**Title: Regression Analysis Case Study**

**1. Introduction**

This case study aims to analyse the impact of advertising expenditure on sales using linear regression. We will use a dataset containing monthly data on advertising expenditure and sales to perform our analysis.

**2. Data Inspection**

We start by inspecting the dataset to understand its structure and contents.

python

Copy code

# Loading data

import pandas as pd

excel\_file = r"C:/Users/user/Music/Regression case studies/Regression1.xlsx"

df = pd.read\_excel(excel\_file)

df

# Creating function to print info on the dataset

def inspect\_data(df):

print('Data Shape')

print('\n')

print(df.shape)

print('\n')

print('Missing Values: ')

print(df.isnull().sum())

print('\n')

print('Data Types: ')

print(df.dtypes)

inspect\_data(df)

**Data Shape**: It helps in summing up the rows and columns.

**Missing Values**: Helps in determining missing values present in the dataset.

**Data Types**: It helps in determining the type of data be it numerical or numerical data for advertising expenditure and sales.

**3. Data Cleaning**

We checked for missing values and duplicate rows to ensure data quality.

python

Copy code

missing\_values = df.isnull().sum()

print("\nMissing values in each column:")

print(missing\_values)

duplicates = df.duplicated()

print("\nDuplicated rows in the dataframe:")

print(df[duplicates])

**Handling Missing Values**: Checking for missing values

**Checking for Duplicates**: Checking for duplicated rows.

The above information will help you clean your data if missing values, duplicates and outliers are present.

**4. Summary Statistics**

Key summary statistics provide an overview of the dataset this can be in form: of percentage, frequency and cumulative for categorical data and mean, standard deviations, minimum and maximum for continuous data.

python

Copy code

# Summary statistics

summary\_stats = df.describe()

print("Summary Statistics:")

print(summary\_stats)

Mean, median, standard deviation, minimum, and maximum values for advertising expenditure and sales.

**5. Data Visualization**

This helps in displaying data in visual forms to identify trends and relationships. This can be inform of: scatter, bar graph, line graph, histogram and many others.

python

Copy code

import matplotlib.pyplot as plt

# Plotting the data

plt.figure(figsize=(12, 6))

# Advertising Expenditure

plt.subplot(2, 1, 1)

plt.plot(df['Month'], df['Advertising Expenditure (X)'], marker='o', color='blue')

plt.title('Advertising Expenditure Over Months')

plt.xlabel('Month')

plt.ylabel('Advertising Expenditure')

# Sales

plt.subplot(2, 1, 2)

plt.plot(df['Month'], df['Sales (Y)'], marker='o', color='green')

plt.title('Sales Over Months')

plt.xlabel('Month')

plt.ylabel('Sales')

plt.tight\_layout()

plt.show()

# Bubble Plot (Scatter plot with varying bubble size)

plt.figure(figsize=(10, 6))

plt.scatter(df['Advertising Expenditure (X)'], df['Sales (Y)'], s=100, alpha=0.5, c='blue', label='Data Points')

plt.title('Advertising Expenditure vs. Sales')

plt.xlabel('Advertising Expenditure')

plt.ylabel('Sales')

plt.legend()

plt.grid(True)

plt.show()

**Trend Over Time**: Plots showed the trends of advertising expenditure and sales over the months. There was a positive relationship between the expenditure and sales.

**Correlation**: The correlation matrix and heatmap indicated a strong positive relationship between advertising expenditure and sales.

python

Copy code

df.corr()

import seaborn as sns

import numpy as np

plt.figure(figsize=(7,5))

sns.heatmap(round(df.corr(),2),annot=True)

plt.show()

**6. Scatter Plot**

A scatter plot of advertising expenditure vs. sales revealed a clear positive linear relationship.

**7. Preparing Data for Regression**

We prepared the data for regression analysis by splitting it into training and testing sets.

python

Copy code

from sklearn.linear\_model import LinearRegression # Importing Linear Regression Model

from sklearn.model\_selection import train\_test\_split # Importing Model Selection to split dataset into training and testing set

from sklearn.metrics import r2\_score # Importing R² score

df1= df.drop("Month", axis=1)

X = df1.drop('Sales (Y)',axis =1 )# Selecting independent variables by removing 'median\_house\_value' column from dataframe

X

y = df1['Advertising Expenditure (X)'] # Selecting dependent variable from the dataframe

y

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X,y, test\_size= 0.3)

**Independent Variable (X)**: Advertising Expenditure

**Dependent Variable (y)**: Sales

**Train-Test Split**: 70% training data, 30% testing data

**8. Model Training**

We trained a linear regression model on the training data.

python

Copy code

lr = LinearRegression() # Initializing the model

lr.fit(X\_train, y\_train) # Fitting the model to the training data

y\_pred = lr.predict(X\_test) # Predicting prices on the testing set

score = r2\_score(y\_test, y\_pred) # Measuring Model performance by comparing the predicted values to the actual values

print(f'\nModel R² Score: {score:.3f}') # Printing R² score

**9. Model Performance**

We evaluated the model's performance using the R² score and examined the regression coefficients and intercept.

python

Copy code

# Printing regression coefficients and intercept

print(f'Coefficients: {lr.coef\_}')

print(f'Intercept: {lr.intercept\_}')

**R² Score**: The model achieved an R² score of 1, indicating a good fit.

**Coefficients**: The model's coefficients showed the expected positive relationship between advertising expenditure and sales.

**Intercept**: The intercept value was b.

The regression equation was as follows: -3.088083 +0 .3350604 advertisement

The positive *b* indicated that any increase in advertisement will increase read to increase in sales.

**NB: The same explanation and interpretation follow it whilst using STATA with the codes that I shared and the one you saved.**