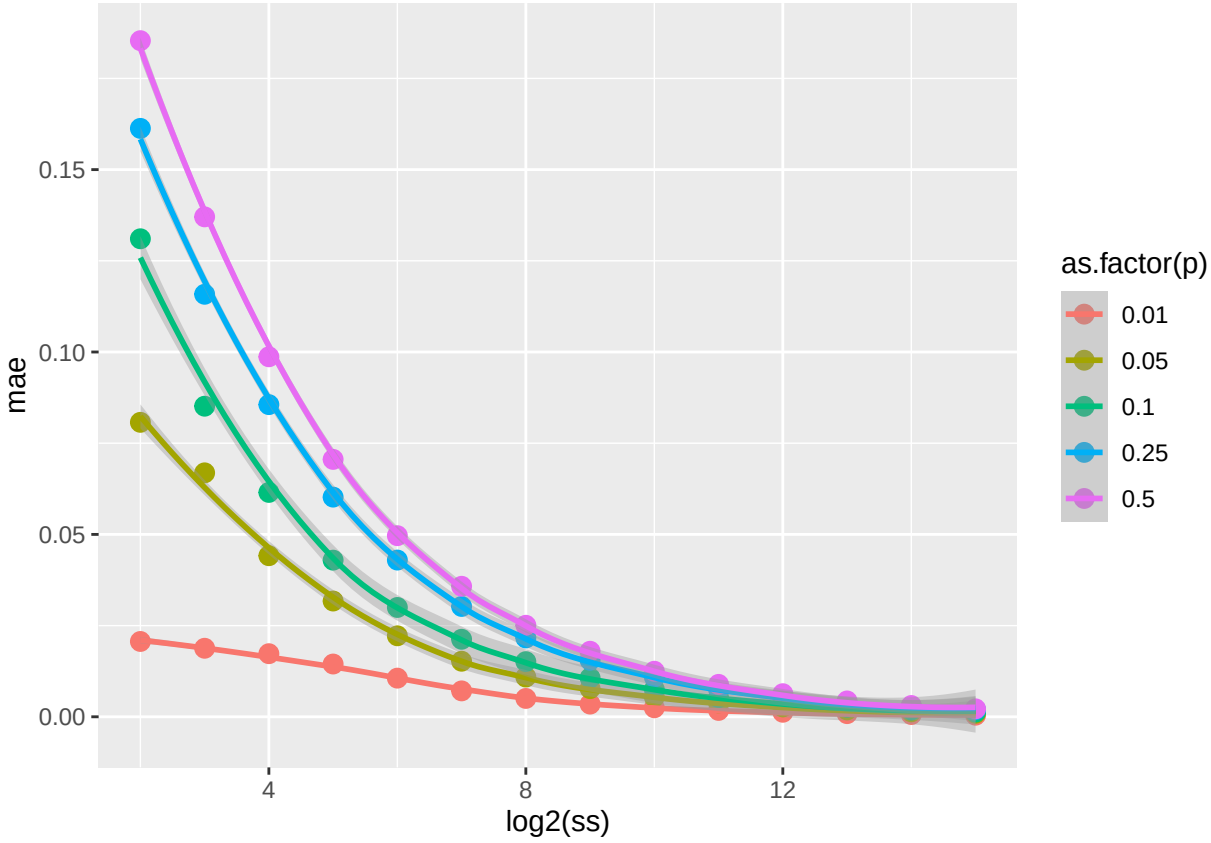
Tonnar Castellano Monte Carlo Error

Last post we discussed how to implement a simulation of a roulette game and what parameters were important in that simulation. This post we will also look at a Monte Carlo simulation; however, we are going to be more interested in the error and how that changes over time. We will look at two error metrics - absolute error and relative error.

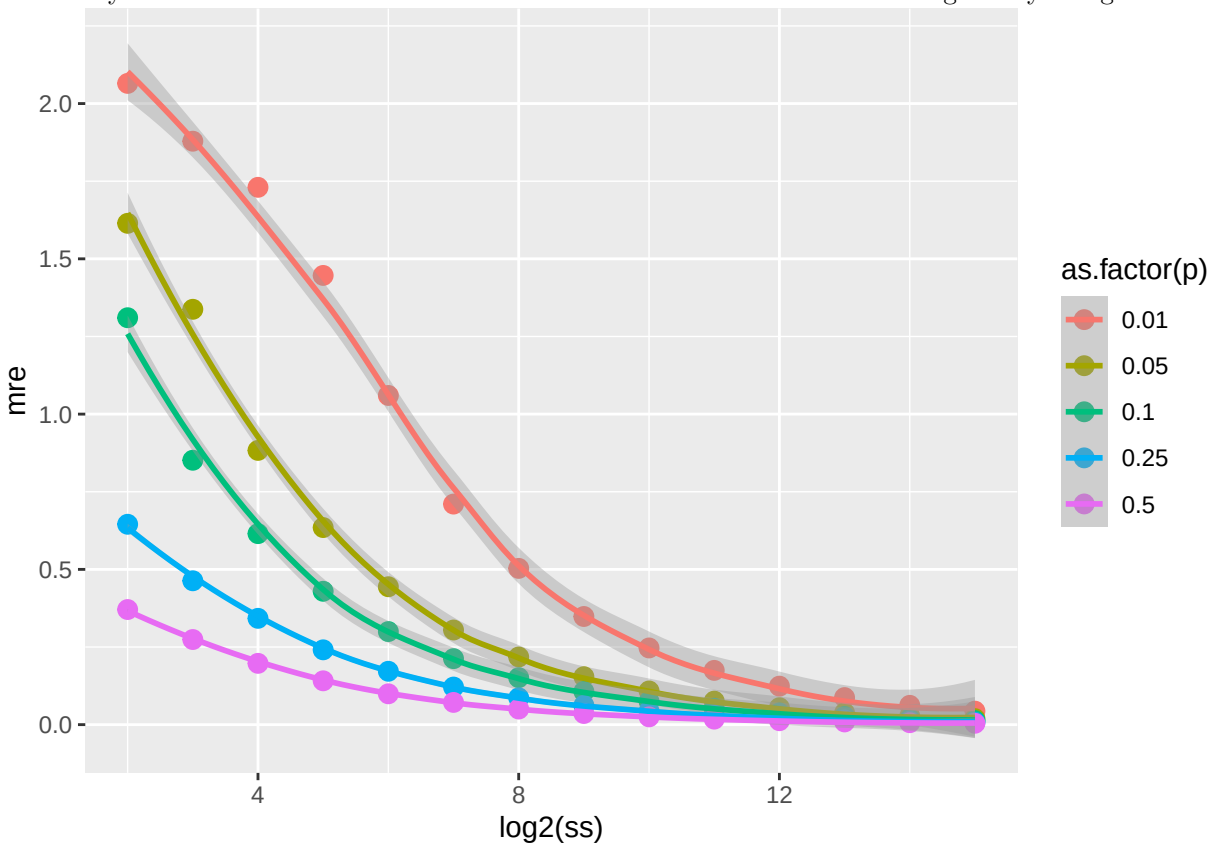
The first type of error I will discuss is absolute error. The formula for absolute error is given as $|p - \hat{p}|$ where p is the known probability and $\hat{p}$ is what our simulation has, well, simulated. The formula is obviously straight forward. We are finding the difference between what should be and what we simulated. Something worth noting is that in the code above $|\hat{p} - p|$ is used but because of the absolute value it does not matter. Below you can see the plot of the absolute error with a regression line helping to show where we could predict our error would fall.



As you can see the purple values with a smaller sample are much higher in terms of the absolute error. The reason is that you have a higher chance of getting a "run" of values. Consider the chance of getting 3 heads in a row $.05^3$ compared to the chance of picking 1 blue out of 99 red balls 3 times in a row with replacement. $.01^3$. As such, you would expect the estimated probability to diverge more frequently from the true probability.

Next measure of error is relative error. Relative error means how much we deviate from the true probability relative to what we expect. This is given by the formula $\frac{|p-\hat{p}|}{p}$. In an intuitive sense, if we have a very low probability for failure we would scale the difference by a lot because it would be high relative to the

expectations. For example in the case of .5 we would expect it to deviate by a lot so we don't weigh each failure by as much. If we have a .01 chance of failure we would want to weigh it by a higher amount.



In the case of the graph above that is exactly what you expected to see. The reason these converge to 0 is that as the iterations increase we get closer and closer to 0 so we get closer and closer to what is expected irregardless of how relative it is to itself.