Simple Linear Regression

In [1]:

```
import pandas as pd
```

In [2]:

```
data1=pd.read_csv("C:/Users/Ashraf/Documents/Datafiles/NewspaperData.csv")
data1.head()
```

Out[2]:

	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]:

```
data1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 34 entries, 0 to 33
Data columns (total 3 columns):
 #   Column     Non-Null Count  Dtype  
---  ------     --------------  -----  
 0   Newspaper  34 non-null     object 
 1   daily      34 non-null     float64
 2   sunday     34 non-null     float64
dtypes: float64(2), object(1)
memory usage: 944.0+ bytes
```

Correlatrion

In [4]:

```
data1.corr()
```

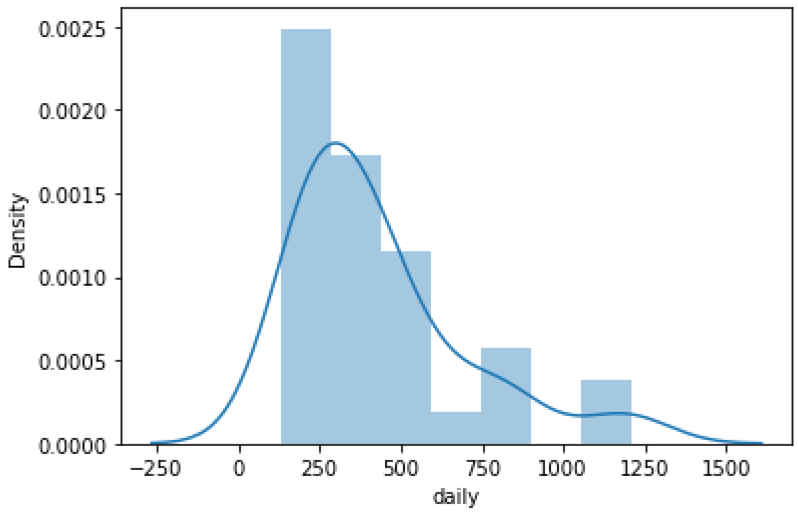
Out[4]:

	daily	sunday
daily	1.000000	0.958154
sunday	0.958154	1.000000

In [5]:

```
import seaborn as sns
sns.distplot(data1['daily']);
```

```
C:\Users\Ashraf\anaconda3\lib\site-packages\seaborn\distributions.py:2619: F
utureWarning: `distplot` is a deprecated function and will be removed in a f
uture version. Please adapt your code to use either `displot` (a figure-leve
l function with similar flexibility) or `histplot` (an axes-level function f
or histograms).
  warnings.warn(msg, FutureWarning)
```



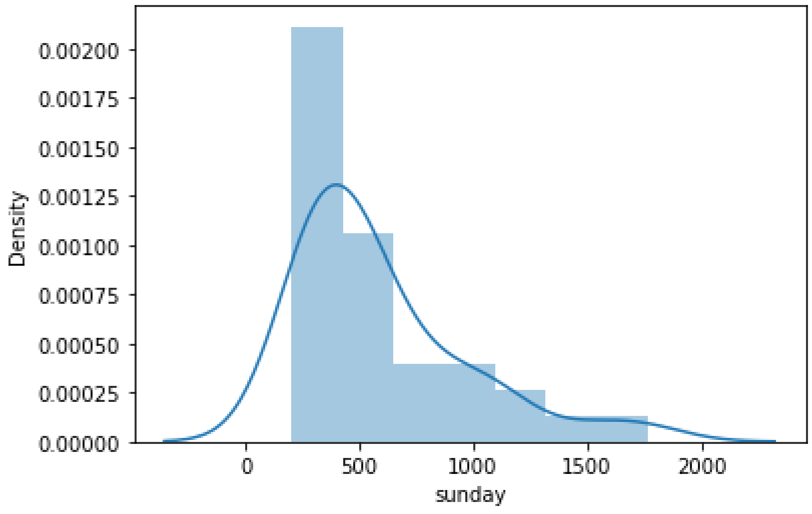
In [6]:

```
sns.distplot(data1['sunday'])
```

```
C:\Users\Ashraf\anaconda3\lib\site-packages\seaborn\distributions.py:2619: F
utureWarning: `distplot` is a deprecated function and will be removed in a f
uture version. Please adapt your code to use either `displot` (a figure-leve
l function with similar flexibility) or `histplot` (an axes-level function f
or histograms).
  warnings.warn(msg, FutureWarning)
```

Out[6]:

```
<AxesSubplot:xlabel='sunday', ylabel='Density'>
```



Fitting a linear regression model

In [7]:

```
import statsmodels.formula.api as smf
```

In [8]:

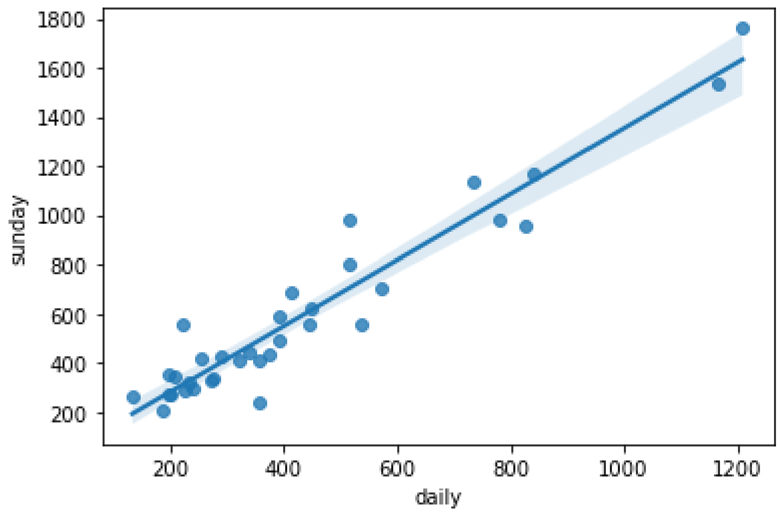
```
model=smf.ols("sunday~daily",data=data1).fit()
```

In [9]:

```
sns.regplot(x="daily",y="sunday",data=data1)
```

Out[9]:

```
<AxesSubplot:xlabel='daily', ylabel='sunday'>
```



In [10]:

```
# Coefficient
model.params
```

Out[10]:

```
Intercept    13.835630
daily         1.339715
dtype: float64
```

In [11]:

```
# t-value and p-value
print(model.tvalues,'\n', model.pvalues)
```

```
Intercept     0.386427
daily        18.934840
dtype: float64
 Intercept    7.017382e-01
daily        6.016802e-19
dtype: float64
```

In [12]:

```
# Rsquared values
(model.rsquared,model.rsquared_adj)
```

Out[12]:

```
(0.9180596895873294, 0.9154990548869335)
```

Predict for new data point

In [13]:

```
# predict for 200 ,250 ,300 daily ciculation
newdata=pd.Series([200,250,300])
```

In [14]:

```
data_pred=pd.DataFrame(newdata, columns=['daily'])
```

In [15]:

```
data_pred
```

Out[15]:

	daily
0	200
1	250
2	300

In [16]:

```
model.predict(data_pred)
```

Out[16]:

```
0    281.778581
1    348.764319
2    415.750057
dtype: float64
```

In [ ]:

```

```