

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

In [2]: df=pd.read_csv("NewspaperData.csv")
print(df.head())

```

	Newspaper	daily	sunday
0	Baltimore Sun	391.952	488.506
1	Boston Globe	516.981	798.298
2	Boston Herald	355.628	235.084
3	Charlotte Observer	238.555	299.451
4	Chicago Sun Times	537.780	559.093

```
In [3]: df.shape
Out[3]: (34, 3)

In [4]: df.describe()

```

	daily	sunday
count	34.000000	34.000000
mean	430.962471	591.202412
std	269.211470	376.418051
min	133.239000	202.614000
25%	233.021500	327.695000
50%	355.235500	436.712500
75%	516.616500	699.735250
max	1209.225000	1762.015000

```
In [5]: df.isnull()
Out[5]: Newspaper daily sunday

```

	Newspaper	daily	sunday
0	False	False	False
1	False	False	False
2	False	False	False
3	False	False	False
4	False	False	False
5	False	False	False
6	False	False	False
7	False	False	False
8	False	False	False
9	False	False	False
10	False	False	False
11	False	False	False
12	False	False	False
13	False	False	False
14	False	False	False
15	False	False	False
16	False	False	False
17	False	False	False
18	False	False	False
19	False	False	False
20	False	False	False
21	False	False	False
22	False	False	False
23	False	False	False
24	False	False	False
25	False	False	False
26	False	False	False
27	False	False	False
28	False	False	False
29	False	False	False
30	False	False	False
31	False	False	False
32	False	False	False
33	False	False	False

```
In [10]: df.isnull().sum()
Out[10]: Newspaper    0
daily        0
sunday       0
dtype: int64

```

```
In [11]: df.head(5)
Out[11]: Newspaper daily sunday

```

	Newspaper	daily	sunday
0	Baltimore Sun	391.952	488.506
1	Boston Globe	516.981	798.298
2	Boston Herald	355.628	235.084
3	Charlotte Observer	238.555	299.451
4	Chicago Sun Times	537.780	559.093

```
In [12]: sns.lmplot(x="daily",y="sunday",data=df)
plt.show()

```

```
In [13]: sns displot(data=df, kind="kde")
Out[13]: <seaborn.axisgrid.FacetGrid at 0x2766a68b750>

```

```
In [21]: sns displot(df["daily"], kind="kde")
plt.axvline(df["daily"].mean())
Out[21]: <matplotlib.lines.Line2D at 0x27672054690>

```

```
In [18]: sns.histplot(df["daily"], kde=True)
Out[18]: <Axes: xlabel='daily', ylabel='Count'>

```

## Model Building

```
In [23]: from sklearn.linear_model import LinearRegression
In [23]: df1=df.drop(labels="Newspaper",axis=1)
print(df1.head())

```

	daily	sunday
0	391.952	488.506
1	516.981	798.298
2	355.628	235.084
3	238.555	299.451
4	537.780	559.093

```
In [26]: import statsmodels.formula.api as smf
linear_model=smf.ols(formula="sunday~daily", data=df1)

```

```
In [27]: linear_model=linear_model.fit()
print(linear_model.params)

```

	Intercept	daily
0	13.8356	1.339715
1	3.297	0.3396
2	0.20635	0.370
3	117.7	965.

```
In [28]: x_test=pd.DataFrame(data=[300,350,585])
print(x_test)

```

	daily
0	300
1	350
2	585

```
In [29]: print(linear_model.summary())

```

OLS Regression Results

Dep. Variable:	sunday	R-squared:	0.918		
Model:	OLS	Adj. R-squared:	0.915		
Method:	Least Squares	F-statistic:	358.5		
Date:	Sat, 08 Nov 2025	Prob (F-statistic):	6.02e-19		
Time:	16:52:11	Log-likelihood:	-206.85		
No. Observations:	34	AIC:	413.7		
Df Residuals:	32	BIC:	420.8		
Df Model:	1				
Covariance Type:	nonrobust				
coef	std err	t	P> t	[0.025 0.975]	
Intercept	13.8356	35.804	0.386	0.702	-59.095 86.766
daily	1.3397	0.071	18.935	0.000	1.196 1.484

Omnibus: 3.297 Durbin-Watson: 2.059  
Prob(Omnibus): 0.194 Prob(J-B): 1.09  
Skew: 0.394 Prob(J-B): 0.370  
Kurtosis: 3.882 Cond. No. 965.

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

```
In [41]: print(linear_model.predict(x_test))

```

	0	1	2
0	415.730057		
1	424.735795		
2	797.568763		

```
dtype: float64

```

## USING SCKIT LEARN

```
In [42]: from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split

```

```
In [53]: X = df[['daily']]
Y = df['sunday']

```

```
In [54]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

```

```
In [55]: X_test = pd.DataFrame(data=[300, 350, 585])
print("\nInput Daily values:")
print(X_test)

```

Input Daily values:

0	300
1	350
2	585

```
In [57]: model = LinearRegression()
model.fit(X_train, y_train)

```

LinearRegression

Parameters

```
In [50]: Y_pred = model.predict(X_test)
print("Unpredicted Sunday values:")
print(y_pred)

```

Predicted Sunday values:  
[413.83820493 482.83380856 807.11314562]

In [ ]: