I analyzed the 23-hour trading period between 2017-09-10 and 2017-10-10 within one-minute blocks for the eight suggested Eurodollars. I only predicted prices within the main trading hour i.e. 7am to 3pm; however, this can be easily generalized to do the rest of the day.

This model updates the covariance matrix through incremental sums, rather than the naïve approach of recalculating covariance at each step. The dataset we are dealing with gets quite large, so it is cumbersome to keep on having to refer to all previous data points to make this calculation.

You must let the model “start-up” for at least seven hours, or else, the covariance matrix will have weird values; in this example the startup period was 14 hours. This is because the overnight hours have sparce transaction distributions.

I checked to see, at the end of the day, if my rolling covariance matrix was accurate by comparing it with a test covariance matrix that I derived using the naïve method for computing the mean. I found that the results were satisfactory, generally within 7 decimal places. I note that the discrepancies are bigger if tested in the beginning of the day.

**Evaluation**:

Once I coded up the model, I had to figure out which etas should be set to zero. Usually only the first (the largest eta) is kept and all the other ones are set to zero. I tested to see if this is true in two ways.

The first way I did it was by calculating the ratios between the slope and standard error of the signal and forward price linear model with no constant. If the standard error was less than the slope, then there is a significant chance that the relation between the signal and future price is non-zero. I reason that the ratio of these two is a good measure of legitimacy i.e. the smaller the ratio the more legitimate. Here are a table of these ratios:



where eta1 implies that the first (i-1) etas were kept and the other etas were set to zero—with the random variable xi always kept. From an eye test, we see that eta 1 outperforms since it does well in all maturities even GEH8.

If you did a sum of these ratios across rows,



eta 1 would have the best score—I am not quite sure what this sum would mean mathematical/statistical though, but it seems intuitive.

The second way is to use correlation to evaluate a signal performance against a forward price. I have constructed a similar matrix to the one above. In this scoring scheme, a higher correlation is better.



The row-sum measure:



Again, eta ­1 seems to do the best. To address the prompt, eta 1 has a higher correlation, thus, is doing better than eta 2 for GEZ8, but eta 1 does worse for GEH9. These differences are very small, however, and both etas pass the ratio test of significance for these products.

From this analysis, I believe that eta 1, only the r.w. part, is the best transformation scheme. It has significant slopes for all maturities and generally has high correlation between forward price and signal. However, I must note that I noticed that eta 2 was the better performer during debugging when I accidentally set the forecast time between midnight and 4pm and another instance where I set it between midnight and 8am. This seems like a plausible possibility given how neck and neck eta 1 and eta 2 are.

Here is the graph of GEZ8 from eta1 to visualize the slope:

