Wavelet Transform

In this example we will be taking a wind speed file and putting it into the Raven wavelet transform. In numerical analysis and functional analysis, a discrete wavelet transform is any wavelet transform for which the wavelets are discretely sampled. In Raven this can be used for further characterization and generation.

Wavelet Algorithm

Below is the wavelet algorithm.

$$y[n] = (x * g)[n] = \sum_{k=-\infty}^{\infty} x[k]g[n-k]$$

Wavelet In Raven

Below is a domonstration on filtering and passing data into Raven via the wavelet transform. The results are also graphed in this example. The original data is in red and the Raven prediction is in blue. As we can see wavelet fits this data very well.

In [1]:
```
%%capture
#Import the necessary libraries and tools to use.
import pandas as pd
import numpy as np
```

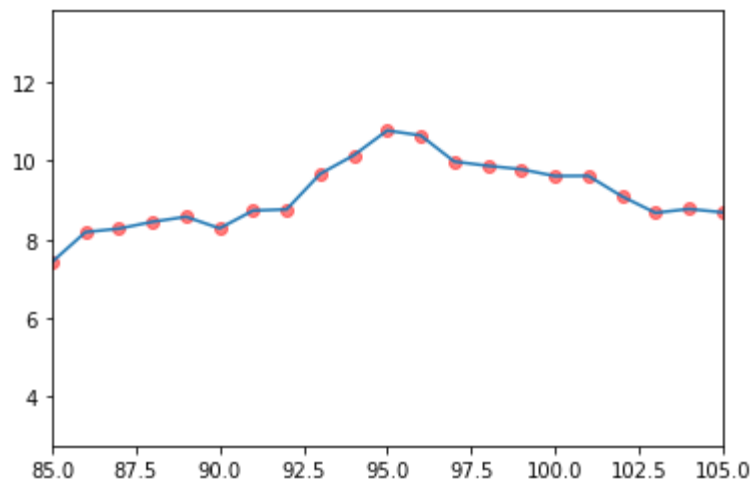
In [2]:
```
#Read the original dataset
data = pd.read_csv("raw_data/T1.csv")
```

In [3]:
```
#Filter the data
dataLatest = data[['LV ActivePower (kW)','Wind Speed (m/s)']]
```

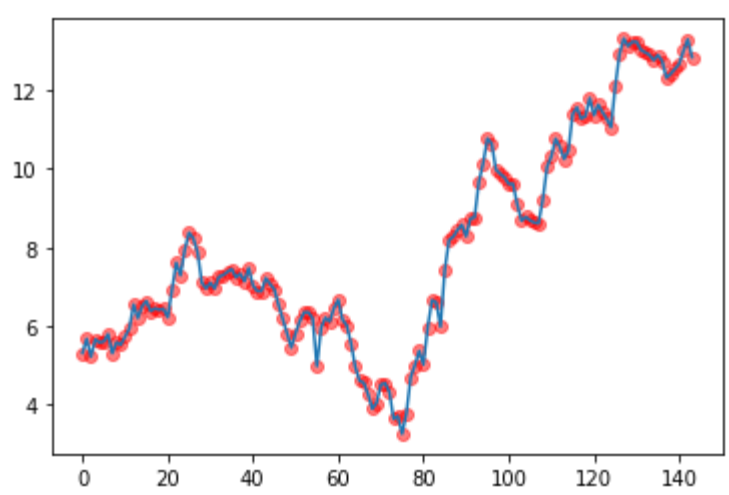
In [4]:
```
#Filter the data to just 145 lines to input into Raven.
df = dataLatest.iloc[0:144]
df.to_csv('raw_data/T2.csv')
```

In [5]:
```
#Import the library necessary to create the graphs.
import matplotlib.pyplot as plt
```

In [6]:
```
#Graph the results from Raven and the original data. This is a specific peak in the data below.
y = df['Wind Speed (m/s)']



dat = pd.read_csv('raw_data/T3.csv')

data = pd.read_csv('raw_data/T4.csv')


plt.plot(dat.seconds, y, 'o', alpha=0.5, color='red')
plt.plot(dat.seconds, dat.signal2)
plt.xlim(85,105)
plt.show()
```



In [7]:
```
#This is a comprehensive Graph that covers the entire dataset input into Raven.
plt.plot(dat.seconds, y, 'o', alpha=0.5, color='red')
plt.plot(dat.seconds, dat.signal2)
plt.show()
```



In [ ]:
```

```