Assignment-04-Simple Linear Regression- 1

In [1]:
```
# import libraries
import pandas as pd
import numpy as np
import seaborn as sns
import statsmodels.formula.api as smf
```

In [3]:
```
# import dataset
dataset=pd.read_csv('Downloads\\delivery_time.csv')
dataset
```

Out[3]:

	Delivery Time	Sorting Time
0	21.00	10
1	13.50	4
2	19.75	6
3	24.00	9
4	29.00	10
5	15.35	6
6	19.00	7
7	9.50	3
8	17.90	10
9	18.75	9
10	19.83	8
11	10.75	4
12	16.68	7
13	11.50	3
14	12.03	3
15	14.88	4
16	13.75	6
17	18.11	7
18	8.00	2
19	17.83	7
20	21.50	5

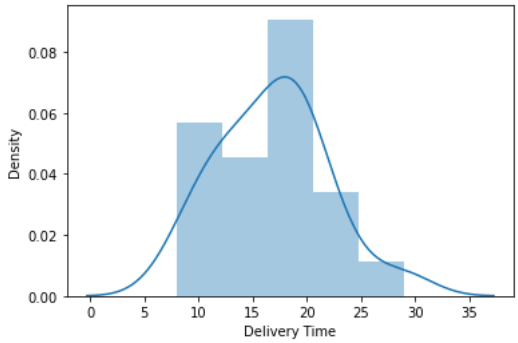
EDA and Data Visualization

In [4]:
```
dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 21 entries, 0 to 20
Data columns (total 2 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Delivery Time  21 non-null     float64
 1   Sorting Time   21 non-null     int64
dtypes: float64(1), int64(1)
memory usage: 464.0 bytes
```

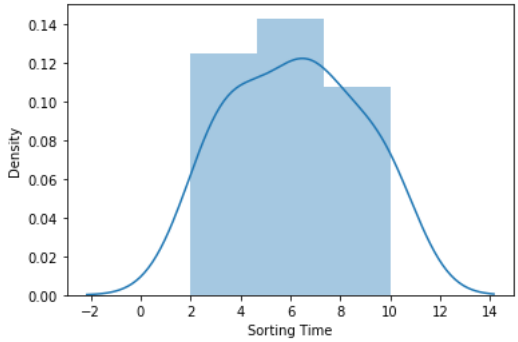
In [6]:
```
1 import warnings
2 warnings.filterwarnings('ignore')
3 sns.distplot(dataset['Delivery Time'])
```

Out[6]: <AxesSubplot:xlabel='Delivery Time', ylabel='Density'>



In [7]:
```
1 import warnings
2 warnings.filterwarnings('ignore')
3 sns.distplot(dataset['Sorting Time'])
```

Out[7]: <AxesSubplot:xlabel='Sorting Time', ylabel='Density'>



Feature Engineering

In [9]:
```
# Rename columns
dataset=dataset.rename({'Delivery Time':'delivery_time','Sorting Time':'sorting_time'},axis=1)
dataset
```

Out[9]:

	delivery_time	sorting_time
0	21.00	10
1	13.50	4
2	19.75	6
3	24.00	9
4	29.00	10
5	15.35	6
6	19.00	7
7	9.50	3
8	17.90	10
9	18.75	9
10	19.83	8
11	10.75	4
12	16.68	7
13	11.50	3
14	12.03	3
15	14.88	4
16	13.75	6
17	18.11	7
18	8.00	2
19	17.83	7
20	21.50	5

Correlation Analysis

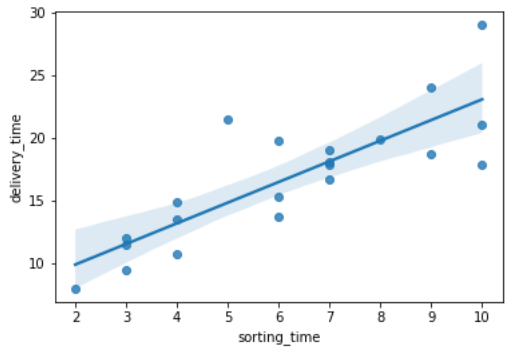
In [10]:
```
dataset.corr()
```

Out[10]:

	delivery_time	sorting_time
delivery_time	1.000000	0.825997
sorting_time	0.825997	1.000000

In [11]:
```
sns.regplot(x=dataset['sorting_time'],y=dataset['delivery_time'])
```

Out[11]: <AxesSubplot:xlabel='sorting_time', ylabel='delivery_time'>



Model Building

In [12]:
```
model=smf.ols("delivery_time~sorting_time",data=dataset).fit()
```

Model Testing

In [13]:
```
# Finding Coefficient Parameters
model.params
```

Out[13]:
```
Intercept       6.582734
sorting_time    1.649020
dtype: float64
```

In [14]:
```
# Finding tvalues and pvalues
model.tvalues , model.pvalues
```

Out[14]:
```
(Intercept       3.823349
 sorting_time    6.387447
 dtype: float64,
 Intercept       0.001147
 sorting_time    0.000004
 dtype: float64)
```

In [15]:
```
# Finding Rsquared Values
model.rsquared , model.rsquared_adj
```

Out[15]:
```
(0.6822714748417231, 0.6655489208860244)
```

Model Predictions

In [17]:
```
# Manual Predictions for say sorting time 5
delivery_time = (6.582734) + (1.649020)*5
delivery_time
```

Out[17]:
```
14.827834
```

In [18]:
```
# Automatic Predictions for say sorting time 5, 8
new_data=pd.Series([5,8])
new_data
```

Out[18]:
```
0    5
1    8
dtype: int64
```

In [19]:
```
data_pred=pd.DataFrame(new_data,columns=['sorting_time'])
data_pred
```

Out[19]:

	sorting_time
0	5
1	8

In [20]:
```
model.predict(data_pred)
```

Out[20]:
```
0    14.827833
1    19.774893
dtype: float64
```

In [ ]:
```

```