8/10

Homework 1

Introduction

I opted to use R over Python or C for this project as I have more prior experience in R. However in general I struggled to force R to use single precision. It was partially possible through the use of external packages to do so though it didn't always work as expected. If I was to redo this project I would have rather used Python. There were some advantages in using R for plotting and data manipulation that came in useful though.

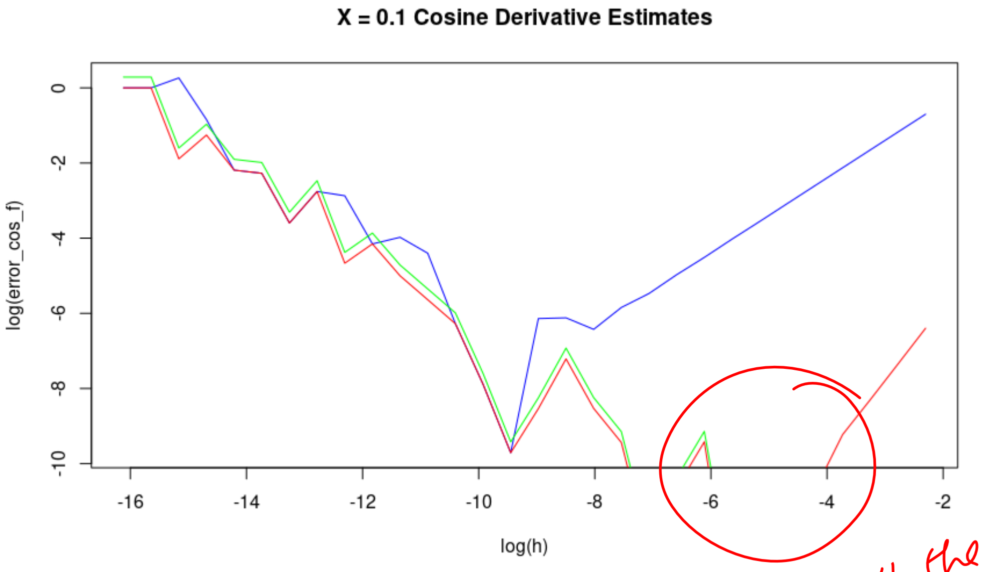
ok!

1

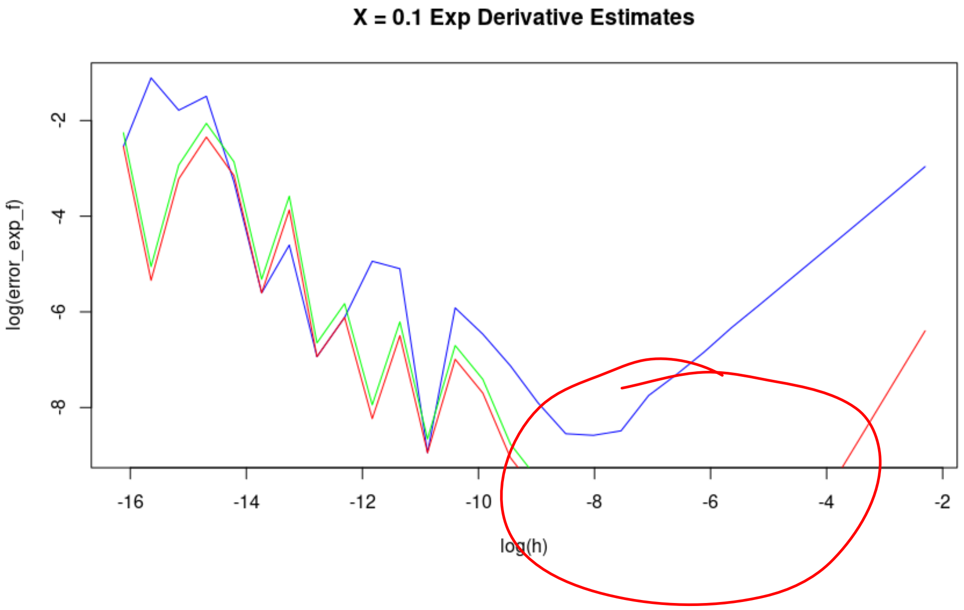
need a key!

The colour correspondence is as follows; blue is the forward difference method, red is the central difference method, green is the extrapolated difference method. In the Log-Log plots of each derivative estimation technique it is clearly visualized where truncation and round off error dominate. When h is too large, the estimate is too "rough" causing truncation error. This is visible in the left portion of the graph trending downwards as h decreases to a minimum value. Eventually, h decreases to a size where the effect of round-off error dominates. The optimal value of h in theory should be around the square root of the machine precision. Which for Single precision would mean the optimal h is between 10^{-3} and 10^{-4}. However it's apparent in my graphs that this switch occurs around 10^{-9}. This is probably due again to issues with R interpreting single precision floats incorrectly.

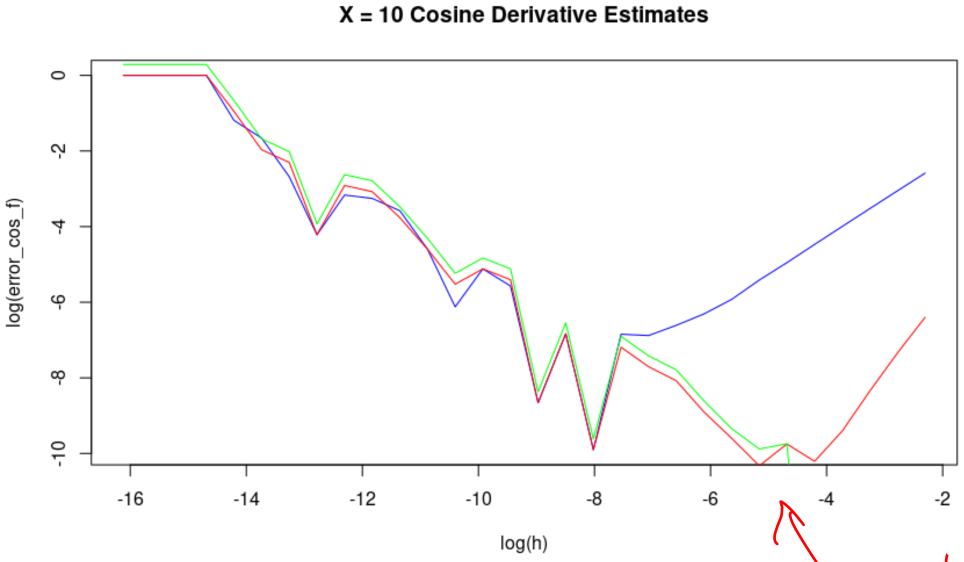
You can change these for double since based on ϵ_m.



all the important info happened outside the plot!

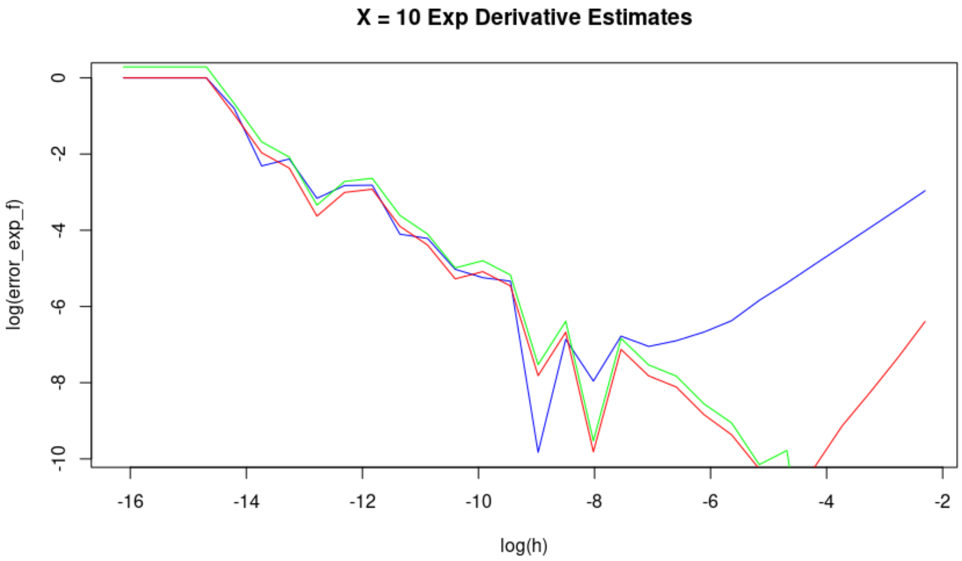


X = 0.1 Exp Derivative Estimates



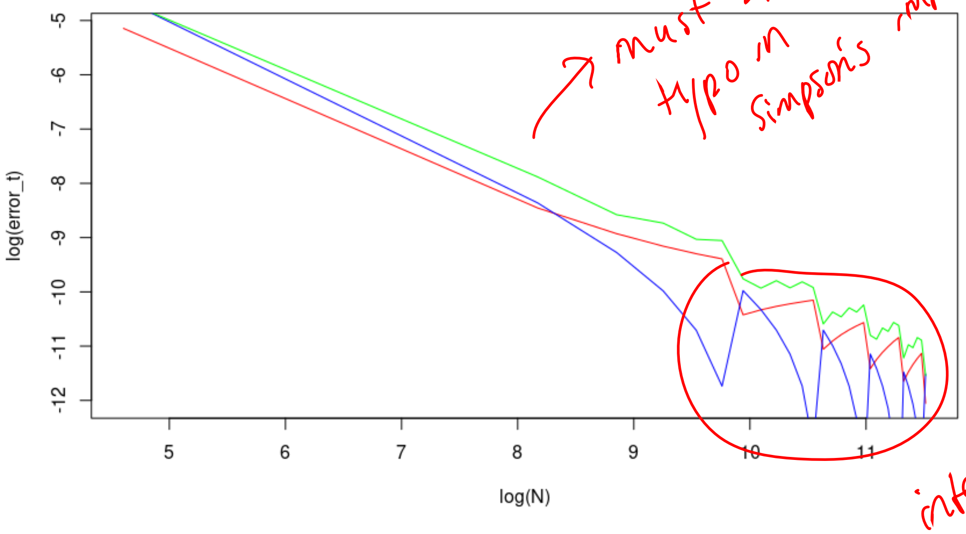
X = 10 Cosine Derivative Estimates

what happened to green?



2

In the code, the colour correspondence is as follows; midpoint is blue, trapezoid is red, and Simpson's is green. The graph indicates the decrease in error of each method as the number of bin's increases. There are several issues with this graph and how it differs from what should be expected. Namely, the Simpson's method should have the least amount of error which it clearly does not in this graph. As the error decreases and the integral converges, eventually the step size gets too small and round off error should kick in as it did in problem one. An effect definitely begins to occur at a similar scale to problem one however the overall error continues to trend down despite the change in behaviour. Again I suspect this to be caused by issues in forcing R to interpret single precision variables. I was unable to find a sufficient work around to this, and had I left things entirely in double precision the Simpson's method performed correctly, however the round-off error would not be visible.



must be typo in Simpson's implementation?

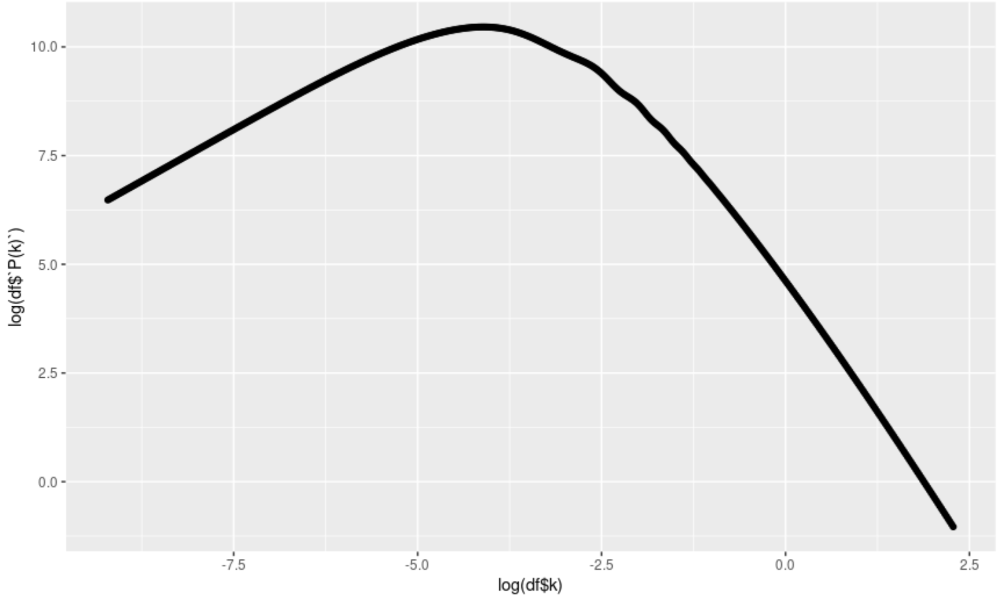
actually these all look like N^{-1} scaling

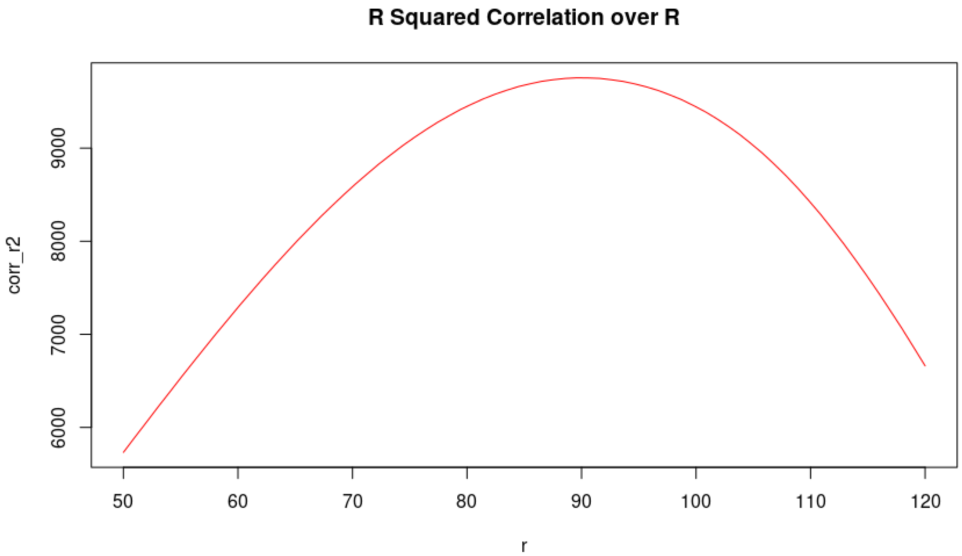
Does that agree w/ any expectation?

interesting that R did this, rather than turn up.

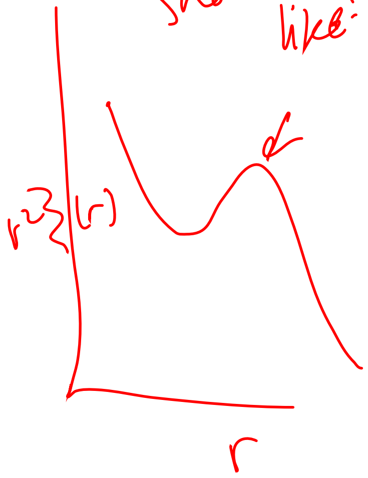
3

For problem 3, after importing the data and stripping the excess columns I interpolated using an built-in function of R. The first plot is a log-log plot of my interpolation to check the behaviour and smoothness.The second plot is my $r^2\xi$ with peak around $r = 93Mpc/h$ with value approximately 9800.





Should look like:



-1