CASE STUDY:

Advertising is a marketing strategy involving paying for space to promote a product, service, or cause. The actual promotional messages are called advertisements, or ads for short. The goal of advertising is to reach people most likely to be willing to pay for a company's products or services and entice them to buy. Data mining can help advertising refine its message and its audiences. There are two downloaded datasets named Advertising_df1_raw.csv and Advertising_df2_raw.csv from different databases. Use both given as datasets where all records of the details being taken and the attributes involved are:

a. TV

b. Radio

c. Newspaper

d. Sales

Question 1

General Knowledge Discuss the ETL concept related to the case study above ETL= Extract,Transform,Load Based on above case study, we have to extract datasets named Advertising_df1_raw.csv and Advertising_df2_raw.csv which contain informations like TV,Radio,Newspaper and Sales data for advertising purpose.

Transformation is the process of converting the extracted data into a format suitable for analysis and decision-making. which include cleaning data by addressing the missing values with measurements of central tendency(mean,mode,median) and merging dataframe(Advertising_df1_raw and Advertising_df2_raw) call df_merge.

Load phase involves storing the transformed data into a target database or data warehouse for analysis and reporting. Saving the cleaned and transformed dataset in a suitable format.In this case we name it as df_merge_clean.

Question 2

Python: Data Preparation

In [1]:
```
# Import Packages
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
```

In [2]:
```
#Import related libraries and Advertising_df1_raw.csv and Advertising_df2_raw.csv datc
df1 = pd.read_csv("Advertising_df1_raw.csv")
df2 = pd.read_csv("Advertising_df2_raw.csv")
```

In [3]:
```
df1.head(7)
```

Out[3]:

	Unnamed: 0	TV	Radio	Newspaper	Sales
0	1	230.1	37.8	69.2	22.1
1	2	44.5	39.3	45.1	10.4
2	3	17.2	45.9	69.3	9.3
3	4	151.5	41.3	58.5	18.5
4	5	180.8	10.8	58.4	12.9
5	6	8.7	48.9	75.0	7.2
6	7	57.5	32.8	23.5	11.8

In [4]:
```
df1.tail(7)
```

Out[4]:

	Unnamed: 0	TV	Radio	Newspaper	Sales
92	93	217.7	33.5	59.0	19.4
93	94	250.9	36.5	72.3	22.2
94	95	107.4	14.0	10.9	11.5
95	96	163.3	31.6	52.9	16.9
96	97	197.6	3.5	5.9	11.7
97	98	184.9	21.0	22.0	15.5
98	99	289.7	42.3	51.2	25.4

In [5]:
```
df2.head(7)
```

Out[5]:

	Unnamed: 0	TV	Radio	Newspaper	Sales
0	100	135.2	41.7	45.9	17.2
1	101	222.4	4.3	49.8	11.7
2	102	296.4	36.3	100.9	23.8
3	103	280.2	10.1	29.7	14.8
4	104	187.9	17.2	17.9	14.7
5	105	238.2	34.3	5.3	20.0
6	106	137.9	46.4	59.0	19.2

In [6]:
```
df2.tail(7)
```

Out[6]:

	Unnamed: 0	TV	Radio	Newspaper	Sales
97	197	94.2	4.9	8.1	9.7
98	198	177.0	9.3	6.4	12.8
99	199	283.6	42.0	66.2	25.5
100	200	232.1	8.6	8.7	13.4
101	138	273.7	28.9	59.7	20.8
102	138	273.7	28.9	59.7	20.8
103	193	17.2	4.1	31.6	5.9

In [7]:
```
#Merge these two files and create/ load the new merged file into your folder
df_merge = pd.concat([df1, df2], axis=0, join='inner')
df_merge
```

Out[7]:

	Unnamed: 0	TV	Radio	Newspaper	Sales
0	1	230.1	37.8	69.2	22.1
1	2	44.5	39.3	45.1	10.4
2	3	17.2	45.9	69.3	9.3
3	4	151.5	41.3	58.5	18.5
4	5	180.8	10.8	58.4	12.9
...	...	...	...	...	...
99	199	283.6	42.0	66.2	25.5
100	200	232.1	8.6	8.7	13.4
101	138	273.7	28.9	59.7	20.8
102	138	273.7	28.9	59.7	20.8
103	193	17.2	4.1	31.6	5.9

203 rows × 5 columns

In [8]:
```
#Remove the unnecessary columns (refer to attributes involved above).
df_merge = df_merge.drop(columns=['Unnamed: 0'])
df_merge = df_merge.reset_index(drop=True)
df_merge
```

Out[8]:

	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	9.3
3	151.5	41.3	58.5	18.5
4	180.8	10.8	58.4	12.9
...	...	...	...	...
198	283.6	42.0	66.2	25.5
199	232.1	8.6	8.7	13.4
200	273.7	28.9	59.7	20.8
201	273.7	28.9	59.7	20.8
202	17.2	4.1	31.6	5.9

203 rows × 4 columns

In [9]:
```
#Explore the merged data. Use interactive EDA using advanced libraries. Show and inter
print("No. of Attributes (Columns): ",len(df_merge.columns))
print("No. of sample (Rows): ",len(df_merge))
```

```
No. of Attributes (Columns):  4
No. of sample (Rows):  203
```

In [10]:
```
df_merge.describe()
```

Out[10]:

	TV	Radio	Newspaper	Sales
count	202.000000	202.000000	203.000000	203.000000
mean	148.009901	23.436139	30.835961	14.091133
std	86.685730	14.799103	21.764217	5.260785
min	0.700000	0.300000	0.300000	1.600000
25%	73.725000	10.025000	12.850000	10.350000
50%	150.650000	23.750000	26.400000	12.900000
75%	220.175000	36.575000	45.100000	17.800000
max	296.400000	49.600000	114.000000	27.000000

In [11]:
```
import ydata_profiling as pp
#Interactive and comprehensive EDA/ data description

# forming ProfileReport and save
# as output.html file
profile = pp.ProfileReport(df_merge)
profile.to_file("output_df_merge.html")
```

Summarize dataset: 100% 30/30 [00:01<00:00, 14.41it/s, Completed]

Generate report structure: 100% 1/1 [00:00<00:00, 1.00it/s]

Render HTML: 100% 1/1 [00:00<00:00, 2.74it/s]

Export report to file: 100% 1/1 [00:00<00:00, 117.27it/s]

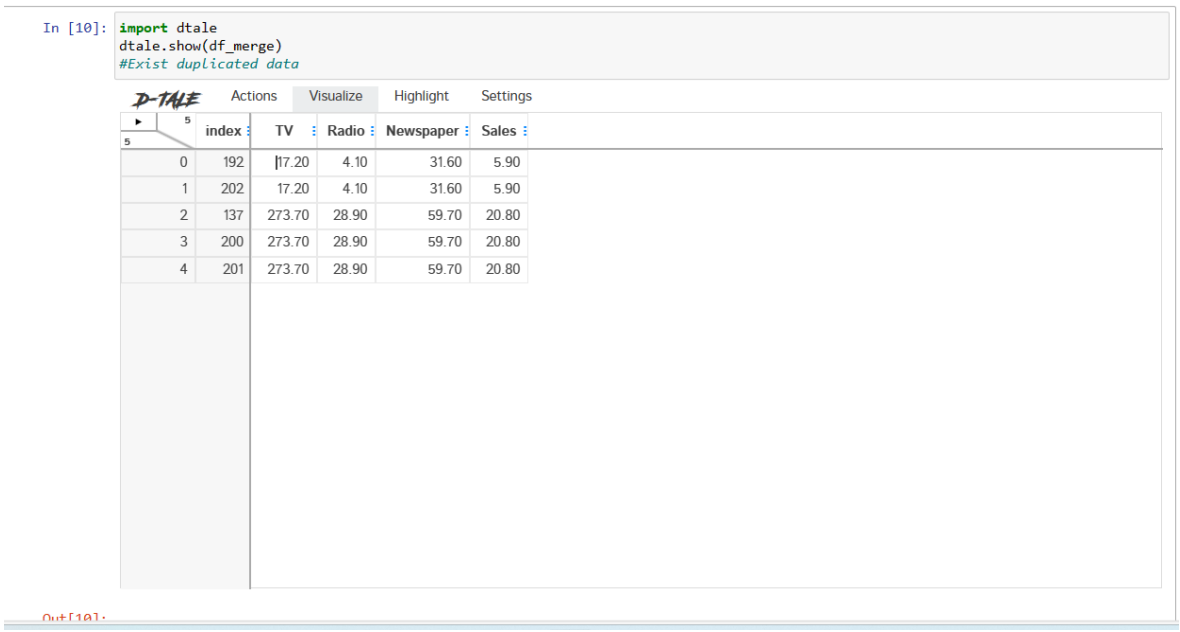
In [12]:
```
import dtale
dtale.show(df_merge)
#Exist duplicated data
```

	TV	Radio	Newspaper	Sales
0	230.10	37.80	69.20	22.10
1	44.50	39.30	45.10	10.40
2	17.20	45.90	69.30	9.30
3	151.50	41.30	58.50	18.50
4	180.80	10.80	58.40	12.90
5	8.70	48.90	75.00	7.20
6	57.50	32.80	23.50	11.80
7	120.20	19.60	11.60	13.20
8	8.60	2.10	1.00	4.80
9	199.80	2.60	21.20	10.60
10	66.10	5.80	24.20	8.60
11	214.70	24.00	4.00	17.40
12	23.80	35.10	65.90	9.20
13	97.50	7.60	7.20	9.70
14	204.10	32.90	46.00	19.00
15	195.40	47.70	52.90	22.40
16	67.80	36.60	114.00	12.50

Out[12]:

In [13]:
```
from PIL import Image
im = Image.open("dtale_missing.png")
im
```

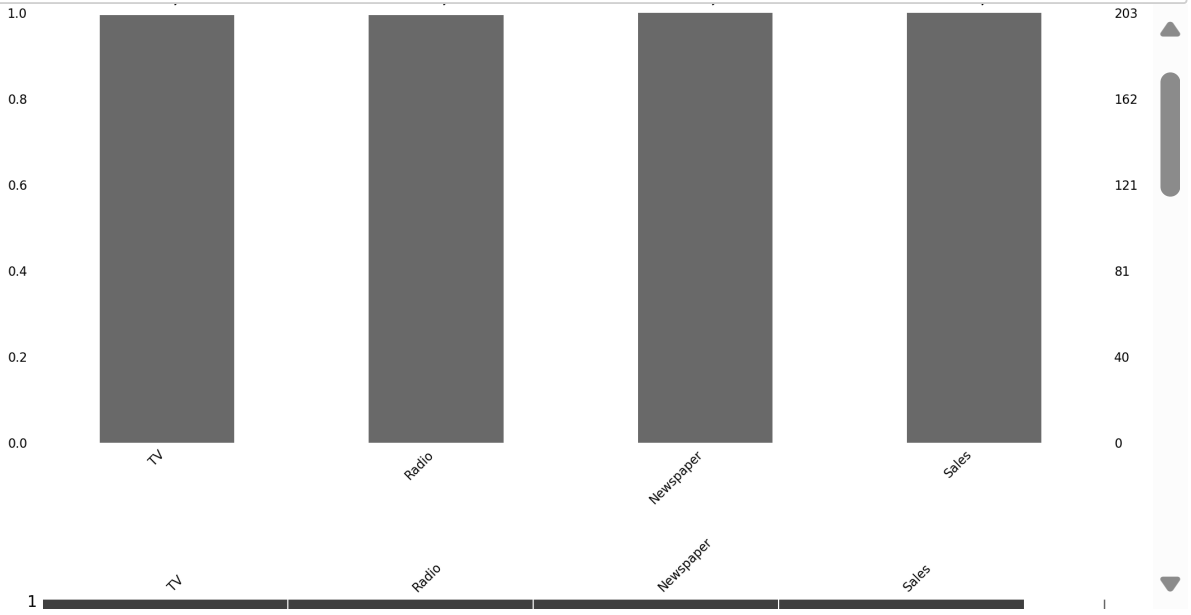
Out[13]:



In [14]:
```
df_merge.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 203 entries, 0 to 202
Data columns (total 4 columns):
 #   Column     Non-Null Count  Dtype  
---  ------     --------------  -----  
 0   TV         202 non-null    float64
 1   Radio      202 non-null    float64
 2   Newspaper  203 non-null    float64
 3   Sales      203 non-null    float64
dtypes: float64(4)
memory usage: 6.5 KB
```

In [15]:
```
#Check and treat missing values and duplicated data (if exist). Discuss them.
import missingno as msno
msno.bar(df_merge)
msno.matrix(df_merge)
msno.dendrogram(df_merge)
msno.heatmap(df_merge)
```



Missingvalues exist in column TV and Radio

In [16]:
```
df_merge.isnull().sum()
```

Out[16]:
```
TV           1
Radio        1
Newspaper    0
Sales        0
dtype: int64
```

In [17]:
```
#Find missing value position
trace_missing_TV = pd.isnull(df_merge['TV'])
df_merge[trace_missing_TV]
```

Out[17]:

	TV	Radio	Newspaper	Sales
115	NaN	35.0	52.7	12.6

In [18]:
```
trace_missing_Radio = pd.isnull(df_merge['Radio'])
df_merge[trace_missing_Radio]
```

Out[18]:

	TV	Radio	Newspaper	Sales
113	209.6	NaN	10.7	15.9

In [19]:
```
#Treating Missing Values
df_merge.TV = df_merge.TV.fillna(df_merge.TV.mean())
df_merge.Radio = df_merge.Radio.fillna(df_merge.Radio.mean())
```

Fill the missing values with mean since it only 1 column missing

In [20]:
```
#Recheck
df_merge.isnull().sum()
```

Out[20]:
```
TV           0
Radio        0
Newspaper    0
Sales        0
dtype: int64
```

In [21]:
```
#Check duplicated rows
df_merge[df_merge.duplicated(keep=False)]
```

Out[21]:

	TV	Radio	Newspaper	Sales
137	273.7	28.9	59.7	20.8
192	17.2	4.1	31.6	5.9
200	273.7	28.9	59.7	20.8
201	273.7	28.9	59.7	20.8
202	17.2	4.1	31.6	5.9

In [22]:
```
#Remove duplicates and keep the first
df_merge.drop_duplicates(keep='first', inplace = True)
```

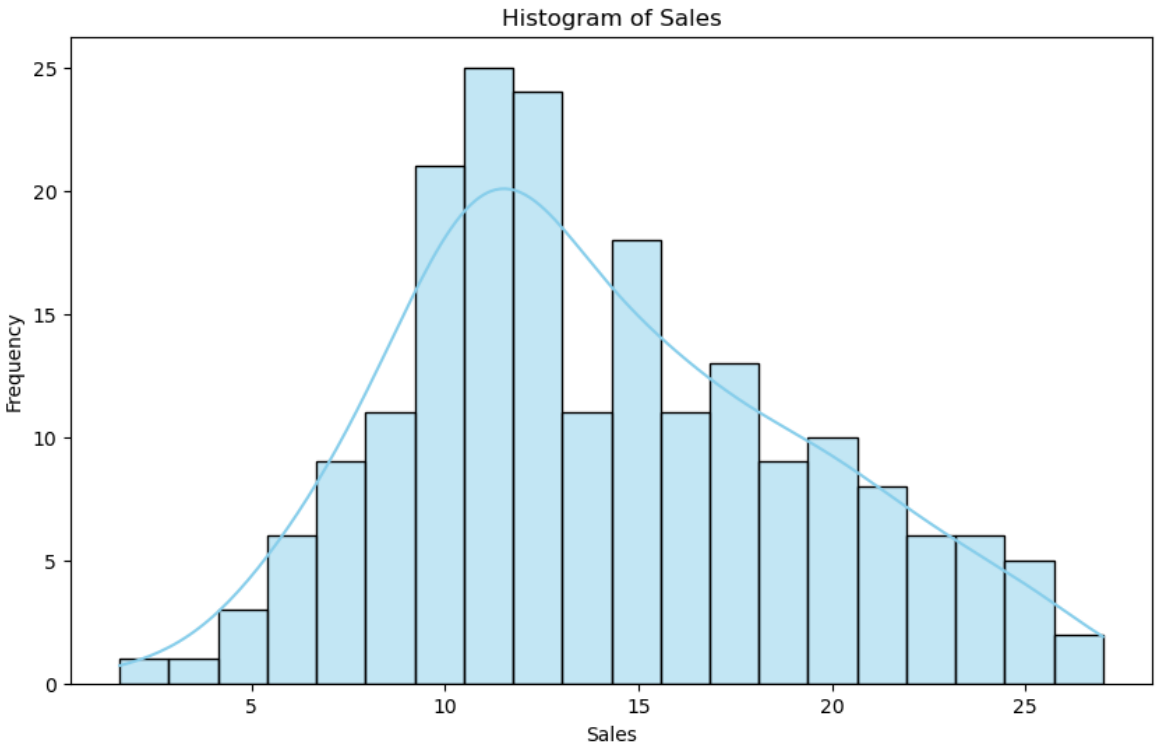
In [23]:
```
#Load new clean data into your folder
df_merge_clean=df_merge.copy()
df_merge_clean
```

Out[23]:

	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	9.3
3	151.5	41.3	58.5	18.5
4	180.8	10.8	58.4	12.9
...	...	...	...	...
195	38.2	3.7	13.8	7.6
196	94.2	4.9	8.1	9.7
197	177.0	9.3	6.4	12.8
198	283.6	42.0	66.2	25.5
199	232.1	8.6	8.7	13.4

200 rows × 4 columns

In [24]:
```
#Construct the histogram for the Sales attribute. Discuss the shape of the data distri
# Constructing the histogram for the 'Sales' attribute
plt.figure(figsize=(10, 6))
sns.histplot(df_merge_clean['Sales'], bins=20, kde=True, color='skyblue')
plt.title('Histogram of Sales')
plt.xlabel('Sales')
plt.ylabel('Frequency')
plt.show()
```

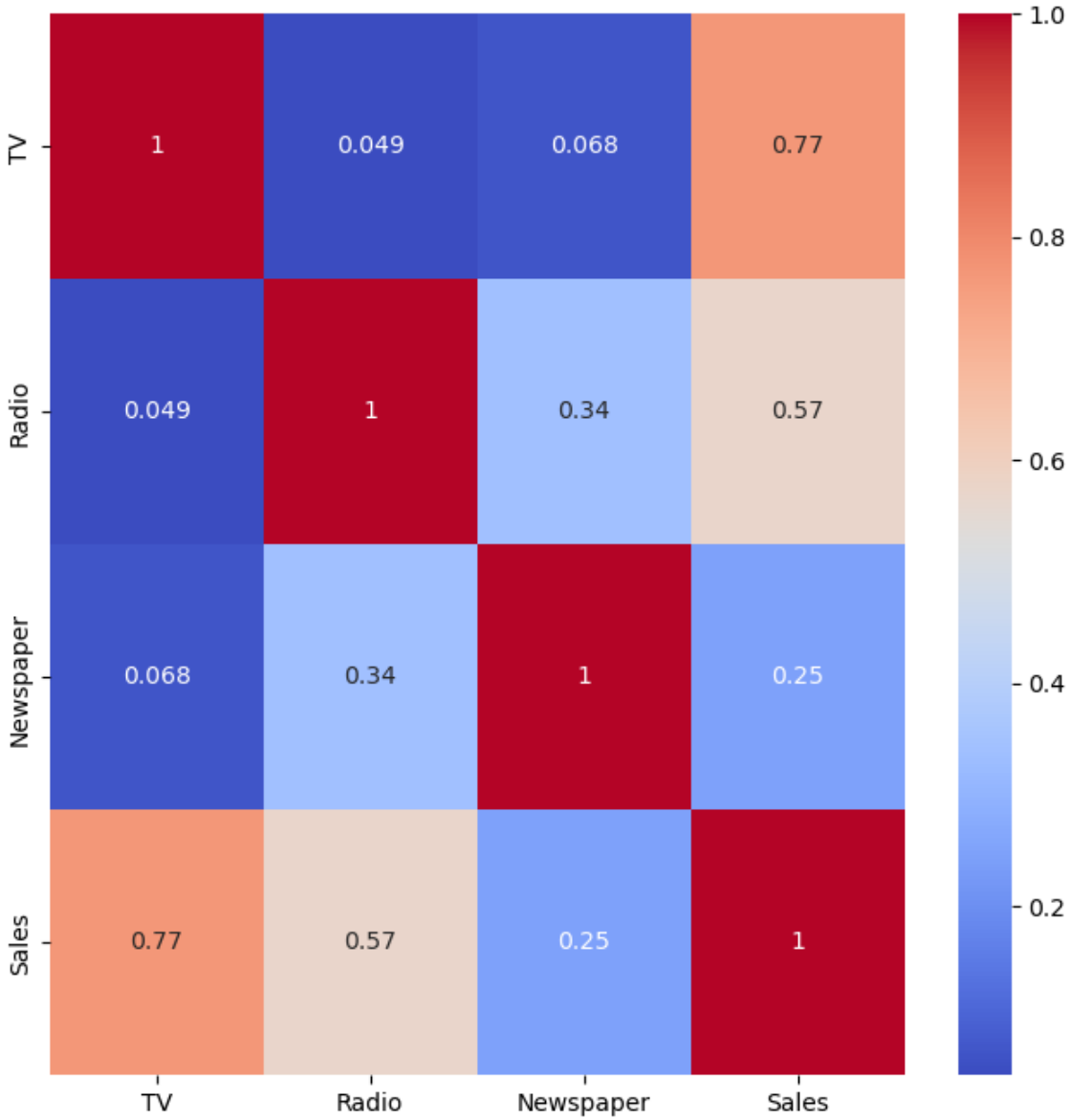


Based on the shape of the data distribution,it shows that the sales is right skewed.

In [25]:
```
#Plot the three-correlation graphs for TV vs Sales, Radio vs Sales and Newspaper vs Sa
# Correlation
df_merge_clean.corr()

# Heatmap Correlation
plt.figure(figsize=(8,8))
sns.heatmap(df_merge_clean.corr(), annot=True, cmap='coolwarm')
```

Out[25]: <AxesSubplot: >



Based on the correlation heatmap,

A correlation coefficient of 0.77 indicates a strong positive linear relationship between TV advertising spending and Sales.

A correlation coefficient of 0.57 indicates a moderate positive linear relationship between Radio advertising spending and Sales.

A correlation coefficient of 0.25 suggests a weak positive linear relationship between Newspaper advertising spending and Sales.

Hence, we conclude that TV advertising have the most influential, with a strong positive correlation with Sales.

In [ ]: