In [1]:

```
import pandas as pd
import numpy as np
import seaborn as sns
import statsmodels.formula.api as smf
```

In [2]:

```
# import dataset
dataset=pd.read_csv('delivery_time.csv')
dataset
```

Out[2]:

| | 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 |

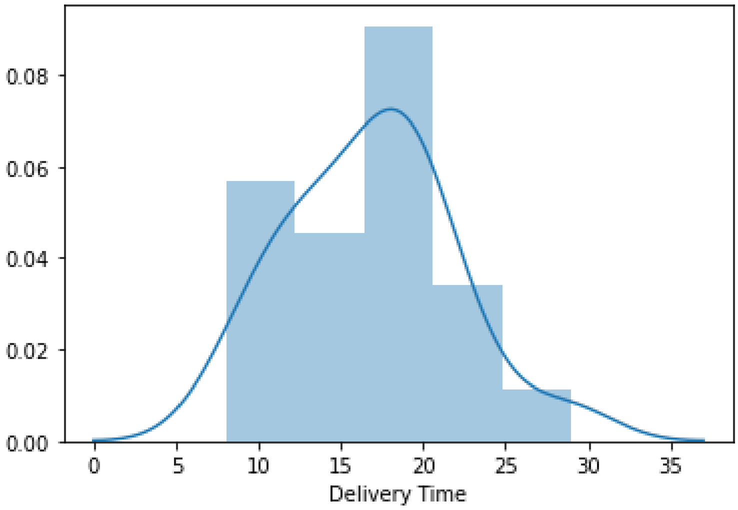
In [3]:

```
dataset.info()
sns.distplot(dataset['Delivery Time'])
```

```
<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
```

Out[3]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x24da73e1f70>
```

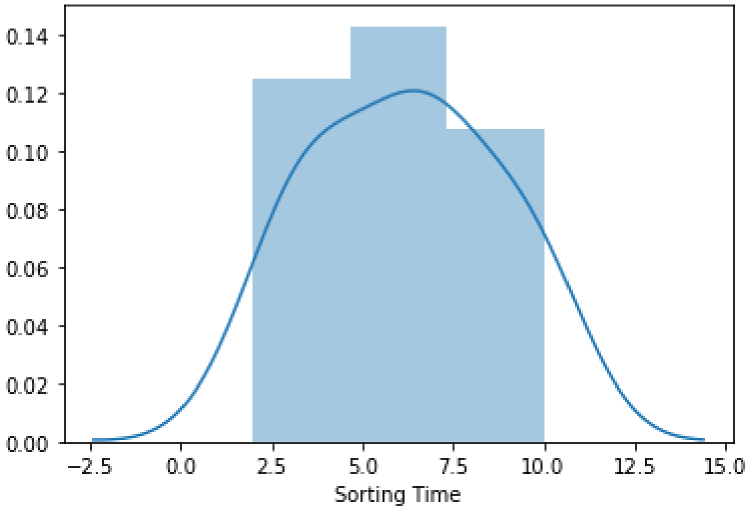


In [4]:

```
sns.distplot(dataset['Sorting Time'])
```

Out[4]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x24da7b76190>
```



In [5]:

```
# Renaming Columns
dataset=dataset.rename({'Delivery Time':'delivery_time', 'Sorting Time':'sorting_time'
},axis=1)
dataset
```

Out[5]:

| | 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 |

In [6]:

```
# correlation analysis
dataset.corr()
```

Out[6]:

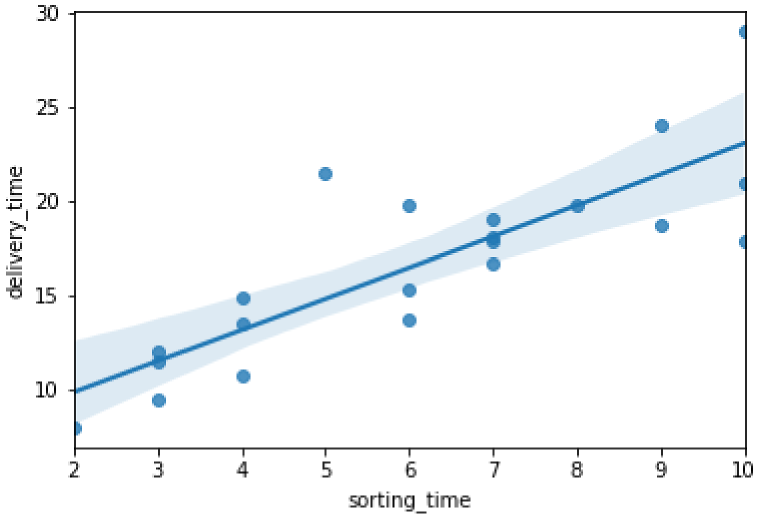
| | delivery_time | sorting_time |
|---|---|---|
| **delivery_time** | 1.000000 | 0.825997 |
| **sorting_time** | 0.825997 | 1.000000 |

In [7]:

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

Out[7]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x24da7be4460>
```



In [8]:

```
# model building
model=smf.ols("delivery_time~sorting_time",data=dataset).fit()
```

In [9]:

```
# model testing
# Finding Coefficient parameters
model.params
```

Out[9]:

```
Intercept       6.582734
sorting_time    1.649020
dtype: float64
```

In [10]:

```
# Finding tvalues and pvalues
model.tvalues , model.pvalues
```

Out[10]:

```
(Intercept       3.823349
 sorting_time    6.387447
 dtype: float64,
 Intercept       0.001147
 sorting_time    0.000004
 dtype: float64)
```

In [11]:

```
# Finding Rsquared Values
model.rsquared , model.rsquared_adj
```

Out[11]:

```
(0.6822714748417231, 0.6655489208860244)
```

In [12]:

```
# model prediction
# Manual prediction for say sorting time 5
delivery_time = (6.582734) + (1.649020)*(5)
delivery_time
```

Out[12]:

```
14.827834
```

In [13]:

```
# Automatic Prediction for say sorting time 5, 8
new_data=pd.Series([5,8])
new_data
```

Out[13]:

```
0    5
1    8
dtype: int64
```

In [14]:

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

Out[14]:

| | sorting_time |
|---|---|
| **0** | 5 |
| **1** | 8 |

In [15]:

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

Out[15]:

```
0    14.827833
1    19.774893
dtype: float64
```