

```
In [69]: import numpy as np #numpy library
import pandas as pd #pandas library
import matplotlib.pyplot as plt #pyplot
import seaborn as sns #seaborn
import plotly.express as px #for visualization
```

```
In [70]: df = pd.read_csv("Advertising.csv")
df
```

Out[70]:

	Unnamed: 0	TV	Radio	Newspaper	Sales
0	1	230.1	37.8	69.2	22.1
1	2	44.5	39.3	45.1	10.4
2	3	17.2	45.9	69.3	9.3
3	4	151.5	41.3	58.5	18.5
4	5	180.8	10.8	58.4	12.9
...	...	...	...	...	...
195	196	38.2	3.7	13.8	7.6
196	197	94.2	4.9	8.1	9.7
197	198	177.0	9.3	6.4	12.8
198	199	283.6	42.0	66.2	25.5
199	200	232.1	8.6	8.7	13.4

200 rows x 5 columns

```
In [71]: df.isnull().sum()
```

Out[71]:

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

```
In [72]: df.shape # before removing null values
(200, 5)
```

```
In [73]: df.head(5)
```

Out[73]:

	Unnamed: 0	TV	Radio	Newspaper	Sales
0	1	230.1	37.8	69.2	22.1
1	2	44.5	39.3	45.1	10.4
2	3	17.2	45.9	69.3	9.3
3	4	151.5	41.3	58.5	18.5
4	5	180.8	10.8	58.4	12.9

```
In [74]: df.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 5 columns):
 #   Column      Non-Null Count  Dtype
---  --
 0   Unnamed: 0  200 non-null   int64
 1   TV          200 non-null   float64
 2   Radio       200 non-null   float64
 3   Newspaper  200 non-null   float64
 4   Sales       200 non-null   float64
dtypes: float64(4), int64(1)
memory usage: 7.9 KB
```

```
In [75]: df.describe()
Out[75]:
```

	Unnamed: 0	TV	Radio	Newspaper	Sales
count	200.000000	200.000000	200.000000	200.000000	200.000000
mean	100.500000	147.042500	23.264000	30.554000	14.022500
std	57.879185	85.854236	14.846809	21.778621	5.217457
min	1.000000	0.700000	0.000000	0.300000	1.600000
25%	50.750000	74.375000	9.975000	12.750000	10.375000
50%	100.500000	149.750000	22.900000	25.750000	12.900000
75%	150.250000	218.825000	36.525000	45.100000	17.400000
max	200.000000	296.400000	49.600000	114.000000	27.000000

```
In [76]: import seaborn as sns
import matplotlib.pyplot as plt

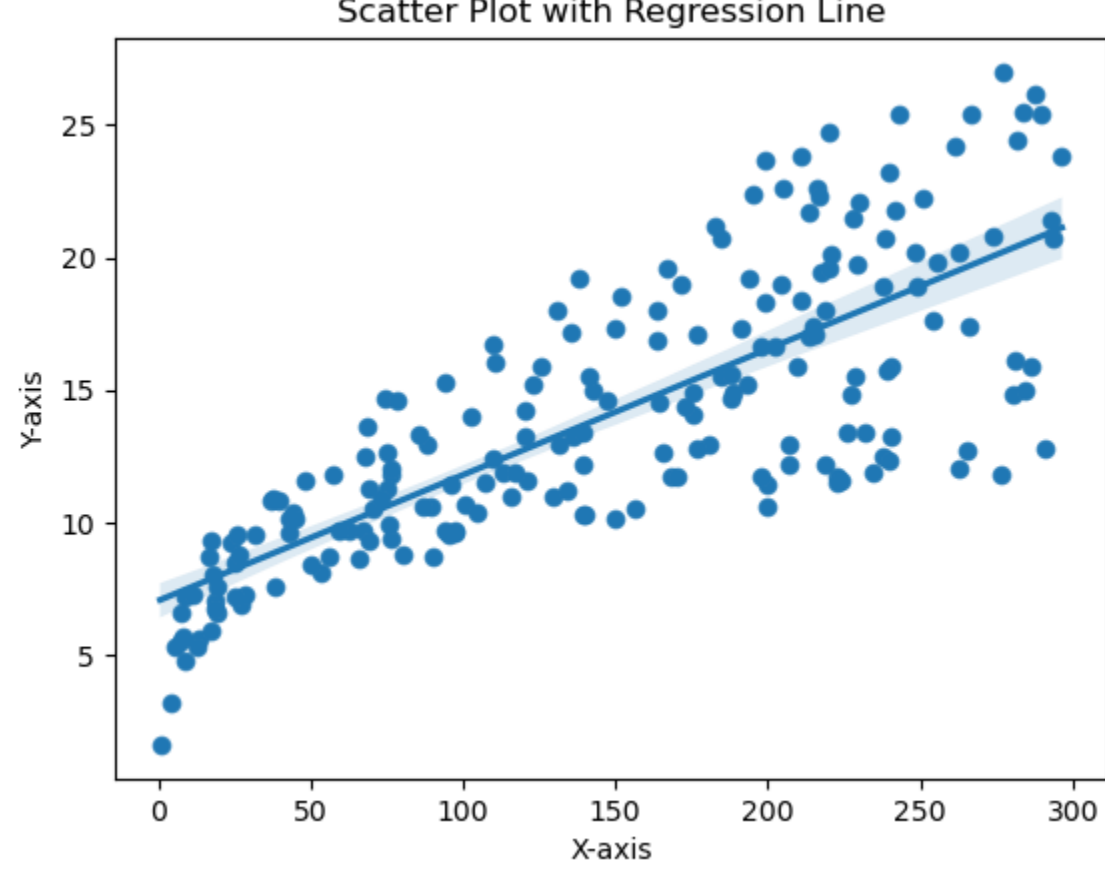
# Sample data
x = df['TV'] # X-axis values
y = df['Sales'] # Y-axis values

# Create scatter plot
sns.scatterplot(x=x, y=y)

# Fit and plot regression line
sns.regplot(x=x, y=y)

# Customize the chart
plt.title("Scatter Plot with Regression Line")
plt.xlabel("X-axis")
plt.ylabel("Y-axis")

# Display the chart
plt.show()
```



```
In [77]: import seaborn as sns
import matplotlib.pyplot as plt

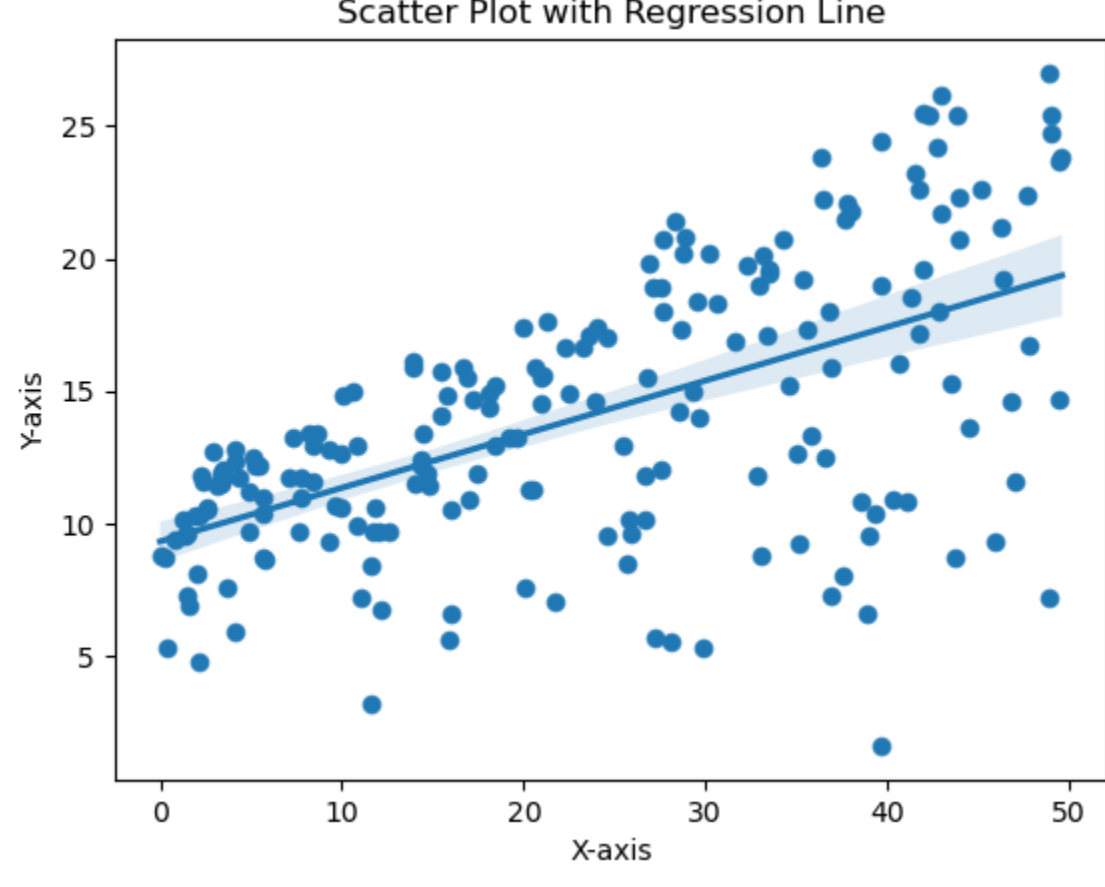
# Sample data
x = df['Radio'] # X-axis values
y = df['Sales'] # Y-axis values

# Create scatter plot
sns.scatterplot(x=x, y=y)

# Fit and plot regression line
sns.regplot(x=x, y=y)

# Customize the chart
plt.title("Scatter Plot with Regression Line")
plt.xlabel("X-axis")
plt.ylabel("Y-axis")

# Display the chart
plt.show()
```



```
In [78]: import seaborn as sns
import matplotlib.pyplot as plt

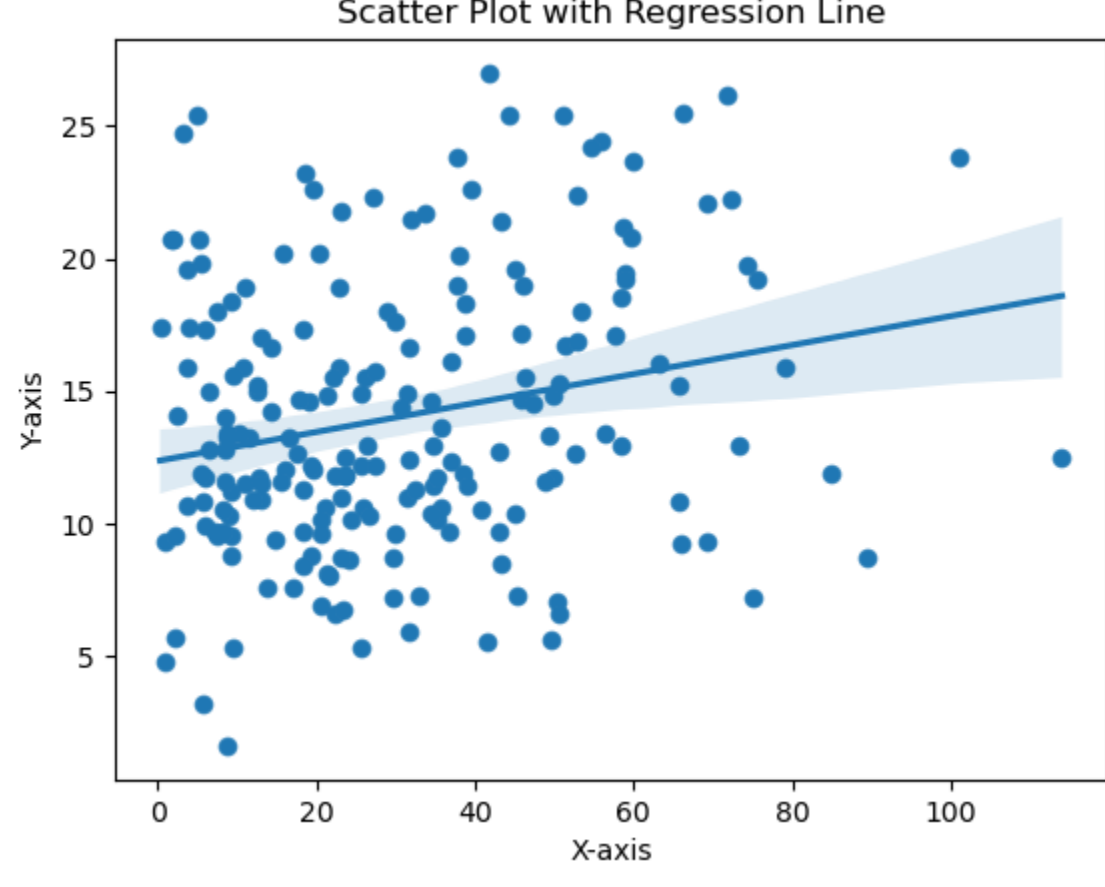
# Sample data
x = df['Newspaper'] # X-axis values
y = df['Sales'] # Y-axis values

# Create scatter plot
sns.scatterplot(x=x, y=y)

# Fit and plot regression line
sns.regplot(x=x, y=y)

# Customize the chart
plt.title("Scatter Plot with Regression Line")
plt.xlabel("X-axis")
plt.ylabel("Y-axis")

# Display the chart
plt.show()
```



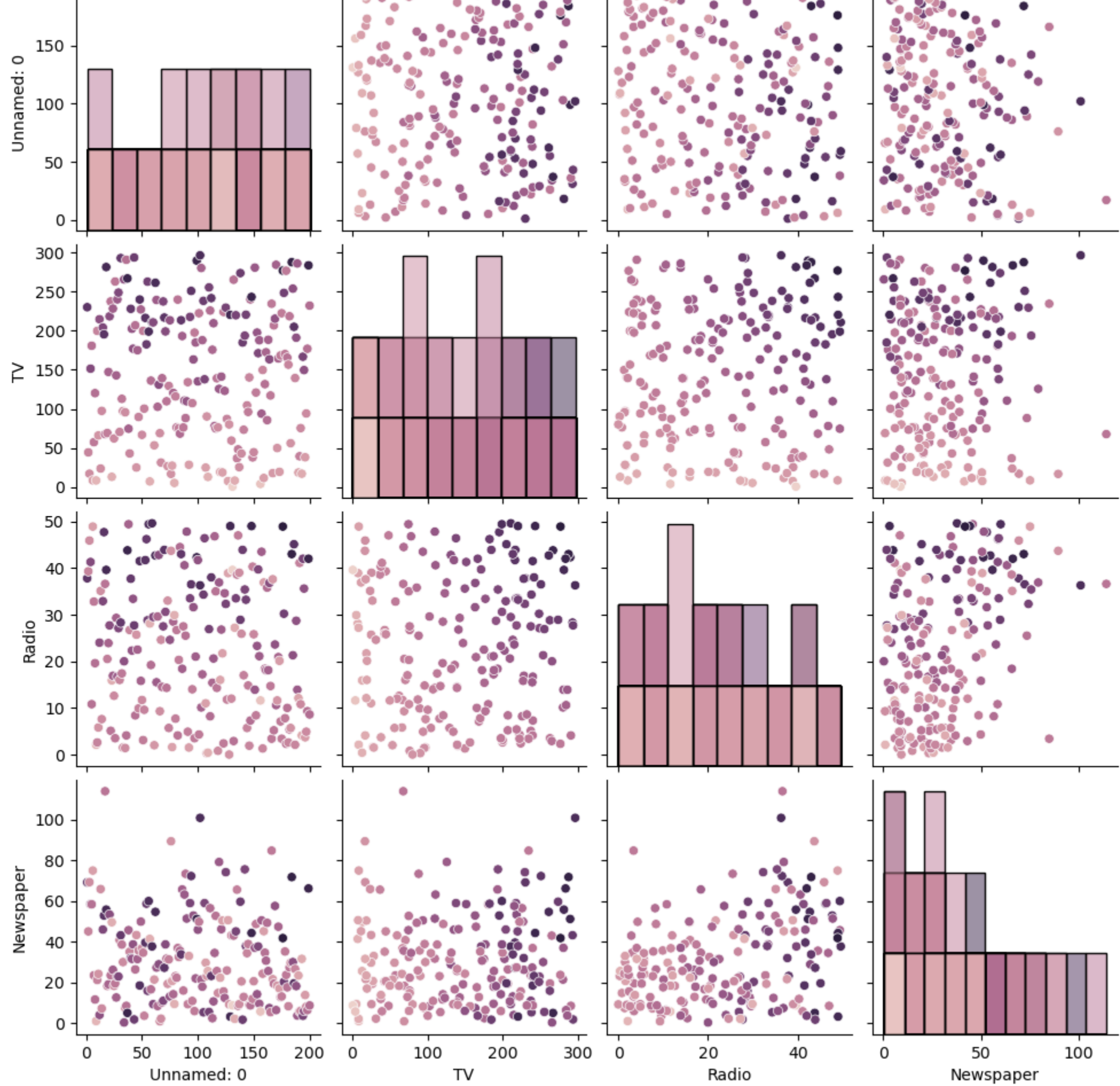
```
In [79]: correlation = df.corr()
correlation.style.background_gradient(cmap = 'BrBG')
```

Out[79]:

	Unnamed: 0	TV	Radio	Newspaper	Sales
Unnamed: 0	1.000000	0.017715	-0.110680	-0.154944	-0.051616
TV	0.017715	1.000000	0.054809	0.056648	0.782224
Radio	-0.110680	0.054809	1.000000	0.354104	0.576223
Newspaper	-0.154944	0.056648	0.354104	1.000000	0.228299
Sales	-0.051616	0.782224	0.576223	0.228299	1.000000

```
In [67]: import seaborn as sns #seaborn
sns.pairplot(df,hue="Sales",diag_kind="hist")
```

```
Out[67]: <seaborn.axisgrid.PairGrid at 0x243d48a9f0>
```



```
In [80]: x= df.drop("Sales", axis=1)
```

```
In [81]: y = df['Sales']
```

```
In [82]: from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split( x, y,test_size= 0.3,random_state= 42)
```

```
In [83]: print(x_train.shape)
print(y_train.shape)
print(x_test.shape)
print(y_test.shape)
print(df.shape)
(140, 4)
(140,)
(60, 4)
(60,)
(200, 5)
```

```
In [84]: #Import Libraries file
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split #Train Test Split
from sklearn.naive_bayes import GaussianNB # Naive Bayes Classifier
from sklearn import preprocessing # Label Encoder
from sklearn.neighbors import KNeighborsClassifier # KNN Classifiers
```

```
In [85]: x = df[['TV','Newspaper','Radio']]
y = df['Sales']
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.30,random_state=48)
x_train.shape
(140, 3)
```

```
In [86]: from sklearn.linear_model import LinearRegression
model=LinearRegression()
```

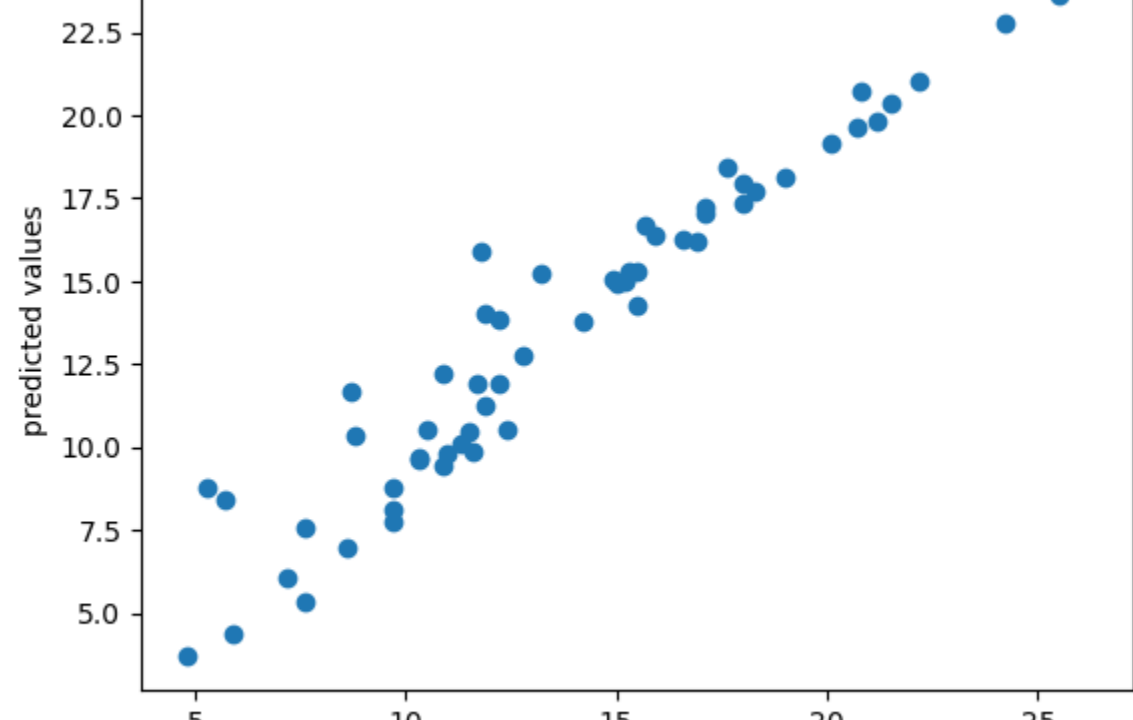
```
In [87]: model.fit(x_train,y_train)
```

```
Out[87]: LinearRegression()
LinearRegression()
```

```
In [88]: predictions=model.predict(x_test)
```

```
In [89]: plt.scatter(y_test,predictions)
plt.xlabel('y_test(true values)')
plt.ylabel('predicted values')
```

```
Out[89]: Text(0, 0.5, 'predicted values')
```



```
In [90]: model.score(x_test,y_test)
```

Out[90]: 0.9268671736178935

```
In [91]: print("ACCURACY IS:",model.score(x_test,y_test)*100)
```

ACCURACY IS: 92.68671736178935

In [ ]: