Original Data: Draft

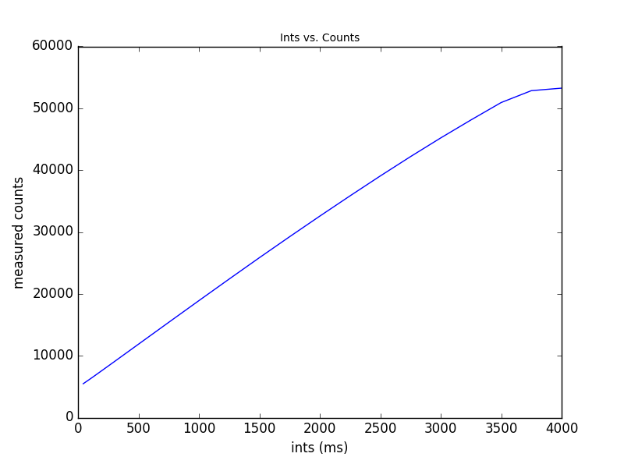
Chris Bohlman

11/03/2016

Context:

The context to this report is to check known data in order to view how the program works. Because the results of the linearity of this data set is know, I wanted to check my program writing to see if my results were on par with what was already observed. Through this, I could then look back upon my programming and see if what I did was correct. Here are the results I gained compared with what was obtained in the original report.

Ints vs. Counts:



Read files in through fits

Adjusted parameters to equal original analysis’ parameters

Analyzed counts in that section of the image for each picture

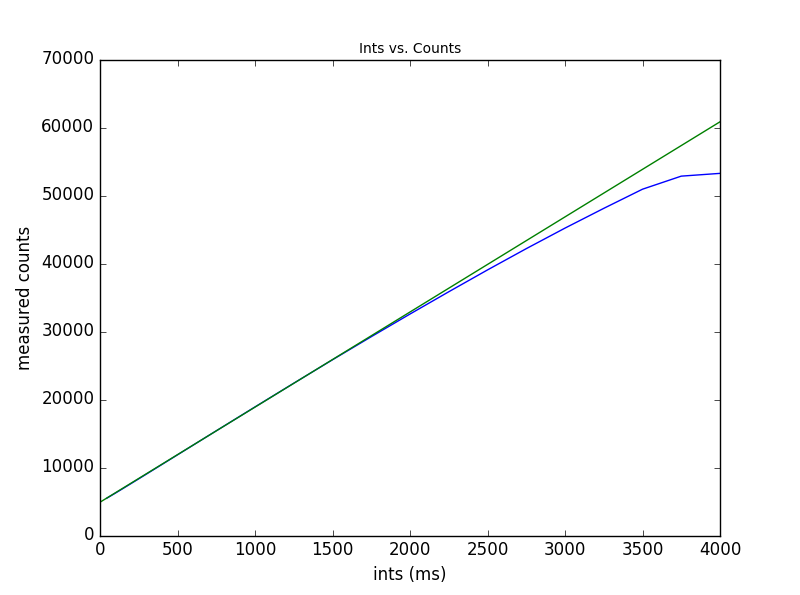
Grabbed ints measurement from each f the headers in the picture.

Data was stored in ‘int’ and ‘counts’ arrays, respectively.

Graphed ints on x axis and counts on y axis.

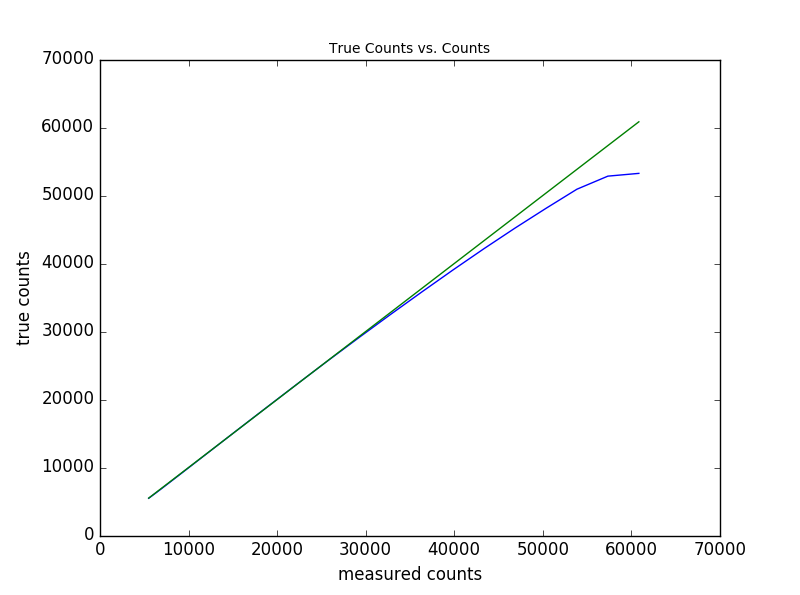
Looks exactly like original data - elaborate

Ints vs. Counts with linear relationship:



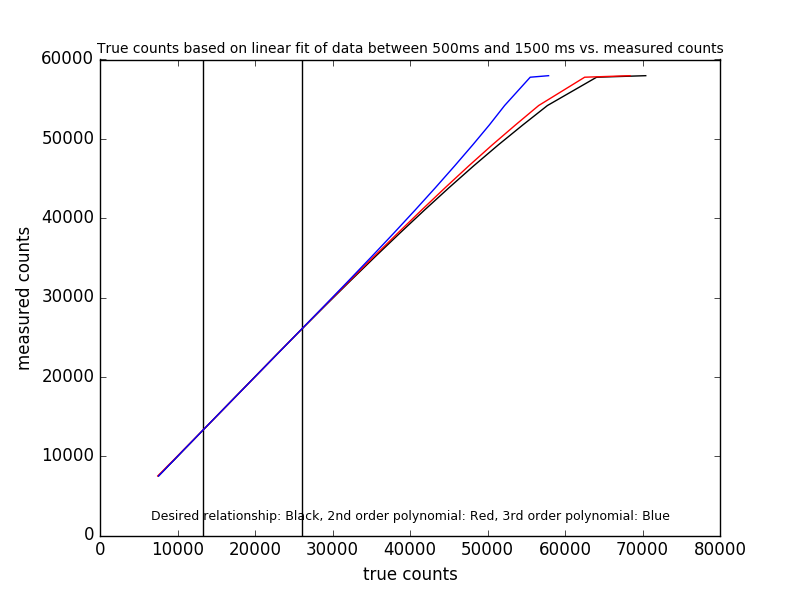
Coefficients for the line: [ 4894.75638298 13.98609149]

True counts vs. Measured Counts with linear:



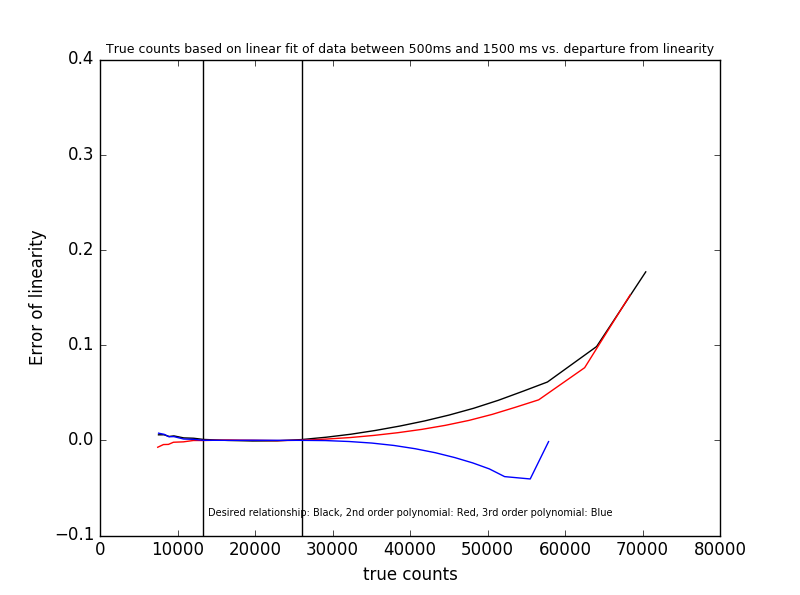
Explanation

True counts vs. Measured Counts with all:



In the end, what was wanted was to see which order equation could model most of the data the best. Therefore, once I had the linear version of the true counts, I calculated the coefficients of a second order equation per the data, and a third order equation to the data. When getting up to higher orders, it was important that I modify the lists of ints/counts a little bit. Because all the ints were the same in groups of 5, and all the coefficients in this group were slightly different, for every int, I averaged out the amount of counts across the five counts for a single int, and made a list of every int without repeats and the subsequent averaged out count. I then used polyfit on these to model second and third order relationships, and redid the first order linear which came up with coefficients that were very close to the coefficients for the previous true counts line, so that did not cause a problem in my data, since that was going to be my desired relationship of data. I then graphed true counts vs counts for every single order the same way I did it in the method of above, and graphed them on the same graph as each other. The vertical lines are representations of where the coefficients for each of the lines were gathered from.

Error of Linearity vs. True Counts



In the end, we wanted to see which order (2nd or 3rd) was closest to the desired relationship of first order. Therefore, stemming from the last graph, I produced an error of linearity plot. Now the true counts axis stayed the same, but for the error of linearity, I used the equation

*Error of linearity* = (*True Counts – Measured Counts)/True Counts*

For every single count, I calculated the error of linearity and stuck that into a list. I then plotted the error of linearity compared to the true counts for first order, second order, and third order, and analyzed the results based off how close the second and third order errors were to the desired relationship of first order.

Analyzing the Data:

From the error of linearity plot, I discovered that the second order equation behaved more closely to the desired relationship. Now, the reason I couldn’t add fourth order is because there were only 23 items on each list (due to me averaging the ints and counts into separate lists), so the fourth order polyfit had some odd behavior. After consolidating my data and limits and methods across three python files, I finally came up with that error of linearity graph, which was a marked improvement over what I was producing previously. However, because I couldn’t get any order to exactly match up to the desired relationship, I was at a loss for things to do. That’s why I am writing this report and organizing all my files: to get some ideas or something, or to notice a method that I did wrong.

Here’s to hoping that happens.