In [1]:

```
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd
import pprint
import xgboost as xgb
from sklearn import linear_model
from sklearn.model_selection import train_test_split
from sklearn import ensemble
```

In [2]:

```
demands = pd.read_csv('demands.csv')
```

In [3]:

```
df_x = demands.iloc[:, 0:1]
df_y = demands.iloc[:, 1]
```

Figure 1

In [4]:

```
plt.scatter(df_x, df_y)
```

Out[4]:

```
<matplotlib.collections.PathCollection at 0x23e14be0e80>
```

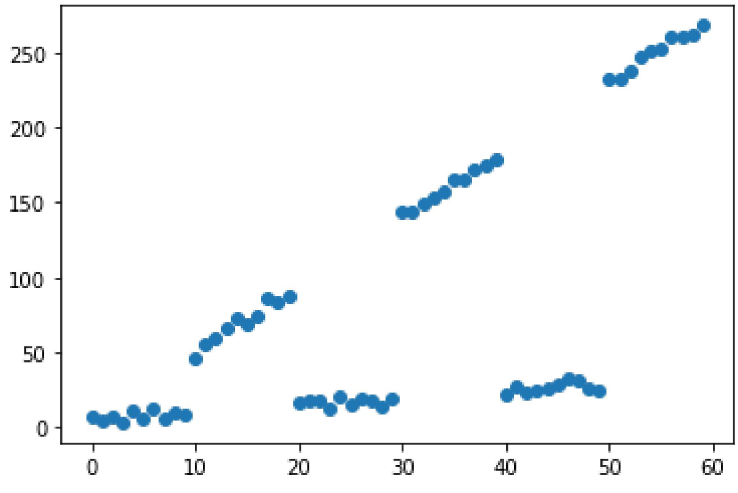


Figure 2

In [5]:

```
model1 = linear_model.LinearRegression()
model1.fit(df_x,df_y)
y_pred = model1.predict(df_x)
plt.scatter(df_x, df_y)
plt.plot(df_x, y_pred, color='red')
plt.show()
```

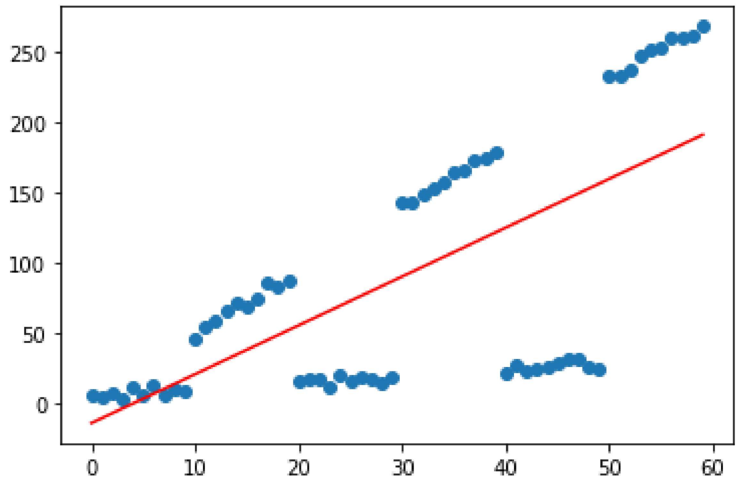


Figure 3

In [6]:

```
params = {'n_estimators': 1, 'max_depth': 1, 'learning_rate': 1, 'loss': 'ls'}
model2 = ensemble.GradientBoostingRegressor(**params)
model2.fit(df_x, df_y)
```

Out[6]:

```
GradientBoostingRegressor(learning_rate=1, max_depth=1, n_estimators=1)
```

In [7]:

```
y_pred = model2.predict(df_x)
plt.scatter(df_x, df_y)
plt.plot(df_x, y_pred, color=’red’)
plt.show()
```

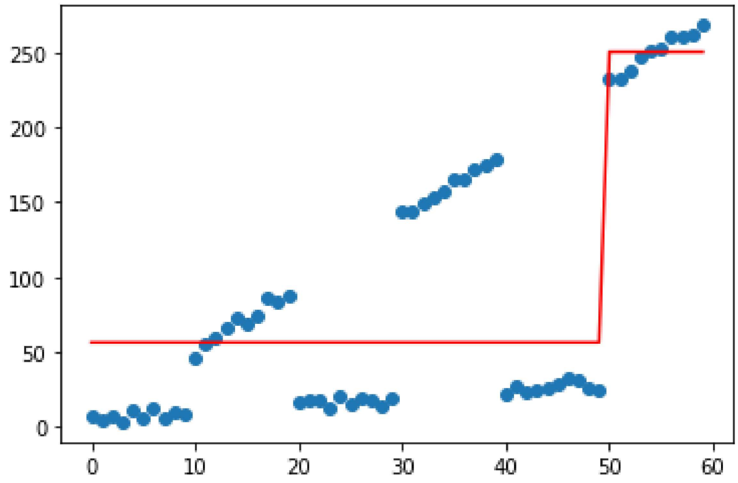


Figure 4

In [8]:

```
params = {’n_estimators’: 2, ’max_depth’: 1, ’learning_rate’: 1, ’loss’: ’ls’}
model3 = ensemble.GradientBoostingRegressor(**params)
model3.fit(df_x, df_y)
```

Out[8]:

```
GradientBoostingRegressor(learning_rate=1, max_depth=1, n_estimators=2)
```

In [9]:

```
y_pred = model3.predict(df_x)
plt.scatter(df_x, df_y)
plt.plot(df_x, y_pred, color=’red’)
plt.show()
```

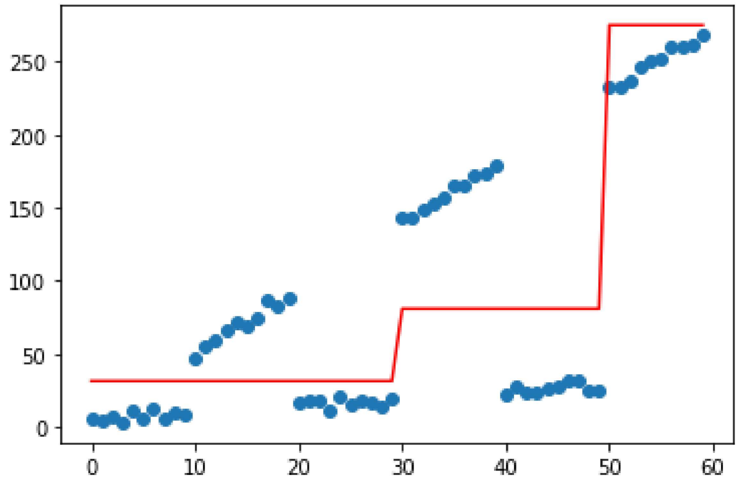


Figure 5

In [10]:

```
params = {’n_estimators’: 5, ’max_depth’: 1, ’learning_rate’: 1, ’loss’: ’ls’}
model4 = ensemble.GradientBoostingRegressor(**params)
model4.fit(df_x, df_y)
```

Out[10]:

```
GradientBoostingRegressor(learning_rate=1, max_depth=1, n_estimators=5)
```

In [11]:

```
y_pred = model4.predict(df_x)
plt.scatter(df_x, df_y)
plt.plot(df_x, y_pred, color=’red’)
plt.show()
```

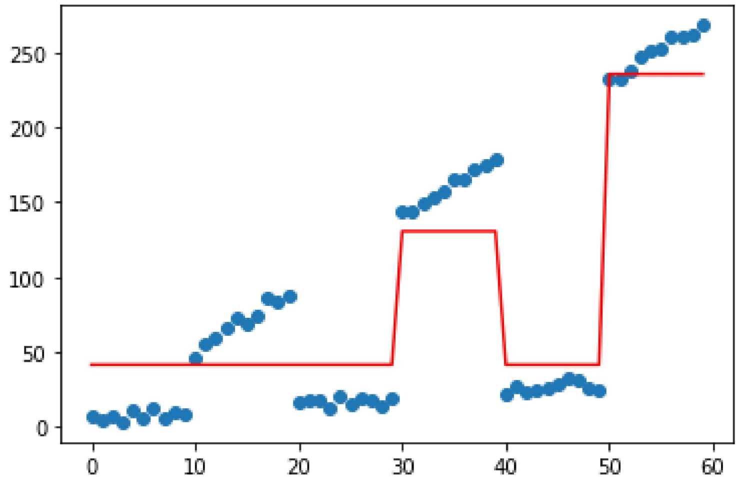


Figure 6

In [12]:

```
params = {’n_estimators’: 10, ’max_depth’: 1, ’learning_rate’: 1, ’loss’: ’ls’}
model5 = ensemble.GradientBoostingRegressor(**params)
model5.fit(df_x, df_y)
```

Out[12]:

```
GradientBoostingRegressor(learning_rate=1, max_depth=1, n_estimators=10)
```

In [13]:

```
y_pred = model5.predict(df_x)
plt.scatter(df_x, df_y)
plt.plot(df_x, y_pred, color='red')
plt.show()
```

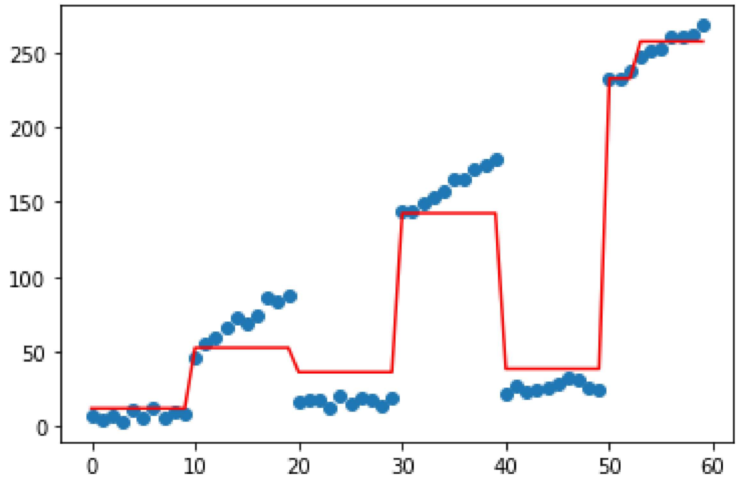


Figure 7

In [14]:

```
params = {'n_estimators': 20, 'max_depth': 1, 'learning_rate': 1, 'loss': 'ls'}
model6 = ensemble.GradientBoostingRegressor(**params)
model6.fit(df_x, df_y)
```

Out[14]:

```
GradientBoostingRegressor(learning_rate=1, max_depth=1, n_estimators=20)
```

In [15]:

```
y_pred = model6.predict(df_x)
plt.scatter(df_x, df_y)
plt.plot(df_x, y_pred, color='red')
plt.show()
```

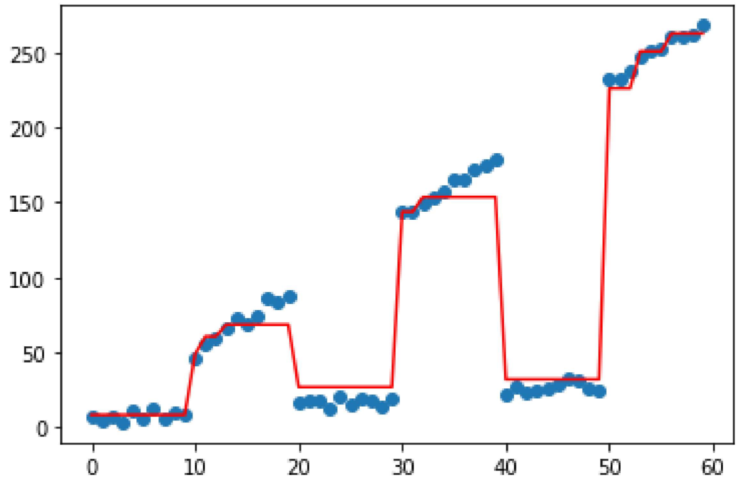


Figure 8

In [16]:

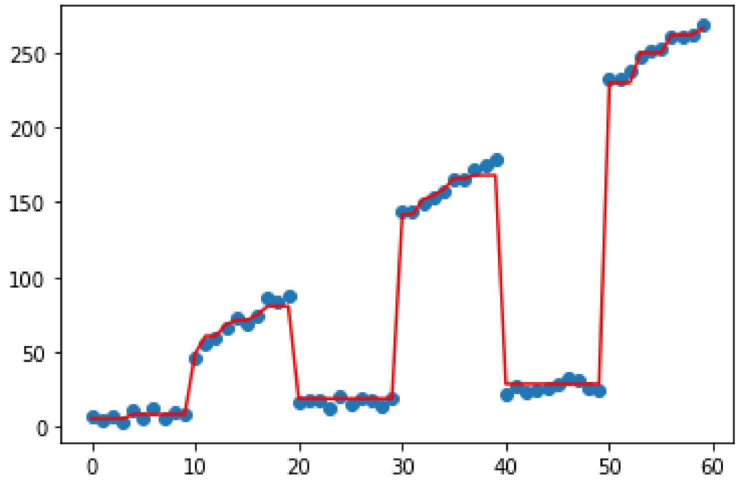
```
params = {'n_estimators': 50, 'max_depth': 1, 'learning_rate': 1, 'loss': 'ls'}
model7 = ensemble.GradientBoostingRegressor(**params)
model7.fit(df_x, df_y)
```

Out[16]:

```
GradientBoostingRegressor(learning_rate=1, max_depth=1, n_estimators=50)
```

In [17]:

```
y_pred = model7.predict(df_x)
plt.scatter(df_x, df_y)
plt.plot(df_x, y_pred, color='red')
plt.show()
```


250
200
150
100
50
0
0
10
20
30
40
50
60