at 0x71a15a69fc90>



[12]:
```python
# the variance inflation factor (optional).

# Import variance_inflation_factor from statsmodels.
from statsmodels.stats.outliers_influence import variance_inflation_factor

# Create a subset of the data with the continous independent variables.
X = data[['Radio','Social_Media']]

# Calculate the variance inflation factor for each variable.
vif = [variance_inflation_factor(X.values, i) for i in range(X.shape[1])]

# Create a DataFrame with the VIF results for the column names in X.
df_vif = pd.DataFrame(vif, index=X.columns, columns = ['VIF'])

# Display the VIF results.
df_vif
```

[12]:
```
                   VIF
Radio         5.170922
Social_Media  5.170922
```

The preceding model only has one continous independent variable, meaning there are no multicollinearity issues.

If a model used both Radio and Social_Media as predictors, there would be a moderate linear relationship between Radio and Social_Media that violates the multicollinearity assumption.

1.5 Step 4: Results and evaluation

1.5.1 Display the OLS regression results

[13]:
```python
# Display the model results summary.

results
```

[13]:
```
<class 'statsmodels.iolib.summary.Summary'>
"""
                            OLS Regression Results
==============================================================================
Dep. Variable:                  Sales   R-squared:                       0.904
Model:                            OLS   Adj. R-squared:                  0.904
Method:                 Least Squares   F-statistic:                     1783.
Date:                Tue, 20 Aug 2024   Prob (F-statistic):          1.63e-288
Time:                        21:00:48   Log-Likelihood:                -2714.0
No. Observations:                 572   AIC:                             5436.
```

```
Df Residuals:                     568   BIC:                             5453.
Df Model:                           3
Covariance Type:            nonrobust
================================================================================
===
                      coef    std err          t      P>|t|      [0.025
0.975]
--------------------------------------------------------------------------------
---
Intercept         218.5261      6.261     34.902      0.000     206.228
230.824
C(TV)[T.Low]     -154.2971      4.929    -31.303      0.000    -163.979
-144.616
C(TV)[T.Medium]   -75.3120      3.624    -20.780      0.000     -82.431
-68.193
Radio               2.9669      0.212     14.015      0.000       2.551
3.383
==============================================================================
Omnibus:                       61.244   Durbin-Watson:                   1.870
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               18.077
Skew:                           0.046   Prob(JB):                     0.000119
Kurtosis:                       2.134   Cond. No.                         142.
==============================================================================

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

The model explains 90.4% of the variation in `Sales`. This makes the model an excellent predictor of `Sales`.

1.5.2 Interpret model coefficients

[14]:
```
# Display the model results summary.

results
```

[14]:
```
<class 'statsmodels.iolib.summary.Summary'>
"""
                            OLS Regression Results
==============================================================================
Dep. Variable:                  Sales   R-squared:                       0.904
Model:                            OLS   Adj. R-squared:                  0.904
Method:                 Least Squares   F-statistic:                     1783.
Date:                Tue, 20 Aug 2024   Prob (F-statistic):          1.63e-288
```

```
Time:                        21:00:48   Log-Likelihood:                -2714.0
No. Observations:                 572   AIC:                             5436.
Df Residuals:                     568   BIC:                             5453.
Df Model:                           3
Covariance Type:            nonrobust
================================================================================
===
                     coef    std err          t      P>|t|      [0.025
0.975]
--------------------------------------------------------------------------------
---
Intercept         218.5261      6.261     34.902      0.000     206.228
230.824
C(TV)[T.Low]     -154.2971      4.929    -31.303      0.000    -163.979
-144.616
C(TV)[T.Medium]   -75.3120      3.624    -20.780      0.000     -82.431
-68.193
Radio               2.9669      0.212     14.015      0.000       2.551
3.383
==============================================================================
Omnibus:                       61.244   Durbin-Watson:                   1.870
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               18.077
Skew:                           0.046   Prob(JB):                     0.000119
Kurtosis:                       2.134   Cond. No.                         142.
==============================================================================

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

Question: What are the model coefficients?

When `TV` and `Radio` are used to predict Sales, the model coefficients are:

- $\beta_0 = 218.5261$
- $\beta_{TV\,Low} = -154.2971$
- $\beta_{TV\,Medium} = -75.3120$
- $\beta_{Radio} = 2.9669$

The relationship between `Sales` and the independent variables as a linear equation:

Sales $= \beta_0 + \beta_1 * X_1 + \beta_2 * X_2 + \beta_3 * X_3$

Sales $= \beta_0 + \beta_{TV\,Low} * X_{TV\,Low} + \beta_{TV\,Medium} * X_{TV\,Medium} + \beta_{Radio} * X_{Radio}$

Sales $= 218.5261 - 154.2971 * X_{TV\,Low} - 75.3120 * X_{TV\,Medium} + 2.9669 * X_{Radio}$

Question: What is the intepretation of the coefficient estimates? Are the coefficients statistically significant?

The default `TV` category for the model is `High` since there are coefficients for the other two `TV` categories, `Medium` and `Low`. Because the coefficients for the `Medium` and `Low TV` categories are negative, that means the average of sales is lower for `Medium` or `Low TV` categories compared to the `High TV` category when `Radio` is at the same level.

The coefficient for `Radio` is positive, confirming the positive linear relationship shown earlier during the exploratory data analysis.

The p-value for all coefficients is 0.000, meaning all coefficients are statistically significant at $p = 0.05$.

1.6 Conclusion

High TV promotional budgets have a substantial positive influence on sales. The model estimates that switching from a high to medium TV promotional budget reduces sales by \$75.3120 million (95% CI $[-82.431, -68.193]$), and switching from a high to low TV promotional budget reduces sales by \$154.297 million (95% CI $[-163.979, -144.616]$). The model also estimates that an increase of \$1 million in the radio promotional budget will yield a \$2.9669 million increase in sales (95% CI $[2.551, 3.383]$).

Thus, it is **recommended** that the business allot a high promotional budget to TV when possible and invest in radio promotions to increase sales.