**The research question:**

The performance of the new approach completely depends on the signal you use. It won't guarantee that you will earn good returns on any given signal. (However, I find that if the signal comes from a factor, PEP's will generally be good and if it is not a factor but the signal makes sense economically, PAP's will generally be good). So, the best approach to go forward is to use as many signals as possible. And show that if you use an array of low-quality signals and follow a standard procedure to form PP’s and get a simple equal weighted average over constructed portfolios, you can earn a good sharp ratio (using either simple factor, unconditional average historical return or null of no predictability as benchmark). This idea can be tested in currencies, US equities and other liquid equities markets, fixed-income securities, options markets and commodities.

Alternatively, I can again use as many signals as possible, form one-period-forward simple linear or long-short portfolios based on signals. Use these portfolios as test assets (universe of investable portfolios). Use momentum or any other signal available as your signal. Apply the Principal portfolios on the factor portfolios instead of single currencies. I believe this approach works better on portfolios compared to single currencies. When working with factor portfolios we can think of PP as a method to time factors.

Again, I can use many signals to compute PP’s over factor portfolios (instead of single currencies) and take simple averages.

Thoughts:

* This idea has some resonance with Kelly’s idea of virtue of complexity. He argues that using more features as inputs helps the return prediction exercise.

**Ideas for extension and improvement:**

* I can use it to investigate its performance at different frequencies. (daily, weekly, monthly). Based on evidence in page 28 of appendix, it seems this approach is works better at higher-frequencies.
* I can use some ML algo to tune the weight of each portfolio when averaging. I will probably have to work with higher frequencies for doing this.
* I can develop a python package for Principal Portfolios.
* Maybe I can improve the way, prediction matrix is calculated. Instead of using a simple average over recent periods, use a fancier method.
* I think I can innovate in building the signal. It can be non-linear.
* This idea of static bets and dynamic bets (timing) can be investigated further.
* In page 26, there is some discussion of how to implement the PP's in a machine learning manner. I can have a setting where I tune hyperparameters K and p first and use them for out-of-sample predictions.
* There is a variant of PEP in which where the expected return is negative, I take a short position. (just the negative of position matrix)
* In the original paper, there is a factor-timing exercise. I can do it for different factors in FX markets.
* Moreover, they compute the ex-post tangency portfolio to investigate how much PP's contribute to the SDF.
* I can get a better estimation of the prediction matrix. First, I have to check whether having access to the true (foresight) prediction matrix improves the results significantly.

Problems with the code:

* There is an issue with the number of eigenvectors for the asymmetric matrix. It can change in some cases and it will make the dataframe to issue an error. It cannot work with even number of assets. Sometimes it can even cause problems with odd number of assets (but rarely)

What to do next:

1. Draw a graph for comparison of equal weighted portfolios.
2. Make the code cleaner.
3. Add expanding window as an option.
4. Look for more signals in the FX market. (use signals to form portfolios) (work with portfolios instead of single currencies)
5. Expand to other markets.