|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **FILTER NAME** | **INPUT LENGTH** | **OUTPUT LENGTH** | **FILTER LENGTH** | **ADVANTAGES/ COMMENTS** | **DISADVANTAGES** |
| Median filter 2, **medFilter2()** | len(x) | len(x) | n > 2  n is odd.  (Reason is that using an even n gives an unbalanced result.) | Replaces extreme noise and outlier values with values like surrounding values.  Same length output. | Tends to lower higher data values near start and end of sequence because of adding zeros.  This can be overcome by having extra data at start and end. First do filtering and after do trimming at start and end. |
| Median filter, **medFilter()** | len(x) | len(x) | n > 2  n is odd.  (Reason is that using an even n gives an unbalanced result.) | Replaces extreme noise and outlier values with values like surrounding values.  Same length output.  **medFilter()** does not tend to lower higher data values near start and end of sequence like **medFilter2()** because averages of start numbers and averages of end numbers are added instead of adding zeros. |  |
| Moving average, small reduction filter, **movAve()** | len(x) | len(x)-n+1 | n > 1  n is odd or even. | Values throughout the sequence are treated equally in the averaging. | Output length is a little shorter. |
| Moving average, large reduction filter, **aveReduct()** | len(x) | ceil(len(x)/n)  Example for len(x)=15,n=4:  ceil(15/4)  =ceil(3.75)=4 | n > 1  n is odd or even. | Greatly reduce length of sequence.  Values throughout the sequence are treated equally in the averaging, unless there are less than n values in the last group to be averaged. | Greatly reducing the length might throw away valuable information. |
| Moving average filter, no reduction, **smooth2()** | len(x) | len(x) | n > 2  n is odd.  (Reason is that using an even n gives an unbalanced result.) | Same length output.  Values near start and end are unchanged. | Values near start and end are unchanged.  **Ways to Overcome:**  1. This can be overcome by having extra data at start and end. First do filtering and after do trimming at start and end.  2. Use other filters before this one. |
| Moving average filter, no reduction, **smooth()** | len(x) | len(x) | n > 2  n is odd.  (Reason is that using an even n gives an unbalanced result.) | Same length output.  Values near start and end are also smoothed. |  |
| Time reduction, first value filter, **timeReduct()** | len(x) | ceil(len(x)/n)  Example for len(x)=15,n=4:  ceil(15/4)  =ceil(3.75)=4 | n > 1  n is odd or even. | **COMMENT**: Same as **aveReduct()** except that the first value of each group is selected, rather than the average. | **OPINION**: I think **aveReduct()** is better because the time average may be a better time representation to associate with other data averaged to remove noise. |
| Shift to new start & end, **scale()** |  |  |  |  |  |
| Shift data to match time, dataTimeShift() |  |  |  |  |  |
|  |  |  |  |  |  |