Activity_Hypothesis testing with Python

November 20, 2023

1 Activity: 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.

[13]:

```
# 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
```

Pandas was used to load the dataset marketing_sales_data.csv as data, now display the first five rows. The variables in the dataset have been adjusted to suit the objectives of this lab. 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.

[14]:
```
# RUN THIS CELL TO IMPORT YOUR DATA.

### YOUR CODE HERE ###
data = pd.read_csv('marketing_sales_data.csv')

# Display the first five rows.
data.head()
### YOUR CODE HERE ###
```

[14]:
```
       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)

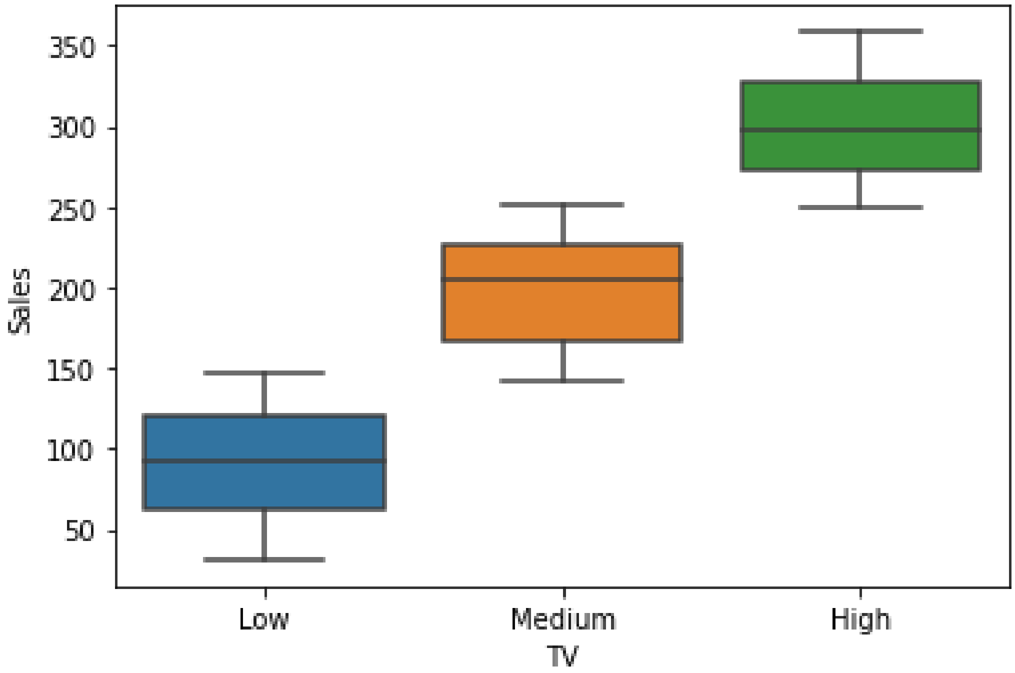
Question: Why is it useful to perform exploratory data analysis before constructing a linear regression model?

[- To know if there's any issues witht the data - To know which relationship has a linear relationship - To know which data is present - To know the mean, and etc.

1.3 Step 2: Data exploration

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

[15]:
```
# Create a boxplot with TV and Sales.
sns.boxplot(x = "TV", y = "Sales", data = data);
### YOUR CODE HERE ###
```



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

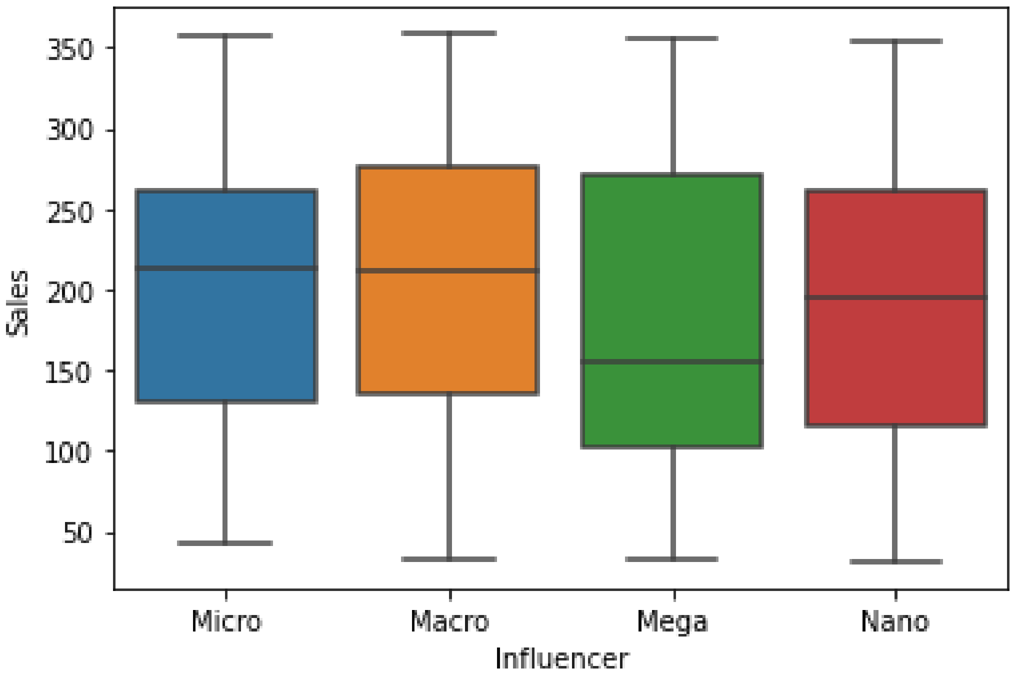
Use `TV` as the `x` argument, `Sales` as the `y` argument, and `data` as the `data` argument.

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

yes huge difference

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

[16]:
```
# Create a boxplot with Influencer and Sales.
sns.boxplot(x = "Influencer", y = "Sales", data = data);
### YOUR CODE HERE ###
```



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

There's some variation, but not so much difference.

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.

[17]:

```
# Drop rows that contain missing data and update the DataFrame.
data = data.dropna()
### YOUR CODE HERE ###


# Confirm the data contains no missing values.
data.isna().sum()
### YOUR CODE HERE ###
```

[17]:

```
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.

[19]:

```
# Define the OLS formula.
import statsmodels.api as sm
from statsmodels.formula.api import ols
### YOUR CODE HERE ###



# Create an OLS model.
model = ols(formula = "Sales ~ C(TV)", data = data).fit()
### YOUR CODE HERE ###



# Fit the model.

### YOUR CODE HERE ###



# Save the results summary.
model_results = model.summary()

### YOUR CODE HERE ###



# Display the model results.
model_results
### YOUR CODE HERE ###
```

[19]:

```
<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:                Thu, 09 Nov 2023   Prob (F-statistic):          8.81e-256
Time:                        11:48:24   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 because it did show a strong relationship to Sales in the analysis. 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?

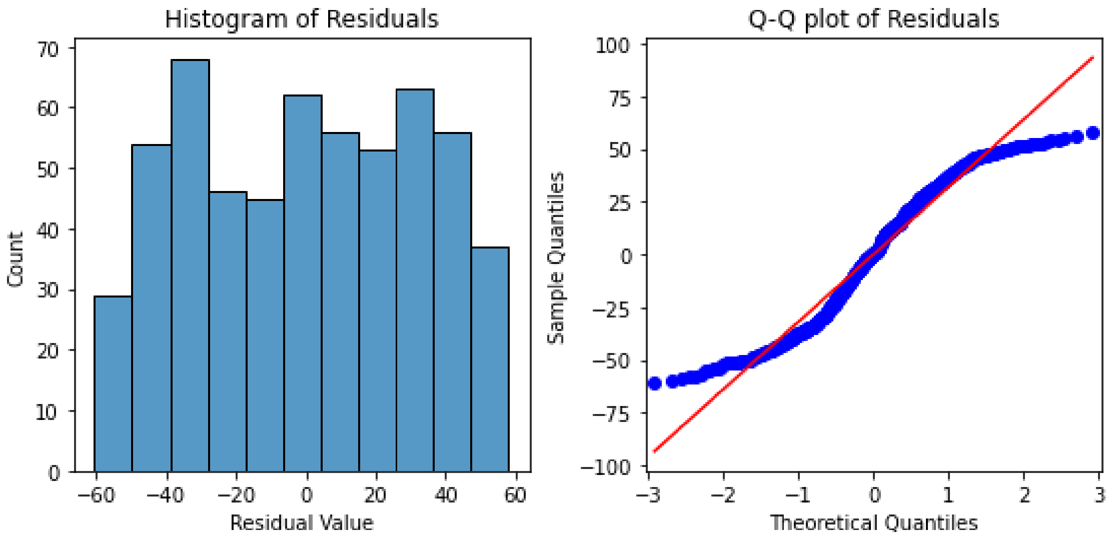
Because your model does not have any continuous independent variables, the linearity assumption is not required.

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.

[32]:

```python
import statsmodels.api as sm
import matplotlib.pyplot as plt
residuals = model.resid

fig, axes = plt.subplots(1, 2, figsize = (8,4))

# Calculate the residuals.
### YOUR CODE HERE ###

# Create a histogram with the residuals.
fig = sns.histplot(residuals,ax=axes[0])
axes[0].set_xlabel("Residual Value")
axes[0].set_title("Histogram of Residuals")

### YOUR CODE HERE ###


# Create a QQ plot of the residuals.
axes[1].set_title("Q-Q plot of Residuals")
fig = sm.qqplot(model.resid, line = 's',ax=axes[1])


### YOUR CODE HERE ###
plt.tight_layout()
```



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?

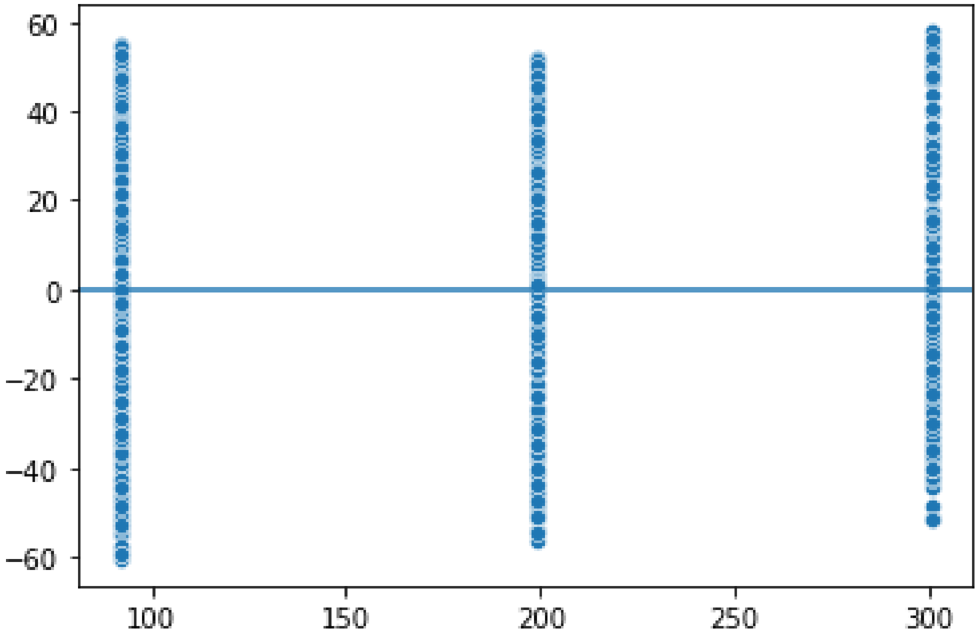
The Q-Q plot shows an “S” shape, not a desired behaviour.

**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.

[39]:

```
fig = sns.scatterplot(x = model.fittedvalues, y = model.resid)


fig.axhline(0)

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?

Yes, because it's spread constantly along the line.

1.5 Step 4: Results and evaluation

First, display the OLS regression results.

[40]:
```
# Display the model results summary.
model.summary()
### YOUR CODE HERE ###
```

[40]:
```
<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:                Thu, 09 Nov 2023   Prob (F-statistic):         8.81e-256
Time:                        13:24:26   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 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?

[C(TV)[T.Low] 208.8133 lower than the C(TV)[T.High] C(TV)[T.Medium] 101.5061 lower than the C(TV)[T.High] [P>|t| = 0.000, so, statistically significant. [-215.353, -202.274] confidence interval for C(TV)[T.Low]. Telling that the sales around the interval 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?

[Granular TV data: Break down TV promotions into more specific categories for a detailed analysis.

Budget breakdown: Include actual TV promotion budgets to better understand the investment in each category.

Location factor: Consider the geographical impact of marketing campaigns on the model's accuracy.

Seasonal influence: Account for the time of year to capture any seasonal trends in TV promotions.]

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.

[41]:

```
# Create an one-way ANOVA table for the fit model.
sm.stats.anova_lm(model, typ=2)
### YOUR CODE HERE ###
```

[41]:

```
                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`. Specify the type of ANOVA test (for example, one-way or two-way), using the `typ` parameter.

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

```
[Null: No difference between Sales and TV promotion budget.
Alternate: Got difference between Sales and TV promotion budget.
```

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

8.805550e-256, the p value is lower than the significance value of 0.05%, therefore rejecting the null hypothesis. Conclusion, there's a difference between Sales and TV promotion budget.

Question: What did the ANOVA test tell you?

there's a staticsally significant difference between Sales and TV promotion budget.

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.

[42]:

```
# Perform the Tukey's HSD post hoc test.
# Import Tukey's HSD function
from statsmodels.stats.multicomp import pairwise_tukeyhsd

# Run Tukey's HSD post hoc test for one-way ANOVA
tukey_oneway = pairwise_tukeyhsd(endog = data["Sales"], groups = data["TV"],␣
 ↪alpha = 0.05)

# Get results (pairwise comparisons)
tukey_oneway.summary()
### YOUR CODE HERE ###
```

[42]:

```
<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?

all 3 rows says thatt tthey are rejecting tthe null hypothesis.

Question: What did the post hoc tell you?**

This way provides more details, as in more ttthan one group can be seen whetther null hypo is rejected or not.

1.6 Considerations

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

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.

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.