1. Import Necessary Library

In [14]:

```
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
import seaborn as sns
from scipy import stats
import statsmodels.formula.api as smf

import warnings
warnings.filterwarnings('ignore')
```

2. Import Data

In [3]:

```
data = pd.read_csv('delivery_time.csv')
data
```

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

3. Data Understanding

3.1 Initial Analysis

In [4]:

```
data.shape
```

Out[4]:

```
(21, 2)
```

In [5]:

```
data.isna().sum()
```

Out[5]:

```
Delivery Time    0
Sorting Time     0
dtype: int64
```

In [6]:

```
data.dtypes
```

Out[6]:

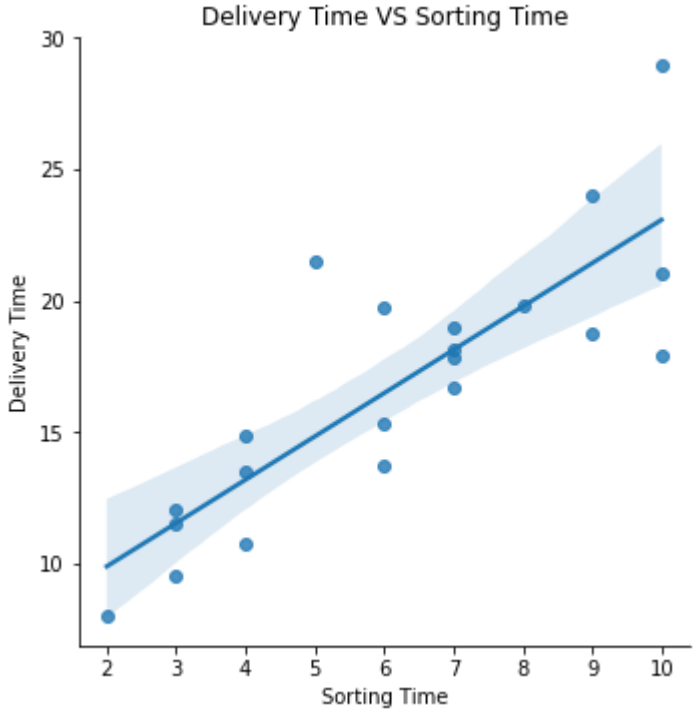
```
Delivery Time    float64
Sorting Time       int64
dtype: object
```

3.2 Assumption Check

3.2.1 Linearity Test - Linearity test has failed

In [13]:

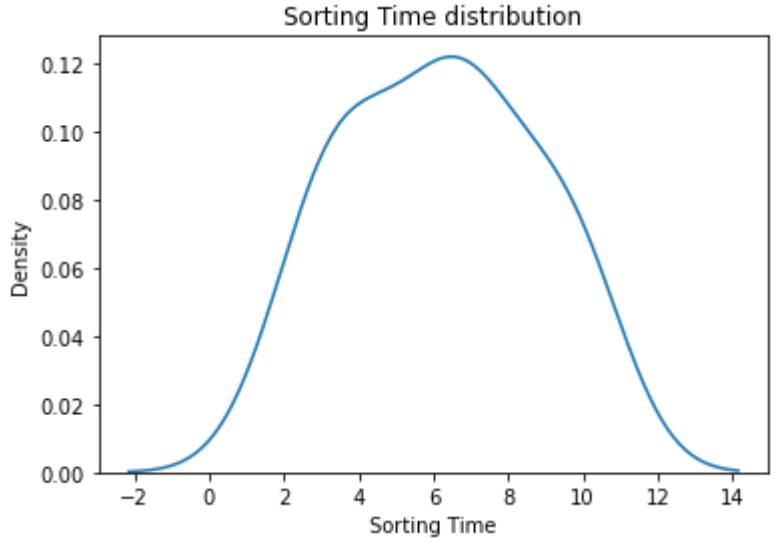
```
sns.lmplot(x = 'Sorting Time',y = 'Delivery Time',data= data)
plt.title('Delivery Time VS Sorting Time')
plt.show()
```



3.2.2 Normality Test - Normality test has failed

In [16]:

```
sns.distplot(a = data['Sorting Time'],hist = False)
plt.title('Sorting Time distribution')
plt.show()
```



4. Data prepration

In [21]:

```
data.rename(columns={'Delivery Time':'Delivery','Sorting Time':'Sorting'},inplace = True)
data.head()
```

Out[21]:

	Delivery	Sorting
0	21.00	10
1	13.50	4
2	19.75	6
3	24.00	9
4	29.00	10

5. Model Building and Model Training

In [23]:

```
linear_model = smf.ols(formula = 'Delivery~Sorting',data = data).fit()
linear_model
```

Out[23]:

```
<statsmodels.regression.linear_model.RegressionResultsWrapper at 0x1c21a4dcb
50>
```

In [24]:

```
linear_model.params
```

Out[24]:

```
Intercept    6.582734
Sorting      1.649020
dtype: float64
```

6. Model Testing

In [25]:

```
data.head()
```

Out[25]:

	Delivery	Sorting
0	21.00	10
1	13.50	4
2	19.75	6
3	24.00	9
4	29.00	10

In [27]:

```
x_test = pd.DataFrame(data= {'Sorting':[11,12,13,14,15,16,17,18,19,20]})
x_test
```

Out[27]:

	Sorting
0	11
1	12
2	13
3	14
4	15
5	16
6	17
7	18
8	19
9	20

In [33]:

```
y_test = linear_model.predict(x_test)
```

In [40]:

```
y_test_1 = pd.DataFrame( data = {'Delivery': y_test})
y_test_1
```

Out[40]:

	Delivery
0	24.721953
1	26.370973
2	28.019993
3	29.669013
4	31.318032
5	32.967052
6	34.616072
7	36.265092
8	37.914112
9	39.563132

In [41]:

```
test_data = pd.concat([x_test,y_test_1],axis = 1)
test_data
```

Out[41]:

	Sorting	Delivery
0	11	24.721953
1	12	26.370973
2	13	28.019993
3	14	29.669013
4	15	31.318032
5	16	32.967052
6	17	34.616072
7	18	36.265092
8	19	37.914112
9	20	39.563132

In [ ]:

```

```