Activity_Evaluate simple linear regression

November 8, 2023

1 Activity: Evaluate simple linear regression

1.1 Introduction

In this activity, you will use simple linear regression to explore the relationship between two continuous variables. To accomplish this, you will perform a complete simple linear regression analysis, which includes creating and fitting a model, checking model assumptions, analyzing model performance, interpreting model coefficients, and communicating results to stakeholders.

For this activity, you are part of an analytics team that provides insights about marketing and sales. You have been assigned to a project that focuses on the use of influencer marketing, and you would like to explore the relationship between marketing promotional budgets and sales. 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, leaders in your company will make decisions about where to focus future marketing efforts, so it is critical to have a clear understanding of the relationship between the different types of marketing and the revenue they generate.

This activity will develop your knowledge of linear regression and your skills evaluating regression results which will help prepare you for modeling to provide business recommendations in the future.

1.2 Step 1: Imports

1.2.1 Import packages

Import relevant Python libraries and packages. In this activity, you will need to use `pandas`, `pyplot` from `matplotlib`, and `seaborn`.

[11]:
```
# Import pandas, pyplot from matplotlib, and seaborn.

import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

1.2.2 Import the statsmodel module and the ols function

Import the statsmodels.api Python module using its common abbreviation, sm, along with the ols() function from statsmodels.formula.api. To complete this, you will need to write the imports as well.

[12]:
```
# Import the statsmodel module.
import statsmodels.api as sm
# Import the ols function from statsmodels.

from statsmodels.formula.api import ols
```

1.2.3 Load the dataset

Pandas was used to load the provided dataset marketing_and_sales_data_evaluate_lr.csv as data, now display the first five rows. This is a fictional dataset that was created for educational purposes. The variables in the dataset have been kept as is to suit the objectives of this activity. 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_and_sales_data_evaluate_lr.csv')

# Display the first five rows.

data.head(5)
```

[3]:

	TV	Radio	Social_Media	Sales
0	16.0	6.566231	2.907983	54.732757
1	13.0	9.237765	2.409567	46.677897
2	41.0	15.886446	2.913410	150.177829
3	83.0	30.020028	6.922304	298.246340
4	15.0	8.437408	1.405998	56.594181

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 promotion budget (in millions of dollars) * Social media promotion budget (in millions of dollars) * Radio promotion budget (in millions of dollars) * Sales

(in millions of dollars)

Each row corresponds to an independent marketing promotion where the business invests in `TV`, `Social_Media`, and `Radio` promotions to increase `Sales`.

The business would like to determine which feature most strongly predicts `Sales` so they have a better understanding of what promotions they should invest in in the future. To accomplish this, you'll construct a simple linear regression model that predicts sales using a single independent variable.

Question: What are some reasons for conducting an EDA before constructing a simple linear regression model?

Perfoming EDA will help me better understand underlying information that may not be present at first glance, and help to ensure accuracy within the regression model.

1.3.2 Explore the data size

Calculate the number of rows and columns in the data.

[5]:
```
# Display the shape of the data as a tuple (rows, columns).

data.shape
```

[5]:
```
(4572, 4)
```

Hint 1

There is an attribute of a pandas DataFrame that returns the dimension of the DataFrame.

Hint 2

The `shape` attribute of a DataFrame returns a tuple with the array dimensions.

Hint 3

Use `data.shape`, which returns a tuple with the number of rows and columns.

1.3.3 Explore the independent variables

There are three continuous independent variables: `TV`, `Radio`, and `Social_Media`. To understand how heavily the business invests in each promotion type, use `describe()` to generate descriptive statistics for these three variables.

[13]:
```
# Generate descriptive statistics about TV, Radio, and Social_Media.

data[['TV','Radio','Social_Media']].describe()
```

[13]:

	TV	Radio	Social_Media
count	4562.000000	4568.000000	4566.000000
mean	54.066857	18.160356	3.323956

```
std     26.125054   9.676958   2.212670
min     10.000000   0.000684   0.000031
25%     32.000000  10.525957   1.527849
50%     53.000000  17.859513   3.055565
75%     77.000000  25.649730   4.807558
max    100.000000  48.871161  13.981662
```

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4572 entries, 0 to 4571
Data columns (total 4 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   TV            4562 non-null   float64
 1   Radio         4568 non-null   float64
 2   Social_Media  4566 non-null   float64
 3   Sales         4566 non-null   float64
dtypes: float64(4)
memory usage: 143.0 KB
```

Hint 1

Subset `data` to only include the columns of interest.

Hint 2

Select the columns of interest using `data[['TV','Radio','Social_Media']]`.

Hint 3

Apply `describe()` to the data subset.

1.3.4 Explore the dependent variable

Before fitting the model, ensure the `Sales` for each promotion (i.e., row) is present. If the `Sales` in a row is missing, that row isn't of much value to the simple linear regression model.

Display the percentage of missing values in the `Sales` column in the DataFrame `data`.

[14]:
```
# Calculate the average missing rate in the sales column.

missing_sales = data.Sales.isna().mean()

# Convert the missing_sales from a decimal to a percentage and round to 2␣
↪decimal place.
missing_sales = round(missing_sales*100 , 2)

# Display the results (missing_sales must be converted to a string to be␣
↪concatenated in the print statement).
```

```
print('Percentage of promotions missing Sales: ' + str(missing_sales) + '%')
```

```
Percentage of promotions missing Sales: 0.13%
```

Question: What do you observe about the percentage of missing values in the `Sales` column?

There is 0.13% of rows that are missing values in Sales.

1.3.5 Remove the missing data

Remove all rows in the data from which `Sales` is missing.

[15]:
```
# Subset the data to include rows where Sales is present.

data = data.dropna(subset = ['Sales'], axis = 0)
```

Hint 1

Refer to the content about removing missing values from a DataFrame.

Hint 2

The `dropna()` function may be helpful.

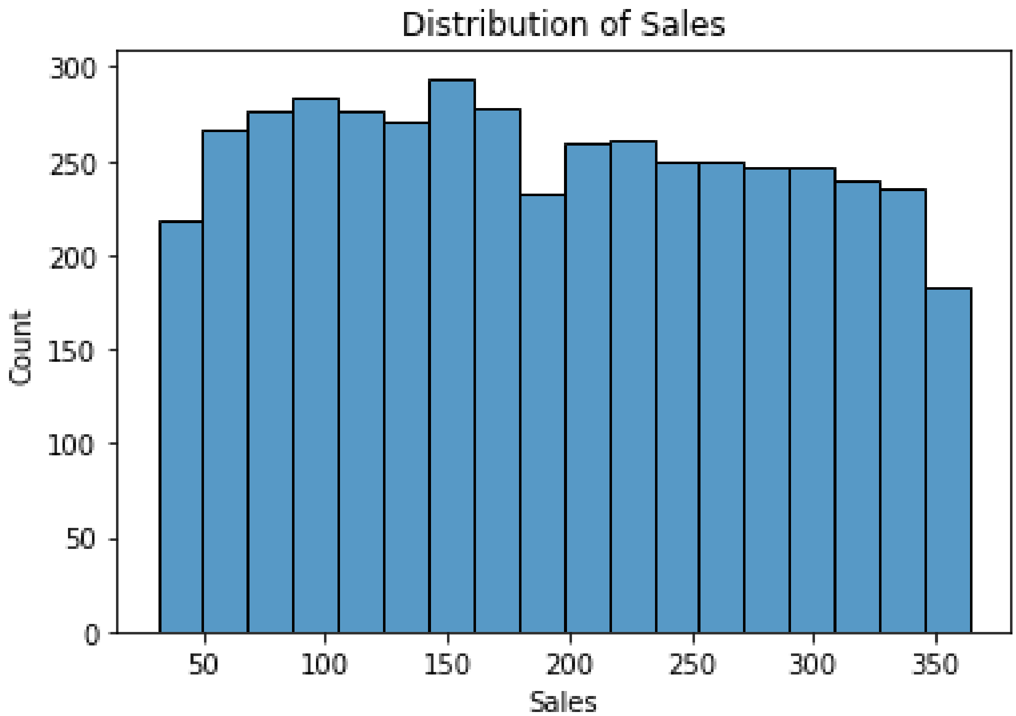
Hint 3

Apply `dropna()` to `data` and use the `subset` and `axis` arguments to drop rows where `Sales` is missing.

1.3.6 Visualize the sales distribution

Create a histogram to visualize the distribution of `Sales`.

[16]:
```
# Create a histogram of the Sales.

fig = sns.histplot(data['Sales'])
# Add a title
fig.set_title('Distribution of Sales');
```



Hint 1

Use the function in the `seaborn` library that allows you to create a histogram.

Hint 2

Call the `histplot()` function from the `seaborn` library and pass in the `Sales` column as the argument.

Hint 3

To get a specific column from a DataFrame, use a pair of single square brackets and place the name of the column, as a string, in the brackets. Be sure that the spelling, including case, matches the data exactly.

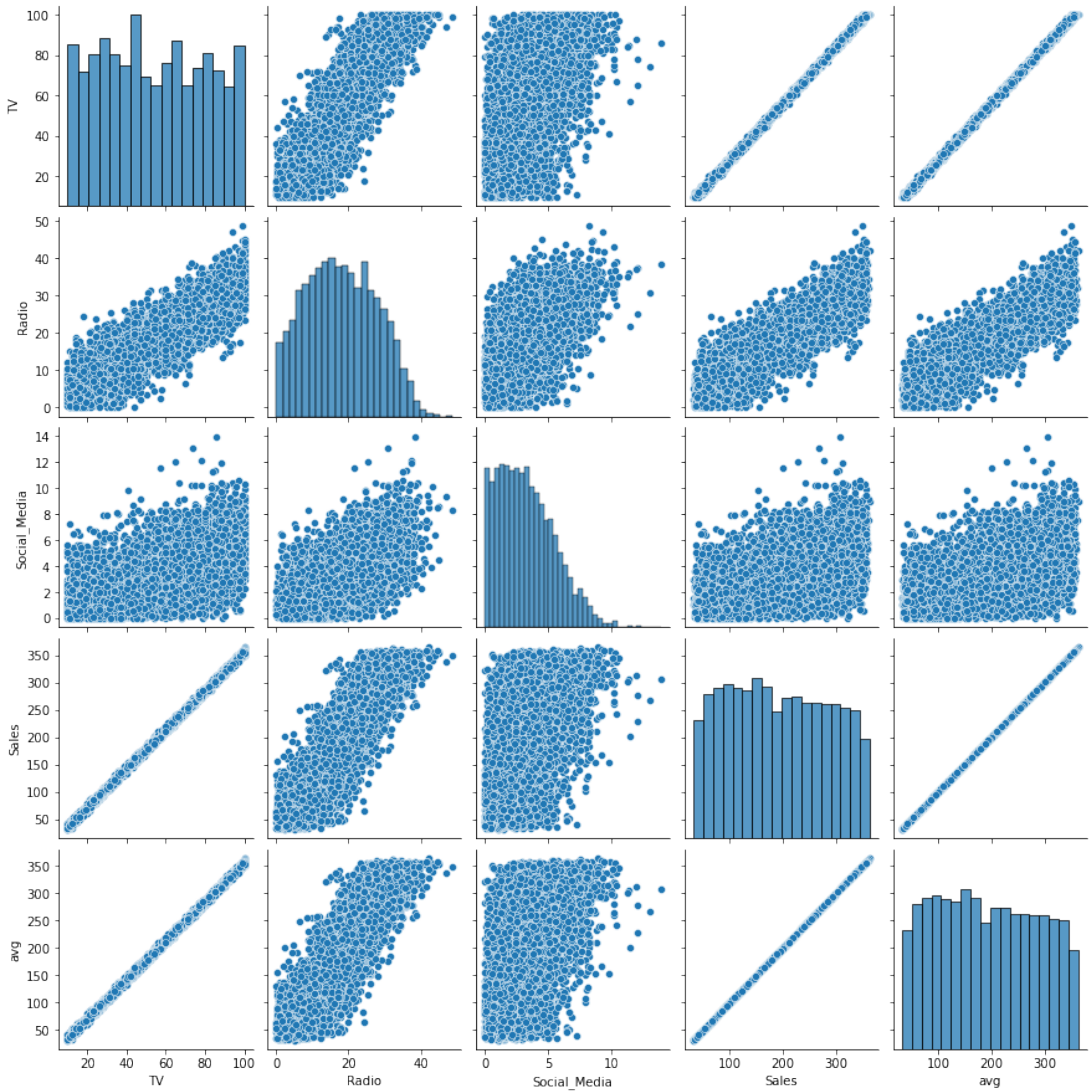
Question: What do you observe about the distribution of `Sales` from the preceding histogram?

Sales follows a pattern of being equally distributed between 25 and 350.

1.4 Step 3: Model building

Create a pairplot to visualize the relationships between pairs of variables in the data. You will use this to visually determine which variable has the strongest linear relationship with `Sales`. This will help you select the X variable for the simple linear regression.

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

sns.pairplot(data);
```



Hint 1

Refer to the video where creating a pairplot is demonstrated.

Hint 2

Use the function in the `seaborn` library that allows you to create a pairplot that shows 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 variable did you select for X? Why?

TV and sales have a clear straight line showing there is a strong linear regression. Stronger than radio and sales.

1.4.1 Build and fit the model

Replace the comment with the correct code. Use the variable you chose for `X` for building the model.

[20]:
```python
# Define the OLS formula.

ols_formula = "Sales ~ TV"

# Create an OLS model.

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

# Fit the model.

model = OLS.fit()

# Save the results summary.

model_results = model.summary()

# Display the model results.

model_results
```

[20]:
```
<class 'statsmodels.iolib.summary.Summary'>
"""
                            OLS Regression Results
==============================================================================
Dep. Variable:                  Sales   R-squared:                       0.999
Model:                            OLS   Adj. R-squared:                  0.999
Method:                 Least Squares   F-statistic:                 4.527e+06
Date:                Wed, 08 Nov 2023   Prob (F-statistic):               0.00
Time:                        20:06:27   Log-Likelihood:                -11393.
No. Observations:                4556   AIC:                         2.279e+04
Df Residuals:                    4554   BIC:                         2.280e+04
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept     -0.1263      0.101     -1.257      0.209      -0.323       0.071
TV             3.5614      0.002   2127.776      0.000       3.558       3.565
```

```
==============================================================================
Omnibus:                        0.051   Durbin-Watson:                   2.002
Prob(Omnibus):                  0.975   Jarque-Bera (JB):                0.030
Skew:                           0.001   Prob(JB):                        0.985
Kurtosis:                       3.012   Cond. No.                         138.
==============================================================================

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

Hint 1

Refer to the video where an OLS model is defined and fit.

Hint 2

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

Hint 3

Replace the `X` in `'Sales ~ X'` with the independent feature you determined has the strongest linear relationship with `Sales`. Be sure the string name for `X` exactly matches the column's name in `data`.

Hint 4

Obtain the model results summary using `model.summary()` and save it. Be sure to fit the model before saving the results summary.

1.4.2 Check model assumptions

To justify using simple linear regression, check that the four linear regression assumptions are not violated. These assumptions are:

- Linearity
- Independent Observations
- Normality
- Homoscedasticity

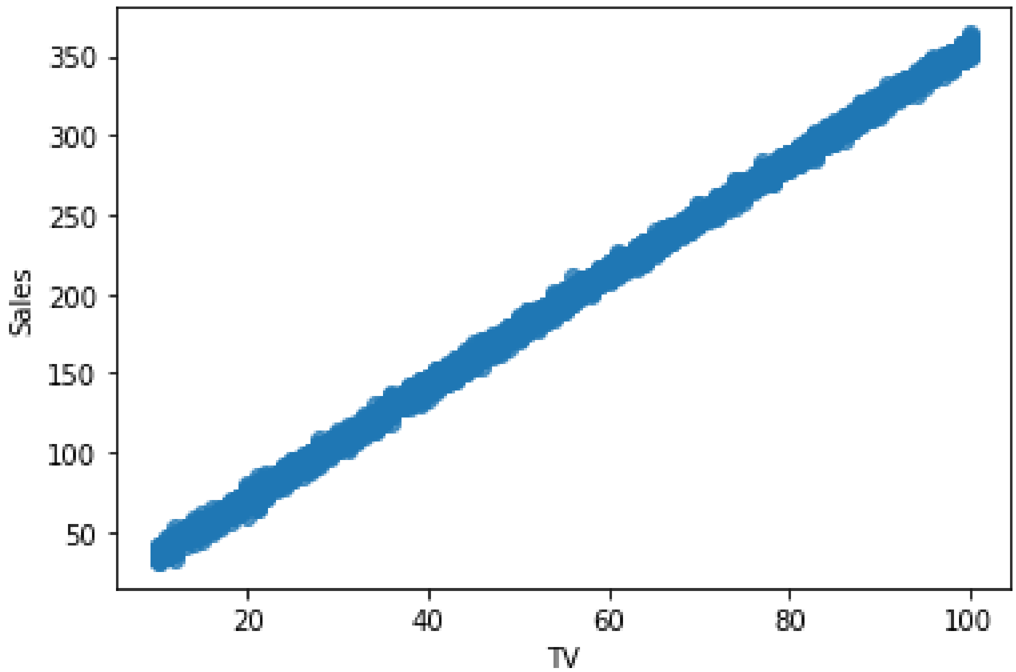
1.4.3 Model assumption: Linearity

The linearity assumption requires a linear relationship between the independent and dependent variables. Check this assumption by creating a scatterplot comparing the independent variable with the dependent variable.

Create a scatterplot comparing the X variable you selected with the dependent variable.

[21]:
```
# Create a scatterplot comparing X and Sales (Y).

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

[21]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f504be68bd0>
```



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 X and Y variables you chose for your simple linear regression as the arguments for `x` and `y`, respectively, in the `scatterplot()` function.

QUESTION: Is the linearity assumption met?

The linearity is met.

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.

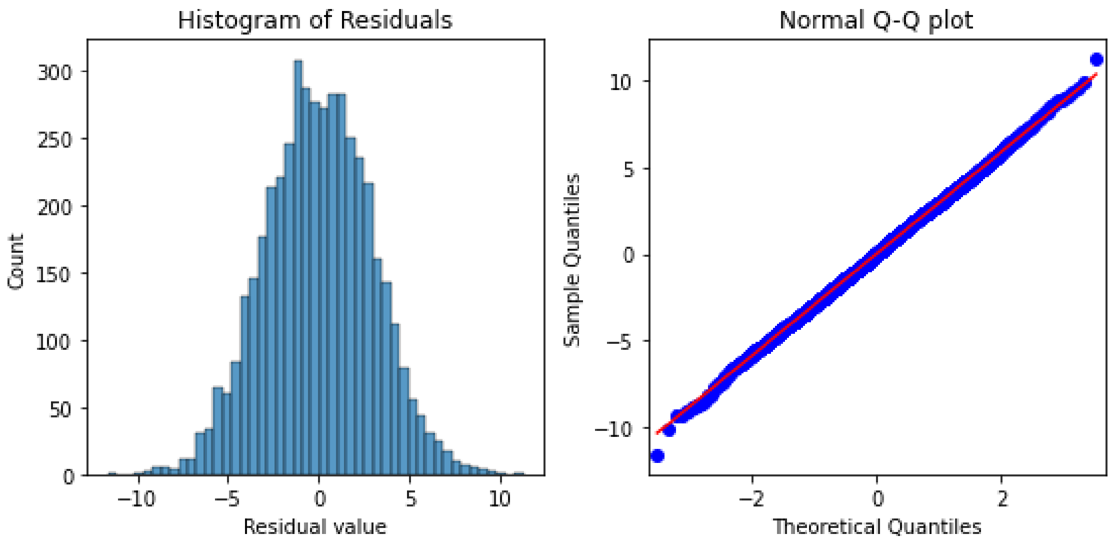
1.4.5 Model assumption: Normality

The normality assumption states that the errors are normally distributed.

Create two plots to check this assumption:

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

[26]:
```
# Calculate the residuals.

residuals = model.resid

# Create a 1x2 plot figures.
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 Q-Q 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 the fit 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?

Yes, both show that they follow assumptions of normality.

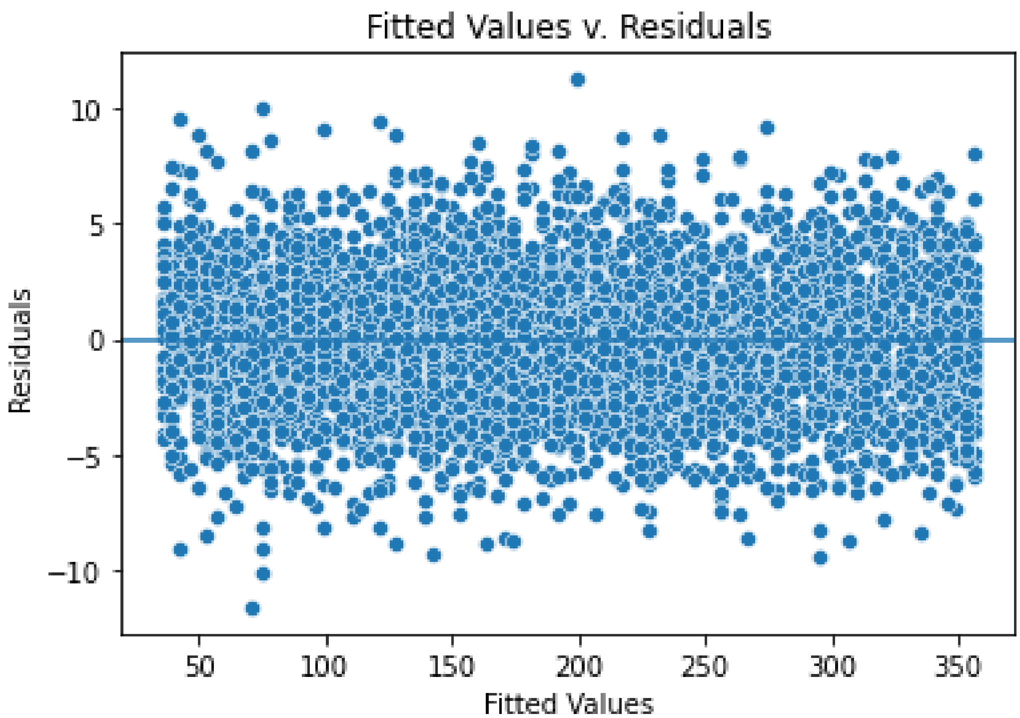
1.4.6 Model assumption: Homoscedasticity

The **homoscedasticity (constant variance) assumption** is that the residuals have a constant variance for all values of `X`.

Check that this 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 $y = 0$.

[32]:
```
# Create a scatterplot with the fitted values from the model and the residuals.

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.

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.

Hint 3

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

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

QUESTION: Is the homoscedasticity assumption met?

The result is a random cloud which means this does meet the assumptions of homoscedasticity

1.5 Step 4: Results and evaluation

1.5.1 Display the OLS regression results

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

Display the OLS regression results from the fitted model object, which includes information about the dataset, model fit, and coefficients.

[33]:
```
# Display the model_results defined previously.

model_results
```

[33]:
```
<class 'statsmodels.iolib.summary.Summary'>
"""
                            OLS Regression Results
==============================================================================
Dep. Variable:                  Sales   R-squared:                       0.999
Model:                            OLS   Adj. R-squared:                  0.999
Method:                 Least Squares   F-statistic:                 4.527e+06
Date:                Wed, 08 Nov 2023   Prob (F-statistic):               0.00
Time:                        20:06:27   Log-Likelihood:                -11393.
No. Observations:                4556   AIC:                         2.279e+04
Df Residuals:                    4554   BIC:                         2.280e+04
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept     -0.1263      0.101     -1.257      0.209      -0.323       0.071
TV             3.5614      0.002   2127.776      0.000       3.558       3.565
==============================================================================
Omnibus:                        0.051   Durbin-Watson:                   2.002
Prob(Omnibus):                  0.975   Jarque-Bera (JB):                0.030
Skew:                           0.001   Prob(JB):                        0.985
Kurtosis:                       3.012   Cond. No.                         138.
==============================================================================

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

Question: The R-squared on the preceding output measures the proportion of variation in the dependent variable (Y) explained by the independent variable (X). What is your intepretation of the model's R-squared?

TV as X shows that a simple linear regression model and R squared results that 99% of the variation is in Sales

1.5.2 Interpret the model results

With the model fit evaluated, assess the coefficient estimates and the uncertainty of these estimates.

Question: Based on the preceding model results, what do you observe about the coefficients?

The coefficients are: intercept is -0.1263 and TV is 3.5614

Question: How would you write the relationship between X and `Sales` in the form of a linear equation?

Sales (millions) = -0.1263+3.5614 * TV (millions)

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

Beta coefficients allow me to make an estimation of the magnitude and direction either positive or negtaive of each independent variable on the dependent variable. These can be used to make insights.

1.5.3 Measure the uncertainty of the coefficient estimates

Model coefficients are estimated. This means there is an amount of uncertainty in the estimate. A p-value and 95% confidence interval are provided with each coefficient to quantify the uncertainty for that coefficient estimate.

Display the model results again.

[34]:
```
# Display the model_results defined previously.

model_results
```

[34]:
```
<class 'statsmodels.iolib.summary.Summary'>
"""
                            OLS Regression Results
==============================================================================
Dep. Variable:                  Sales   R-squared:                       0.999
Model:                            OLS   Adj. R-squared:                  0.999
Method:                 Least Squares   F-statistic:                 4.527e+06
Date:                Wed, 08 Nov 2023   Prob (F-statistic):               0.00
Time:                        20:06:27   Log-Likelihood:                -11393.
No. Observations:                4556   AIC:                         2.279e+04
Df Residuals:                    4554   BIC:                         2.280e+04
Df Model:                           1
Covariance Type:            nonrobust
```

```
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept     -0.1263      0.101     -1.257      0.209      -0.323       0.071
TV             3.5614      0.002   2127.776      0.000       3.558       3.565
==============================================================================
Omnibus:                        0.051   Durbin-Watson:                   2.002
Prob(Omnibus):                  0.975   Jarque-Bera (JB):                0.030
Skew:                           0.001   Prob(JB):                        0.985
Kurtosis:                       3.012   Cond. No.                         138.
==============================================================================

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

Question: Based on this model, what is your interpretation of the p-value and confidence interval for the coefficient estimate of X?

TV has a p-value of 0.000 and a 95% confidence interval. This means there is a 95% chance the interval contains the true parameter value of the slope.

Question: Based on this model, what are you interested in exploring?

using TV and Radio as variables, and using visualizations to see how the work together in the formula

Question: What recommendations would you make to the leadership at your organization?

TV has the strongest linear regression and therefore the comapny should invest in a budget for a promotional campaign, the model shows there will be an estimated 3.5614 million dollars in more sales.

1.6 Considerations

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

Sales follow a pattern of equal sales but through visualization I found that TV has the strongest linear regression even stronger than Radio.

What findings would you share with others?

Though Radio has a stong linear regression TV is a better option to look more into.

How would you frame your findings to stakeholders?

TV is the strongest choice from the models predictions, its showing a 99% variation sales this means thats its predicting that TV budget alone will make a great choice for promoting.

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

Dale, D.,Droettboom, M., Firing, E., Hunter, J. (n.d.). *Matplotlib.Pyplot.Axline — Matplotlib 3.5.0 Documentation*.

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.