Prodigy Infotect Data Science Internship Task-2

- Created Date:09.03.2024
- Created By: Blessy Louis
- edited on:09.03.2024

Importing necessary packages for Analysis:

1. Pandas: Is a Python library used for working with data sets. It has functions for analyzing, cleaning, exploring, and manipulating data.
2. Numpy:NumPy enhances Python's mathematical operations on arrays and matrices by providing a powerful data structure, a vast library of high-level functions, and efficient calculations.
3. matplotlib.pyplot: Matplotlib is an amazing visualization library in Python for 2D plots of arrays. Matplotlib is a multi-platform data visualization library built on NumPy arrays.Matplotlib consists of several plots like line, bar, scatter, histogram, etc.
4. Stats: Scipy is a powerful library in Python that provides many useful functions for scientific computing. One of its sub-modules, scipy. stats, contains a variety of statistical functions and probability distributions that are commonly used in data analysis.
5. seaborn:Python Seaborn library is a widely popular data visualization library that is commonly used for data science and machine learning tasks. You build it on top of the matplotlib data visualization library and can perform exploratory analysis.
6. Plotly-express:Plotly Express is the easy-to-use, high-level interface to Plotly, which operates on a variety of types of data and produces easy-to-style figures.

In [ ]:

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from scipy import stats
import seaborn as sns
import plotly.express as px
```

Load Dataset:

The dataset used for this Analysis is the sales prices incured from differnt modes of advertisement , that include features like:

- TV: Amount spent on advertisement on tv
- Radio: Amount spent on advertisement on radio

- Newspaper: Amount spent on advertisement on Newspaper
- sales: total sales

In [ ]:

```
df=pd.read_csv('advertising.csv')
sales=df['Sales']
sales
```

Out[ ]:

```
0      22.1
1      10.4
2      12.0
3      16.5
4      17.9
       ...
195     7.6
196    14.0
197    14.8
198    25.5
199    18.4
Name: Sales, Length: 200, dtype: float64
```

Displaying first 5 rows of the dataset

In [ ]:

```
df.head()
```

Out[ ]:

	TV	Radio	Newspaper	Sales
0	230.1	37.8	69.2	22.1
1	44.5	39.3	45.1	10.4
2	17.2	45.9	69.3	12.0
3	151.5	41.3	58.5	16.5
4	180.8	10.8	58.4	17.9

Displaying the number of rows and columns of the dataset

In [ ]:

```
df.shape
```

Out[ ]:

```
(200, 4)
```

The dataset contains 200 rows and 4 features/cloumns

Data Description: Features in the data

```
In [ ]: df.columns
```

```
Out[ ]: Index(['TV', 'Radio', 'Newspaper', 'Sales'], dtype='object')
```

Checking for null values

```
In [ ]: df.isnull().sum()
```

```
Out[ ]: TV           0
        Radio        0
        Newspaper    0
        Sales        0
        dtype: int64
```

clearly , we observe that there is no null values in the dataset , since the count of the number of null values for each feature is zero.

Displaying the information about the data

```
In [ ]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 4 columns):
 #   Column     Non-Null Count  Dtype  
---  ------     --------------  -----  
 0   TV         200 non-null    float64
 1   Radio      200 non-null    float64
 2   Newspaper  200 non-null    float64
 3   Sales      200 non-null    float64
dtypes: float64(4)
memory usage: 6.4 KB
```

The above output gives us the information about the number of enteries in the dataset i,e. there are a total of 200 ranging from 0-199, the dataset includes a total of 4 columns /features we also have the information about each feature , we can see the number of enteries , its a non null and the datatype . we can see that the datatypes available in the dataset are: float64, and the memory used is 6.4kb

Understanding some basic descriptive statistics of the dataset

In [ ]:
```
df.describe()
```

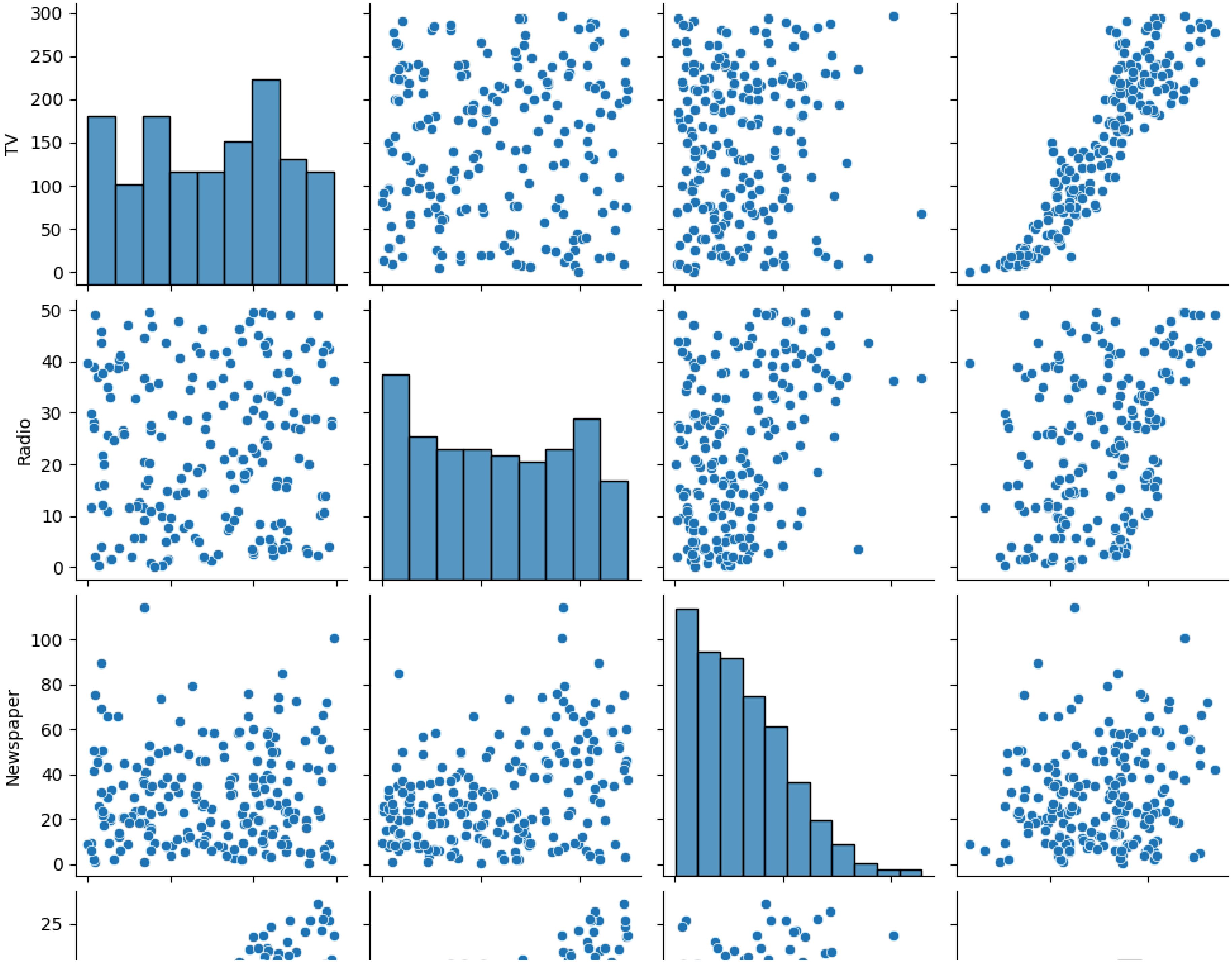
Out[ ]:

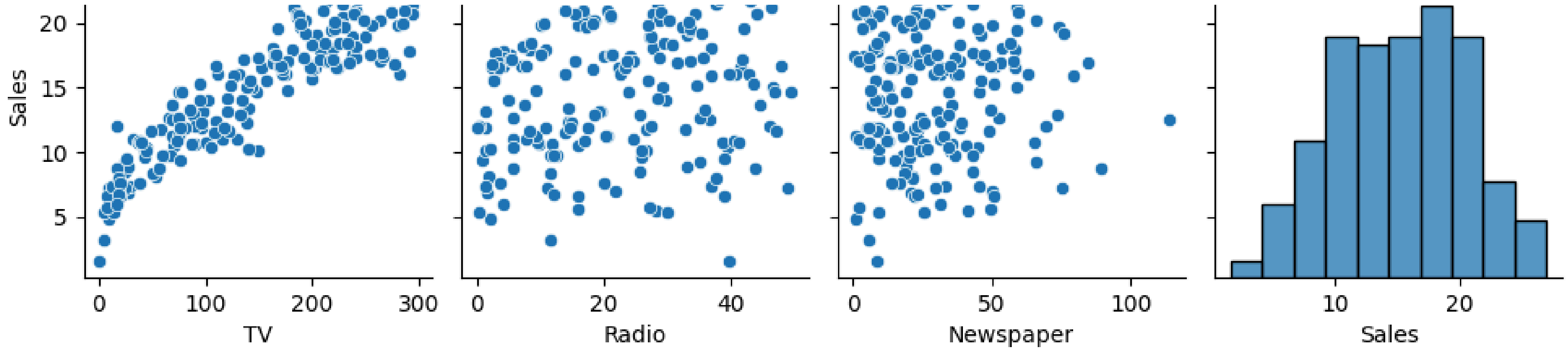
	TV	Radio	Newspaper	Sales
count	200.000000	200.000000	200.000000	200.000000
mean	147.042500	23.264000	30.554000	15.130500
std	85.854236	14.846809	21.778621	5.283892
min	0.700000	0.000000	0.300000	1.600000
25%	74.375000	9.975000	12.750000	11.000000
50%	149.750000	22.900000	25.750000	16.000000
75%	218.825000	36.525000	45.100000	19.050000
max	296.400000	49.600000	114.000000	27.000000

The output provides information about each feature's count, minimum, maximum,mean standard deviation, and the different quartile values

Graphical presentation

In [ ]:
```
sns.pairplot(df)
plt.show()
```


TV
300
250
200
150
100
50
0
Radio
50
40
30
20
10
0
Newspaper
100
80
60
40
20
0
25

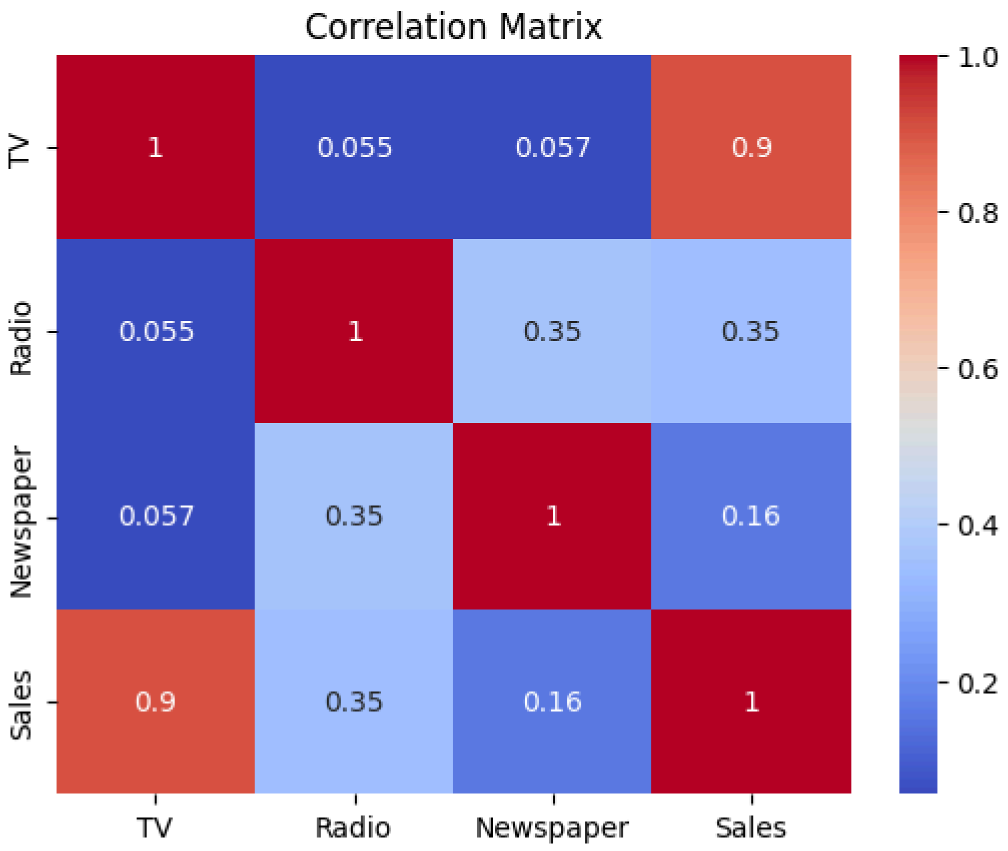


- The pairplot shows pairwise relationships between the variables (TV, Radio, Newspaper, and Sales).
- It helps in visualizing the correlations between variables and identifying potential patterns.

Inference: We can observe the distribution of each variable and see if there's any correlation between advertising channels and sales.

In [ ]:

```
correlation_matrix = df.corr()
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm')
plt.title('Correlation Matrix')
plt.show()
```



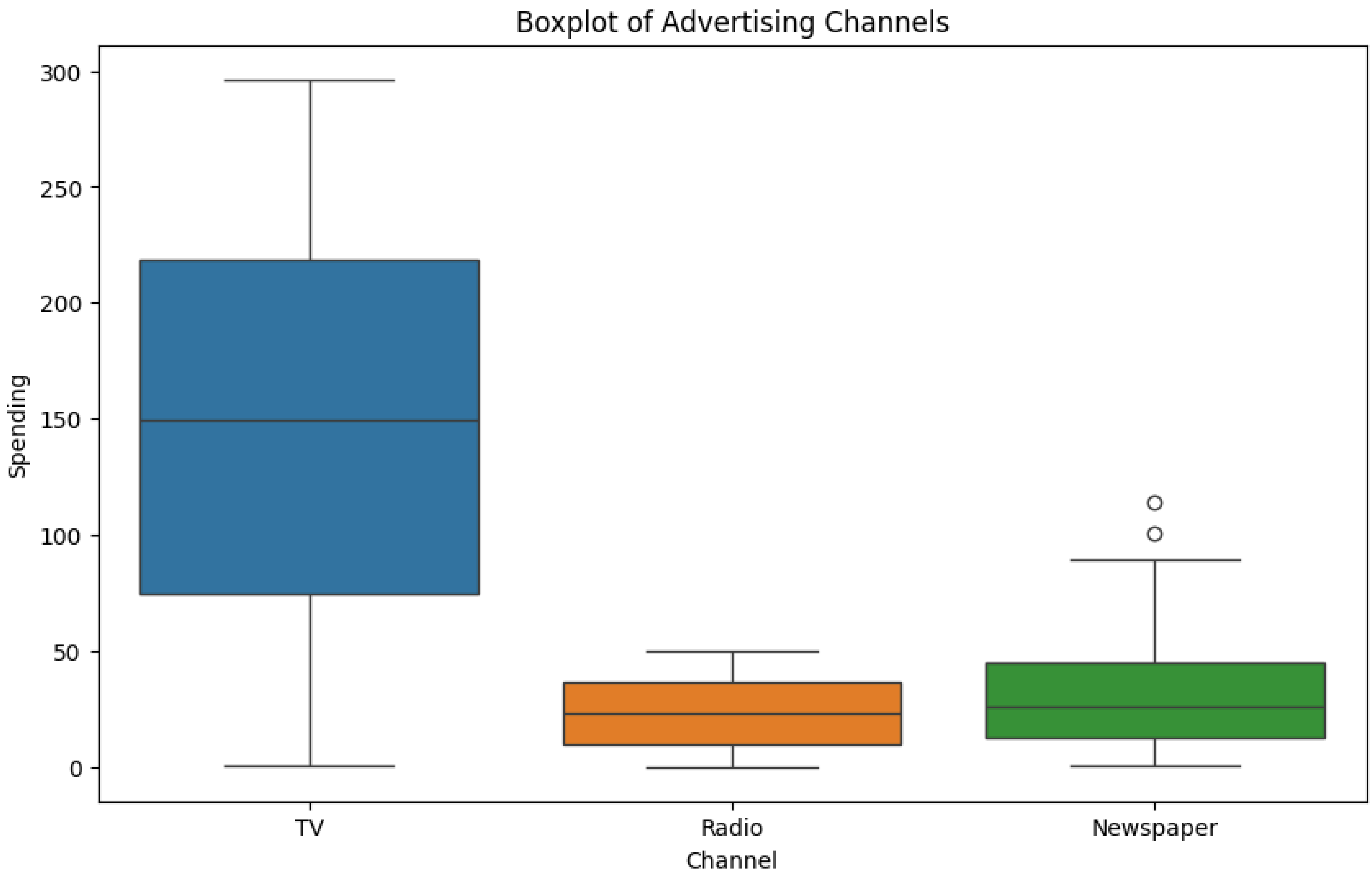
- The correlation matrix visualizes the correlation coefficients between all pairs of variables.
- Values range from -1 to 1, where 1 indicates a perfect positive correlation, -1 indicates a perfect negative correlation, and 0 indicates no correlation.

Inference: We can see which advertising channels have stronger correlations with sales. A high positive correlation indicates that increasing spending on that channel tends to increase sales.

clearly we see that there is a high positive correlation (0.9) Between TV and Sales The second highest is Radio with a low positive correlation(0.35) in relationship with Sales The least correlated (0.16) is newspaper and sales

In [ ]:
```
plt.figure(figsize=(10, 6))
sns.boxplot(data=df[['TV', 'Radio', 'Newspaper']])
```

```
plt.title('Boxplot of Advertising Channels')
plt.xlabel('Channel')
plt.ylabel('Spending')
plt.show()
```

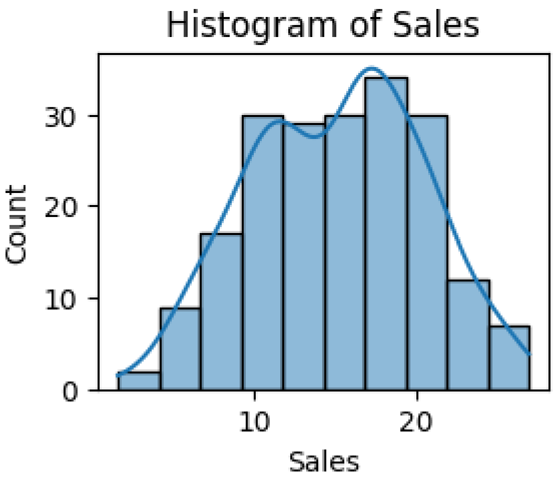


- The boxplot visualizes the distribution of advertising spending for each channel (TV, Radio, Newspaper).
- It helps in identifying outliers and comparing the distributions between different channels.

Inference: We can observe the spread of spending across different advertising channels. It can help us identify which channel has higher variability or median spending. clearly we see that TV Channel of advertisement has highest spread of money spent , followed by newspaper

In [ ]:
```
plt.subplot(2, 2, 1)
sns.histplot(df['Sales'], kde=True)
plt.title('Histogram of Sales')
```

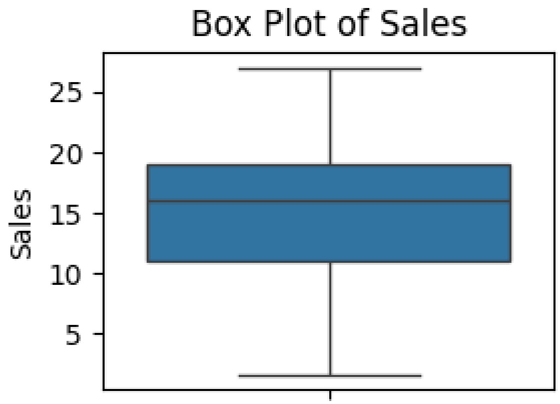
Out[ ]:
```
Text(0.5, 1.0, 'Histogram of Sales')
```



Clearly we observe that the distribution of sales is left skewed and since the the shape of the graph is almost resembling normal , so the distribution maybe mesokurtic in nature.

In [ ]:
```
plt.subplot(2, 2, 2)
sns.boxplot(y=df['Sales'])
plt.title('Box Plot of Sales')
```

Out[ ]:
```
Text(0.5, 1.0, 'Box Plot of Sales')
```

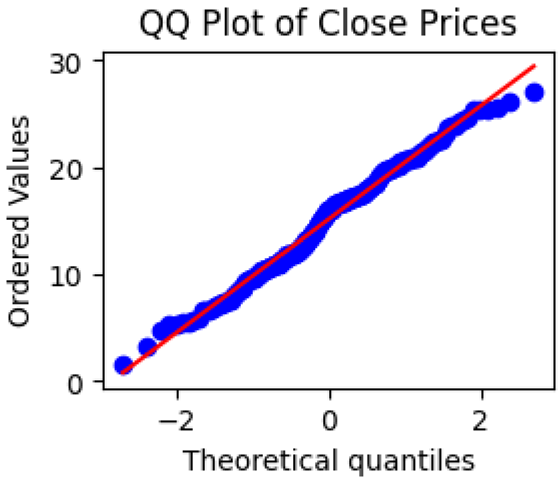


The above graph represents the boxplot of the sales distribution , the distribution has no outliers , a most of the data values lie beyond the median value.

The QQ plot compares sales data's distribution against a theoretical normal distribution. If points follow a straight line, it suggests normality. If deviations are significant, they suggest departures from normality. Outliers are far from the line, and the fit indicates the normal distribution's fit.

In [ ]:
```
plt.subplot(2, 2, 3)
stats.probplot(df['Sales'], dist="norm", plot=plt)
plt.title('QQ Plot of Close Prices')
```

Out[ ]:
```
Text(0.5, 1.0, 'QQ Plot of Close Prices')
```



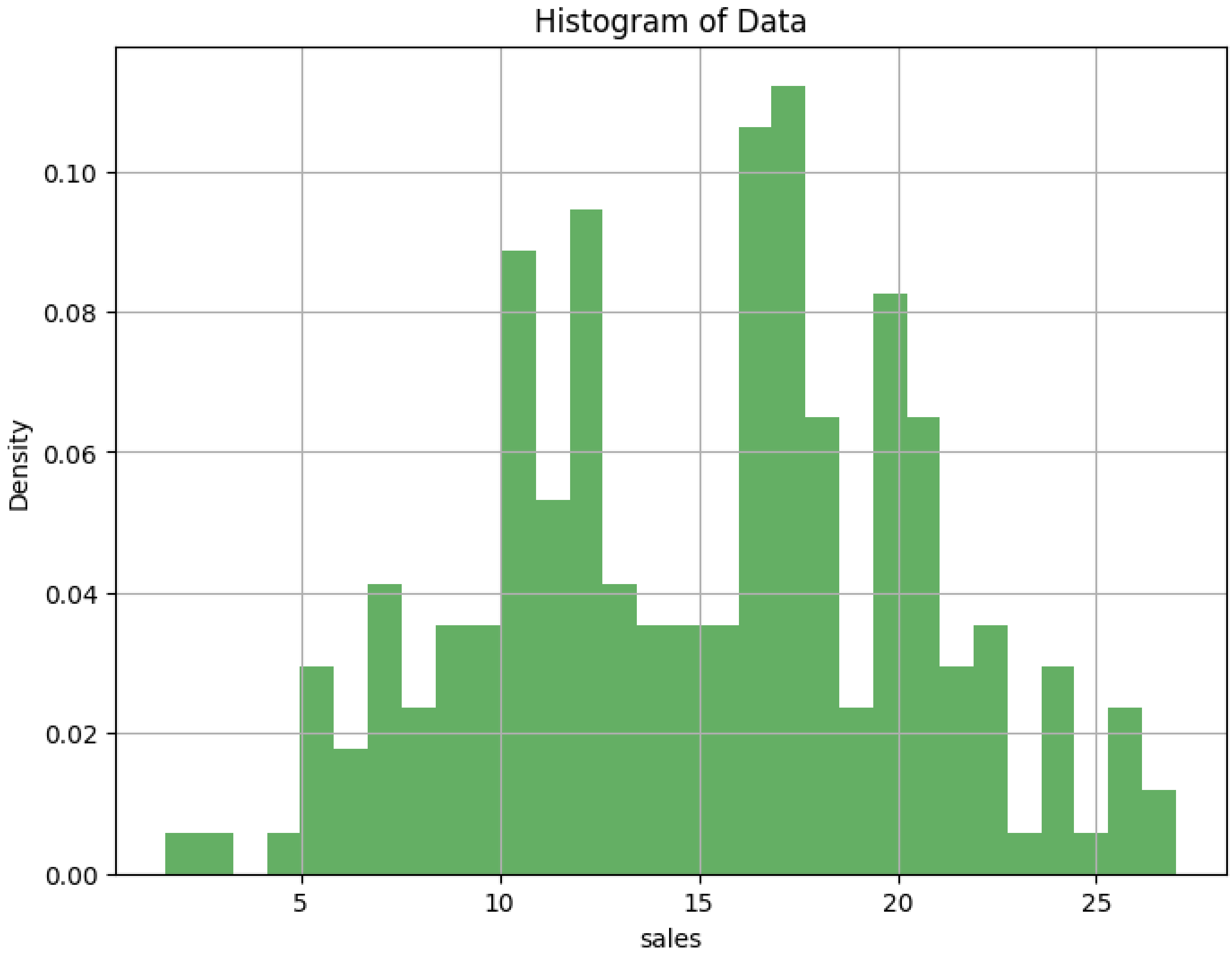
We, clearly observe that , all point follow the straight line , hence we can conclude that the sales distribution is normally distributed

In [ ]:
```
# Descriptive Statistics
mean = np.mean(sales)
std_dev = np.std(sales)
skewness = np.mean((sales - mean) ** 3) / (std_dev ** 3)
kurtosis = np.mean((sales - mean) ** 4) / (std_dev ** 4) - 3

print("Descriptive Statistics:")
print(f"Mean: {mean:.2f}")
print(f"Standard Deviation: {std_dev:.2f}")
print(f"Skewness: {skewness:.2f}")
print(f"Kurtosis: {kurtosis:.2f}")

# the skewness is negative, the distribution is skewed to the left

# If the kurtosis is positive, the distribution has heavier tails and a sharper peak than the normal distribution. This is called
# If the kurtosis is negative, the distribution has lighter tails and a flatter peak than the normal distribution. This is called
# If the kurtosis is close to zero, the distribution is similar to the normal distribution and is called mesokurtic.
```

```
Descriptive Statistics:
Mean: 15.13
Standard Deviation: 5.27
Skewness: -0.07
Kurtosis: -0.65
```

From the results , we observe that the sales of this dataset is negatively skewed , and platykurtic in nature

In [ ]:
```
plt.figure(figsize=(8, 6))
plt.hist(sales, bins=30, density=True, alpha=0.6, color='g')
plt.title('Histogram of Data')
plt.xlabel('sales')
plt.ylabel('Density')
plt.grid(True)
plt.show()
```



The code generates a sales data histogram using matplotlib, providing a visual representation of the data's distribution. The histogram's shape, density, number of bins, grid lines, and title and labels help interpret the data, providing insights into patterns, central tendency, and variability.

The graph clearly predicts that the distribution of sales in normal , and the maximum denisty is of data values is beyond 0.10

Parametric Method analysis:

One-way ANOVA:The one-way analysis of variance (ANOVA) is used to determine whether there are any statistically significant differences between the means of three or more independent (unrelated) groups. This guide will provide a brief introduction to the one-way ANOVA, including the assumptions of the test and when you should use this test

Here, the three classes are class-A;TV Class-B;Radio class-C;Newspaper

Hypothesis: H0: The mean of prices of all three groups are same vs H1: The mean of prices of atleast a pair of groups is not same

In [ ]:

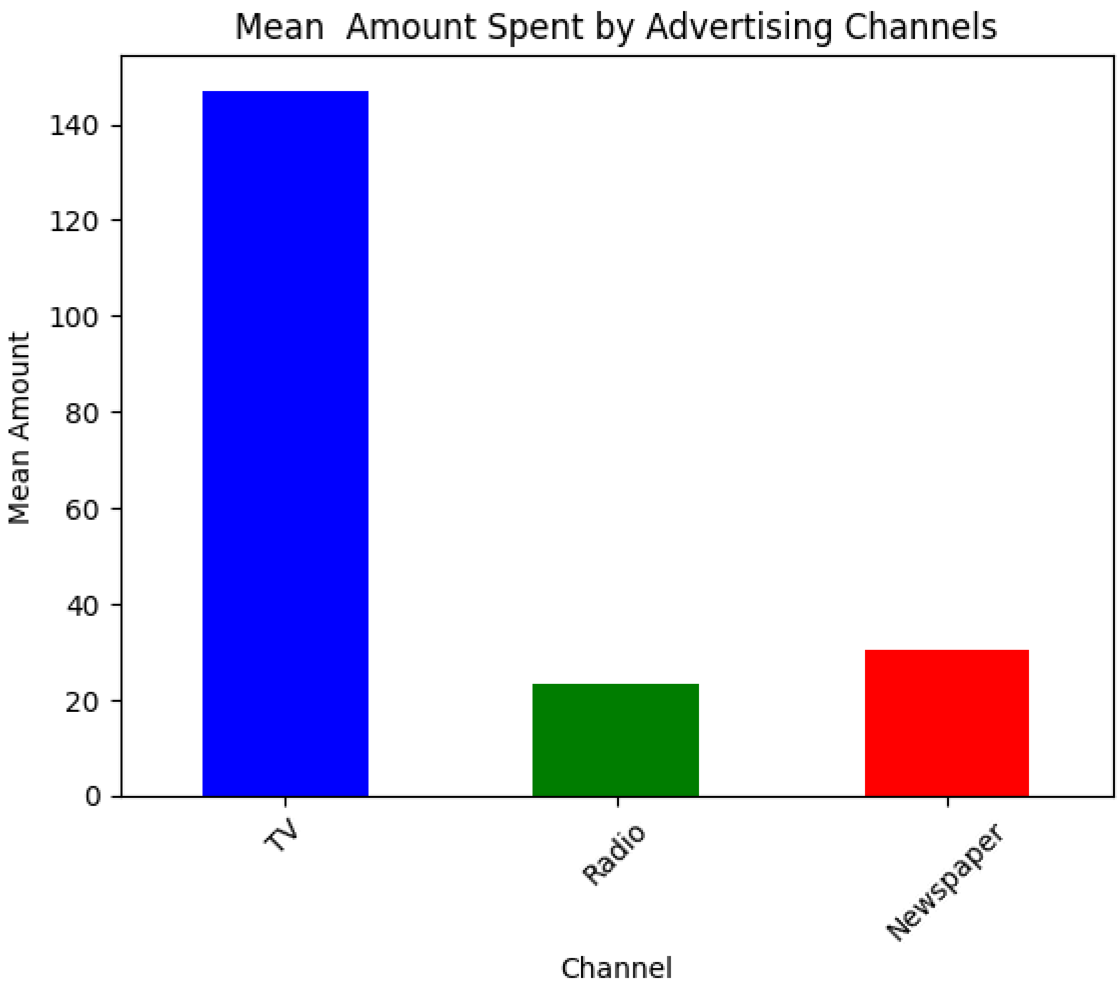
```
from scipy.stats import f_oneway

# Perform one-way ANOVA
f_statistic, p_value = f_oneway(df['TV'], df['Radio'], df['Newspaper'])
print("ANOVA p-value:", p_value)
```

```
ANOVA p-value: 4.552931539744962e-103
```

The p-value resulted is very small and almost equal to zero , therefore we reject H0 and conclude that , The mean of prices of atleast a pair of groups is not same

In [ ]:

```
mean_sales_by_channel = df[['TV', 'Radio', 'Newspaper']].mean()
mean_sales_by_channel.plot(kind='bar', color=['blue', 'green', 'red'])
plt.title('Mean  Amount Spent by Advertising Channels')
plt.xlabel('Channel')
plt.ylabel('Mean Amount')
plt.xticks(rotation=45)
plt.show()
```

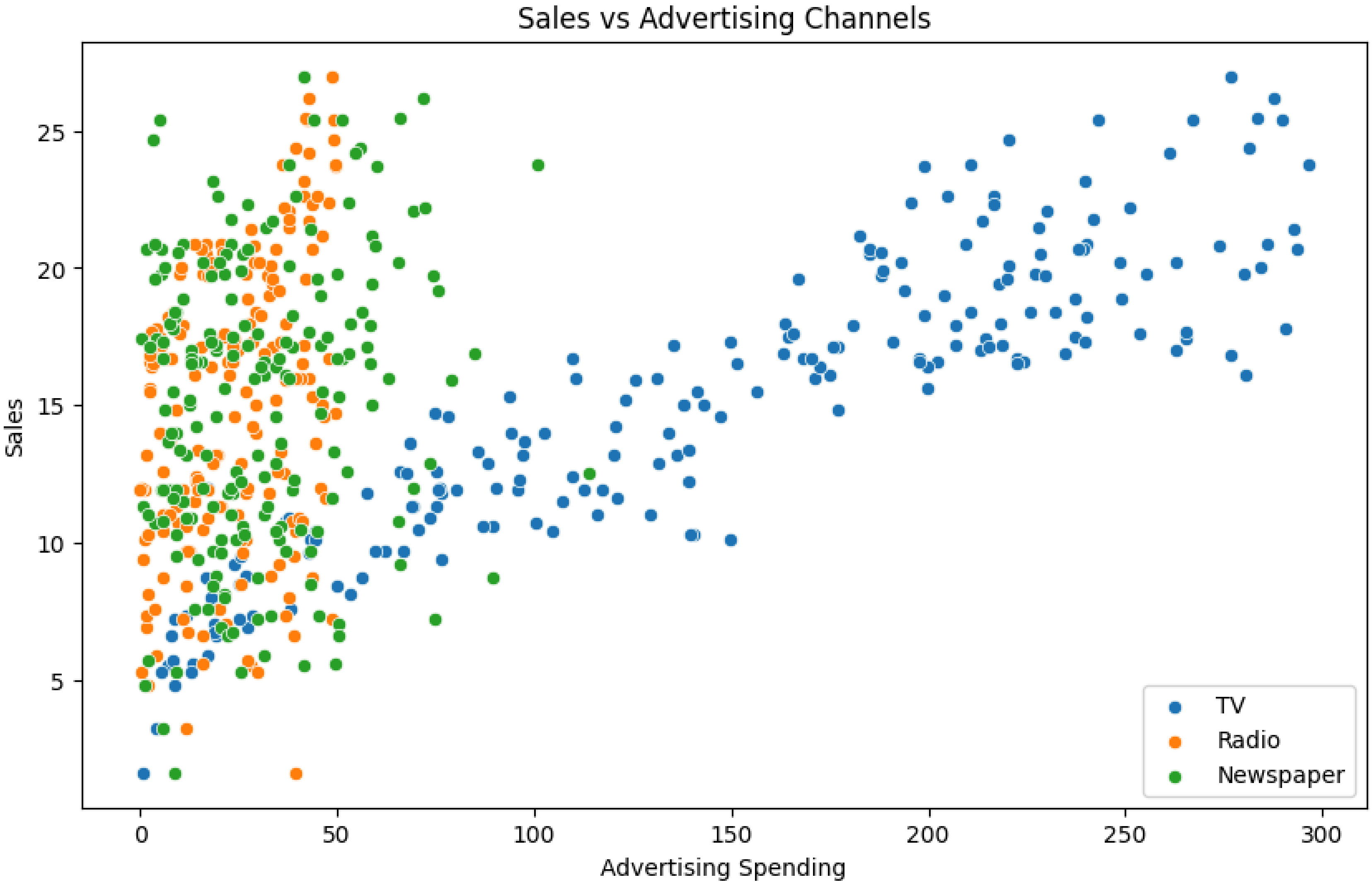


- The barplot shows the mean sales for each advertising channel.
- It helps in comparing the average amount spent through different advertising channels.

Inference: We can identify which channel, on average, Has spent more. This information can help in resource allocation and decision-making.

We observe that TV channel has the most amount to be spent on

In [ ]:
```
plt.figure(figsize=(10, 6))
sns.scatterplot(x='TV', y='Sales', data=df, label='TV')
sns.scatterplot(x='Radio', y='Sales', data=df, label='Radio')
sns.scatterplot(x='Newspaper', y='Sales', data=df, label='Newspaper')
plt.title('Sales vs Advertising Channels')
plt.xlabel('Advertising Spending')
plt.ylabel('Sales')
plt.legend()
plt.show()
```



- The scatterplot visualizes the relationship between advertising spending and sales for each channel.
- It helps in identifying any patterns or trends in how sales vary with advertising spending.

Inference: We can observe if there's a linear relationship between advertising spending and sales for each channel. A positive slope indicates that increasing spending leads to higher sales.

Conclusion:

The exploratory data analysis (EDA) and ANOVA test conducted on the dataset reveal significant differences in sales among the advertising channels (TV, Radio, Newspaper). The ANOVA test yielded an extremely small p-value (4.55e-103), indicating strong evidence against the null hypothesis. Graphical representations, including boxplots, barplots, and scatterplots, provided visual insights into the distribution of advertising spending, mean sales by channel, and the relationship between spending and sales. These findings suggest that at least one advertising channel has a different effect on sales compared to the others. This analysis underscores the importance of optimizing marketing strategies and resource allocation to maximize sales effectiveness.