Exemplar_Perform multiple linear regression

September 21, 2024

1 Exemplar: 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.

[1]:

```
# Import libraries and modules.

### YOUR CODE HERE ###

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

Load the dataset `marketing_sales_data.csv` as `data` and display the first five rows. The variables in the dataset have been adjusted to suit the objectives of this lab.

[2]:

```
# Load the data.

### YOUR CODE HERE ###

data = pd.read_csv('marketing_sales_data.csv')

# Display the first five rows.

### YOUR CODE HERE ###

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

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?

Potential reasons include:

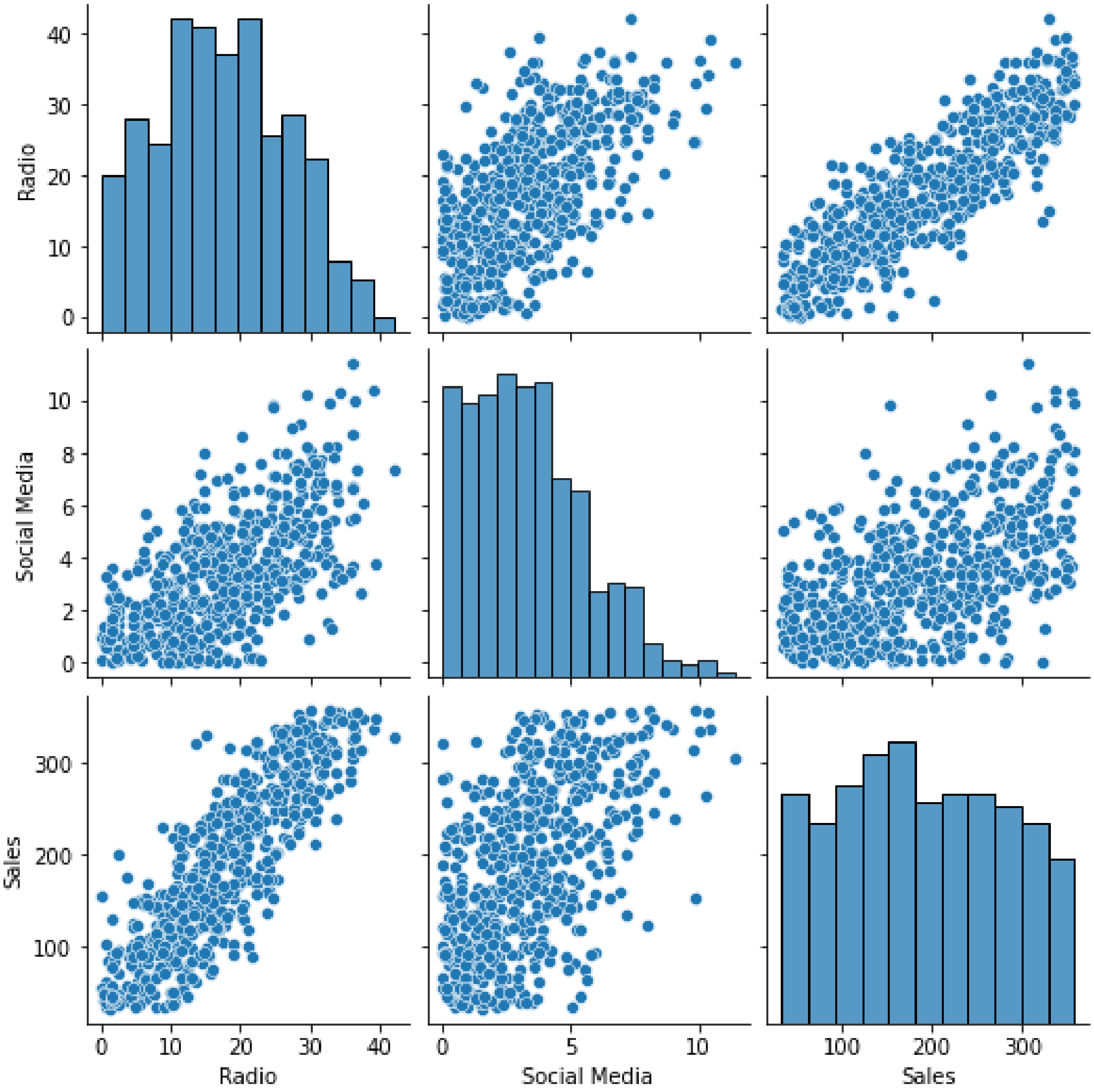
- Understanding which variables are present in the data
- Reviewing the distribution of features, such as minimum, mean, and maximum values

- Plotting the relationship between the independent and dependent variables to visualize which features have a linear relationship
- Identifying issues with the data, such as incorrect values (e.g., typos) or missing values

1.3.2 Create a pairplot of the data

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

[3]:
```
# Create a pairplot of the data.

### YOUR CODE HERE ###

sns.pairplot(data);
```



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?

Radio and Social Media both appear to have linear relationships with Sales. Given this, Radio and Social Media may be useful as independent variables in a multiple linear regression model estimating Sales.

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

[4]:

```python
# Calculate the mean sales for each TV category.

### YOUR CODE HERE ###

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

print('')

# Calculate the mean sales for each Influencer category .

### YOUR CODE HERE ###

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`?

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`.

These results can be investigated further when fitting the multiple linear regression model.

1.3.4 Remove missing data

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

[5]:
```python
# Drop rows that contain missing data and update the DataFrame.

### YOUR CODE HERE ###

data = data.dropna(axis=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.

[6]:

```python
# Rename all columns in data that contain a space.

### YOUR CODE HERE ###

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`.

[7]:

```python
# Define the OLS formula.

### YOUR CODE HERE ###

ols_formula = 'Sales ~ C(TV) + Radio'

# Create an OLS model.

### YOUR CODE HERE ###

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

# Fit the model.

### YOUR CODE HERE ###

model = OLS.fit()
```

```python
# Save the results summary.

### YOUR CODE HERE ###

model_results = model.summary()

# Display the model results.

### YOUR CODE HERE ###

model_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:                Thu, 18 May 2023   Prob (F-statistic):         1.63e-288
Time:                        21:06:21   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?

- `TV` was selected, 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.

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.

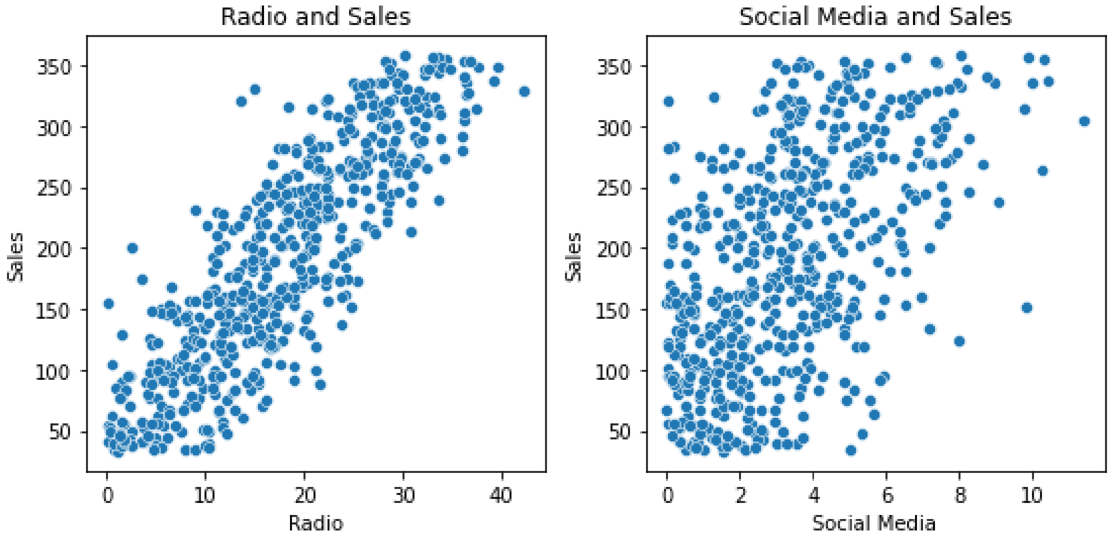
1.4.3 Model assumption: Linearity

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.

[8]:

```python
# Create a scatterplot for each independent variable and the dependent variable.

### YOUR CODE HERE ###

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

# Create a scatterplot between Radio and Sales.
```

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

# Set the title of the first plot.
axes[0].set_title("Radio and Sales")

# Create a scatterplot between Social Media and Sales.
sns.scatterplot(x = data['Social_Media'], y = data['Sales'],ax=axes[1])

# Set the title of the second plot.
axes[1].set_title("Social Media and Sales")

# Set the xlabel of the second plot.
axes[1].set_xlabel("Social Media")

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

The linearity assumption holds for `Radio`, as there is a clear linear relationship in the scatterplot between `Radio` and `Sales`. `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

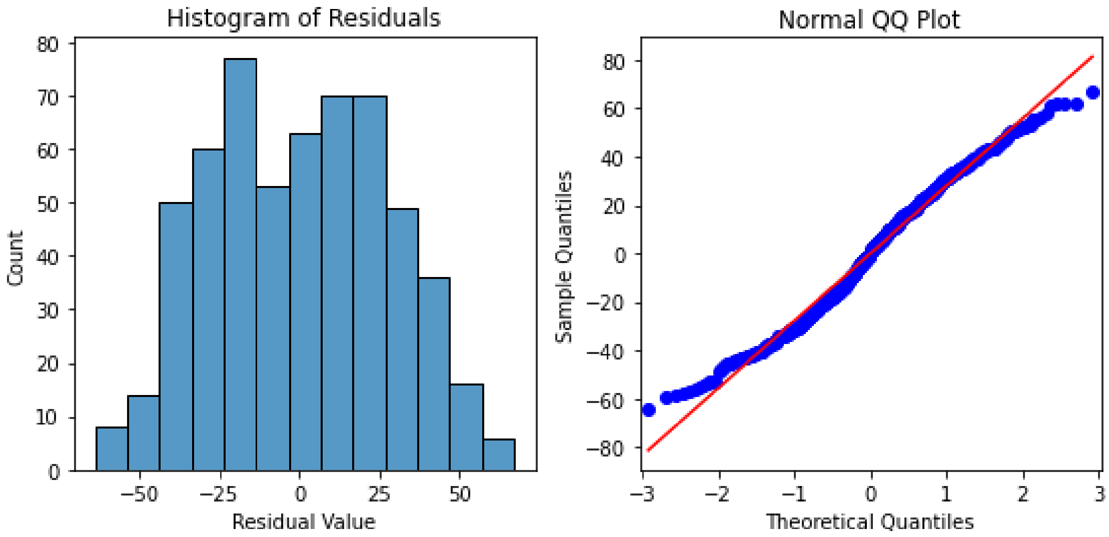
Create the following plots to check the **normality assumption**:

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

[9]:

```python
# Calculate the residuals.

### YOUR CODE HERE ###

residuals = model.resid

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

# Create a histogram with the residuals.

### YOUR CODE HERE ###

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.

### YOUR CODE HERE ###

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

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

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

# Show the plot.
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?

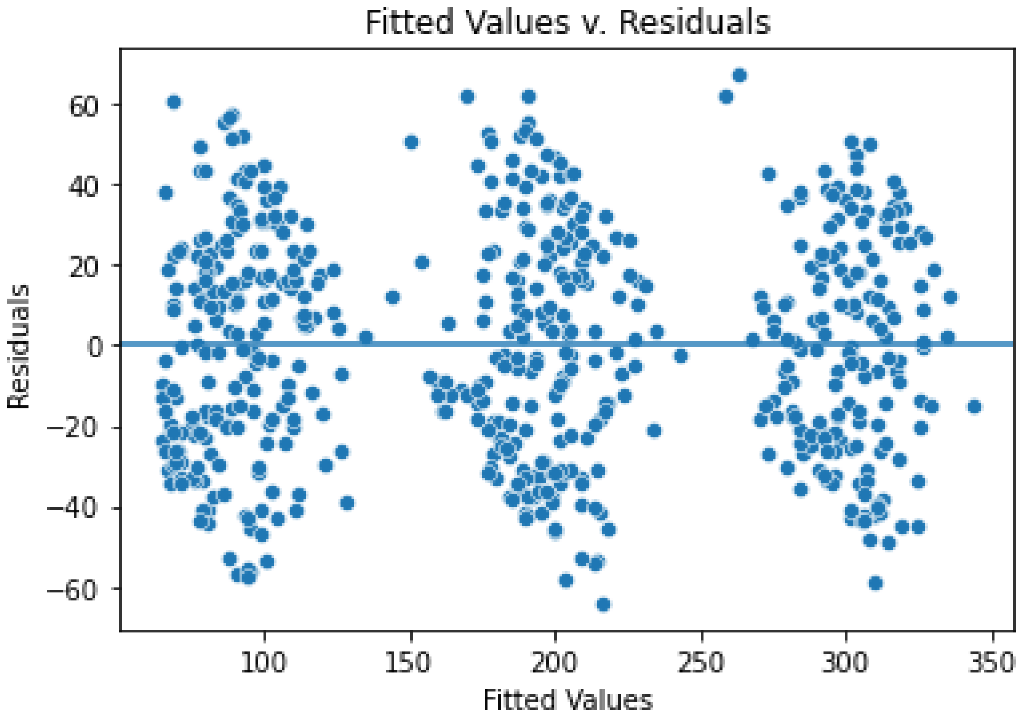
The histogram of the residuals are approximately normally distributed, which supports that the normality assumption is met for this model. The residuals in the Q-Q plot form a straight line, further supporting that this assumption is met.

1.4.6 Model assumption: Constant variance

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 $y = 0$.

[10]:

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

### YOUR CODE HERE ###

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.

### YOUR CODE HERE ###

fig.axhline(0)

# Show the plot.
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 the 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 7: Is the constant variance assumption met?

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:

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

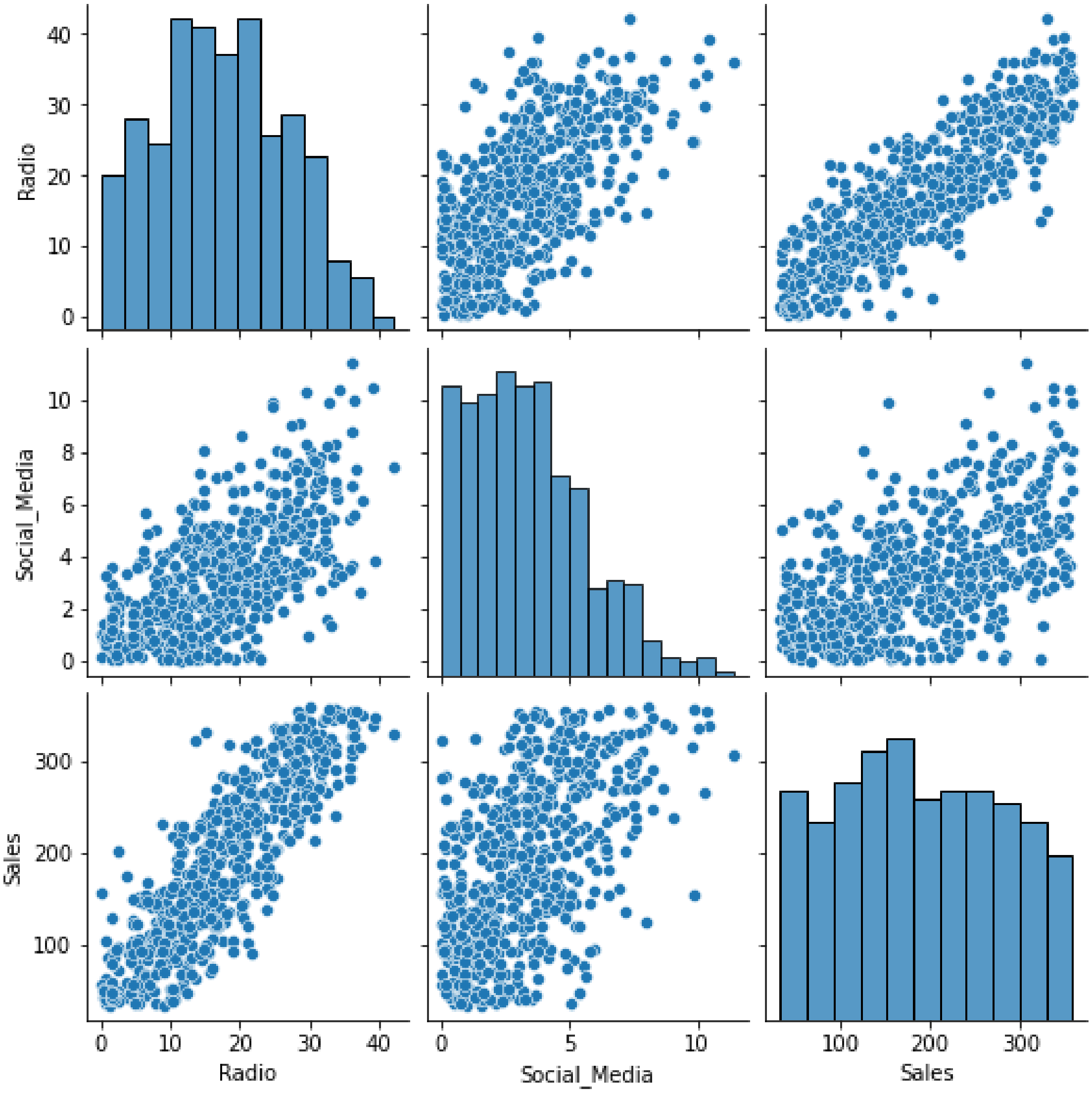
Use one of these two methods to check your model's no multicollinearity assumption.

[11]:

```
# Create a pairplot of the data.

### YOUR CODE HERE ###

sns.pairplot(data)
```

[11]:

```
<seaborn.axisgrid.PairGrid at 0x7effc5289810>
```



[12]:

```
# Calculate the variance inflation factor (optional).

### YOUR CODE HERE ###

# 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
```

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?

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. Furthermore, the variance inflation factor when both `Radio` and `Social_Media` are included in the model is 5.17 for each variable, indicating high multicollinearity.

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.

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

### YOUR CODE HERE ###

model_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:                Thu, 18 May 2023   Prob (F-statistic):          1.63e-288
Time:                        21:06:21   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?

Using `TV` and `Radio` as the independent variables results in a multiple linear regression model with $R^2 = 0.904$. In other words, 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

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

Again, display the OLS regression results.

[14]:

```
# Display the model results summary.

### YOUR CODE HERE ###

model_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:                Thu, 18 May 2023   Prob (F-statistic):          1.63e-288
Time:                        21:06:21   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:

- $\beta_0 = 218.5261$
- $\beta_{TVLow} = -154.2971$
- $\beta_{TVMedium} = -75.3120$
- $\beta_{Radio} = 2.9669$

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

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

$\text{Sales} = \beta_0 + \beta_{TVLow} * X_{TVLow} + \beta_{TVMedium} * X_{TVMedium} + \beta_{Radio} * X_{Radio}$

$\text{Sales} = 218.5261 - 154.2971 * X_{TVLow} - 75.3120 * X_{TVMedium} + 2.9669 * X_{Radio}$

Question: What is your 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.

For example, the model predicts that a `Low TV` promotion is 154.2971 lower on average compared to a `high TV` promotion given the same `Radio` promotion.

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 $p = 0.05$. The 95% confidence intervals for each coefficient should be reported when presenting results to stakeholders.

For example, there is a 95% chance that the interval $[-163.979, -144.616]$ contains the true parameter of the slope of β_{TVLow}, which is the estimated difference in promotion sales when a `Low`

`TV` promotion is chosen instead of a `High TV` promotion.

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

Beta coefficients allow you to estimate the magnitude and direction (positive or negative) of the effect of each independent variable on the dependent variable. The coefficient estimates can be converted to explainable insights, such as the connection between an increase in TV promotional budgets and sales mentioned previously.

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

Potential areas to explore include:

- Providing the business with the estimated sales given different TV promotions and radio budgets
- Additional plots to help convey the results, such as using the `seaborn` `regplot()` to plot the data with a best fit regression line

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

Given how accurate `TV` was as a predictor, the model could be improved by getting a more granular view of the `TV` promotions, such as by considering more categories or the actual `TV` promotional budgets.

Furthermore, additional variables, such as the location of the marketing campaign or the time of year, could increase model accuracy.

1.6 Considerations

What are some key takeaways that you learned from this lab?

- Multiple linear regression is a powerful tool to estimate a dependent continous variable from several independent variables.
- Exploratory data analysis is useful for selecting both numeric and categorical features for multiple linear regression.
- Fitting multiple linear regression models may require trial and error to select variables that fit an accurate model while maintaining model assumptions.

What findings would you share with others?

According to the model, high TV promotional budgets result in significantly more sales than medium and low TV promotional budgets. For example, the model predicts that a `Low TV` promotion is 154.2971 lower on average than a `high TV` promotion given the same `Radio` promotion.

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 $p = 0.05$. The 95% confidence intervals for each coefficient should be reported when presenting results to stakeholders.

For example, there is a 95% chance the interval $[-163.979, -144.616]$ contains the true parameter of the slope of β_{TVLow}, which is the estimated difference in promotion sales when a low TV promotional budget is chosen instead of a high TV promotion budget.

How would you frame your findings to stakeholders?

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.

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.