Activity_Run simple linear regression

November 7, 2023

1 Activity: Run simple linear regression

1.1 Introduction

As you're learning, simple linear regression is a way to model the relationship between two variables. By assessing the direction and magnitude of a relationship, data professionals are able to uncover patterns and transform large amounts of data into valuable knowledge. This enables them to make better predictions and decisions.

In this lab, you are part of an analytics team that provides insights about your company's sales and marketing practices. You have been assigned to a project that focuses on the use of influencer marketing. For this task, you will explore the relationship between your radio promotion budget and your 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, company leaders will make decisions about where to focus future marketing resources. Therefore, it is critical to provide them with a clear understanding of the relationship between types of marketing campaigns and the revenue generated as a result of this investment.

1.2 Step 1: Imports

Import relevant Python libraries and modules.

[3]:
```
# Import relevant Python libraries and modules.

import pandas as pd
import seaborn as sns
# OLS function import.
from statsmodels.formula.api import ols
```

The dataset provided is a .csv file (named `marketing_sales_data.csv`), which contains information about marketing conducted in collaboration with influencers, along with corresponding sales. Assume that the numerical variables in the data are expressed in millions of dollars. 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.

Note: This is a fictional dataset that was created for educational purposes and modified for this lab.

[4]:
```
# RUN THIS CELL TO IMPORT YOUR DATA.

### YOUR CODE HERE ###
sales = pd.read_csv("marketing_sales_data.csv")
```

Hint 1

Refer to what you learned about loading data in Python.

Hint 2

There is a function in the `pandas` library that allows you to read data from a .csv file and load the data into a DataFrame.

Hint 3

Use the `read_csv()` function from the `pandas` library.

1.3 Step 2: Data exploration

To get a sense of what the data includes, display the first 10 rows of the data.

[4]:
```
# Display the first 10 rows of the data.

sales.head(10)
```

[4]:

	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
5	High	25.561910	5.459718	Micro	261.966812
6	High	37.263819	6.886535	Nano	349.861575
7	Low	13.187256	2.766352	Macro	140.415286
8	High	29.520170	2.333157	Nano	264.592233
9	Low	3.773287	0.135074	Nano	55.674214

Hint 1

Refer to what you learned about exploring datasets in Python.

Hint 2

There is a function in the `pandas` library that allows you to get a specific number of rows from the top of a DataFrame.

Hint 3

Use the `head()` function from the `pandas` library.

Question: What do you observe about the different variables included in the data?

Observations: Influencers are classed by size of following, EX: Micro,Mega, Nano, etc. Tv is listed from Low to high, sales are accrued from promotion.

Next, to get a sense of the size of the dataset, identify the number of rows and the number of columns.

[6]:
```
# Display number of rows, number of columns.

sales.shape
```

[6]:
```
(572, 5)
```

Hint 1

Refer to what you learned about exploring datasets in Python.

Hint 2

There is a property in every DataFrame in `pandas` that gives you access to the number of rows and the number of columns as a tuple.

Hint 3

Use the `shape` property.

Question: How many rows and columns exist in the data?

572 Rows and 5 Columns

Now, check for missing values in the rows of the data. This is important because missing values are not that meaningful when modeling the relationship between two variables. To do so, begin by getting Booleans that indicate whether each value in the data is missing. Then, check both columns and rows for missing values.

[7]:
```
# Start with .isna() to get booleans indicating whether each value in the data␣
↪is missing.

sales.isna()
```

[7]:

	TV	Radio	Social Media	Influencer	Sales
0	False	False	False	False	False
1	False	False	False	False	False
2	False	False	False	False	False
3	False	False	False	False	False
4	False	False	False	False	False
..	…	…	…	…	…
567	False	False	False	False	False
568	False	False	False	False	False
569	False	False	False	False	False
570	False	False	False	False	False
571	False	False	False	False	False

```
[572 rows x 5 columns]
```

If you would like to read more about the `isna()` function, refer to its documentation in the references section of this lab.

[19]:
```
# Use .any(axis=1) to get booleans indicating whether there are any missing␣
↪values along the columns in each row.

sales.any(axis=1)
```

[19]:
```
0      True
1      True
2      True
3      True
4      True
       …
567    True
568    True
569    True
570    True
571    True
Length: 572, dtype: bool
```

If you would like to read more about the `any()` function, refer to its documentation in the references section of this lab.

[5]:
```
# Use .sum() to get the number of rows that contain missing values.

sales.isna().any(axis=1).sum()
```

[5]:
```
3
```

If you would like to read more about the `sum()` function, refer to its documentation in the references section of this lab.

Question: How many rows containing missing values?

There are 3 missing values.

Next, drop the rows that contain missing values. Data cleaning makes your data more usable for analysis and regression. Then, check to make sure that the resulting data does not contain any rows with missing values.

[7]:
```
# Use .dropna(axis=0) to indicate that you want rows which contain missing␣
↪values to be dropped. To update the DataFrame, reassign it to the result.

sales_final = sales.dropna(axis=0)
```

[8]:
```
# Start with .isna() to get booleans indicating whether each value in the data␣
↪is missing.
# Use .any(axis=1) to get booleans indicating whether there are any missing␣
↪values along the columns in each row.
# Use .sum() to get the number of rows that contain missing values


sales_final.isna().any(axis=1).sum()
```

[8]:
```
0
```

[15]:
```
sales_final.any(axis=1)
```

[15]:
```
0      True
1      True
2      True
3      True
4      True
       …
567    True
568    True
569    True
570    True
571    True
Length: 569, dtype: bool
```

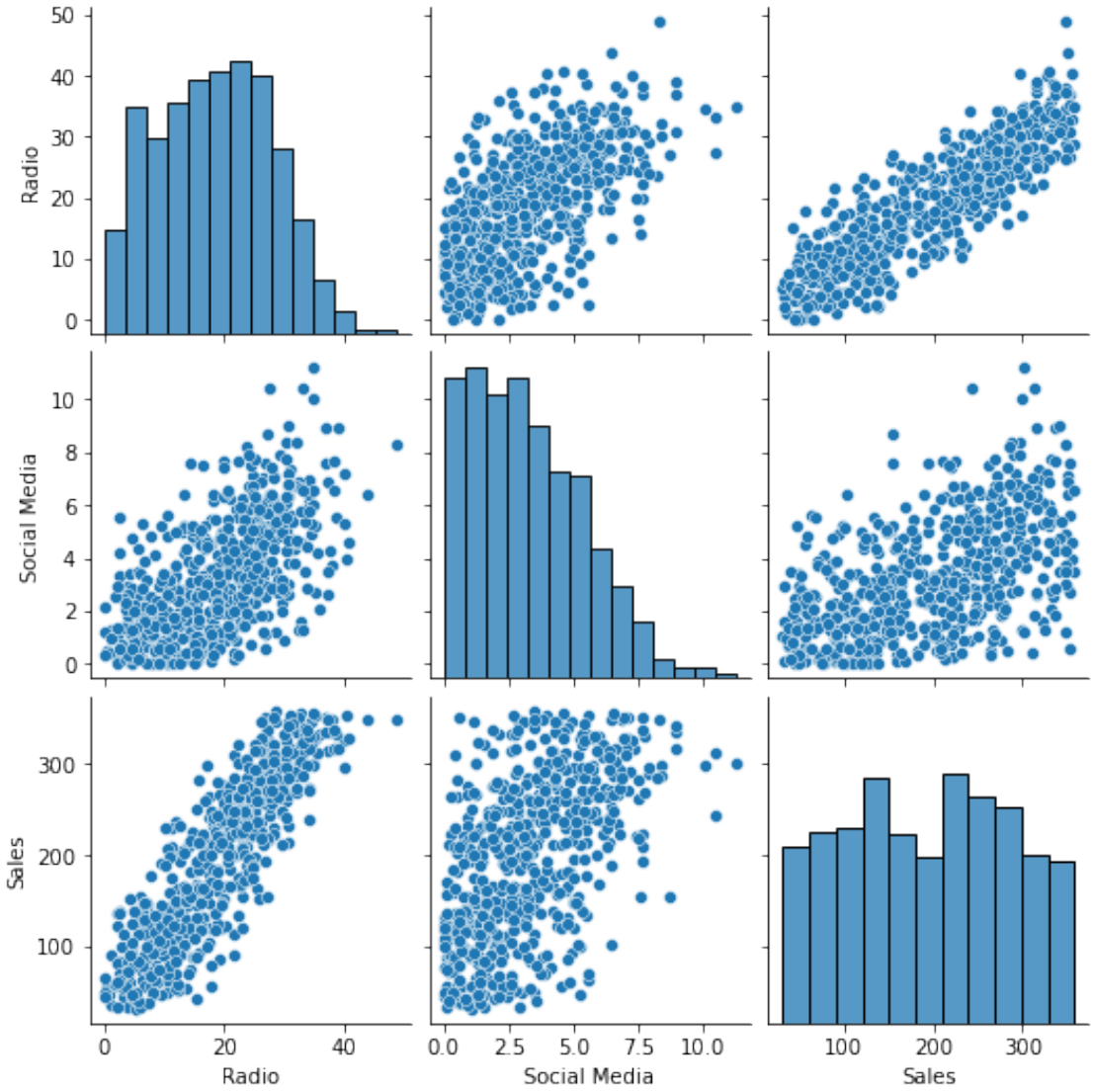
The next step for this task is checking model assumptions. To explore the relationship between radio promotion budget and sales, model the relationship using linear regression. Begin by confirming whether the model assumptions for linear regression can be made in this context.

Note: Some of the assumptions can be addressed before the model is built. These will be addressed in this section. After the model is built, you will finish checking the assumptions.

Create a plot of pairwise relationships in the data. This will help you visualize the relationships and check model assumptions.

[20]:
```
# Create plot of pairwise relationships.

sns.pairplot(sales_final)
```

[20]:
```
<seaborn.axisgrid.PairGrid at 0x7f49863b4f50>
```



Hint 1

Refer to the video section about creating a plot that shows the relationships between pairs of variables.

Hint 2

There is a function in the `seaborn` library that you can call to create a plot that shows the relationships between pairs of variables.

Hint 3

Call the `pairplot()` function from the `seaborn` library.

Question: Is the assumption of linearity met?

Yes there is the assumption of linearity between radio and Sales the scatter plot shows there's a

positive association.

Hint 1

Refer to the video section about checking model assumptions for linear regression.

Hint 2

Use the scatterplot of `Sales` over `Radio` found in the preceding plot of pairwise relationships.

Hint 3

Check the scatterplot of `Sales` over `Radio` found in the plot of pairwise relationships. If the data points cluster around a line, that indicates that the assumption of linearity is met. Alternatively, if the data points resemble a random cloud or a curve, then a linear model may not fit the data.

1.4 Step 3: Model building

Select only the columns that are needed for the model.

[9]:
```
# Select relevant columns.
# Save resulting DataFrame in a separate variable to prepare for regression.

ols_sales = sales_final[["Radio", "Sales"]]
```

Hint 1

Refer to the video about selecting multiple columns from a DataFrame.

Hint 2

Use two pairs of square brackets around the names of the columns that should be selected.

Hint 3

Make sure column names are spelled exactly as they are in the data.

Now, display the first 10 rows of the new DataFrame to better understand the data.

[37]:
```
# Display first 10 rows of the new DataFrame.

ols_sales.head(10)
```

[37]:

	Radio	Sales
0	1.218354	90.054222
1	14.949791	222.741668
2	10.377258	102.774790
3	26.469274	328.239378
4	36.876302	351.807328
5	25.561910	261.966812
6	37.263819	349.861575
7	13.187256	140.415286
8	29.520170	264.592233

```
9    3.773287    55.674214
```

Hint 1

Refer to the video about displaying contents of a DataFrame.

Hint 2

There is a function in the `pandas` library that allows you to display the first n number of rows of a DataFrame, where n is a number of your choice.

Hint 3

Call the `head()` function from the `pandas` library and pass in the number of rows from the top that you want to display.

Next, write the linear regression formula for modeling the relationship between the two variables of interest.

[38]:
```
# Write the linear regression formula.
# Save it in a variable.

ols_formula = "Sales ~ Radio"
```

Hint 1

Refer to the video section where model building for linear regression is discussed.

Hint 2

Save the formula as string.

Hint 3

Use a tilde to separate the y variable from the x variable so that the computer understands which is which. Make sure the spelling of each variable exactly matches the corresponding column from the data.

Now, implement the ordinary least squares (OLS) approach for linear regression.

[39]:
```
# Implement OLS.

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

Hint 1

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

Hint 2

There is a function from the `statsmodels` library that can be called to implement OLS.

Hint 3

You can call the `ols()` function from the `statsmodels` library.

Now, create a linear regression model for the data and fit the model to the data.

[40]:
```
# Fit the model to the data.
# Save the fitted model in a variable.

model = OLS.fit()
```

Hint 1

Refer to the video section where model building for linear regression is discussed.

Hint 2

There is a function from the `statsmodels` library that can be called to fit the model.

Hint 3

Call the `fit()` function from the `statsmodels` library.

1.5 Step 4: Results and evaluation

Begin by getting a summary of the results from the model.

[41]:
```
# Get summary of results.

model.summary()
```

[41]:
```
<class 'statsmodels.iolib.summary.Summary'>
"""
                            OLS Regression Results
==============================================================================
Dep. Variable:                  Sales   R-squared:                       0.757
Model:                            OLS   Adj. R-squared:                  0.757
Method:                 Least Squares   F-statistic:                     1768.
Date:                Tue, 07 Nov 2023   Prob (F-statistic):          2.07e-176
Time:                        00:06:05   Log-Likelihood:                -2966.7
No. Observations:                 569   AIC:                             5937.
Df Residuals:                     567   BIC:                             5946.
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept     41.5326      4.067     10.211      0.000      33.544      49.521
Radio          8.1733      0.194     42.048      0.000       7.791       8.555
==============================================================================
Omnibus:                        2.267   Durbin-Watson:                   1.880
Prob(Omnibus):                  0.322   Jarque-Bera (JB):                2.221
Skew:                          -0.102   Prob(JB):                        0.329
Kurtosis:                       2.772   Cond. No.                         45.7
==============================================================================
```

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

Hint 1

You may find it helpful to refer back to the video section where getting model results is discussed.

Hint 2

There is a function from the `statsmodels` library that can be called to get the summary of results from a model.

Hint 3

Call the `summary()` function from the `statsmodels` library.

Next, analyze the bottom table from the results summary. Based on the table, identify the coefficients that the model determined would generate the line of best fit. The coefficients are the y-intercept and the slope.

Question: What is the y-intercept?

The Y intercept is 41.5326.

Question: What is the slope?

The slope is 8.1733.

Question: What linear equation would you write to express the relationship between sales and radio promotion budget? Use the form of y = slope * x + y-intercept?

sales=8.1733* radio promotion budget + 41.5326

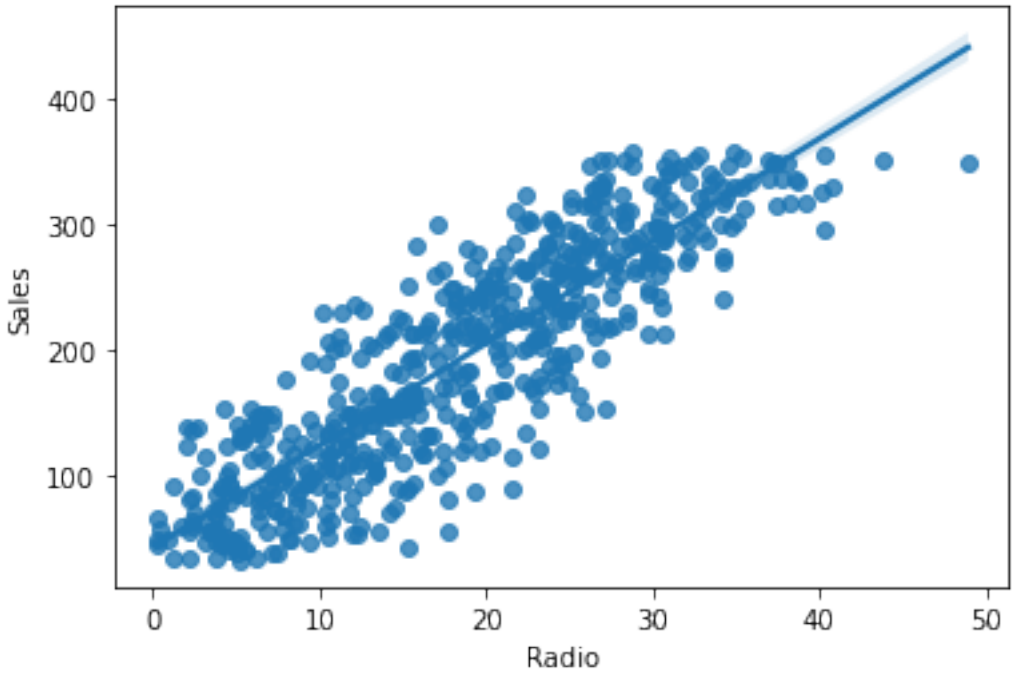
Question: What does the slope mean in this context?

If a company has a budget of 1 million dollars more for promotion, using the radio sales would have increased by 8.1733 million dollars on average.

Now that you've built the linear regression model and fit it to the data, finish checking the model assumptions. This will help confirm your findings. First, plot the OLS data with the best fit regression line.

[42]:
```
# Plot the OLS data with the best fit regression line.

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

[42]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f4986f17910>
```



Hint 1

Refer to the video about plotting data with the best fit regression line.

Hint 2

There is a function from the `seaborn` library that can be useful here.

Hint 3

Call the `regplot()` function from the `seaborn` library.

Question: What do you observe from the preceding regression plot?

It follows linear regression between sales and radio, which confirms the assumption of linearity.

Now, check the normality assumption. Get the residuals from the model.

[43]:
```
# Get the residuals from the model.

residuals = model.resid
```

Hint 1

Refer to the video about accessing residuals.

Hint 2

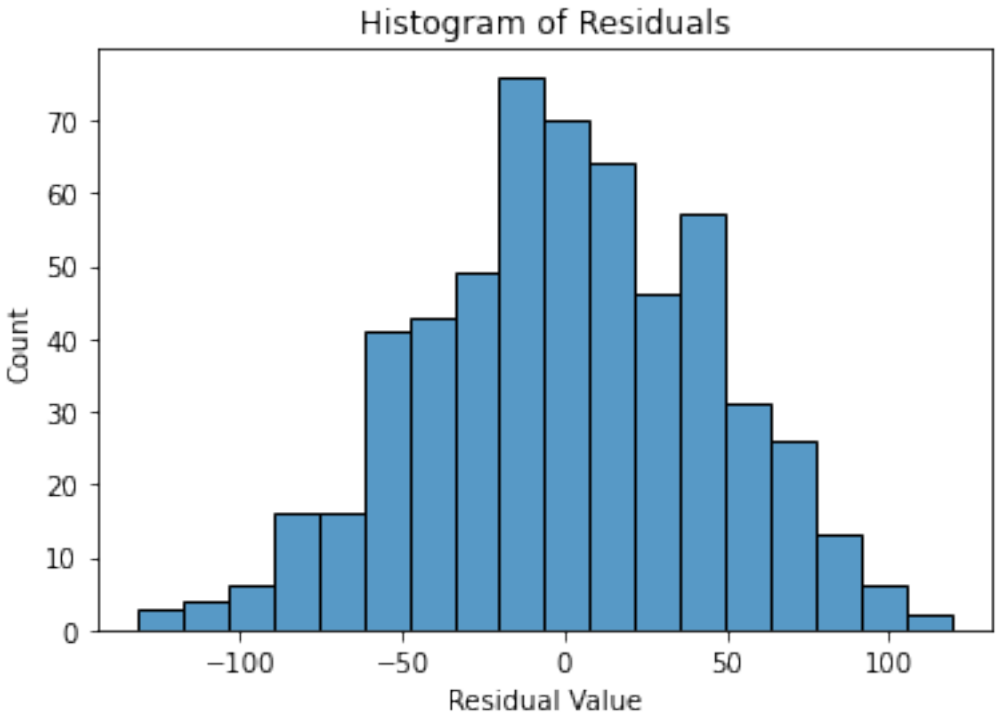
There is an attribute from the `statsmodels` library that can be called to get the residuals from a fitted model.

Hint 3

Call the `resid` attribute from the `statsmodels` library.

Now, visualize the distribution of the residuals.

[44]:
```
# Visualize the distribution of the residuals.

import matplotlib.pyplot as plt
fig = sns.histplot(residuals)
fig.set_xlabel("Residual Value")
fig.set_title("Histogram of Residuals")
plt.show()
```



Hint 1

Refer to the video about visualizing residuals.

Hint 2

There is a function from the `seaborn` library that can be called to create a histogram.

Hint 3

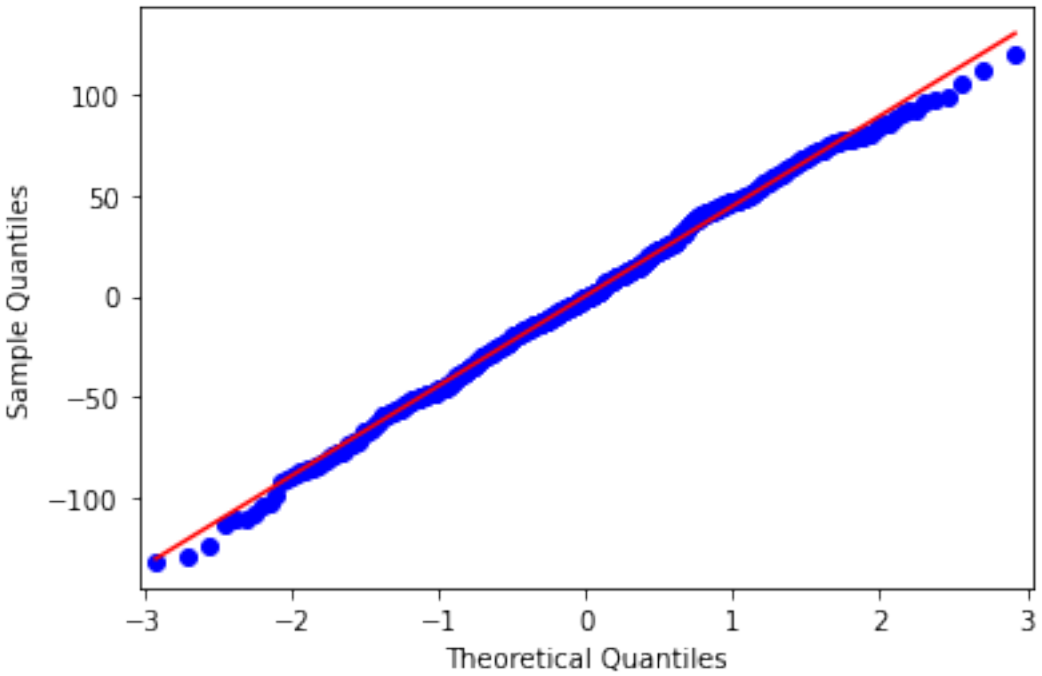
Call the `histplot()` function from the `seaborn` library.

Question: Based on the visualization, what do you observe about the distribution of the residuals?

The residual values follow a standard bell curve, showing assumption of normality is met.

Next, create a Q-Q plot to confirm the assumption of normality.

[47]:
```
# Create a Q-Q plot.
import statsmodels.api as sm
fig = sm.qqplot(model.resid, line = 's')
plt.show()
```



Hint 1

Refer to the video about creating a Q-Q plot.

Hint 2

There is a function from the `statsmodels` library that can be called to create a Q-Q plot.

Hint 3

Call the `qqplot()` function from the `statsmodels` library.

Question: Is the assumption of normality met?

Yes as It does follow the assumption of normality

Now, check the assumptions of independent observation and homoscedasticity. Start by getting the fitted values from the model.

[51]:
```
X = ols_sales["Radio"]

Fitted_values = model.predict(X)
```

[54]:
```
# Get fitted values.
X = ols_sales["Radio"]

Fitted_values = model.predict(X)
```

Hint 1

Refer to the video about calculating fitted values.

Hint 2

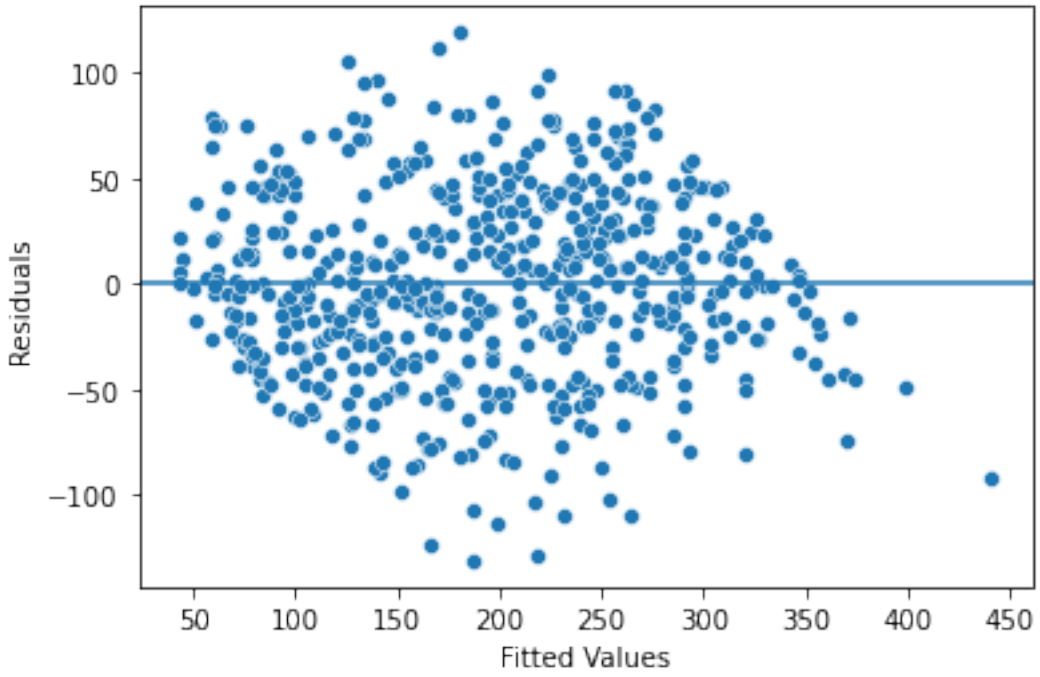
There is a function from the `statsmodels` library that can be called to calculate fitted values from the model.

Hint 3

Call the `predict()` function from the `statsmodels` library. Make sure to pass in the column from `ols_data` corresponding to the x variable.

Next, create a scatterplot of the residuals against the fitted values.

[55]:
```
# Create a scatterplot of residuals against fitted values.



fig = sns.scatterplot(x=Fitted_values, y=residuals)

# Reference line
fig.axhline(0)

#Set axis labels
fig.set_xlabel("Fitted Values")
fig.set_ylabel("Residuals")

plt.show()
```



Hint 1

Refer to the video about visualizing residuals against fitted values.

Hint 2

There is a function from the `seaborn` library that can be called to create a scatterplot.

Hint 3

Call the `scatterplot()` function from the `seaborn` library.

Question: Are the assumptions of independent observation and homoscedasticity met?

Yes as the scatter plot does not follow any specific pattern. This means the assumption is not violated and therefore homoscedasticity assumptions are met.

1.6 Considerations

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

Using visualizations and EDA I can verify linear regression to ensure the proper relationships between 2 variables to then create a model.

How would you present your findings from this lab to others?

Results from the entire analysis and the model for regression, there is a distinct relationship between radio promotion and sales. If a company uses 1 million dollars to use promotion through radio there's an average increase of 8.1733 million dollars in sales.

What summary would you provide to stakeholders?

Based on the information and data given I have performed regression analysis, my analysis concludes that there is a direct relationship with radio promotion and sales. For companies who have invested 1 million dollars on average into radio promotion they saw a sales increase of 8.1733 million dollars on average.

References

Pandas.DataFrame.Any — Pandas 1.4.3 Documentation.

Pandas.DataFrame.Isna — Pandas 1.4.3 Documentation.

Pandas.Series.Sum — Pandas 1.4.3 Documentation.

Saragih, H.S. *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.