importing required packages

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import sklearn as sk
import seaborn as sns
import warnings
import scipy
import numpy as np
from sklearn.preprocessing import MinMaxScaler
import seaborn as sns
import matplotlib.pyplot as plt
warnings.filterwarnings('ignore')
```

loading dataset

```
sales=pd.read_csv('Sales.csv')
print(sales)
```

```
        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
..     ...    ...        ...    ...
195   38.2    3.7       13.8    7.6
196   94.2    4.9        8.1   14.0
197  177.0    9.3        6.4   14.8
198  283.6   42.0       66.2   25.5
199  232.1    8.6        8.7   18.4

[200 rows x 4 columns]
```

reading dataset

```
sales.head(10)
```

```
      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
5    8.7   48.9       75.0    7.2
6   57.5   32.8       23.5   11.8
7  120.2   19.6       11.6   13.2
8    8.6    2.1        1.0    4.8
9  199.8    2.6       21.2   15.6
```

```
sales.tail(10)
```

```
        TV  Radio  Newspaper  Sales
190   39.5   41.1        5.8   10.8
191   75.5   10.8        6.0   11.9
192   17.2    4.1       31.6    5.9
193  166.8   42.0        3.6   19.6
194  149.7   35.6        6.0   17.3
195   38.2    3.7       13.8    7.6
196   94.2    4.9        8.1   14.0
197  177.0    9.3        6.4   14.8
198  283.6   42.0       66.2   25.5
199  232.1    8.6        8.7   18.4
```

statistical analysis

```
sales.describe()
```

```
                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
```

```
sales.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
```

```
sales.shape
```

```
(200, 4)
```

DATASET PREPROCESSING

finding null values

```
sales.isnull()
```

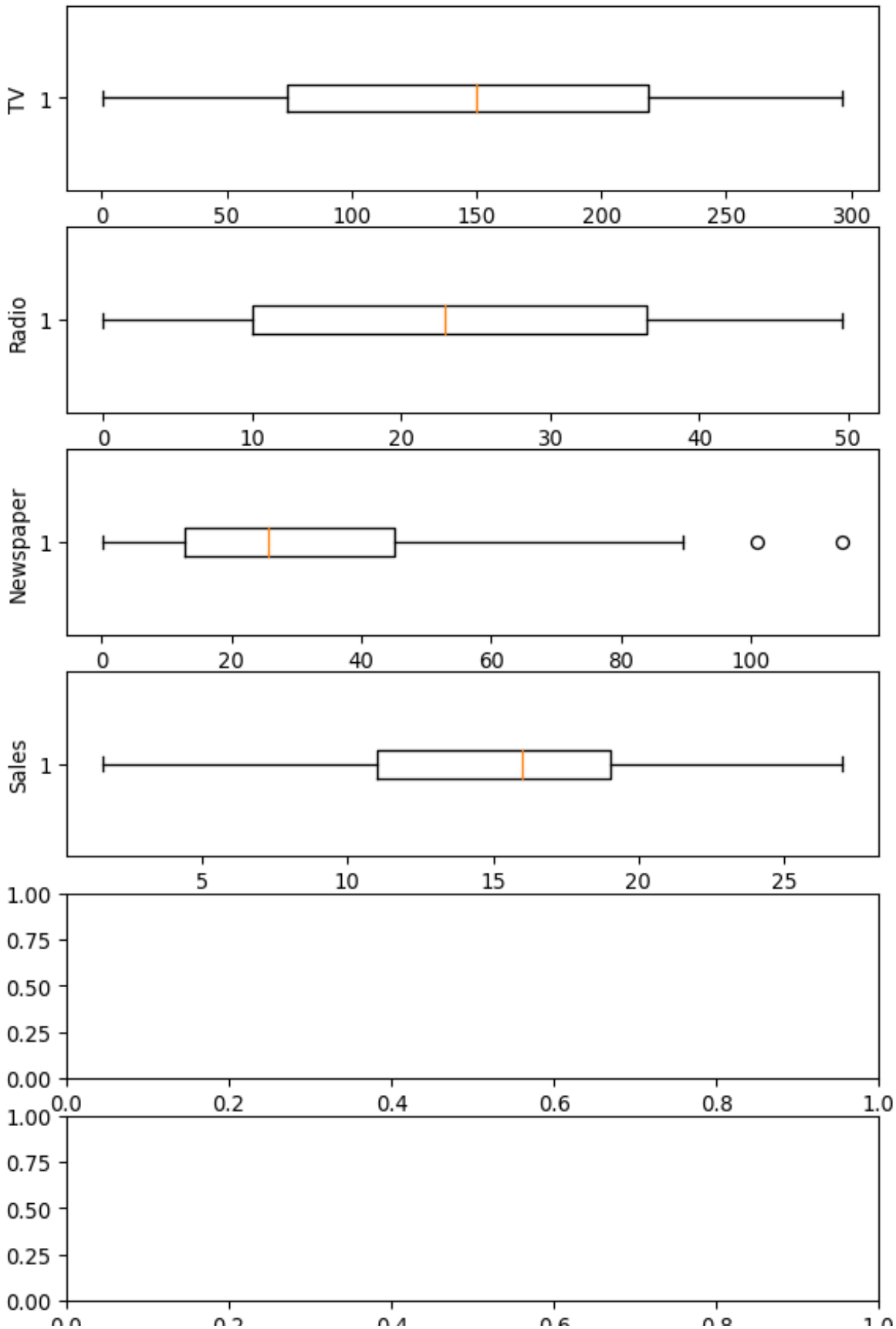
```
        TV  Radio  Newspaper  Sales
0    False  False      False  False
1    False  False      False  False
2    False  False      False  False
3    False  False      False  False
4    False  False      False  False
..     ...    ...        ...    ...
195  False  False      False  False
196  False  False      False  False
197  False  False      False  False
198  False  False      False  False
199  False  False      False  False

[200 rows x 4 columns]
```

```
sales.isna().sum()
```

```
TV           0
Radio        0
Newspaper    0
Sales        0
dtype: int64
```

checking for outliers

```
fig, axs = plt.subplots(9,1,dpi=95, figsize=(7,17))
i = 0
for col in sales.columns:
    axs[i].boxplot(sales[col], vert=False)
    axs[i].set_ylabel(col)
    i+=1
plt.show()
```

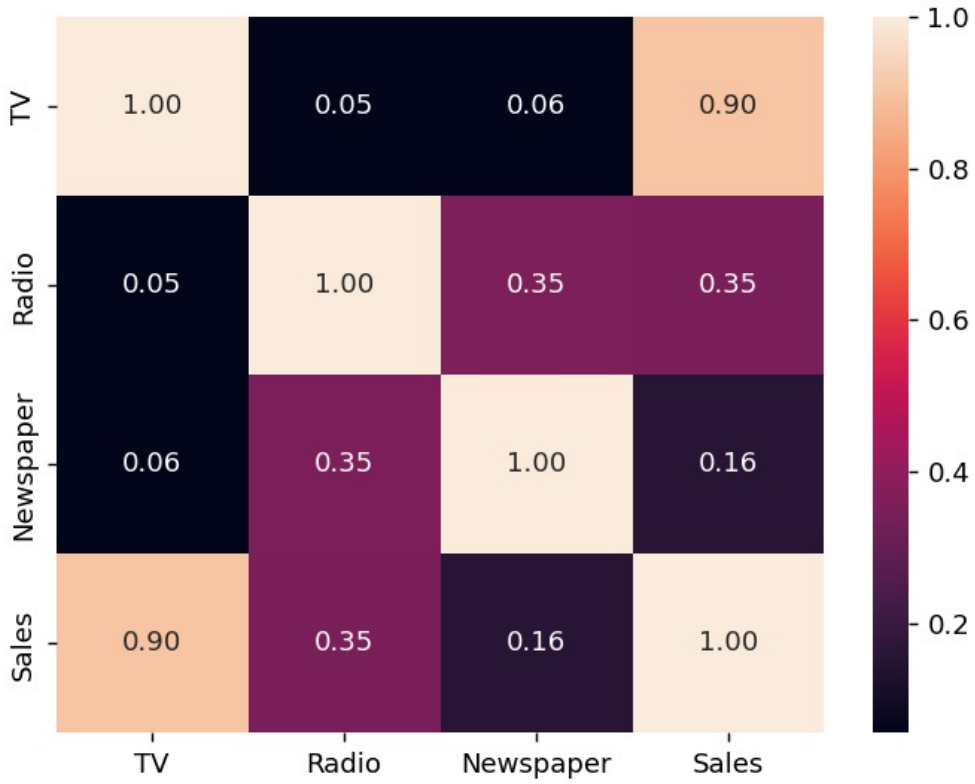

TV
1
0
50
100
150
200
250
300
Radio
1
0
10
20
30
40
50
Newspaper
1
0
20
40
60
80
100
Sales
1
5
10
15
20
25
1.00
0.75
0.50
0.25
0.00
0.0
0.2
0.4
0.6
0.8
1.0
1.00
0.75
0.50
0.25
0.00

drop the outliers

```
q1, q3 = np.percentile(sales['TV'], [25, 75])
iqr = q3 - q1
lower_bound = q1 - (1.5 * iqr)
upper_bound = q3 + (1.5 * iqr)

# Drop the outliers
clean_data = sales[(sales['TV'] >= lower_bound)
                & (sales['TV'] <= upper_bound)]

q1, q3 = np.percentile(clean_data['Radio'], [25, 75])
iqr = q3 - q1
lower_bound = q1 - (1.5 * iqr)
upper_bound = q3 + (1.5 * iqr)

# Drop the outliers
clean_data = clean_data[(clean_data['Radio'] >= lower_bound)
                        & (clean_data['Radio'] <= upper_bound)]
q1, q3 = np.percentile(clean_data['Newspaper'], [25, 75])
iqr = q3 - q1
lower_bound = q1 - (1.5 * iqr)
upper_bound = q3 + (1.5 * iqr)

# Drop the outliers
clean_data = clean_data[(clean_data['Newspaper'] >= lower_bound)
                        & (clean_data['Newspaper'] <= upper_bound)]

q1, q3 = np.percentile(clean_data['Sales'], [25, 75])
iqr = q3 - q1
lower_bound = q1 - (1.5 * iqr)
upper_bound = q3 + (1.5 * iqr)

# Drop the outliers
clean_data = clean_data[(clean_data['Sales'] >= lower_bound)
                        & (clean_data['Sales'] <= upper_bound)]
```

finding correlation

```
corr = sales.corr()

plt.figure(dpi=130)
sns.heatmap(sales.corr(), annot=True, fmt= '.2f')
plt.show()
```


TV
Radio
Newspaper
Sales
1.00
0.05
0.06
0.90
0.05
1.00
0.35
0.35
0.06
0.35
1.00
0.16
0.90
0.35
0.16
1.00
TV
Radio
Newspaper
Sales
1.0
0.8
0.6
0.4
0.2