Activity_Perform multiple linear regression

November 10, 2023

1 Activity: Perform multiple linear regression

1.1 Introduction

As you have learned, multiple linear regression helps you estimate the linear relationship between one continuous dependent variable and two or more independent variables. For data science professionals, this is a useful skill because it allows you to compare more than one variable to the variable you're measuring against. This provides the opportunity for much more thorough and flexible analysis.

For this activity, you 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. They previously had you work on finding a single variable that predicts sales, and now they are hoping to expand this analysis to include other variables that can help them target their marketing efforts.

To address the business' request, you 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.

[6]:
```
# Import libraries and modules.

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

1.2.2 Load dataset

`Pandas` was used to load the dataset `marketing_sales_data.csv` as `data`, now display the first five rows. The variables in the dataset have been adjusted to suit the objectives of this lab. As shown in this cell, the dataset has been automatically loaded in for you. You do not need to download the .csv file, or provide more code, in order to access the dataset and proceed with this lab. Please continue with this activity by completing the following instructions.

[3]:
```
# RUN THIS CELL TO IMPORT YOUR DATA.

### YOUR CODE HERE ###
data = pd.read_csv('marketing_sales_data.csv')

# Display the first five rows.

data.head(5)
```

[3]:

	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

1.3.1 Familiarize yourself with the data's features

Start with an exploratory data analysis to familiarize yourself with the data and prepare it for modeling.

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)

Question: What are some purposes of EDA before constructing a multiple linear regression model?

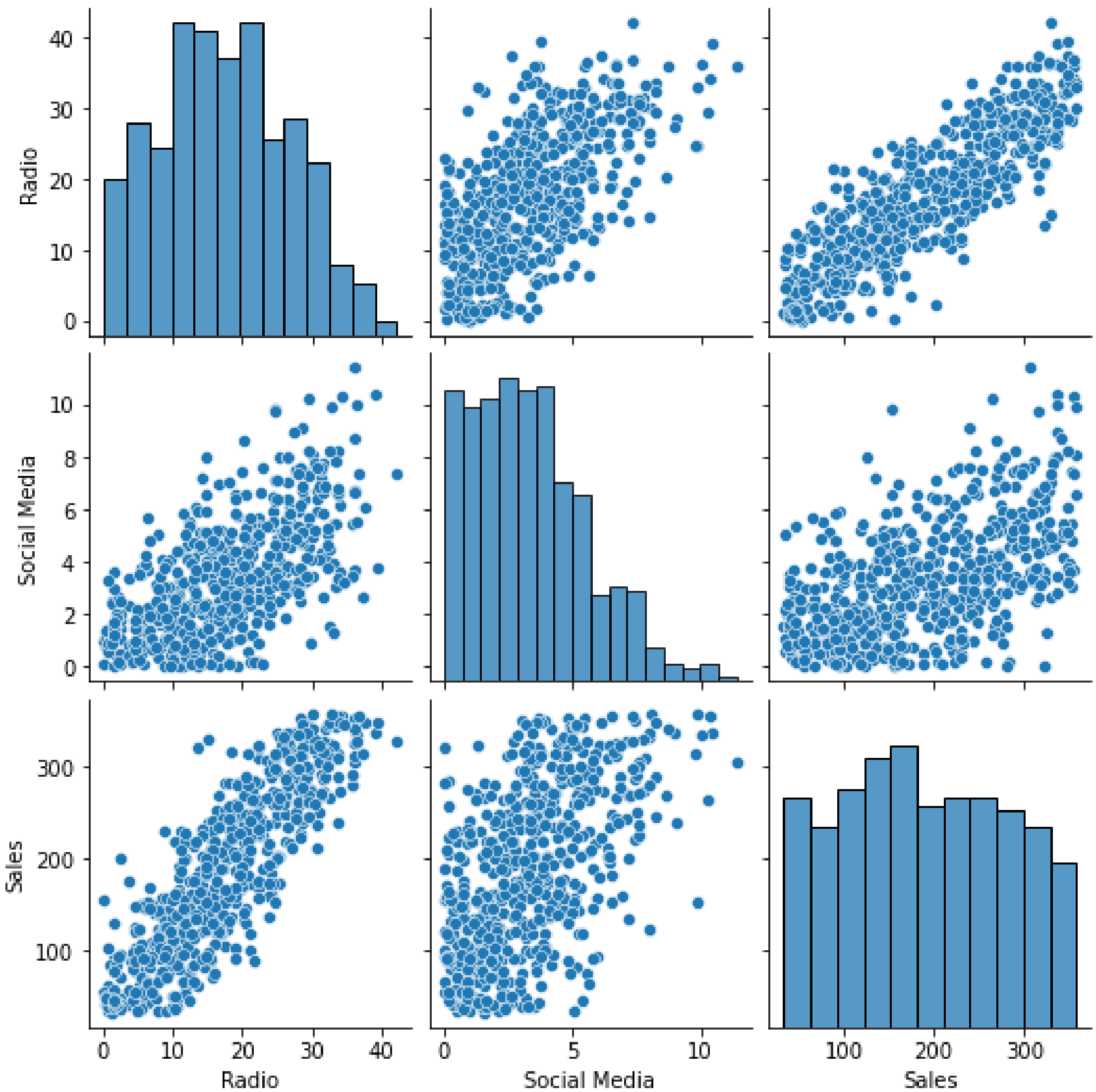
[Write your response here. Double-click (or enter) to edit.]

1.3.2 Create a pairplot of the data

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

[4]:
```python
# Create a pairplot of the data.


sns.pairplot(data)
```

[4]: <seaborn.axisgrid.PairGrid at 0x7f35a8daa7d0>



Hint 1

Refer to the content where creating a pairplot is demonstrated.

Hint 2

Use the function in the `seaborn` library that allows you to create a pairplot showing the relationships between variables in the data.

Hint 3

Use the `pairplot()` function from the `seaborn` library and pass in the entire DataFrame.

Question: Which variables have a linear relationship with `Sales`? Why are some variables in the data excluded from the preceding plot?

There's a relationship between radio, Social media and sales. There are some vaiables exclude due to the variables not being numeric.

1.3.3 Calculate the mean sales for each categorical variable

There are two categorical variables: `TV` and `Influencer`. To characterize the relationship between the categorical variables and `Sales`, find the mean `Sales` for each category in `TV` and the mean `Sales` for each category in `Influencer`.

[7]:
```
# Calculate the mean sales for each TV category.

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


# Calculate 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
```

Hint 1

Find the mean `Sales` when the `TV` promotion is `High`, `Medium`, or `Low`.

Find the mean `Sales` when the `Influencer` promotion is `Macro`, `Mega`, `Micro`, or `Nano`.

Hint 2

Use the `groupby` operation in `pandas` to split an object (e.g., data) into groups and apply a calculation to each group.

Hint 3

To calculate the mean `Sales` for each `TV` category, group by `TV`, select the `Sales` column, and then calculate the mean.

Apply the same process to calculate the mean `Sales` for each `Influencer` category.

Question: What do you notice about the categorical variables? Could they be useful predictors of `Sales`?

Average sales are higher for TV when they fall into the high category. Tv seems to be a strong predictor. Influencer has a close average all the way through between the categories they fall under indicating this is not a strong predictor.

1.3.4 Remove missing data

This dataset contains rows with missing values. To correct this, drop all rows that contain missing data.

[9]:
```
# Drop rows that contain missing data and update the DataFrame.

data_final = data.dropna(axis=0)

data_final.isna().any(axis=0).sum()
```

[9]:
```
0
```

Hint 1

Use the `pandas` function that removes missing values.

Hint 2

The `dropna()` function removes missing values from an object (e.g., DataFrame).

Hint 3

Use `data.dropna(axis=0)` to drop all rows with missing values in `data`. Be sure to properly update the DataFrame.

1.3.5 Clean column names

The `ols()` function doesn't run when variable names contain a space. Check that the column names in `data` do not contain spaces and fix them, if needed.

[10]:
```
# Rename all columns in data that contain a space.

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

Hint 1

There is one column name that contains a space. Search for it in `data`.

Hint 2

The `Social Media` column name in `data` contains a space. This is not allowed in the `ols()` function.

Hint 3

Use the `rename()` function in `pandas` and use the `columns` argument to provide a new name for `Social Media`.

1.4 Step 3: Model building

1.4.1 Fit a multiple linear regression model that predicts sales

Using the independent variables of your choice, fit a multiple linear regression model that predicts `Sales` using two or more independent variables from `data`.

[12]:
```python
# 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.

summary = model.summary()

# Display the model results.

summary
```

[12]:
```
<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:                Fri, 10 Nov 2023   Prob (F-statistic):         1.63e-288
Time:                        12:45:38   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.
"""
```

Hint 1

Refer to the content that discusses model building for linear regression.

Hint 2

Use the `ols()` function imported earlier—which creates a model from a formula and DataFrame—to create an OLS model.

Hint 3

You previously learned how to specify in `ols()` that a feature is categorical.

Be sure the string names for the independent variables match the column names in `data` exactly.

Question: Which independent variables did you choose for the model, and why?

I chose to use TV and radio because they had a strong linear regression to sales. And TV had distinct averages that were not close together when it came to sales and categories.

1.4.2 Check model assumptions

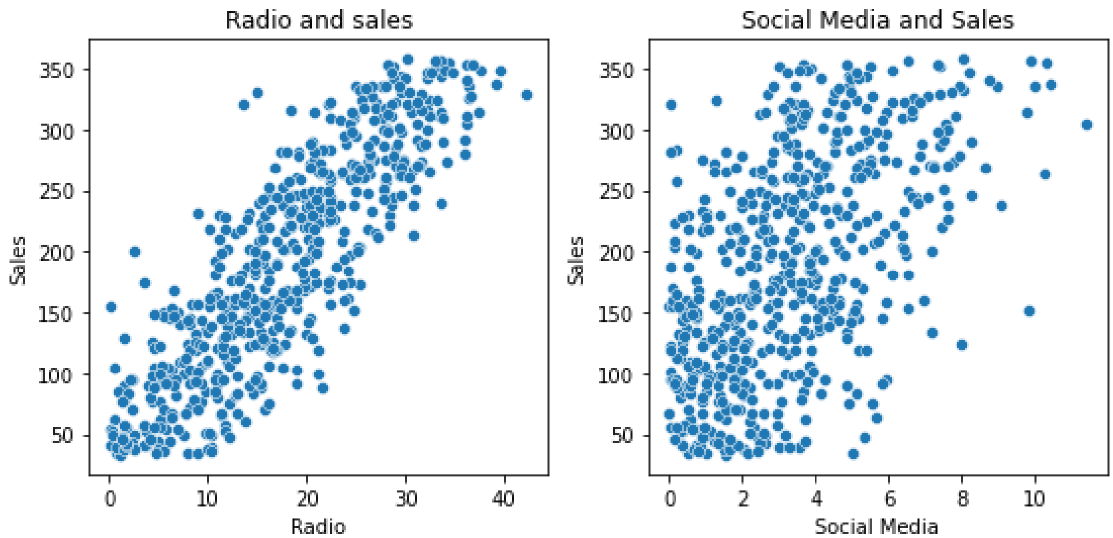
For multiple linear regression, there is an additional assumption added to the four simple linear regression assumptions: **multicollinearity**.

Check that all five multiple linear regression assumptions are upheld for your model.

1.4.3 Model assumption: Linearity

Create scatterplots comparing the continuous independent variable(s) you selected previously with `Sales` to check the linearity assumption. Use the pairplot you created earlier to verify the linearity assumption or create new scatterplots comparing the variables of interest.

[15]:
```
# Create a scatterplot for each independent variable and the dependent variable.

# Plot figure
fig, axes = plt.subplots(1, 2, figsize = (8,4))

# Scatterplot Radio and Sales
sns.scatterplot(x = data['Radio'], y = data['Sales'],ax=axes[0])

# Title 1
axes[0].set_title("Radio and sales")

# x label 1
axes[0].set_xlabel("Radio")

# Scatterplot Social media and sales
sns.scatterplot(x = data['Social_Media'], y = data['Sales'],ax=axes[1])

#Title 2
axes[1].set_title("Social Media and Sales")

# x label 2
axes[1].set_xlabel("Social Media")

# Tight layout to clean up plot space
plt.tight_layout()
```



Hint 1

Use the function in the `seaborn` library that allows you to create a scatterplot to display the values for two variables.

Hint 2

Use the `scatterplot()` function in `seaborn`.

Hint 3

Pass the independent and dependent variables in your model as the arguments for `x` and `y`, respectively, in the `scatterplot()` function. Do this for each continous independent variable in your model.

Question: Is the linearity assumption met?

Linearity is shown in Radio but the linearity is violated when it comes to social media.

1.4.4 Model assumption: Independence

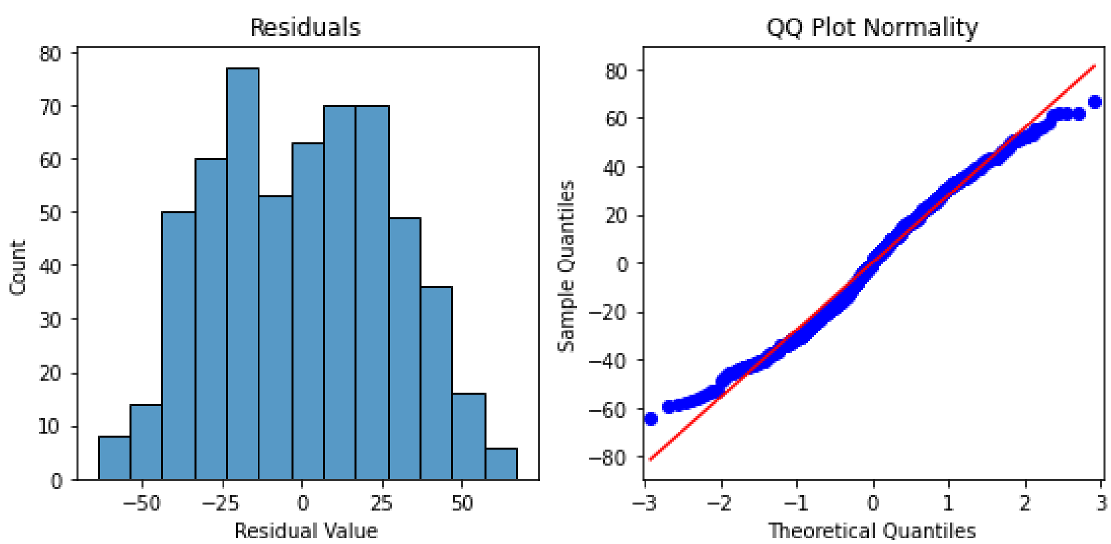
The **independent observation assumption** states that each observation in the dataset is independent. As each marketing promotion (i.e., row) is independent from one another, the independence assumption is not violated.

1.4.5 Model assumption: Normality

Create the following plots to check the **normality assumption**:

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

[18]:
```
# Calculate the residuals.

residuals = model.resid

# Plot figure
fig, axes = plt.subplots(1, 2, figsize = (8,4))

# Create a histogram with the residuals.

sns.histplot(residuals, ax=axes[0])
axes[0].set_xlabel("Residual Value")
axes[0].set_title("Residuals")
# Create a Q-Q plot of the residuals.

sm.qqplot(residuals, line='s',ax = axes[1])
axes[1].set_title("QQ Plot Normality")

plt.tight_layout()

plt.show()
```



Hint 1

Access the residuals from the fit model object.

Hint 2

Use `model.resid` to get the residuals from a fit model called `model`.

Hint 3

For the histogram, pass the residuals as the first argument in the `seaborn` `histplot()` function.

For the Q-Q plot, pass the residuals as the first argument in the `statsmodels` `qqplot()` function.

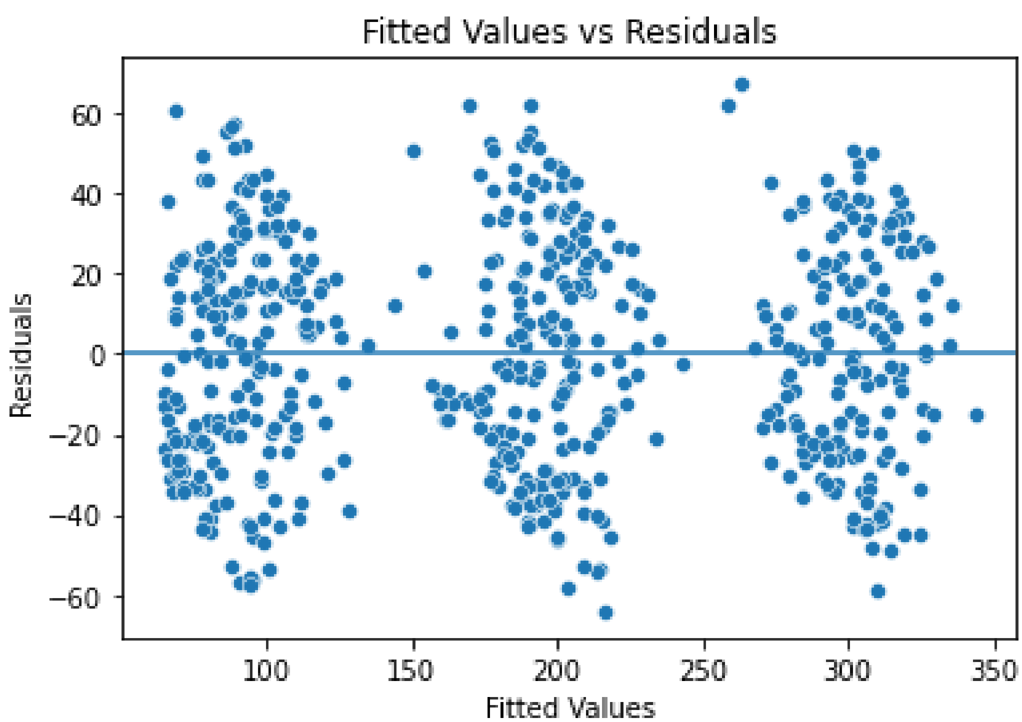
Question: Is the normality assumption met?

Both plots show there is normality within the distribution.

1.4.6 Model assumption: Constant variance

Check 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 $y = 0$.

[20]:
```
# Create a scatterplot with the fitted values from the model and the residuals.

fig = sns.scatterplot(x = model.fittedvalues, y = model.resid)

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

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

plt.show()
```



Hint 1

Access the fitted values from the model object fit earlier.

Hint 2

Use `model.fittedvalues` to get the fitted values from a fit model called `model`.

Hint 3

Call the `scatterplot()` function from the `seaborn` library and pass in the fitted values and residuals.

Add a line to a figure using the `axline()` function.

Question: Is the constant variance assumption met?

Yes, through each of the categories of TV it follows the assumption.

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.

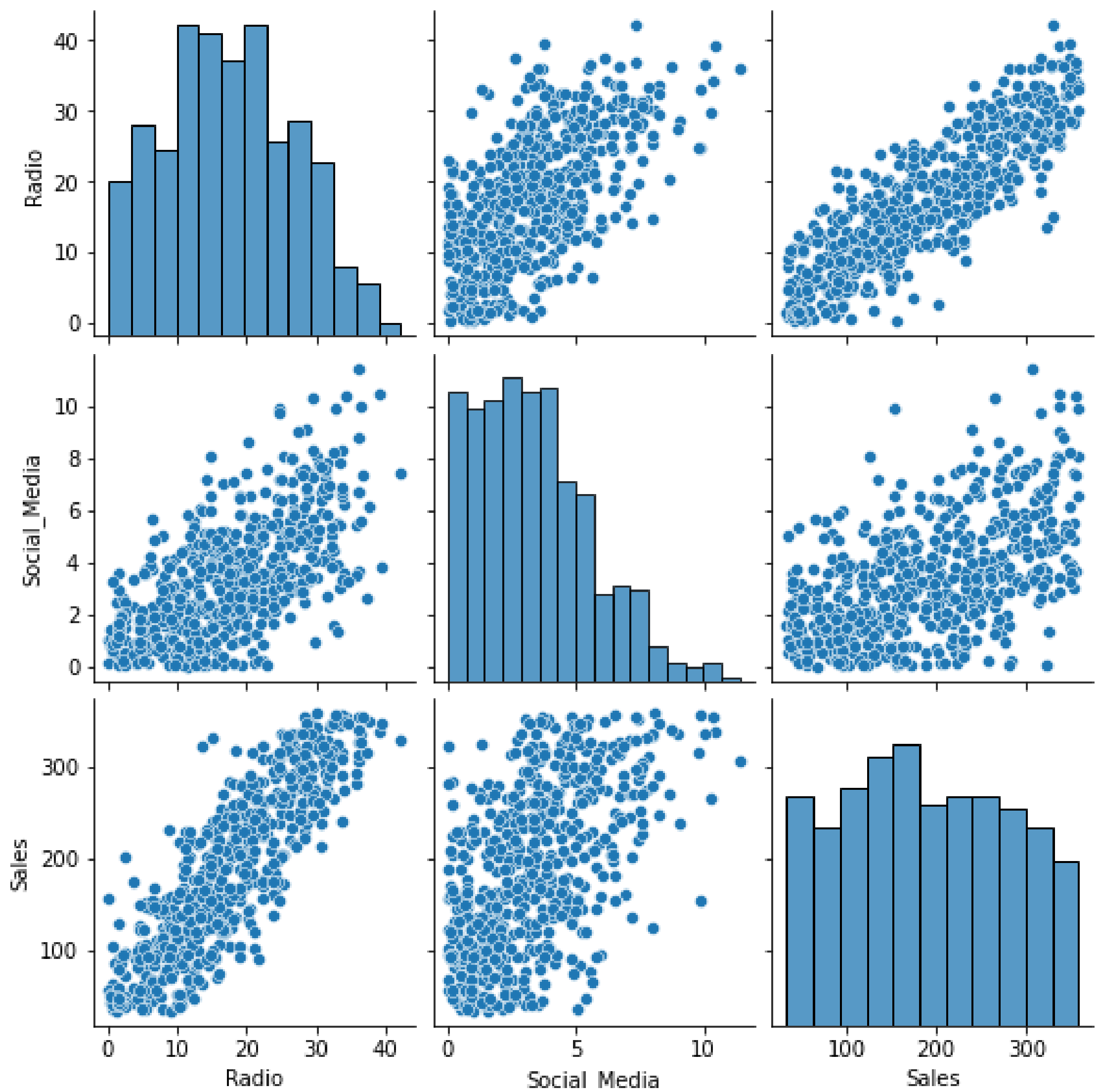
Two common ways to check for multicollinearity are to:

- Create scatterplots to show the relationship between pairs of independent variables
- Use the variance inflation factor to detect multicollinearity

Use one of these two methods to check your model's no multicollinearity assumption.

[21]:
```
# Create a pairplot of the data.

sns.pairplot(data)
```

[21]:
```
<seaborn.axisgrid.PairGrid at 0x7f359e0ea150>
```



[22]:
```
# Calculate 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'])
```

```python
# Display the VIF results.
df_vif
```

[22]:
```
                   VIF
Radio         5.170922
Social_Media  5.170922
```

Hint 1

Confirm that you previously created plots that could check the no multicollinearity assumption.

Hint 2

The `pairplot()` function applied earlier to `data` plots the relationship between all continous variables (e.g., between `Radio` and `Social Media`).

Hint 3

The `statsmodels` library has a function to calculate the variance inflation factor called `variance_inflation_factor()`.

When using this function, subset the data to only include the continous independent variables (e.g., `Radio` and `Social Media`). Refer to external tutorials on how to apply the variance inflation factor function mentioned previously.

Question 8: Is the no multicollinearity assumption met?

There is only one continous independent variable, this means there is no multicollinearity issues.

1.5 Step 4: Results and evaluation

1.5.1 Display the OLS regression results

If the model assumptions are met, you can interpret the model results accurately.

First, display the OLS regression results.

[23]:
```python
# Display the model results summary.

summary
```

[23]:
```
<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:                Fri, 10 Nov 2023   Prob (F-statistic):          1.63e-288
Time:                        12:45:38   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 is your interpretation of the model's R-squared?

The R-squared result is 90.4% of variance in sales. Which means it is the best predictor of sales.

1.5.2 Interpret model coefficients

With the model fit evaluated, you can look at the coefficient estimates and the uncertainty of these estimates.

Again, display the OLS regression results.

[24]:
```python
# Display the model results summary.

summary
```

[24]:
```
<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:                Fri, 10 Nov 2023   Prob (F-statistic):          1.63e-288
Time:                        12:45:38   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?

coefficients: Intercept - 218.5261 TV LOW - -154.2971 TV Medium - -75.3120 Radio - 2.9669

Question: How would you write the relationship between `Sales` and the independent variables as a linear equation?

Sales = 218.5261 - 154.2971* -75.3120 + 2.669

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

The coefficients all have a p-vlaue of 0.000 which means they all are statistically significant.

Question: Why is it important to interpret the beta coefficients?

Beta coefficients help describe estimates wether they are positive or negative of the independent variables on the dependent variables.

Question: What are you interested in exploring based on your model?

I'm interested in seeing further visualizations on how sales can be increased with better predictions and furthering the math to ensure the best outcomes.

Question: Do you think your model could be improved? Why or why not? How?

I think there is always room for improvment though given the model is at 95% there could always be more information to be uncovered.

1.6 Conclusion

What are the key takeaways from this lab?

The use of python makes EDA and regression models much easier and streamlined so it's a lot more digestable and understanding the information can be faster

What results can be presented from this lab?

Results are showing that with higher promo budgets we can achieve a significant increase in sales. Much higher than medium amounts and on lower amounts the averages are 154.2971 lower than higher budgets.

How would you frame your findings to external stakeholders?

Higher promotional budgets are predicted to increase sales by $75 million. There is a 95 percent chance of that increase with higher budgets for television promotions.

References Saragih, H.S. (2020). *Dummy Marketing and Sales Data*.

Congratulations! You've completed this lab. However, you may not notice a green check mark next to this item on Coursera's platform. Please continue your progress regardless of the check mark. Just click on the "save" icon at the top of this notebook to ensure your work has been logged.