Yield Curve Forecasting

Final Report

Group 24

Guillermo Marin Bull

Dhruv Chakervarti

Naveen Varma Chekuri

Aadithya Narayanan

Table of contents:

1. Introduction

Topic:

To develop a model that can explain and forecast yield curve changes across the different maturity ranges using economic and financial data. To produce forecasts of the level and the changes of the yield curve on all relevant maturity ranges (1, 2, 3, 6 months, 1, 2, 3, 5, 7, 10, 20, 30 years) for the next 1, 3, 6 months, 1 year.

Background of the project:

Forecasting movements in the term structure is of prime importance to active fixed income portfolio managers, who would attempt to time the market, and adjust the positions in their portfolio to take advantage of market events. While it may not be possible to churn their positions as frequently, due to inhibiting transaction costs and compromised benchmark tracking, the manager might be able to take positions with derivatives to take advantage of the movement and to contribute positively to the incremental performance.

The term structure, or the yield curve is composed of `on the run' Treasury bonds of varying tenors all of which may not move in tandem. However, a singular value decomposition yields three possible vectors that explain close to 99.5% of the variation.

These Principal Components can be called the level, slope and curvature factors in the yield curve (in accordance with the research done by Litterman , Scheinkman(1991) and Knez, Litterman , Scheinkman(1994)) .

We attempt to establish a dynamic and robust relationship between the three movements by using the Nelson Siegel Svensson model. Then we use several factors ranging from analyst consensus on macroeconomic expectations and lagged realizations to financial factors, as variables captured by our model. For smoothening the curve, we incorporate time variation of the coefficients overall creating a state space model for the purpose of our project.

We have done literature reviews to take inspiration from the author's ideas and to do checks in comparison to their results.

2. Literature Review

We will review 6 papers that have relevant takeaways for our project:

The first paper defines a methodology to address the problem of decomposition, specially in terms of Yield curve modelling.
The second paper on the other hand seeks to provide a dynamic Vector Autoregressive framework and fundamental AR(1) models to analyze the movements in PCA components.

The third paper inspires us to model the term structure factors through a dynamic term structure model that allows for more flexible dynamics for the evolution of these factors by modelling them through a Bayesian framework.

Fourth paper describes the effects of inflation uncertainty and premia in a Fixed Income context, which can be useful for understanding the usefulness of inflation and other macroeconomic variables when trying to explain the underlying factors that explain the movements of the UST yield curve.

The fifth paper on analyzing the dynamic inflation premium and inflation uncertainty inherent in term structure inspires us to devise models that would augment our current framework to accommodate for the measurements in the expectations and uncertainty.

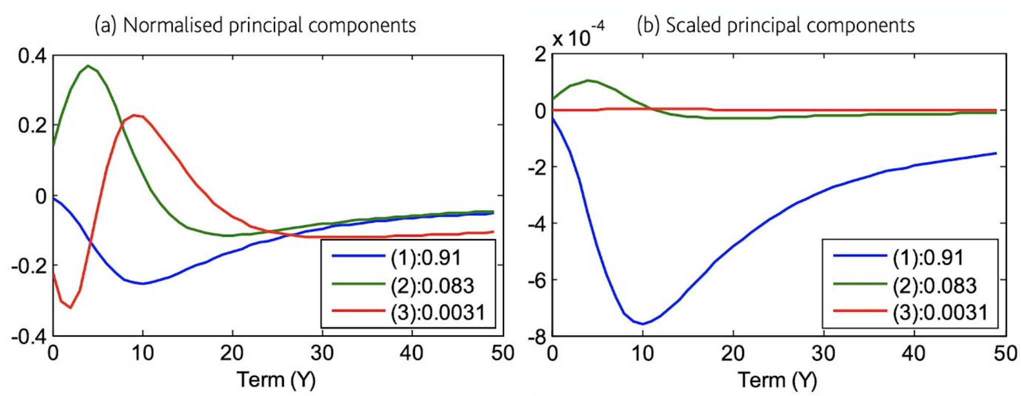
The last paper proposes a different methodology, decomposing the yield curve with a Nelson-Siegel alike model, and constructing an AR(1) over those factors.

We seek to include ideas from these papers and define processes that would enable us to make robust forecasts of the yield curve.

1) Principal Component Analysis for Yield Curve Modelling (Moodys, 2014)

The main idea in this paper is to describe and elaborate the PCA methodology into the fixed income world, and the different issues that the construction of this model can deliver.
The most important concept in PCA is to reduce a high dimensionality problem into a lower level of orthogonal factors, by using the correlation between the variables, and therefore, explaining the variance of a variable with the most important factors. This model is in fact very intuitive in yield curve context, as there are 3 factors that explain the large degree of the variance in the movement of rates: Level, slope and curvature.

But, why reduce to a simpler model by picking only a few factors instead of choosing all factors and explaining 100% of the variance (full statistical model) ?
A full statistical model would be unnecessarily flexible, as a few number of factors can explain most of the variations of the yield curve. In fact, the yield curve explained in the paper can be 99% explained by only 3 principal components.



Yet, this model has limitations. As the PCA model is constructed with historical data, it can only assemble yield curves that are similar to those that are in the sample set. Therefore, one alternative to this approach is to use a synthetically modelled yield curve dynamics, using some stochastic model to generate infinite yield curves, but it has to be calibrated properly.

One method to expand the universe of yield curves in the historic PCA approach is to include fictitious yield curves movements (like a 100 basis point parallel shift) in the context of stress testing. This would require to consider more principal components for the model to have a good fit, but would increase flexibility into a broader range of yield curve movements.

This paper introduces a dimension reduction technique that we will use in order to analyze the yield curve dynamics and the factors that explain it.

2) Forecasting the yield curve with dynamic factors (Reschenhofer & Stark, 2019)

Most models that describe the yield curve are actually useless when trying to forecast it, but dynamic factor models seem to be an exception. This paper will test the hypothesis that dynamic factor models can outperform a random walk process.

Diebold and Li used Autoregressive models into individual factors for predicting them. The paper will work through their findings and test the existence of forecasting power in the US Treasury market.
Diebold and Li modeled the yield curve with this formula (based on Nelson-Siegel approach):

$$y_t(\tau) = \beta_{1t} + \beta_{2t}\frac{1-\exp(-\lambda_t\tau)}{\lambda_t\tau} + \beta_{3t}\left(\frac{1-\exp(-\lambda_t\tau)}{\lambda_t\tau} - \exp(-\lambda_t\tau)\right) + u_t(\tau)$$

This 3 B 's can be interpreted as the 3 most relevant factors that explain the variance in the yield curve: Level (B_1) , Slope (B_2) and Curvature (B_3), and trying to forecast the yield curve is just an attempt to forecast these 3 factors.

Therefore, the authors tested different models and how their out-of-sample forecasts behaved when compared with a simple random walk pattern.
More specifically, they used 5 models to forecast the 1 and 12 month ahead yields, for the 3 months / 6months / 36 months / 120 months US Treasury bonds/bills.

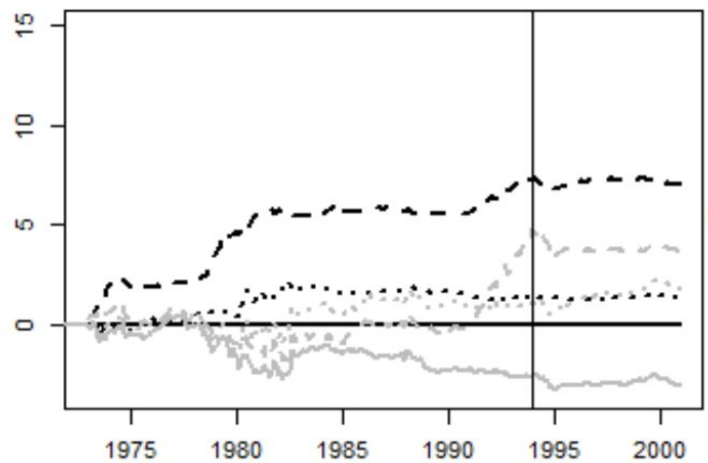
The models tested for each maturity/forecast are the following:

a) Random-walk forecast based on yields (Black, solid line)
b) AR(1) model for the yields (Black, dashed)
c) AR(1) model for the differenced yields (Black, dotted)
d) Random-walk forecast based on fitted 3-factor model (Gray, solid line)
e) AR(1) model for the factors (Gray, dashed)
f) AR(1) model for the differenced yields (Gray, dotted)

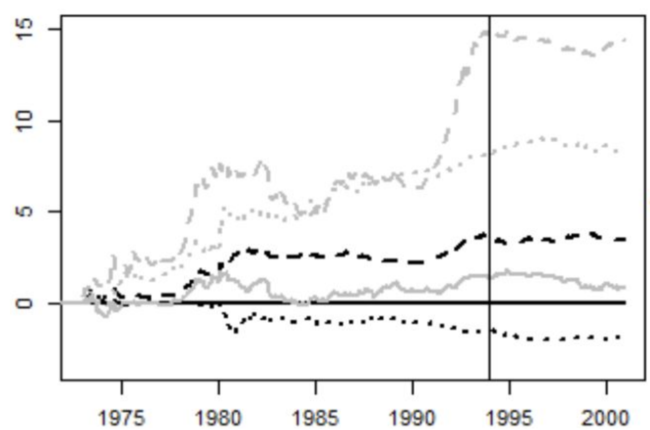
All the following graphs represent the accumulated errors when compared when compared with the Random-Walk forecast based on yields (Black, solid line)

1-Month-ahead Forecasting

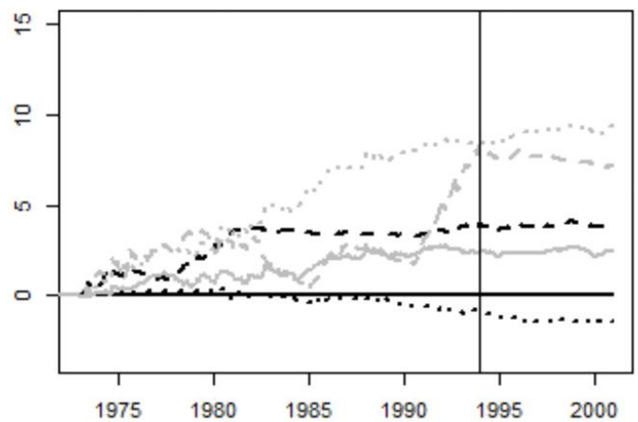
a) 3 month maturity: Random-walk forecast based on the 3 factor model had less errors when compared with Random-walk forecast based on the yields. The other models behaved poorly in this test.



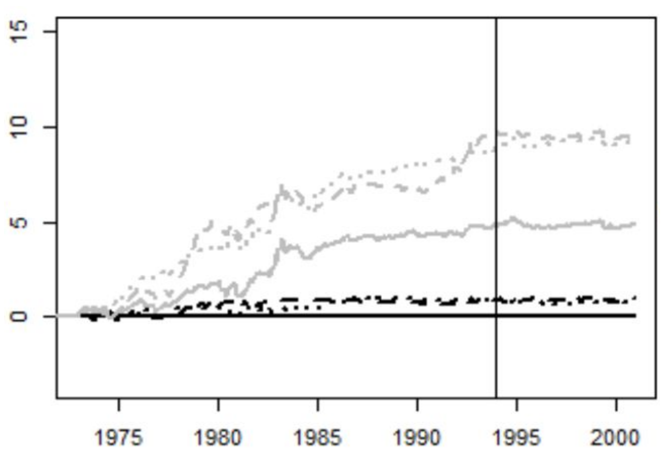
b) 6 month maturity: Forecasts based on AR(1) models for the differenced yields had a better performance than the benchmark. The other models behaved poorly in this test.



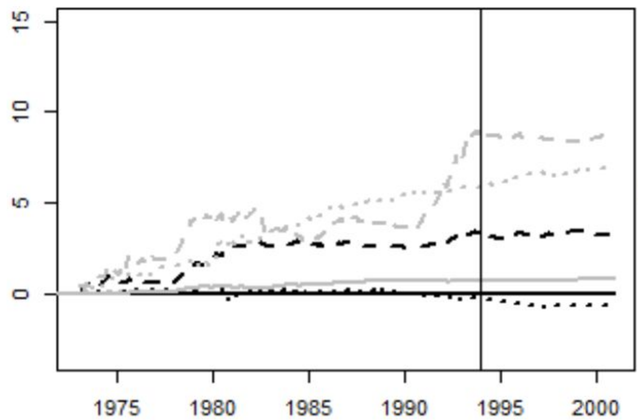
c) <u>36 month maturity:</u> Forecasts based on AR(1) models for the differenced yields had a better performance than the benchmark. The other models behaved poorly in this test.



d) <u>120 month maturity:</u> None of the models was able to perform better than the benchmark

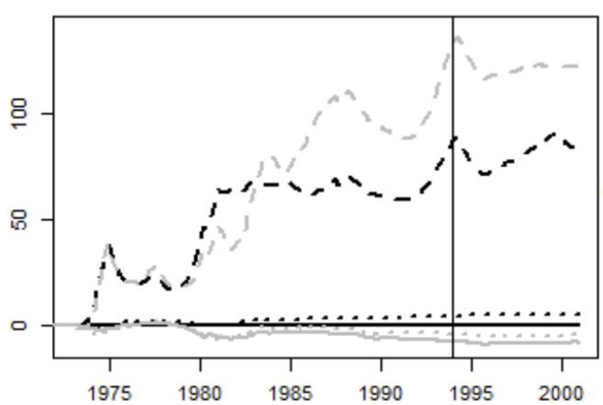


e) <u>Average performance:</u> Only Forecasts based on AR(1) models for the differenced yields had a better performance that the benchmark (on average), considering the previous 4 tests.

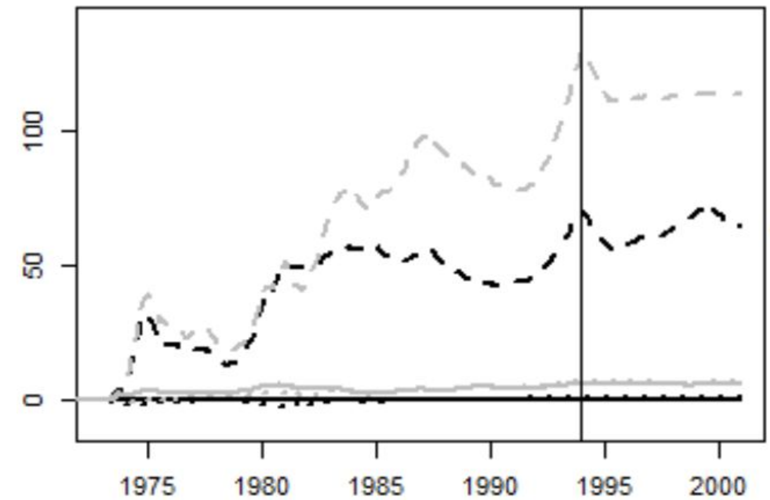


12-Month-ahead Forecasting

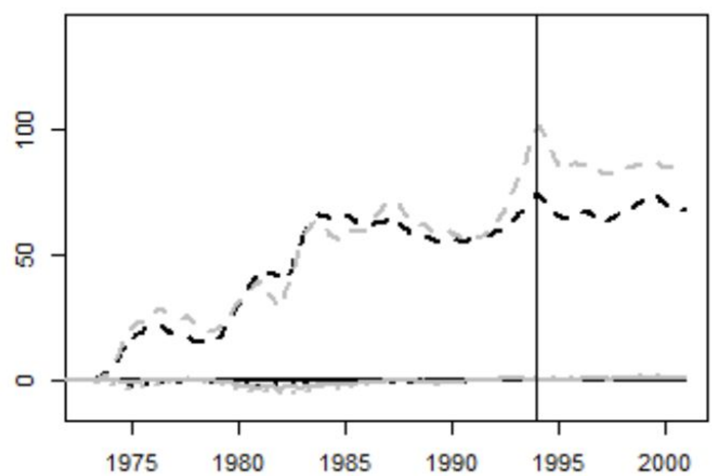
a) <u>3 month maturity:</u> Both Random-Walk based on fitted 3 factor model and AR(1) model based on factors had a better performance when compared to the benchmark.



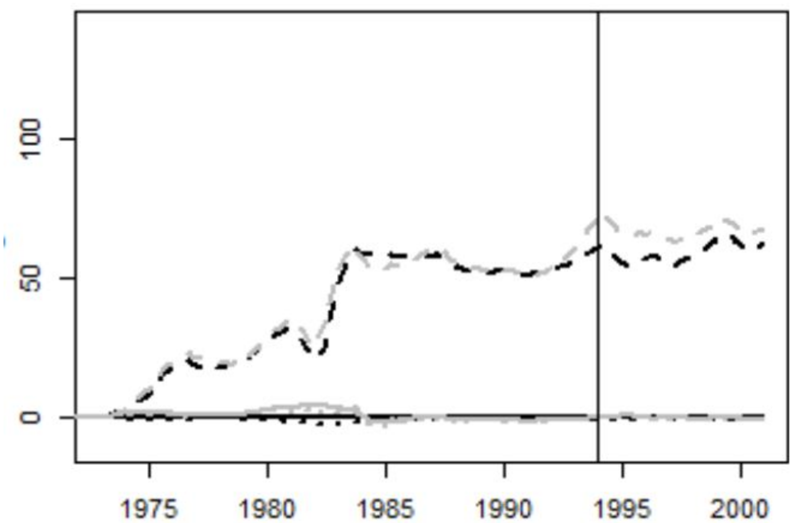
b) <u>6 month maturity:</u> No factor behaves better than the benchmark



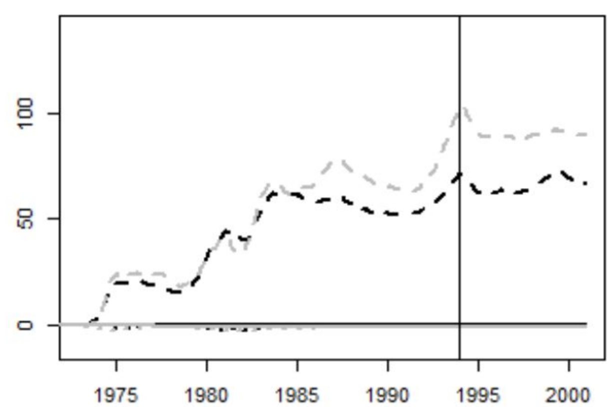
c) <u>36 month maturity:</u> No factor behaves better than the benchmark



d) <u>120 month maturity:</u> No factor behaves better than the benchmark



e) <u>Average performance:</u> No strategy delivers consistently better forecasts than the benchmark



The main finding of this paper is that previous studies about dynamic factor models that claimed to outperform random walk forecasts and other techniques, actually do not. Only AR(1) forecasts on differenced yields for the 1 month-ahead forecast have some predictive power, and factor-based forecasts are particularly bad in comparison to Random walk or yield estimates.

3) The macroeconomy and the yield curve: a dynamic latent factor approach (Francis X. Diebold, Glenn D. Rudebusch & S. Boragan Aruoba)

The paper provides a characterization of the dynamic interactions between the macroeconomy and the yield curve. The idea of the paper was to estimate a model that summarizes the yield curve using latent factors (specifically, level, slope, and curvature) and also includes observable macroeconomic variables (specifically, real activity, inflation, and the monetary policy instrument). The paper emphasises on exploring the bidirectional causality, i.e. not just the effect of macroeconomic variables on yield but also yield on macroeconomic variables. For the purpose of the research the authors maintain a no arbitrage assumption. No-arbitrage factor models often appear to fit the cross-section of yields at a particular point in time, but they do less well in describing the dynamics. The dynamic fit was crucial to the goal of relating the evolution of the yield curve over time to movements in macroeconomic variables. Moreover, the authors also emphasise that there may be a loss of efficiency in not imposing the restriction of no arbitrage if it is valid, but this must be weighed against the possibility of misspecification if transitory arbitrage opportunities are not eliminated immediately/due to illiquidity in thinly traded regions of the yield curve.

The authors achieve their objective by following three steps:

- In this section, we introduce a factor model of the yield curve without macroeconomic variables, which is useful for two key reasons:
  - First, methodologically, such a model proves to be a convenient vehicle for introducing the state-space framework that we use throughout the paper.
  - Second, and substantively, the estimated yields-only model serves as a useful benchmark to which we subsequently compare our full model that incorporates macroeconomic variables.

  To estimate this model, the paper introduces a unified state-space modeling approach that lets us simultaneously fit the yield curve at each point in time and estimate the underlying dynamics of the factors. This one-step approach improves upon the two-step estimation procedure of Diebold and Li (2002) and provides a unified framework in which to examine the yield curve and the macroeconomy.

  Factor Model representation/Nelson–Siegel representation:

$$y(\tau) = \beta_1 + \beta_2\left(\frac{1-e^{-\lambda\tau}}{\lambda\tau}\right) + \beta_3\left(\frac{1-e^{-\lambda\tau}}{\lambda\tau} - e^{-\lambda\tau}\right)$$

  β1; β2; and β3 are time-varying level, slope, and curvature factors and the terms that multiply these factors are factor loadings

  Replacing the coefficients we get:

$$y_t(\tau) = L_t + S_t\left(\frac{1-e^{-\lambda\tau}}{\lambda\tau}\right) + C_t\left(\frac{1-e^{-\lambda\tau}}{\lambda\tau} - e^{-\lambda\tau}\right)$$

where Lt; St; and Ct are the time-varying β1; β2; and β3

VAR(1) process/Unified state-space system:

$$\begin{pmatrix} y_t(\tau_1) \\ y_t(\tau_2) \\ \vdots \\ y_t(\tau_N) \end{pmatrix} = \begin{pmatrix} 1 & \frac{1-e^{-\tau_1\lambda}}{\tau_1\lambda} & \frac{1-e^{-\tau_1\lambda}}{\tau_1\lambda} - e^{-\tau_1\lambda} \\ 1 & \frac{1-e^{-\tau_2\lambda}}{\tau_2\lambda} & \frac{1-e^{-\tau_2\lambda}}{\tau_2\lambda} - e^{-\tau_2\lambda} \\ \vdots & \vdots & \vdots \\ 1 & \frac{1-e^{-\tau_N\lambda}}{\tau_N\lambda} & \frac{1-e^{-\tau_N\lambda}}{\tau_N\lambda} - e^{-\tau_N\lambda} \end{pmatrix} \begin{pmatrix} L_t \\ S_t \\ C_t \end{pmatrix} + \begin{pmatrix} \varepsilon_t(\tau_1) \\ \varepsilon_t(\tau_2) \\ \vdots \\ \varepsilon_t(\tau_N) \end{pmatrix}$$

Which can be simplified as:

$$y_t = \Lambda f_t + \varepsilon_t.$$

If the dynamic movements of Lt; St; and Ct follow a vector autoregressive process of first order, then the model immediately forms a state-space system.
Hence, Lt; St; and Ct can be derived by:

$$\begin{pmatrix} L_t - \mu_L \\ S_t - \mu_S \\ C_t - \mu_C \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \begin{pmatrix} L_{t-1} - \mu_L \\ S_{t-1} - \mu_S \\ C_{t-1} - \mu_C \end{pmatrix} + \begin{pmatrix} \eta_t(L) \\ \eta_t(S) \\ \eta_t(C) \end{pmatrix}$$

Which can be simplified as:

$$(f_t - \mu) = A(f_{t-1} - \mu) + \eta_t.$$

For linear least-squares optimality of the Kalman filter, the paper considers that the white noise transition and measurement disturbances be orthogonal to each other and to the initial state:

$$\begin{pmatrix} \eta_t \\ \varepsilon_t \end{pmatrix} \sim WN\left[\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} Q & 0 \\ 0 & H \end{pmatrix}\right]$$

The assumption of a diagonal H matrix, which implies that the deviations of yields of various maturities from the yield curve are uncorrelated, an unrestricted Q matrix, which is potentially non diagonal, allows the shocks to the three term structure factors to be correlated.

- In this step, the paper incorporates macroeconomic variables and estimates a ‘‘yieldsmacro’’ model. To complement the nonstructural nature of the yield curve representation, the paper forms a simple nonstructural VAR representation of the macroeconomy. The focus of the examination was the nature of the linkages between the factors driving the yield curve and macroeconomic fundamentals.

The paper in this step introduces three key macroeconomic variables to the state space model: manufacturing capacity, utilization (CUt); the federal funds rate (FFRt); and annual price inflation (INFLt). The paper does not have any analytical criteria on selecting these parameters but the authors state that the parameters are the minimum set of fundamentals needed to capture basic macroeconomic dynamics. The model now is:

$$(f_t - \mu) = A(f_{t-1} - \mu) + \eta_t,$$

$$y_t = \Lambda f_t + \varepsilon_t,$$

$$\begin{pmatrix} \eta_t \\ \varepsilon_t \end{pmatrix} \sim WN\left[\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} Q & 0 \\ 0 & H \end{pmatrix}\right],$$

where $f'_t = (L_t, S_t, C_t, CU_t, FFR_t, INFL_t)$

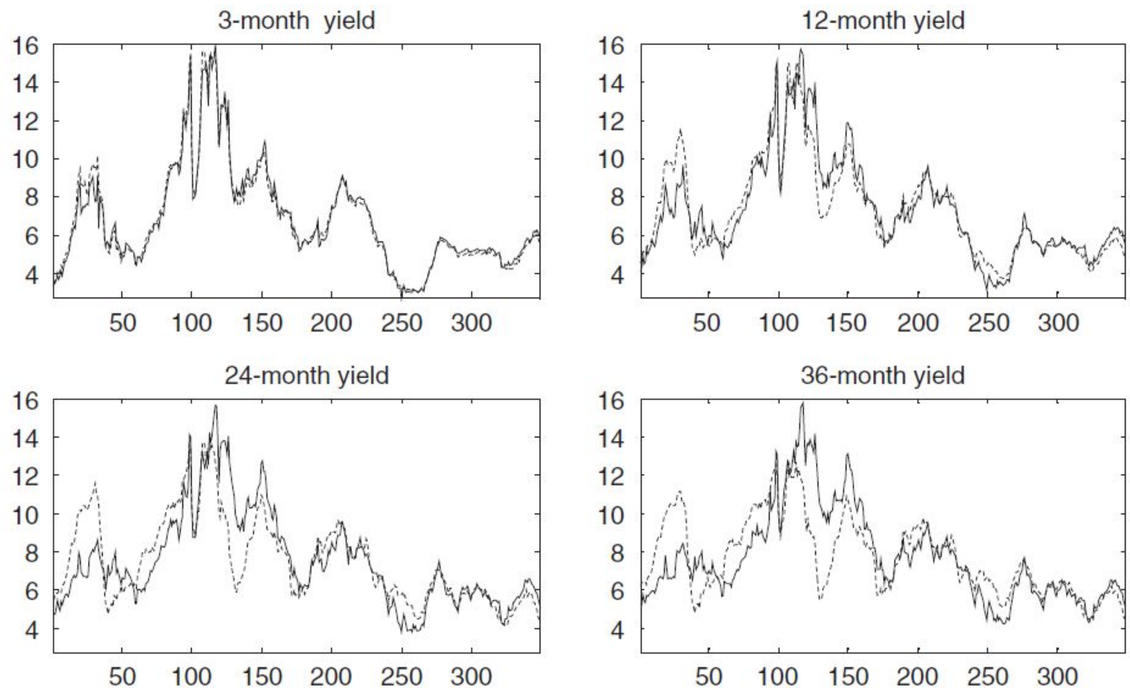
- In this section the paper relates the framework to the expectations hypothesis, which has been studied intensively in macroeconomics. The expectation hypotheses of the term structure states that movements in long rates are due to movements in expected future short rates. Any term of risk premia is constant. Theoretical bond yield consistent with the expectations hypothesis :

$$y_t(\tau)^{EH} \equiv (1/\tau) \sum_{i=0}^{\tau-1} \mathrm{E}_t y_{t+i}(1) + c_\tau$$

where c is a term premium that may vary with the maturity t but assumed to be constant through time.

The underlying assumption of the research is that the expectations hypothesis holds with the actual bond yields. For testing this theory, the paper constructs the expected future 1-month yields by iterating forward the estimated yields-macro model to get yt(1) and then compute the theoretical bond yields at each point in time using the above equation.

The results show that there are large deviations between the theoretical and actual yields, especially at longer maturities.



The author concludes the paper by sharing the following observations of the results.
The paper points out that:

- There is strong evidence of macroeconomic effects on the future yield curve and somewhat weaker evidence of yield curve effects on future macroeconomic developments. However, Market yields contain important predictive information about the Fed's policy rate.
- The expectations hypothesis may hold reasonably well during certain periods, but that it does not hold across the entire sample.
- The state space framework is more efficient than a factor based approach.

4)The term structure and Inflation uncertainty (Breach, D'amico, Orphanides)

This paper develops and estimates a quadratic gaussian model of the US term structure that can accommodate rich dynamics inflation risk premia by time varying market prices of inflation risk and incorporating survey information on inflation uncertainty in the estimation of our model.

While acknowledging the dynamic nature of inflation risk premia, the paper attributes this variation to the changes in the levels of actual and perceived inflation uncertainty. We briefly describe the building blocks of the quadratic gaussian model first, and subsequently derive the dynamics of the inflation risk premia by taking into account the certain and uncertain dynamics of inflation.
Given z_t are factors that are hidden in the nominal yield curve.

$$\begin{bmatrix} dx_t \\ dz_t \end{bmatrix} = \begin{bmatrix} \kappa^x_{2\times 2} & 0_{2\times 1} \\ 0_{1\times 2} & \kappa^z_{1\times 1} \end{bmatrix} \begin{bmatrix} (\mu_x - x_t)_{2\times 1} \\ (\mu_z - z_t)_{1\times 1} \end{bmatrix} dt + \begin{bmatrix} \Sigma^x_{2\times 2} & 0_{2\times 1} \\ \sigma_{x,z} & \sigma_z \end{bmatrix} \begin{bmatrix} dB^x_t \\ dB^z_t \end{bmatrix}$$

The nominal pricing kernel under the physical measure is given by:

$$\frac{dM^N_t}{M^N_t} = -r^N_t dt - \lambda^{N\prime}_t dB^x_t$$

Where the nominal short rate is affine in the state variables x_t as defined above. Please note that the Brownian shocks may be correlated as previously defined.

Additionally, the dynamics of evolution of market's price of risk (lambda) is also affined in x_t. The log price follows the dynamics. Please note that the prices are driven by an additional orthogonal brownian motion with a standard deviation of omega. Please note that the brownian motion associated with

$$d\log Q_t = \pi_t dt + \lambda^{q\prime}_t dB^{x^*}_t + \omega_t dB^{\perp}_t$$

The pi_t in the above equation is again affine in xt. The lambda and omega terms that are essentially the conditional volatility are affine in x_t.

The bond pricing and equations defining the inflation expectations and uncertainty are defined appropriately. The inflation risk premium is defined by:
Nominal Short rate- Real Short rate-Expected inflation rate.

State Space formulation of our problem and the data:

An augmented extended Kalman FIlter is used for the estimation of the equations and state matrix is defined as

$$s_t = [x_t^*, vech(x_t^* x_t^{*\prime}), q_t]'$$

Observed equations, as the name says is what we observe: in the paper, the observation variables are: nominal yields of all maturities under consideration and the forecasts of short rates for periods 6 months, 12 months and a longer term forecast, survey of inflation expectations and a measure of inflation uncertainty.

The evolution of the state matrix and the observation equations are:

$$s_t = G_h + \Gamma_h s_{t-h} + \eta_t^s, \qquad o_t = a + F s_t + \varepsilon_t$$

The seven measurement equations above relate the treasury yields to the two state variables x_t. Specifically, we use the 3- and 6-month Treasury bill rates from the Federal Reserve Board's H.15 release and convert them to continuously compounded basis. The other yields of a longer maturity curve is fitted using the zero coupon using the methodology identical to that of the Federal Reserve Board. The yields are sampled at weekly frequency , and CPI data is included at a monthly frequency (CPI is released at the last day of the month).

Results:

A visual description of the main findings is presented in Figures 1, 2, and 3. Specifically, Figure 1 shows the decomposition of the 10-year nominal yield into three components: The real yield (including the RRP), the expected inflation at the pertinent horizon, and the corresponding IRP. Figure 2 focuses on the four components of the 10-year nominal yield, as in addition to the expected inflation rate and IRP (also shown in Figure 1), it shows the expected future short real rate and the RRP separately. Finally, Figure 3 summarizes the overall fit of the full model, as it compares the model-implied one-year inflation variance, 5-year real yield, one-year expected inflation, and one-year expectation of the nominal short-term rate to their counterparts in the data (shown in orange).

Figure 1

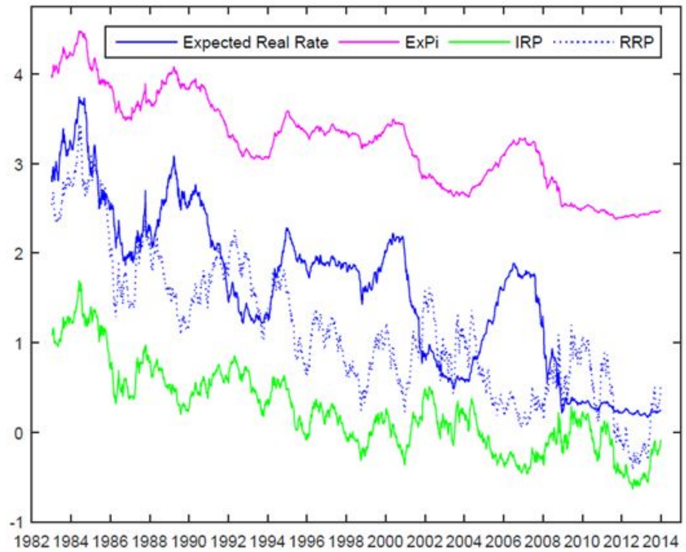
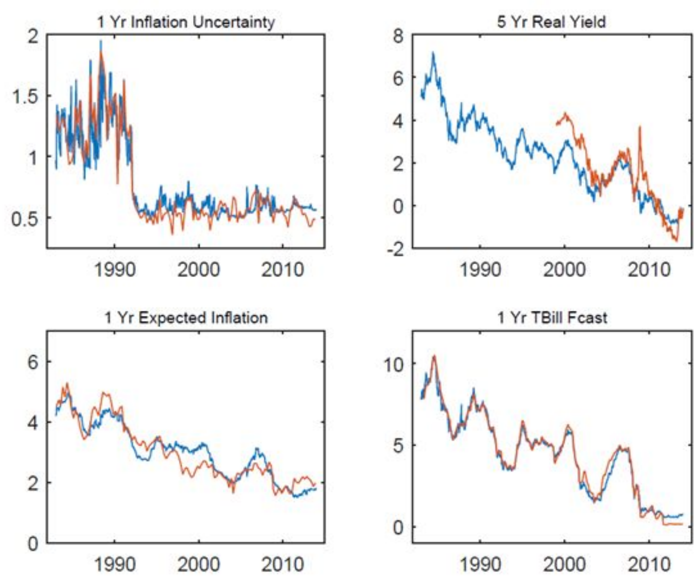


Figure 2



5) The Fed and Interest Rates - A High Frequency Identification

This paper studies monetary-policy shocks to the interest rates, as a result of federal funds target movements.

The interest rate response to the unexpected federal rate changes is different from the response to target changes. The long term rates seems to be more affected by the unexpected federal funds target changes and many estimates suggest that.

The interest rate shocks are studied for a period which is 2 days prior to and one day after the change.

Below table 1 shows the interest changes across maturities. We see that the interest rate changes are more when the target rates are unexpected (as evident from the part B vs part A)

	Euro		Treasury			
Statistic	1 mo	3 mo	3 mo	1 yr	3 yr	10 yr
A. *Yield Change on Target Change:*						
b	0.52	0.46	0.37	0.37	0.29	0.19
t	9.1	8.0	7.6	6.7	5.0	3.5
R^2	0.39	0.32	0.28	0.24	0.15	0.08
B. *Yield Change on One-Month Euro Change:*						
b		0.91	0.62	0.72	0.67	0.52
t		22.1	6.8	10.8	8.8	5.2
R^2		0.87	0.54	0.63	0.55	0.39

TABLE 1:

			Interest Rates			
Statistic	Constant	Target	3 mo	2 yr	5 yr	10 yr
b	0.29	−0.037	0.12	0.87	−0.87	0.24
t	3.1	−3.7	2.0	6.7	−3.5	1.57

TABLE 2:

Table 2 shows that the long-term rates are far more important than the short-term rates in forecasting Fed moves. On regressing the one month yields with the federal target changes, the coefficients turn out to be 0.06 with a t stat of 0.8, where are insignificant compared to the longer maturities.

This finding is not in line with the general expectation that the short term interest changes will forecast target rate changes. From the results it is evident that the Fed responds to expected inflation information embedded in the long term rates, as such the long term interest are more affected by the target rate changes.

Finally the paper concludes that the Fed responds to long term interest rates, as they carry the inflation information and to the slope of the term structure which represents real economic activity. Short term interest rates do not help to forecast target changes.

6) Forecasting the Term Structure of Government Bond Yields (Diebolt, Li)

In this paper, the authors argue that using a similar approach as Nelson-Siegel (1987), the parameters obtained by their exponential function can be interpreted as the 3 predominant factors of the yield curve: Level, Slope and Curvature.

$$y_t(\tau) = \beta_{1t} + \beta_{2t}\left(\frac{1-e^{-\lambda_t\tau}}{\lambda_t\tau}\right) + \beta_{3t}\left(\frac{1-e^{-\lambda_t\tau}}{\lambda_t\tau} - e^{-\lambda_t\tau}\right) .$$

In this parametric formula, *tau* stands for the time to maturity of each point in the yield curve, and *lambda* is the exponential decay of each factor. This *lambda* is fixed into $\lambda = 0.0609$, as the author claims that this value is the one that maximizes the curvature factor in 30 months (a commonly used assumption).

Authors propose an autoregressive model (1) for those factors, in order to estimate them. For accomplishing so, they apply Ordinary Least Squares to the yield for each month (and therefore estimate the Betas).
The obtained Time Series of Betas are also tested for unit root, using the Dickey Fuller Test. This test concluded that $\beta 1$, $\beta 2$ may have unit roots, but $\beta 3$ does not.

They also construct a VAR forecast, but they argue that it should be inferior to the AR1 model, as they are prone to overfitting.

They finally conclude that the 1month forecasts are not better than a Random Walk, but the 1 year results are significatively superior.

3. Data Description

We will use the bloomberg api and write search queries that would pull the data from the different sources listed below.

1. Yields on the on the run treasury notes, can be obtained from the FED website
2. Sell side analyst consensus on expected economic outcomes (Bloomberg) - ticker: CESI USD
3. Inflation, USD currency liquidity and Unemployment data from Bloomberg

4. Methodology

Our first step will be to construct the Principal component factors (PCA) over the American yield curve, in order to understand in a more concise way what are the drivers of it and their relevance. By decomposing the yield curve into orthogonal components (that according to literature, capture nearly 99.5% of the variation), we significantly reduce the dimensionality of if, and therefore any analysis that wants to be pursued becomes more simpler.

If the literature review concludes that only 3 PCA are enough to explain almost all the variation still is valid (as it should be), we may go forward and continue with the Diebolt & Li paper implementation.
As explained in the literature review, Diebolt, Li (2003) proposes that if the PCA are constructed with a non-orthogonal methodology, dependence between factors and lagged values of them can be exploited into predicting future values of themselves.

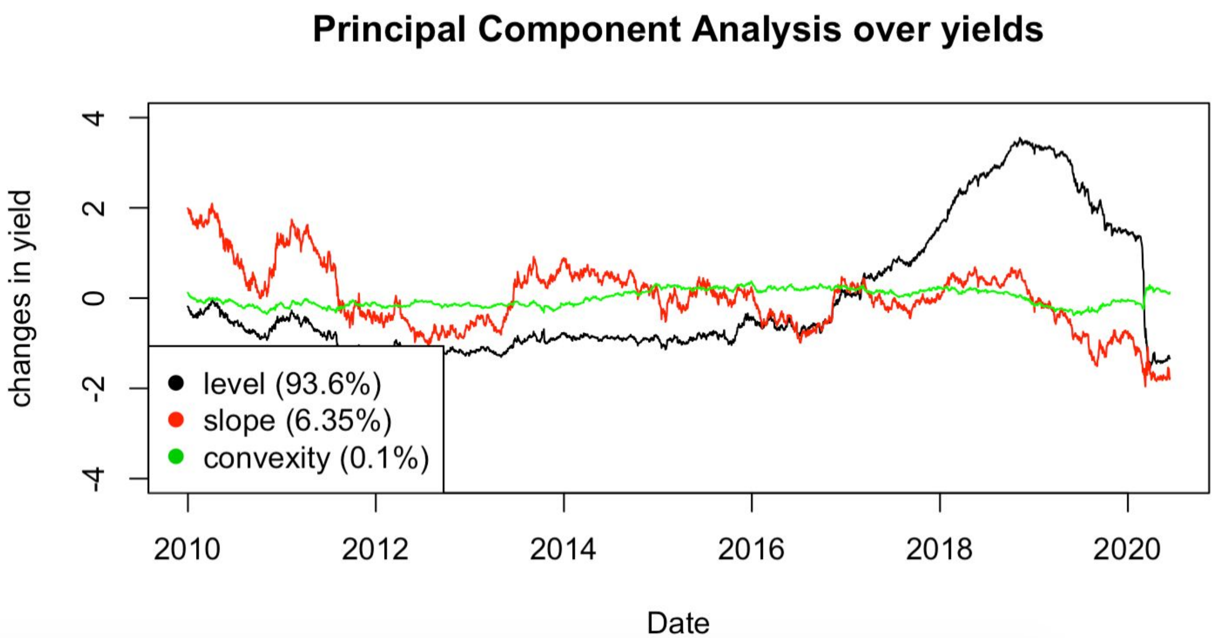
After constructing these factors in a non-orthogonal way, we need to check if there is dependence between the factors, with a Granger test. If this test has significant results, we can move forward into constructing a VAR model, where we will construct forecasts over each of the factors with the help of lagged values not only of the factor itself but also, from the other factors.

At a later stage, an empirical performance of the VAR model will be realized over 1 month , 3 month, 6 month and 12 month forecasts, to check the predictive power of the model over those timeframes.

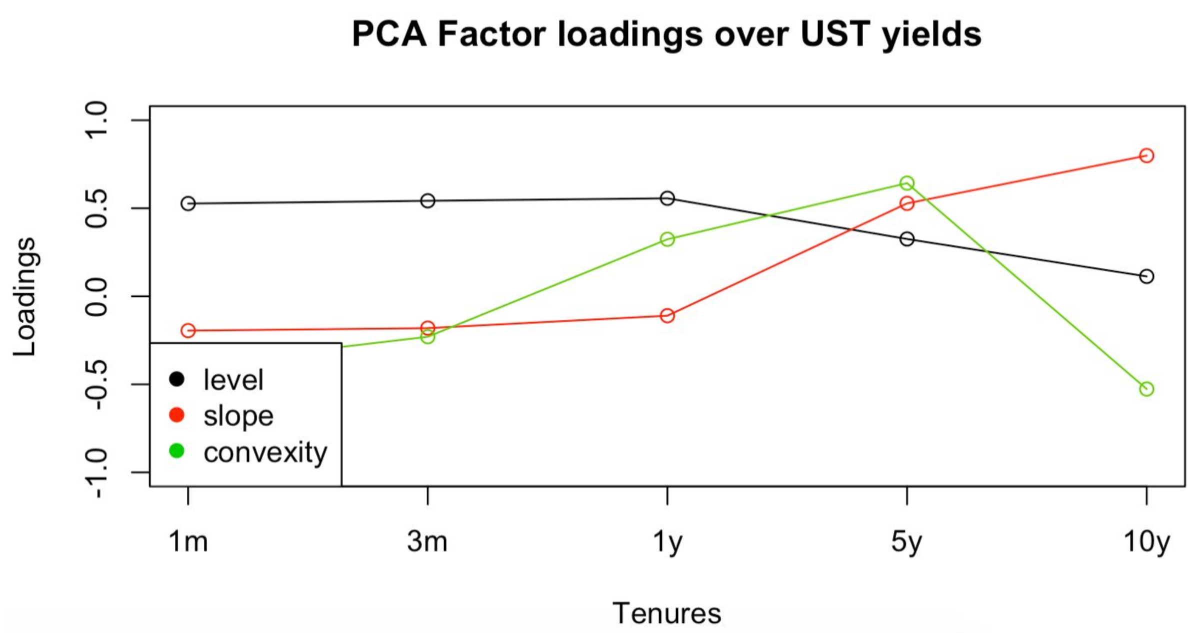
5. Preliminary Results

As previously stated, our first step would be to construct 2 Principal Component Analysis models over 1 month / 3months / 1year / 5 year / 10 year tenures; one PCA model over the yields and one PCA model over the change in yields.

a) PCA over yields: The yield curve was decomposed and 3 most relevant PC's explain 99.9% of the variation. This PC's can be related to Level, Slope and Curvature

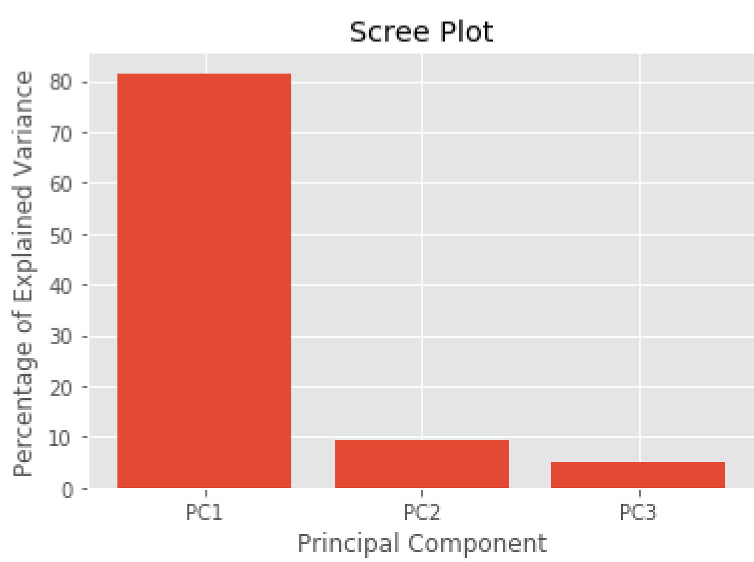


	PC1 (level)	PC2 (slope)	PC3 (curvature)	PC4	PC5
1m	0.5266783	-0.1949882	-0.3898944	-0.46352405	0.56366422
3m	0.5423184	-0.1807643	-0.2296361	-0.06881344	-0.78469547
1y	0.5565929	-0.1101572	0.3238013	0.71369732	0.25270264
5y	0.3254928	0.5277132	0.6425968	-0.44786449	-0.04538736
10y	0.1129580	0.7991796	-0.5267559	0.26544999	0.02483985

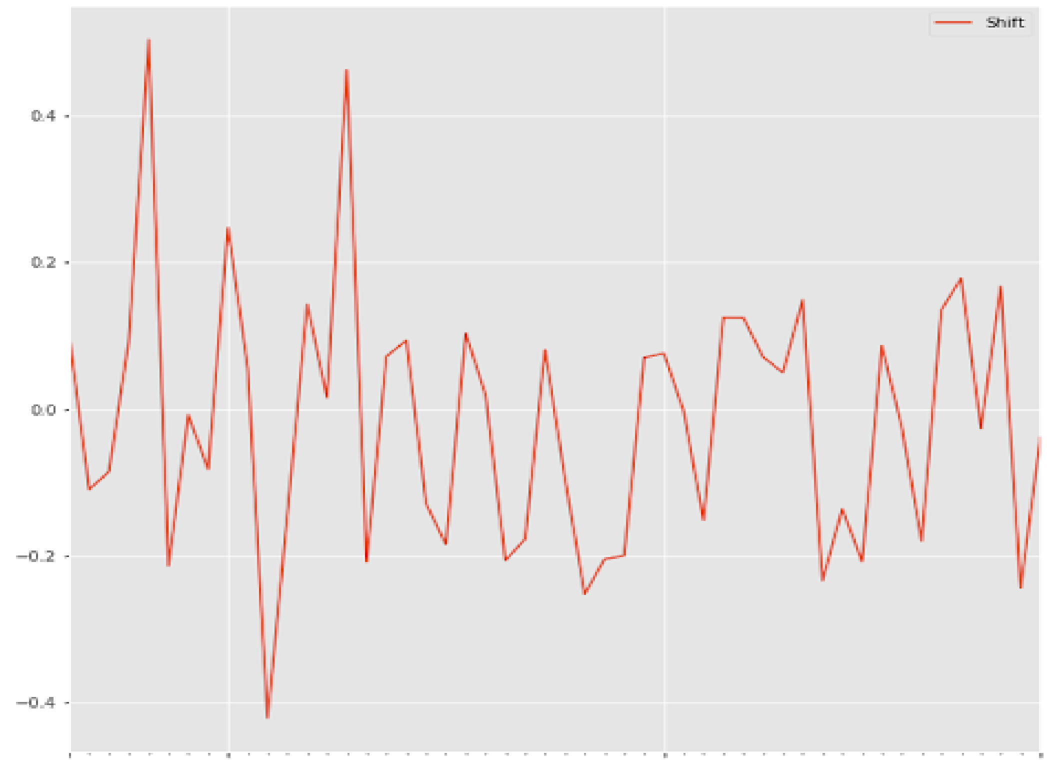


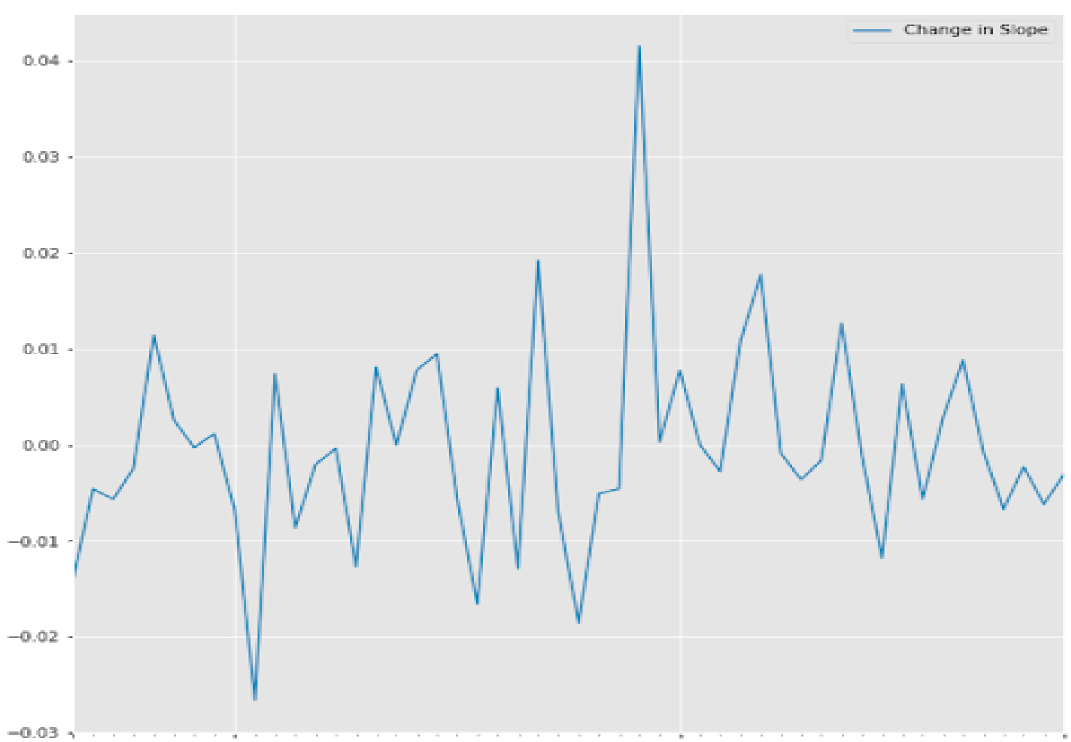
A Box-Pierce test was performed over each series of decomposed yields to check the orthogonality of them. They all proved to be significantly orthogonal.

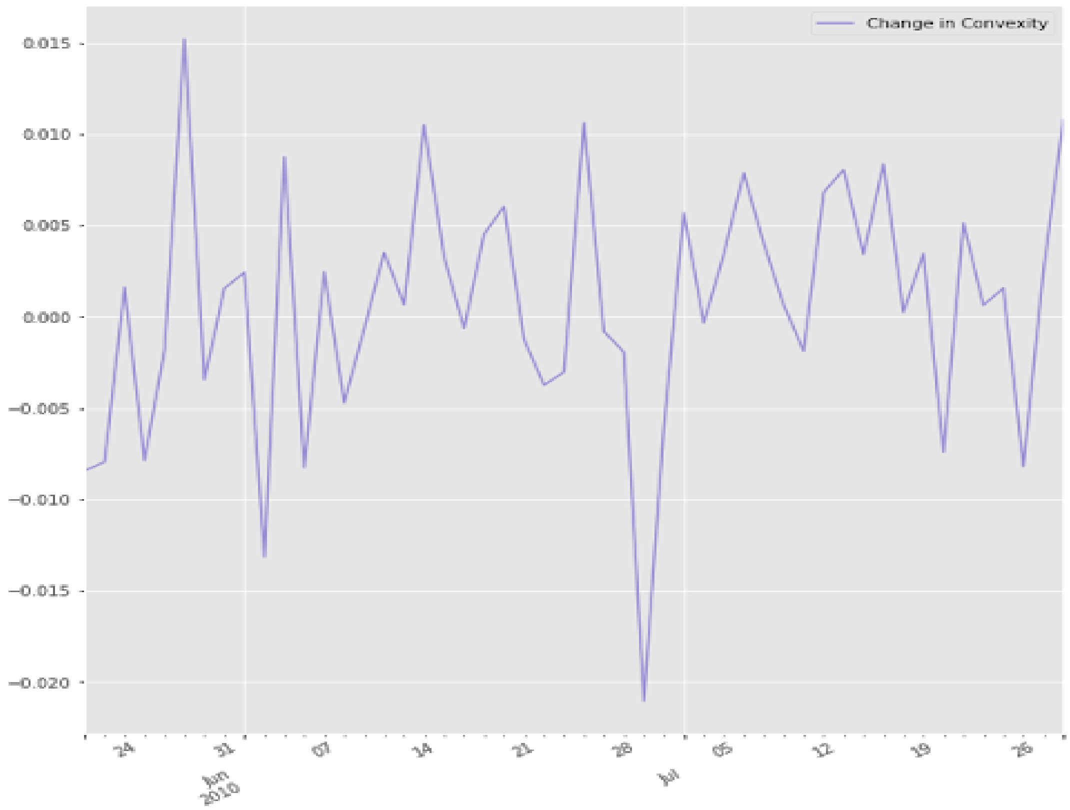
b) PCA over differenced yields:. We do this because we believe the forecasting power of the changes in yield would be more than just the yield level. The results of this analysis were three principal components which were the change in level (shift), change in slope and the change in the curvature.



Percentage explained by Principal Components: array([81.5, 9.5, 5.2])







In the absence of explanatory variables, we examine the significance of autocorrelation evident in each of the principal components. In addition for this analysis, we perform the Box-Pierce test results of which are as follows:

Box - Pierce Test:

df = 5	X Squared	P value
PCA 1 - Level	2720.3	1.98E-03
PCA 2 - Slope	2689.3	2.20E-16
PCA 3 - Curvature	2700.9	6.21E-06

c) Instead of taking daily data into consideration, to match the timeline of the macroeconomic data, we take weekly yields into consideration in our model. We extract the orthogonal components from the weekly yields by taking a PCA. We then take the differences in the PCAs for further analysis, the results are as follows:

	L_diff	S_diff	C_diff
0	0.082341	-0.003128	-0.064619
1	-0.280592	0.074014	0.011800
2	0.105177	-0.096673	-0.019089
3	-0.042912	-0.293664	-0.094564
4	0.036672	-0.153342	-0.146871
...	...	...	...
1032	-0.026414	0.010531	-0.010687
1033	0.019205	0.012490	0.044325
1034	-0.050934	0.035111	0.006148
1035	0.020706	-0.004415	-0.003037
1036	0.022915	-0.024811	-0.015700

1037 rows × 3 columns

Even though theoretically the orthogonality of the factors should disregard a VAR model (since there is no dependability), for the purpose of completion we will still construct the VAR model and check on its results.

OLS Regression Results

Dep. Variable:	L_diff	**R-squared (uncentered):**	0.013
Model:	OLS	**Adj. R-squared (uncentered):**	0.010
Method:	Least Squares	**F-statistic:**	4.510
Date:	Fri, 16 Oct 2020	**Prob (F-statistic):**	0.00376
Time:	22:19:13	**Log-Likelihood:**	189.43
No. Observations:	1036	**AIC:**	-372.9
Df Residuals:	1033	**BIC:**	-358.0
Df Model:	3		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
L_diff	0.0707	0.040	1.770	0.077	-0.008	0.149
S_diff	0.1317	0.064	2.071	0.039	0.007	0.257
C_diff	-0.3284	0.116	-2.823	0.005	-0.557	-0.100

Omnibus:	77.467	**Durbin-Watson:**	2.013
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	346.424
Skew:	0.151	**Prob(JB):**	5.96e-76
Kurtosis:	5.817	**Cond. No.**	4.08

OLS Regression Results

Dep. Variable:	S_diff	**R-squared (uncentered):**	0.004
Model:	OLS	**Adj. R-squared (uncentered):**	0.001
Method:	Least Squares	**F-statistic:**	1.365
Date:	Fri, 16 Oct 2020	**Prob (F-statistic):**	0.252
Time:	22:19:14	**Log-Likelihood:**	736.81
No. Observations:	1036	**AIC:**	-1468.
Df Residuals:	1033	**BIC:**	-1453.
Df Model:	3		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
L_diff	-0.0218	0.024	-0.927	0.354	-0.068	0.024
S_diff	-0.0757	0.037	-2.018	0.044	-0.149	-0.002
C_diff	-0.0052	0.069	-0.076	0.940	-0.140	0.129

Omnibus:	93.495	**Durbin-Watson:**	1.995
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	307.844
Skew:	0.409	**Prob(JB):**	1.42e-67
Kurtosis:	5.542	**Cond. No.**	4.08

OLS Regression Results

Dep. Variable:	C_diff	**R-squared (uncentered):**	0.012
Model:	OLS	**Adj. R-squared (uncentered):**	0.009
Method:	Least Squares	**F-statistic:**	4.301
Date:	Fri, 16 Oct 2020	**Prob (F-statistic):**	0.00502
Time:	22:19:14	**Log-Likelihood:**	1436.8
No. Observations:	1036	**AIC:**	-2868.
Df Residuals:	1033	**BIC:**	-2853.
Df Model:	3		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
L_diff	0.0353	0.012	2.945	0.003	0.012	0.059
S_diff	0.0139	0.019	0.727	0.467	-0.024	0.051
C_diff	-0.1038	0.035	-2.973	0.003	-0.172	-0.035

Omnibus:	75.561	**Durbin-Watson:**	2.012
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	178.042
Skew:	-0.420	**Prob(JB):**	2.18e-39
Kurtosis:	4.849	**Cond. No.**	4.08

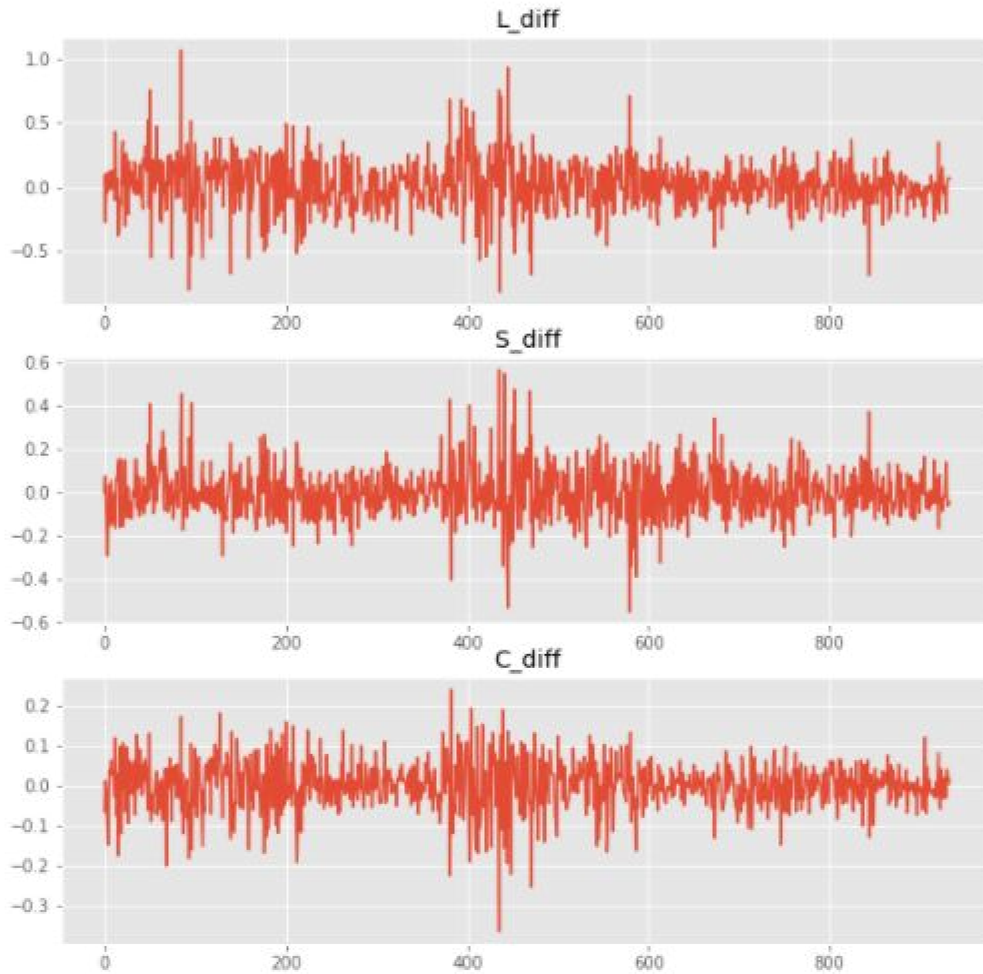
The results for the VAR model are as follows:

```
  Summary of Regression Results
==================================
Model:                         VAR
Method:                        OLS
Date:           Fri, 16, Oct, 2020
Time:                     23:13:28
--------------------------------------------------------------------
No. of Equations:         3.00000    BIC:                   -13.5157
Nobs:                     932.000    HQIC:                  -13.5543
Log likelihood:           2372.00    FPE:                1.26808e-06
AIC:                     -13.5780    Det(Omega_mle):     1.25189e-06
--------------------------------------------------------------------
Results for equation L_diff
============================================================================
              coefficient       std. error           t-stat            prob
----------------------------------------------------------------------------
const            0.009122         0.006709            1.360           0.174
L1.L_diff       -0.099252         0.042787           -2.320           0.020
L1.S_diff       -0.042692         0.067348           -0.634           0.526
L1.C_diff        0.325689         0.121560            2.679           0.007
============================================================================

Results for equation S_diff
============================================================================
              coefficient       std. error           t-stat            prob
----------------------------------------------------------------------------
const           -0.002086         0.003949           -0.528           0.597
L1.L_diff        0.041527         0.025183            1.649           0.099
L1.S_diff       -0.021809         0.039639           -0.550           0.582
L1.C_diff       -0.085454         0.071545           -1.194           0.232
============================================================================

Results for equation C_diff
============================================================================
              coefficient       std. error           t-stat            prob
----------------------------------------------------------------------------
const           -0.000004         0.002058           -0.002           0.999
L1.L_diff       -0.027443         0.013124           -2.091           0.037
L1.S_diff       -0.026154         0.020657           -1.266           0.205
L1.C_diff       -0.034018         0.037285           -0.912           0.362
============================================================================

Correlation matrix of residuals
            L_diff    S_diff    C_diff
L_diff    1.000000 -0.560602  0.482754
S_diff   -0.560602  1.000000 -0.312741
C_diff    0.482754 -0.312741  1.000000
```



However, as predicted, we encountered significant dispersions in our predictions for in-sample tests and so we had to search for alternative more efficient approaches for our project.

Final approach for the project:

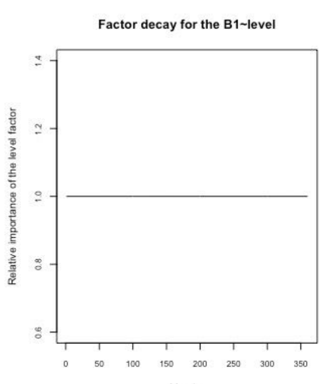
We also implemented the idea stated by Diebold,Li (2003) regarding the computation of a parametric function (inspired in Nelson-Siegel work) and further interpretation of the 3 Betas obtained as Level, Slope and Curvature.
For doing so, we used the same database we used before regarding the UST yields in between Jan-2000 and Sept-2020.
This methodology will create dependent factors (as opposed to PCA), which can help into predicting rates with a VAR model.

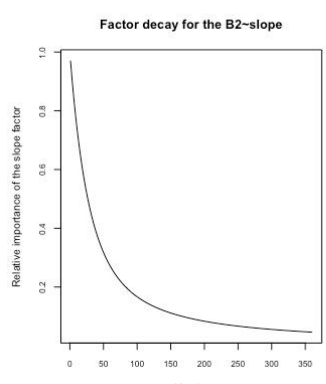
We filtered the dataset and made our calculations using weekly data. Once we accomplished this, we estimated **β**1, **β**2, **β**3 and **β**4 for each week, using OLS approach and the parametric function proposed by the paper.

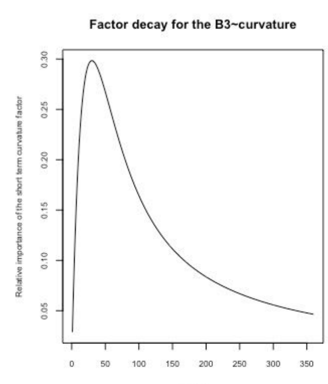
$$St = B_1 + B_2 \frac{(1 - e^{-\lambda_1 t})}{\lambda_1 t} + B_3 \left[\frac{(1 - e^{-\lambda_1 t})}{\lambda_1 t} - e^{-\lambda_1 t} \right] + B_4 \left[\frac{(1 - e^{-\lambda_2 t})}{\lambda_2 t} - e^{-\lambda_2 t} \right]$$

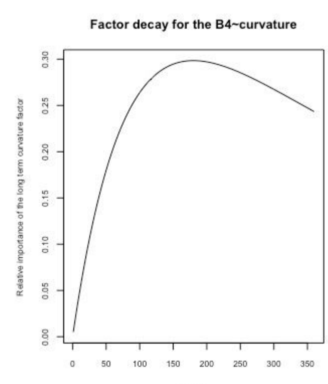
The intuition of this parameters is simple:

β1 is the level, as its flat and non-dependent from the time. Therefore its value across all the yield curve is constant
β2 is the slope , as its behavior resembles the difference between the long term of the yield curve minus the beginning of it (360 months vs 3 months).
β3 is the short term curvature over the yield curve. The value of λ_1 is 0.059776 in order to create a hump in the month 30, as Diebolt,Li suggested.
β4 is the long term curvature over the yield curve. The value of λ_2 is 0.0099626 in order to create a hump in the month 200. This idea was raised by our team to account for the long term hump that's seen on the yield curves and usually is disregarded



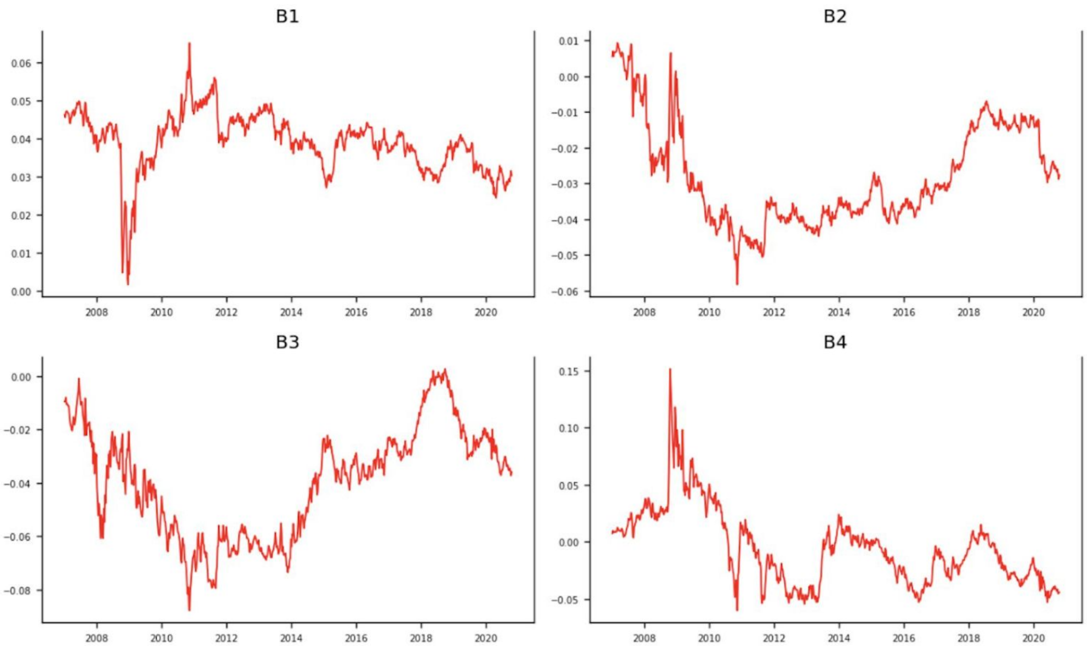






This betas were estimated for each week according with the spot rates and the λ_1 and λ_2 just proposed, by fitting the data.

The results are the time series of the 4 Beta parameters



Is worth mentioning that the slope parameter (B2) for being interpreted as the the slope must be considered as the negative of the B2 values; so for an actual visualization of the slope factor, it should be adjusted properly.

One singularity of these 4 factors is that they are not orthogonal, as opposed to the PCA procedure priorly described. Therefore, the factors can have a causality embedded between them, dependence that we will test with a Granger test.

	B1_x	B2_x	B3_x	B4_x
B1_y	1.0000	0.0000	0.0055	0.0011
B2_y	0.0001	1.0000	0.0000	0.0000
B3_y	0.0231	0.0203	1.0000	0.0058
B4_y	0.0021	0.0011	0.0094	1.0000

Looking at the above, a causation may be established:

This Granger test output shows the p-values for the causality between factors. For instance, B1_x ∩ B2_y value of 0.0001 means that the Null Hypothesis that the B1 can't predict B2 has a p-value of 0.0001 so it should be disregarded. Therefore, B1 has a significant predictability over B2.

With this table we can conclude some interesting things:
B1 can be caused by B2, B3 and B4.
B2 can be caused by B1,B3 and B4
B3 can be caused by B1,B2,B.
B4 seems to be caused by B1,B2,B3.
Indicatively a VAR model appears to be useful in establishing a relationship with a reasonable significance level.

The Akaike measure shows that the best possible lag is 1 period-lag.

Therefore we will try a VAR model with 1 lag for the 4 betas

```
  Summary of Regression Results
==================================
Model:                         VAR
Method:                        OLS
Date:           Sun, 29, Nov, 2020
Time:                     21:37:04
--------------------------------------------------------------------
No. of Equations:         4.00000    BIC:                   -42.5315
Nobs:                     119.000    HQIC:                  -42.8089
Log likelihood:           1903.00    FPE:                2.11854e-19
AIC:                     -42.9986    Det(Omega_mle):     1.79696e-19
--------------------------------------------------------------------
Results for equation B1
========================================================================
           coefficient       std. error           t-stat            prob
------------------------------------------------------------------------
const         0.008957         0.002848            3.145           0.002
L1.B1         0.756545         0.068910           10.979           0.000
L1.B2         0.100801         0.050552            1.994           0.046
L1.B3        -0.088047         0.037598           -2.342           0.019
L1.B4        -0.017296         0.016196           -1.068           0.286
========================================================================

Results for equation B2
========================================================================
           coefficient       std. error           t-stat            prob
------------------------------------------------------------------------
const        -0.009573         0.002967           -3.227           0.001
L1.B1         0.218978         0.071774            3.051           0.002
L1.B2         0.807514         0.052653           15.336           0.000
L1.B3         0.124079         0.039160            3.168           0.002
L1.B4         0.013293         0.016869            0.788           0.431
========================================================================

Results for equation B3
========================================================================
           coefficient       std. error           t-stat            prob
------------------------------------------------------------------------
const        -0.005272         0.004285           -1.231           0.219
L1.B1         0.054261         0.103658            0.523           0.601
L1.B2        -0.004610         0.076043           -0.061           0.952
L1.B3         0.939829         0.056557           16.617           0.000
L1.B4        -0.006710         0.024363           -0.275           0.783
========================================================================

Results for equation B4
========================================================================
           coefficient       std. error           t-stat            prob
------------------------------------------------------------------------
const        -0.017141         0.009550           -1.795           0.073
L1.B1         0.526979         0.231059            2.281           0.023
L1.B2        -0.197047         0.169505           -1.162           0.245
L1.B3         0.230502         0.126068            1.828           0.067
L1.B4         0.980674         0.054307           18.058           0.000
========================================================================
```

This VAR model is clear into identifying some important conclusions:

- The level factor (B1) can be explained only by one specific lagged value at the 1% significance: itself. Slope and short term curvature lagged values are only significant at the 5% confidence and long term curvature factor seems useless. Another interesting feature is that the constant parameter (intercept) has a positive and significant parameter. This is natural, as a persistent negative level over time is not something to be expected for a yield curve, as the time value of the money should be a positive and discounting factor instead of an amplifying one.

- The slope factor (B2) can be predicted with 1% significance using lagged values of level, slope and short term curvature. Again, long term curvature doesn't seem important. The intercept is also significant and negative, pointing out that the slope tends to decrease over time.

- Both short term and long term curvature can only be predicted with a high level of significance with their own lagged values; level and slope doesn’t seem to be important factors for them.

Economic parameters

We tried to include the economic parameters like money supply, CPI, industrial production change, unemployment rate and GDP in the VAR for betas but found that their coefficients are very insignificant in predicting betas. Our VAR results showed economic parameters had no significance in our model as can be seen below. As such we chose to omit the economic variables from our VAR for betas.

Results for equation PCA1

	coefficient	std. error	t-stat	prob
const	0.002765	0.010561	0.262	0.793
L1.CPI YOY Index	-0.003366	0.001861	-1.809	0.071
L1.GDP CURY Index	0.001680	0.001001	1.678	0.093
L1.FDTR Index	-0.002344	0.001362	-1.721	0.085
L1.US Unemployment rate	0.000071	0.001237	0.057	0.954
L1.Inudstrial Production change Prev Month	-0.000502	0.001586	-0.317	0.752
L1.M2 Money Supply	-0.000029	0.000015	-1.918	0.055
L1.PCA1	0.058668	0.017942	3.270	0.001
L1.PCA2	0.094482	0.028173	3.354	0.001
L1.PCA3	-0.092217	0.045941	-2.007	0.045

Results for equation PCA2

	coefficient	std. error	t-stat	prob
const	-0.006348	0.006687	-0.949	0.342
L1.CPI YOY Index	0.001155	0.001179	0.980	0.327
L1.GDP CURY Index	0.001085	0.000634	1.711	0.087
L1.FDTR Index	-0.001004	0.000862	-1.164	0.244
L1.US Unemployment rate	0.000300	0.000783	0.383	0.702
L1.Inudstrial Production change Prev Month	0.002370	0.001004	2.360	0.018
L1.M2 Money Supply	0.000006	0.000010	0.576	0.564
L1.PCA1	0.060372	0.011360	5.314	0.000
L1.PCA2	0.087896	0.017837	4.928	0.000
L1.PCA3	0.008633	0.029087	0.297	0.767

Results for equation PCA3

	coefficient	std. error	t-stat	prob
const	0.003208	0.003687	0.870	0.384
L1.CPI YOY Index	0.000286	0.000650	0.441	0.659
L1.GDP CURY Index	-0.000121	0.000350	-0.347	0.729

This is how we regressed the economic parameters and the lagged beta parameters:

B1(t) = B1(t-1) + B2(t-1) + B3(t-1) + B4(t-1) + GDP(t-1) + Money supply(t-1) + Unemployment(t-1) + CPI(t-1) + Industrial Production Change(t-1)

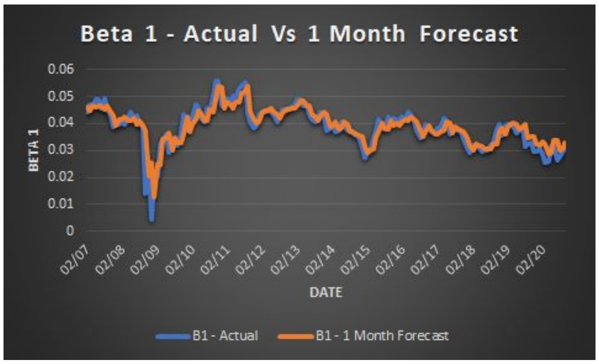
B2(t) = B1(t-1) + B2(t-1) + B3(t-1) + B4(t-1) + GDP(t-1) + Money supply(t-1) + Unemployment(t-1) + CPI(t-1) + Industrial Production Change(t-1)

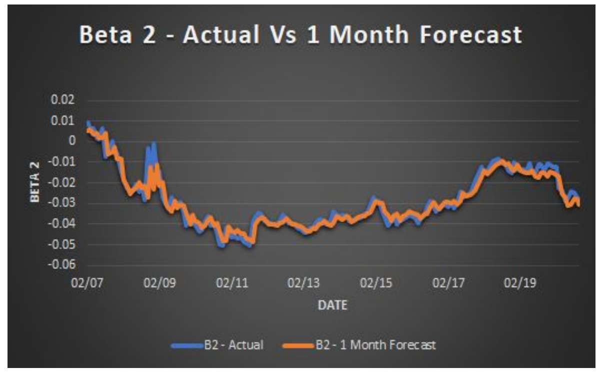
B3(t) = B1(t-1) + B2(t-1) + B3(t-1) + B4(t-1) + GDP(t-1) + Money supply(t-1) + Unemployment(t-1) + CPI(t-1) + Industrial Production Change(t-1)

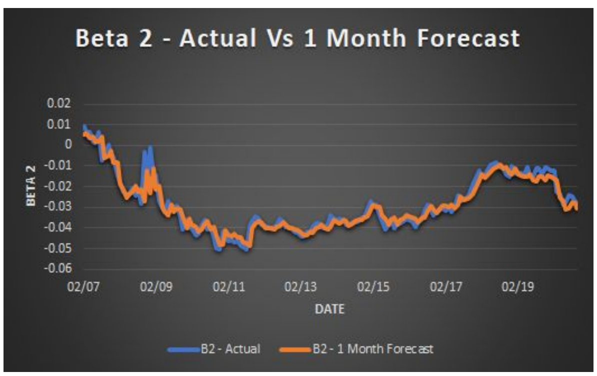
B4(t) = B1(t-1) + B2(t-1) + B3(t-1) + B4(t-1) + GDP(t-1) + Money supply(t-1) + Unemployment(t-1) + CPI(t-1) + Industrial Production Change(t-1)

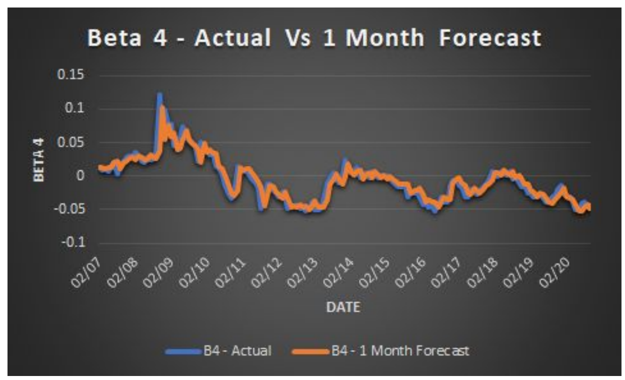
5. Results:

Beta Forecasting: In terms of predicting Beta coefficients, the VAR(1) model does a phenomenal job. We used the forecasted values of all beta coefficients uptil 12 months as inputs for the NSS model.



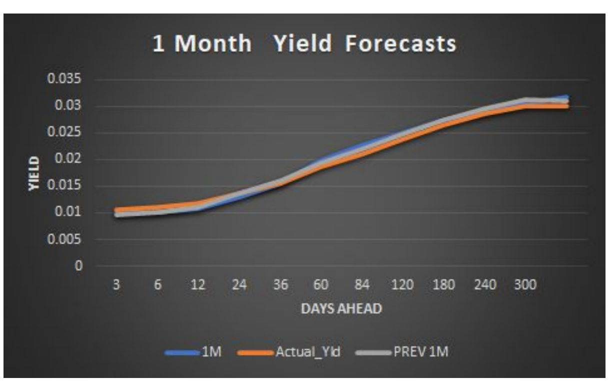


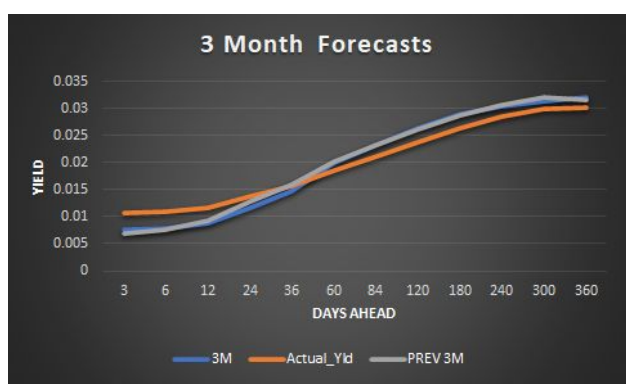


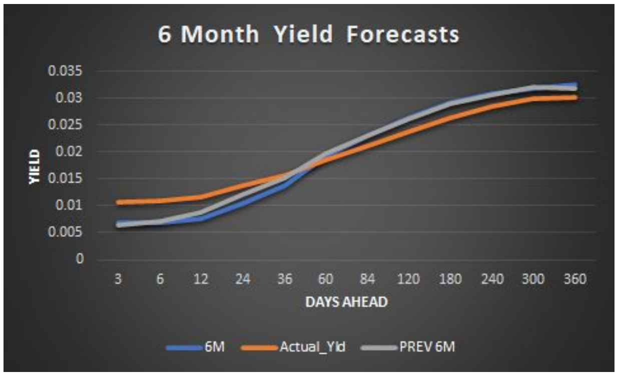


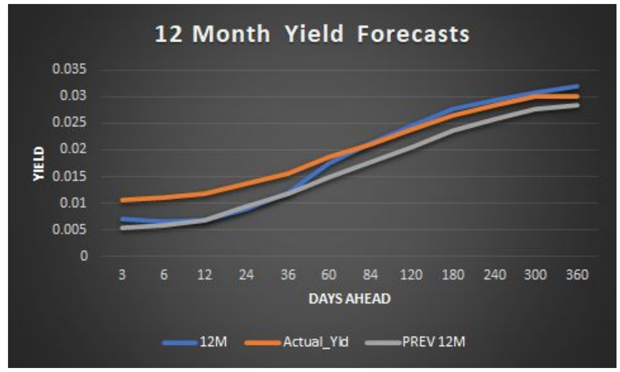
We then tried to perform a comparative analysis between a time where the economic scenario was stable like in 2017 versus the current unprecedented time we are in 2020. The results are as can be seen as follows:

As of 05/31/2017:

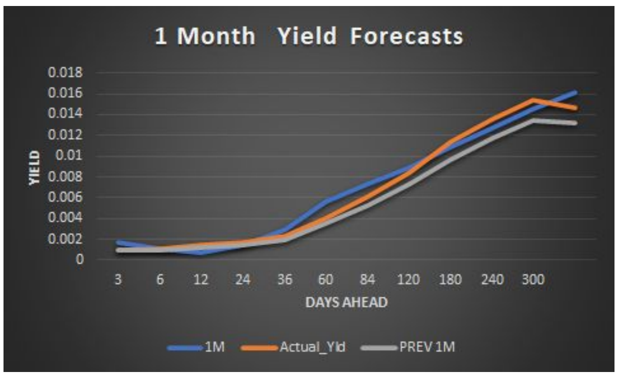


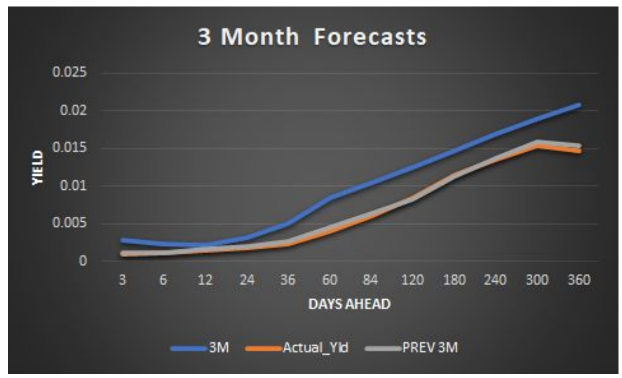


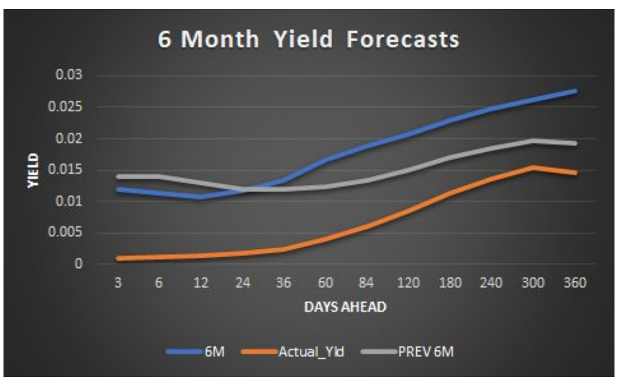


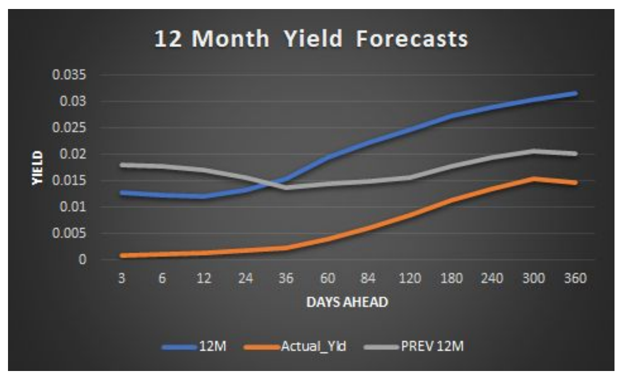


As of 08/26/2020:









6. Conclusion

As per our observation of the in-sample forecasts:

1. For the beta coefficients, we observe that the model provides accurate predictions.
2. The model is capable of accurately predicting yield curve movements in the shorter term. The predictions for longer term show higher dispersion, especially in unprecedented times of extreme economic movements, for instance the pandemic. However, when the economy is stable, the model does incorporate micro changes and predict the longer term movements in the yields relatively well.

In the times of economic uncertainty we can run the strategy again to factor in the changes and better predict the yield curve. However, we need to find a more efficient way to incorporate uncertainty for predictions.

However, predicting Yield curve continues to be a guessing game. Though a lot of research has been done on the subject there is no one model in the market that is able to accurately estimate the yield curves. Fund houses would make millions and billions of dollars with that model. Also the factors that effect the yield are very diverse and some may not even have been researched upon and are yet to be discovered. Our idea here was to give a close approximation of what the yield curve would look like in the future. However, estimates one year from now are highly unreliable and we could not find a model that can predict for that long in the future per our research.

7.References

1. Moody's Analytics, 2014, *Principal Component Analysis for Yield Curve Modelling*
2. Erhard Reschenhofer, Thomas Stark, 2019, *Forecasting the Yield Curve with dynamic Factors*
3. Francis X. Diebold, Glenn D. Rudebuschb, S. Boragan Aruoba, 2005, *The macroeconomy and the yield curve: a dynamic latent factor approach*
4. Breach, D'amico, Orphanides *,The term structure and Inflation uncertainty*
5. John Cochrane, Monika Piazzesi, 2002, *The Fed and Interest Rates: A high Frequency Identification*
6. Francis X. Diebold, Canlin Li, 2003, *Forecasting the term structure of government yield bonds*
7. Jonathan Hartley, Krista Schwarz, 2019, *Predictable End-of-Month Treasury Returns*
8. Grace Xing Hu, Jun Pan, Jiang Wang, 2013, *Noise as information for Illiquidity*
9. Francis X. Diebold, Glenn D. Rudebuschb, S. Boragan Aruoba, 2005, *The macroeconomy and the yield curve: a dynamic latent factor approach*

8. Appendix

Code1: PCA over differenced yields (Python)

```
import os
os.chdir("/Users/idhru/Desktop/Practice/AFP files")
import pandas as pd
import numpy as np
import warnings
warnings.filterwarnings('ignore')
import scipy
import sklearn.decomposition as skdec
import statsmodels.api as sm
import matplotlib
import matplotlib.pyplot as plt
import matplotlib.patches as mpathes
matplotlib.style.use('ggplot')
get_ipython().run_line_magic('matplotlib', 'inline')
from IPython.display import display, HTML, Image

# # 1 Month

onemonth = pd.read_excel('1 Month.xlsx', sheet_name = "Worksheet",skiprows =
range(1,6))
onemonth.columns = ["Date","PX_MID","YLD_CNV_MID"]
onemonth = onemonth[['Date','PX_MID']]
onemonth.columns = ["Date","1m"]
onemonth


# # 3 Month
threemonth = pd.read_excel('3 Month.xlsx', sheet_name = "Worksheet",skiprows
= range(1,6))
threemonth.columns = ["Date","PX_MID","YLD_CNV_MID"]
threemonth = threemonth[['Date','PX_MID']]
```

```
threemonth.columns = ["Date","3m"]
threemonth


# # 1 Year
oneyear = pd.read_excel('1 Year.xlsx',skiprows = range(1,6))
oneyear.columns = ["Date","PX_MID","YLD_CNV_MID"]
oneyear = oneyear[['Date','PX_MID']]
oneyear.columns = ["Date","1Y"]
oneyear


# # 5 Year
fiveyear = pd.read_excel('5 Year.xlsx',skiprows = range(1,6))
fiveyear.columns = ["Date","PX_MID","YLD_CNV_MID"]
fiveyear = fiveyear[['Date','PX_MID']]
fiveyear.columns = ["Date","5Y"]
fiveyear


# # 10 Year
tenyear = pd.read_excel('10 Year.xlsx',skiprows = range(1,6))
tenyear.columns = ["Date","PX_MID","YLD_CNV_MID"]
tenyear = tenyear[['Date','PX_MID']]
tenyear.columns = ["Date","10Y"]
tenyear


# # Merged data
df=pd.merge(onemonth,threemonth, on = "Date")
df=pd.merge(df,oneyear, on = "Date")
df=pd.merge(df,fiveyear, on = "Date")
df=pd.merge(df,tenyear, on = "Date")
df=df.sort_values(by="Date")
df=df.reset_index(drop=True)
```

```
# # Creating Difference Matrix
mat = df[['1m','3m','1Y','5Y','10Y']]
mat_diff = mat.diff(axis=0)
mat_diff = mat_diff.dropna()
mat_diff


# # PCA
from sklearn import preprocessing
from sklearn.decomposition import PCA
pca = PCA(n_components = 3)
pca_data = pca.fit_transform(mat_diff.T)
principalDf = pd.DataFrame(data = pca_data , columns = ['principal component 1', 'principal component 2','principal component 3'])
principalDf
per_var = np.round(pca.explained_variance_ratio_* 100, decimals=1)
labels = ['PC' + str(x) for x in range(1, len(per_var)+1)]

plt.bar(x=range(1,len(per_var)+1), height=per_var, tick_label=labels)
plt.ylabel('Percentage of Explained Variance')
plt.xlabel('Principal Component')
plt.title('Scree Plot')
plt.show()



# ## Fraction explained by Principal Components

per_var
# Output : array([81.5,  9.5,  5.2])


# ## Scores obtained using Principal Components

principalDf2 = np.dot(mat_diff,principalDf)
```

```
principalDf2 = pd.DataFrame(data = principalDf2 , columns =
['PCA_vec1','PCA_vec2','PCA_vec3'])
principalDf2 = pd.concat([df["Date"][:2722],principalDf2], axis =1)
principalDf2


# ## Plots for a subset
p= principalDf2.iloc[100:150,:]
pd.DataFrame({"Shift": p.iloc[:,1].values, "Change in Slope":
p.iloc[:,2].values,"Change in Convexity": p.iloc[:,3].values}, index=
p.iloc[:,0].values).plot(figsize=(10,40), subplots=True)
```

Code 2: PCA over yields (R)

```
library("readxl")
library(data.table)
sci.option=999

#Load data
#1m
data1m=as.data.table(read_xlsx("1 month.xlsx",skip=4,col_names = T))
data1m=data1m[,-2]
colnames(data1m)=c("Date","1m")
# setorder(data1m,Date)

#3m
data3m=as.data.table(read_xlsx("3 Month.xlsx",skip=4,col_names = T))
data3m=data3m[,-2]
colnames(data3m)=c("Date","3m")

#1y
data1y=as.data.table(read_xlsx("1 year.xlsx",skip=4,col_names = T))
data1y=data1y[,-2]
colnames(data1y)=c("Date","1y")
```

```
#5y
data5y=as.data.table(read_xlsx("5 year.xlsx",skip=4,col_names = T))
data5y=data5y[,-2]
colnames(data5y)=c("Date","5y")

#10y
data10y=as.data.table(read_xlsx("10 year.xlsx",skip=4,col_names = T))
data10y=data10y[,-2]
colnames(data10y)=c("Date","10y")

#Format the dates
data1m$Date=as.Date(as.character(data1m$Date,"%Y-%m-%d"))
data3m$Date=as.Date(as.character(data3m$Date,"%Y-%m-%d"))
data1y$Date=as.Date(as.character(data1y$Date,"%Y-%m-%d"))
data5y$Date=as.Date(as.character(data5y$Date,"%Y-%m-%d"))
data10y$Date=as.Date(as.character(data10y$Date,"%Y-%m-%d"))

#Merge all rates data
x=merge(data1m,data3m, by="Date")
x=merge(x,data1y, by="Date")
x=merge(x,data5y, by="Date")
x=merge(x,data10y, by="Date")

#means of each rate
means=sapply(x,mean)

#Demean the rates
PCA=matrix(0,nrow=2723,ncol=1)
PCA=as.data.table(PCA)
PCA$Date=x$Date
PCA$`1m`=x$`1m`-means[2]
PCA$`3m`=x$`3m`-means[3]
PCA$`1y`=x$`1y`-means[4]
PCA$`5y`=x$`5y`-means[5]
PCA$`10y`=x$`10y`-means[6]
PCA$V1=NULL
```

PCA

```
#Cov matrix over demeaned rates
Cov_matrix=cov(PCA[,-1])

#Eigenvalues + Eigenvectors
PCA_eigenvalues=eigen(Cov_matrix)$values
PCA_eigenvectors=-eigen(Cov_matrix)$vectors
rownames(PCA_eigenvectors)=c("1m","3m","1y","5y","10y")
colnames(PCA_eigenvectors)=c("PC1 (level)","PC2 (slope)","PC3
(curvature)","PC4","PC5")
PCA_eigenvectors

#PCA1 contribution to variance
PCA_eigenvalues[1]^2/(t(PCA_eigenvalues)%*%PCA_eigenvalues)

#PCA2 contribution to variance
PCA_eigenvalues[2]^2/(t(PCA_eigenvalues)%*%PCA_eigenvalues)

#PCA3 contribution to variance
PCA_eigenvalues[3]^2/(t(PCA_eigenvalues)%*%PCA_eigenvalues)

#Transform data
PCA1=t(t(PCA_eigenvectors[,1:3])%*%t(PCA[,-1]))
PCA1=t(t(as.matrix(PCA_eigenvectors[,1:3]))%*%t(PCA[,-1]))

tenures=c("1m","3m","1y","5y","10y")
#plot
plot(x=PCA$Date, y=PCA1[,1],type="l", col= "black",ylab="changes in yield",
    xlab="Date",main="Principal Component Analysis over yields",ylim=c(-4,4))
lines(x= PCA$Date, y= PCA1[,2], type="l", col="red")
lines(x= PCA$Date, y= PCA1[,3], type="l", col="green")
legend("bottomleft",legend=c("level (93.6%)","slope (6.35%)","convexity
(0.1%)"),col=c(1:3),pch = c(19,19))
```

```
plot(PCA_eigenvectors[,1],type="o",ylim=c(-1,1),main="PCA Factor loadings over
UST yields",ylab="Loadings",
    xaxt = "n",xlab="Tenures")
lines(PCA_eigenvectors[,2],type="o",col="red")
lines(PCA_eigenvectors[,3],type="o",col="chartreuse3")
axis(1, at=1:5, labels=tenures)
legend("bottomleft",legend=c("level","slope","convexity"),col=c(1:3),pch =
c(19,19))



#Box-Pierce test
#Level
Box.test(PCA1[,1],type="Box-Pierce")

        Box-Pierce test

data:  PCA1[, 1]
X-squared = 2720.3, df = 1, p-value < 2.2e-16

#Slope
Box.test(PCA1[,2],type="Box-Pierce")

        Box-Pierce test

data:  PCA1[, 2]
X-squared = 2689.3, df = 1, p-value < 2.2e-16


#Curvature
Box.test(PCA1[,3],type="Box-Pierce")

Box-Pierce test
```

data: PCA1[, 3]
X-squared = 2700.9, df = 1, p-value < 2.2e-16

Code 3: VAR using Diebolt, Li (2003) approach (Python)

```
import pandas as pd
import numpy as np
import scipy as sp
from scipy.interpolate import CubicSpline
from scipy.optimize import root
from scipy.optimize import minimize
spt= pd.read_csv("Spot_Curve_Weekly.csv")

def nss_model(t,b1,b2,b3,b4,lam1,lam2):
    st= b1+
b2*((1-np.exp(-lam1*t))/(lam1*t))+b3*(((1-np.exp(-lam1*t))/(lam1*t))-np.exp(-lam1*t))+b4*(((1-np.exp(
-lam2*t))/(lam2*t))-np.exp(-lam2*t))
    return st



def est_param_nss(y,t,lam1,lam2):

    on= np.ones(t.shape[0])
    X=
np.vstack([on,(1-np.exp(-lam1*t))/(lam1*t),((1-np.exp(-lam1*t))/(lam1*t))-np.exp(-lam1*t),((1-np.exp(-l
am2*t))/(lam2*t))-np.exp(-lam2*t)])
    X= np.transpose(X)
    X= np.matrix(X)
    B= np.linalg.inv(X.T*X)*X.T*y

    return np.array(B.T)[0]

#Computing the values of l1 and l2 that define humps, capturing medium and long term term effects

#first hump at 30 months

t=30
lam1= minimize(lambda l:-1*(((1-np.exp(-l*t))/(l*t))-np.exp(-l*t)),x0=[0.6]).x[0]
t=180
```

```
lam2= minimize(lambda l:-1*(((1-np.exp(-l*t))/(l*t))-np.exp(-l*t)),x0=[0.6]).x[0]

beta_df=
pd.DataFrame({'Date':spt.iloc[:,0].values,'B1':np.zeros(spt.shape[0]),'B2':np.zeros(spt.shape[0]),'B3':
np.zeros(spt.shape[0]),'B4':np.zeros(spt.shape[0])})
t= np.array([3,6,12,2*12,3*12,5*12,7*12,10*12,15*12,20*12,25*12,30*12])
for i in range(beta_df.shape[0]):
   y= np.matrix(spt.iloc[i,1:].values).T
   beta_df.iloc[i,1:]= est_param_nss(y,t,lam1,lam2)

import matplotlib.pyplot as plt
from statsmodels.tsa.api import VAR
from statsmodels.tsa.stattools import adfuller
from statsmodels.tools.eval_measures import rmse, aic
import datetime as dt
beta_changes= pd.read_csv("betas.csv")
beta_df['Date']= pd.to_datetime(beta_df['Date'],format= '%d/%m/%Y')
beta_df= beta_df.set_index(beta_df['Date'])
beta_df= beta_df.iloc[:,1:]
beta_df



fig, axes = plt.subplots(nrows=2, ncols=2, dpi=120, figsize=(10,6))

for i, ax in enumerate(axes.flatten()):
   data = beta_df[beta_df.columns[i]]
   ax.plot(data, color='red', linewidth=1)
   ax.set_title(beta_df.columns[i])
   ax.spines["top"].set_alpha(0)
   ax.tick_params(labelsize=6)
plt.tight_layout()



#Testing Causality

from statsmodels.tsa.stattools import grangercausalitytests
maxlag=12
def grangers_causation_matrix(data, variables, test='ssr_chi2test', verbose=False):
   df = pd.DataFrame(np.zeros((len(variables), len(variables))), columns=variables, index=variables)
   for c in df.columns:
      for r in df.index:
         test_result = grangercausalitytests(data[[r, c]], maxlag=maxlag, verbose=False)
         p_values = [round(test_result[i+1][0][test][1],4) for i in range(maxlag)]
```

```
            if verbose: print(f'Y = {r}, X = {c}, P Values = {p_values}')
            min_p_value = np.min(p_values)
            df.loc[r, c] = min_p_value
    df.columns = [var + '_x' for var in variables]
    df.index = [var + '_y' for var in variables]
    return df
grangers_causation_matrix(beta_df, variables = beta_df.columns)

def adfuller_test(series, signif=0.05, name='', verbose=False):

    r = adfuller(series, autolag='AIC')
    output = {'test_statistic':round(r[0], 4), 'pvalue':round(r[1], 4), 'n_lags':round(r[2], 4), 'n_obs':r[3]}
    p_value = output['pvalue']
    def adjust(val, length= 6): return str(val).ljust(length)

    # Print Summary
    print(f'    Augmented Dickey-Fuller Test on "{name}"', "\n   ", '-'*47)
    print(f' Null Hypothesis: Data has unit root. Non-Stationary.')
    print(f' Significance Level    = {signif}')
    print(f' Test Statistic        = {output["test_statistic"]}')
    print(f' No. Lags Chosen       = {output["n_lags"]}')

    for key,val in r[4].items():
        print(f' Critical value {adjust(key)} = {round(val, 3)}')

    if p_value <= signif:
        print(f" => P-Value = {p_value}. Rejecting Null Hypothesis.")
        print(f" => Series is Stationary.")
    else:
        print(f" => P-Value = {p_value}. Weak evidence to reject the Null Hypothesis.")
        print(f" => Series is Non-Stationary.")

for name, column in beta_df.iteritems():
    adfuller_test(column, name=column.name)
    print('\n')

df_differenced = beta_df.diff().dropna()

for name, column in df_differenced.iteritems():
    adfuller_test(column, name=column.name)
    print('\n')

model = VAR(df_differenced)
for i in [1,2,3,4,5,6,7,8,9,10,11,12]:
```

```
    result = model.fit(i)
    print('Lag Order =', i)
    print('AIC : ', result.aic)
    print('BIC : ', result.bic)
    print('FPE : ', result.fpe)
    print('HQIC: ', result.hqic, '\n')

model_fitted = model.fit(5)
model_fitted.summary()

model_fitted.fittedvalues.to_csv("fit.csv")
beta_df.to_csv("org.csv")
```

Python Code:

```
# In[1]:


import os
os.chdir("/Users/idhru/Desktop/AFP")
import pandas as pd
import numpy as np
import warnings
warnings.filterwarnings('ignore')
import scipy
import sklearn.decomposition as skdec
import statsmodels.api as sm
import matplotlib
import matplotlib.pyplot as plt
import matplotlib.patches as mpathes
matplotlib.style.use('ggplot')
get_ipython().run_line_magic('matplotlib', 'inline')
from IPython.display import display, HTML, Image
from sklearn.preprocessing import StandardScaler


# In[2]:


data = pd.read_csv('treasury_data.csv')


# # Weekly data

# In[3]:


data = data.iloc[::5,:].reset_index()
data
```

```
# In[4]:


mat = data[['3_Mo','6_Mo','1_Yr','2_Yr','3_Yr','5_Yr','7_Yr','10_Yr','20_Yr','30_Yr']]
mat = mat.dropna()
mat


# In[5]:


col = mat.columns
col


# # Standardizing Data

# In[6]:


mat = StandardScaler().fit_transform(mat)
mat = pd.DataFrame(mat)
mat.columns = col
mat


# # PCA

# In[7]:


from sklearn import preprocessing
from sklearn.decomposition import PCA
pca = PCA(n_components = 3)
```

```
#pca = PCA()
pca_data = pca.fit_transform(mat)
principalDf = pd.DataFrame(data = pca_data, columns = ['L', 'S','C'])
principalDf


# In[8]:


per_var = np.round(pca.explained_variance_ratio_* 100, decimals=1)
labels = ['PC' + str(x) for x in range(1, len(per_var)+1)]

plt.bar(x=range(1,len(per_var)+1), height=per_var, tick_label=labels)
plt.ylabel('Percentage of Explained Variance')
plt.xlabel('Principal Component')
plt.title('Scree Plot')
plt.show()


# In[9]:


per_var


# # AR Process

# In[10]:


principalDf2 = principalDf.diff(-1).dropna().copy()


# In[11]:
```

```
principalDf2.columns = ['L_diff','S_diff','C_diff']
principalDf2


# In[12]:


import statsmodels.api as sm

x_var = principalDf2[['L_diff','S_diff','C_diff']].shift(-1)
x_var = x_var.dropna()
model = sm.OLS(principalDf2.L_diff[0:(len(principalDf2)-1)], x_var).fit()
model.summary()


# In[13]:


modelS= sm.OLS(principalDf2.S_diff[0:(len(principalDf2)-1)], x_var).fit()
modelS.summary()


# In[14]:


modelC = sm.OLS(principalDf2.C_diff[0:(len(principalDf2)-1)], x_var).fit()
modelC.summary()


# In[ ]:




# # Rough
```

```
# In[ ]:




# In[15]:

from sklearn.model_selection import train_test_split
train, test = train_test_split(principalDf2, test_size=0.1, shuffle=False)


# In[16]:

from statsmodels.tsa.api import VAR
model = VAR(train)
results = model.fit()
results.summary()


# In[17]:

results.plot()[0:3]
```

```
#Nelson - Siegel Betas plots (useful for interpretation) - R based code

l=0.0597760632423479
t=seq(1,360,1)
l2=0.00996267604057902l
getwd()



par(mfrow=c(1,4))
plot(y=rep(1,360),x=t,type="l", main="Factor decay for the B1~level",
    xlab="Months",ylab="Relative importance of the level factor",cex.axis=0.8)


plot((y=1-exp(-l*t))/(l*t),x=t,type="l", main="Factor decay for the B2~slope",
    xlab="Months",ylab="Relative importance of the slope factor",cex.axis=0.8)


plot((y=1-exp(-l*t))/(l*t)-exp(-l*t),x=t,type="l", main="Factor decay for the
B3~curvature",
    xlab="Months",ylab="Relative importance of the short term curvature
factor",cex.axis=0.8,cex.lab=0.8)

plot((y=1-exp(-l2*t))/(l2*t)-exp(-l2*t),x=t,type="l", main="Factor decay for the
B4~curvature",
    xlab="Months",ylab="Relative importance of the long term curvature
factor",cex.axis=0.8,cex.lab=0.8)
```