```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
```

```
from google.colab import files
uploaded= files.upload()
```

Browse... advertising.csv

advertising.csv(application/vnd.ms-excel) - 4062 bytes, last modified: n/a - 100% done

```
Saving advertising.csv to advertising.csv
```

```
df1=pd.read_csv("advertising.csv",encoding= "latin1")
```

```
man=df1
```

```
man.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

```
man.head(20)
```

	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

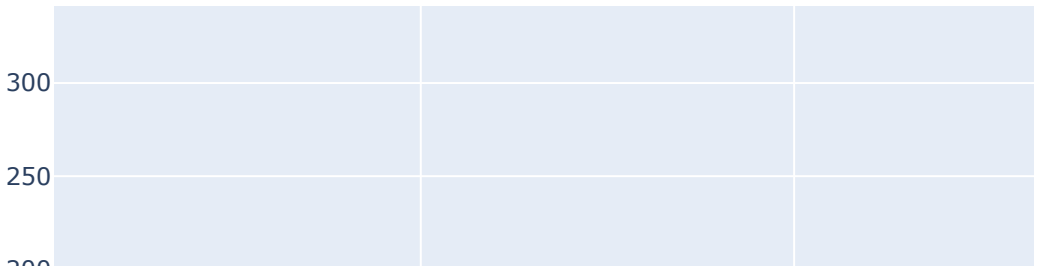
✓ 0s completed at 6:21 PM

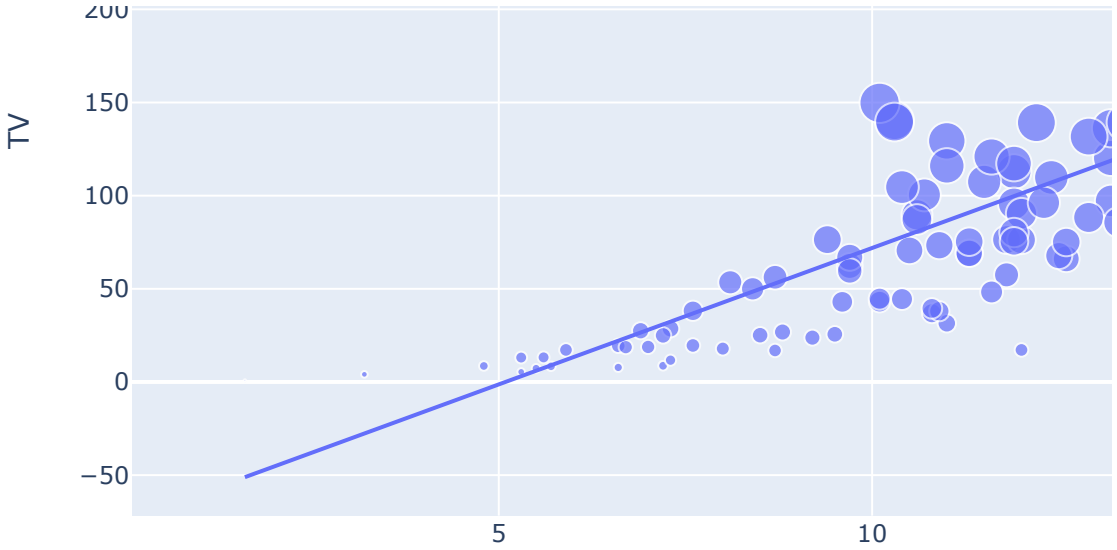
6	57.5	32.8	23.5	11.8
7	120.2	19.6	11.6	13.2
8	8.6	2.1	1.0	4.8
9	199.8	2.6	21.2	15.6
10	66.1	5.8	24.2	12.6
11	214.7	24.0	4.0	17.4
12	23.8	35.1	65.9	9.2
13	97.5	7.6	7.2	13.7
14	204.1	32.9	46.0	19.0
15	195.4	47.7	52.9	22.4
16	67.8	36.6	114.0	12.5
17	281.4	39.6	55.8	24.4

```
man.isnull().sum()
```

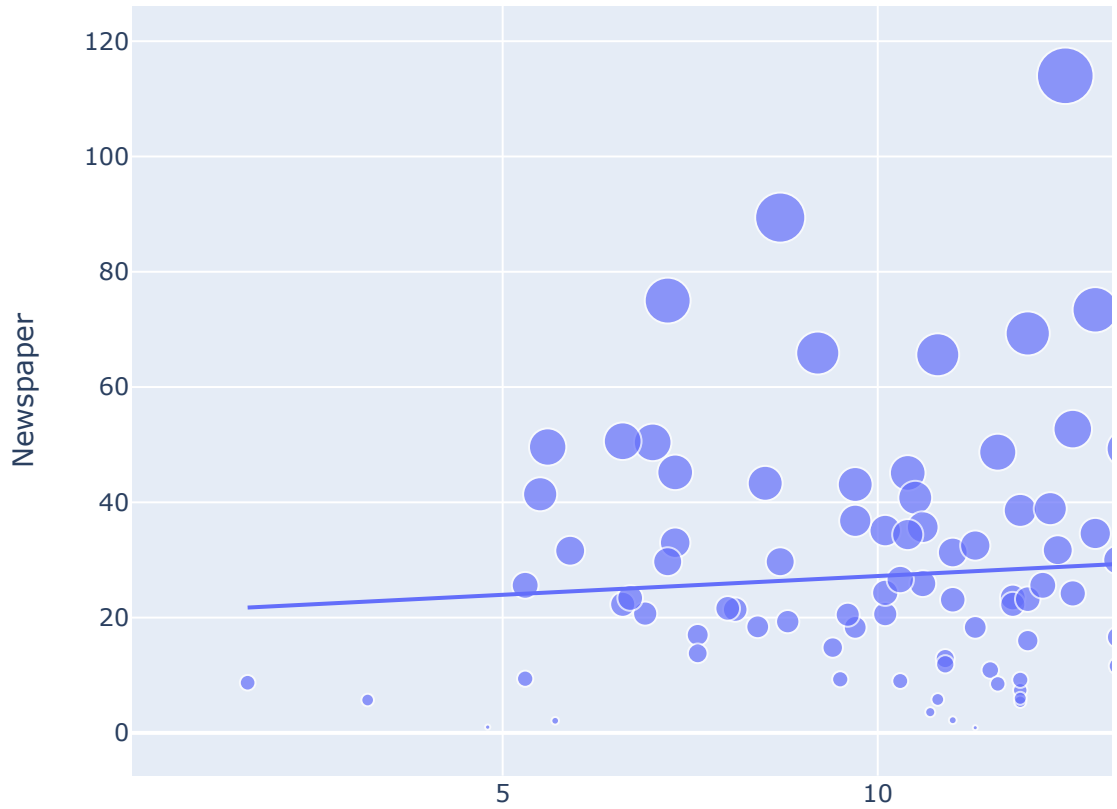
```
TV           0
Radio        0
Newspaper    0
Sales        0
dtype: int64
```

```
import plotly.express as px
import plotly.graph_objects as go
figure = px.scatter(data_frame = man, x="Sales",
                    y="TV", size="TV", trendline="ols")
figure.show()
```

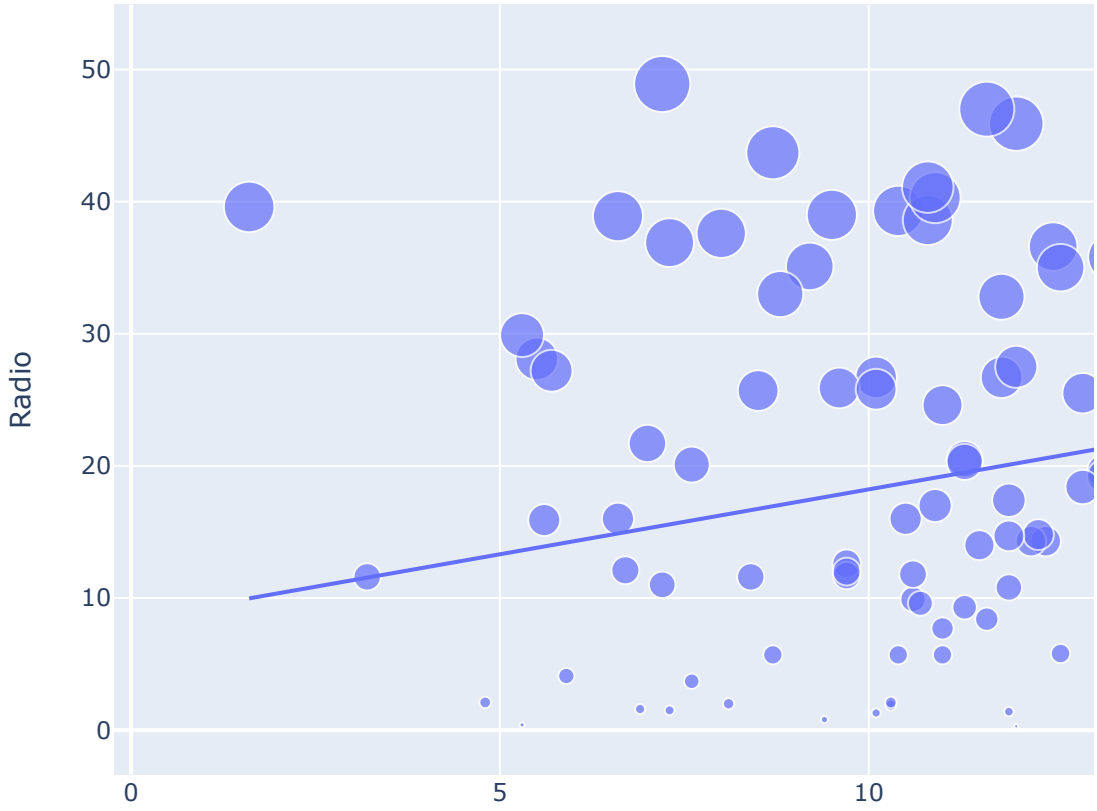





```
figure = px.scatter(data_frame = man, x="Sales",
                    y="Newspaper", size="Newspaper", trendline="ols")
figure.show()
```



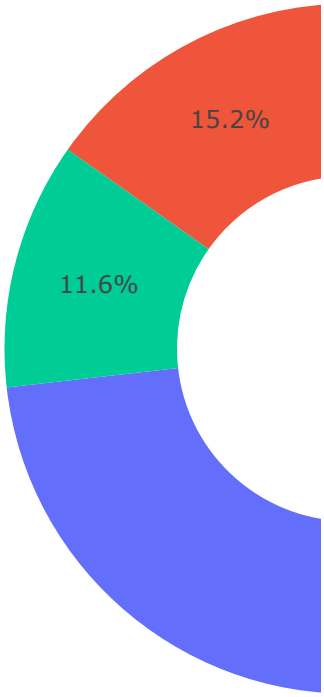
```
figure = px.scatter(data_frame = man, x="Sales",
                    y="Radio", size="Radio", trendline="ols")
figure.show()
```




```
TV = man["TV"].sum()
Radio =man["Radio"].sum()
Newspaper = man["Newspaper"].sum()


labels = ['TV','Radio','Newspaper']
values = [TV , Radio, Newspaper]

fig = px.pie(man, values=values, names=labels,
             title='sales  From Various addverising media ', hole=0.5)
fig.show()
```


```
correlation = man.corr()
print(correlation["Sales"].sort_values(ascending=False))
```

```
Sales        1.000000
TV           0.901208
Radio        0.349631
Newspaper    0.157960
Name: Sales, dtype: float64
```

```
x = np.array(man.drop(["Sales"], 1))
y = np.array(man["Sales"])
xtrain, xtest, ytrain, ytest = train_test_split(x, y,
                                                test_size=0.2,
                                                random_state=42)
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: FutureWarning:

In a future version of pandas all arguments of DataFrame.drop except for the argument
```

```
model = LinearRegression()
model.fit(xtrain, ytrain)
print(model.score(xtest, ytest))
```

```
0.9059011844150825
```

```
features = np.array([[230.1, 37.8, 69.2]]) ## here the features are taken as [TV,Radio,News
print(model.predict(features))
```

```
[21.37254028]
```