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.

In [1]:
```
# Import libraries and modules.


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

In [2]:
```
# Load the data.

df = pd.read_csv('Fk8EtlRPRyWPBLZUT-cl2Q_83304da42a7a433aa14111ee7a7c79f1_

# Display the first five rows.

df.head()
```

Out[2]:

	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

Data exploration

In [3]:
```
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 [4]:
```
df.isna().any(axis = 0).sum()
```

Out[4]: 3

drop misssing values

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

In [6]:
```
df.head()
```

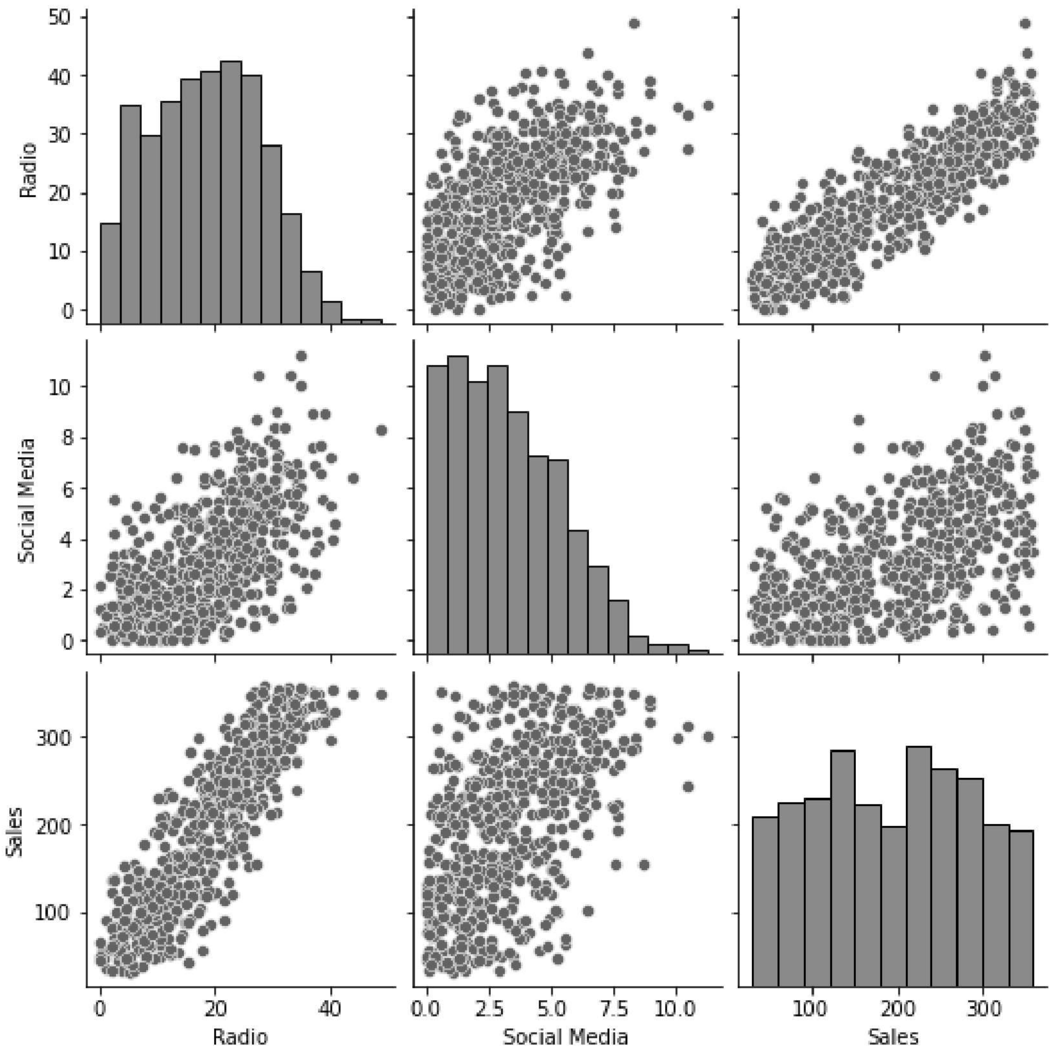
Out[6]:

	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 [7]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 569 entries, 0 to 571
Data columns (total 5 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   TV            569 non-null    object 
 1   Radio         569 non-null    float64
 2   Social Media  569 non-null    float64
 3   Influencer    569 non-null    object 
 4   Sales         569 non-null    float64
dtypes: float64(3), object(2)
memory usage: 26.7+ KB
```

In [8]:
```
# Create a pairplot of the data.

sns.pairplot(df);
```



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.

In [9]:

```
print(df.groupby('TV')['Sales'].mean())

print('')

# Calculate the mean sales for each Influencer category .

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

```
TV
High      300.529591
Low        91.716309
Medium    199.023461
Name: Sales, dtype: float64

Influencer
Macro    206.641805
Mega     180.385096
Micro    198.655080
Nano     189.742830
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.

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

In [10]:

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


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

Model building

Fit a multiple linear regression model that predicts sales

In [11]:

```
# Define the OLS formula.

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

# Create an OLS model.


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


model = OLS.fit()

# Save the results summary.


model_results = model.summary()

# Display the model results.


model_results
```

Out[11]:

OLS Regression Results

Dep. Variable:	Sales	**R-squared:**	0.904
Model:	OLS	**Adj. R-squared:**	0.904
Method:	Least Squares	**F-statistic:**	1782.
Date:	Thu, 01 Jun 2023	**Prob (F-statistic):**	1.61e-287
Time:	21:50:14	**Log-Likelihood:**	-2701.4
No. Observations:	569	**AIC:**	5411.
Df Residuals:	565	**BIC:**	5428.
Df Model:	3		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	217.6367	6.577	33.089	0.000	204.718	230.556
C(TV)[T.Low]	-152.0897	5.160	-29.474	0.000	-162.225	-141.954
C(TV)[T.Medium]	-73.4835	3.587	-20.484	0.000	-80.530	-66.437
Radio	2.8864	0.217	13.306	0.000	2.460	3.312

Omnibus:	35.219	**Durbin-Watson:**	1.949
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	13.863
Skew:	0.087	**Prob(JB):**	0.000976
Kurtosis:	2.255	**Cond. No.**	155.

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

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.

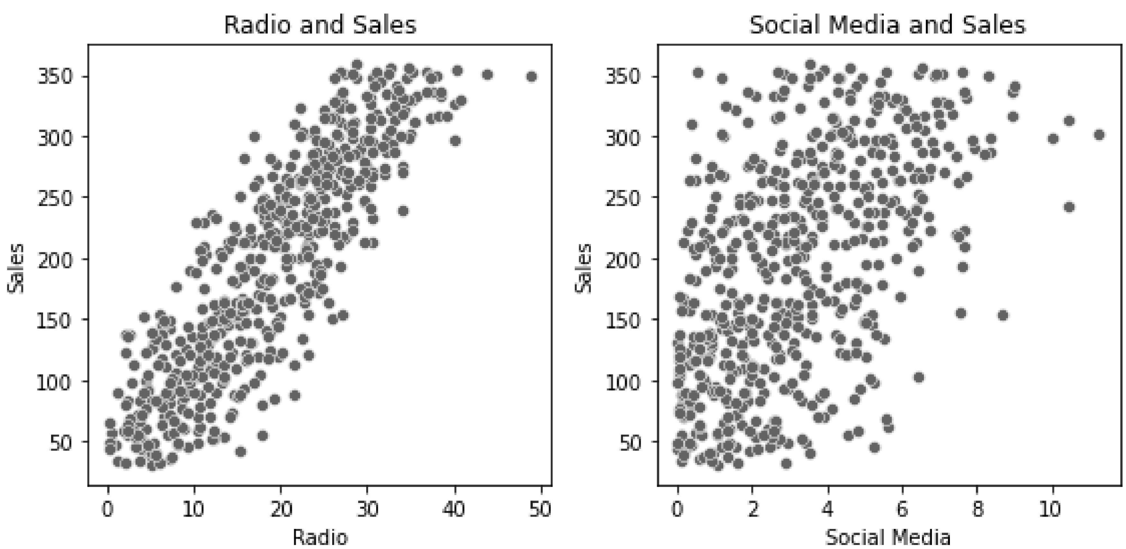
Social Media was not selected because it did not increase model performance and it was later determined to be correlated with another independent variable: Radio. Influencer was not selected because it did not show a strong relationship to Sales in the preceding analysis

Check model assumptions

Model assumption: Linearity

In [12]:

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


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

# Create a scatterplot between Radio and Sales.
sns.scatterplot(x = df['Radio'], y = df['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 = df['Social_Media'], y = df['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
plt.tight_layout()
```

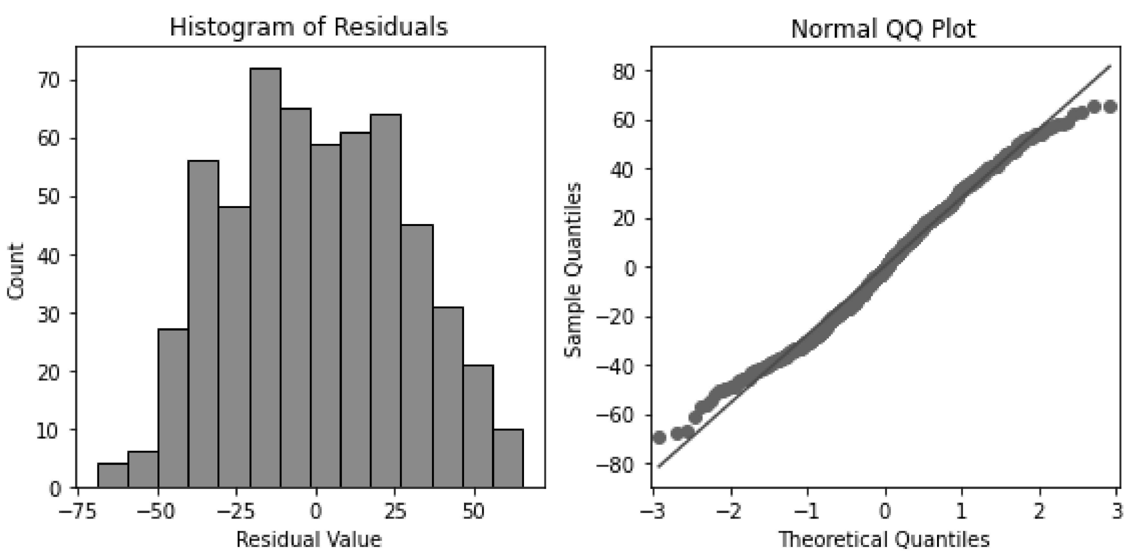


The linearity assumption holds for Radio, as there is a clear linear relationship in the scatterplot between Radio and Sales. Social Media was not included in the preceding multiple linear regression model, but it does appear to have a linear relationship with Sales

Model assumption: Normality

In [13]:

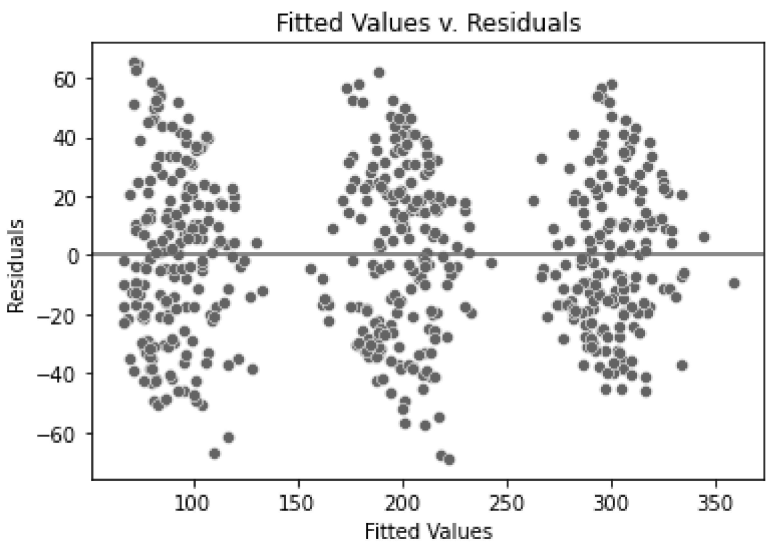
```
# Calculate the residuals.

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 QQ Plot")

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

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



Model assumption: Constant variance

In [14]:

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

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 bel

fig.axhline(0)

# Show the plot.
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.

Model assumption: No multicollinearity

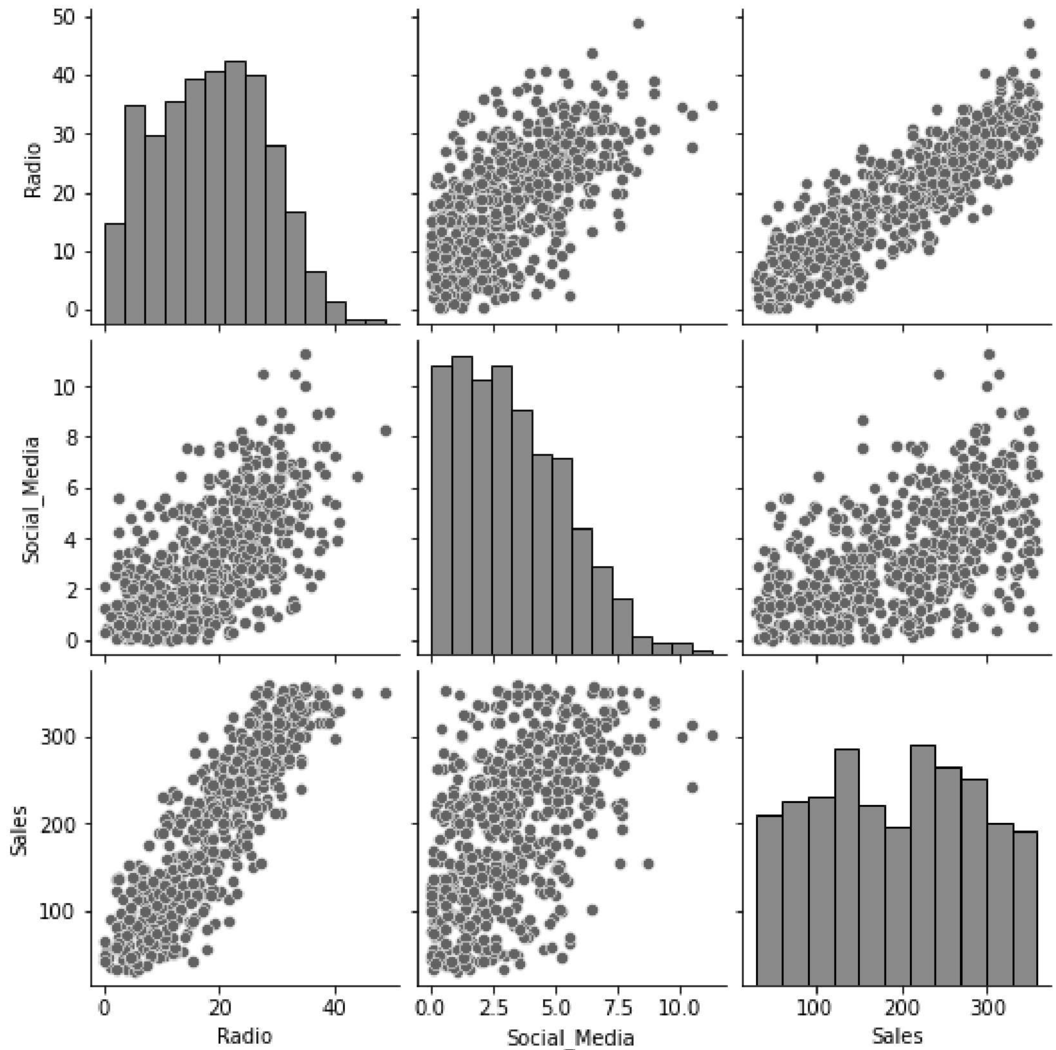
The no multicollinearity assumption states that no two independent variables (Xi and Xj) can be highly correlated with each other.

Two common ways to check for multicollinearity are to:

1. Create scatterplots to show the relationship between pairs of independent variables

In [15]:
```
# Create a pairplot of the data.

sns.pairplot(df)
```

Out[15]: <seaborn.axisgrid.PairGrid at 0x184d9036820>



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.

Display the OLS regression results

In [16]:

```
# Display the model results summary.


model_results
```

Out[16]:

OLS Regression Results

Dep. Variable:	Sales	**R-squared:**	0.904
Model:	OLS	**Adj. R-squared:**	0.904
Method:	Least Squares	**F-statistic:**	1782.
Date:	Thu, 01 Jun 2023	**Prob (F-statistic):**	1.61e-287
Time:	21:50:14	**Log-Likelihood:**	-2701.4
No. Observations:	569	**AIC:**	5411.
Df Residuals:	565	**BIC:**	5428.
Df Model:	3		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
Intercept	217.6367	6.577	33.089	0.000	204.718	230.556
C(TV)[T.Low]	-152.0897	5.160	-29.474	0.000	-162.225	-141.954
C(TV)[T.Medium]	-73.4835	3.587	-20.484	0.000	-80.530	-66.437
Radio	2.8864	0.217	13.306	0.000	2.460	3.312

Omnibus:	35.219	**Durbin-Watson:**	1.949
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	13.863
Skew:	0.087	**Prob(JB):**	0.000976
Kurtosis:	2.255	**Cond. No.**	155.

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

Result and evaluations

Using TV and Radio as the independent variables results in a multiple linear regression model with $R2$=0.904 . In other words, the model explains 90.4% of the variation in Sales. This makes the model an excellent predictor of Sales.

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

Sales=$\beta 0$+$\beta 1*X1$+$\beta 2*X2$+$\beta 3*X3$

Sales=$\beta 0$+$\beta TVLow*XTVLow$+$\beta TVMedium*XTVMedium$+$\beta Radio*XRadio$

Sales=218.5261−154.2971$*XTVLow$−75.3120$*XTVMedium$+2.9669$*XRadio$

findings

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 $75.3120 million (95 and switching from a high to low TV promotional budget reduces by$ 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]).

In [ ]:

```
1
```