Hi Sunny –

Thanks so much for putting together the most recent analysis. Seeing the final dataframe makes it easier to understand what’s likely going on internally with the model and also helps guide direction on where we should take it.

After looking things over… a couple things come to mind:

1 - The results seem to be the same for each maxlag increment, which tells me that each iteration is landing on the same lag to use (lag of 1 for example).

2 - The standard error values for the 450+ tickers vary a lot between all the tickers. So if our hypothesis is that a low standard error means the ticker better contributes to the modeling of csi1, then it would seem that we can prioritize these low standard error tickers for our model.

These two things seem pretty unrelated. I’d say let’s skip #1 and I’ll try and research this one some more independently. I think I need to study the model’s internal mechanics more to better understand why the model prefers a lag of 1 for each iteration.

Let’s instead focus on #2 and come up with some theories here. For next steps, can we set up a loop that will first run the model in the same way as we have now… where each batch of ~40 tickers is run and the stderr value for each ticker in relation to csi1 is saved. Then, using the results, can we isolate a subset of tickers that have the lowest stderr scores and run the model again to see if we can beat the results from the previous iterations where the tickers selected were essentially random? We can use MSE as our scoring mechanism where the theory we’re trying to prove is that using only the lowest stderr tickers will result in better performance for the model.

I’m thinkig we can first test with the lowest 5 tickers per stderr scores, then 10, then 20, 30, and finally 40. Doing so we’d be able to see if:

1) Generally speaking, using tickers with the lowest stderr scores consistently leads to a model that’s more accurate than using random tickers.

2) Having more tickers is superior to less tickers (did 40 work better than 5 and 10 for example)

Finally, we can then test these results against the performance of the model using the tickers that csi1 causes (ie all the csi1\_x tickers) to see which performs better… whether that be either tickers with lowest stderr scores or ones that are determined through causality.

Also, judging by the results it looks like the variable that leads to the most accurate results is having trend set to “ctt”. From there, having ic set to “aic” or “fpe” seems to be the second most important (and doesn’t seem to matter which). Going forward, lets use these settings for consistency reasons.