In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings("ignore")
```

In [2]:
```
df=pd.read_csv("Sales.csv")
```

In [3]:
```
df
```

Out[3]:

	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 × 4 columns

In [4]:
```
df.head()
```

Out[4]:

	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

In [5]:
```
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
```

In [6]:
```
df.describe()
```

Out[6]:

	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

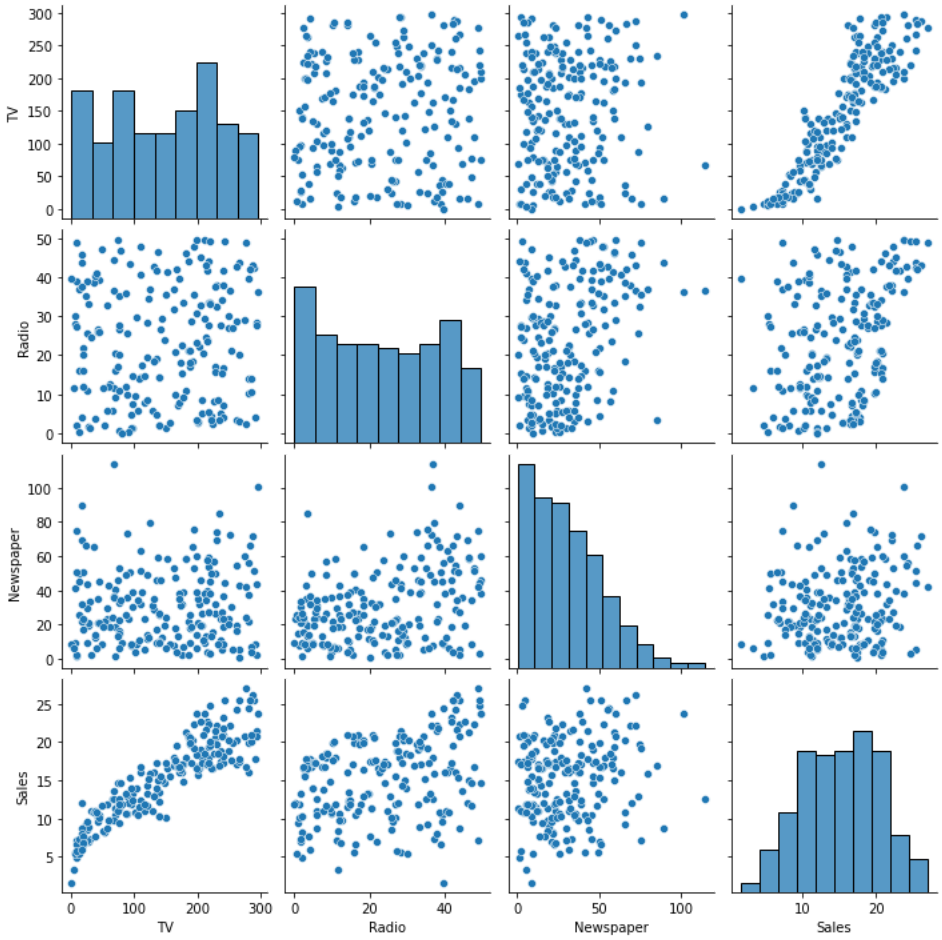
In [7]:
```
df.corr()
```

Out[7]:

	TV	Radio	Newspaper	Sales
TV	1.000000	0.054809	0.056648	0.901208
Radio	0.054809	1.000000	0.354104	0.349631
Newspaper	0.056648	0.354104	1.000000	0.157960
Sales	0.901208	0.349631	0.157960	1.000000

In [8]:
```
sns.pairplot(df)
```

Out[8]:
```
<seaborn.axisgrid.PairGrid at 0x1999d51caf0>
```



In [9]:
```
x=df.iloc[:,:-1]
```

In [10]:
```
x
```

Out[10]:

	TV	Radio	Newspaper
0	230.1	37.8	69.2
1	44.5	39.3	45.1
2	17.2	45.9	69.3
3	151.5	41.3	58.5
4	180.8	10.8	58.4
...	...	...	...
195	38.2	3.7	13.8
196	94.2	4.9	8.1
197	177.0	9.3	6.4
198	283.6	42.0	66.2
199	232.1	8.6	8.7

200 rows × 3 columns

In [11]:
```
y=df["Sales"]
```

In [12]:
```
y
```

Out[12]:
```
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
```

In [13]:
```
from sklearn.model_selection import train_test_split
```

In [14]:
```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=1)
```

In [15]:
```
#creating the MLmodel
from sklearn.linear_model import LinearRegression
```

In [16]:
```
#creating instance of model /inheritance of model
lr=LinearRegression()
```

In [17]:
```
#train the model
lr.fit(x_train,y_train)
```

Out[17]:
```
LinearRegression()
```

In [18]:
```
#predict the model
y_pred=lr.predict(x_test)
```

In [19]:
```
#evaluate the model
from sklearn.metrics import r2_score
```

In [20]:
```
r2_score(y_test,y_pred)
```

Out[20]:
```
0.8747226291661847
```

In [21]:
```
lr.coef_
```

Out[21]:
```
array([ 0.05507865,  0.10308563, -0.00090115])
```

In [22]:
```
#loss function
from sklearn.metrics import mean_absolute_error
```

In [23]:
```
mean_absolute_error(y_test,y_pred)
```

Out[23]:
```
1.2754390912939684
```

In [24]:
```
from sklearn.metrics import mean_squared_error
```

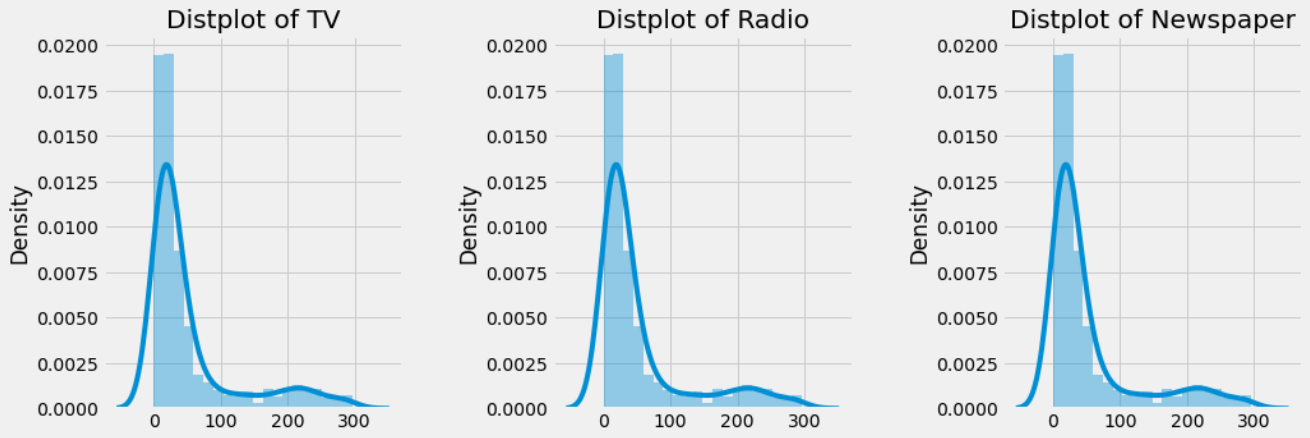
In [25]:
```
mean_squared_error(y_test,y_pred)
```

Out[25]:
```
2.409333612892368
```

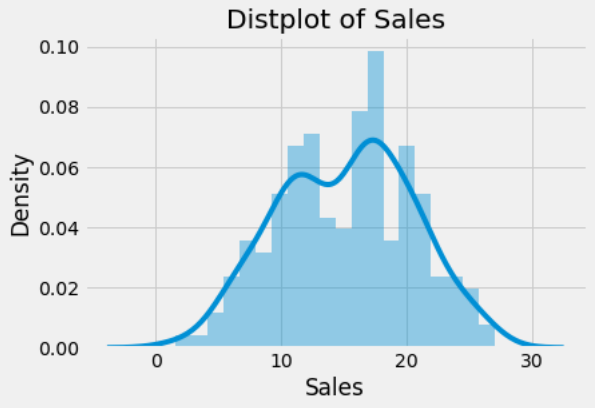
In [26]:
```
np.sqrt(mean_squared_error(y_test,y_pred))
```

Out[26]:
```
1.5522028259516758
```

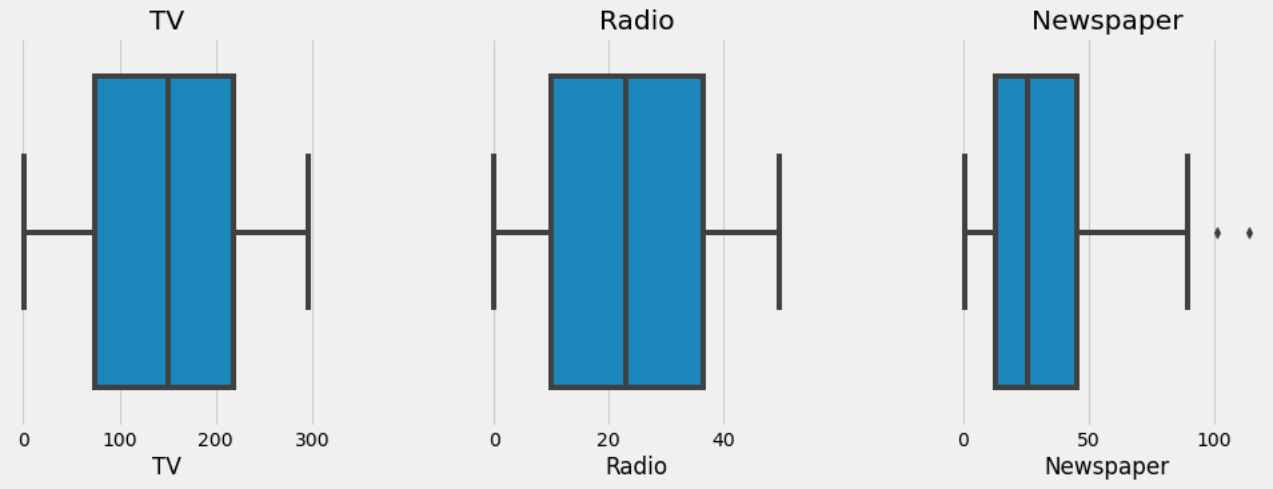
In [27]:
```
plt.style.use('fivethirtyeight')

plt.figure(1 , figsize = (15 , 5))

n = 0
for x in ['TV' , 'Radio' , 'Newspaper']:

    n += 1
    plt.subplot(1 , 3 , n)
    plt.subplots_adjust(hspace = 0.5 , wspace = 0.5)
    sns.distplot(df , bins = 20 )
    plt.title('Distplot of {}'.format(x))
```



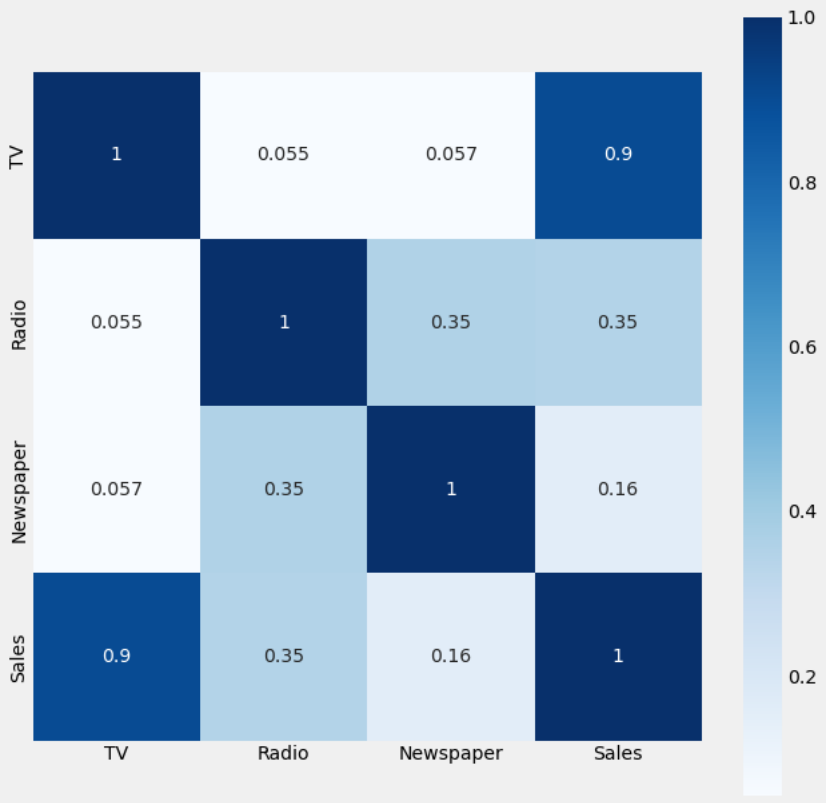
In [28]:
```
sns.distplot(df['Sales'] , bins = 20)
plt.title('Distplot of Sales')
plt.show()
```



In [29]:
```
plt.figure(1 , figsize = (15 , 5))

n = 0
for x in  ['TV' , 'Radio' , 'Newspaper']:

    n += 1
    plt.subplot(1 , 3 , n)
    plt.subplots_adjust(hspace = 0.5 , wspace = 0.5)
    sns.boxplot(x =df[x])

    plt.title('{}'.format(x))

plt.show()
```



In [30]:
```
corrmat = df.corr()
plt.figure(1 , figsize = (10 , 10 ))
sns.heatmap(corrmat,
            cmap = sns.color_palette("Blues", as_cmap=True) ,
            annot = True , square = True)
plt.show()
```



In [31]:
```
plt.figure(1 , figsize = (15 , 5))

n = 0
for x in  ['TV' , 'Radio' , 'Newspaper']:

    n += 1

    plt.subplot(1 , 3 , n)
    plt.subplots_adjust(hspace = 0.5 , wspace = 0.5)

    if x == 'TV':
        sns.regplot(x = x , y = 'Sales' , data = df, color = 'g')
    else:
        sns.regplot(x = x , y = 'Sales' , data =df, color = 'r')

    plt.title('{}'.format(x))

plt.show()
```