In [1318...
```
# Import libraries
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from statsmodels.formula.api import ols
import statsmodels.api as sm
from statsmodels.stats.multicomp import pairwise_tukeyhsd
```

In [1320...
```
data = pd.read_csv(r"C:\Users\HP\Desktop\Advance Data Analyst\5. Simplify Complex Data Relationships\4. Module 4\2. Analysis of variance\Files\marketing_sales_data.csv")
data.head(10)
```

Out[1320...

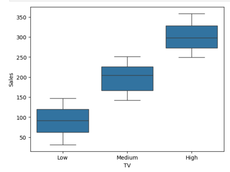
	TV	Radio	Social Media	Influencer	Sales
0	Low	1.218354	1.270444	Micro	90.054222
1	Medium	14.949791	0.274451	Macro	222.741668
2	Low	10.377258	0.061984	Mega	102.774790
3	High	26.469274	7.070945	Micro	328.239378
4	High	36.876302	7.618605	Mega	351.807328
5	High	25.561910	5.459718	Micro	261.966812
6	High	37.263819	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

In [1322...
```
# Display the first 10 rows of the data
data.head(10)
```

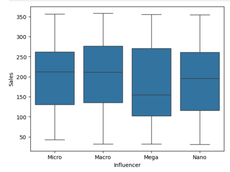
Out[1322...

	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

In [1324...
```
# Create a boxplot with TV and Sales.
sns.boxplot(x = "TV", y = "Sales", data = data);
```



In [1326...
```
# Create a boxplot with Influencer and Sales.
sns.boxplot(x = "Influencer", y = "Sales", data = data);
```



In [1328...
```
# Drop rows that contain missing data and update the DataFrame.

data = data.dropna(axis=0)


# Confirm the data contain no missing values.

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

Out[1328...
```
TV              0
Radio           0
Social Media    0
Influencer      0
Sales           0
dtype: int64
```

In [1330...
```
# Define the OLS formula.

ols_formula = 'Sales ~ C(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
```

Out[1330...

OLS Regression Results

Dep. Variable:	Sales	R-squared:	0.874
Model:	OLS	Adj. R-squared:	0.874
Method:	Least Squares	F-statistic:	1971.
Date:	Fri, 27 Sep 2024	Prob (F-statistic):	8.81e-256
Time:	17:28:50	Log-Likelihood:	-2778.9
No. Observations:	569	AIC:	5564.
Df Residuals:	566	BIC:	5577.
Df Model:	2		
Covariance Type:	nonrobust		

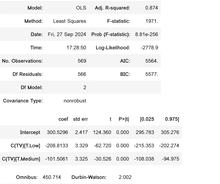
	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	300.5296	2.417	124.360	0.000	295.783	305.276
C(TV)[T.Low]	-208.8133	3.329	-62.720	0.000	-215.353	-202.274
C(TV)[T.Medium]	-101.5061	3.325	-30.526	0.000	-108.038	-94.975

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

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

In [1332...
```
# 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 QQ plot of the residuals.

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()
```

In [1334...
```
# Create a scatter plot 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()
```



In [1336...
```
# Display the model results summary.

model_results
```

Out[1336...

OLS Regression Results

Dep. Variable:	Sales	R-squared:	0.874
Model:	OLS	Adj. R-squared:	0.874
Method:	Least Squares	F-statistic:	1971.
Date:	Fri, 27 Sep 2024	Prob (F-statistic):	8.81e-256
Time:	17:28:50	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

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

In [1338...
```
# Create an one-way ANOVA table for the fit model.

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

Out[1338...

	sum_sq	df	F	PR(>F)
C(TV)	4.052692e+06	2.0	1971.455737	8.805550e-256
Residual	5.817569e+05	566.0	NaN	NaN

In [1340...
```
# Perform the Tukey's HSD post hoc test.

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

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

Out[1340...

Multiple Comparison of Means - Tukey HSD, FWER=0.05

group1	group2	meandiff	p-adj	lower	upper	reject
High	Low	-208.8133	0.0	-216.6367	-200.9898	True
High	Medium	-101.5061	0.0	-109.3202	-93.6921	True
Low	Medium	107.3072	0.0	99.7066	114.9077	True

In [ ]:
```
# some key takeaways

# 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.
```