Program 1

import pandas as pd

import numpy as np

import scipy.stats as stats

np.random.seed(42)

df = pd.DataFrame({

'TV\_Ad\_Spend': np.random.uniform(50, 300, 150),

'Radio\_Ad\_Spend': np.random.uniform(10, 50, 150),

'Region': np.random.choice(['North', 'South', 'West', 'East'], 150)

})

df['Sales'] = 50 + df['TV\_Ad\_Spend'] \* 0.9 + df['Radio\_Ad\_Spend'] \* 1.3 + np.random.normal(0, 25, 150)

df.loc[df['Region'] == 'South', 'Sales'] += 30

tv\_spend = df['TV\_Ad\_Spend']

sales = df['Sales']

correlation\_coefficient, p\_value = stats.pearsonr(tv\_spend, sales)

print(f"Correlation Coefficient: {correlation\_coefficient:.4f}\nP-value: {p\_value:.4f}")

Program 2

import pandas as pd

import matplotlib.pyplot as plt

import numpy as np

np.random.seed(42)

df = pd.DataFrame({

'TV\_Ad\_Spend':np.random.uniform(50, 300, 150),

'Radio\_Ad\_Spend': np.random.uniform(10, 50, 150),

'Region': np.random.choice(['North', 'South', 'West', 'East'], 150)

})

df['Sales'] = 50 + df['TV\_Ad\_Spend'] \* 0.9 + df['Radio\_Ad\_Spend'] \* 1.3 + np.random.normal(0, 25, 150)

df.loc[df['Region'] == 'South', 'Sales'] += 30

plt.scatter(df['TV\_Ad\_Spend'], df['Sales'])

plt.title('Sales vs. TV Ad Spend')

plt.xlabel('TV Ad Spend ($)')

plt.ylabel('Sales ($)')

plt.show()

Program 3

import pandas as pd

import numpy as np

import statsmodels.formula.api as smf

import matplotlib.pyplot as plt

np.random.seed(42)

df = pd.DataFrame({

'TV\_Ad\_Spend': np.random.uniform(50, 300, 150),

'Radio\_Ad\_Spend': np.random.uniform(10, 50, 150),

'Region': np.random.choice(['North', 'South', 'West', 'East'], 150)

})

df['Sales'] = 50 + df['TV\_Ad\_Spend'] \* 0.9 + df['Radio\_Ad\_Spend'] \* 1.3 + np.random.normal(0, 25, 150)

df.loc[df['Region'] == 'South', 'Sales'] += 30

model = smf.ols('Sales ~ TV\_Ad\_Spend', data=df).fit()

print(model.summary())

plt.scatter(df['TV\_Ad\_Spend'], df['Sales'])

plt.plot(df['TV\_Ad\_Spend'], model.predict(df), color='red')

plt.title('Sales vs. TV Ad Spend with Regression Line')

plt.xlabel('TV Ad Spend ($)')

plt.ylabel('Sales ($)')

plt.show()

Program 4

import pandas as pd

import numpy as np

import scipy.stats as stats

np.random.seed(42)

df = pd.DataFrame({

'TV\_Ad\_Spend': np.random.uniform(50, 300, 150),

'Radio\_Ad\_Spend': np.random.uniform(10, 50, 150),

'Region': np.random.choice(['North', 'South', 'West', 'East'], 150)

})

df['Sales'] = 50 + df['TV\_Ad\_Spend'] \* 0.9 + df['Radio\_Ad\_Spend'] \* 1.3 + np.random.normal(0, 25, 150)

df.loc[df['Region'] == 'South', 'Sales'] += 30

sales\_south = df[df['Region'] == 'South']['Sales']

sales\_east = df[df['Region'] == 'East']['Sales']

t\_stat, p\_value = stats.ttest\_ind(sales\_south, sales\_east)

print(f"T-statistic: {t\_stat:.4f}\nP-value: {p\_value:.4f}")

Program 5

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

np.random.seed(42)

df = pd.DataFrame({

'TV\_Ad\_Spend': np.random.uniform(50, 300, 150),

'Radio\_Ad\_Spend': np.random.uniform(10, 50, 150),

'Region': np.random.choice(['North', 'South', 'West', 'East'], 150)

})

df['Sales'] = 50 + df['TV\_Ad\_Spend'] \* 0.9 + df['Radio\_Ad\_Spend'] \* 1.3 + np.random.normal(0, 25, 150)

df.loc[df['Region'] == 'South', 'Sales'] += 30

print(df['Region'].value\_counts())

plt.bar(df['Region'].value\_counts().index, df['Region'].value\_counts().values)

plt.title('Number of Sales Observations per Region')

plt.xlabel('Region')

plt.ylabel('Count')

plt.show()

Program 6

import pandas as pd

import numpy as np

import statsmodels.formula.api as smf

np.random.seed(42)

df = pd.DataFrame({

'TV\_Ad\_Spend': np.random.uniform(50, 300, 150),

'Radio\_Ad\_Spend': np.random.uniform(10, 50, 150),

'Region': np.random.choice(['North', 'South', 'West', 'East'], 150)

})

df['Sales'] = (

50 + df['TV\_Ad\_Spend'] \* 0.9 + df['Radio\_Ad\_Spend'] \* 1.3

+ np.random.normal(0, 25, 150)

)

df.loc[df['Region'] == 'South', 'Sales'] += 30

mlr = smf.ols('Sales ~ TV\_Ad\_Spend + C(Region)', data=df).fit()

print("\n--- Multiple Linear Regression Summary ---")

print("R-squared:", round(mlr.rsquared, 3))

print("Coefficients:\n", mlr.params.round(3))

print("\nP-values:\n", mlr.pvalues.round(4))

Program 7

import pandas as pd

import numpy as np

import statsmodels.formula.api as smf

import matplotlib.pyplot as plt

import seaborn as sns

np.random.seed(42)

df = pd.DataFrame({

'TV\_Ad\_Spend': np.random.uniform(50, 300, 150),

'Radio\_Ad\_Spend': np.random.uniform(10, 50, 150),

'Region': np.random.choice(['North', 'South', 'West', 'East'], 150)

})

df['Sales'] = 50 + df['TV\_Ad\_Spend'] \* 0.9 + df['Radio\_Ad\_Spend'] \* 1.3 + np.random.normal(0, 25, 150)

df.loc[df['Region'] == 'South', 'Sales'] += 30

# Build a full model with all our main predictors

full\_model = smf.ols(formula='Sales ~ TV\_Ad\_Spend + Radio\_Ad\_Spend + C(Region)', data=df).fit()

# Print the full OLS summary for detailed interpretation

print("--- Full OLS Regression Results ---")

print(full\_model.summary())

Program 8

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

np.random.seed(42)

df = pd.DataFrame({

'TV\_Ad\_Spend': np.random.uniform(50, 300, 150),

'Radio\_Ad\_Spend': np.random.uniform(10, 50, 150),

'Region': np.random.choice(['North', 'South', 'West', 'East'], 150)

})

df['Sales'] = 50 + df['TV\_Ad\_Spend'] \* 0.9 + df['Radio\_Ad\_Spend'] \* 1.3 + np.random.normal(0, 25, 150)

df.loc[df['Region'] == 'South', 'Sales'] += 30

print(df['Sales'].describe())

plt.hist(df['Sales'], bins=20)

plt.title('Histogram of Sales')

plt.xlabel('Sales')

plt.ylabel('Frequency')

plt.show()

Program 9

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

np.random.seed(42)

df = pd.DataFrame({

'TV\_Ad\_Spend': np.random.uniform(50, 300, 150),

'Radio\_Ad\_Spend': np.random.uniform(10, 50, 150),

'Region': np.random.choice(['North', 'South', 'West', 'East'], 150)

})

df['Sales'] = 50 + df['TV\_Ad\_Spend'] \* 0.9 + df['Radio\_Ad\_Spend'] \* 1.3 + np.random.normal(0, 25, 150)

df.loc[df['Region'] == 'South', 'Sales'] += 30

df.boxplot(column='Sales', by='Region')

plt.title('Sales Distribution by Region')

plt.suptitle('')

plt.xlabel('Region')

plt.ylabel('Sales')

plt.show()

Program 10

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

np.random.seed(42)

df = pd.DataFrame({

'TV\_Ad\_Spend': np.random.uniform(50, 300, 150),

'Radio\_Ad\_Spend': np.random.uniform(10, 50, 150),

'Region': np.random.choice(['North', 'South', 'West', 'East'], 150)

})

df['Sales'] = 50 + df['TV\_Ad\_Spend'] \* 0.9 + df['Radio\_Ad\_Spend'] \* 1.3 + np.random.normal(0, 25, 150)

df.loc[df['Region'] == 'South', 'Sales'] += 30

pd.plotting.scatter\_matrix(df[['TV\_Ad\_Spend','Radio\_Ad\_Spend','Sales']])

plt.suptitle('Pair Plot of Numerical Variables')

plt.show()

print(df[['TV\_Ad\_Spend','Radio\_Ad\_Spend','Sales']].corr())

Program 11

import pandas as pd

import numpy as np

np.random.seed(42)

df = pd.DataFrame({

'TV\_Ad\_Spend': np.random.uniform(50, 300, 150),

'Radio\_Ad\_Spend': np.random.uniform(10, 50, 150),

'Region': np.random.choice(['North', 'South', 'West', 'East'], 150)

})

df['Sales'] = 50 + df['TV\_Ad\_Spend'] \* 0.9 + df['Radio\_Ad\_Spend'] \* 1.3 + np.random.normal(0, 25, 150)

df.loc[df['Region'] == 'South', 'Sales'] += 30

sales = df['Sales']

print(f"Mean: {sales.mean():.2f}\nMedian: {sales.median():.2f}\nMode: {sales.mode()[0]:.2f}")

print(f"Variance: {sales.var():.2f}\nStd Dev: {sales.std():.2f}\nRange: {sales.max()-sales.min():.2f}")

Program 12

import pandas as pd

import numpy as np

import scipy.stats as stats

np.random.seed(42)

df = pd.DataFrame({

'TV\_Ad\_Spend': np.random.uniform(50, 300, 150),

'Radio\_Ad\_Spend': np.random.uniform(10, 50, 150),

'Region': np.random.choice(['North', 'South', 'West', 'East'], 150)

})

df['Sales'] = 50 + df['TV\_Ad\_Spend'] \* 0.9 + df['Radio\_Ad\_Spend'] \* 1.3 + np.random.normal(0, 25, 150)

df.loc[df['Region'] == 'South', 'Sales'] += 30

groups = [df[df['Region'] == r]['Sales'] for r in ['North','South','East','West']]

f\_stat, p\_value = stats.f\_oneway(\*groups)

print(f"F-statistic: {f\_stat:.4f}\nP-value: {p\_value:.4f}")