Exemplar_Hypothesis testing with Python

January 7, 2024

1 Exemplar: Hypothesis testing with Python

1.1 Introduction

As you've been learning, analysis of variance (commonly called ANOVA) is a group of statistical techniques that test the difference of means among three or more groups. It's a powerful tool for determining whether population means are different across groups and for answering a wide range of business questions.

In this activity, you are a data professional working with historical marketing promotion data. You will use the data to run a one-way ANOVA and a post hoc ANOVA test. Then, you will communicate your results to stakeholders. These experiences will help you make more confident recommendations in a professional setting.

In your dataset, each row corresponds to an independent marketing promotion, where your business uses TV, social media, radio, and influencer promotions to increase sales. You have previously provided insights about how different promotion types affect sales; now stakeholders want to know if sales are significantly different among various TV and influencer promotion types.

To address this request, a one-way ANOVA test will enable you to determine if there is a statistically significant difference in sales among groups. This includes: * Using plots and descriptive statistics to select a categorical independent variable * Creating and fitting a linear regression model with the selected categorical independent variable * Checking model assumptions * Performing and interpreting a one-way ANOVA test * Comparing pairs of groups using an ANOVA post hoc test * Interpreting model outputs and communicating the results to nontechnical stakeholders

1.2 Step 1: Imports

Import pandas, pyplot from matplotlib, seaborn, api from statsmodels, ols from statsmodels.formula.api, and pairwise_tukeyhsd from statsmodels.stats.multicomp.

[1]:
```
# Import libraries and packages.

### 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
from statsmodels.stats.multicomp import pairwise_tukeyhsd
```

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	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

The features in the data are: * TV promotion budget (in Low, Medium, and High categories) * Social media promotion budget (in millions of dollars) * Radio promotion budget (in millions of dollars) * Sales (in millions of dollars) * Influencer size (in Mega, Macro, Nano, and Micro categories)

Question: Why is it useful to perform exploratory data analysis before constructing a linear regression model?

Potential reasons include:

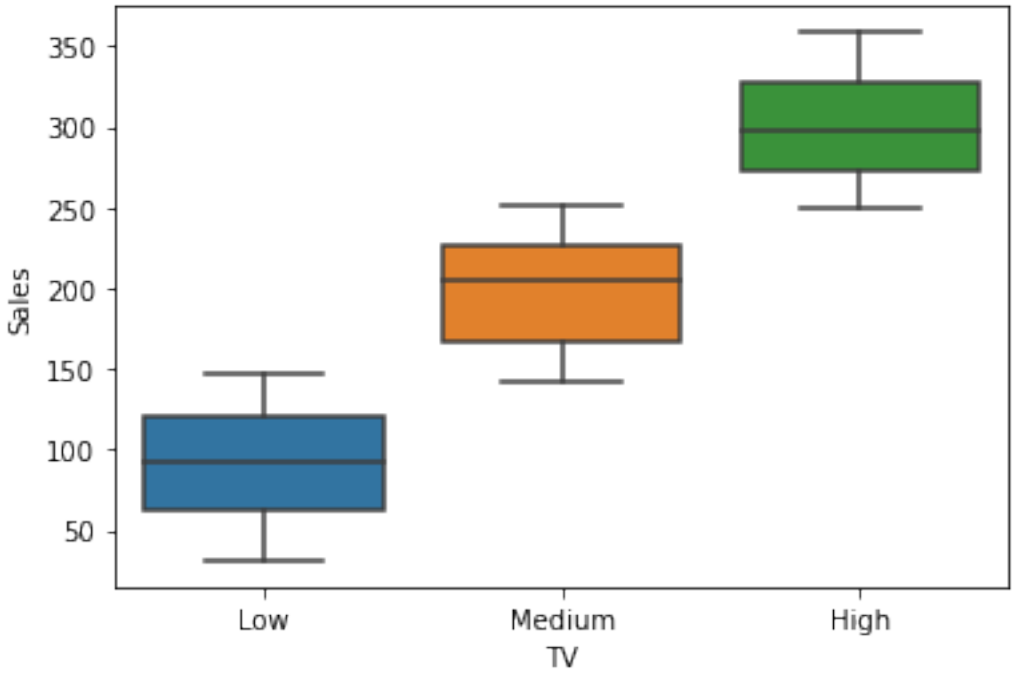
- To understand which variables are present in the data
- To consider the distribution of features, such as minimum, mean, and maximum values
- To plot the relationship between the independent and dependent variables and visualize which features have a linear relationship
- To identify issues with the data, such as incorrect or missing values.

1.3 Step 2: Data exploration

First, use a boxplot to determine how `Sales` vary based on the `TV` promotion budget category.

[3]:
```
# Create a boxplot with TV and Sales.

```

```
### YOUR CODE HERE ###

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



Hint 1

There is a function in the `seaborn` library that creates a boxplot showing the distribution of a variable across multiple groups.

Hint 2

Use the `boxplot()` function from `seaborn`.

Hint 3

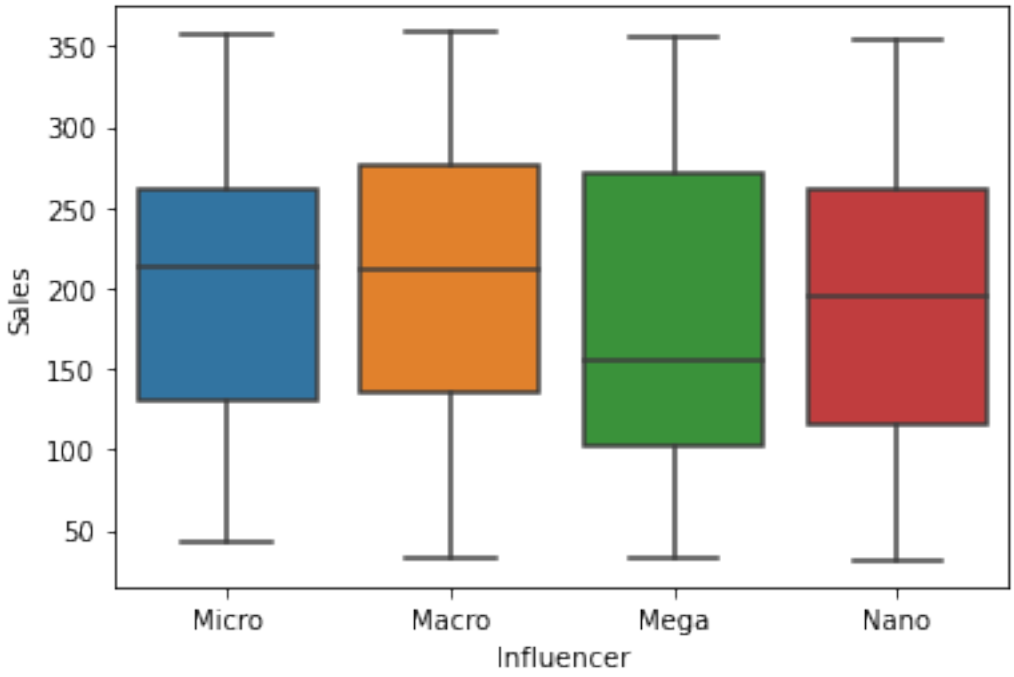
Use `TV` as the `x` argument, `Sales` as the `y` argument, and `data` as the `data` argument.

Question: Is there variation in `Sales` based off the `TV` promotion budget?

There is considerable variation in `Sales` across the `TV` groups. The significance of these differences can be tested with a one-way ANOVA.

Now, use a boxplot to determine how `Sales` vary based on the `Influencer` size category.

[4]:
```
# Create a boxplot with Influencer and Sales.

### YOUR CODE HERE ###

sns.boxplot(x = "Influencer", y = "Sales", data = data);
```



Question: Is there variation in `Sales` based off the `Influencer` size?

There is some variation in `Sales` across the `Influencer` groups, but it may not be significant.

1.3.1 Remove missing data

You may recall from prior labs that this dataset contains rows with missing values. To correct this, drop these rows. Then, confirm the data contains no missing values.

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

### YOUR CODE HERE ###

data = data.dropna(axis=0)


# Confirm the data contain no missing values.

### YOUR CODE HERE ###

data.isnull().sum(axis=0)
```

[5]:
```
TV              0
Radio           0
Social Media    0
```

```
Influencer      0
Sales           0
dtype: int64
```

Hint 1

There is a `pandas` function that removes missing values.

Hint 2

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

Hint 3

Verify the data is updated properly after the rows containing missing data are dropped.

1.4 Step 3: Model building

Fit a linear regression model that predicts `Sales` using one of the independent categorical variables in `data`. Refer to your previous code for defining and fitting a linear regression model.

[6]:
```
# Define the OLS formula.

### YOUR CODE HERE ###

ols_formula = 'Sales ~ C(TV)'

# Create an OLS model.

### YOUR CODE HERE ###

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

# Fit the model.

### YOUR CODE HERE ###

model = OLS.fit()

# Save the results summary.

### YOUR CODE HERE ###

model_results = model.summary()

# Display the model results.

### YOUR CODE HERE ###
```

```
model_results
```

[6]:
```
<class 'statsmodels.iolib.summary.Summary'>
"""
                            OLS Regression Results
==============================================================================
Dep. Variable:                  Sales   R-squared:                       0.874
Model:                            OLS   Adj. R-squared:                  0.874
Method:                 Least Squares   F-statistic:                     1971.
Date:                Tue, 25 Jul 2023   Prob (F-statistic):          8.81e-256
Time:                        21:45:48   Log-Likelihood:                -2778.9
No. Observations:                 569   AIC:                             5564.
Df Residuals:                     566   BIC:                             5577.
Df Model:                           2
Covariance Type:            nonrobust
================================================================================
===
                     coef    std err          t      P>|t|      [0.025
0.975]
--------------------------------------------------------------------------------
---
Intercept         300.5296      2.417    124.360      0.000     295.783
305.276
C(TV)[T.Low]     -208.8133      3.329    -62.720      0.000    -215.353
-202.274
C(TV)[T.Medium]  -101.5061      3.325    -30.526      0.000    -108.038
-94.975
==============================================================================
Omnibus:                      450.714   Durbin-Watson:                   2.002
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               35.763
Skew:                          -0.044   Prob(JB):                     1.71e-08
Kurtosis:                       1.775   Cond. No.                         3.86
==============================================================================

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

Hint 1

Refer to code you've written to fit linear regression models.

Hint 2

Use the `ols()` function from `statsmodels.formula.api`, which creates a model from a formula and DataFrame, to create an OLS model.

Hint 3

Use `C()` around the variable name in the ols formula to indicate a variable is categorical.

Be sure the variable string names exactly match the column names in `data`.

Question: Which categorical variable did you choose for the model? Why?

- `TV` was selected as the preceding analysis showed a strong relationship between the `TV` promotion budget and the average `Sales`.
- `Influencer` was not selected because it did not show a strong relationship to `Sales` in the analysis.

1.4.1 Check model assumptions

Now, check the four linear regression assumptions are upheld for your model.

Question: Is the linearity assumption met?

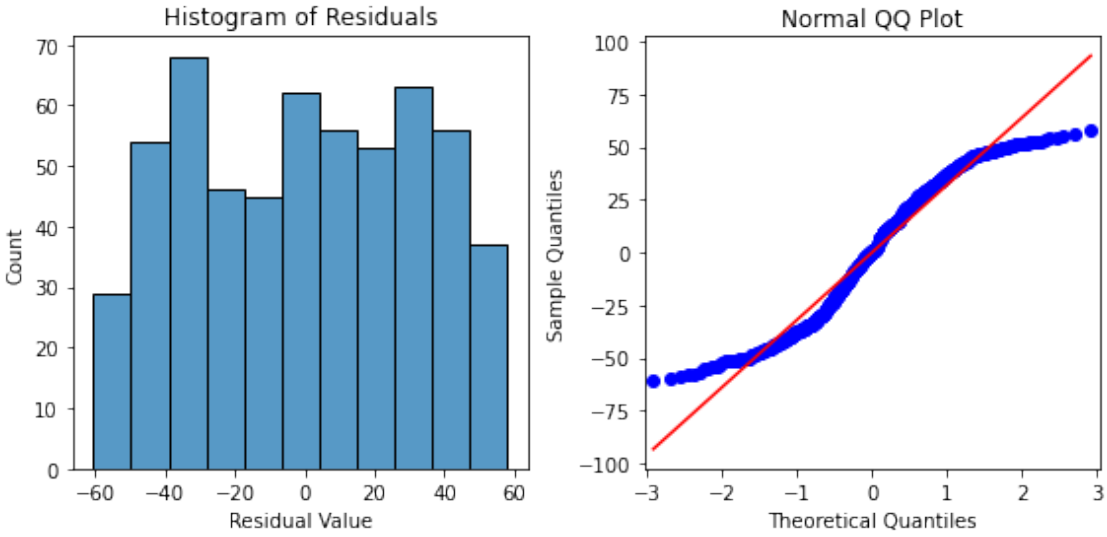
Because your model does not have any continuous independent variables, the linearity assumption is not required.

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

Next, verify that the normality assumption is upheld for the model.

[7]:
```
# 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 QQ plot of the residuals.

### YOUR CODE HERE ###
```

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

# Set the title of the QQ 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 QQ-plot, pass the residuals as the first argument in the `statsmodels` `qqplot()` function.

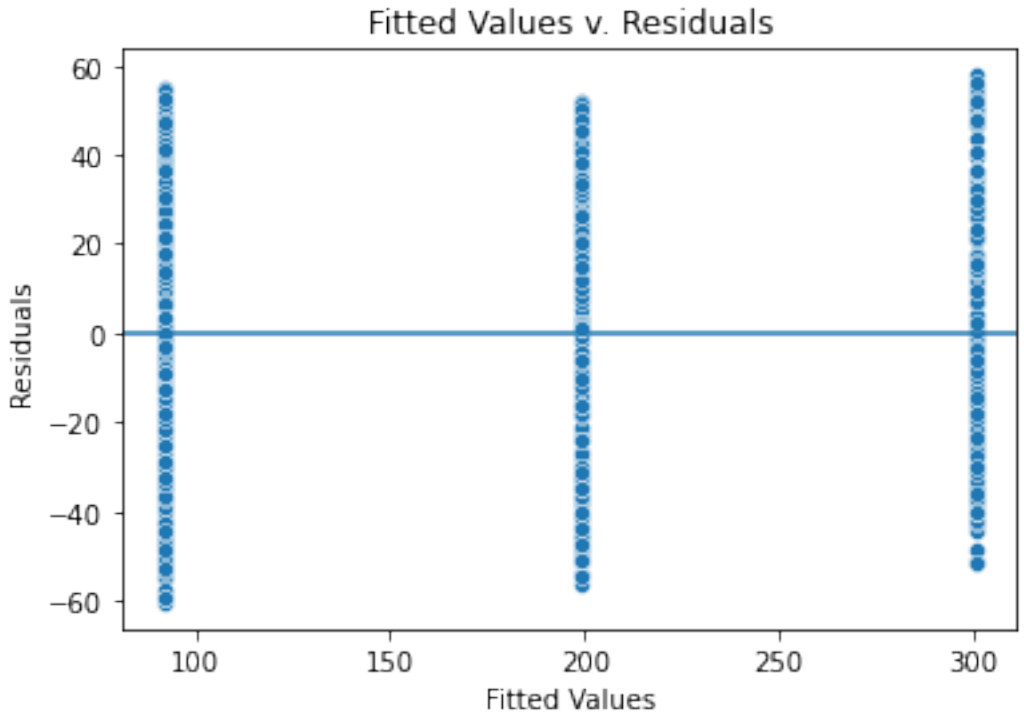
Question: Is the normality assumption met?

There is reasonable concern that the normality assumption is not met when `TV` is used as the independent variable predicting `Sales`. The normal q-q forms an 'S' that deviates off the red diagonal line, which is not desired behavior.

However, for the purpose of the lab, continue assuming the normality assumption is met.

Now, verify the constant variance (homoscedasticity) assumption is met for this model.

[8]:
```
# Create a scatter plot 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: Is the constant variance (homoscedasticity) assumption met?

The variance where there are fitted values is similarly distributed, validating that the constant variance assumption is met.

1.5 Step 4: Results and evaluation

First, display the OLS regression results.

[9]:
```
# Display the model results summary.

### YOUR CODE HERE ###

model_results
```

[9]:
```
<class 'statsmodels.iolib.summary.Summary'>
"""
                            OLS Regression Results
==============================================================================
Dep. Variable:                  Sales   R-squared:                       0.874
Model:                            OLS   Adj. R-squared:                  0.874
Method:                 Least Squares   F-statistic:                     1971.
Date:                Tue, 25 Jul 2023   Prob (F-statistic):          8.81e-256
Time:                        21:45:48   Log-Likelihood:                -2778.9
No. Observations:                 569   AIC:                             5564.
Df Residuals:                     566   BIC:                             5577.
Df Model:                           2
Covariance Type:            nonrobust
==============================================================================
===
                     coef    std err          t      P>|t|      [0.025
0.975]
------------------------------------------------------------------------------
```

```
---
Intercept          300.5296      2.417    124.360      0.000     295.783
305.276
C(TV)[T.Low]      -208.8133      3.329    -62.720      0.000    -215.353
-202.274
C(TV)[T.Medium]   -101.5061      3.325    -30.526      0.000    -108.038
-94.975
==============================================================================
Omnibus:                      450.714   Durbin-Watson:                   2.002
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               35.763
Skew:                          -0.044   Prob(JB):                     1.71e-08
Kurtosis:                       1.775   Cond. No.                         3.86
==============================================================================

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` as the independent variable results in a linear regression model with $R^2 = 0.874$. In other words, the model explains 87.4% of the variation in `Sales`. This makes the model an effective predictor of `Sales`.

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

The default `TV` category for the model is `High`, because there are coefficients for the other two `TV` categories, `Medium` and `Low`. According to the model, `Sales` with a `Medium` or `Low` `TV` category are lower on average than `Sales` with a `High` `TV` category. For example, the model predicts that a `Low` `TV` promotion would be 208.813 (in millions of dollars) lower in `Sales` on average than a `High` `TV` promotion.

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 instance, there is a 95% chance the interval $[-215.353, -202.274]$ contains the true parameter of the slope of $\beta_{TV\,Low}$, which is the estimated difference in promotion sales when a `Low` `TV` promotion is chosen instead of a `High` `TV` promotion.

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 with a more granular view of the `TV` promotions, such as additional categories or the actual `TV` promotion budgets. Further, additional variables, such as the location of the marketing campaign or the time of year, may increase model accuracy.

1.5.1 Perform a one-way ANOVA test

With the model fit, run a one-way ANOVA test to determine whether there is a statistically significant difference in `Sales` among groups.

[10]:
```
# Create an one-way ANOVA table for the fit model.

### YOUR CODE HERE ###

sm.stats.anova_lm(model, typ=2)
```

[10]:

	sum_sq	df	F	PR(>F)
C(TV)	4.052692e+06	2.0	1971.455737	8.805550e-256
Residual	5.817589e+05	566.0	NaN	NaN

Hint 1

Review what you've learned about how to perform a one-way ANOVA test.

Hint 2

There is a function in `statsmodels.api` (i.e. `sm`) that peforms an ANOVA test for a fit linear model.

Hint 3

Use the `anova_lm()` function from `sm.stats`.

Question: What are the null and alternative hypotheses for the ANOVA test?

The null hypothesis is that there is no difference in `Sales` based on the `TV` promotion budget.

The alternative hypothesis is that there is a difference in `Sales` based on the `TV` promotion budget.

Question: What is your conclusion from the one-way ANOVA test?

The F-test statistic is 1971.46 and the p-value is $8.81 * 10^{-256}$ (i.e., very small). Because the p-value is less than 0.05, you would reject the null hypothesis that there is no difference in `Sales` based on the `TV` promotion budget.

Question: What did the ANOVA test tell you?

The results of the one-way ANOVA test indicate that you can reject the null hypothesis in favor of the alternative hypothesis. There is a statistically significant difference in `Sales` among `TV` groups.

1.5.2 Perform an ANOVA post hoc test

If you have significant results from the one-way ANOVA test, you can apply ANOVA post hoc tests such as the Tukey's HSD post hoc test.

Run the Tukey's HSD post hoc test to compare if there is a significant difference between each pair of categories for TV.

[11]:
```
# Perform the Tukey's HSD post hoc test.

### YOUR CODE HERE ###

tukey_oneway = pairwise_tukeyhsd(endog = data["Sales"], groups = data["TV"])

# Display the results
tukey_oneway.summary()
```

[11]:
```
<class 'statsmodels.iolib.table.SimpleTable'>
```

Hint 1

Review what you've learned about how to perform a Tukey's HSD post hoc test.

Hint 2

Use the `pairwise_tukeyhsd()` function from `statsmodels.stats.multicomp`.

Hint 3

The `endog` argument in `pairwise_tukeyhsd` indicates which variable is being compared across groups (i.e., `Sales`). The `groups` argument in `pairwise_tukeyhsd` tells the function which variable holds the group you're interested in reviewing.

Question: What is your interpretation of the Tukey HSD test?

The first row, which compares the `High` and `Low` `TV` groups, indicates that you can reject the null hypothesis that there is no significant difference between the `Sales` of these two groups.

You can also reject the null hypotheses for the two other pairwise comparisons that compare `High` to `Medium` and `Low` to `Medium`.

Question: What did the post hoc tell you?

A post hoc test was conducted to determine which `TV` groups are different and how many are different from each other. This provides more detail than the one-way ANOVA results, which can at most determine that at least one group is different. Further, using the Tukey HSD controls for the increasing probability of incorrectly rejecting a null hypothesis from peforming multiple tests.

The results were that `Sales` is not the same between any pair of `TV` groups.

1.6 Considerations

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

- Box-plots are a helpful tool for visualizing the distribution of a variable across groups.
- One-way ANOVA can be used to determine if there are significant differences among the means of three or more groups.
- ANOVA post hoc tests provide a more detailed view of the pairwise differences between groups.

What summary would you provide to stakeholders? Consider the statistical significance of key relationships and differences in distribution.

High TV promotion budgets result in significantly more sales than both medium and low TV promotion budgets. Medium TV promotion budgets result in significantly more sales than low TV promotion budgets.

Specifically, following are estimates for the difference between the mean sales resulting from different pairs of TV promotions, as determined by the Tukey's HSD test:

- Estimated difference between the mean sales resulting from High and Low TV promotions: \$208.81 million (with 95% confidence that the exact value for this difference is between 200.99 and 216.64 million dollars).
- Estimated difference between the mean sales resulting from High and Medium TV promotions: \$101.51 million (with 95% confidence that the exact value for this difference is between 93.69 and 109.32 million dollars).
- difference between the mean sales resulting from Medium and Low TV promotions: \$107.31 million (with 95% confidence that the exact value for this difference is between 99.71 and 114.91 million dollars).

The linear regression model estimating `Sales` from `TV` had an R-squared of $0.871, making it a fairly accurate estimator. The model showed a statistically significant relationship between the `TV` promotion budget and `Sales`.

The results of the one-way ANOVA test indicate that the null hypothesis that there is no difference in Sales based on the TV promotion budget can be rejected. Through the ANOVA post hoc test, a significant difference between all pairs of TV promotions was found.

The difference in the distribution of sales across TV promotions was determined significant by both a one-way ANOVA test and a Tukey's HSD test.

Reference 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.