In [1]:
```
import pandas as pd
import pandas as pd
import numpy as np
import seaborn as sns
import plotly.express as px
import plotly.graph_objects as go
import sklearn
from sklearn.linear_model import LinearRegression
from sklearn import metrics
from sklearn.metrics import r2_score
```

In [2]:
```
data=pd.read_csv("Advertising.csv")
```

In [3]:
```
data
```

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	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 [4]:
```
data.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	9.3
3	151.5	41.3	58.5	18.5
4	180.8	10.8	58.4	12.9

In [5]:
```
data.shape
```

Out[5]:
```
(200, 4)
```

In [6]:
```
data.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 [7]:
```
data.describe()
```

Out[7]:

	TV	radio	newspaper	sales
count	200.000000	200.000000	200.000000	200.000000
mean	147.042500	23.264000	30.554000	14.022500
std	85.854236	14.846809	21.778621	5.217457
min	0.700000	0.000000	0.300000	1.600000
25%	74.375000	9.975000	12.750000	10.375000
50%	149.750000	22.900000	25.750000	12.900000
75%	218.825000	36.525000	45.100000	17.400000
max	296.400000	49.600000	114.000000	27.000000

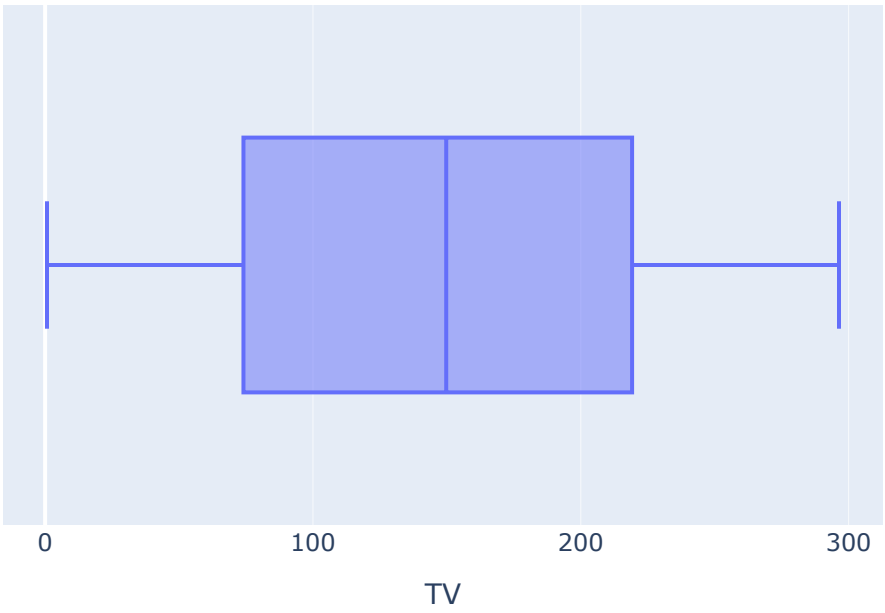
In [8]:
```
data.nunique()
```

Out[8]:
```
TV           190
radio        167
newspaper    172
sales        121
dtype: int64
```

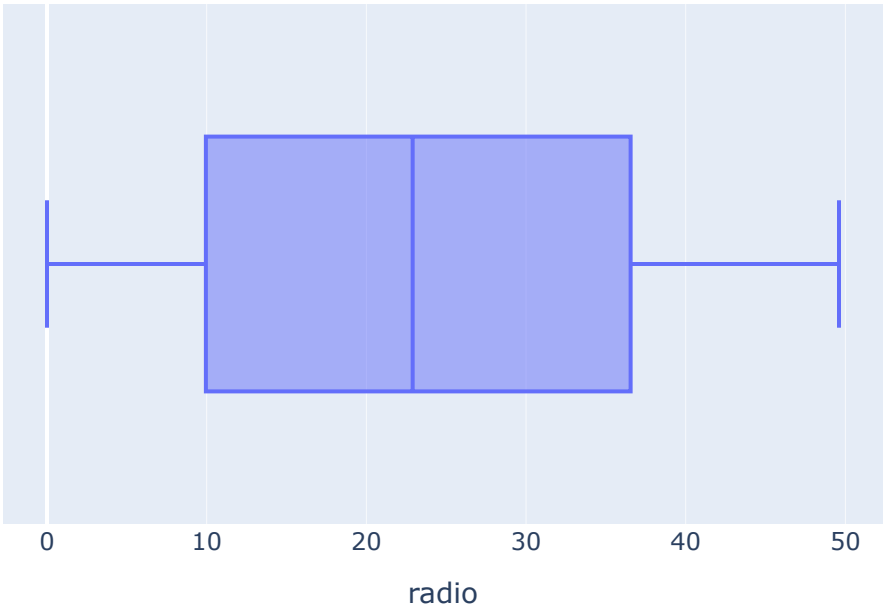
In [9]:
```
data.isnull().sum()
```

Out[9]:
```
TV           0
radio        0
newspaper    0
sales        0
dtype: int64
```

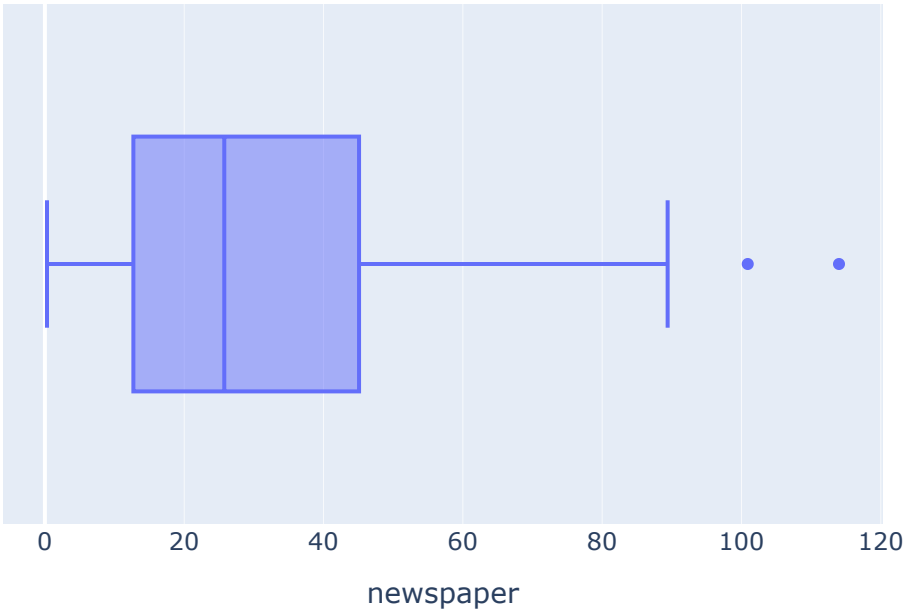
In [10]:
```
fig=px.box(data_frame=data,x='TV')
fig.update_layout(width=600,height=400)
fig.show()
```



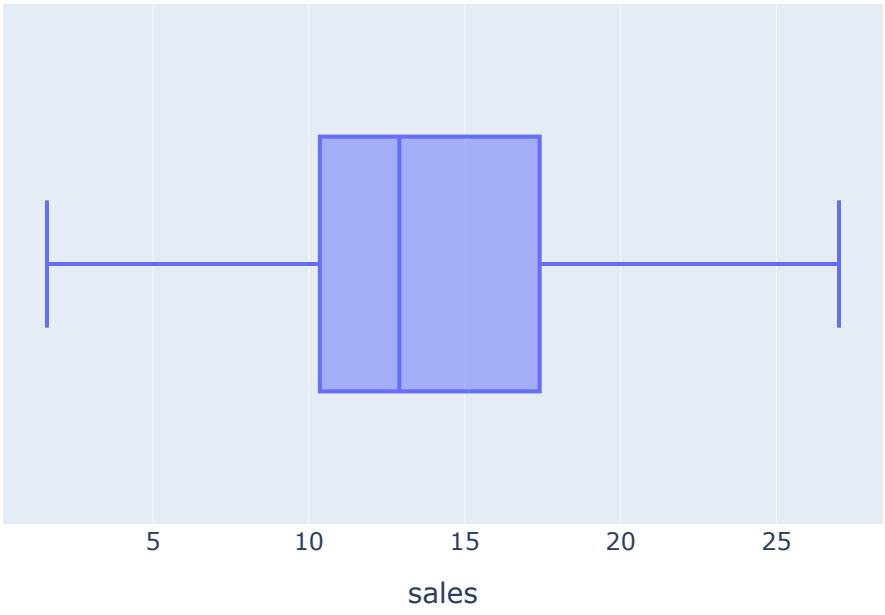
In [11]:
```
fig=px.box(data_frame=data,x='radio')
fig.update_layout(width=600,height=400)
fig.show()
```



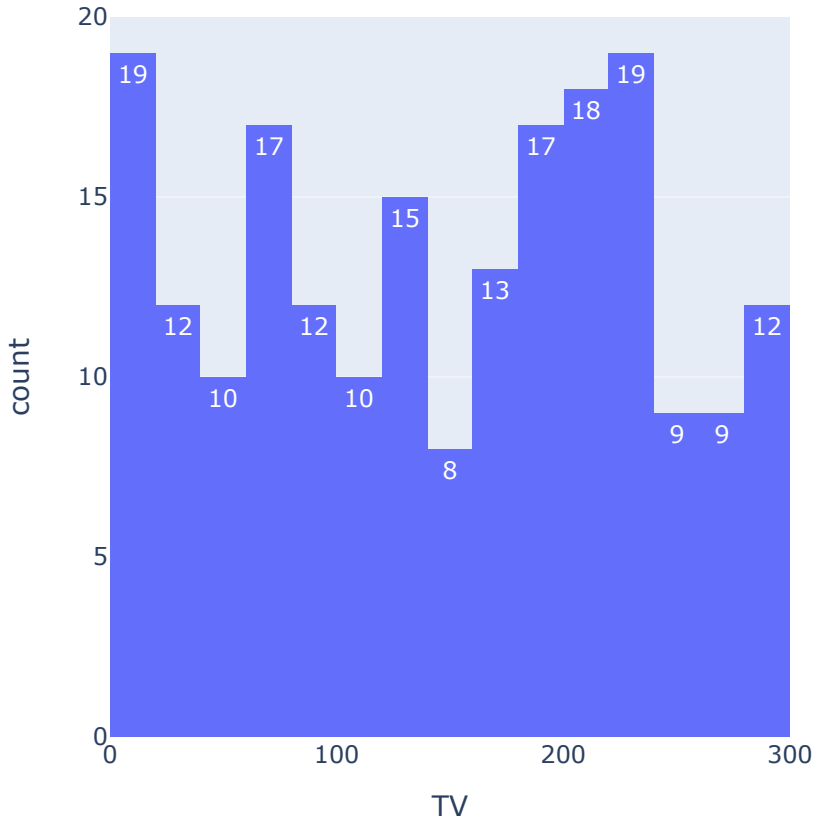
In [12]:
```
fig=px.box(data_frame=data,x='newspaper')
fig.update_layout(width=600,height=400)
fig.show()
```



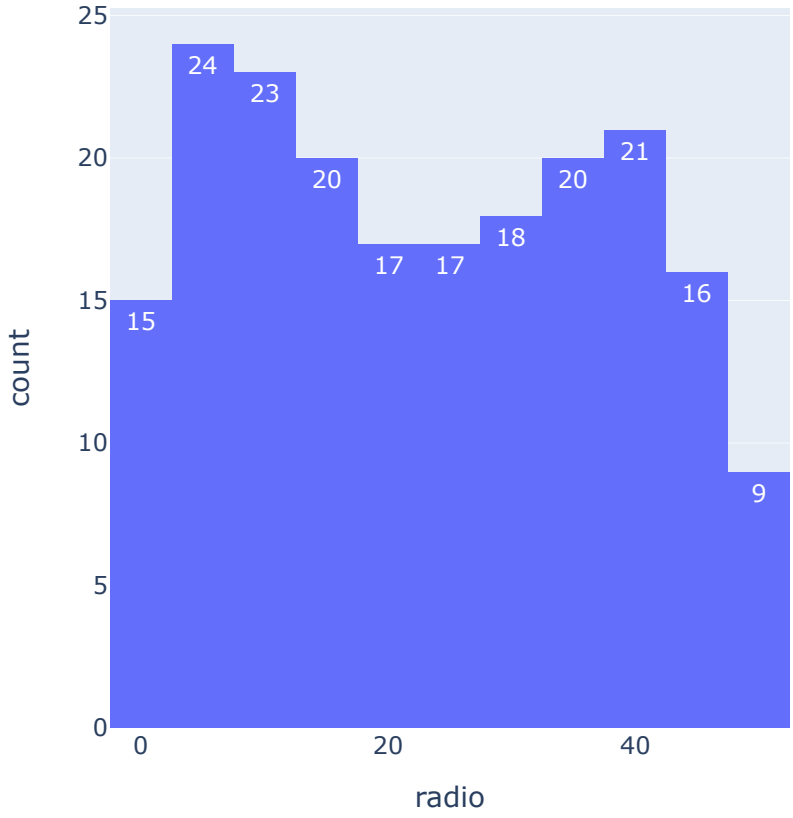
In [13]:
```
fig=px.box(data_frame=data,x='sales')
fig.update_layout(width=600,height=400)
fig.show()
```



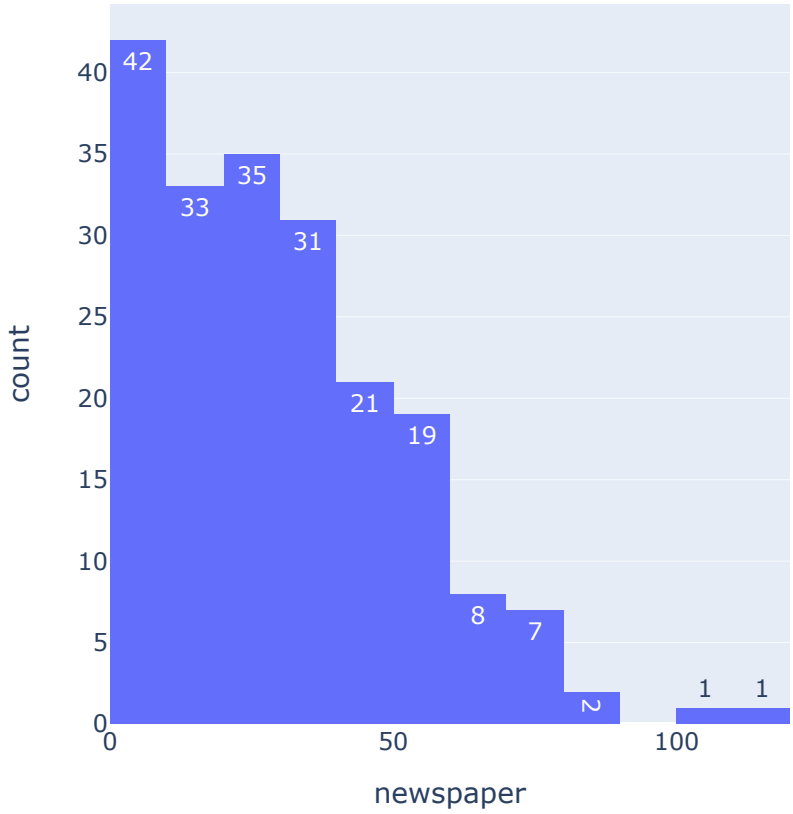
In [14]:
```
fig=px.histogram(data_frame=data,x='TV',text_auto=True,nbins=20)
fig.update_layout(width=500,height=500)
fig.show()
```



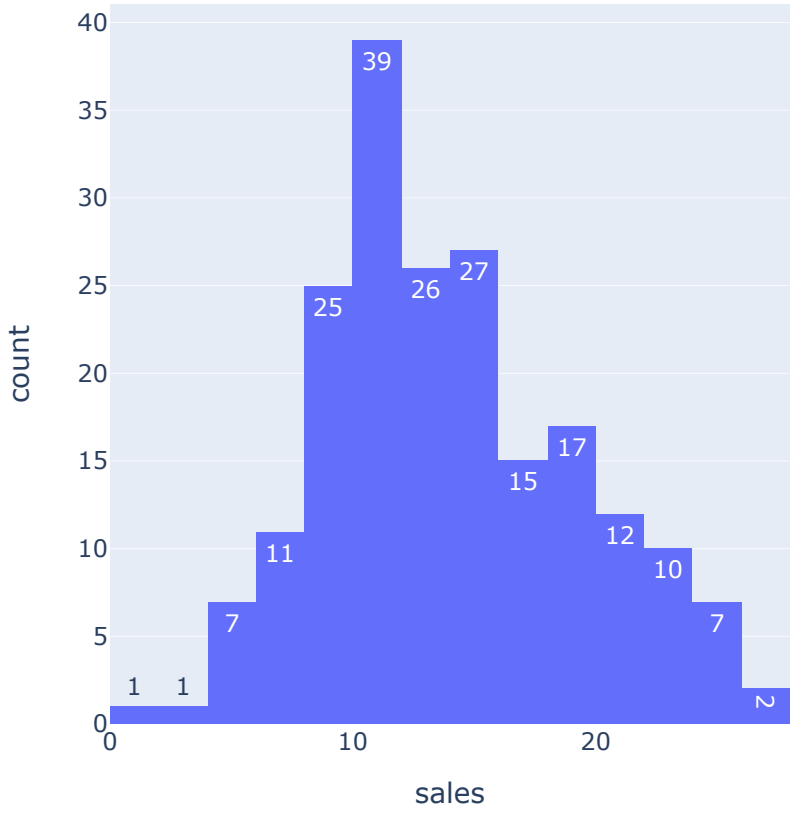
In [15]:
```
fig=px.histogram(data_frame=data,x='radio',text_auto=True,nbins=20)
fig.update_layout(width=500,height=500)
fig.show()
```



In [16]:
```
fig=px.histogram(data_frame=data,x='newspaper',text_auto=True,nbins=20)
fig.update_layout(width=500,height=500)
fig.show()
```



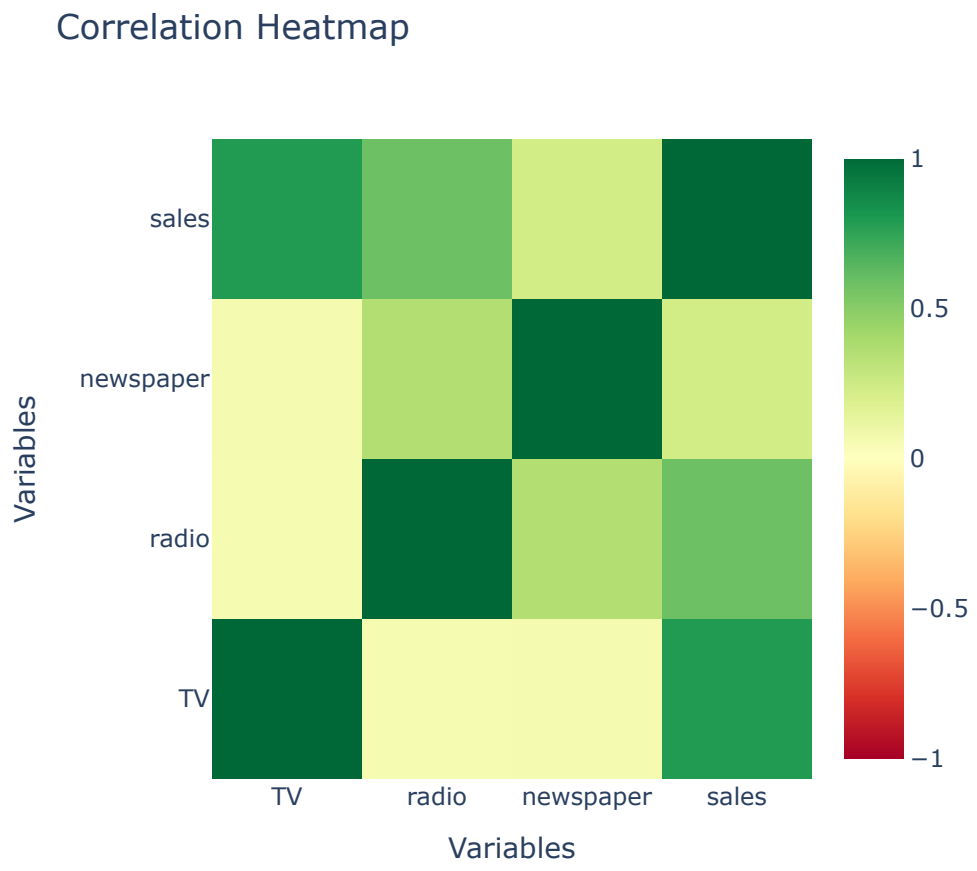
In [17]:
```
fig=px.histogram(data_frame=data,x='sales',text_auto=True,nbins=20)
fig.update_layout(width=500,height=500)
fig.show()
```



In [18]:
```
Correlation=data.corr(method='pearson')
print(Correlation)
```

```
                TV     radio  newspaper     sales
TV        1.000000  0.054809   0.056648  0.782224
radio     0.054809  1.000000   0.354104  0.576223
newspaper 0.056648  0.354104   1.000000  0.228299
sales     0.782224  0.576223   0.228299  1.000000
```

In [19]:
```
fig=go.Figure(go.Heatmap(x=Correlation.columns,y=Correlation.columns,z=Correla
fig.update_layout(title='Correlation Heatmap',xaxis_title='Variables',yaxis_ti
fig.show()
```



In [20]:
```
X = data.drop(labels='sales',axis=1)
Y = data['sales']
```

In [21]:
```
X
```

Out[21]:

	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 [22]:
```
Y
```

Out[22]:
```
0      22.1
1      10.4
2       9.3
3      18.5
4      12.9
       ...
195     7.6
196     9.7
197    12.8
198    25.5
199    13.4
Name: sales, Length: 200, dtype: float64
```

In [23]:
```
from sklearn.model_selection import train_test_split
```

In [24]:
```
x_train, x_test, y_train, y_test = train_test_split(X, Y, test_size=0.2)
```

In [25]:
```
lr = LinearRegression()
```

In [26]:
```
lr.fit(x_train, y_train)
```

Out[26]:
```
LinearRegression
LinearRegression()
```

In [27]:
```
lr.coef_
```

Out[27]:
```
array([ 0.04553342,  0.19141297, -0.00531119])
```

In [28]:
```
pred=lr.predict(x_test)
```

In [29]:
```
rmse = sklearn.metrics.mean_squared_error(y_test,pred)
np.sqrt(rmse)
```

Out[29]:
```
1.660548074403325
```

In [30]:
```
r2_score(y_test, pred)
```

Out[30]:
```
0.876357165205376
```

In [31]:
```
import statsmodels.formula.api as smf
```

In [32]:
```
lr = smf.ols('sales ~ TV + radio + newspaper',data).fit()
lr.summary()
```

Out[32]:

OLS Regression Results

Dep. Variable:	sales	**R-squared:**	0.897
Model:	OLS	**Adj. R-squared:**	0.896
Method:	Least Squares	**F-statistic:**	570.3
Date:	Sat, 11 Nov 2023	**Prob (F-statistic):**	1.58e-96
Time:	22:22:15	**Log-Likelihood:**	-386.18
No. Observations:	200	**AIC:**	780.4
Df Residuals:	196	**BIC:**	793.6
Df Model:	3		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	2.9389	0.312	9.422	0.000	2.324	3.554
TV	0.0458	0.001	32.809	0.000	0.043	0.049
radio	0.1885	0.009	21.893	0.000	0.172	0.206
newspaper	-0.0010	0.006	-0.177	0.860	-0.013	0.011

Omnibus:	60.414	**Durbin-Watson:**	2.084
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	151.241
Skew:	-1.327	**Prob(JB):**	1.44e-33
Kurtosis:	6.332	**Cond. No.**	454.

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

In [ ]:
```
```