University of Warwick
Department of Physics

Author:
Jason Clarke

Course:
MMath G103

Student Number:
1205556

C Programming Project 3 Report

January 2014

1 Running the program

Write 1205556_proj3 a b where a is the window size, and b is the number of bins in the histogram. In the specification, we are told to validate the date so that the window is not too large or too small. I arbitrarily decided that bigger than 1/20th of the input size is too large and smaller than 1/200th of the input size is too small. Since these values are plucked from thin air I decided to make the program just give a warning and a suggestion, but allow the program to continue if the value used is between 1 and the input size. Else the program will exit.
I arbitrarily set the maximum number of histogram bins to 1000. Using a number not between 1 and 1000 will cause the program to exit.

2 Notes on Calculations

1.To take the boxcar average, I first calculated the number of segments I would require based on the window size w by division:

nosegments = ceil(((double)inputsize)/((double)w));

The ceiling function ensures a segment for remainder terms if the window size does not divide the input size exactly. I calculated the average of all but the last segment, which are certainly complete, with a for loop, dealing with the last segment separately. I then found out the size of the last segment by using a modulo operation:

int remainder = inputsize%w;

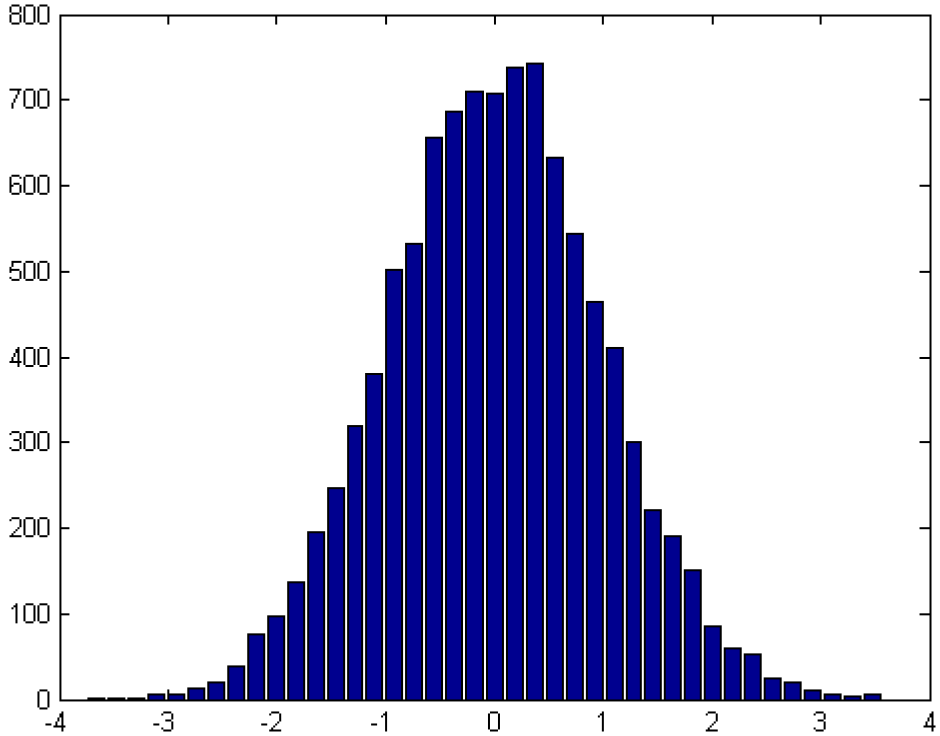
If the window size divides the input size exactly, then the modulo operation gives 0. However in this case the last segment will have exactly w terms, not 0. So I had to write a quick if statement to correct this. Now knowing the number of terms and their values I could take the average of the last segment easily.

2.I found $S_{\overline{y}}$ in the following way. Starting with the first segment, I took the average value of the segment, multiplied by the number of values in the segment (w for all but the last segment), and then added this value to a cumulative total. Repeating this process for all segments, the resultant cumulative total is equal to the sum of all the average values added individually. However, with this method, all the average values are summed in their window groups, and the groups are then summed together.

3.To find $S_{t\overline{y}}$ I summed each term individually. The term $\overline{y}_i$ is described as:
boxcardata[i/w]
where i, w are ints. Note that integer division is used, so the result of i/w here is floor(i/w). In boxcardata[k], the average value of the k-th segment is held. So if $\overline{y}_i$ belongs to the k-th segment its value is boxcardata[k]. $\overline{y}_i$ belongs to the k-th segment if lies between $k * w \leq i \leq ((k + 1) * w) - 1$. For all these values of i, floor(i/w) is k. So boxcardata[i/w] yields the correct value.

3 Graphical Results

Using value 128 for the window size, 40 for the number of bins, and proj3input yields the following histogram of detrended data. The width of the histogram represents the width of the bins. The x-axis shows the range of values for each bin. The y-axis shows the frequency of detrended data elements in each bin.



4 Extension: Compatibility with Matlab

I wrote an extension to the code which writes an m-file called histogram.m which can then be used to draw the histogram produced from the arguments. The extension also creates an m-file called detrend.m which shows how the data is detrended. The x-axis represents time. The y-axis represents the value of the data element. The dark blue line is the initial data. The bumpy pink line is the boxcar-smoothed data. The red line is the least-square line. The green line is the detrended data. The cyan line is the mean of the detrended data (which is zero as one might hope).