SD21063 TEAN JIN HE Data Mining Lab Report 1

November 20, 2023

1 Data Mining Lab Report 1

NAME: TEAN JIN HE

MATRIC ID: SD21063¶

SECTION: 02G

1.0.1 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

2 Question 1

2.0.1 Discuss the ETL concept related to the case study above.

Extract, transform, and load called ETL is a technique for integrating data from several sources into a single one.

The process of continuously extracting data from several sources is automated in the extract stage. In this instance, the two CSV files are for preliminary development purposes; we automate the extraction procedure to establish a more dependable and effective workflow.

Data processing is done on the raw data during the staging transform. In this instance, the data is combined and converted for the specific analytical use case in mind. In this instance, we verify that the data match the format, much like the immune system does. In the event that it is inconsistent, we sterilise it. If there is a common perspective among the data, we may occasionally combine the data.

The cleansed data is finally stored in the load stage. In this situation, the data was cleaned and then saved into a different CSV file.

3 Question 2

3.1 Python: Data Preparation

4 Import related libraries and datasets

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

5 Read first 7 and last 7 rows

[2]:
```
# Remove the unnecessary columns
df1 = pd.read_csv('Advertising_df1_raw.csv')
df1.head(7)
```

[2]:

	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

[3]:
```
df1.tail(7)
```

[3]:

	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

[4]:
```
# Remove the unnecessary columns
df2= pd.read_csv('Advertising_df2_raw.csv')
df2.head(7)
```

[4]:

	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
```

[5]:
```
df2.tail(7)
```

[5]:
```
     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
```

6 Merge these two files and create

[6]:
```
import pandas as pd
df_Advertising=pd.concat([df1,df2])
df_Advertising.head()
```

[6]:
```
   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
```

7 Load the new merged file into your folder

[7]:
```
df_Advertising.to_csv('Merge_Advertising.csv')
df_Advertising
```

[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 x 5 columns]
```

8 Remove the unnecessary columns

[8]:
```
df_Advertising = df_Advertising.drop('Unnamed: 0', axis = 1)
df_Advertising
```

[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
..     …     …          …      …
99   283.6   42.0       66.2   25.5
100  232.1    8.6        8.7   13.4
101  273.7   28.9       59.7   20.8
102  273.7   28.9       59.7   20.8
103   17.2    4.1       31.6    5.9

[203 rows x 4 columns]
```

9 Explore the merged data and Interpret

[9]:
```
print('No of attributes: ', len(df_Advertising.columns))
```

```
No of attributes:  4
```

There are 4 attributes which are TV, Radio, Newspaper and Sales.

[10]:
```
print('No of rows: ', len(df_Advertising))
```

```
No of rows:  203
```

There are total of 203 rows data.

[11]:
```
print(df_Advertising.info())
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 203 entries, 0 to 103
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: 7.9 KB
None
```

[12]:
```
df_Advertising.describe()
```

[12]:

	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

Based on the summary statistics of all the numerical variables like the mean, median (50%), minimum values and maximum values which are along with the standard deviation. Then, We can also calculate the IQR using the 25th and 75th percentile values.

[33]:
```
import ydata_profiling as pp
#Interactive and comprehensive EDA/ data description

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

```
D:\anaconda\Lib\site-packages\numba\core\decorators.py:262:
NumbaDeprecationWarning:

numba.generated_jit is deprecated. Please see the documentation at:

https://numba.readthedocs.io/en/stable/reference/deprecation.html#deprecation-

of-generated-jit for more information and advice on a suitable replacement.

D:\anaconda\Lib\site-packages\visions\backends\shared\nan_handling.py:50:
NumbaDeprecationWarning:

The 'nopython' keyword argument was not supplied to the 'numba.jit'

decorator. The implicit default value for this argument is currently False, but

it will be changed to True in Numba 0.59.0. See

https://numba.readthedocs.io/en/stable/reference/deprecation.html#deprecation-

of-object-mode-fall-back-behaviour-when-using-jit for details.

2023-11-20 09:38:28,270 - INFO     - Pandas backend loaded 1.5.3
2023-11-20 09:38:28,280 - INFO     - Numpy backend loaded 1.23.5
2023-11-20 09:38:28,282 - INFO     - Pyspark backend NOT loaded
```

```
2023-11-20 09:38:28,282 - INFO     - Python backend loaded
Summarize dataset:   0%|          | 0/5 [00:00<?, ?it/s]
Generate report structure:   0%|          | 0/1 [00:00<?, ?it/s]
Render HTML:   0%|          | 0/1 [00:00<?, ?it/s]
Export report to file:   0%|          | 0/1 [00:00<?, ?it/s]
```

10 D-Tale

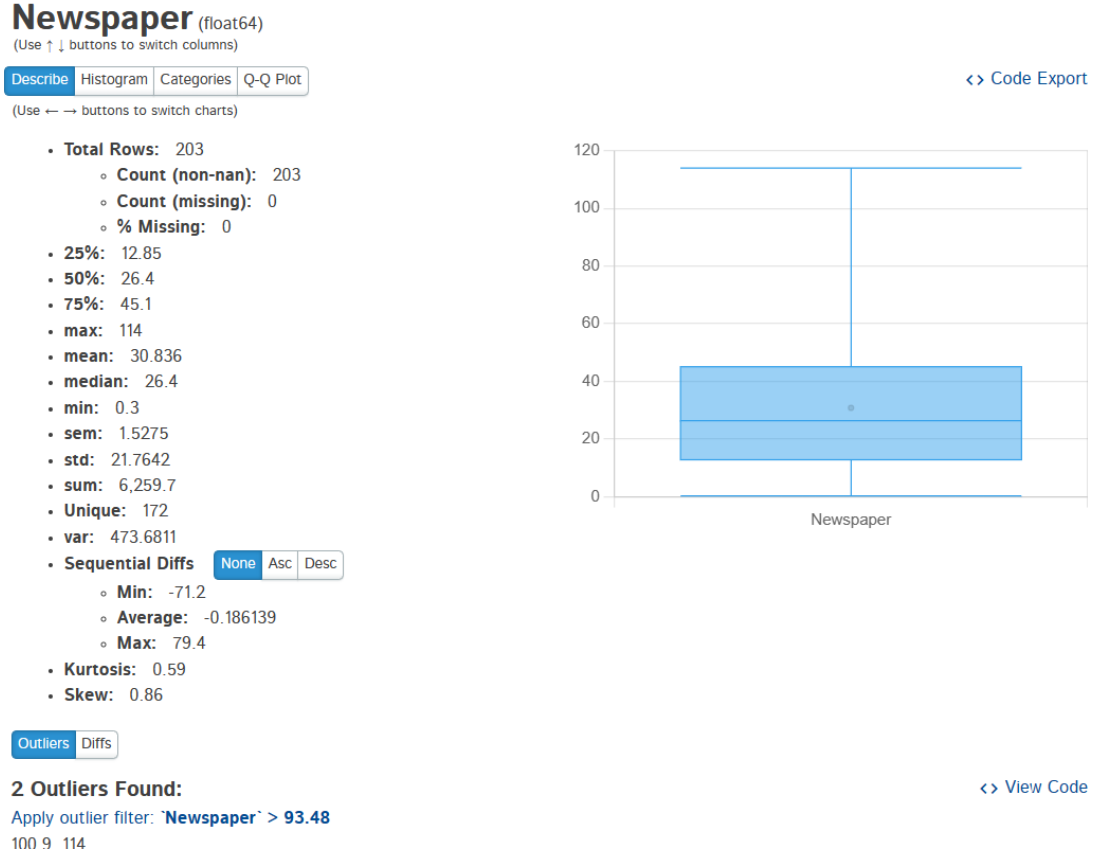
[14]:
```
import dtale

dtale.show(df_Advertising)
```

```
<IPython.lib.display.IFrame at 0x12aebdb3750>
```

[14]:

[41]:
```
from PIL import Image
im = Image.open("outliers.png")
im
```

[41]:



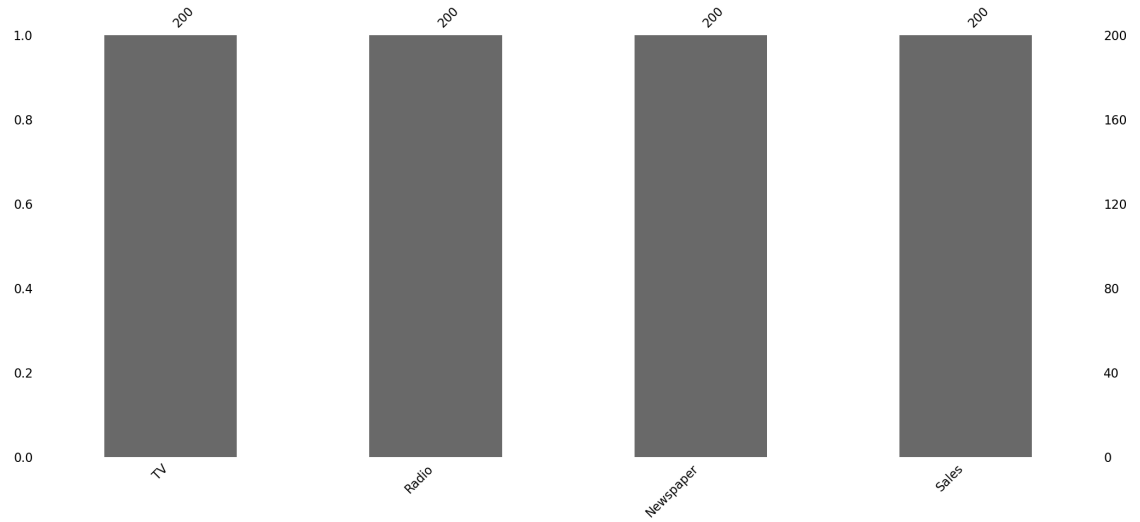
Based on the observation, we can see that there are two missing values in newspaper which are 100.9 and 114.

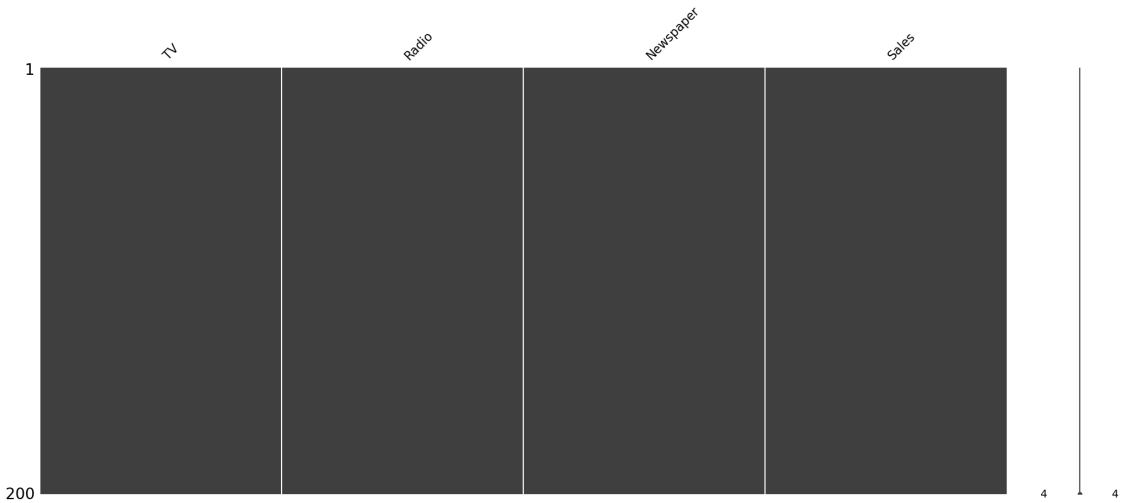
11 Missingno

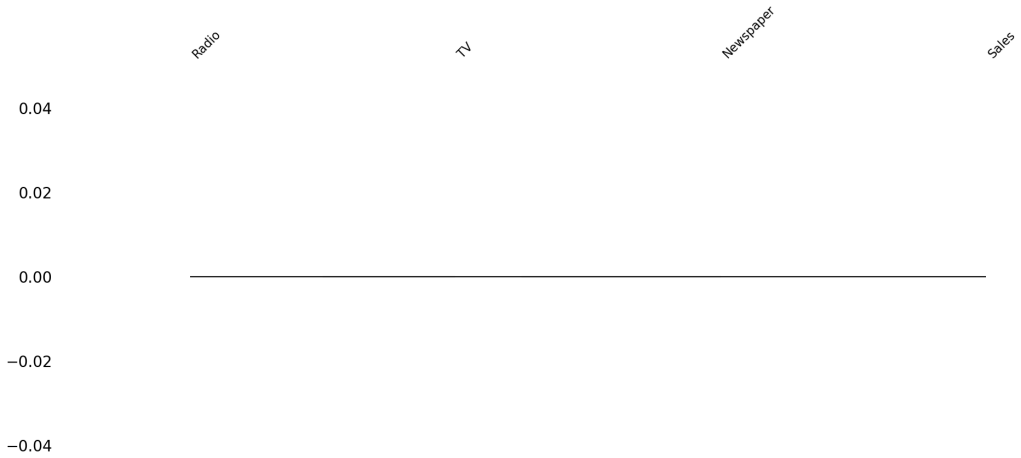
[34]:
```
import missingno as msno

msno.bar(df_Advertising)
msno.matrix(df_Advertising)
msno.dendrogram(df_Advertising)
msno.heatmap(df_Advertising)
```

```
D:\anaconda\Lib\site-packages\scipy\cluster\hierarchy.py:2846: UserWarning:

Attempting to set identical low and high ylims makes transformation singular;
automatically expanding.

D:\anaconda\Lib\site-packages\seaborn\matrix.py:309: UserWarning:

Attempting to set identical low and high xlims makes transformation singular;
automatically expanding.

D:\anaconda\Lib\site-packages\seaborn\matrix.py:309: UserWarning:

Attempting to set identical low and high ylims makes transformation singular;
automatically expanding.
```

[34]:
```
<AxesSubplot: >
```




TV
Radio
Newspaper
Sales
1
200
4
4


Radio
TV
Newspaper
Sales
0.04
0.02
0.00
−0.02
−0.04

12 SweetViz

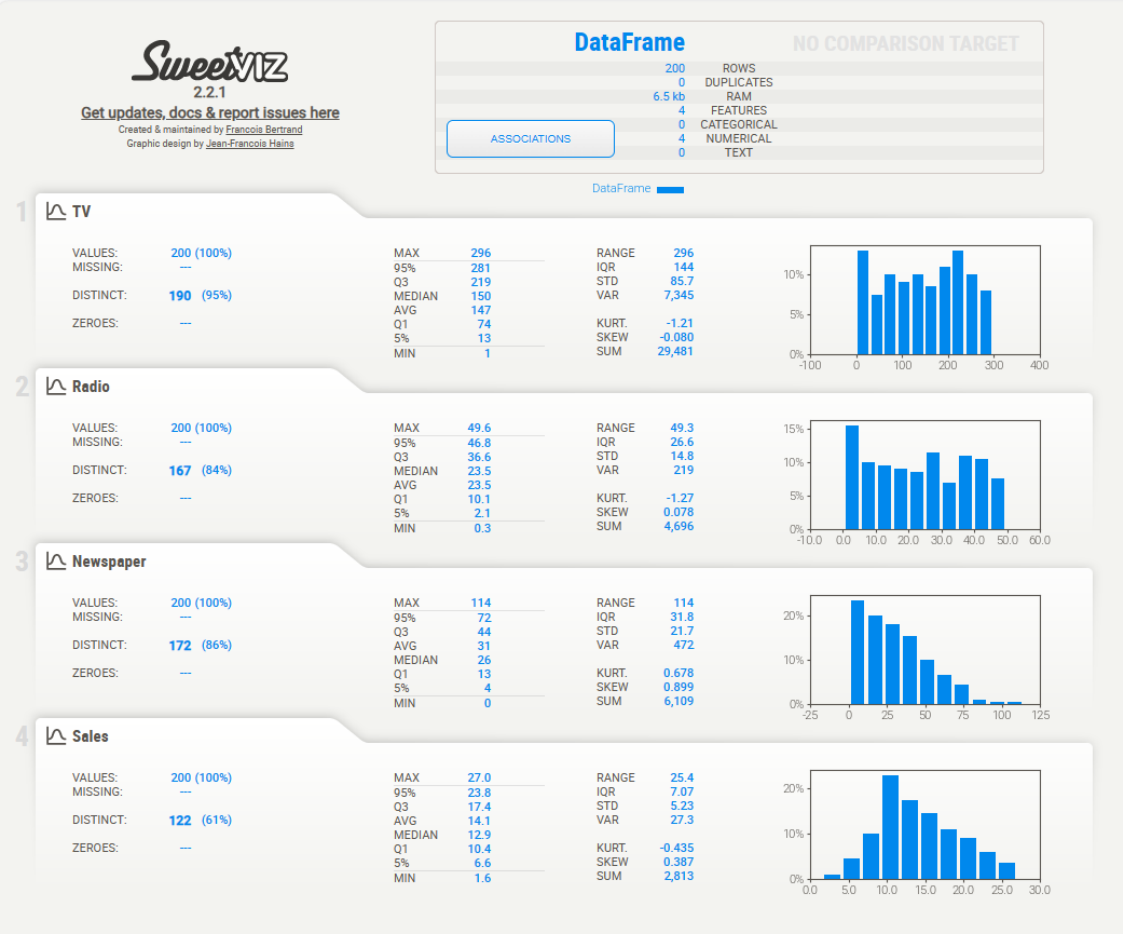
[42]:
```
import sweetviz as sv

report = sv.analyze(df_Advertising)
report.show_html()
```

```
                                             |                                       ␣
↪           | [  0%]   00:00 ->…
```

```
Report SWEETVIZ_REPORT.html was generated! NOTEBOOK/COLAB USERS: the web browser
MAY not pop up, regardless, the report IS saved in your notebook/colab files.
```

[43]:
```
from PIL import Image
im1 = Image.open("sweetviz.png")
im1
```

[43]:



Based on this html, we can see TV has the maximum values which having 296 higher than Radio (49.6), Newspaper(114) and Sales(27). Other than that, based on the histogram, TV is the bimodal distribution and Newspaper and Sales are the right skewed distribution.

[37]:
```
import sketch

df_Advertising.sketch.ask('What are the max values of each numerical column?')
```

```
<IPython.core.display.HTML object>
```

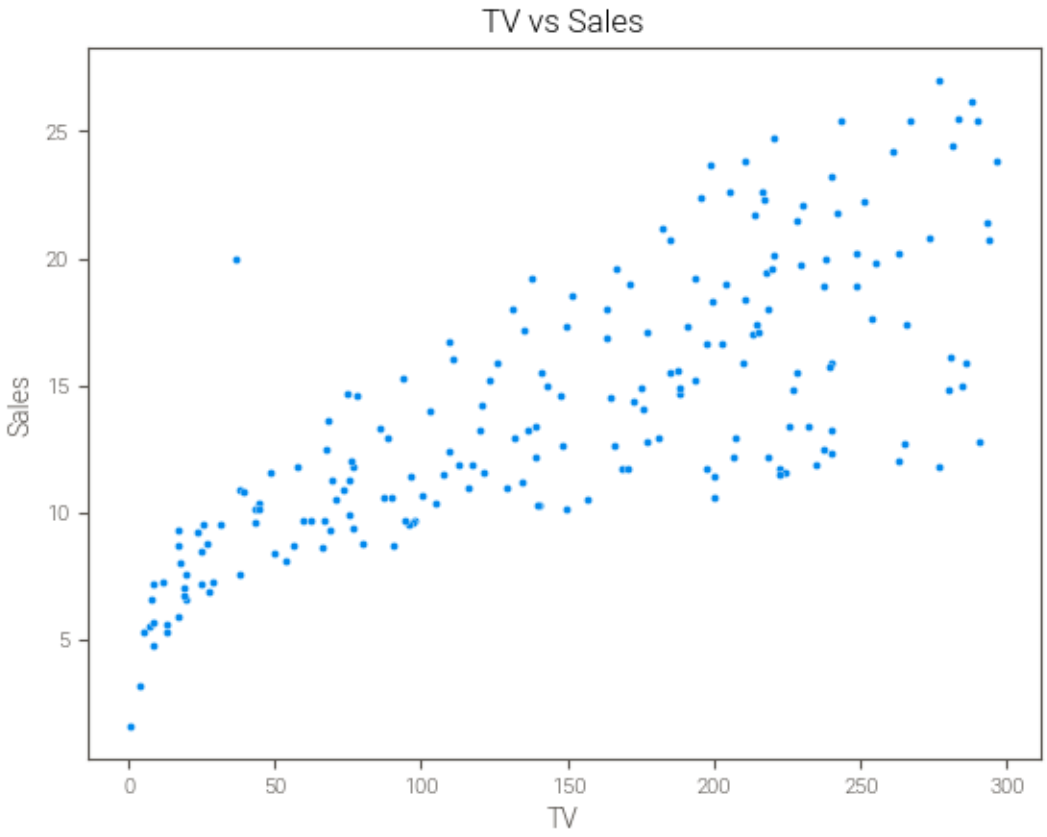
[38]:
```
df_Advertising.sketch.howto("""How do I plot KM against Price
                using a scatter plot and have the coloured in rainbow?""")
```

```
<IPython.core.display.HTML object>
```

[44]:
```
import matplotlib.pyplot as plt
import seaborn as sns

# Create a scatter plot of KM against Price
```

```
sns.scatterplot(x='TV', y='Sales', data=df_Advertising, palette='rainbow')

# Add labels and title
plt.title('TV vs Sales')
plt.xlabel('TV')
plt.ylabel('Sales')

# Show the plot
plt.show()
```

```
C:\Users\user\AppData\Local\Temp\ipykernel_8084\1406416016.py:5: UserWarning:

Ignoring `palette` because no `hue` variable has been assigned.
```



Based on the plot, there is a positive correlation distribution between TV vs Sales.

13 Check and treat missing values

[16]:
```
df_Advertising[df_Advertising.isnull().any(axis=1)]
```

[16]:
```
      TV  Radio  Newspaper  Sales
14  209.6    NaN       10.7   15.9
16    NaN   35.0       52.7   12.6
```

There are 2 missing values which are row 14 and row 16.

[17]:
```
#Showing the data infomation there is how many null values
df_Advertising.isnull().sum()
```

[17]:
```
TV           1
Radio        1
Newspaper    0
Sales        0
dtype: int64
```

[18]:
```
# To find the location of missing values in row
trace_missing_TV = pd.isnull(df_Advertising['TV'])
df_Advertising[trace_missing_TV]
```

[18]:
```
     TV  Radio  Newspaper  Sales
16 NaN   35.0       52.7   12.6
```

[19]:
```
trace_missing_Radio = pd.isnull(df_Advertising['Radio'])
df_Advertising[trace_missing_Radio]
```

[19]:
```
      TV  Radio  Newspaper  Sales
14  209.6    NaN       10.7   15.9
```

[20]:
```
#filling missing values
df_Advertising.TV = df_Advertising.TV.fillna(df_Advertising.TV.mean())
```

[21]:
```
df_Advertising.Radio = df_Advertising.Radio.fillna(df_Advertising.Radio.mean())
```

[22]:
```
df_Advertising.isnull().sum()
```

[22]:
```
TV           0
Radio        0
Newspaper    0
Sales        0
dtype: int64
```

14 Check for the duplicated rows

[23]:
```
df_Advertising[df_Advertising.duplicated(keep=False)]
```

[23]:
```
        TV  Radio  Newspaper  Sales
38   273.7   28.9       59.7   20.8
93    17.2    4.1       31.6    5.9
101  273.7   28.9       59.7   20.8
102  273.7   28.9       59.7   20.8
103   17.2    4.1       31.6    5.9
```

[24]:
```
# Remove the duplicated data and keep first
df_Advertising.drop_duplicates(keep='first', inplace=True)
```

[25]:
```
# Recheck
df_Advertising[df_Advertising.duplicated(keep=False)]
```

[25]:
```
Empty DataFrame
Columns: [TV, Radio, Newspaper, Sales]
Index: []
```

15 Load new clean data into your folder

[26]:
```
df_Advertising = df_Advertising.reset_index(drop=True)
```

[27]:
```
# Copy the cleaned data into new data frame
df_Advertising_clean = df_Advertising.copy()
df_Advertising_clean
```

[27]:
```
        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 x 4 columns]
```

[28]:
```
# Either way, you may create/ load new dataset for data_clean
df_Advertising.to_csv('df_Advertising_clean.csv', index=False)
```

[29]:
```
df_Advertising_clean = pd.read_csv('df_Advertising_clean.csv')
df_Advertising_clean.head()
```

[29]:

	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

[30]:
```
df_Advertising_clean.describe()
```

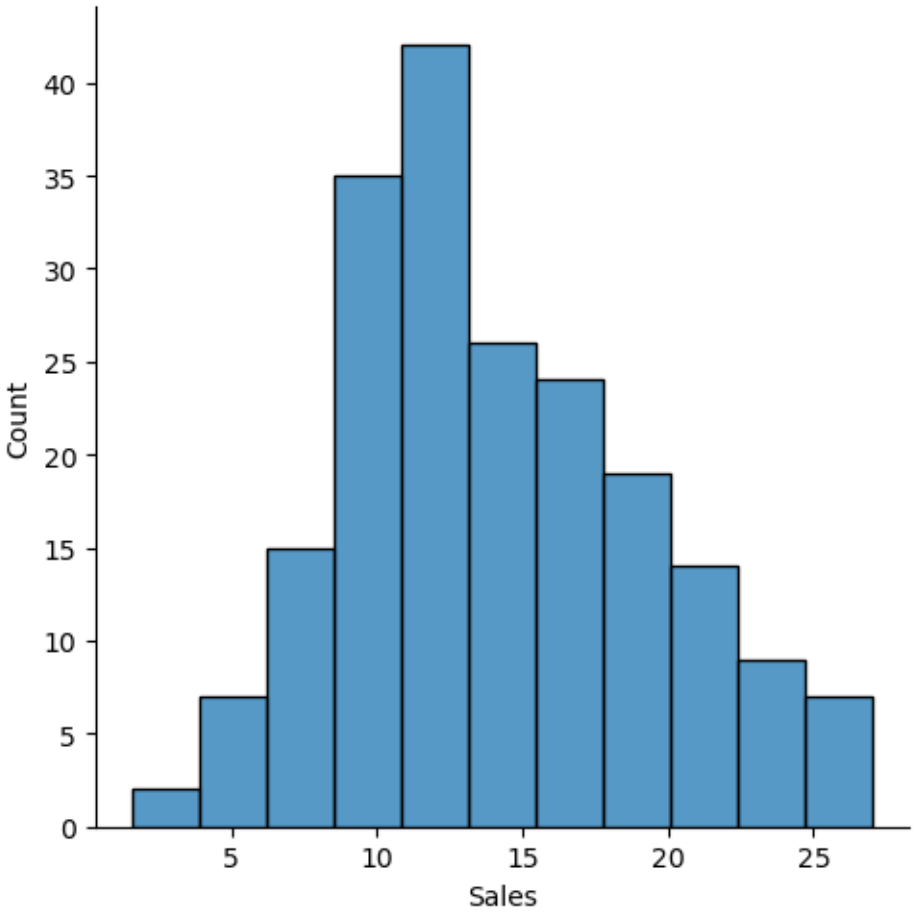
[30]:

	TV	Radio	Newspaper	Sales
count	200.000000	200.000000	200.000000	200.000000
mean	147.407050	23.478181	30.543500	14.065000
std	85.701881	14.799796	21.733844	5.225217
min	0.700000	0.300000	0.300000	1.600000
25%	74.375000	10.075000	12.750000	10.375000
50%	149.750000	23.518069	26.050000	12.900000
75%	218.825000	36.650000	44.500000	17.450000
max	296.400000	49.600000	114.000000	27.000000

16 Construct the histogram for the Sales attribute

[31]:
```
import seaborn as sns # interactive visualisation
sns.displot(df_Advertising_clean['Sales'])
```

[31]:
```
<seaborn.axisgrid.FacetGrid at 0x12aebdf7e50>
```

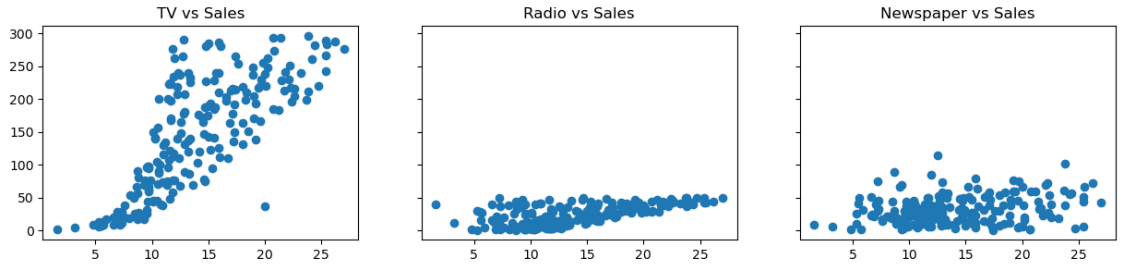


Based on the histogram above, the shape of the data distribution is right skewed distributions which the peak of the graph lies on the left side of the center. So, it can be say as a positively skewed histogram.

17 Plot the three-correlation graphs for TV vs Sales, Radio vs Sales and Newspaper vs Sales

[32]:
```
# Plotting TV vs Sales, Radio vs Sales and Newspaper vs Sales
import matplotlib.pyplot as plt
fig, (ax1, ax2, ax3) = plt.subplots(1, 3, sharey = True, figsize=(15,3))
ax1.scatter(df_Advertising_clean['Sales'], df_Advertising_clean['TV'])
ax1.set_title('TV vs Sales')
ax2.scatter(df_Advertising_clean['Sales'], df_Advertising_clean['Radio'])
ax2.set_title('Radio vs Sales')
ax3.scatter(df_Advertising_clean['Sales'], df_Advertising_clean['Newspaper'])
ax3.set_title('Newspaper vs Sales')
```

[32]: `Text(0.5, 1.0, 'Newspaper vs Sales')`



Based on the graph above, the graph of TV vs Sales has a strong positive linear association between the two variables with a potential outliers. The graph of Radio vs Sales has no correlation linear association between two variables. The graph of Newspaper vs Sales is no correlation between this two variables.

[ ]: