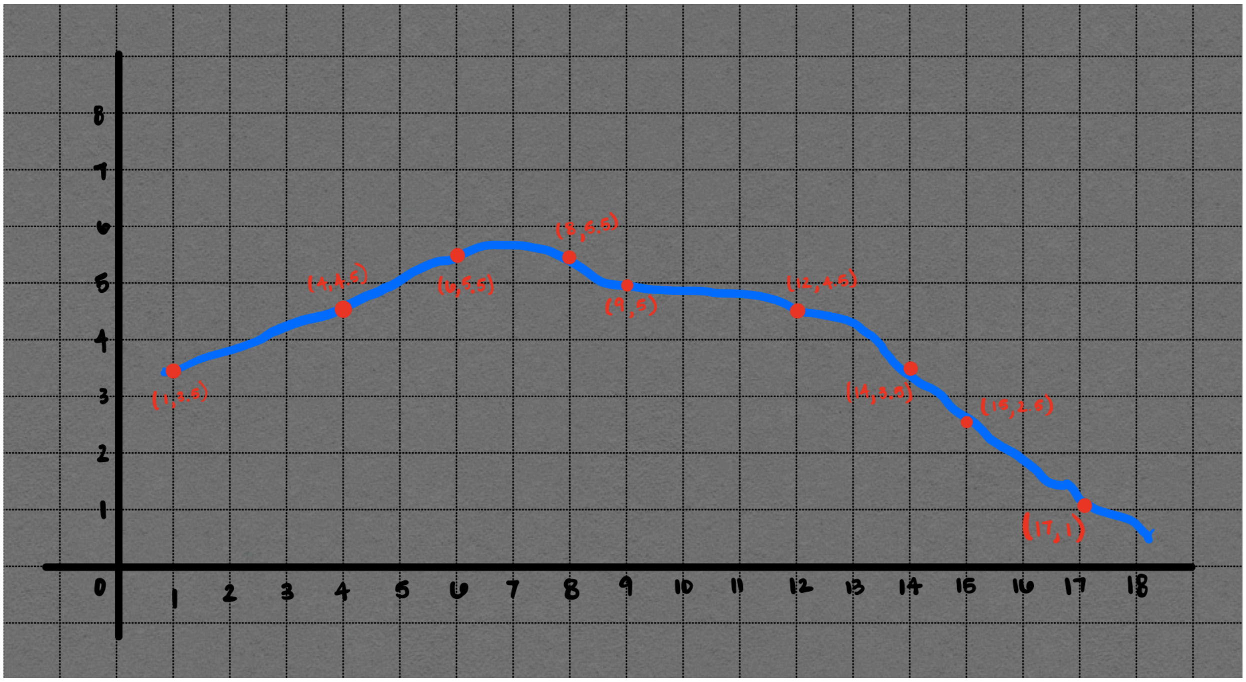
HW 1: Ch19.4 Cubic Splines

Bailey Williams

In this assignment, we use Python to fit a cubic spline to the data points obtained by tracing hand profile onto graph paper.

Data Points

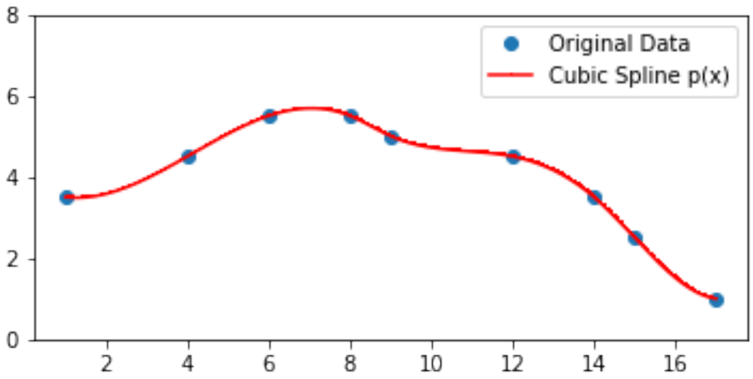
In the figure below, the plot of the data points is shown from an image of my hand profile.



Python Code and Output

In [11]:

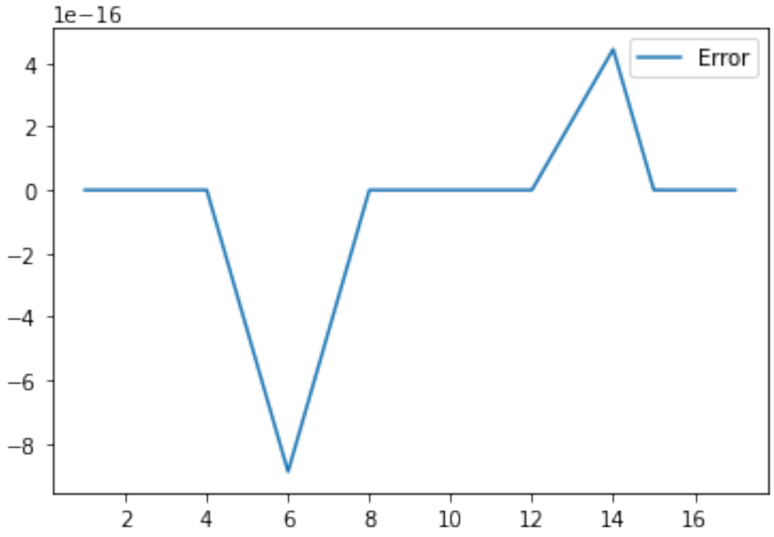
```
import numpy as np
import matplotlib.pyplot as plt
from scipy.interpolate import interp1d

#Enter Data
x = [1, 4, 6, 8, 9, 12, 14, 15, 17]
y = [3.5, 4.5, 5.5, 5.5, 5, 4.5, 3.5, 2.5, 1]

#Command for cubic splines spline polynomial p(x)
p = interp1d(x, y, kind = 'cubic')

#Create vector of 500 points between 0[n] and x[n-1] on x-axis
n = len(x)
xnew = np.linspace(x[0], x[n-1], num=500, endpoint=True)

#Plot the original data together with p(x) sampled at the 500 points
plt.plot(x, y, 'o', xnew, p(xnew), '-,r')
plt.legend(['Original Data', 'Cubic Spline p(x)'], loc = 'best')
plt.axis('square')
plt.ylim(0,8);
plt.show()
```



Error Code and Output

In [17]:

```
plt.plot(x, p(x)-y)
plt.legend(['Error'], loc = 'best')
plt.show()
```



Discussion of Results

As seen in the graph above, the python calculated cubic spline interpolates the data points which provides a smooth trend in the data without large ocillations between data points. This spline provides a reasonably accurate representation or estimation between data points, but not outside the range of the data points (extrapolation). The beauty of the cubic spline interpolant is how well it approximates a function with little error.

In [ ]: