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

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	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]:
```
data.shape
```

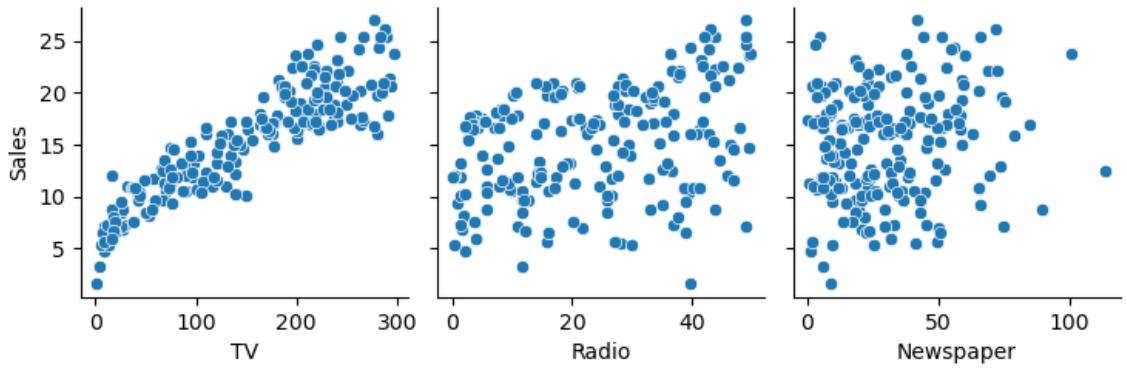
Out[4]:
```
(200, 4)
```

In [5]:
```
data.describe()
```

Out[5]:

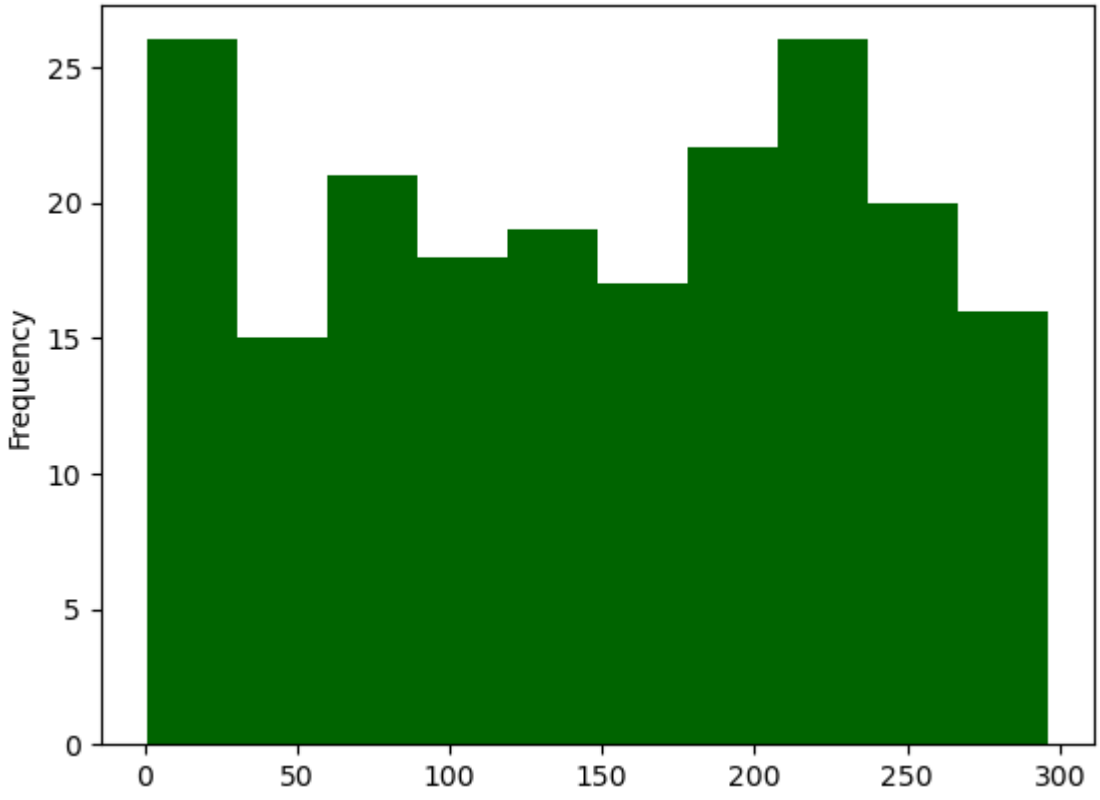
	TV	Radio	Newspaper	Sales
count	200.000000	200.000000	200.000000	200.000000
mean	147.042500	23.264000	30.554000	15.130500
std	85.854236	14.846809	21.778621	5.283892
min	0.700000	0.000000	0.300000	1.600000
25%	74.375000	9.975000	12.750000	11.000000
50%	149.750000	22.900000	25.750000	16.000000
75%	218.825000	36.525000	45.100000	19.050000
max	296.400000	49.600000	114.000000	27.000000

In [6]:
```
sns.pairplot(data,x_vars = ['TV','Radio','Newspaper'], y_vars = 'Sales', ki
plt.show()
```



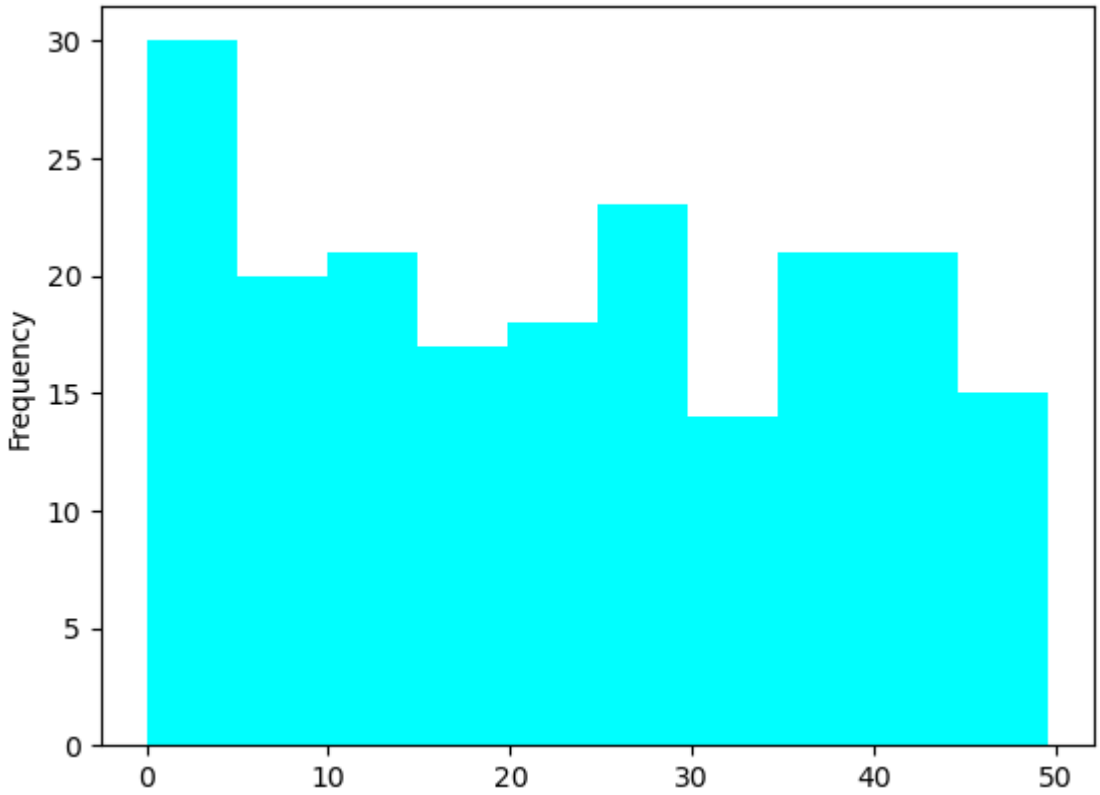
In [7]:
```
data['TV'].plot.hist(bins=10,color="darkgreen",xlabel='TV')
```

Out[7]: <Axes: ylabel='Frequency'>



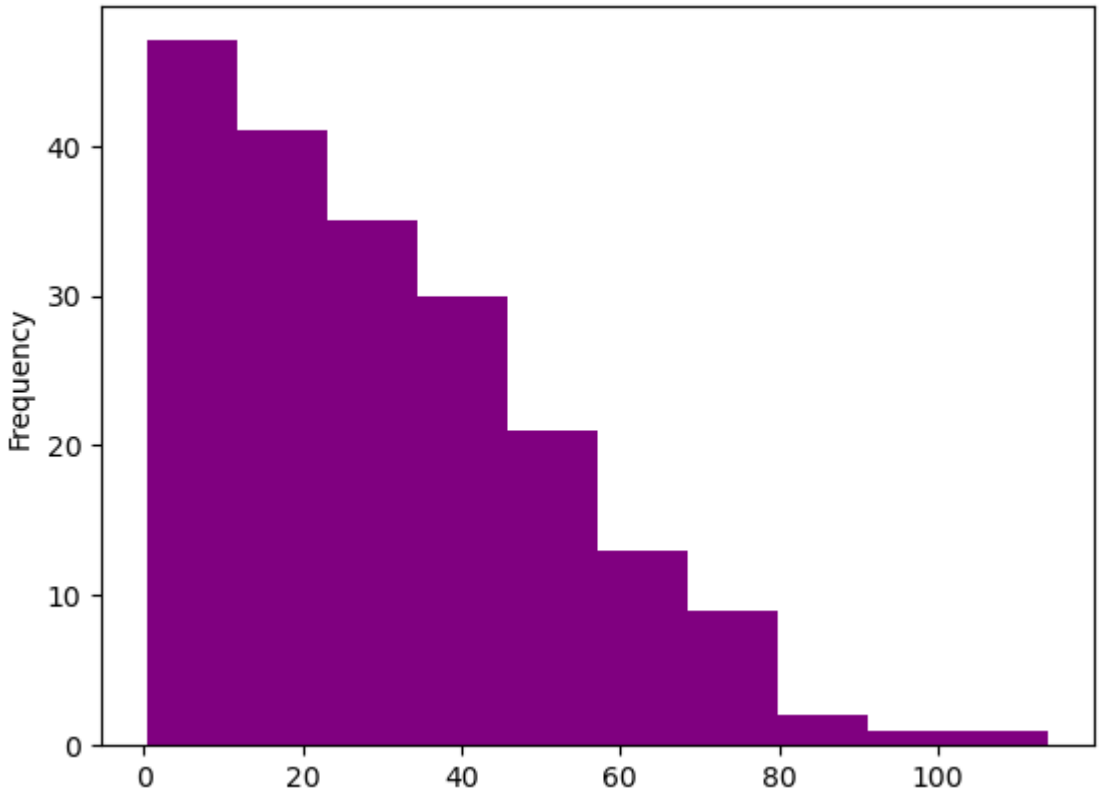
In [8]:
```
data['Radio'].plot.hist(bins=10,color="cyan",xlabel='Radio')
```

Out[8]:
```
<Axes: ylabel='Frequency'>
```

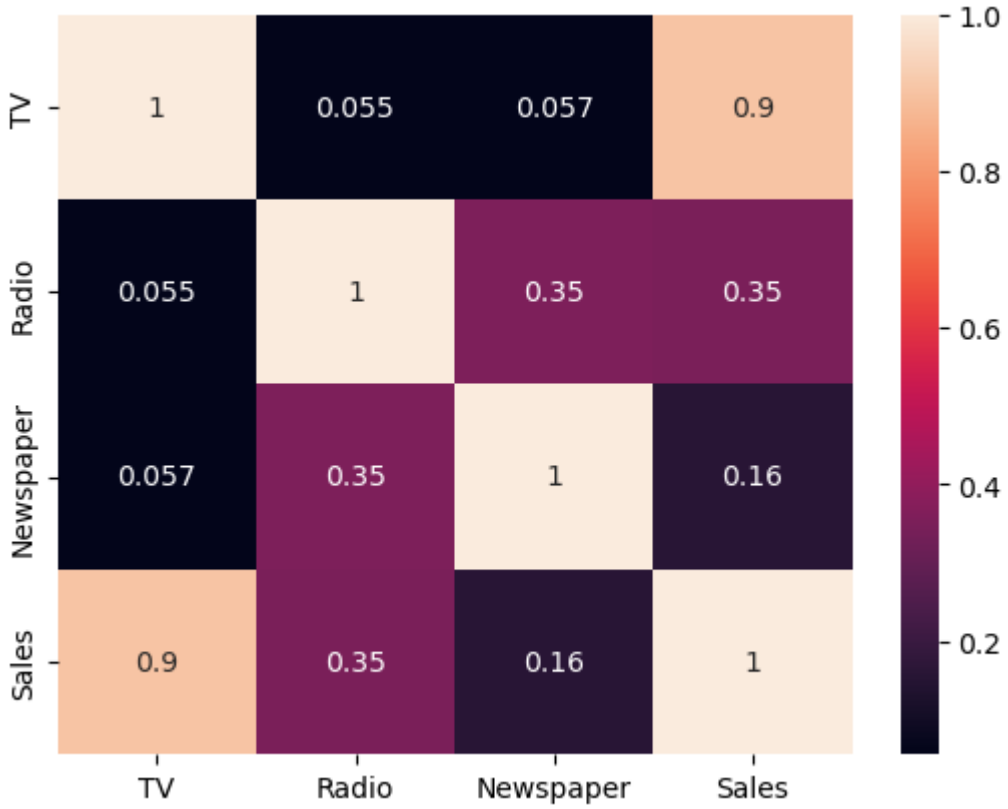


In [9]:
```
data['Newspaper'].plot.hist(bins=10,color="purple",xlabel='Newspaper')
```

Out[9]:
```
<Axes: ylabel='Frequency'>
```



In [10]:
```
sns.heatmap(data.corr(),annot=True)
plt.show()
```



In [11]:
```
from sklearn.model_selection import train_test_split
X_train,X_test,Y_train,Y_test = train_test_split(data[['TV']],data[['Sales'
```

In [12]:
```
X_train
```

Out[12]:

	TV
86	76.3
182	56.2
69	216.8
125	87.2
42	293.6
...	...
50	199.8
184	253.8
165	234.5
7	120.2
70	199.1

160 rows × 1 columns

In [13]:
```
Y_train
```

Out[13]:

	Sales
86	12.0
182	8.7
69	22.3
125	10.6
42	20.7
...	...
50	16.4
184	17.6
165	16.9
7	13.2
70	18.3

160 rows × 1 columns

In [14]:
```
X_test
```

Out[14]:

	TV
96	197.6
5	8.7
116	139.2
35	290.7
183	287.6
160	172.5
54	262.7
134	36.9
90	134.3
191	75.5
139	184.9
142	220.5
177	170.2
26	142.9
89	109.8
140	73.4
171	164.5
23	228.3
132	8.4
37	74.7
151	121.0
28	248.8
85	193.2
93	250.9
174	222.4
75	16.9
18	69.2
105	137.9
121	18.8
130	0.7
33	265.6
46	89.7
168	215.4
169	284.3
11	214.7
166	17.9
81	239.8
111	241.7
67	139.3

	TV
147	243.2

In [15]:
```
Y_test
```

Out[15]:

	Sales
96	16.7
5	7.2
116	12.2
35	17.8
183	26.2
160	16.4
54	20.2
134	10.8
90	14.0
191	11.9
139	20.7
142	20.1
177	16.7
26	15.0
89	16.7
140	10.9
171	17.5
23	20.5
132	5.7
37	14.7
151	11.6
28	18.9
85	20.2
93	22.2
174	16.5
75	8.7
18	11.3
105	15.0
121	7.0
130	1.6
33	17.4
46	10.6
168	17.1
169	20.0
11	17.4
166	8.0
81	17.3
111	21.8
67	13.4

	Sales
147	25.4

In [16]:
```
from sklearn.linear_model import LinearRegression
model = LinearRegression()
model.fit(X_train,Y_train)
```

Out[16]:
```
LinearRegression()
```
In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [17]:
```
res = model.predict(X_test)
res
```

Out[17]:
```
array([[18.02667063],
       [ 7.45221584],
       [14.75749032],
       [23.23832622],
       [23.06479096],
       [16.62159485],
       [21.670911  ],
       [ 9.03082688],
       [14.48319266],
       [11.19162071],
       [17.31573587],
       [19.30859236],
       [16.49284288],
       [14.96461305],
       [13.11170435],
       [11.07406457],
       [16.17376193],
       [19.74522945],
       [ 7.43542211],
       [11.14683742],
       [13.73867044],
       [20.89280131],
       [17.78036252],
       [21.01035745],
       [19.41495268],
       [ 7.91124458],
       [10.83895229],
       [14.68471748],
       [ 8.0176049 ],
       [ 7.00438292],
       [21.83325044],
       [11.98652414],
       [19.02309887],
       [22.88005988],
       [18.98391349],
       [ 7.9672237 ],
       [20.38898927],
       [20.49534959],
       [14.76308824],
       [20.57931827]])
```

In [18]:
```
print("Accuracy Score :: ",model.score(X_test,Y_test)*100)
```

```
Accuracy Score ::  80.25426506084597
```

In [19]:
```
model.coef_
```

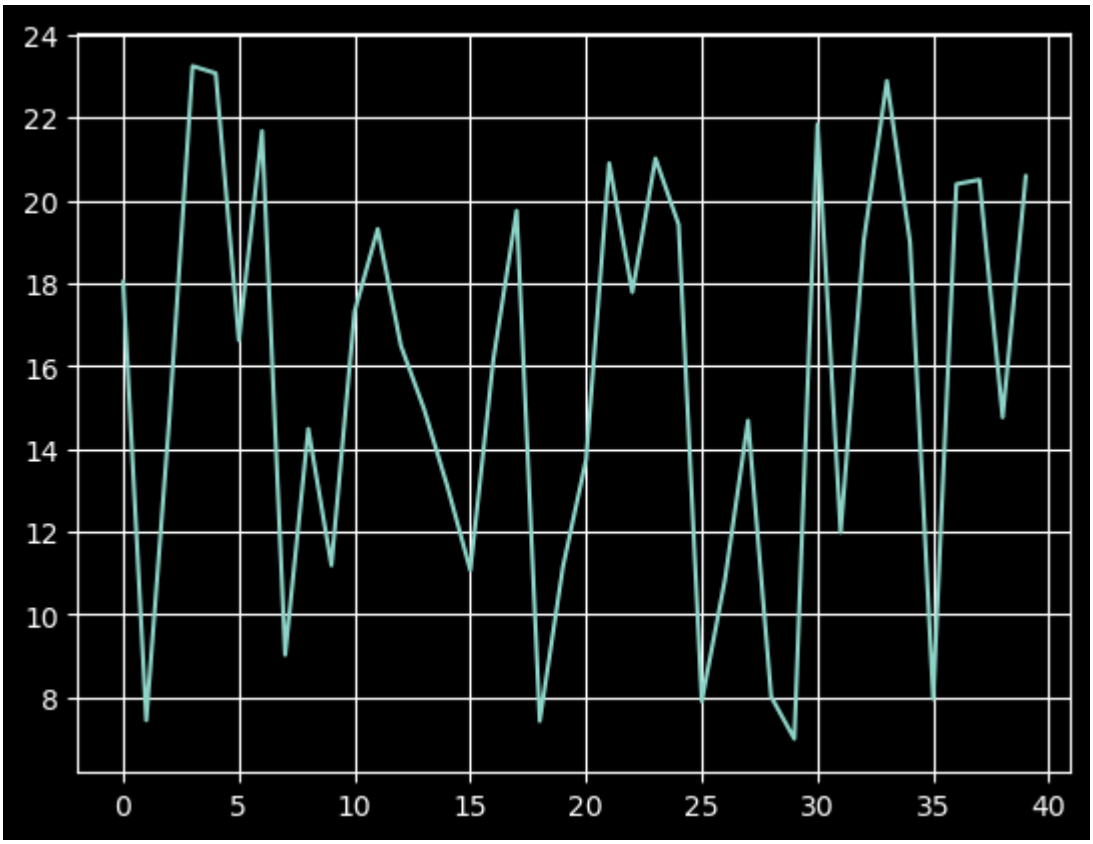
Out[19]:
```
array([[0.05597911]])
```

In [20]:
```
model.intercept_
```

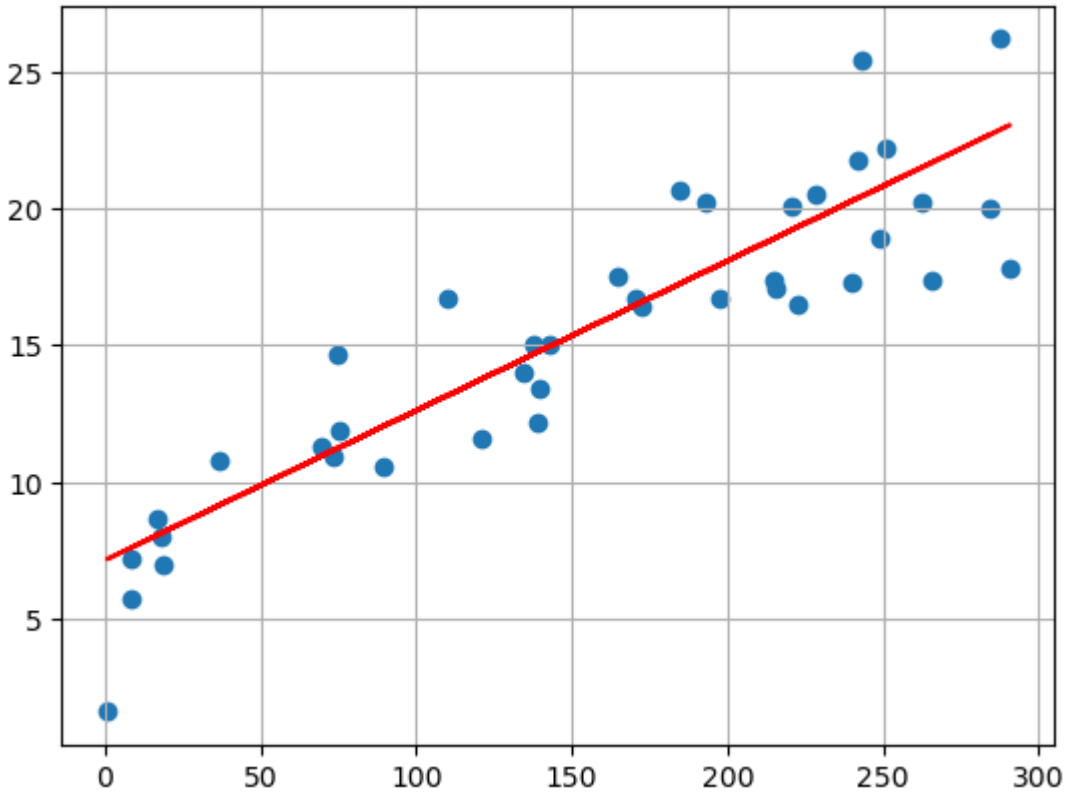
Out[20]:
```
array([6.96519754])
```

In [21]:
```
plt.style.use('dark_background')
plt.grid()
plt.plot(res)
```

Out[21]:
```
[<matplotlib.lines.Line2D at 0x1ed38d20510>]
```



In [22]:
```
plt.style.use('default')
plt.grid()
plt.scatter(X_test,Y_test)
plt.plot(X_test,7.14382225 + 0.05473199 * X_test , 'r')
plt.show()
```



In [ ]:
```

```