Principal Components Analysis of Yield Curves

Dipan Biswas
dipanbiswasiitkgp@gmail.com

Abstract

It is widely accepted fact that the evolution of yield curves is driven by some independent factors. Through Singular Value Decomposition and subsequent Principal Components Analysis, this paper tries to (1) examine how many principal components are required to explain variance in yield curves, (2) see if these factors are correlated for yield curves of a few developed economies, (3) regress macro-economic indicators against these factors, (4) see if a global set of factors can be developed that explains variance for entire set of yield curves, (5) analyze stability of factor loadings across time periods, (6) project factors into the future and develop yield curve predictions for near term, (7) study shape and frequency distribution of factors derived from percentage change in yields and simulate yield curves from known distributions, (8) synthetically create pure factor portfolios by combining yields of different maturities.

Keywords: *Yield Curves; Principal Components Analysis*

1. Introduction

It is widely accepted that the evolution of the yield curve is driven by some independent factors. There is however no unanimous consensus on the number of factors that actually impact the shape of the curve. Also, different studies have different inferences on the impact of each factor. However, most researchers point to the fact that two to three factors can explain almost all variance within the yield curve. Such factors are generally extracted by Principal Components Analysis. Litterman and Sheinkman [1991] used the terms level, steepness and curvature to name the three factors that they found explain over 95% of the variance of US treasuries across various periods. They hypothesized that hedging with these 3 factors would create a more perfect hedge than duration hedging. This research then paved way for many such researches to understand the nature of these factors.

Barber and Copper [2012] used cubic spline interpolation as suggested by Bliss [1997] to create a common set of nodes for entire historical data between 1992 and 2001. This then allowed them to run statistical studies on the consistency of these factors between various windows. They found that first 2 principal components explain 93% of the variance at 90% confidence level. They introduced a statistical element to principal components which was earlier missing. However, whether or not we can assume normality with principal components is still a matter of debate.

Novosyolov and Satchkov [2008] attempted to model jointly the global term structure of interest rates. Modelling the curves of various developed economies together is important in the sense that the global economy is more inter-connected than ever before. But as they rightly point out, there is not much research on this subject. They model both government and LIBOR curves of USA, EU, UK, Japan and Canada. Instead of using the spot rates themselves, they used percentage change in the spot rates as the raw data. They feel it removes a lot of autocorrelation in the spot data by doing an order 1 differencing and also taking percentages removes order of magnitude differences that may occur with using absolute changes of rates in different countries. In one style, they take all the raw data together and find significant factors for all of them. In another, they find significant factors in each treasury, put the factors together and find secondary set of factors that explain these primary factors. They find that 16 principal components explain most of the variance and that the second style is better as they preserve the characteristics of each factor – level, steepness, curvature. This research, although a fresh thought, can be easily extended by looking at the stability of these factors across time periods and investigating its predictive power.

Riesman and Zohar [2004] have tried to predict short-term developments in the term structure. They break down the yield curves into principal components. They use ARMA processes to model each factor and recombine them linearly by factor loadings. The reason we can do this is because Principal Components decomposition is an ad-hoc process and as such, there is no assumption on the dynamics of the underlying process. They use these predictions to design portfolios for trading and then use out-of-sample data to find out the alpha they generate by this mechanism. However, there is not much emphasis on the model-fit and quality of predictions. This research can thus be taken forward by not only doing error tests on the model but by seeing stability of the models themselves in various time periods.

Scenario simulation is a widely used technique in calculating Value-at-Risk of bond portfolios. However, Monte Carlo simulations of yield curve is still not a well-researched topic. Jamshedian and Zhu [1997] alludes to using principal components for simulating yield curves. But they resort to joint probability distribution method of all the nodes to simulate the curves. This is a computationally expensive technique as the number of simulations to produce a sufficiently large sample is huge (as they point out ~10^6). Also, this requires assuming distributions for all the nodes which may be unnecessary because a few factors can explain the variance in all the nodes. Also, there is not much reference in the paper or any other research papers about the frequency distribution of the percentage change in principal components themselves. This could be an interesting area to explore if known distributions can be used to simulate the factors and recombining them can give us simulated yield curves.

There has been substantial research about the effect of macroeconomic news on yields. Gurkainak [2014] found that the effect on daily variations because of the news is very little. However, Ang and Piazzesi (2003) find that macroeconomic variables explain 85% of the variance in short and medium maturities and 60% in long maturities. It would be interesting to see which macro-economic variables each individual factors correspond to the most. If we find very high correlation of a factor with one or a group of related macro-economic variables, we may be able to understand the factors better. This may help us in explaining the movements of yield curves with actual movements in the real world economy.

The majority of the research, till date, with principal components and yield curves, however, is to find out effective ways of hedging yield curve risk. There is a plethora of good research on this subject. Papers include Golub and Tilman [1997], Falkenstein and Hanweck [1997], Carcano [2009], Carcano and Dall'O [2010] among others. Since, this is already a well-researched area, this project won't go into the topics of hedging and risk management while admitting it is probably one of the most common and practical uses of Principal Components Analysis of yield curves.

2. Principal Components Analysis

The main idea behind Principal Components Analysis (PCA) is that a system with high dimensionality can be reduced to a lower dimension system with reasonable accuracy by taking advantage of the correlations that exist within the variables. As mentioned before, one of the major uses of PCA in finance is to model yield curve dynamics. Since points in a yield curve seldom move independently, PCA can be used extract factors which determine the movements of the yield curve as a whole. The reduced system, after applying PCA, has fewer sources of uncertainty that needs to be studied. Hence, studying the principal components is an effective proxy for analyzing the yield curves as a whole.

We extract the Principal Components form the yield curves by performing Singular Value Decomposition on raw data matrix of the yield curves or rate of change of yield curves. Suppose we have a t x n matrix F, where t = 1,…,T, which is nothing but the period of study under consideration and n = 1,….,N, which are the maturities that make up a single yield curve. Thus, the rows represent the date index and the columns represent the maturities.

$$F = [F_{t,n}]$$

The first step is to transform the data matrix such that it has 0 mean and unit variance. This is done by subtracting the column means from each column and then dividing it by the standard deviation of the columns. Then we apply Singular Value Decomposition (SVD) on the transformed matrix. SVD reduces a t x n matrix F to three components as follows:

$$F = USV^T$$

U is a t x n orthonormal matrix, S is a n x n diagonal matrix containing the singular values and V is a n x n orthonormal matrix. The columns of U are eigenvalues of of FF^T and the columns of V are eigenvectors of F^TF. The diagonal values of S, called the Singular Values are all non-negative. They are the positive square root of the eigenvalues of the matrix F^TF. The matrix produced by multiplying U and S gives the principal components columns, P which has dimensions t x n. A system with n vectors will thus have n principal components. The matrix V is called factor loadings. It is important to note that V is time invariant. Thus, factor loadings can be considered a list of coefficients when multiplied with the principal components matrix P, gives back the raw data. We should also note that since matrix F is scaled, F^TF is equal to Cov(F). The proportion of variance explained by each PC is equal to $S(i)^{\frac{1}{2}}/\Sigma S(i)$, where S(i) is the i^{th} principal component. This metric can be used to choose the appropriate number of principal components and discard the rest. Suppose, we choose the first 3 principal components. Then, our principal components matrix is reduced to P’ with dimension t x 3. Similarly, we choose the first 3 columns of the factor loadings matrix V. Thus, we have V’ with dimension n x 3. Multiplying, P’ and $(V')^T$ should give back F’ which should be reasonably close to our original matrix F.

3. Data Collection and Pre-processing

Quandl.com is a marketplace for financial data. It lists several data aggregators who compile financial and economic data from all over the world. Quandl.com also provides seamless API services through which data can be imported into R, Python and Matlab using API function calls. The historical yield curve data of USA, Canada, Japan and Switzerland are imported from Global Yields Curves data aggregator listed on Quandl.com. We import it into R and do the subsequent analysis there. The data history is different for different countries. To achieve consistency we truncate the data from January 2005 to December 2019 for all the countries.

In addition to this, macroeconomic data is imported into R from FRED database, which is also listed on Quandl.com. In particular, we take GDP, Unemployment Rate, CPI, Industrial Production, Capacity Utilisation and WTI Oil prices of USA. Gold prices are imported from London Bullion Market Association, also listed on Quandl.com.

Quandl.com APIs can be called in R by installing the Quandl package. In addition, we use several other packages in R and use their functions to make our calculations, analysis and visualizations easier and faster. The packages used in this project are Quandl, dplyr, zoo, ggplot, corrplot, tseries, TSA, forecast, PerformanceAnalytics, TTR, Mass, LaplacesDemon, NormalLaplace, quantmod and tidyr.

4. Principal Components Analysis of US Treasuries

We will apply PCA over US Treasuries yields from January 2000 to December 2019. The yield curve consists of 1 year, 2 year, 3 year, 5 year, 7 year, 10 year, 20 year and 30 year maturities. First, we will fit a cubic spline through the points to fill the gaps and get maturities for all 30 years. We do that using the *spline* function in R which is a part of stats package. Cubic spline is the most commonly used spline for yield curve interpolation. Fig. 1 is the yield curve of 02-01-2002 and its cubic spline.

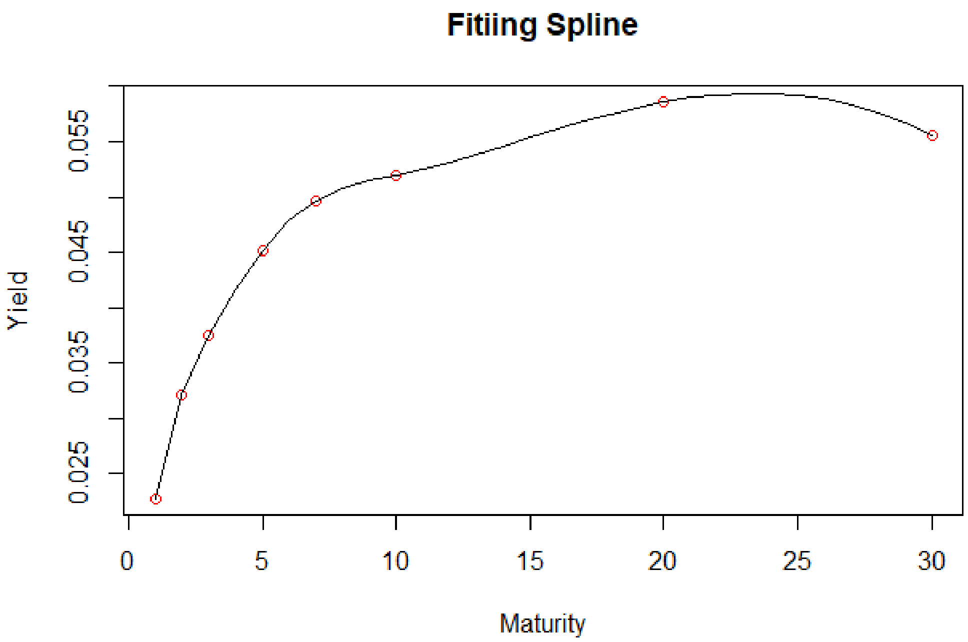


Fig. 1: Cubic Spline Interpolation

We fit the PCA model on the interpolated yield curves. 30 maturities will give us 30 Principal Components. We will look at the cumulative variance explained by each successive Principal Component and take a call on how many Principal Components we will need to capture most of the variance in the data and assume that the rest of the variance is due to random shocks. Fig. 2 plots first 10 Principal Components and their respective cumulative variance explained.

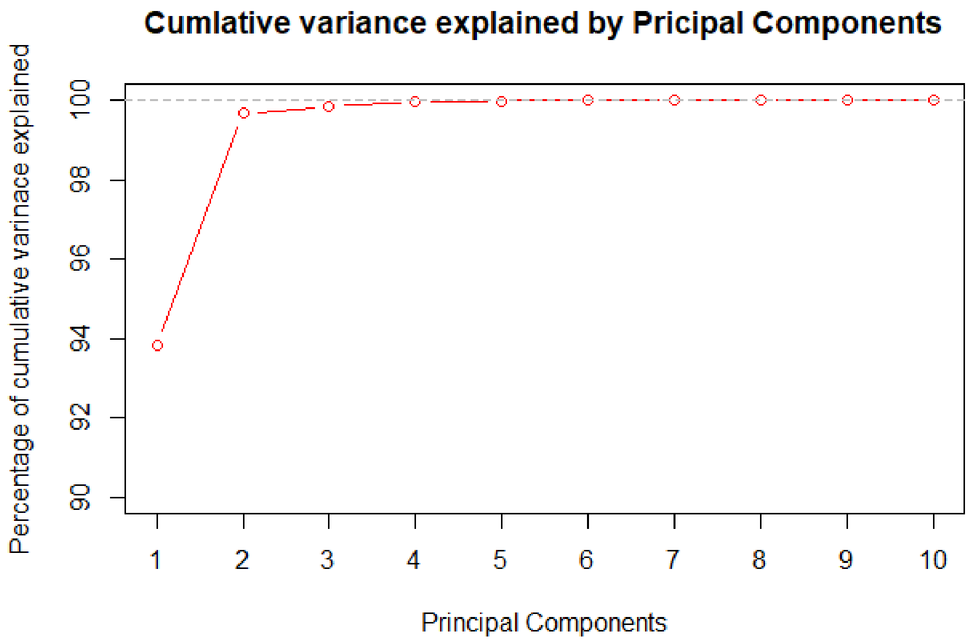


Fig. 2 – Cumulative variance explained by first 10 Principal Component

We can see that first 3 Principal Components (PC) explain almost all 99% of the variance in the data. Specifically, the first PC explains 93.8%, the second PC explains 5.8% and the third PC explains 1.8% of the variance in the data respectively. Thus, we will take the first 3 and discard the rest. All the subsequent analysis with be done with the first 3 Principal Components.

Let us now see how the PCs evolve over time. Fig. 4 plots the time series of PCs.

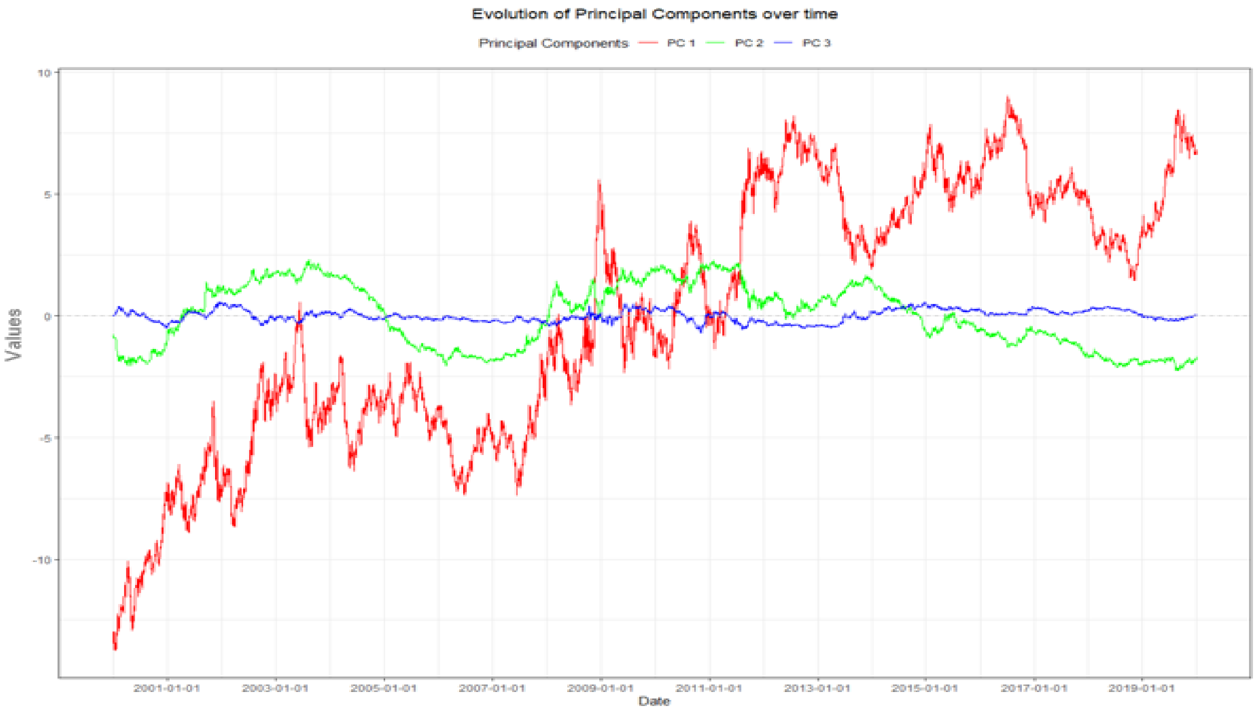


Fig. 4: Time series of 3 Principal Components for US Treasuries

The first PC (red line) which explains around 94% of the variance in the yields data shows a clear upward trend from 2000 to 2020. On closer observation, we can clearly see two different regimes in the PC 1 time series. Between 2002 and 2008, the time series stays below 0. Then, there is a transition phase between 2008 and 2011. After 2011, PC 1 stays above 0 and oscillates there. If we correlate PC 1 with the actual yields, we can say in periods where yield was high (early 2000s), PC 1 was negative and then it turned positive in a low yield environment. Also, although negative yields are not uncommon these days, but most economists believe that deep negative yields are impossible due to government intervention. Therefore, the upside of PC 1 may be capped. This means if PC 1 is extrapolated upwards with the long term trend, it may take the yields to impossible territories. This may suggest that the long term trend of PC 1 will not continue in the near future.

The second PC (green line) is somewhat oscillatory with no clear trend. However, the period of oscillation is different in different windows. The third PC has much less magnitudes than the other 2 PCs and is also oscillatory in nature with no clear trend.

Since, the 3 PCs together account for 99% of the total variance in the data, if we were to recreate the yield curves using PCs, it should fit well on the actual yields. To demonstrate this, we take 4 yield curves from randomly chosen dates within our data. We plot the yield curves and then overlay the derived yield curves from PCs over them in Fig. 5.

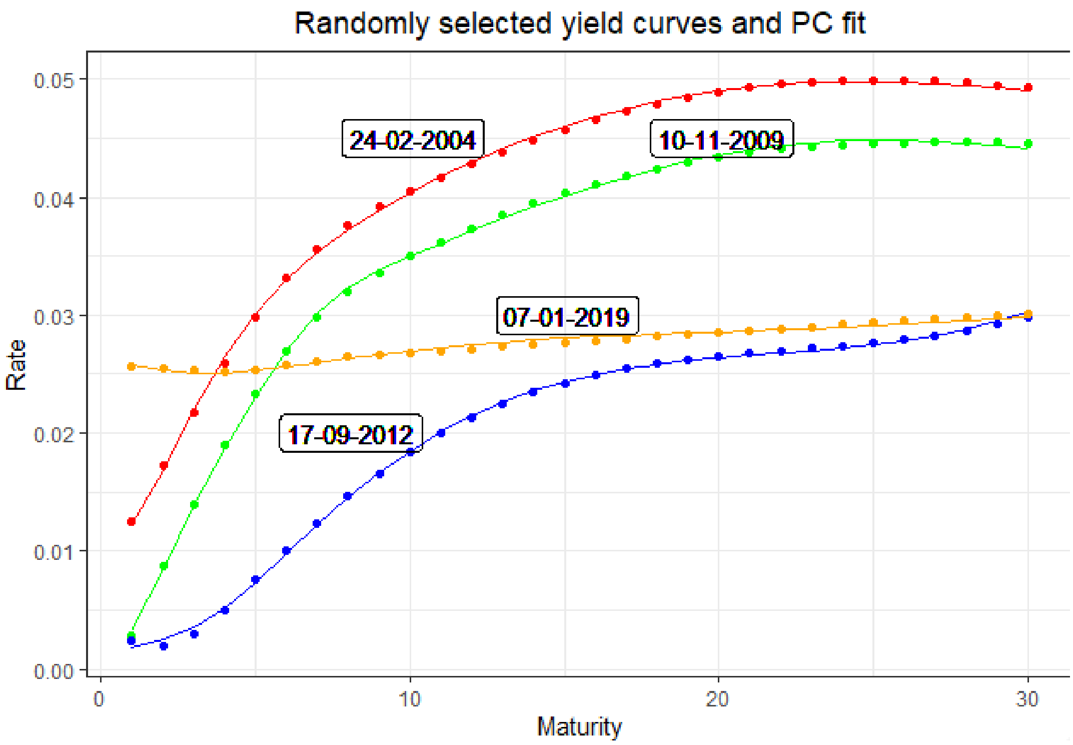


Fig. 5: Yield curves and PCs fit

We can see it is a near perfect fit. This proves that very little information is lost by choosing first 3 PCs instead of all 30.

PCs by definition are orthogonal to each other. This means there should uncorrelated to each other. This is demonstrated in Fig. 6. This also means that any recombination of the PCs can be done linearly without any fear of multicollinearity.

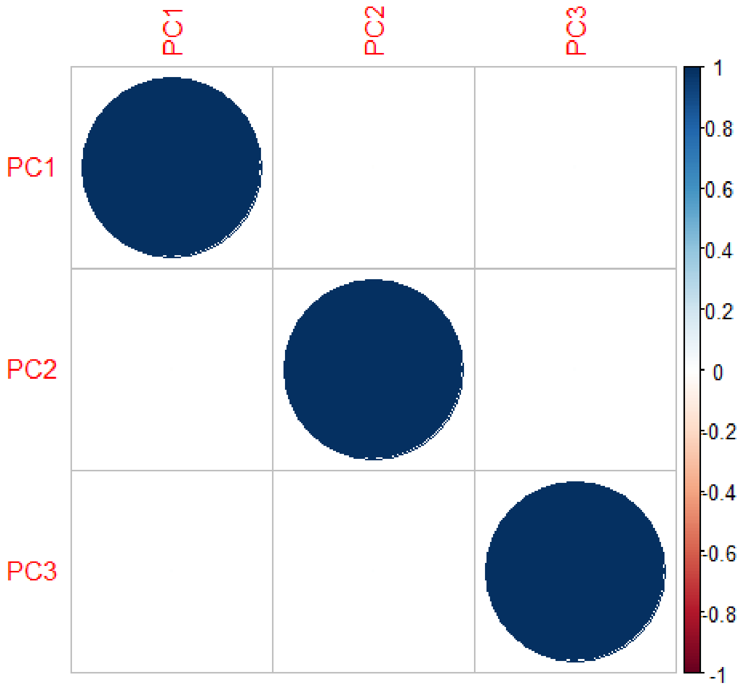
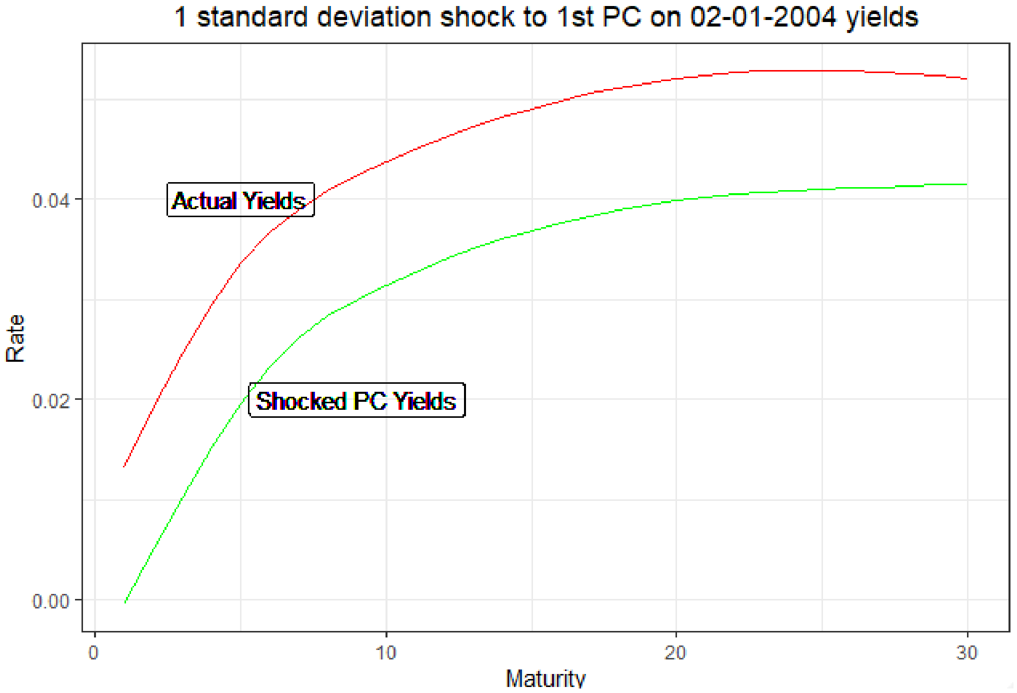
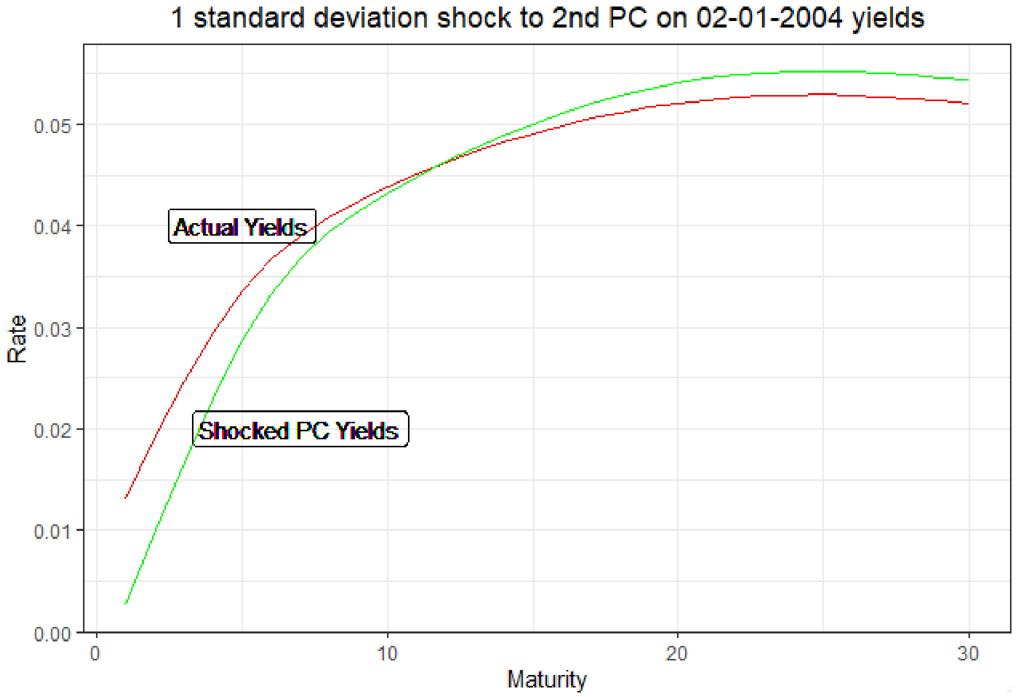


Fig 6: Correlation plot of the 3 PCs

Next, we see how the three PCs actually influence the yield curves. For that, we again select a yield curve on a random date. The value of 3 PCs on this date is then taken. Keeping, the other two PCs constant, we shock one PC by +1 standard deviation of the PC. We calculate the yield curve from this new set of 1 shocked and 2 unchanged PC. We plot this yield curve vis-à-vis the original yield curve and see the effect of shocking 1 PC has on it. We repeat this exercise for all 3 PCs.





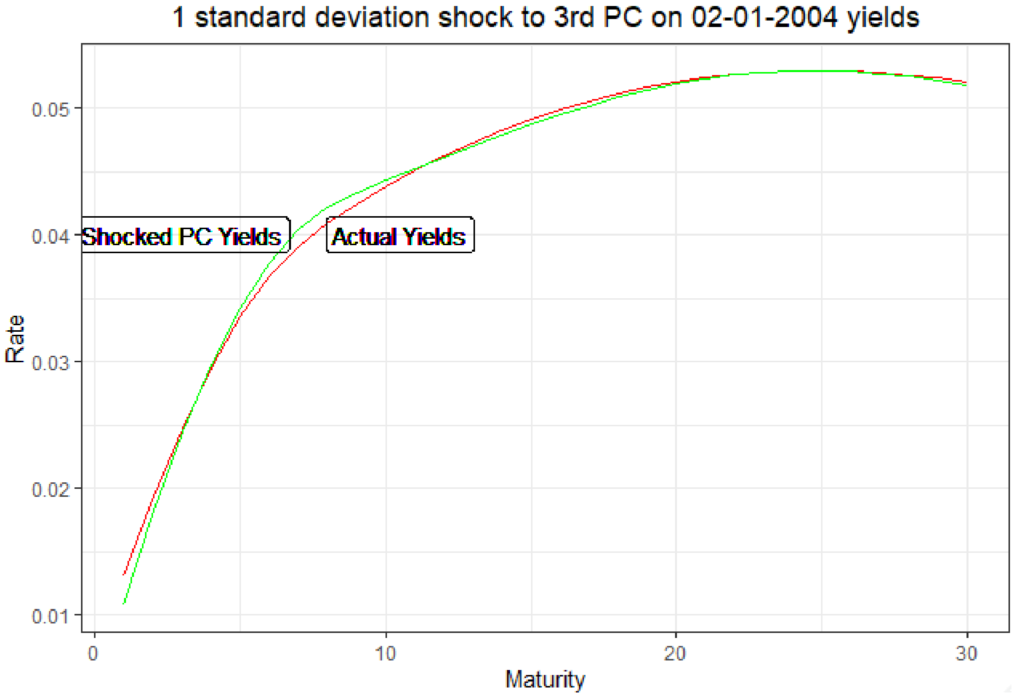


Fig 6: Effect of shocking PCs on a yield curve

From Fig. 6, we see that the first PC has the effect of almost a parallel shift in the level of the yield curve. This PC is thus usually referred as Level. We thus note that 93.8% of yield curve movement is parallel shift. The second PC has the effect of determining the slope of the yield curve. It has a tendency to twist the curve at around the 12th maturity and thus make the yield curve more or less steep. It is referred to as Slope. Also, around 5.8% of yield curve movements is changing its slope. The third PC determines how humped or flat the yield curve will be. It is called Curvature. 1.87% of yield curve movement is thus changing its curvature.

Having analyzed the PCs, we now a look at factor loadings. The factor loading when multiplied with the PCs gives back the yield curves. As mentioned before, the factor loadings are time invariant. The assumption is that the functional relation of principal components, which evolve over time and the yield curves remain constant. We will test this assumption by dividing our period under study (2000 – 2019) into two windows: 2000 – 2008 and 2009 – 2020. We will try to explore whether there is any change in the factor loadings in pre and post crisis regimes. We will do that by visual inspection and not depend on any statistical test, although that may be the logical next step.

Fig. 7 shows the eigenvalues of the respective data matrices of two part-windows and one full window. We see that although directionally there is not much dissimilarity, the second eigenvalue for the 3 windows are distant from each other. Hence, this inspection doesn't assure us that the functional relationship is period agnostic because the roots of the equations are somewhat different. Fig. 8 shows cumulative variance explained by different PCs for 3 different windows. We notice that for the window 2009-2019, the variance explained by first PC is a lot less than that explained by the first PC in the other two windows. This suggests that the effect of parallel shift on yield curve movements has reduced post crisis. However, the cumulative variance explained by first and second PCs are nearly coincident for all 3 windows. Thus, post crisis, the decrease in effect of parallel shift is taken up by increase in effect of slope. The effect of third PC is more or less constant on all the three windows and thus, the effect of curvature on yield curves has not gone through major changes.

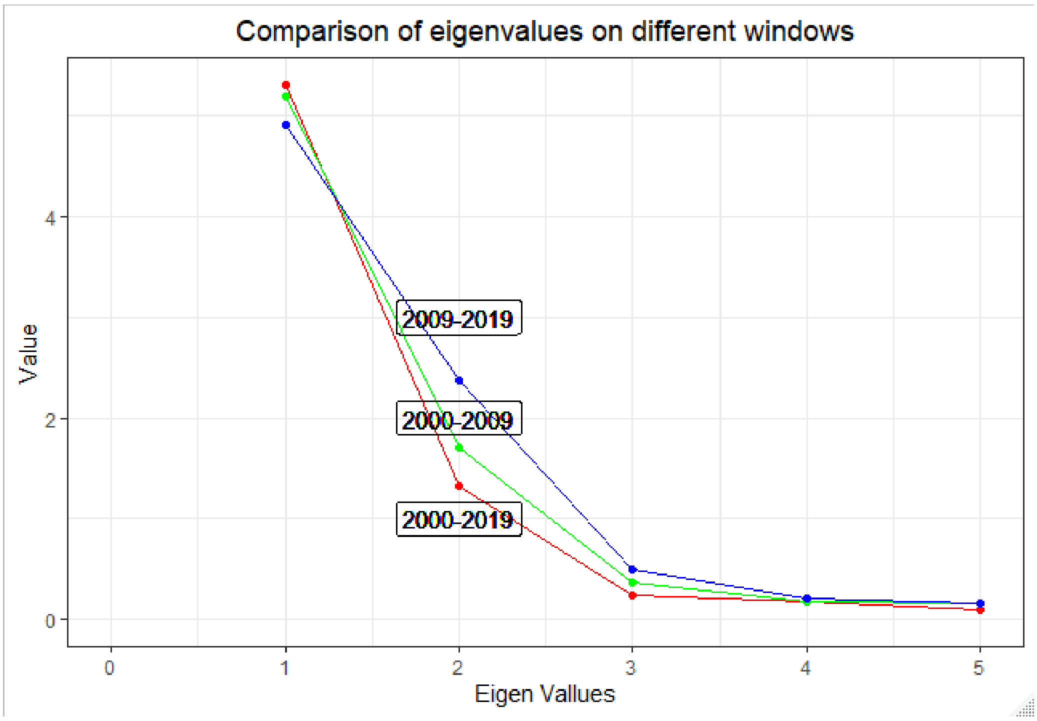


Fig. 7: Comparison of eigenvalues of data matrices from 3 different windows

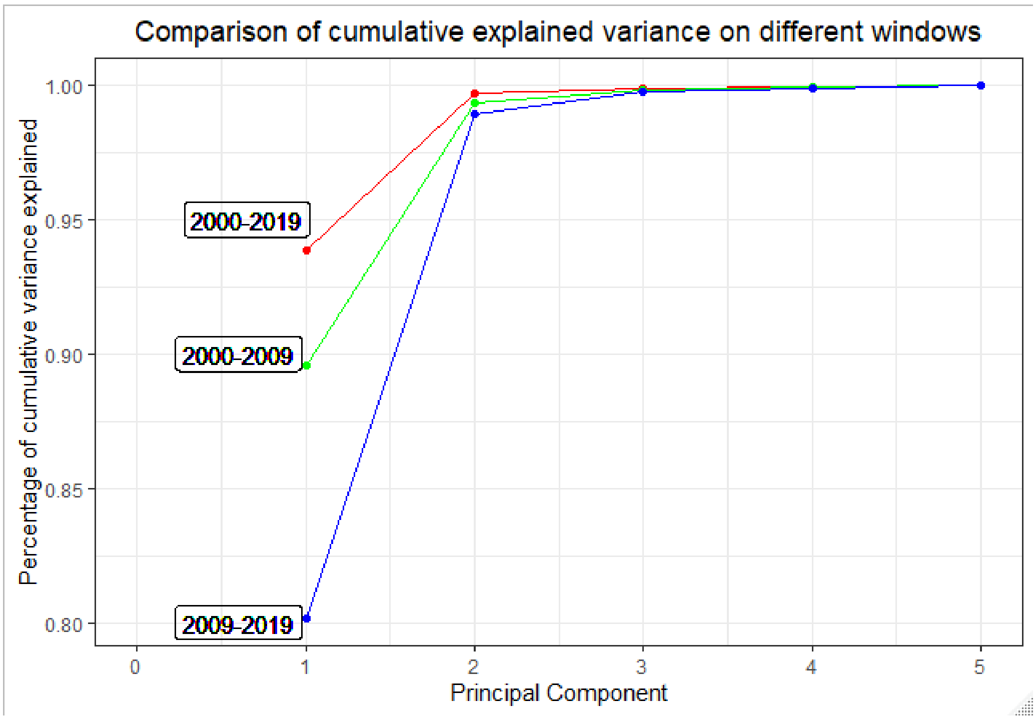


Fig. 8: Comparison of cumulative explained variance on different windows

Factor loadings are also called principal directions. The idea is that the original data undergoes a transformation of axes. The PCs represent the projection of the original data on the transformed axes and the factor loading represent the direction of the axes. Thus, if the data structure does not change, the principal directions should not change as well. We test this assumption visually by plotting factor loading i.e. principal directions for the 3 periods.

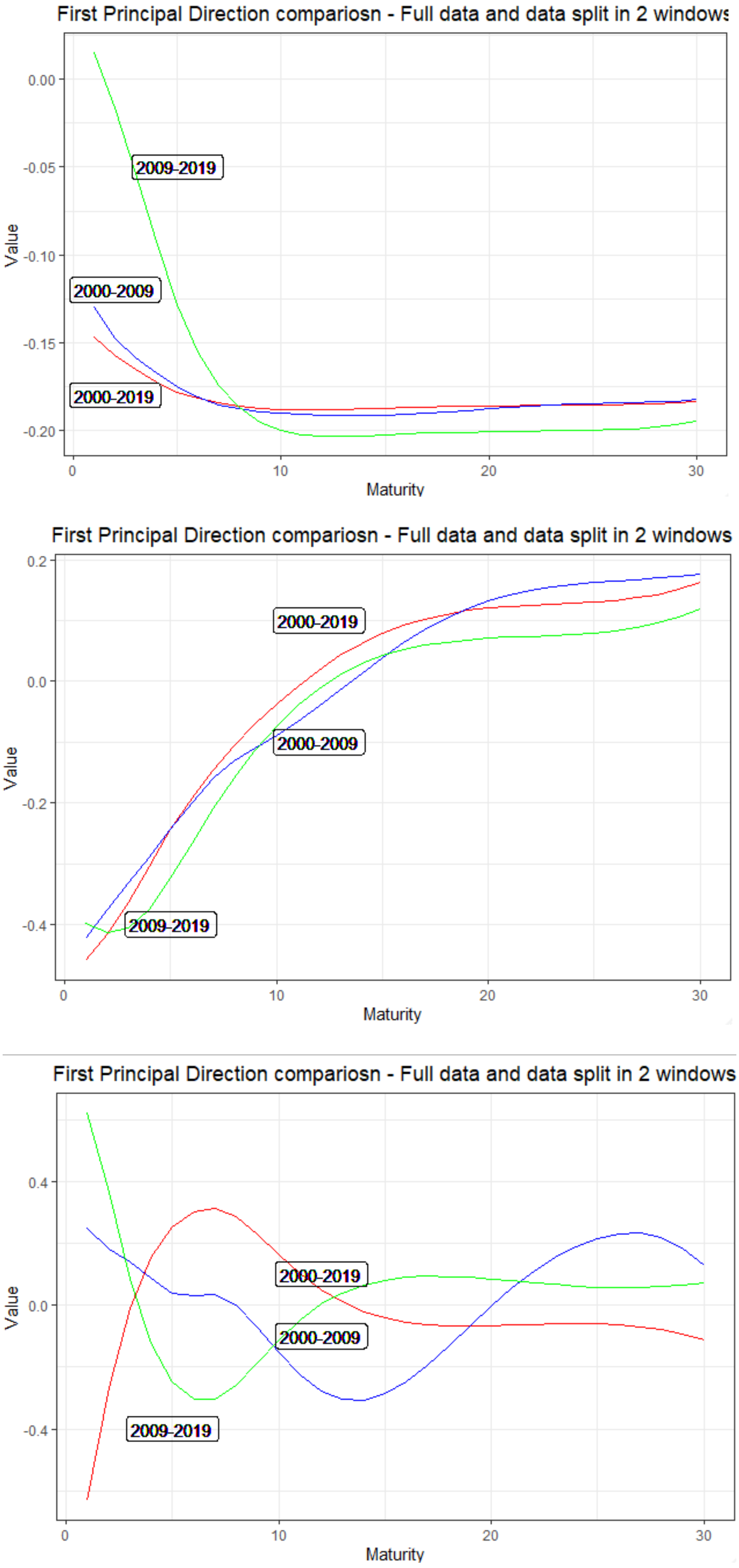


Fig. 9: Principal directions in 3 different windows

We see that there is a clear discrepancy in the first principal direction pre and post crisis, especially for the first few maturities. The discrepancy is less for the longer maturities. For the first maturity, even the signs are different. The second principal direction is well behaved across the windows. They tend to move together across maturities and are not far apart. The third principal direction also seems to be different for different windows. Even here, the signs of the first maturity are opposite. The relationship improves for the longer maturities like in the case of the first principal direction.

Next, we see if there is any difference in the evolution of PCs. Fig. 10 plots the PCs for the full data and the windowed data together,

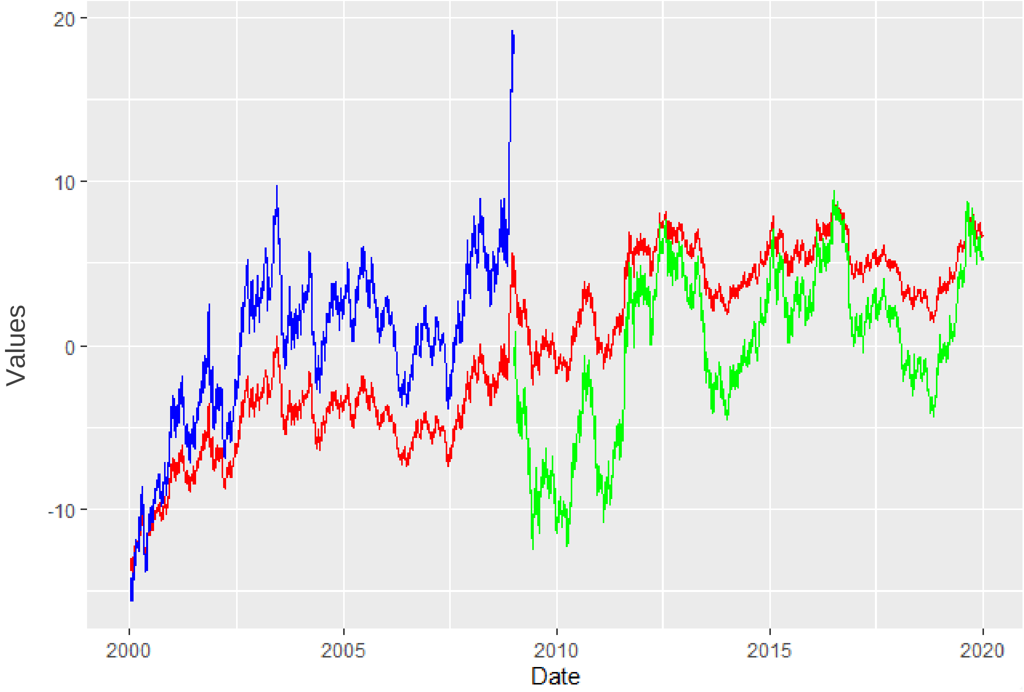


Fig. 10a: First PC for the three windows

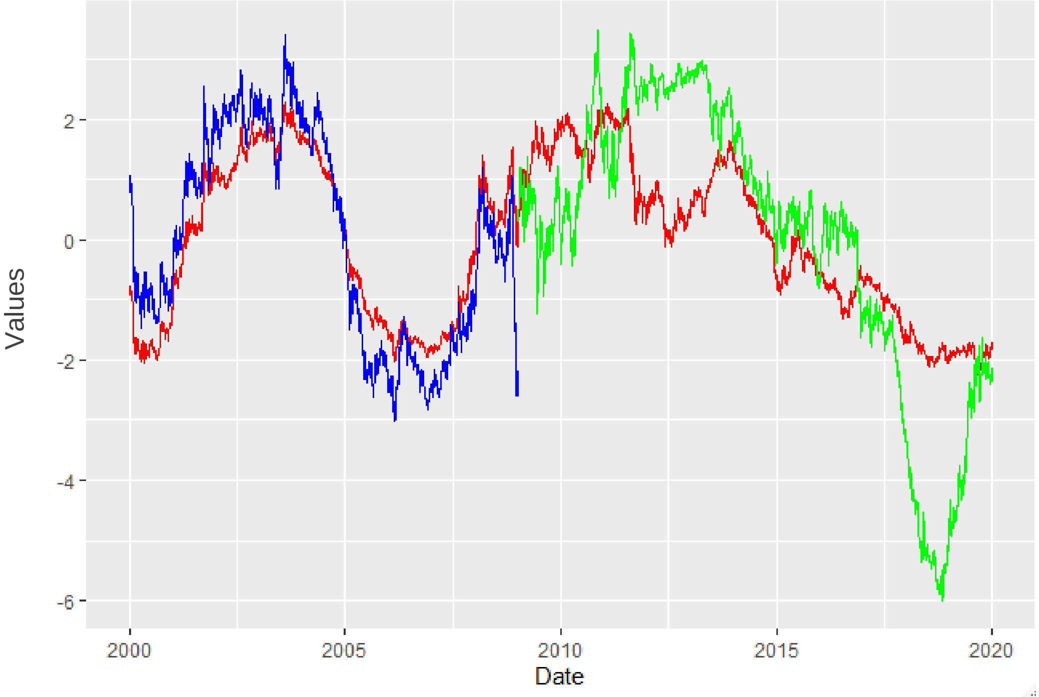


Fig. 10b: Second PC for the three windows

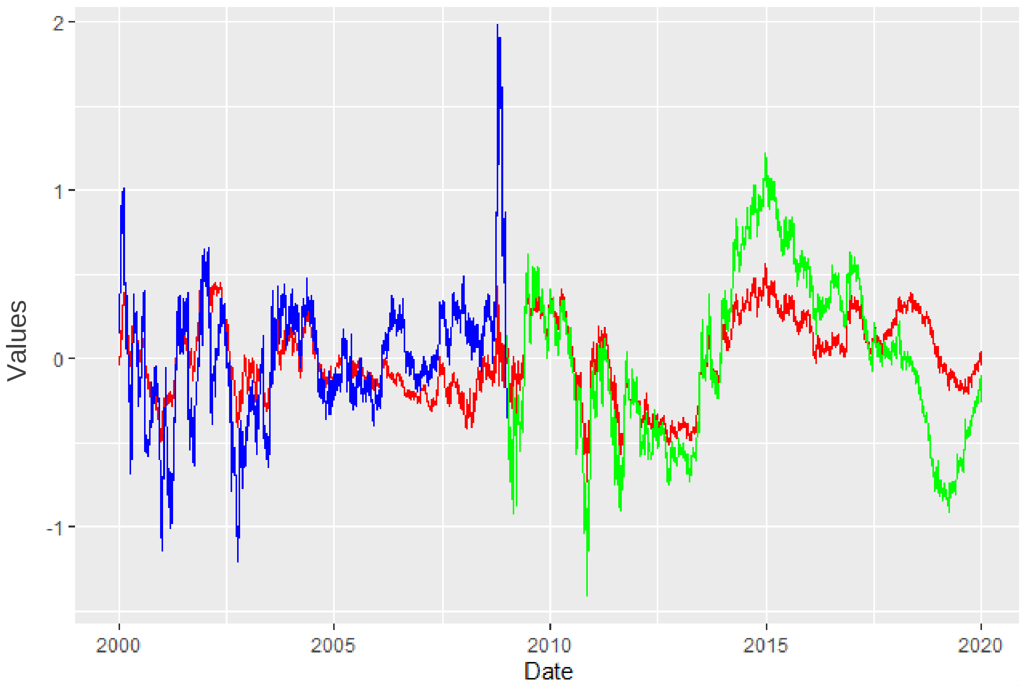


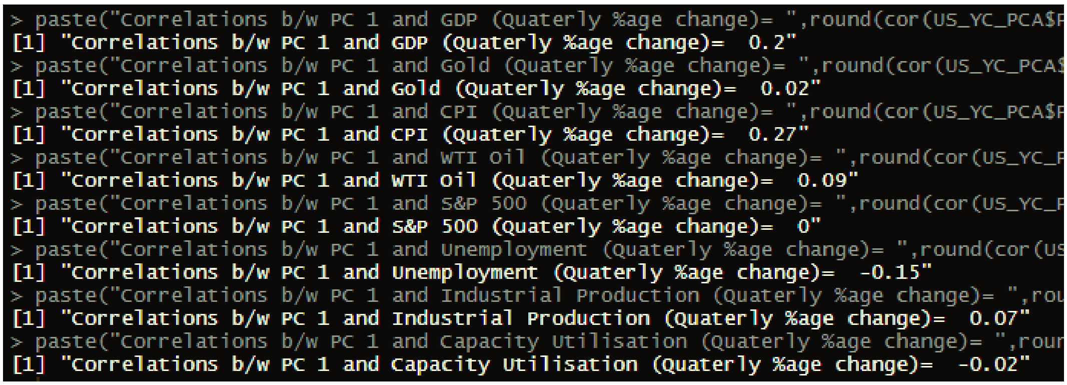
Fig. 10c: Third PC for the three windows

We see that for all the three PCs, movement in the all the three widows is either parallel or conjoint. A few of the peaks and troughs are much more pronounced in the windowed PCs than that in the PC for full data. Given that the windowed PCs when multiplied by its factor loadings and the PCs for the full data multiplied by the factor loading reproduce the same yield curves, we may suspect that the differences in factor loadings and evolution of PCs is due scaling differences rather than structural differences. An interesting study would be transform the factor loading and PC matrix by the same scaling factor and choose the factor in such a way that the squared error between the windowed PCs and full data PCs become minimum. If discrepancies is solely due to scaling, applying inverse of the scaling factor on the factor loadings should produce the same or very nearly same factor loadings for both windowed and full data. However, this would still not explain the reversal in signs.

5. Effect of Macro-economic variables on Principal Components of US Treasuries

In the last section, we saw that 3 PCs are sufficient to explain most of the variation in yield curves. The 3 PCs namely Level, Slope and Curvature has a unique effect on the shape of the yield curve. Now, the PCs are derived by a mathematical process and as such have no economic value attached to them. They are not observable in the real world. Hence, in this section we try to regress some economic indicators against the PCs to see whether we can describe yield curve movements through them.

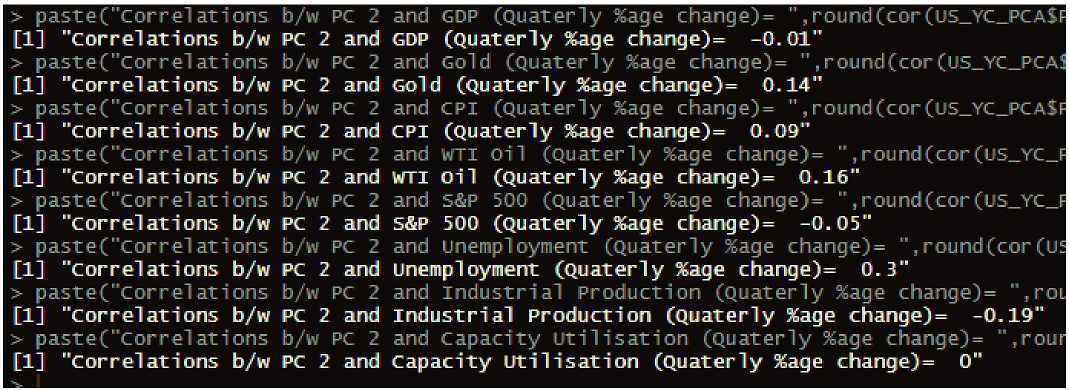
Since some of the major indicators are recorded quarterly, we calculate the percentage quarterly change of all the variables. Our period under consideration is 2000 to 2019. Thus, we have 80 quarters or 80 data points. The difference in frequency of recording of different variables poses a problem because yield curves are recorded daily and as such there will be short term fluctuations that won't be captured using quarterly change. Assuming that we are only bothered about long term fluctuations, we have the following variables that we calculate quarterly percentage of: PC 1, PC 2, PC 3, USA GDP, USA CPI, S&P 500, Gold (in Dollars), USA Unemployment Rate, USA Industrial Production, WTI Oil Price and USA Capacity Utilization. Since, 1st and 2nd PC explain 98% of the variation, any major change in the yields must be through these 2 PCs. First, we will list out all the individual correlations and then plot each indicator against the PCs to have a visual inspection.



```
> paste("Correlations b/w PC 1 and GDP (Quaterly %age change)= ",round(cor(US_YC_PCA$F
[1] "Correlations b/w PC 1 and GDP (Quaterly %age change)=  0.2"
> paste("Correlations b/w PC 1 and Gold (Quaterly %age change)= ",round(cor(US_YC_PCA$
[1] "Correlations b/w PC 1 and Gold (Quaterly %age change)=  0.02"
> paste("Correlations b/w PC 1 and CPI (Quaterly %age change)= ",round(cor(US_YC_PCA$F
[1] "Correlations b/w PC 1 and CPI (Quaterly %age change)=  0.27"
> paste("Correlations b/w PC 1 and WTI Oil (Quaterly %age change)= ",round(cor(US_YC_F
[1] "Correlations b/w PC 1 and WTI Oil (Quaterly %age change)=  0.09"
> paste("Correlations b/w PC 1 and S&P 500 (Quaterly %age change)= ",round(cor(US_YC_F
[1] "Correlations b/w PC 1 and S&P 500 (Quaterly %age change)=  0"
> paste("Correlations b/w PC 1 and Unemployment (Quaterly %age change)= ",round(cor(US
[1] "Correlations b/w PC 1 and Unemployment (Quaterly %age change)=  -0.15"
> paste("Correlations b/w PC 1 and Industrial Production (Quaterly %age change)= ",rou
[1] "Correlations b/w PC 1 and Industrial Production (Quaterly %age change)=  0.07"
> paste("Correlations b/w PC 1 and Capacity Utilisation (Quaterly %age change)= ",rour
[1] "Correlations b/w PC 1 and Capacity Utilisation (Quaterly %age change)=  -0.02"
```

Fig 11: Output of correlation between the 1st PC and variables

CPI and GDP are the only 2 variable who have a significantly positive correlation with PC 1. All the other variables have correlation that can be termed as insignificant.

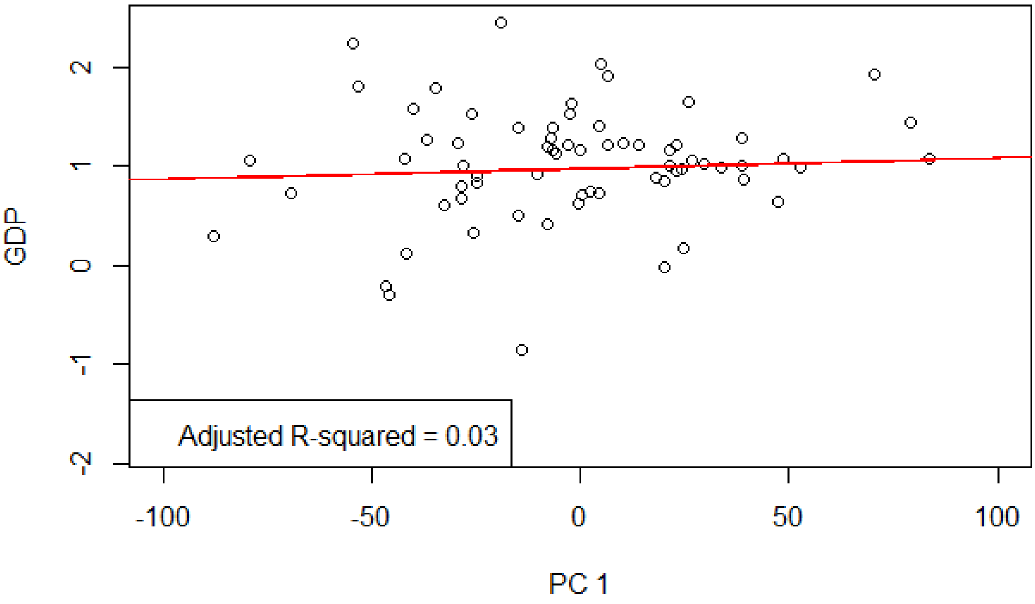


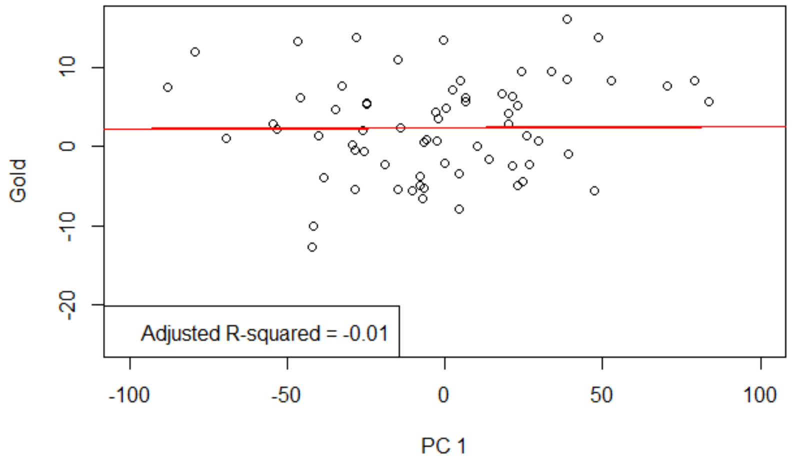
```
> paste("Correlations b/w PC 2 and GDP (Quaterly %age change)= ",round(cor(US_YC_PCA$F
[1] "Correlations b/w PC 2 and GDP (Quaterly %age change)=  -0.01"
> paste("Correlations b/w PC 2 and Gold (Quaterly %age change)= ",round(cor(US_YC_PCA$
[1] "Correlations b/w PC 2 and Gold (Quaterly %age change)=  0.14"
> paste("Correlations b/w PC 2 and CPI (Quaterly %age change)= ",round(cor(US_YC_PCA$F
[1] "Correlations b/w PC 2 and CPI (Quaterly %age change)=  0.09"
> paste("Correlations b/w PC 2 and WTI Oil (Quaterly %age change)= ",round(cor(US_YC_F
[1] "Correlations b/w PC 2 and WTI Oil (Quaterly %age change)=  0.16"
> paste("Correlations b/w PC 2 and S&P 500 (Quaterly %age change)= ",round(cor(US_YC_F
[1] "Correlations b/w PC 2 and S&P 500 (Quaterly %age change)=  -0.05"
> paste("Correlations b/w PC 2 and Unemployment (Quaterly %age change)= ",round(cor(US
[1] "Correlations b/w PC 2 and Unemployment (Quaterly %age change)=  0.3"
> paste("Correlations b/w PC 2 and Industrial Production (Quaterly %age change)= ",rou
[1] "Correlations b/w PC 2 and Industrial Production (Quaterly %age change)=  -0.19"
> paste("Correlations b/w PC 2 and Capacity Utilisation (Quaterly %age change)= ",rour
[1] "Correlations b/w PC 2 and Capacity Utilisation (Quaterly %age change)=  0"
>
```

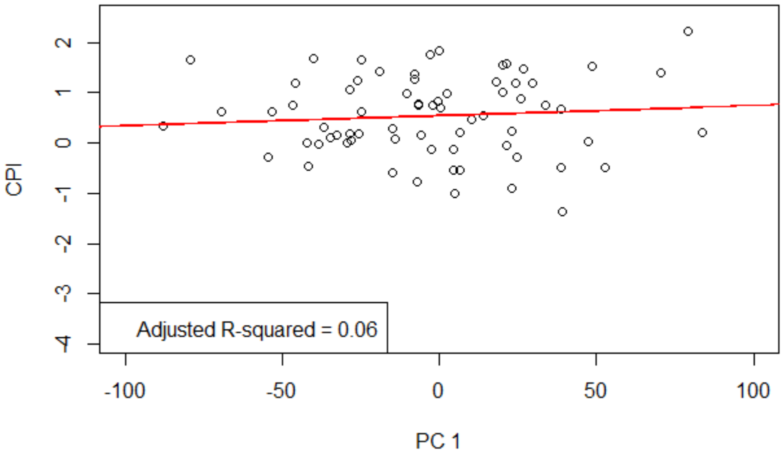
Fig 12: Output of correlation between the 2nd PC and variables

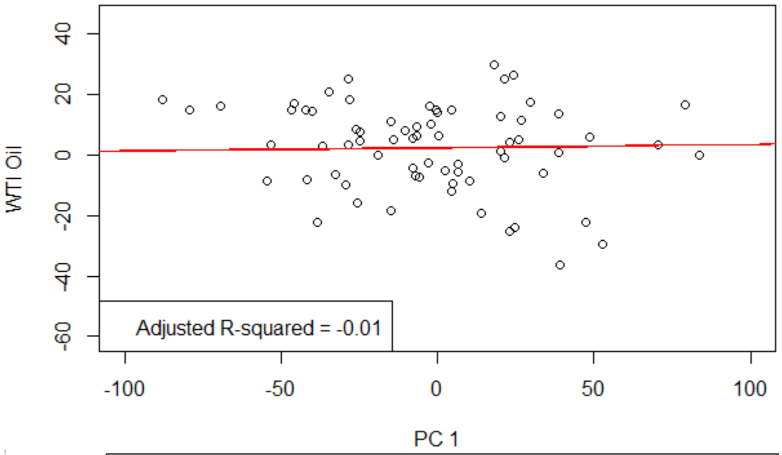
Unemployment Rate and Industrial Production has slightly significant correlation with PC 2. All other variables have little or no correlation.

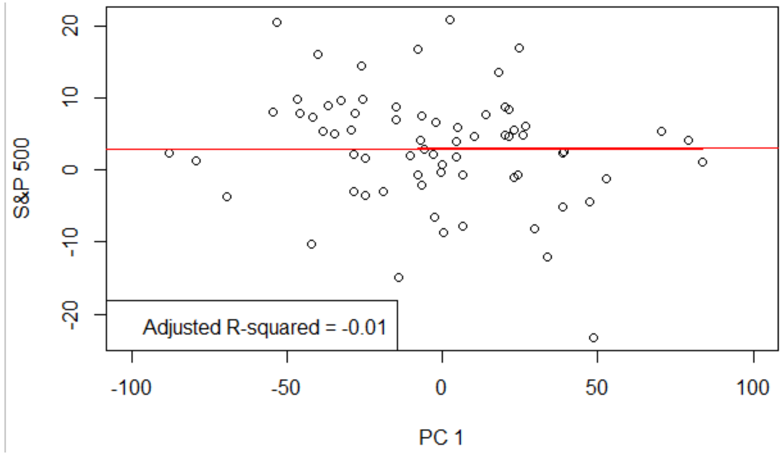
We will now plot each variable against PCs to see if there is any non-linear or higher order relationship




Gold
10
0
-10
-20
Adjusted R-squared = -0.01
-100
-50
0
50
100
PC 1


CPI
2
1
0
-1
-2
-3
-4
Adjusted R-squared = 0.06
-100
-50
0
50
100
PC 1


WTI Oil
40
20
0
-20
-40
-60
Adjusted R-squared = -0.01
-100
-50
0
50
100
PC 1


S&P 500
20
10
0
-10
-20
Adjusted R-squared = -0.01
-100
-50
0
50
100
PC 1

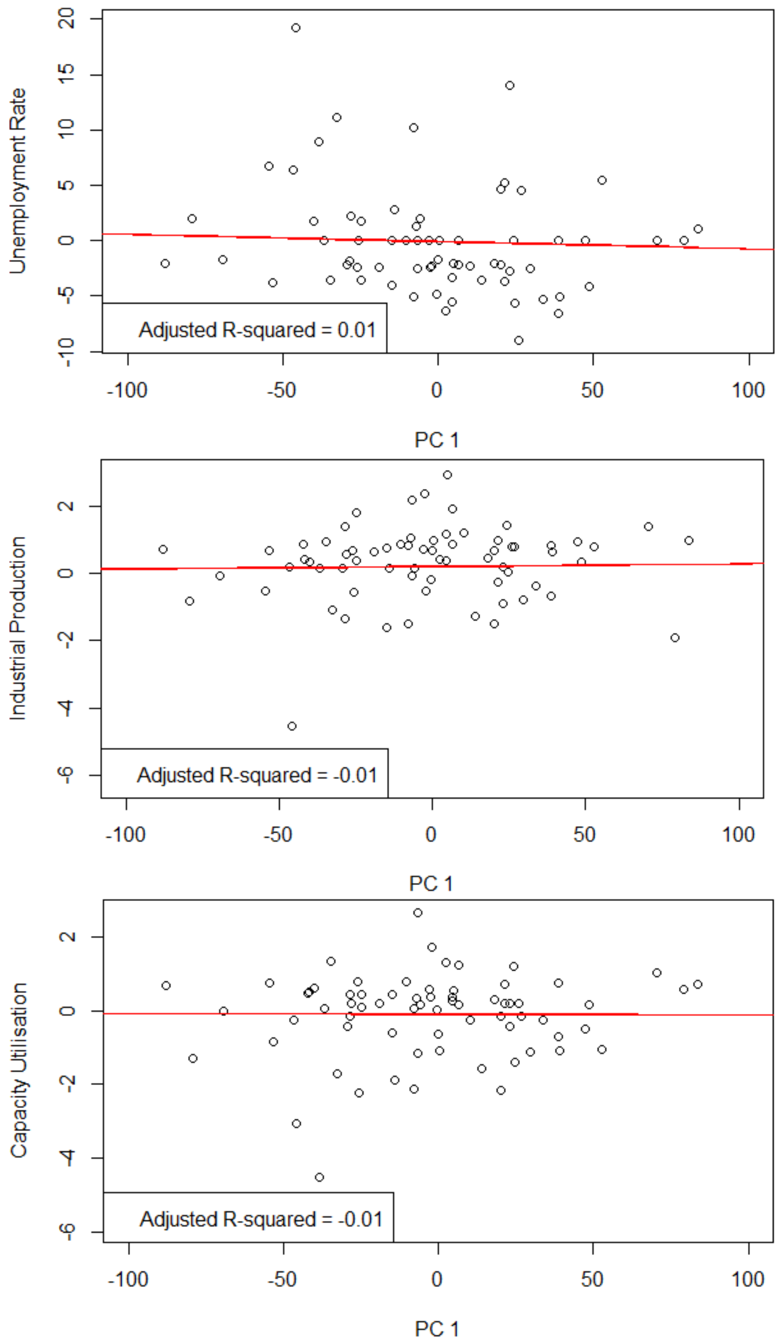
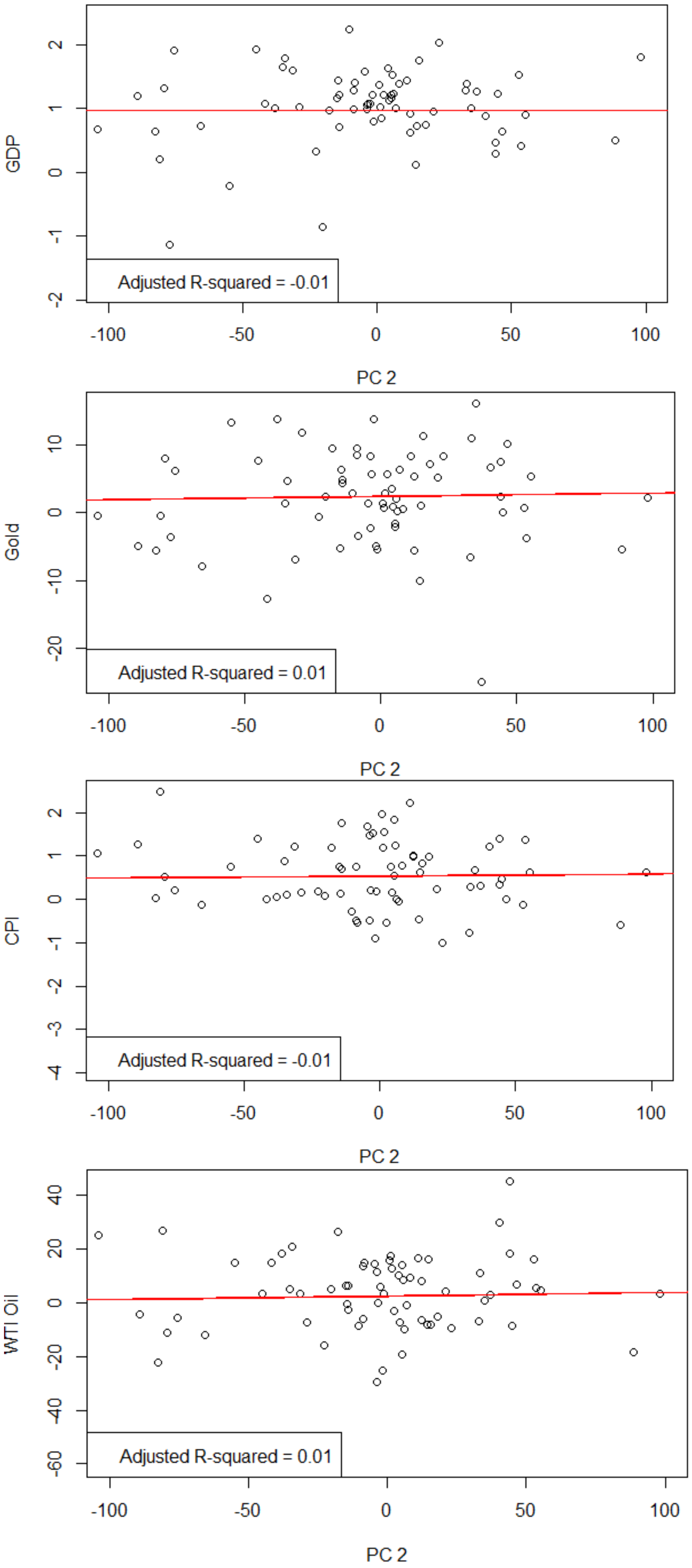


Fig 13: Variables plotted against PC 1

From these plots it is fair to conclude that no variable has a strong predictive relationship with PC 1.


GDP
2
1
0
-1
-2
Adjusted R-squared = -0.01
-100
-50
0
50
100
PC 2
Gold
10
0
-10
-20
Adjusted R-squared = 0.01
-100
-50
0
50
100
PC 2
CPI
2
1
0
-1
-2
-3
-4
Adjusted R-squared = -0.01
-100
-50
0
50
100
PC 2
WTI Oil
40
20
0
-20
-40
-60
Adjusted R-squared = 0.01
-100
-50
0
50
100
PC 2

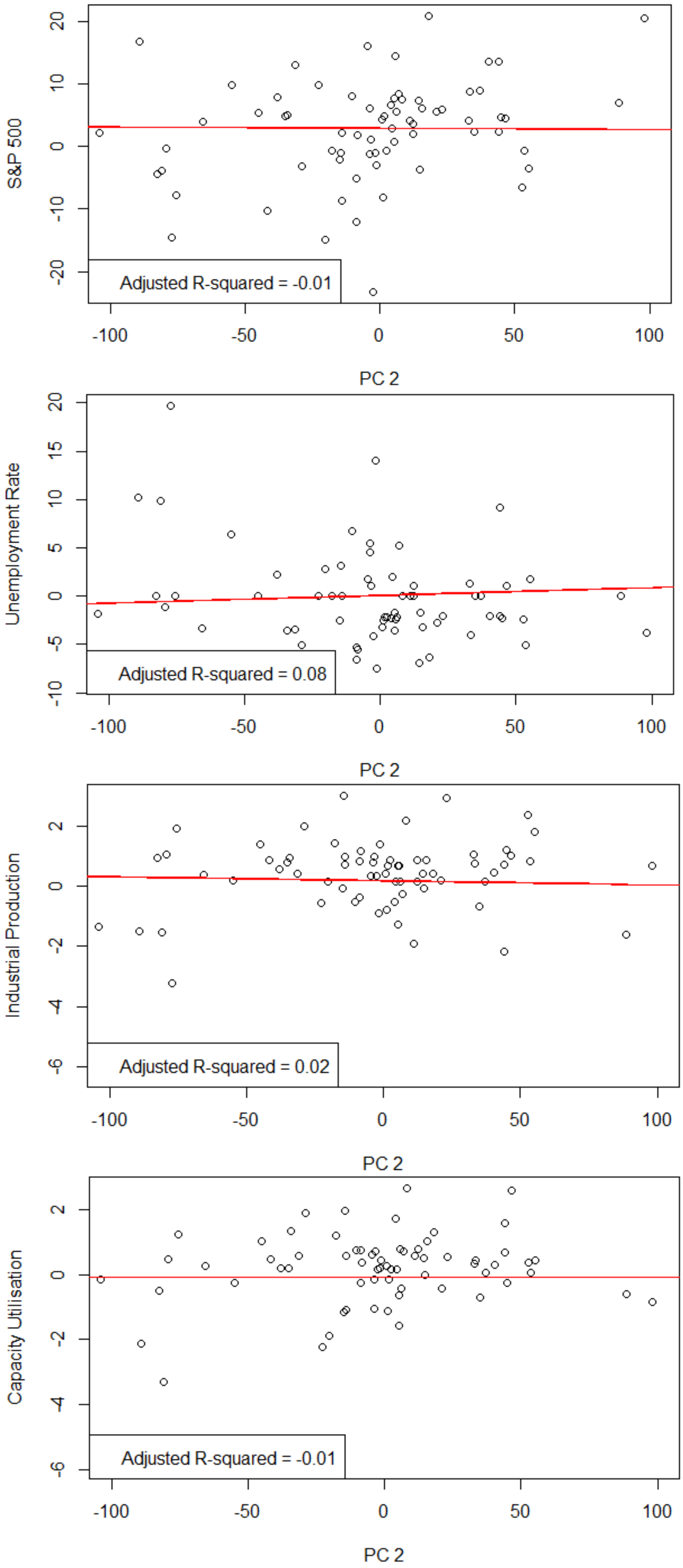
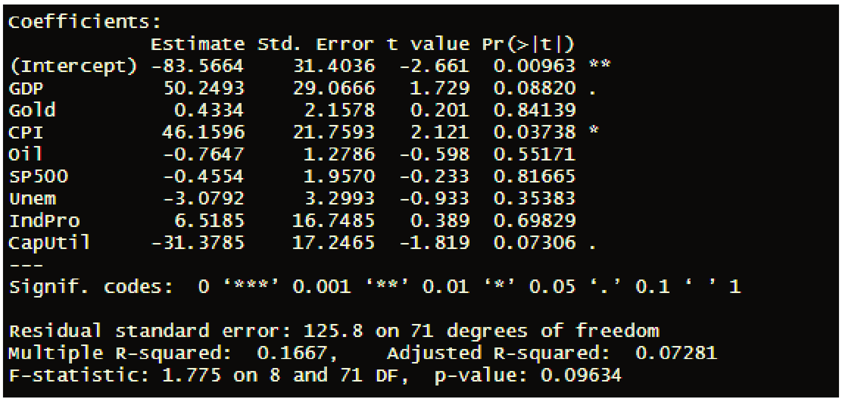


Fig 14: Variables plotted against PC 2

We may also conclude from Fig. 14 that no variable has a strong predictive relationship with PC 2.

Now, we try to do a regression analysis on the PCs. Having found no bivariate relationship, regression will tell us whether there is a multivariate relationship between the macro-economic variables and the PCs.

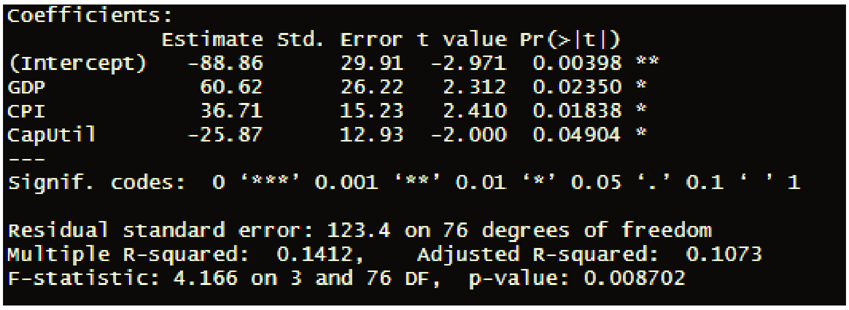


```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -83.5664    31.4036  -2.661  0.00963 **
GDP          50.2493    29.0666   1.729  0.08820 .
Gold          0.4334     2.1578   0.201  0.84139
CPI          46.1596    21.7593   2.121  0.03738 *
Oil          -0.7647     1.2786  -0.598  0.55171
SP500        -0.4554     1.9570  -0.233  0.81665
Unem         -3.0792     3.2993  -0.933  0.35383
IndPro        6.5185    16.7485   0.389  0.69829
CapUtil     -31.3785    17.2465  -1.819  0.07306 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 125.8 on 71 degrees of freedom
Multiple R-squared:  0.1667,    Adjusted R-squared:  0.07281
F-statistic: 1.775 on 8 and 71 DF,  p-value: 0.09634
```

Fig 15: Regression of all variables against PC 1

Fig. 15 shows that a few variables namely GDP, CPI and Capacity Utilization have significant coefficients when regressed against PC 1. However, the model-fit metric Adjusted R-squared of 0.07 suggests that the model is performing very poorly to predict PC 1 with these variables. We will try and remove the insignificant variables to see if the results are better.

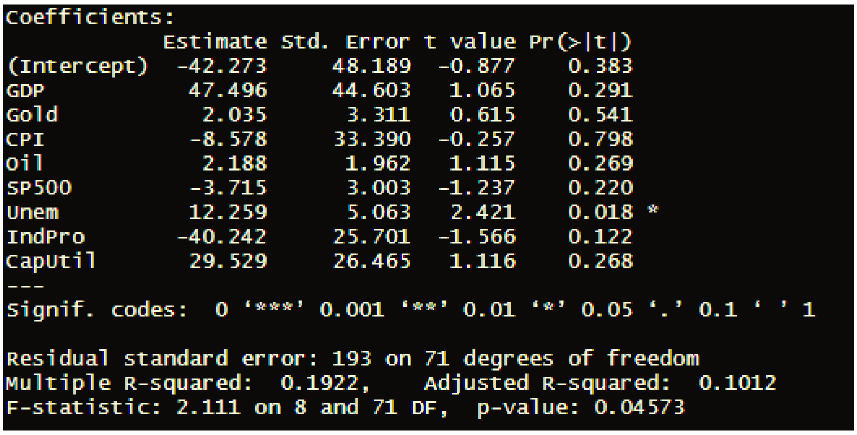


```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   -88.86      29.91  -2.971  0.00398 **
GDP            60.62      26.22   2.312  0.02350 *
CPI            36.71      15.23   2.410  0.01838 *
CapUtil       -25.87      12.93  -2.000  0.04904 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 123.4 on 76 degrees of freedom
Multiple R-squared:  0.1412,    Adjusted R-squared:  0.1073
F-statistic: 4.166 on 3 and 76 DF,  p-value: 0.008702
```

Fig 16: Regression of significant variables against PC 1

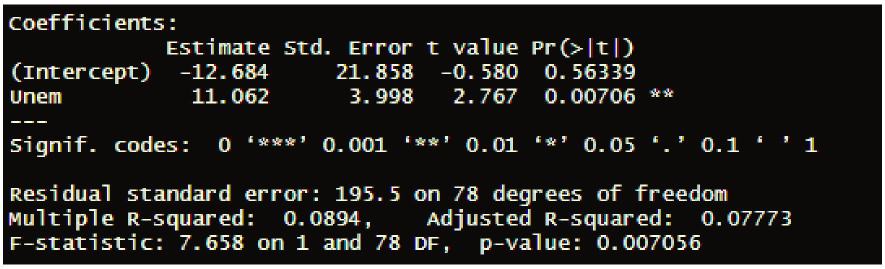
Fig. 16 shows that although the coefficients are significant, the model is still performing poorly. The Adjusted R-squared improvement is not enough to ascertain good predictive power to these variables.



```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -42.273     48.189  -0.877    0.383
GDP           47.496     44.603   1.065    0.291
Gold           2.035      3.311   0.615    0.541
CPI           -8.578     33.390  -0.257    0.798
Oil            2.188      1.962   1.115    0.269
SP500         -3.715      3.003  -1.237    0.220
Unem          12.259      5.063   2.421    0.018 *
IndPro       -40.242     25.701  -1.566    0.122
CapUtil       29.529     26.465   1.116    0.268
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 193 on 71 degrees of freedom
Multiple R-squared:  0.1922,    Adjusted R-squared:  0.1012
F-statistic: 2.111 on 8 and 71 DF,  p-value: 0.04573
```

Fig 17: Regression of all variables against PC 2



```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)   
(Intercept)  -12.684     21.858  -0.580  0.56339   
Unem          11.062      3.998   2.767  0.00706 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 195.5 on 78 degrees of freedom
Multiple R-squared:  0.0894,	Adjusted R-squared:  0.07773 
F-statistic: 7.658 on 1 and 78 DF,  p-value: 0.007056
```

Fig 18: Regression of significant variable against PC 2

Fig 17 and Fig 18 also shows that model-fit is not good enough to ascertain predictive power to the variables for PC 2.

Thus, we are not successful in showing that the economic indicators have any significant relationship with the PCs. However, it may interesting to see whether the PCs are lag or lead indicators for macro-economic variables. Studies often show that market movements precede macro-economic events and developments. The yield curves are considered by many analysts as a proxy for future state of economy and inflation regimes. We keep that as a possible next step of this research.

6. Principal Components Correlation Analysis of Yield Curves from USA, Canada, Switzerland and Japan

In this section we analyze PCs of yield curves from four developed economies – USA, Canada, Switzerland and Japan. The procedure to extract PCs remain the same. We take daily yield curves of all the four countries from January 2005 to December 2019. This arbitrary choice of period is to ensure that we have data with as less missing values as possible in all four countries. Then as before, we fit a cubic spline to interpolate all maturities from 1 to 30. Having done that, we apply PCA on the yields. Fig. 19 demonstrates the explained variance of the first 10 PCs in the four countries.

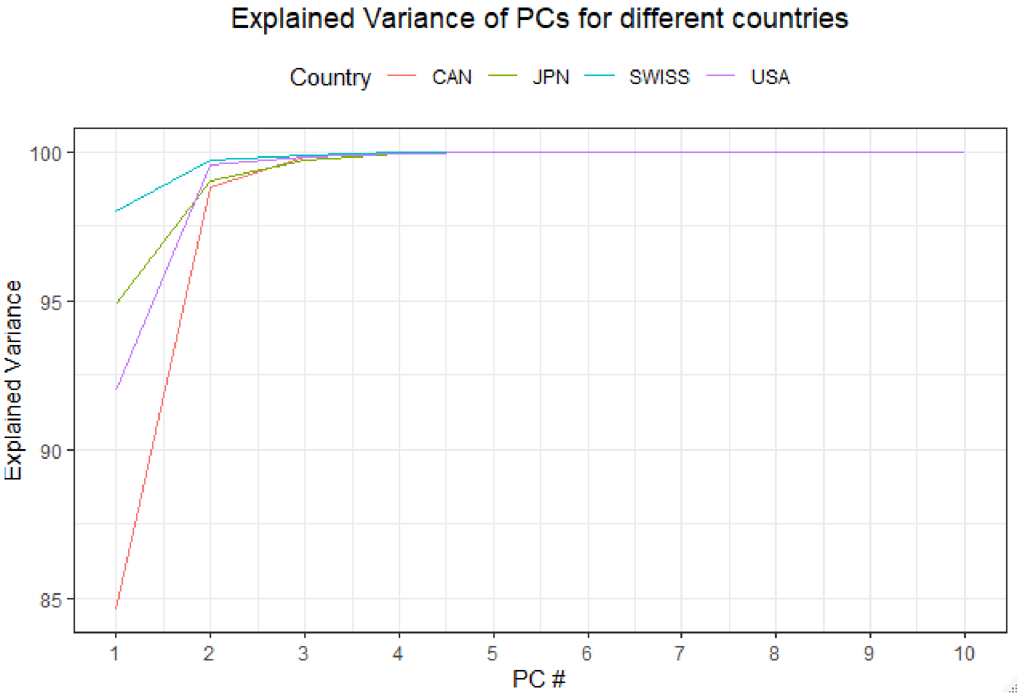
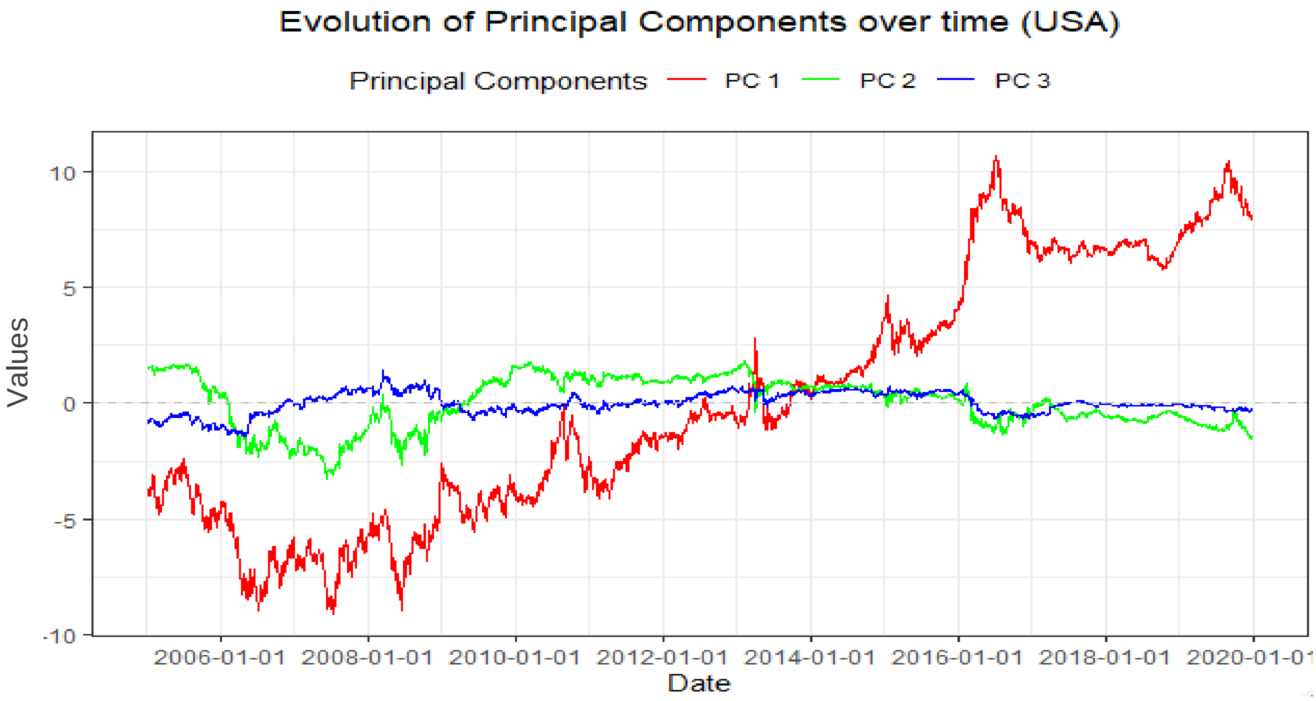
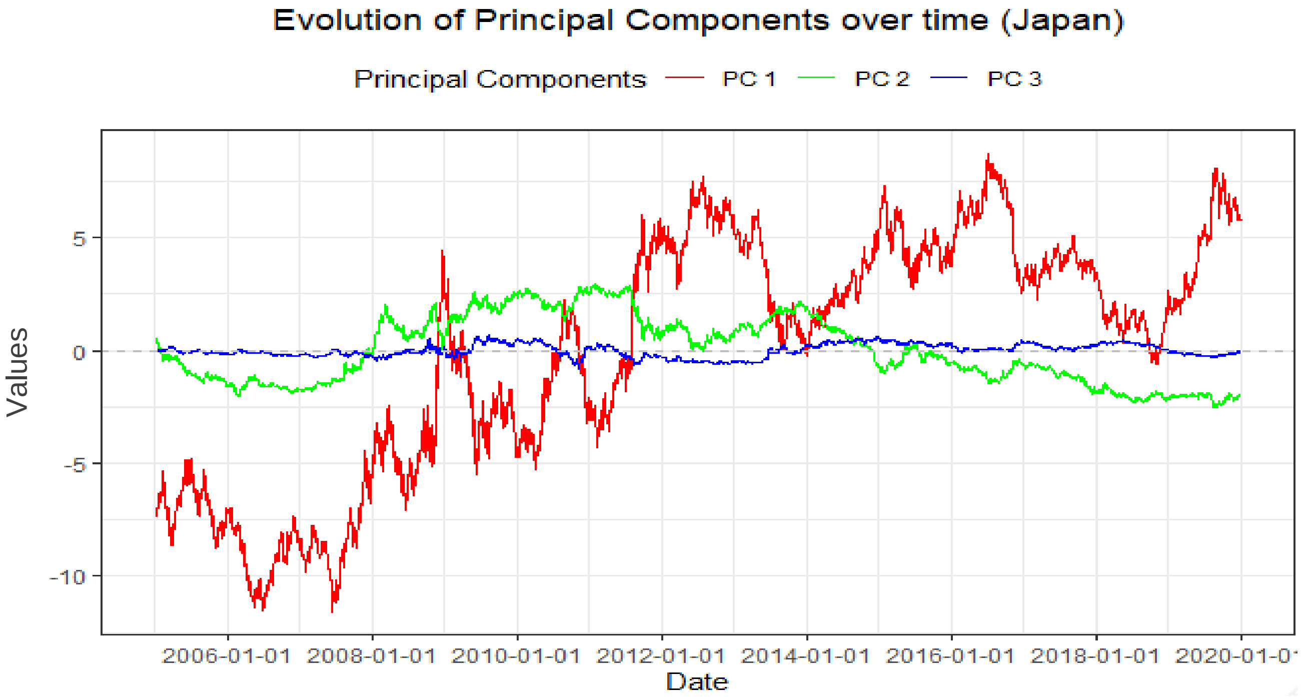
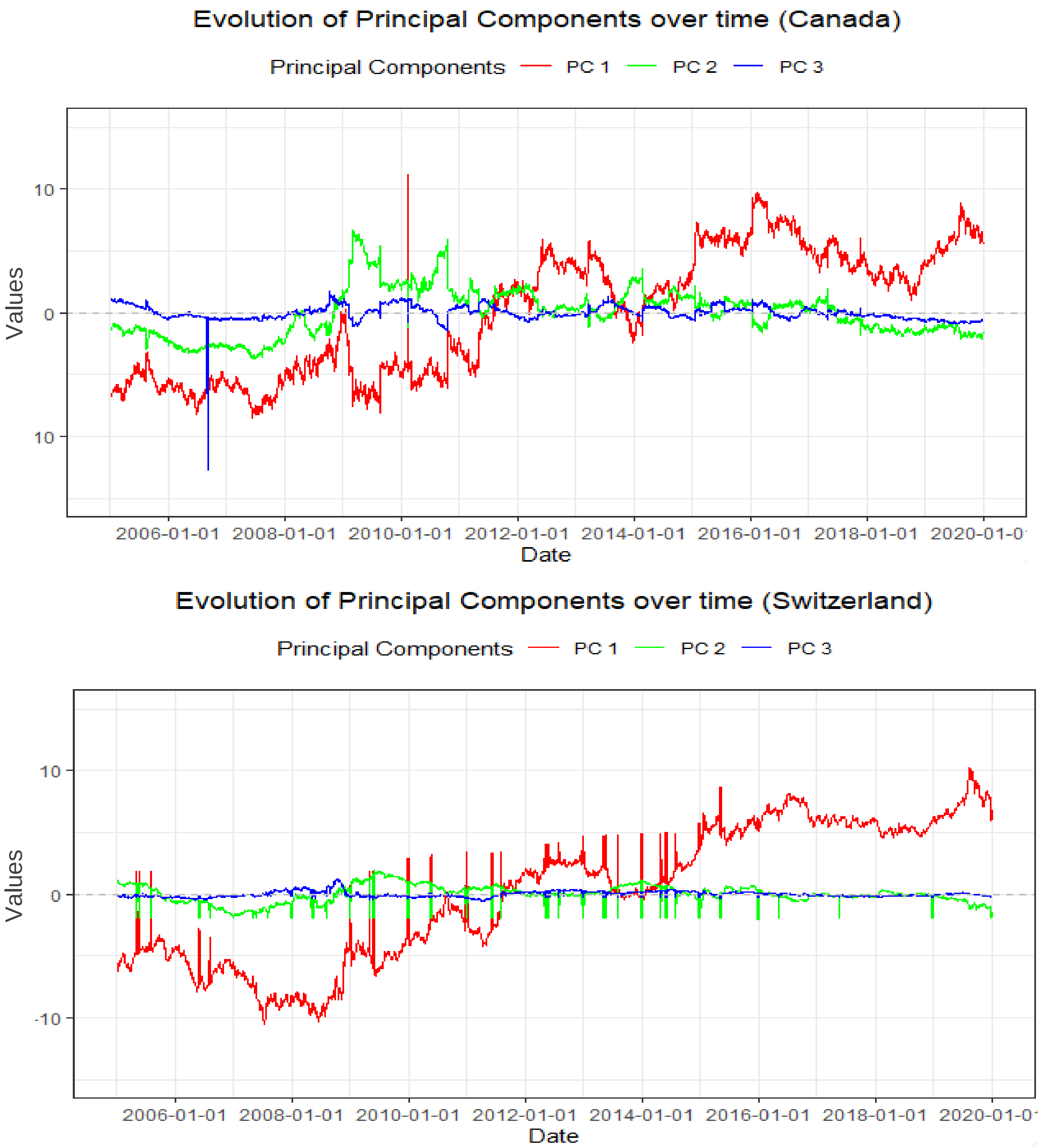


Fig. 19: Explained variance of first 10 PCs for USA, Canada, Switzerland and Japan

We see that in all four countries, the first 3 PCs are sufficient to capture most of the variance in the data. For Canada only, the first PC explains less than 90% of the variance. But explained variance of all four countries converge to around 99% at PC 3.







Fig, 20: Time series of PCs for USA, Japan, Canada and Switzerland

Fig. 20 depicts evolution of the 3 PCs for 4 countries. The common theme in all the graphs is that we have an upward trending first PC. The second and third PC in all four countries are somewhat oscillating and do not have a clear trend. All four countries show at least 2 clear regimes in the evolution of first PC. This may indicate that there exists some correlation in the yield curves across the four countries. To investigate this further, we check the correlation plots of the 3 PCs.

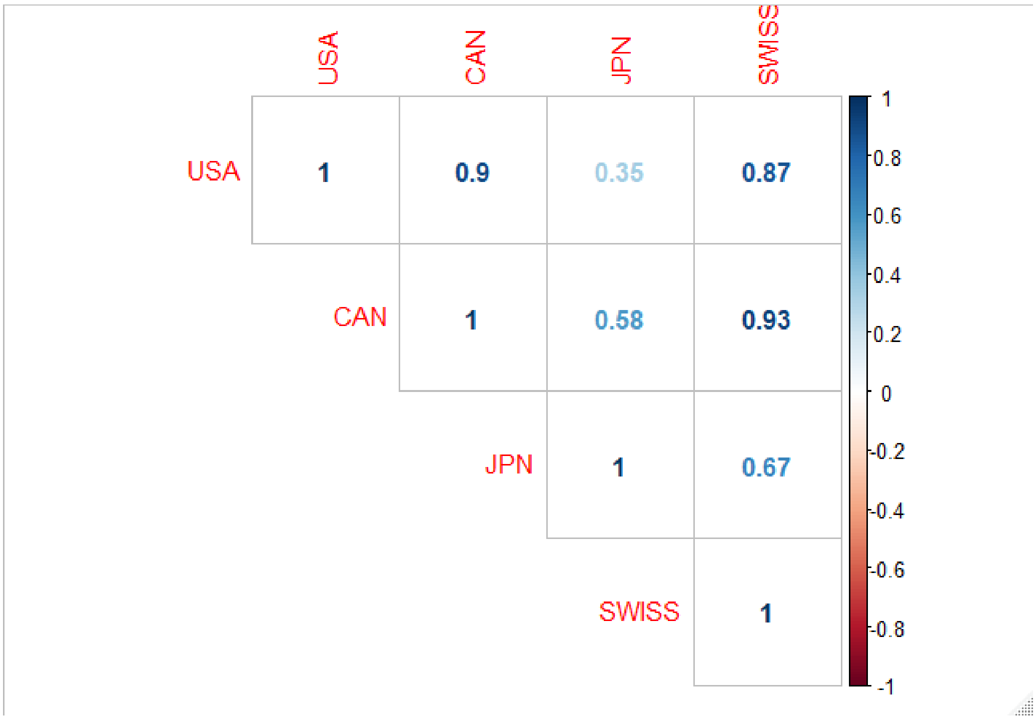


Fig. 21: Correlation plot of first PC for all four countries

We see that the first PC is positively correlated for all four countries. Especially USA, Canada and Switzerland exhibit very high correlations, upwards of 0.8. Remembering that the first PC denotes parallel shift of yield curves, we may say that the yield curves parallel movements are highly correlated across these economies.

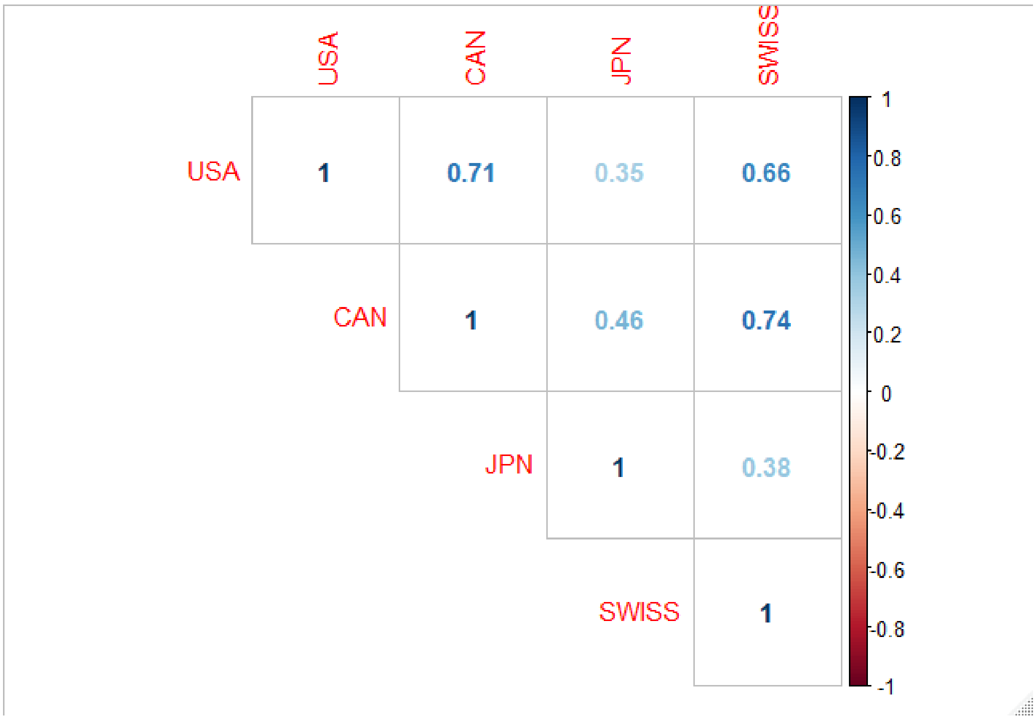
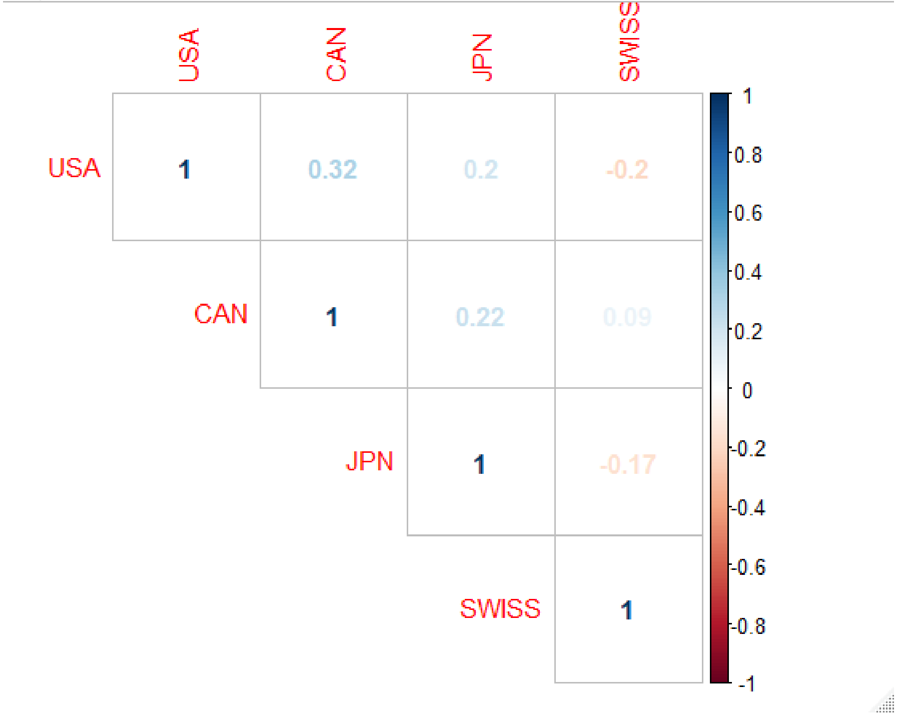


Fig. 22: Correlation plot of second PC for all four countries

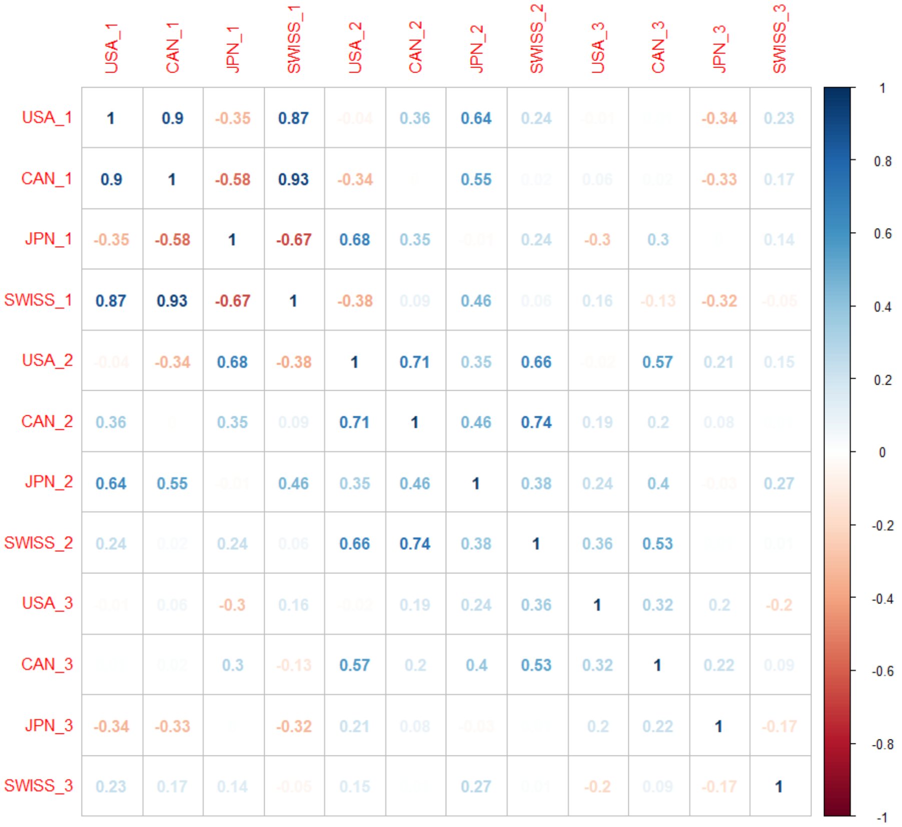
We find that even the second PCs exhibit somewhat high correlation among each other, though less than the first PCs. The second PC determines slope of the yield curves. Thus, we may conclude that alterations in slopes of yield curves happen somewhat in a similar way.



	USA	CAN	JPN	SWISS
USA	1	0.32	0.2	-0.2
CAN		1	0.22	0.09
JPN			1	-0.17
SWISS				1

Fig 23: Correlation plot of third PC for all four countries

The third PCs exhibits much less correlation than the first two. The third PC determines the curvature of the yield curve. However, we may recall that the third PC explains around 1% of the explained variance and is, thus, less important that the first two where, clearly, significant co-movement occurs.



	USA_1	CAN_1	JPN_1	SWISS_1	USA_2	CAN_2	JPN_2	SWISS_2	USA_3	CAN_3	JPN_3	SWISS_3
USA_1	1	0.9	-0.35	0.87	-0.04	0.36	0.64	0.24	-0.01	0.01	-0.34	0.23
CAN_1	0.9	1	-0.58	0.93	-0.34	0	0.55	0.02	0.06	0.02	-0.33	0.17
JPN_1	-0.35	-0.58	1	-0.67	0.68	0.35	-0.01	0.24	-0.3	0.3	0	0.14
SWISS_1	0.87	0.93	-0.67	1	-0.38	0.09	0.46	0.06	0.16	-0.13	-0.32	-0.05
USA_2	-0.04	-0.34	0.68	-0.38	1	0.71	0.35	0.66	-0.02	0.57	0.21	0.15
CAN_2	0.36	0	0.35	0.09	0.71	1	0.46	0.74	0.19	0.2	0.08	0.01
JPN_2	0.64	0.55	-0.01	0.46	0.35	0.46	1	0.38	0.24	0.4	-0.03	0.27
SWISS_2	0.24	0.02	0.24	0.06	0.66	0.74	0.38	1	0.36	0.53	0.01	0.01
USA_3	-0.01	0.06	-0.3	0.16	-0.02	0.19	0.24	0.36	1	0.32	0.2	-0.2
CAN_3	0.01	0.02	0.3	-0.13	0.57	0.2	0.4	0.53	0.32	1	0.22	0.09
JPN_3	-0.34	-0.33	0	-0.32	0.21	0.08	-0.03	0.01	0.2	0.22	1	-0.17
SWISS_3	0.23	0.17	0.14	-0.05	0.15	0.01	0.27	0.01	-0.2	0.09	-0.17	1

Fig. 24: Correlation plot of all 3 PCs in all four countries

Fig. 24 shows correlation of 12 PCs across four countries with each other. We can see there is little correlation between different sets of PCs in the same country or in 2 different countries. PCs with the same number across countries are correlated.

Having established the possibility that yield curves across the four countries show co-movement, we will now try to create global PCs to capture this movement. We have a total of 12 PCs. If co-movement were to exist, applying PCA on these 12 PCs should reduce the number of PCs even further and give us a set of global PCs that can explain movements in yield curves in all four countries.

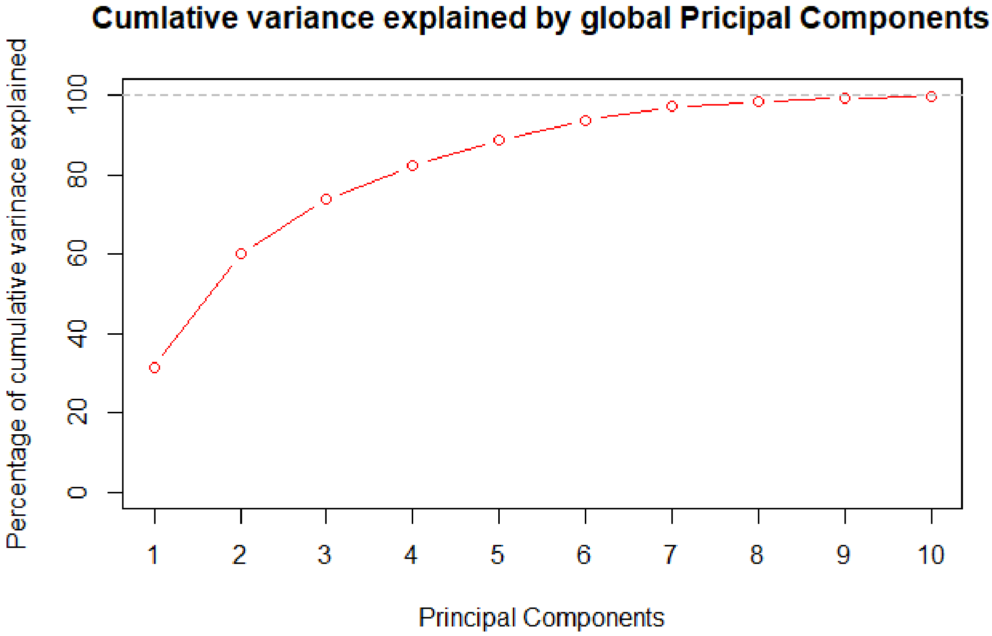


Fig. 25: Explained variance of the global PCs

We see that at least 6 PCs are required to explain around 95% of the variance. The time series of these 6 PCs are shown in Fig 26.

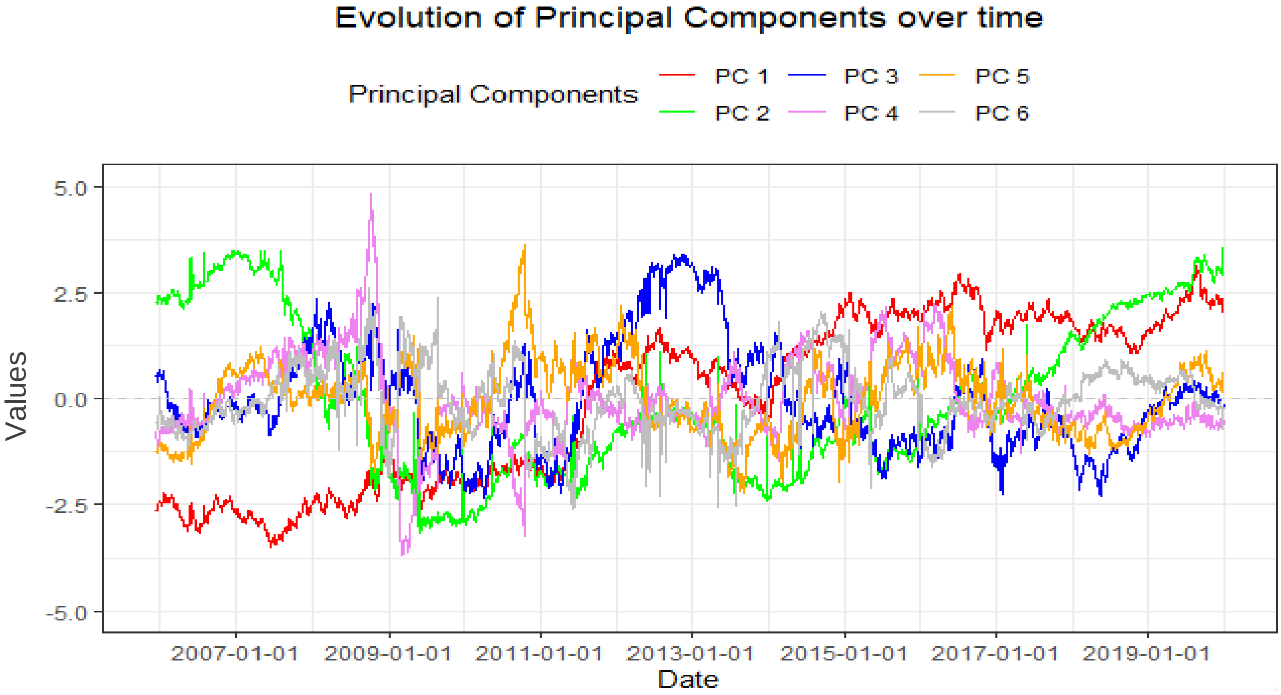


Fig. 26: Time series of 6 global PCs

7. Projecting Yield Curves using Principal Components

Projecting yield curves into the future is a common exercise in financial institutions. Good predictions about movement of yield curves can help mitigate risk through hedging or position management and can help investors take sensible investment decisions. Projections are usually done on individual maturities with interest rate models or on the yield curve using key rates method. The interest rate models are univariate and do not take into account the correlation structure of the yield curves. On the other hand, key rates method assume the entire yield curve follows a few key rates within the curve and all other rates are a derivative of these key rates. Also, key rates method can only account for parallel movement. Thus, around one-tenth of the movements of yield curve is not accounted for in this method. PCs on the other hand offer a good proxy to project yield curves. If we can successfully project PCs, we can easily recombine then to get projected yield curves. The advantage is that the entire correlation structure of the yield curves is taken into account. Also, all types of possible yield curve movements are modelled. The assumption in method, is the functional relationship of the PCs with the yield curves stays the same. This assumption is fairly consistent when our period of projections is not very long.

For this analysis, we take US treasuries data from January 2013 to December 2018 and fit our time series model over the same. We test the predictions over January 2019 to June 2009 data. Fig 27 shows the time series of the first PC.

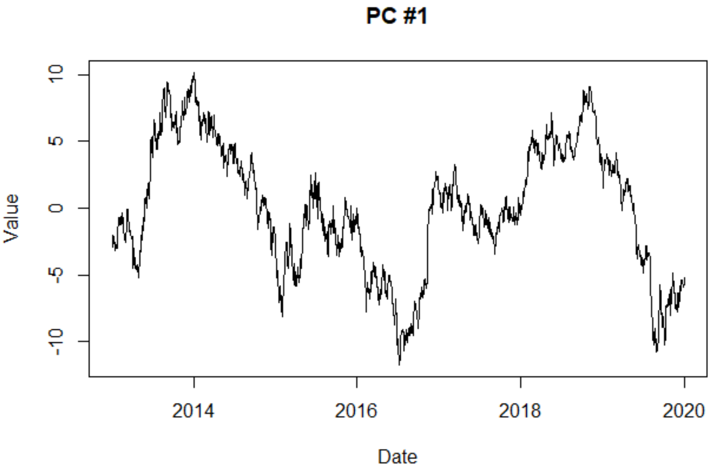


Fig 27: Time series of first Principal Component

We apply a moving average filter over the data to remove trends. We subtract 50-day moving average from the data. Fig 28 shows the resultant time series.

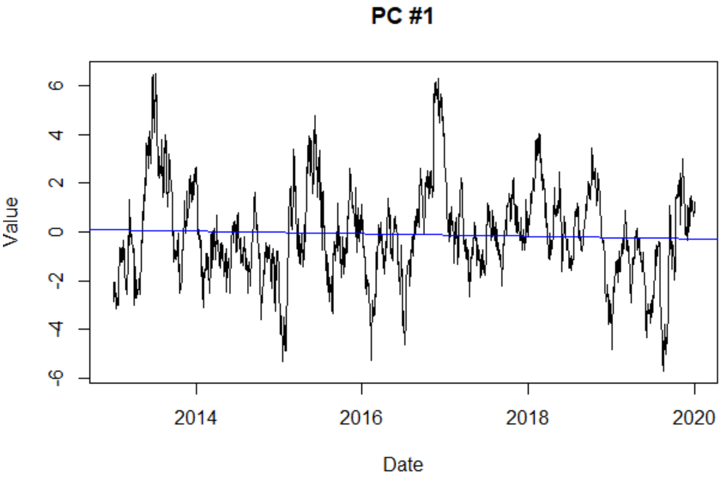
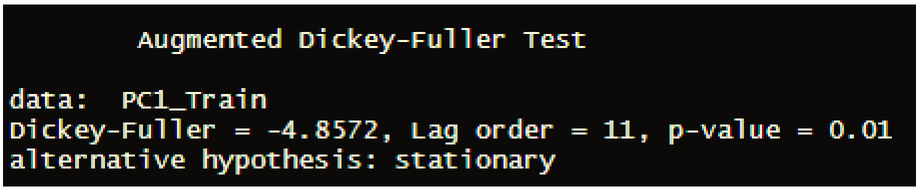


Fig 28: First PC with 50 day Moving Average removed

We divide this data into training and out of sample test as mentioned above. On the training data we perform Augmented Dickey-Fuller (ADF) test to ascertain stationarity of the data. Fig 29 shows the results of the ADF test.



```
        Augmented Dickey-Fuller Test

data:  PC1_Train
Dickey-Fuller = -4.8572, Lag order = 11, p-value = 0.01
alternative hypothesis: stationary
```

Fig 29: ADF test on PC 1 - train

We see that the p-value suggests that we reject the null hypothesis that the data has unit roots i.e. data is not stationary. Next, we plot the ACF and PACF of the training data to get an idea of the ARIMA parameters that we need to train the model.

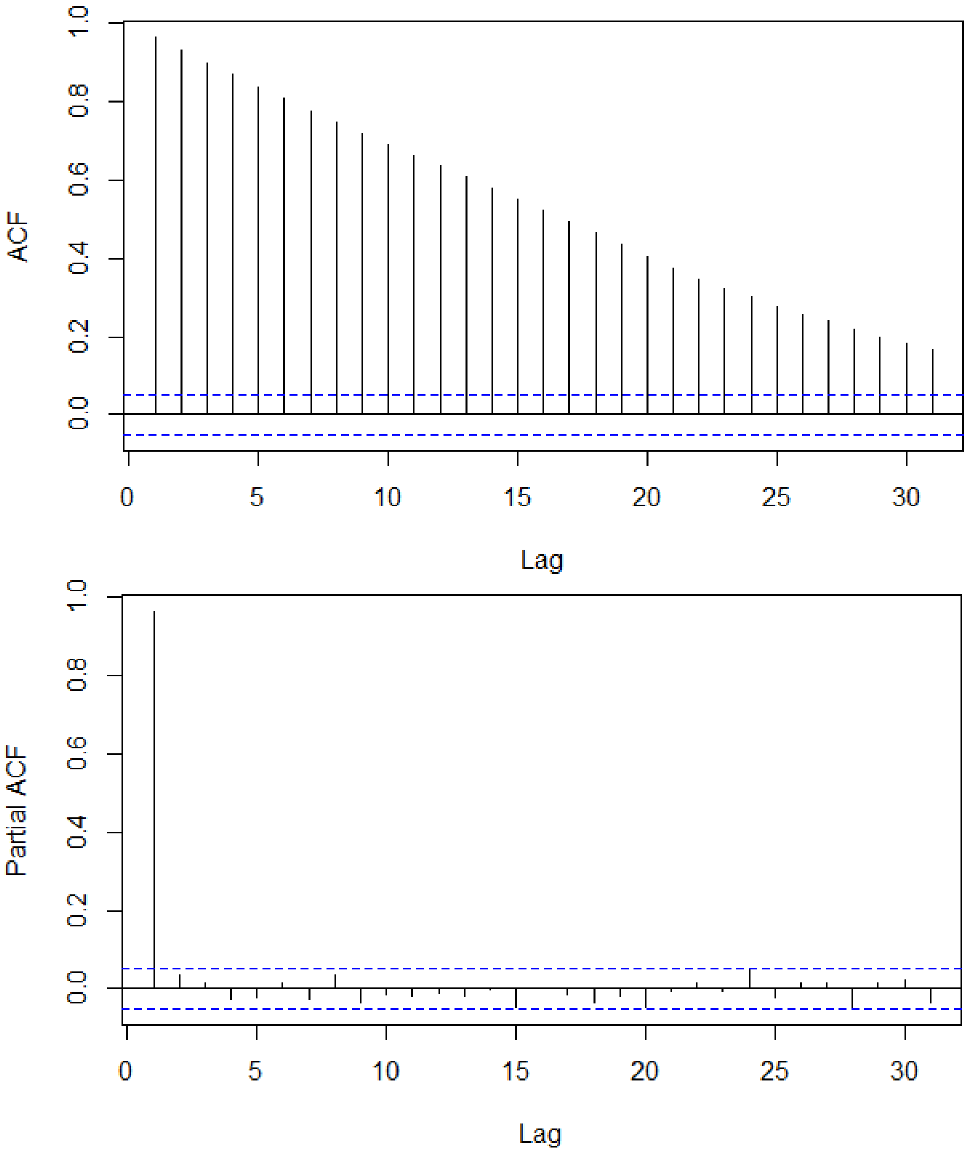
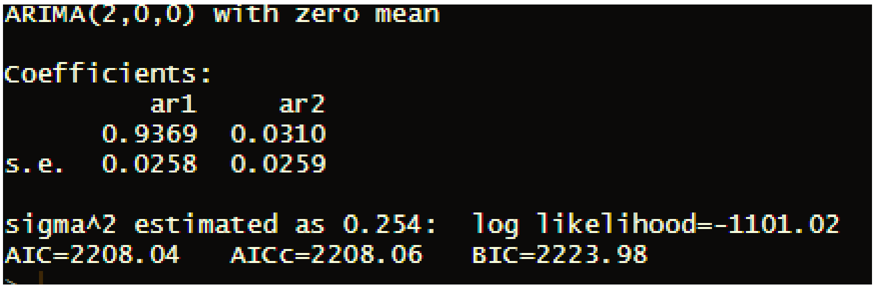


Fig 30: ACF and PACF of PC 1 – train

The ACF plot shows a slowly decaying curve with lag and the PACF has a spike at the first lag. This shows tremendous autoregressive behavior in the data. We use *auto.arima* function in R to get the best ARIMA model parameters and compare it with our intuition.



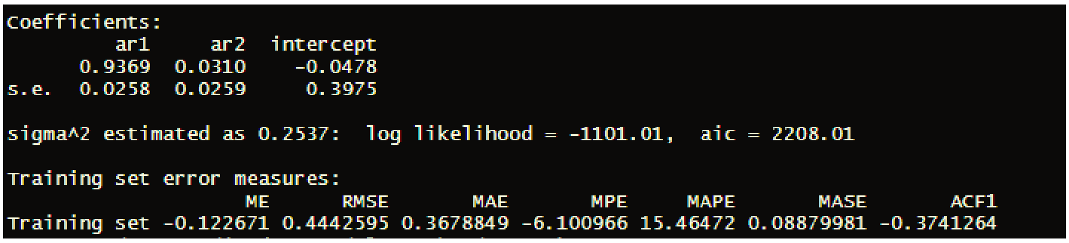
```
ARIMA(2,0,0) with zero mean

Coefficients:
         ar1     ar2
      0.9369  0.0310
s.e.  0.0258  0.0259

sigma^2 estimated as 0.254:  log likelihood=-1101.02
AIC=2208.04   AICc=2208.06   BIC=2223.98
```

Fig 31: *auto.arima* results on PC 1 – train

The results of *auto.arima* also tell us to fit an autoregressive ARIMA(2,0,0) model on the data. The resulting model has the following parameters.



```
Coefficients:
         ar1     ar2  intercept
      0.9369  0.0310    -0.0478
s.e.  0.0258  0.0259     0.3975

sigma^2 estimated as 0.2537:  log likelihood = -1101.01,  aic = 2208.01

Training set error measures:
                    ME      RMSE       MAE       MPE     MAPE       MASE       ACF1
Training set -0.122671 0.4442595 0.3678849 -6.100966 15.46472 0.08879981 -0.3741264
```

Fig 32: ARIMA(2,0,0) on PC 1 – train

The coefficients are all significant. We now see the predictive power of this model on the data.

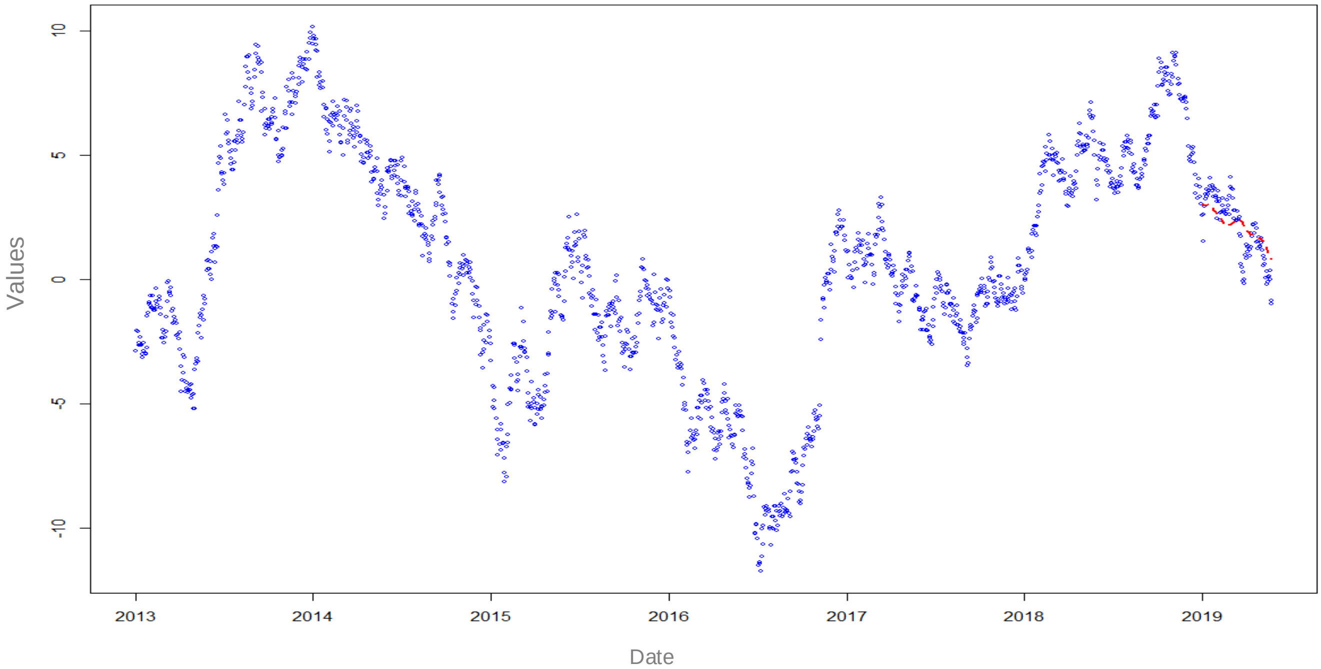


Fig 33: Train data, test data and model prediction (in red) for PC 1

We see the predictions are directionally consistent with the observed data. It is less volatile with lesser number of shorter peaks and troughs but capture long term trends well.

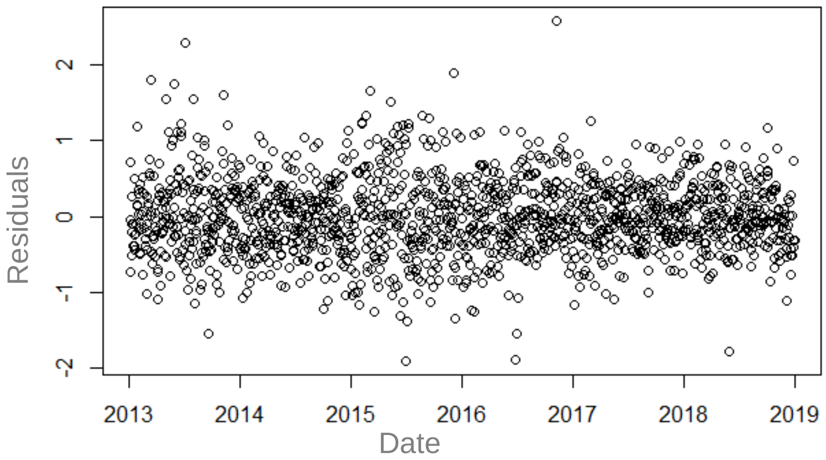
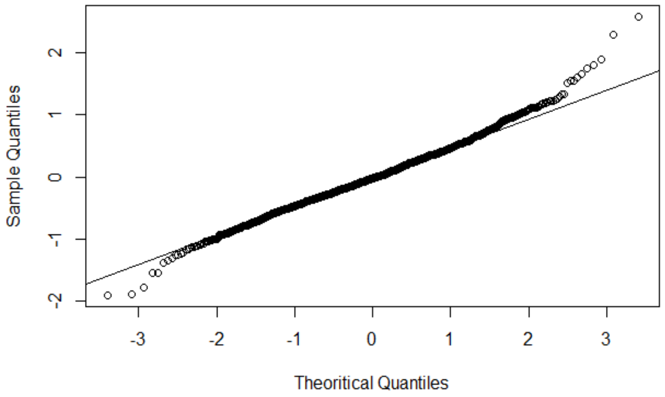
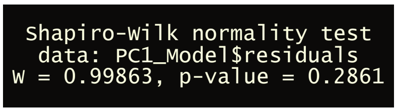


Fig 34: Residual plot of ARIMA(2,0,0) model for PC 1 - train





```
 Shapiro-Wilk normality test
  data: PC1_Model$residuals
W = 0.99863, p-value = 0.2861
```

Fig 35: QQ Plot and Shapiro-Wilk test of residuals for PC 1 model

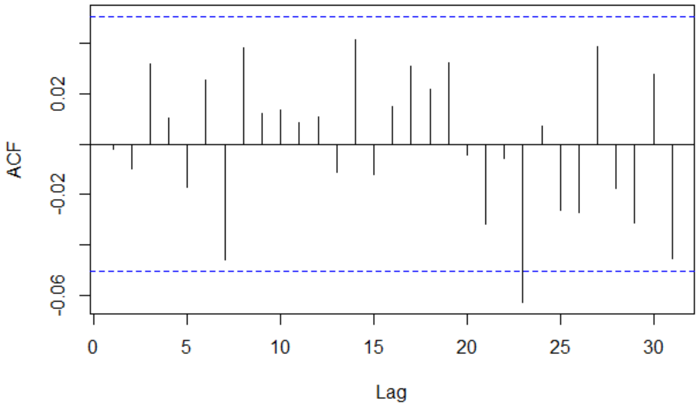


Fig 36: ACF Plot of residuals for PC 1 model

Finally, we check the residuals of the model to see if they are indeed random. Visual inspection of the residuals in Fig 34 shows no discernable irregularities in the pattern. The Q-Q plot in Fig 35 has a few deviations near the tail but overall follow the expected line. The Shapiro-Wilk test has a p-value of 0.28 meaning that we cannot reject the null hypothesis that the data is normally distributed. Fig 36 is the ACF plot of the residuals and it shows there is no significant correlation at any lags.

We move on to modelling PC 2. We follow the exact same procedure of data preparation. Fig. 37 shows the time series of PC 2. Fig. 38 is PC 2 data with 50 day Moving Average removed. Since we will not have moving average data for the first few days, it appears as anomaly in the chart. This will be reflected in residuals as well. We will ignore those.

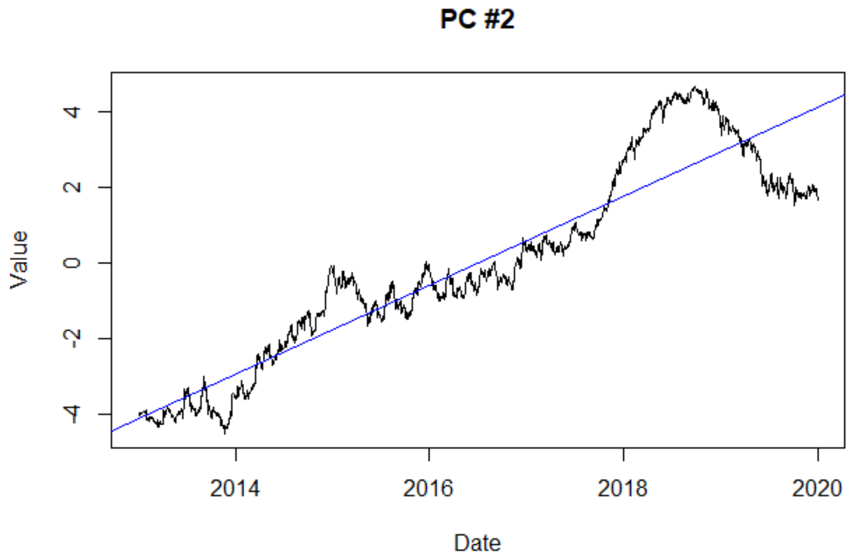


Fig. 37: Time series of PC 2

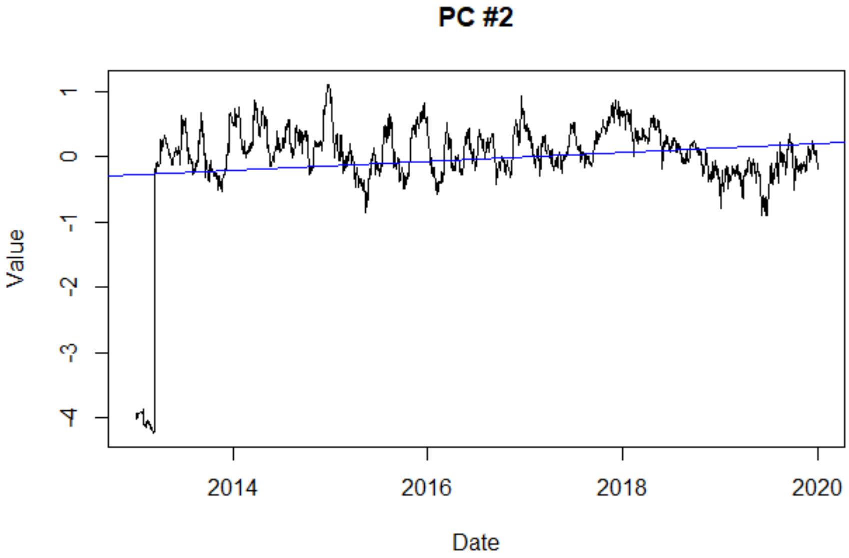


Fig. 38: PC 2 with 50 day moving average removed

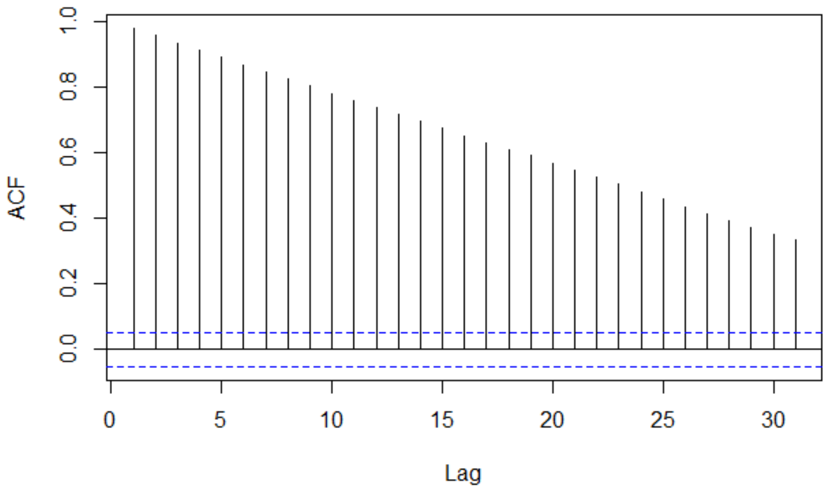


Fig 39: ACF plot of PC 2 – train

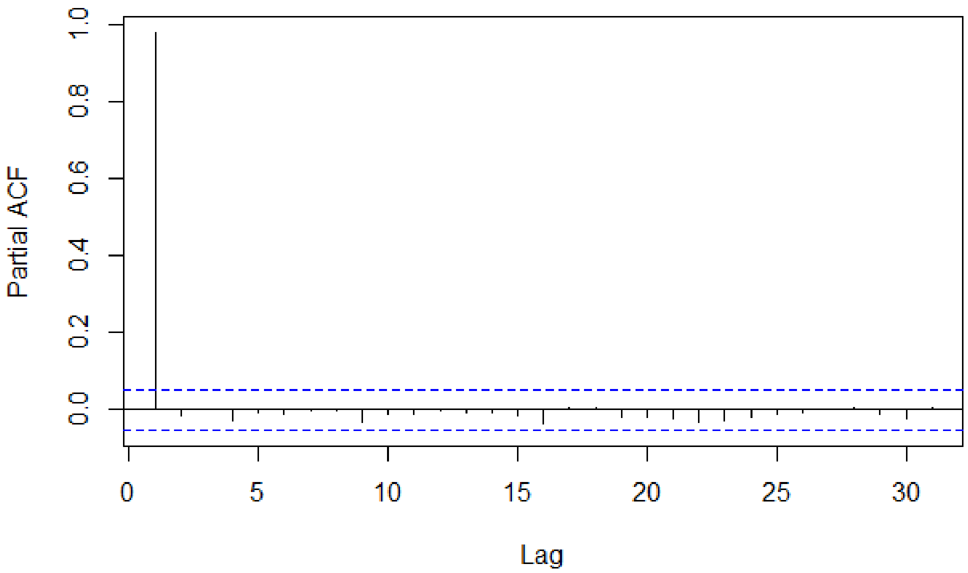
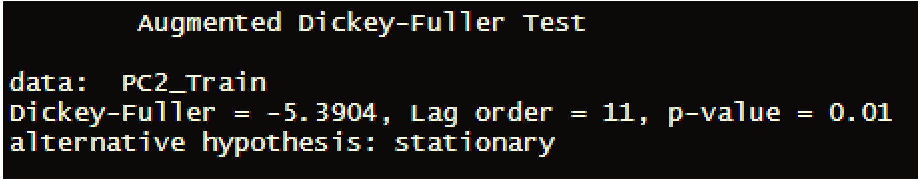


Fig. 40: PACF plot of PC 2 – train

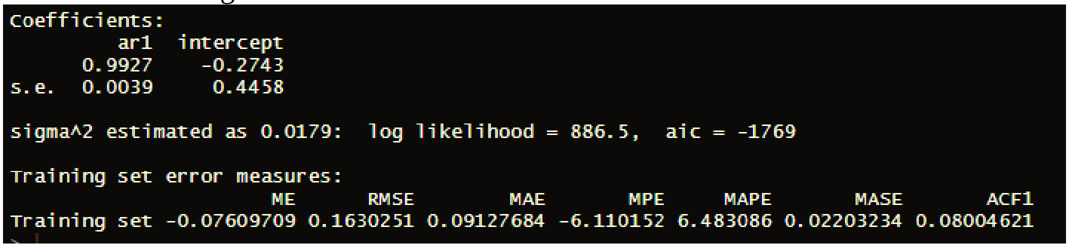
The ACF and PACF charts (Fig. 39 and Fig. 40) shows presence of strong autoregressive nature in the data. Fig. 41 is the ADF test which ascertains that the data is stationary.



```
        Augmented Dickey-Fuller Test

data:  PC2_Train
Dickey-Fuller = -5.3904, Lag order = 11, p-value = 0.01
alternative hypothesis: stationary
```

Fig 41: ADF Test of PC 2 – train



```
Coefficients:
         ar1  intercept
      0.9927    -0.2743
s.e.  0.0039     0.4458

sigma^2 estimated as 0.0179:  log likelihood = 886.5,  aic = -1769

Training set error measures:
                      ME      RMSE        MAE       MPE     MAPE       MASE       ACF1
Training set -0.07609709 0.1630251 0.09127684 -6.110152 6.483086 0.02203234 0.08004621
```

Fig 42: ARIMA(1,0,0) on PC 2 – train

We fit a ARIMA (1,0,0) model to the data. The coefficients are significant. We see how the model fits to out of sample data.

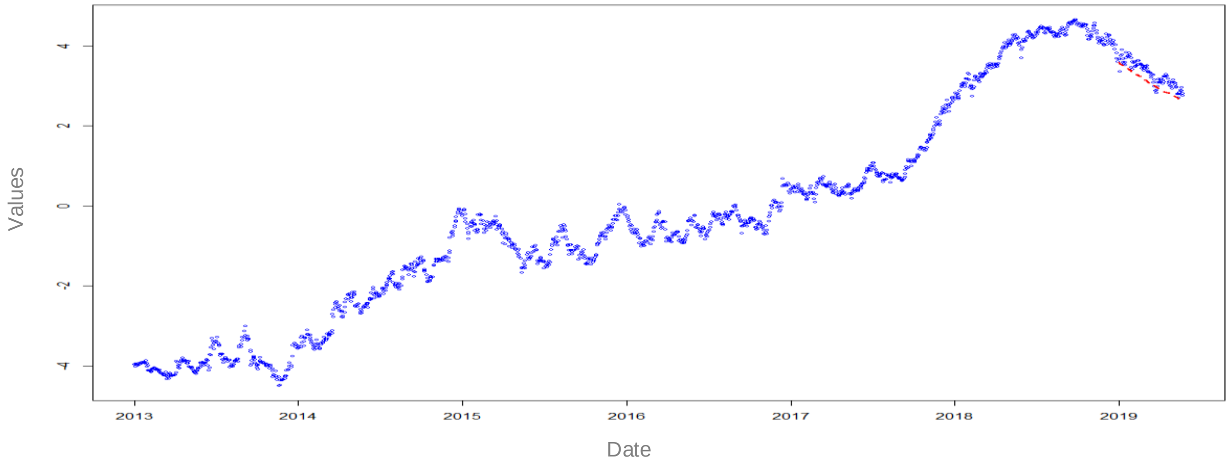


Fig 43: Train data, test data and model prediction (red line) for PC 2

In Fig. 43, we see that the predictions are directionally consistent with the observed data. We look at the residual plots to see if there is any anomaly.

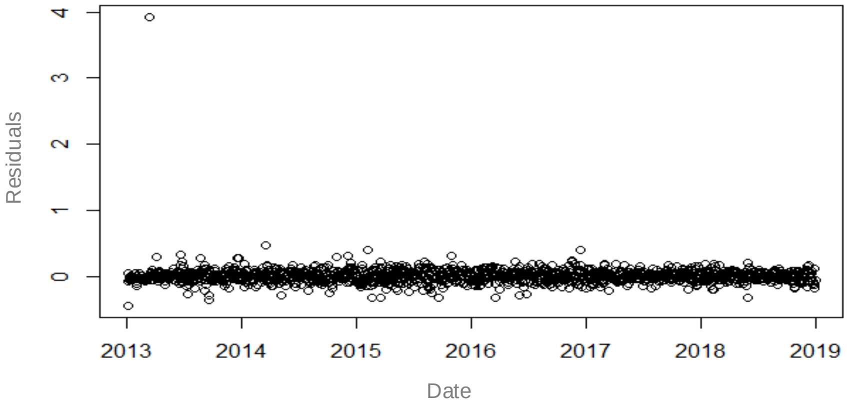
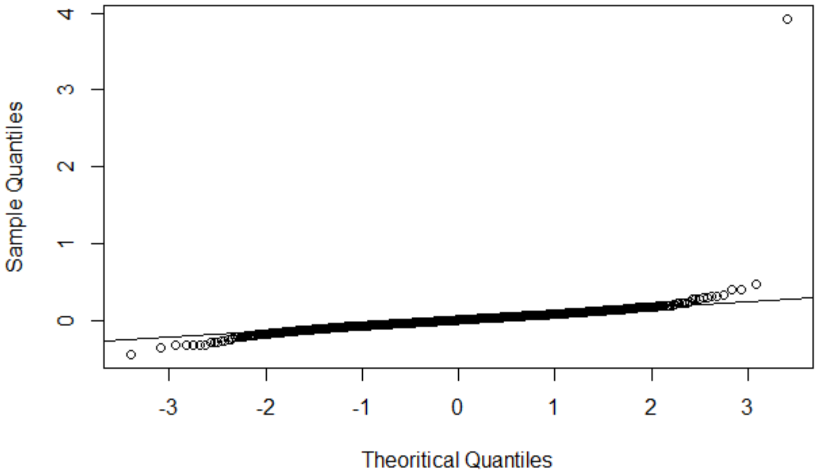
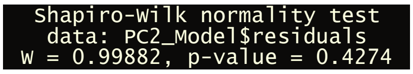


Fig 43: Residuals of the model for PC 2





```
Shapiro-Wilk normality test
data:  PC2_Model$residuals
W = 0.99882, p-value = 0.4274
```

Fig 44: QQ plot and Shapiro-Wilk test of residuals of the model for PC 2

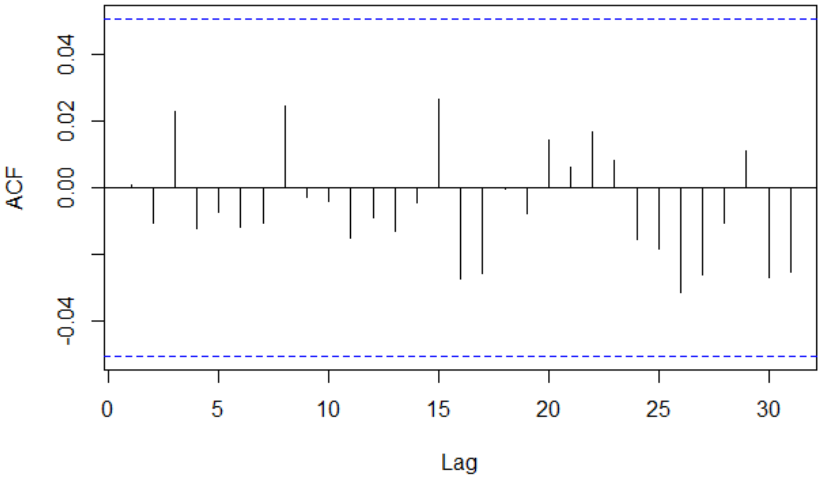


Fig 45: ACF plot of residuals of the model for PC 2

The residual plot in Fig. 43 shows a few extreme values. However, those may be due to the problem mentioned earlier. The Q-Q plot in Fig. 44 is well behaved barring the same extreme values, further validated by the Shapiro-Wilk test. The ACF plot of residuals in Fig. 45 shows no autocorrelation at any lag.

Finally, we model PC 3 using the same procedure. Fig. 46 shows the time series of PC 2. Fig. 47 is PC 2 data with 50 day Moving Average removed.

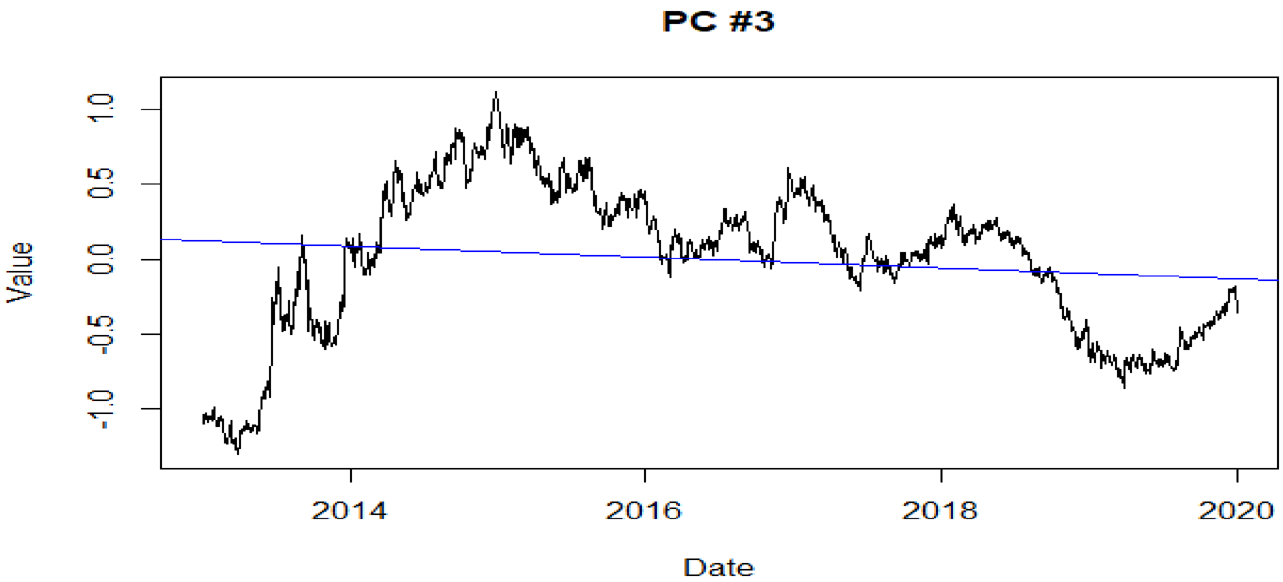


Fig. 46: Time series of PC 3

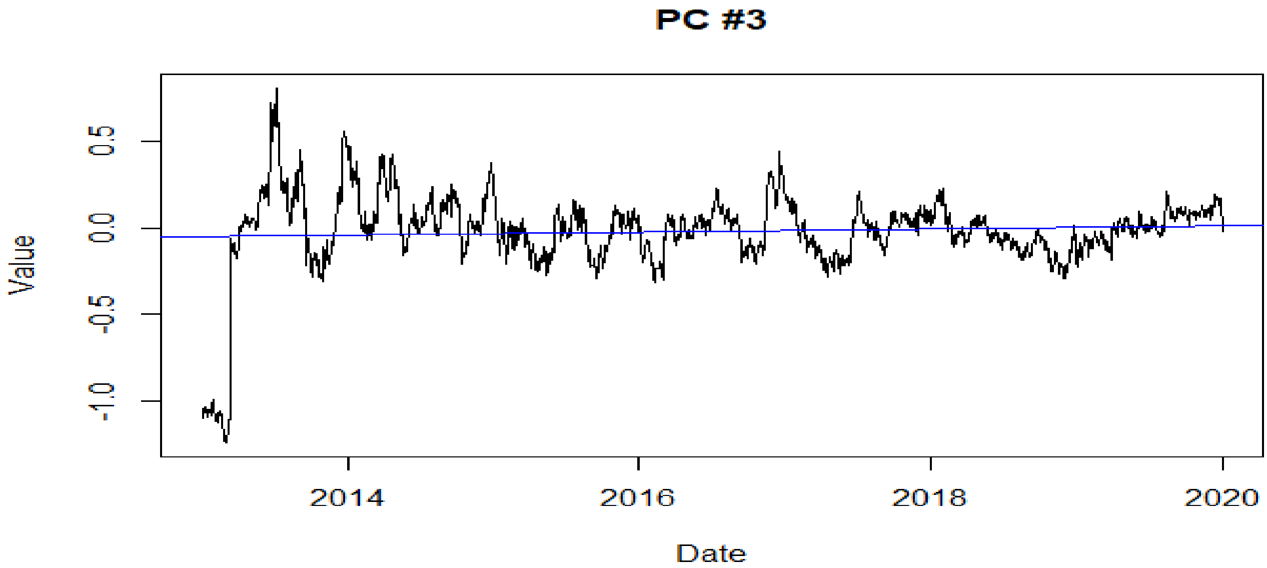
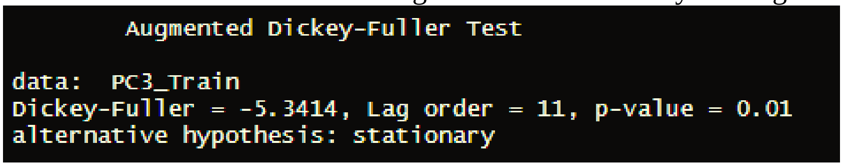


Fig. 47: PC 3 with 50 day moving average removed

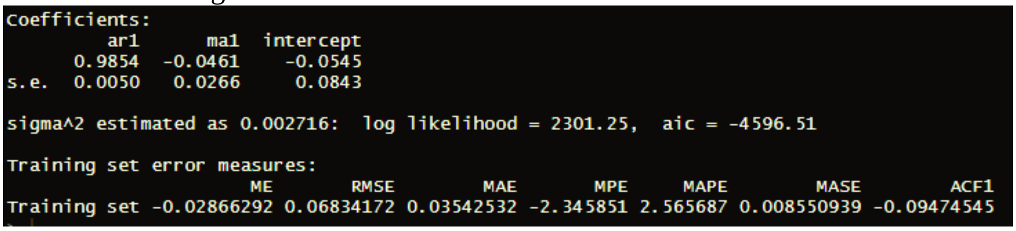


```
        Augmented Dickey-Fuller Test

data:  PC3_Train
Dickey-Fuller = -5.3414, Lag order = 11, p-value = 0.01
alternative hypothesis: stationary
```

Fig. 48: ADF test of PC 3 – train

ADF test shows that the data is stationary. We fit a ARIMA (1,0,1) model to the data. The coefficients are significant.



```
Coefficients:
         ar1      ma1  intercept
      0.9854  -0.0461    -0.0545
s.e.  0.0050   0.0266     0.0843

sigma^2 estimated as 0.002716:  log likelihood = 2301.25,  aic = -4596.51

Training set error measures:
                      ME       RMSE        MAE       MPE     MAPE        MASE        ACF1
Training set -0.02866292 0.06834172 0.03542532 -2.345851 2.565687 0.008550939 -0.09474545
```

Fig. 49: ARIMA(1,0,1) model PC 3

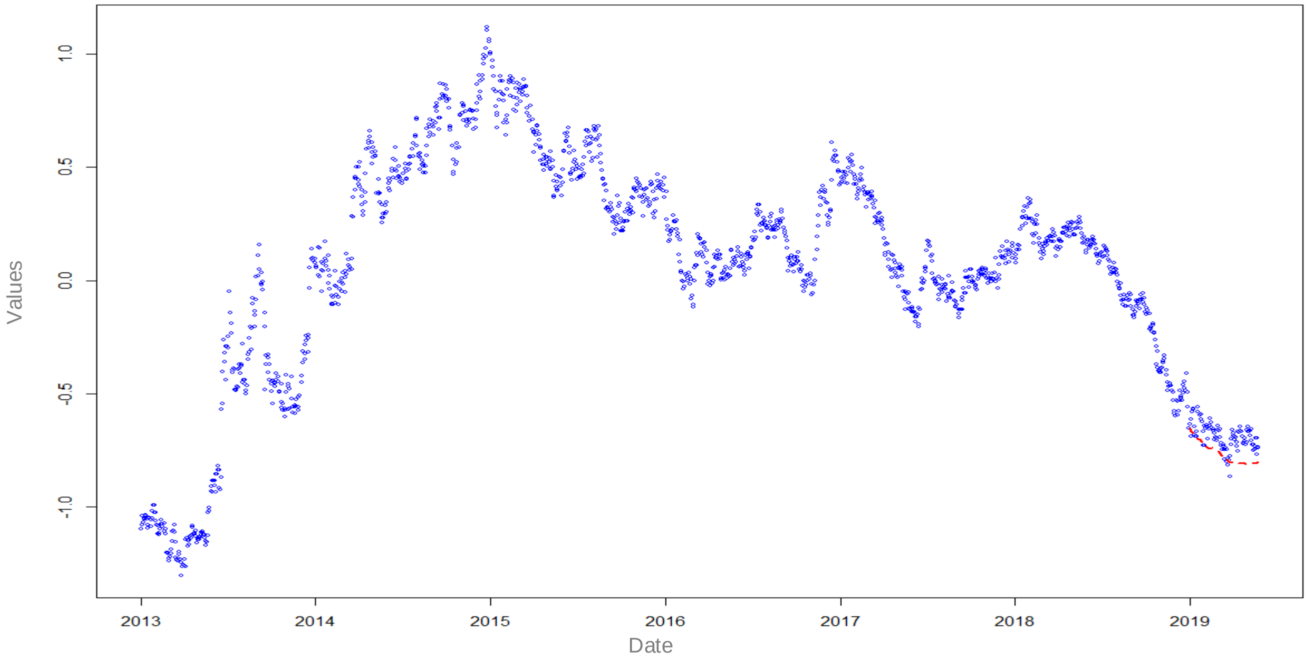


Fig. 50: Train, test and model prediction of PC 3

Again, in Fig 50, we see that the predictions and the observed data is directionally consistent.

Finally, we multiply the projected PCs with the factor loadings to get projected yield curves for January 2019 to December 2019. To demonstrate the accuracy of prediction, we plot 2 maturities – 10 year and 20 year – predicted values and observed values.

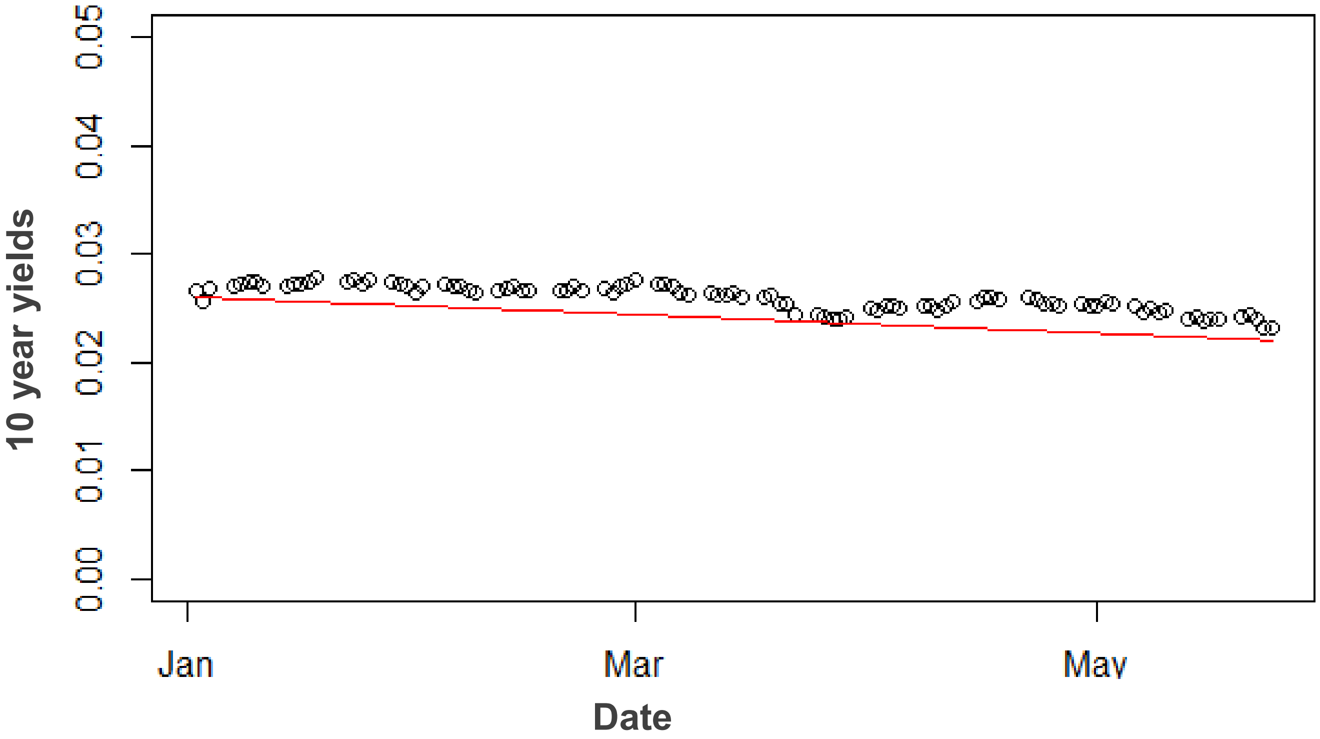


Fig. 51: 10 year maturity yields – Predicted (red line), actual (black circles)

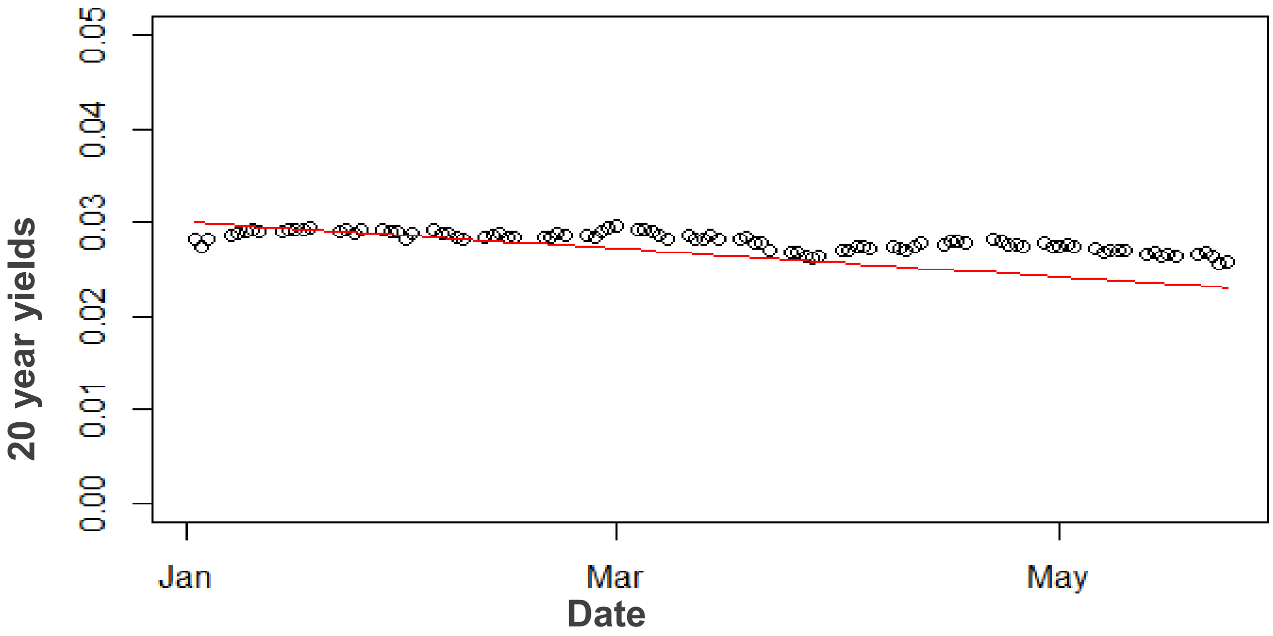


Fig 52: 20 year maturity yields – Predicted (red line), actual (black circles)

We can see that the predictions for the short term, January 2019 to May 2019, are quite close and directionally consistent with observed values.

8. Simulating Yield Curves from Principal Components

Financial analysts simulate returns for stocks all the time. Monte Carlo Simulations are very common in the financial planning industry. They help institutions to measure risk, set expectations based on confidence levels and decide on adequate capital requirements. On the other hand, risk analysis on yield curves is mainly done using scenario analysis by shocking the key rates. The population of outcomes derived using this technique is very limited compared to that achieved using Monte Carlo Simulations. In this section, we try to simulate yield curves in a similar way to stock returns using the PCs.

Generally, we derive stock return simulations from Normal distributions. Although, extensive literature exists on the fact that stock returns may exhibit fatter tails and excess kurtosis, since the return histogram exhibit definite bell shaped structure Normal distribution is an easy way to approximate the returns. We will see if the histogram of the PCs from the yield curve show similar bell shaped structure.

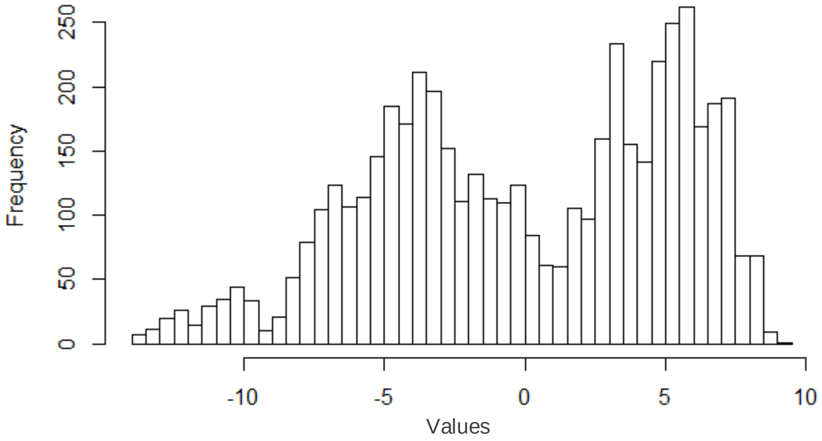


Fig. 53: Histogram of PC 1

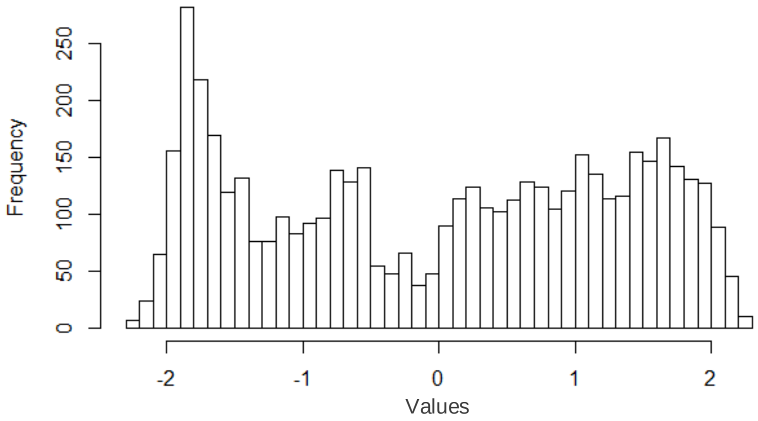


Fig 54: Histogram of PC 2

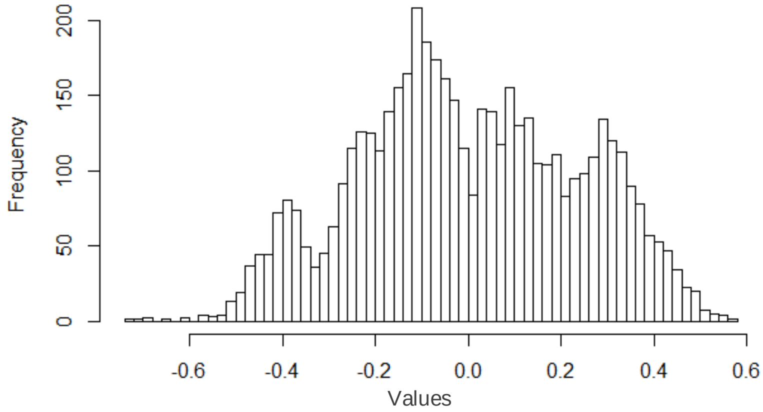


Fig 55: Histogram of PC 3

In Fig 53, 54 and 55, we find that none of the histograms for the 3 PCs are unimodal. Hence, we cannot use any of the common distributions to simulate PCs.

In stocks, we generally simulate stock returns instead of stock prices. Similarly, we will try to simulate percentage change in yield curves (PCYC) instead. First, we calculate PCYC, apply SVD, extract the PCs and look at their histograms.

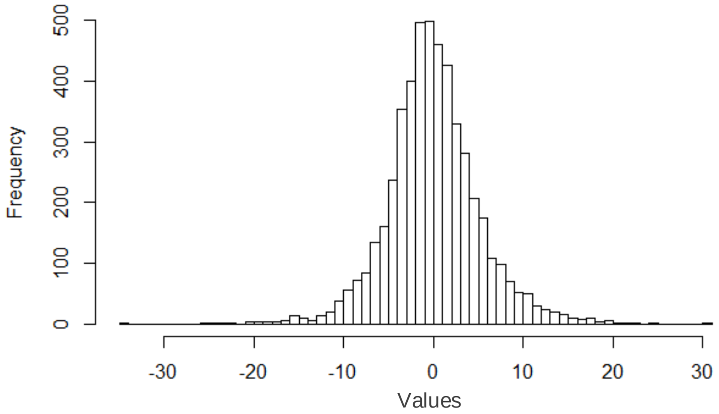


Fig 56: Histogram of PCYC PC 1

We see that for PCYCs, the histogram of PC 1 is perfectly unimodal. Similarly, in Fig 57 and 58, we see that the histogram of PC 2 and PC 3 also show perfect unimodal structure. Thus, they may conform to known distributions.

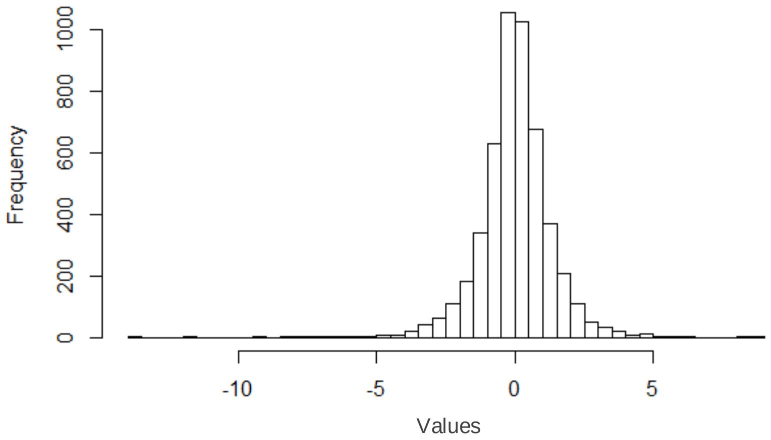


Fig 57: Histogram of PCYC PC 2

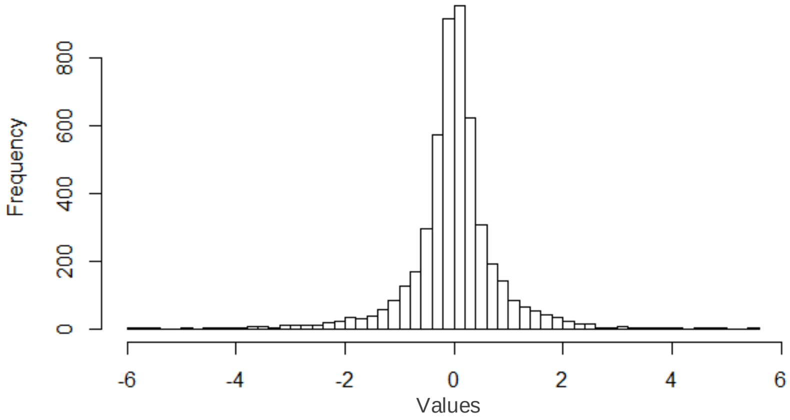
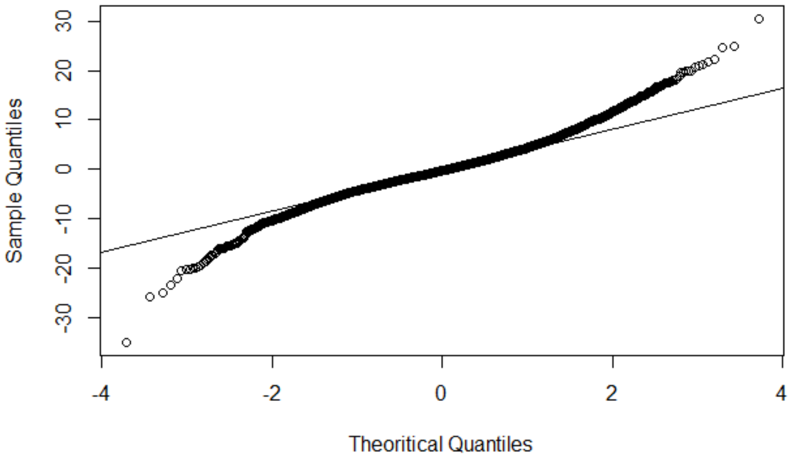
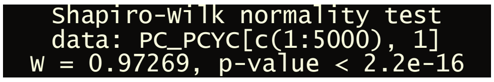


Fig 58: Histogram of PCYC PC 3

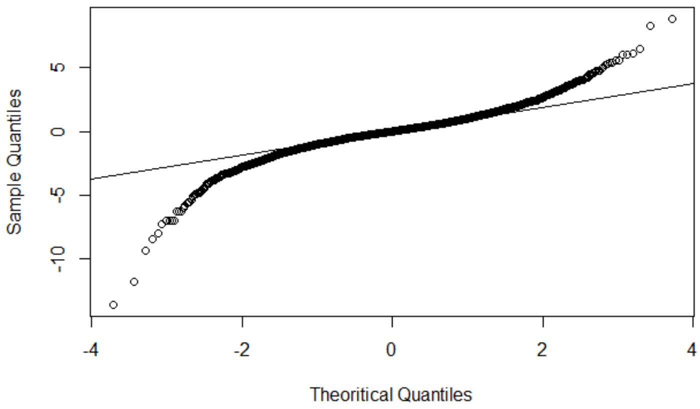


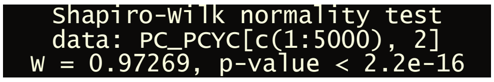


```
  Shapiro-Wilk normality test
  data: PC_PCYC[c(1:5000), 1]
W = 0.97269, p-value < 2.2e-16
```

Fig 59: Q-Q plot and Shapiro-Wilk test of PCYC PC 1

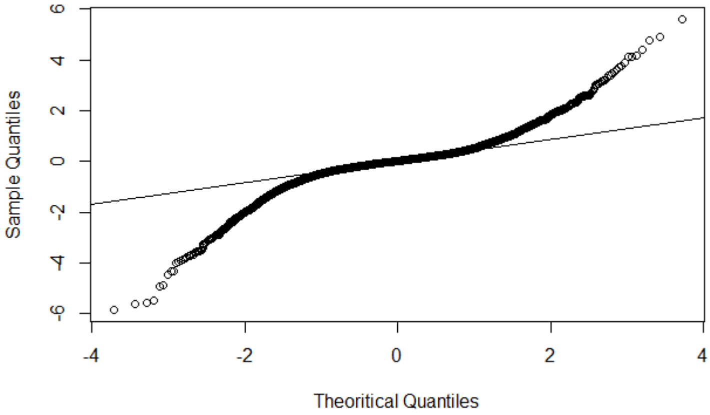
In Fig 59, 60 and 61, we see that the Q-Q plot of the PCYC PCs show considerable deviations at tails. Shapiro-Wilk test rejects the null hypothesis that the data is normal. This tells us that although bell-shaped, Normal distributions may not be good fit for these PCs. The first PC has a skewness of 0.17 and excess kurtosis of 2.5. The second PC has a skewness of -0.57 and excess kurtosis of 7.78. The third PC has a skewness -0.47 and excess kurtosis of 7.22. While the skewness is not far from 0 for any of the PCs, the excess kurtosis suggests that while we may assume that the distributions are not skewed, they are definitely leptokurtic and in a way that vastly deviates from Normality assumptions.

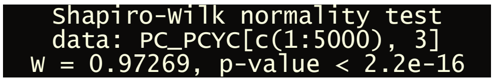




```
Shapiro-Wilk normality test
data: PC_PCYC[c(1:5000), 2]
W = 0.97269, p-value < 2.2e-16
```

Fig 60: Q-Q plot and Shapiro-Wilk test of PCYC PC 2





```
Shapiro-Wilk normality test
data: PC_PCYC[c(1:5000), 3]
W = 0.97269, p-value < 2.2e-16
```

Fig 61: Q-Q plot and Shapiro-Wilk test of PCYC PC 3

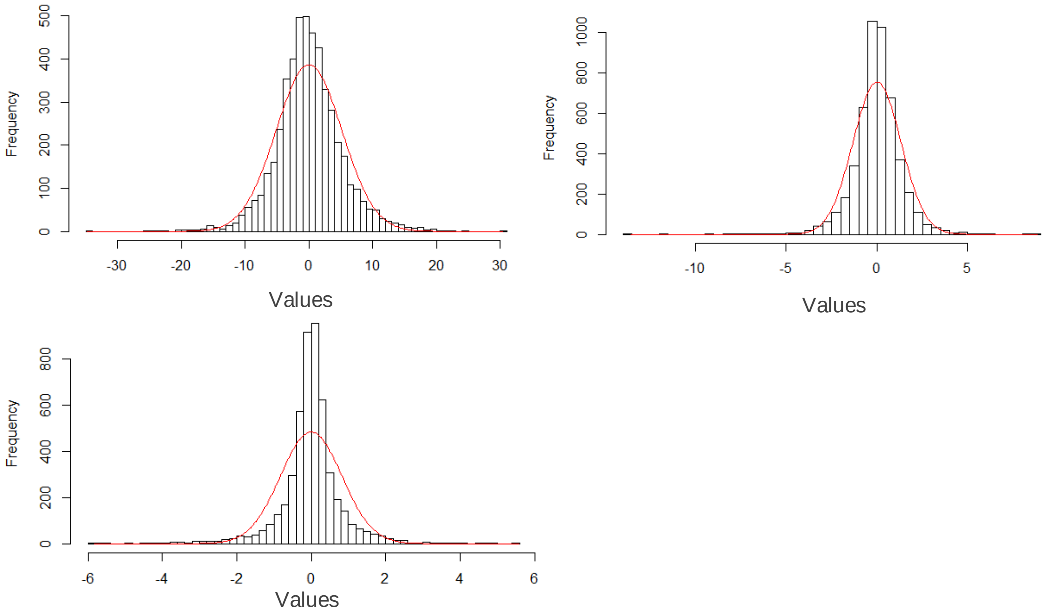


Fig 62: From left, Normal fit over histogram of PCYC PC 1, PCYC PC 2 and PCYC PC 3

We see from the Normal fit in Fig 62 that all 3 PCs exhibit significant leptokurtosis. A suitable distribution to fit these histograms is the Laplace distribution. The Laplace distribution requires two parameters – location and scale. The pdf of the Laplace distribution is as follows:

$$P(x) = \frac{1}{2b}\exp(-\frac{|x-\mu|}{b})$$

The Laplace distribution requires a scale and location parameters. To estimate the appropriate parameters, we write a custom function *bestfit* that estimate these two parameters. The function minimizes the mean squared error between the quantile values of the observed data and that of the Laplace distribution.

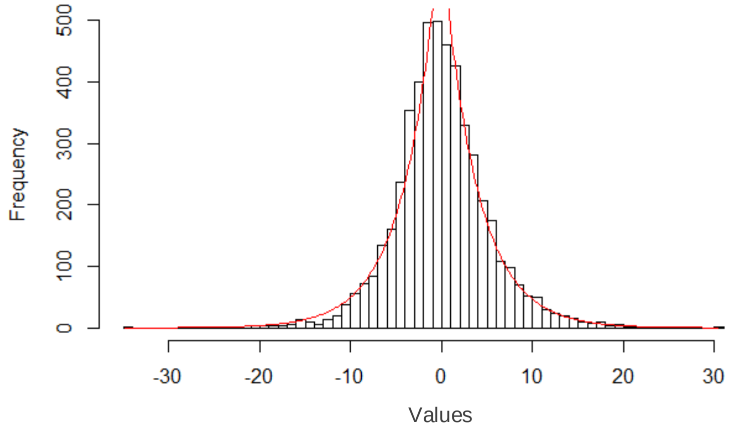
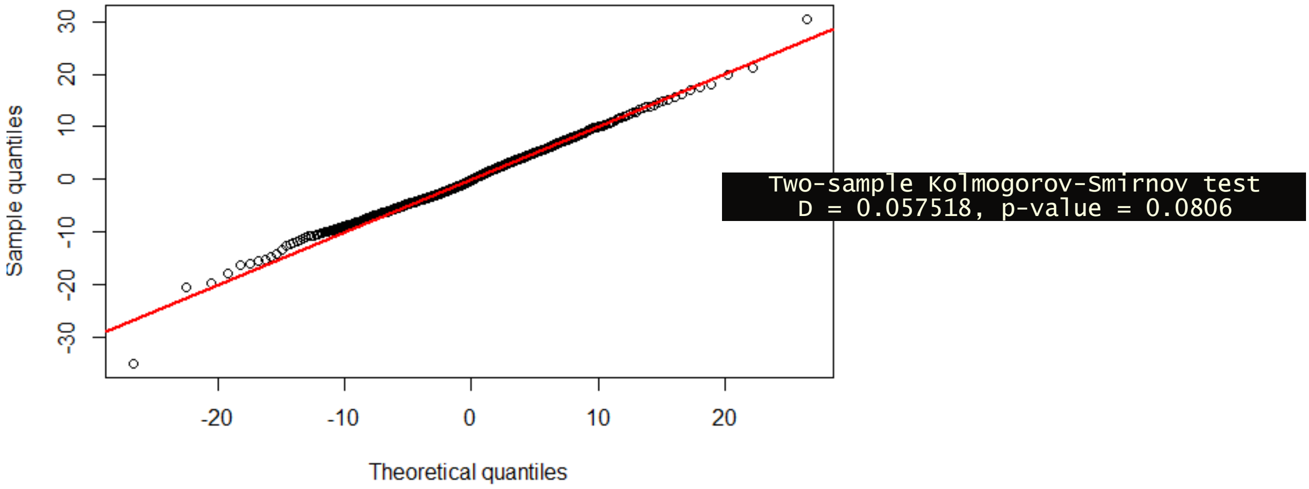


Fig 63: Laplace distribution fit over PCYC PC 1



```
Two-sample Kolmogorov-Smirnov test
D = 0.057518, p-value = 0.0806
```

Fig 64: Q-Q plot and Kolmogorov-Smirnoff test of Laplace distribution and PCYC PC 1

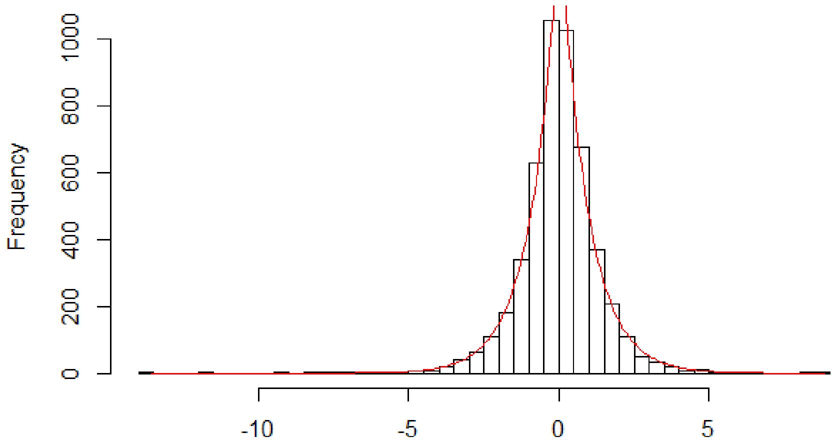
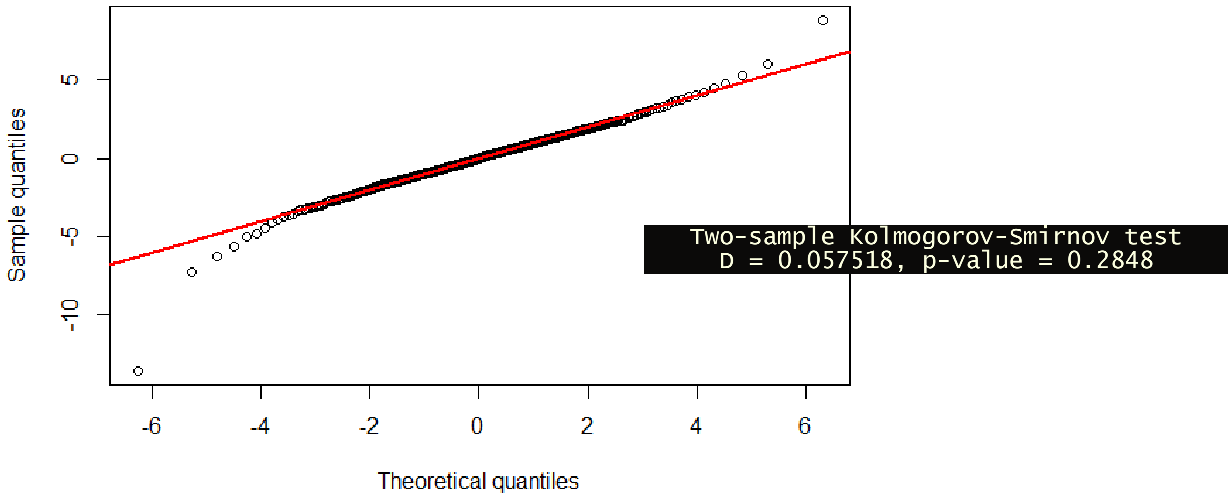


Fig 65: Laplace distribution fit over PCYC PC 2



```
Two-sample Kolmogorov-Smirnov test
D = 0.057518, p-value = 0.2848
```

Fig 66: Q-Q plot and Kolmogorov-Smirnoff test of Laplace distribution and PCYC PC 2

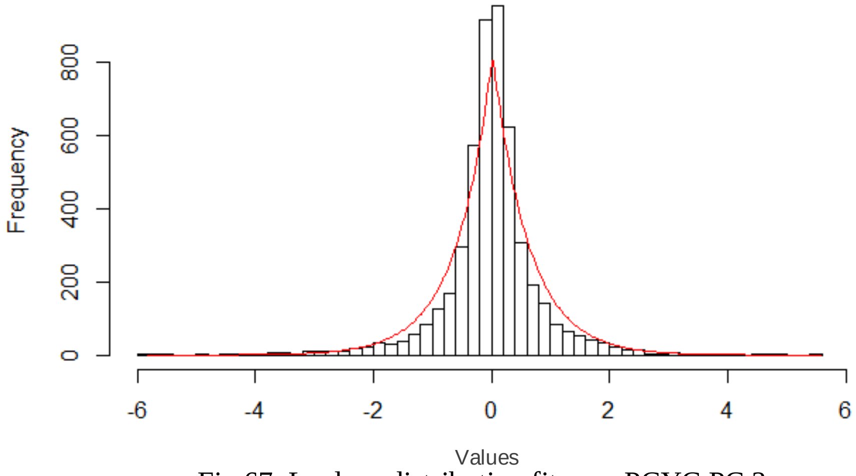
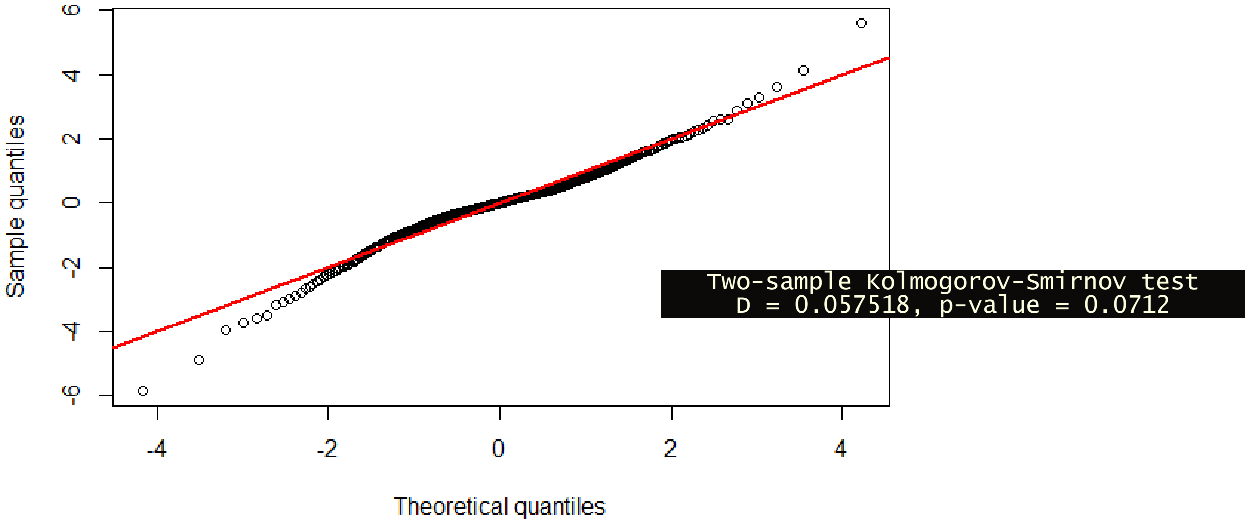


Fig 67: Laplace distribution fit over PCYC PC 3



```
Two-sample Kolmogorov-Smirnov test
D = 0.057518, p-value = 0.0712
```

Fig 68: Q-Q plot and Kolmogorov-Smirnoff test of Laplace distribution and PCYC PC 3

Being satisfied with the Q-Q plots and Kolmogorov-Smirnoff tests that Laplace distribution fits well over the data, we proceed to create yield curve simulations with Laplace distribution. Since, we simulate percentage changes in yield curves, we run 1000 simulations and then apply those changes on the last observed yield curve i.e 31 December 2019 to get 1000 simulations for the next day in Fig 69.

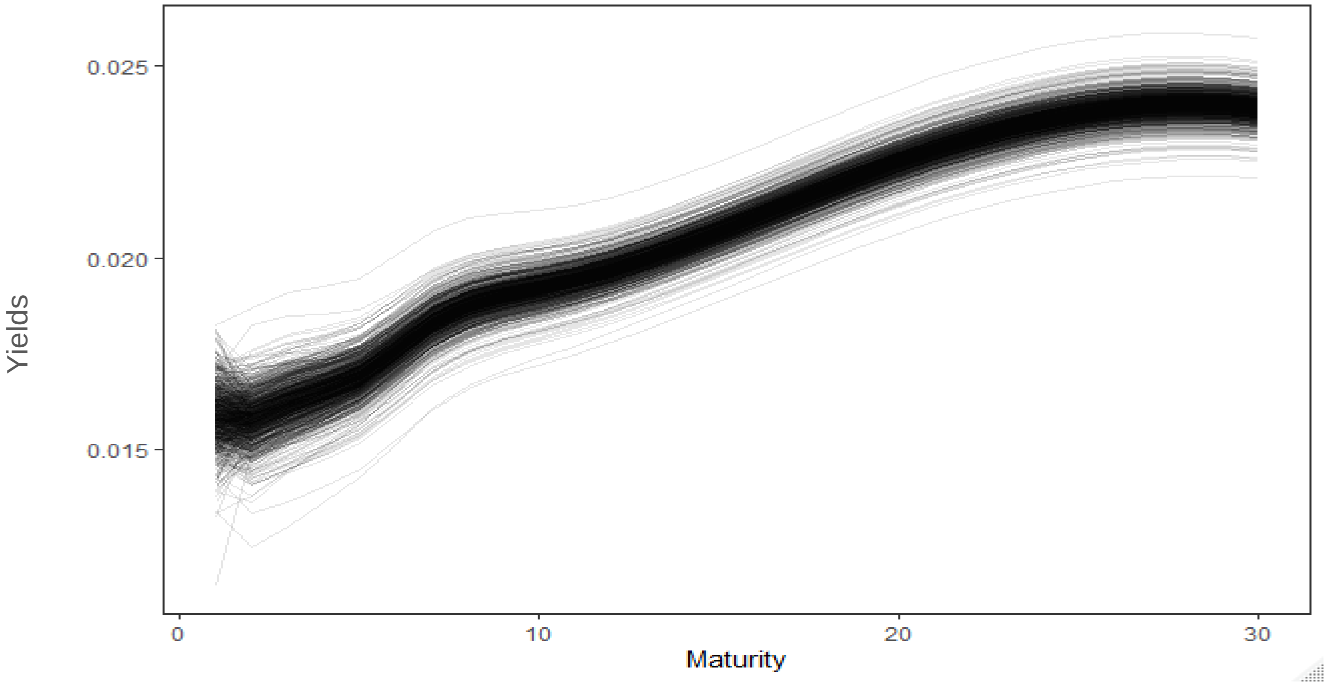
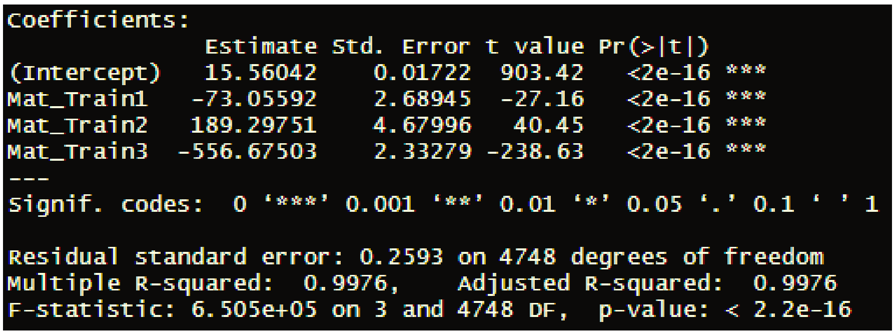


Fig 69: Yield Curve Simulations using Laplace distribution

9. Creating factor portfolios from US Treasuries

Factor models have long been used to describe stock returns. Fama-French three factor model is one such popular technique. Factors include Growth, Momentum, Value, Dividend Yield, etc. Fund houses have created factor portfolios by combining stocks in a way that overweighs one particular factor. In this way, an investor that wants exposure to a factor and not any particular stock, can buy the factor portfolio. Similarly, we have earlier demonstrated how the three PCs of the yield curve represent 3 distinct movements of the yield curves. These can be thought of as independent factors influencing the movements of yield curves. We may create factor portfolios of yield curves that has exposure to only 1 factor and thus move only in one particular way. This may help in several different circumstances. For example, if someone wants to hedge against parallel movement of yield curves, he can use PC 1 factor portfolio which will have only parallel movements and be agnostic to twists and inversions.

To create factor portfolio, we will combine a few maturities with weights which will make them equal to a particular PC. For this we choose 3 maturities – 3 year, 5 year and 10 year. Why we choose only 3, no more or no less will be discussed later. We regress them against PC 1.



```
Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)   15.56042    0.01722  903.42   <2e-16 ***
Mat_Train1   -73.05592    2.68945  -27.16   <2e-16 ***
Mat_Train2   189.29751    4.67996   40.45   <2e-16 ***
Mat_Train3  -556.67503    2.33279 -238.63   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.2593 on 4748 degrees of freedom
Multiple R-squared:  0.9976,    Adjusted R-squared:  0.9976 
F-statistic: 6.505e+05 on 3 and 4748 DF,  p-value: < 2.2e-16
```

Fig 70: Model fit of maturities against PC 1

In Fig 70, we see that we are able to achieve a very high Adjusted R-Square of 0.997. The portfolio we created will thus have the following weights for 3 year, 5 year and 10 year maturities respectively : - 73.06, 189.30 and -556.68.

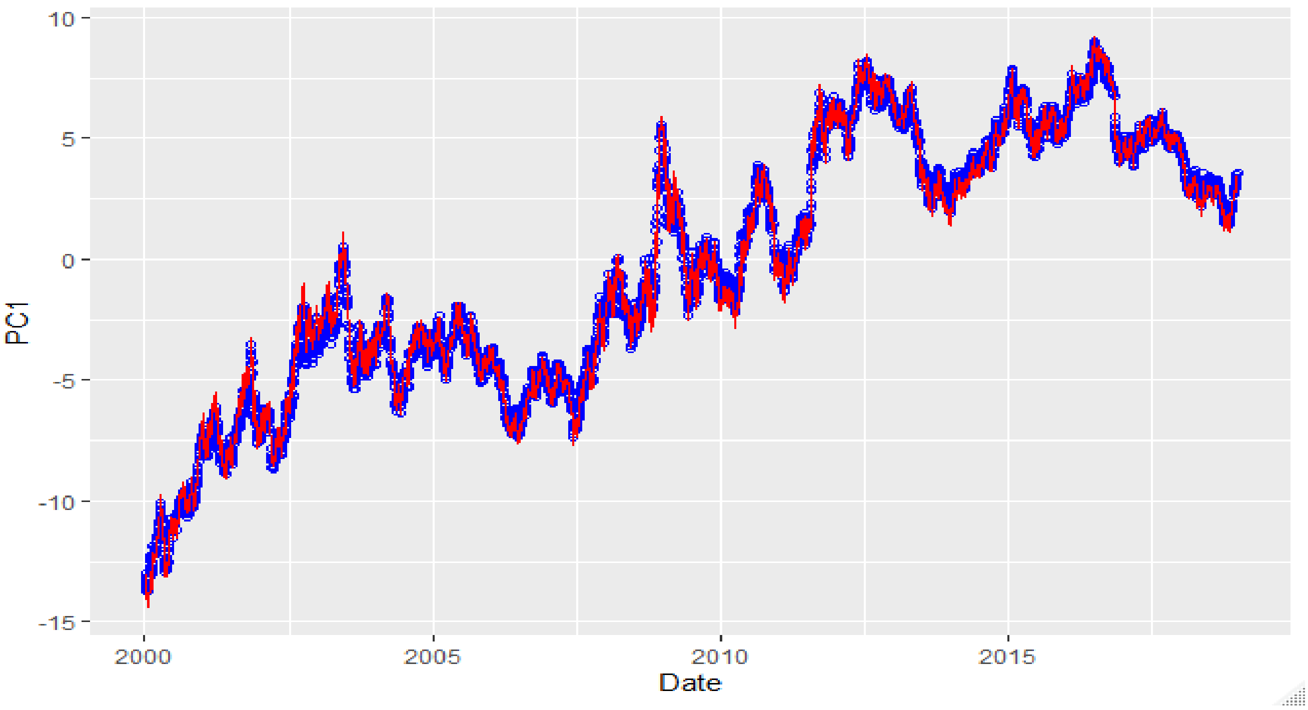


Fig 71: Observed values of PC 1 and Model Fit

In Fig 71, we see that the modelled values fit the observed data nearly perfectly.

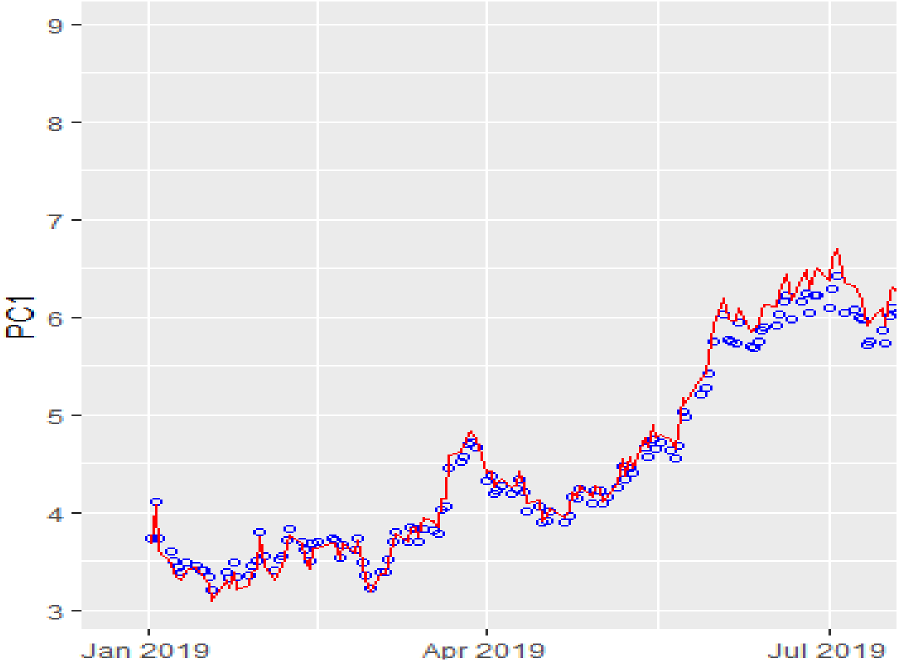
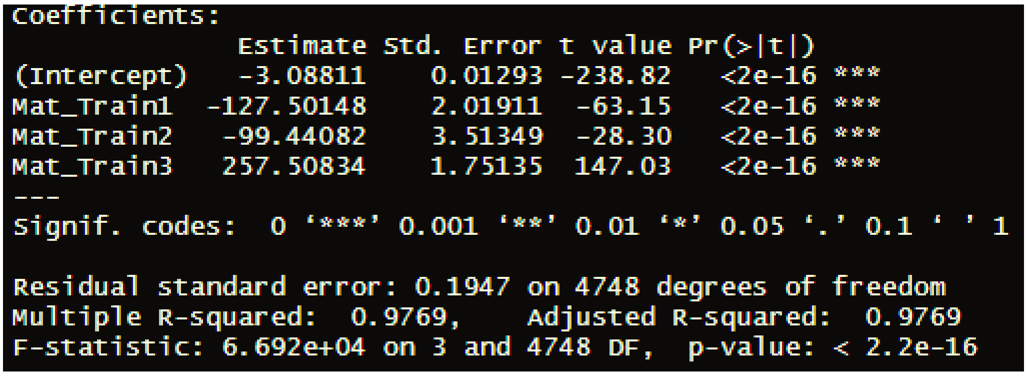


Fig 72: PC 1 model fit on Out of Sample data

In fig 72, we see that the figure for out-of-sample data is quite good and our portfolio is able to recreate PC 1 quite nicely. We repeat the similar exercise for PC 2.



```
Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)   -3.08811    0.01293 -238.82   <2e-16 ***
Mat_Train1  -127.50148    2.01911  -63.15   <2e-16 ***
Mat_Train2   -99.44082    3.51349  -28.30   <2e-16 ***
Mat_Train3   257.50834    1.75135  147.03   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.1947 on 4748 degrees of freedom
Multiple R-squared:  0.9769,	Adjusted R-squared:  0.9769 
F-statistic: 6.692e+04 on 3 and 4748 DF,  p-value: < 2.2e-16
```

Fig 73: Model fit of maturities against PC 2

The portfolio we created will have the following weights for 3 year, 5 year and 10 year maturities respectively to recreate PC 2: - 127.5, -99.44 and 257.51.

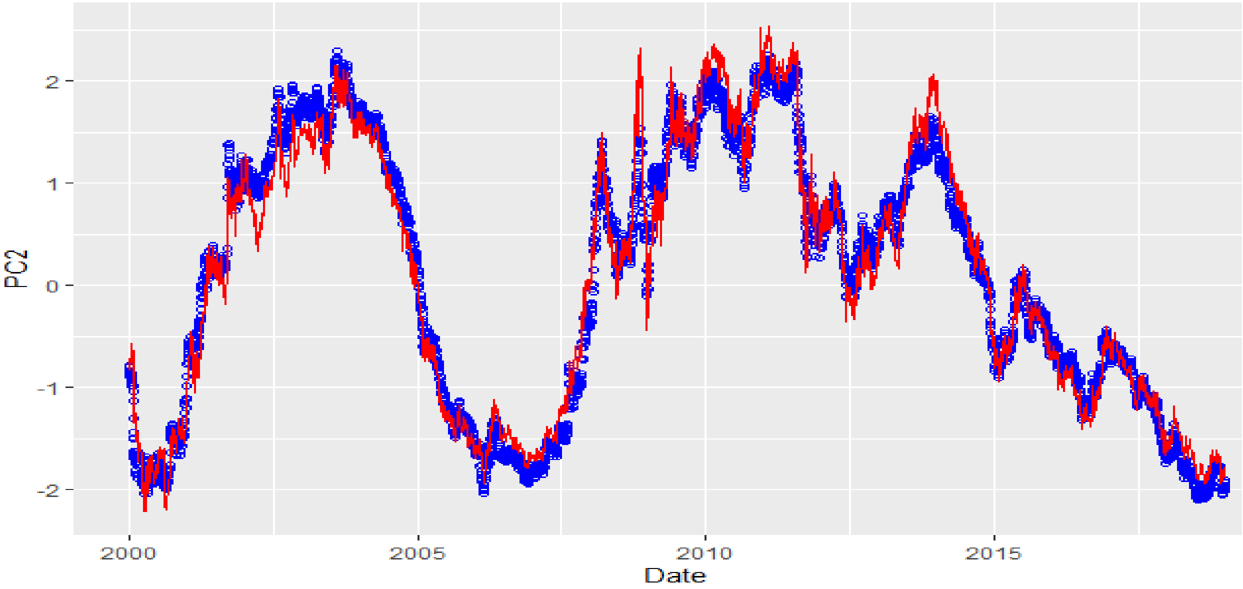


Fig 74: Observed values of PC 2 and Model Fit

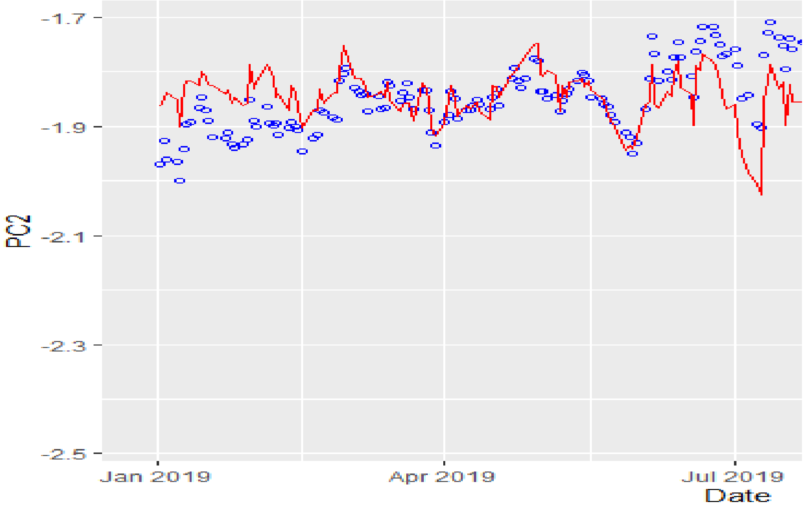
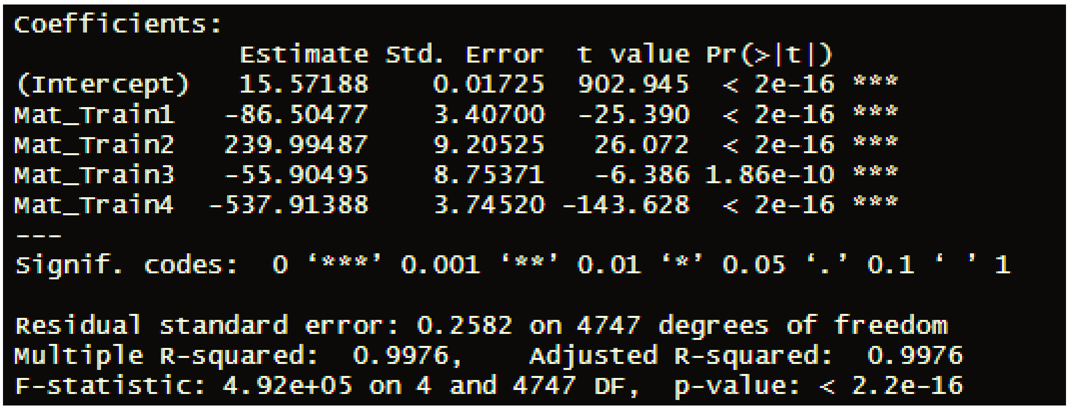


Fig 75: PC 1 model fit on Out of Sample data

Increasing the number of maturities used in regression from 3 doesn't significantly increase Adjusted R-squared as seen in Fig 76.

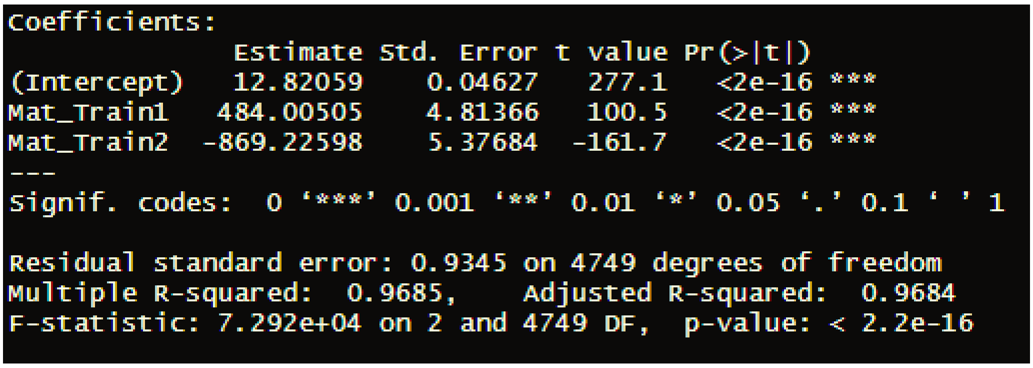


```
Coefficients:
               Estimate Std. Error  t value Pr(>|t|)    
(Intercept)    15.57188    0.01725  902.945  < 2e-16 ***
Mat_Train1    -86.50477    3.40700  -25.390  < 2e-16 ***
Mat_Train2    239.99487    9.20525   26.072  < 2e-16 ***
Mat_Train3    -55.90495    8.75371   -6.386 1.86e-10 ***
Mat_Train4   -537.91388    3.74520 -143.628  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.2582 on 4747 degrees of freedom
Multiple R-squared:  0.9976,	Adjusted R-squared:  0.9976 
F-statistic: 4.92e+05 on 4 and 4747 DF,  p-value: < 2.2e-16
```

Fig 76: Model fit of 3 year, 5 year, 7 year and 10 year maturity on PC 1

However, decreasing number of maturities from 3 to 2 significantly decreases the goodness of fit statistic as seen in Fig 77. Hence, we choose 3 maturities to create factor portfolios.



```
Coefficients:
               Estimate Std. Error t value Pr(>|t|)    
(Intercept)    12.82059    0.04627   277.1   <2e-16 ***
Mat_Train1    484.00505    4.81366   100.5   <2e-16 ***
Mat_Train2   -869.22598    5.37684  -161.7   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.9345 on 4749 degrees of freedom
Multiple R-squared:  0.9685,	Adjusted R-squared:  0.9684 
F-statistic: 7.292e+04 on 2 and 4749 DF,  p-value: < 2.2e-16
```

Fig 77: Model fit of 3 year and 5 year maturity on PC 1

10. Conclusion

In this paper, we delved on the various uses of Principal Component analysis of yield curves. We analyzed the time series of PCs from 2000 to 2019 and saw that there are two clear regimes of yields. We demonstrated how the PCs clearly affect yield curve movements in distinct ways. We discussed about the stability of relationship between the factors and yields themselves. We saw that no macro-economic indicator could effectively explain any of the PCs derived from US treasuries. However, we found that the PCs of a few developed economies are highly correlated and thus, we derived global factors that can explain movement of yield curves from all these countries. We projected the yield curves by applying ARIMA models on the PCs. We simulated the yields using Laplace distribution after finding that the distribution of percentage change in yields was leptokurtic. Finally, we created factor portfolios from yields that only responds to one particular type of movement of the yield curve.

There are several enhancement possible to this research. Some of them are mention in the respective sections. But we have tried to encapsulate all the possible avenues of Principal Components usage into one.

Appendix

Source Code in R

```
library(Quandl)
library(dplyr)
library(zoo)
library(ggplot2)
library(corrplot)
library(tseries)
library(TSA)
library(forecast)
library(PerformanceAnalytics)
library(TTR)
library(MASS)
library(LaplacesDemon)
library(NormalLaplace)
library(quantmod)
library(tidyr)
library(reshape2)

Quandl.api_key("PqwdR9zg7uFyWWv4enza")

setwd('C:/Users/HP/Desktop/WQU/Capstone Project/Code and Data')

# USA_YC = Quandl("YC/USA",type = 'raw')
# GBR_YC = Quandl("YC/GBR",type = 'raw')
# CAN_YC = Quandl("YC/CAN",type = 'raw')
# FRA_YC = Quandl("YC/FRA",type = 'raw')
# JPN_YC = Quandl("YC/JPN",type = 'raw')
# GRC_YC = Quandl("YC/GRC",type = 'raw')
# BEL_YC = Quandl("YC/BEL",type = 'raw')
# SWISS_YC = Quandl("YC/CHE",type = 'raw')
#
# save(USA_YC,GBR_YC,CAN_YC,FRA_YC,JPN_YC,GRC_YC,BEL_YC,SWISS_YC,file = "YieldCurve.RData")

# US_GDP = Quandl("FRED/GDP",type = 'raw')
# US_CPI = Quandl("RATEINF/CPI_USA",type = 'raw')
# US_Oil_WTI = Quandl("FRED/WTISPLC",type = 'raw')
# # US_Gold = Quandl("FRED/GOLDPMGBD228NLBM",type = 'raw')
# US_Gold = Quandl("LBMA/GOLD",type = 'raw')
# US_UNEM = Quandl("FRED/UNRATE",type = 'raw')
# US_IND_PRO = Quandl("FRED/INDPRO",type = 'raw')
# US_CAP_UTIL = Quandl("FRED/CAPUTLB50001SQ",type = 'raw')
# US_SP500 = read.csv(file = 'GSPC.csv')

#     save(US_GDP,US_CPI,US_Oil_WTI,US_Gold,US_UNEM,US_IND_PRO,US_CAP_UTIL,US_SP500,file     =
#'EconomicIndicators.RData')
######################################
load('YieldCurve.RData')
load('EconomicIndicators.RData')
#########################################################################
####

#Data Prepreocessing
#USA
USA_YC_OOS = USA_YC[240:1,c(1,5:12)]
USA_YC_OOS[,2:9] = USA_YC_OOS[,2:9]/100
rownames(USA_YC_OOS) = 1:nrow(USA_YC_OOS)

USA_YC = USA_YC[5242:241,c(1,5:12)]
USA_YC[,2:9] = USA_YC[,2:9]/100
rownames(USA_YC) = 1:nrow(USA_YC)

USA_YC[is.na(USA_YC[,9]),9] = USA_YC[is.na(USA_YC[,9]),8]
```

```
x = c(1,2,3,5,7,10,20,30)
# y = USA_YC[1,-1]
# spline(x,y,n=30)

USA_YC_SPline = matrix(rep(0,30*nrow(USA_YC)),nrow = nrow(USA_YC),ncol = 30)
for (i in 1:nrow(USA_YC)) {
 y = USA_YC[i,-1]
 USA_YC_SPline[i,] = spline(x,y,n=30)$y
}

plot(x,USA_YC[500,-1],type = 'p',xlab = 'Maturity',
   ylab = 'Yield', main = 'Fitiing Spline',col = 'red')
lines(c(1:30), USA_YC_SPline[500,])


#PCA of USA - Full data
date_USA = USA_YC$Date
y.pca = prcomp(USA_YC_SPline,scale = T,center = T)
cumsum(y.pca$sdev^2)/sum(y.pca$sdev^2)

#Taking the first 3 factor loadings - Full data
factor_loadings_USA = y.pca$rotation[,1:3]
PC_USA = y.pca$x[,1:3]
scale = y.pca$scale
center = y.pca$center

PCA_YC_USA = PC_USA%*%t(factor_loadings_USA)
PCA_YC_USA = apply(PCA_YC_USA,1,function(x) x*scale+center)
PCA_YC_USA = t(PCA_YC_USA)

#Plots - Output
#1. Explained variance
plot(c(1:10),(cumsum(y.pca$sdev^2)/sum(y.pca$sdev^2)*100)[1:10],ylim = c(90,100),type = 'b',xlab
= 'Principal Components',
   ylab = 'Percentage of cumulative varinace explained', main = 'Cumlative variance explained by
Pricipal Components',col = 'red',
   xaxt = 'n')
axis(side = 1, at = c(1:10))
abline(h = 100,col = 'gray',lty=2)


#2. Time Series of principal components
a = data.frame(Date = rep(date_USA,3),Values = c(PC_USA[,1],PC_USA[,2],PC_USA[,3]),
          PC = c(rep('PC 1',length(date_USA)),rep('PC 2',length(date_USA)),rep('PC 3',length(date_USA))))
ggplot(data = a) +
 geom_line(aes(x = Date, y = Values, colour = PC)) +
 geom_hline(yintercept = 0,linetype = 'dashed', color = 'grey') +
 scale_x_date(date_breaks = '2 years') +
 labs(title = "Evolution of Principal Components over time", x = "Date", y = "", color = "Principal
Components") +
 scale_color_manual(labels = c("PC 1", "PC 2", 'PC 3'), values = c("red", "green",'blue')) +
 theme_bw() +
 theme(legend.position = "top",plot.title = element_text(hjust = 0.5))

#3. PCA Fit on 4 random YCs
rand_rows = c(1035,2467,3182,4756)
rand_dates = date_USA[rand_rows]
b = data.frame(Maturity = c(1:30))
for (i in 1:4) {
 b = cbind(b,data.frame(a = USA_YC_SPline[rand_rows[i],]),data.frame(b =
PCA_YC_USA[rand_rows[i],]))
 names(b)[(i*2):(i*2+1)] <- c(paste('Observed-',rand_dates[i],sep = ''),paste(rand_dates[i],'-PCA',sep
= ''))
}

ggplot(data = b) +
 geom_line(aes(x=b[,1], y=b[,2]), color = 'red') +
 geom_point(aes(x=b[,1], y=b[,3]), color = 'red' ) +
 geom_line(aes(x=b[,1], y=b[,4]), color = 'green') +
 geom_point(aes(x=b[,1], y=b[,5]), color = 'green') +
 geom_line(aes(x=b[,1], y=b[,6]), color = 'blue') +
 geom_point(aes(x=b[,1], y=b[,7]),  color = 'blue') +
 geom_line(aes(x=b[,1], y=b[,8]), color = 'orange') +
 geom_point(aes(x=b[,1], y=b[,9]),  color = 'orange') +
 geom_label(label='24-02-2004', x=10.1,y=0.045,label.size = 0.2,color = "black",fill=NA) +
 geom_label(label='10-11-2009', x=20.1,y=0.045,label.size = 0.2,color = "black",fill=NA) +
 geom_label(label='17-09-2012', x=8.1,y=0.02,label.size = 0.2,color = "black",fill=NA) +
 geom_label(label='07-01-2019', x=15.1,y=0.03,label.size = 0.2,color = "black",fill=NA) +
 labs(title = "Randomly selected yield curves and PC fit", x = "Maturity", y = "Rate") +
 theme_bw() +
 theme(plot.title = element_text(hjust = 0.5))


#4. Correlation among PCs
corrplot(cor(PC_USA))

#5. Effect of individual PC on Yield Curves
row_num = 1000
c = data.frame(Maturity = c(1:30), YC = USA_YC_SPline[row_num,])
PC_C = PC_USA[row_num,]

PC_C_Shocked = c(PC_USA[row_num,1]+sd(PC_USA[,1]),PC_USA[row_num,2],PC_USA[row_num,3])
YC_Shocked = t(PC_C_Shocked%*%t(factor_loadings_USA))*scale+center

c = cbind(c,YC_Level = YC_Shocked)

PC_C_Shocked = c(PC_USA[row_num,1],PC_USA[row_num,2]+sd(PC_USA[,2]),PC_USA[row_num,3])
YC_Shocked = t(PC_C_Shocked%*%t(factor_loadings_USA))*scale+center

c = cbind(c,YC_Slope = YC_Shocked)

PC_C_Shocked = c(PC_USA[row_num,1],PC_USA[row_num,2],PC_USA[row_num,3]+sd(PC_USA[,3]))
YC_Shocked = t(PC_C_Shocked%*%t(factor_loadings_USA))*scale+center

c = cbind(c,YC_Curvature = YC_Shocked)

#PC1
ggplot(data = c) +
 geom_line(aes(x=c[,1], y=c[,2]), color = 'red') +
 geom_line(aes(x=c[,1], y=c[,3]), color = 'green') +
 labs(title = "1 standard deviation shock to 1st PC on 02-01-2004 yields", x = "Maturity", y = "Rate") +
 geom_label(label='Actual Yields', x=5,y=0.04,label.size = 0.2,color = "black",fill=NA) +
 geom_label(label='Shocked PC Yields', x=9,y=0.02,label.size = 0.2,color = "black",fill=NA) +
 theme_bw() +
 theme(plot.title = element_text(hjust = 0.5))
#PC2
ggplot(data = c) +
 geom_line(aes(x=c[,1], y=c[,2]), color = 'red') +
 geom_line(aes(x=c[,1], y=c[,4]), color = 'green') +
 labs(title = "1 standard deviation shock to 2nd PC on 02-01-2004 yields", x = "Maturity", y = "Rate") +
 geom_label(label='Actual Yields', x=5,y=0.04,label.size = 0.2,color = "black",fill=NA) +
 geom_label(label='Shocked PC Yields', x=7,y=0.02,label.size = 0.2,color = "black",fill=NA) +
 theme_bw() +
 theme(plot.title = element_text(hjust = 0.5))
#PC3
ggplot(data = c) +
 geom_line(aes(x=c[,1], y=c[,2]), color = 'red') +
 geom_line(aes(x=c[,1], y=c[,5]), color = 'green') +
 labs(title = "1 standard deviation shock to 3rd PC on 02-01-2004 yields", x = "Maturity", y = "Rate") +
 geom_label(label='Actual Yields', x=10.5,y=0.04,label.size = 0.2,color = "black",fill=NA) +
 geom_label(label='Shocked PC Yields', x=3,y=0.04,label.size = 0.2,color = "black",fill=NA) +
 theme_bw() +
 theme(plot.title = element_text(hjust = 0.5))


#6. Compare factor loadings Full data vs 2000-2008 end vs 2009 start-Data End
USA_YC_1Half = USA_YC_SPline[1:2251,]
USA_YC_2Half = USA_YC_SPline[2252:5002,]

Expl_Var = data.frame(Full = cumsum(y.pca$sdev^2)/sum(y.pca$sdev^2))

#y.pca = prcomp(USA_YC_SPline,scale = T,center = T)
sd_full = y.pca$sdev[1:10]

y.pca = prcomp(USA_YC_1Half,scale = T,center = T)
sd_half1 = y.pca$sdev[1:10]
Expl_Var$Half1 = cumsum(y.pca$sdev^2)/sum(y.pca$sdev^2)
factor_loadings_Half1 = y.pca$rotation[,1:3]
PC_USA_Half1 = y.pca$x[,1:3]

y.pca = prcomp(USA_YC_2Half,scale = T,center = T)
sd_half2 = y.pca$sdev[1:10]
Expl_Var$Half2 = cumsum(y.pca$sdev^2)/sum(y.pca$sdev^2)
factor_loadings_Half2 = y.pca$rotation[,1:3]
PC_USA_Half2 = y.pca$x[,1:3]

#Comparing eigen values
ggplot(data = data.frame(Full = sd_full,Half1 = sd_half1,Half2 = sd_half2)) +
 geom_line(aes(x=c(1:10), y=Full),color = 'red') +
 geom_line(aes(x=c(1:10), y=Half1),color = 'green') +
 geom_line(aes(x=c(1:10), y=Half2),color = 'blue') +
 geom_point(aes(x=c(1:10), y=Full),color = 'red') +
 geom_point(aes(x=c(1:10), y=Half1),color = 'green') +
 geom_point(aes(x=c(1:10), y=Half2),color = 'blue') +
 xlim(0,5) +
 labs(title = "Comparison of eigenvalues on different windows", x = "Eigen Vallues",
    y = "Value") +
 geom_label(label='2000-2019', x=2,y=1,label.size = 0.2,color = "black",fill=NA) +
 geom_label(label='2000-2009', x=2,y=2,label.size = 0.2,color = "black",fill=NA) +
 geom_label(label='2009-2019', x=2,y=3,label.size = 0.2,color = "black",fill=NA) +
 theme_bw() +
 theme(plot.title = element_text(hjust = 0.5))


#Explained variance chart
ggplot(data = Expl_Var) +
 geom_line(aes(x=c(1:30), y=Full),color = 'red') +
 geom_line(aes(x=c(1:30), y=Half1),color = 'green') +
 geom_line(aes(x=c(1:30), y=Half2),color = 'blue') +
 geom_point(aes(x=c(1:30), y=Full),color = 'red') +
 geom_point(aes(x=c(1:30), y=Half1),color = 'green') +
 geom_point(aes(x=c(1:30), y=Half2),color = 'blue') +
 xlim(0,5) +
 labs(title = "Comparison of cumulative explained variance on different windows", x = "Principal
Component",
    y = "Percentage of cumulative variance explained") +
 geom_label(label='2000-2019', x=.65,y=0.95,label.size = 0.2,color = "black",fill=NA) +
 geom_label(label='2000-2009', x=.6,y=0.9,label.size = 0.2,color = "black",fill=NA) +
 geom_label(label='2009-2019', x=.6,y=0.8,label.size = 0.2,color = "black",fill=NA) +
 theme_bw() +
 theme(plot.title = element_text(hjust = 0.5))

#Factor loadings comparison

#Direction 1
ggplot(data.frame(Maturity = c(1:30), Full = factor_loadings_USA[,1], Half1 =
factor_loadings_Half1[,1],Half2 = factor_loadings_Half2[,1])) +
 geom_line(aes(x = Maturity, y = Full), color = 'red') +
 geom_line(aes(x = Maturity, y = Half1), color = 'blue') +
 geom_line(aes(x = Maturity, y = Half2), color = 'green') +
 labs(title = "First Principal Direction compariosn - Full data and data split in 2 windows", x = "Maturity",
    y = "Value") +
 geom_label(label='2000-2019', x=2,y=-0.18,label.size = 0.2,color = "black",fill=NA) +
 geom_label(label='2000-2009', x=2,y=-0.12,label.size = 0.2,color = "black",fill=NA) +
 geom_label(label='2009-2019', x=5,y=-0.05,label.size = 0.2,color = "black",fill=NA) +
 theme_bw() +
 theme(plot.title = element_text(hjust = 0.5))

#Direction2
ggplot(data.frame(Maturity = c(1:30), Full = factor_loadings_USA[,2], Half1 =
factor_loadings_Half1[,2],Half2 = factor_loadings_Half2[,2])) +
 geom_line(aes(x = Maturity, y = Full), color = 'red') +
 geom_line(aes(x = Maturity, y = Half1), color = 'blue') +
 geom_line(aes(x = Maturity, y = Half2), color = 'green') +
 labs(title = "First Principal Direction compariosn - Full data and data split in 2 windows", x = "Maturity",
    y = "Value") +
 geom_label(label='2000-2019', x=12,y=0.1,label.size = 0.2,color = "black",fill=NA) +
 geom_label(label='2000-2009', x=12,y=-0.1,label.size = 0.2,color = "black",fill=NA) +
 geom_label(label='2009-2019', x=5,y=-0.4,label.size = 0.2,color = "black",fill=NA) +
 theme_bw() +
 theme(plot.title = element_text(hjust = 0.5))

#Direction3
ggplot(data.frame(Maturity = c(1:30), Full = factor_loadings_USA[,3], Half1 =
factor_loadings_Half1[,3],Half2 = factor_loadings_Half2[,3])) +
 geom_line(aes(x = Maturity, y = Full), color = 'red') +
 geom_line(aes(x = Maturity, y = Half1), color = 'blue') +
 geom_line(aes(x = Maturity, y = Half2), color = 'green') +
 labs(title = "First Principal Direction compariosn - Full data and data split in 2 windows", x = "Maturity",
    y = "Value") +
 geom_label(label='2000-2019', x=12,y=0.1,label.size = 0.2,color = "black",fill=NA) +
 geom_label(label='2000-2009', x=12,y=-0.1,label.size = 0.2,color = "black",fill=NA) +
 geom_label(label='2009-2019', x=5,y=-0.4,label.size = 0.2,color = "black",fill=NA) +
 theme_bw() +
 theme(plot.title = element_text(hjust = 0.5))


#Principal Components Comparison
#PC1
ggplot() +
 geom_line(data = data.frame(Date = date_USA,PC1 = PC_USA[,1]),aes(x = Date, y = PC1),col = 'red') +
 geom_line(data = data.frame(Date = date_USA[1:2251],PC1 = PC_USA_Half1[,1]),aes(x = Date, y =
PC1),col = 'blue') +
 geom_line(data = data.frame(Date = date_USA[2252:5002],PC1 = PC_USA_Half2[,1]),aes(x = Date, y =
PC1),col = 'green')

#PC2
ggplot() +
 geom_line(data = data.frame(Date = date_USA,PC1 = PC_USA[,2]),aes(x = Date, y = PC1),col = 'red') +
 geom_line(data = data.frame(Date = date_USA[1:2251],PC1 = PC_USA_Half1[,2]),aes(x = Date, y =
PC1),col = 'blue') +
 geom_line(data = data.frame(Date = date_USA[2252:5002],PC1 = PC_USA_Half2[,2]),aes(x = Date, y =
PC1),col = 'green')

#PC3
ggplot() +
 geom_line(data = data.frame(Date = date_USA,PC1 = PC_USA[,3]),aes(x = Date, y = PC1),col = 'red') +
 geom_line(data = data.frame(Date = date_USA[1:2251],PC1 = -PC_USA_Half1[,3]),aes(x = Date, y =
PC1),col = 'blue') +
 geom_line(data = data.frame(Date = date_USA[2252:5002],PC1 = -PC_USA_Half2[,3]),aes(x = Date, y =
PC1),col = 'green')
```

```
#######################################################################
####################################################
#####Time Series Principal Components#################################

#Take only 2013 - 2018 data and 2019 data as out of sample

USA_YC_SPline_TS = USA_YC_SPline[3253:5002,]
y.pca = prcomp(USA_YC_SPline_TS,scale = T,center = T)
y.pca$sdev[1:10]
cumsum(y.pca$sdev^2)/sum(y.pca$sdev^2)
factor_loadings_TS = y.pca$rotation[,1:3]
PC_USA_TS = y.pca$x[,1:3]
scale = y.pca$scale
center = y.pca$center


# PC_USA_Train = PC_USA[1:1500,]
# PC_USA_OOS = PC_USA[1501:1750,]
date_TS = date_USA[3253:5002]

plot(date_TS,PC_USA_TS[,1],type = 'l',main ='PC #1',xlab= 'Date', ylab = 'Value')
abline(reg=lm(PC_USA_TS[,1]~date_TS), col = 'blue')
#PC 1
Mov_Avg =  sapply(SMA(PC_USA_TS[,1],n=50), function(x) ifelse(is.na(x),0,x))
a = PC_USA_TS[,1] - sapply(SMA(PC_USA_TS[,1],n=50), function(x) ifelse(is.na(x),0,x))

plot(date_TS,a,type = 'l',main ='PC #1',xlab= 'Date', ylab = 'Value')
abline(reg=lm(a~date_TS), col = 'blue')
#lines(date_train,SMA(PC_USA_Train[,1],n=50))

#a = PC_USA_Train[,1] - sapply(SMA(PC_USA_Train[,1],n=50), function(x) ifelse(is.na(x),0,x))

PC1_Train = a[1:1500]
PC1_Test = a[-c(1:1500)]

acf(PC1_Train)
pacf(PC1_Train)

# acf(PC1_Train^2)
# pacf(PC1_Train^2)

adf.test(PC1_Train)
#adf.test(diff(a))

# acf(diff(PC_USA_Train[,1]))
# pacf(diff(PC_USA_Train[,1]))

auto.arima(PC1_Train)

plot(PC1_Train)
#plot(diff(PC_USA_Train[,1],lag = 300))
PC1_Model = arima(PC1_Train,order = c(2,0,0))
summary(PC1_Model)
PC1_pred <- predict(PC1_Model, n.ahead = 100)
plot(c(PC1_Train,PC1_Test), col = 'blue')
lines(c(rep(NA,length(PC1_Train)),PC1_pred$pred), col = 'red')

plot(c(PC_USA_Train[,1],PC1_pred$pred))
lines(c(rep(0,length(PC_USA_Train[,1]))))

McLeod.Li.test(PC1_Model)

acf(PC1_Model$residuals)

plot(date_TS[1:1500],PC1_Model$residuals)
```

```
qqnorm(PC1_Model$residuals,xlab = 'Theoritical Quantiles', ylab = 'Sample Quantiles', main = 'QQPlot
of PCA 1 Model')
qqline(PC1_Model$residuals, distribution = qnorm,probs = c(0.25, 0.75), qtype = 7)

#plot.ts(SMA(PC_USA_Train[,1], n=200))

# acf(diff(PC_USA_Train[,1]))
# pacf(diff(PC_USA_Train[,1]))

plot(date_TS[1:1600],c(PC1_Train,PC1_Test[1:100])+Mov_Avg[1:1600], type = 'p', col = 'blue', cex = .5)
lines(date_TS[1501:1600],PC1_pred$pred+Mov_Avg[1501:1600],col= 'red', lty = 2, lwd = 2)

#PC2
plot(date_TS,PC_USA_TS[,2],type = 'l',main ='PC #2',xlab= 'Date', ylab = 'Value')
abline(reg=lm(PC_USA_TS[,2]~date_TS), col = 'blue')

Mov_Avg =  sapply(SMA(PC_USA_TS[,2],n=50), function(x) ifelse(is.na(x),0,x))
a = PC_USA_TS[,2] - sapply(SMA(PC_USA_TS[,2],n=50), function(x) ifelse(is.na(x),0,x))

plot(date_TS,a,main ='PC #2',xlab= 'Date', ylab = 'Value', type ='l')
abline(reg=lm(a~date_TS), col = 'blue')
#lines(date_train,SMA(PC_USA_Train[,1],n=50))

#a = PC_USA_Train[,1] - sapply(SMA(PC_USA_Train[,1],n=50), function(x) ifelse(is.na(x),0,x))

PC2_Train = a[1:1500]
PC2_Test = a[-c(1:1500)]

acf(PC2_Train)
pacf(PC2_Train)

adf.test(PC2_Train)
#adf.test(diff(PC_USA_Train[,2]))

auto.arima(PC2_Train)
plot(PC2_Train)
#plot(diff(PC_USA_Train[,1],lag = 300))
PC2_Model = arima(PC2_Train,order = c(1,0,0))
summary(PC2_Model)
PC2_pred <- predict(PC2_Model, n.ahead = 100)
plot(c(PC2_Train,PC2_Test))
lines(c(rep(0,length(PC2_Train)),PC2_pred$pred), col = 'red')

McLeod.Li.test(PC2_Model)

acf(PC2_Model$residuals)

plot(date_TS[1:1500],PC2_Model$residuals)

qqnorm(PC2_Model$residuals,xlab = 'Theoritical Quantiles', ylab = 'Sample Quantiles', main = 'QQPlot
of PCA 1 Model')
qqline(PC2_Model$residuals, distribution = qnorm,probs = c(0.25, 0.75), qtype = 7)

plot(date_TS[1:1600],c(PC2_Train,PC2_Test[1:100])+Mov_Avg[1:1600], type = 'p', col = 'blue', cex = .5)
lines(date_TS[1501:1600],PC2_pred$pred+Mov_Avg[1501:1600],col= 'red', lty = 2, lwd = 2)

#PC 3

plot(date_TS,PC_USA_TS[,3],type = 'l',main ='PC #3',xlab= 'Date', ylab = 'Value')
abline(reg=lm(PC_USA_TS[,3]~date_TS), col = 'blue')

Mov_Avg =  sapply(SMA(PC_USA_TS[,3],n=50), function(x) ifelse(is.na(x),0,x))
a = PC_USA_TS[,3] - sapply(SMA(PC_USA_TS[,3],n=50), function(x) ifelse(is.na(x),0,x))

plot(date_TS,a,main ='PC #3',xlab= 'Date', ylab = 'Value',type ='l')
abline(reg=lm(a~date_TS), col = 'blue')
```

```
#lines(date_train,SMA(PC_USA_Train[,1],n=50))

#a = PC_USA_Train[,1] - sapply(SMA(PC_USA_Train[,1],n=50), function(x) ifelse(is.na(x),0,x))

PC3_Train = a[1:1500]
PC3_Test = a[-c(1:1500)]

acf(PC3_Train)
pacf(PC3_Train)

adf.test(PC3_Train)
#adf.test(diff(PC_USA_Train[,2]))

auto.arima(PC3_Train)
plot(PC3_Train)
#plot(diff(PC_USA_Train[,1],lag = 300))
PC3_Model = arima(PC3_Train,order = c(1,0,1))
summary(PC3_Model)
PC3_pred <- predict(PC3_Model, n.ahead = 100)
plot(c(PC3_Train,PC3_Test))
lines(c(rep(0,length(PC3_Train)),PC3_pred$pred), col = 'red')

McLeod.Li.test(PC3_Model)

acf(PC3_Model$residuals)

plot(date_train,PC3_Model$residuals)

qqnorm(PC3_Model$residuals,xlab = 'Theoritical Quantiles', ylab = 'Sample Quantiles', main = 'QQPlot
of PCA 1 Model')
qqline(PC3_Model$residuals, distribution = qnorm,probs = c(0.25, 0.75), qtype = 7)


plot(date_TS[1:1600],c(PC3_Train,PC3_Test[1:100])+Mov_Avg[1:1600], type = 'p', col = 'blue', cex = .5)
lines(date_TS[1501:1600],PC3_pred$pred+Mov_Avg[1501:1600],col= 'red', lty = 2, lwd = 2)

#Recombination
PC_Pred = cbind(PC1_pred$pred,PC2_pred$pred,PC3_pred$pred)
PCA_YC_Pred = PC_Pred%*%t(factor_loadings_TS)
PCA_YC_Pred = apply(PCA_YC_Pred,1,function(x) x*scale+center)
PCA_YC_Pred = t(PCA_YC_Pred)

#Test 10th year maturity
plot(date_TS[1501:1600],USA_YC_SPline_TS[1501:1600,10],ylim = c(0,0.05))
lines(date_TS[1501:1600],PCA_YC_Pred[,10], col='red')

#Test 20th year maturity
plot(date_TS[1501:1600],USA_YC_SPline_TS[1501:1600,20],ylim = c(0,0.05))
lines(date_TS[1501:1600],PCA_YC_Pred[,20], col='red')

##############Frequency Distribution of Yields and Percentage Change in
Yields################################
date_USA = USA_YC$Date
y.pca = prcomp(USA_YC_SPline,scale = T,center = T)
cumsum(y.pca$sdev^2)/sum(y.pca$sdev^2)
factor_loadings_YC = y.pca$rotation[,1:3]
PC_YC = y.pca$x[,1:3]
scale_YC = y.pca$scale
center_YC = y.pca$center

hist(PC_YC[,1], breaks = 50)
hist(PC_YC[,2], breaks = 50)
hist(PC_YC[,3], breaks = 50)

PC_PCYC = cbind(diff(PC_YC[,1])/PC_YC[-length(PC_YC[,1]),1],diff(PC_YC[,2])/PC_YC[-
length(PC_YC[,2]),2],
           diff(PC_YC[,3])/PC_YC[-length(PC_YC[,3]),3])

PCYC_USA = apply(USA_YC_SPline,2,function(x) diff(x)/x[-length(x)])
y.pca = prcomp(PCYC_USA,scale = T,center = T)
cumsum(y.pca$sdev^2)/sum(y.pca$sdev^2)
factor_loadings_PCYC = y.pca$rotation[,1:3]
PC_PCYC = y.pca$x[,1:3]
scale_PCYC = y.pca$scale
center_PCYC = y.pca$center

hist(PC_PCYC[,1], breaks = 50)
hist(PC_PCYC[,2], breaks = 50)
hist(PC_PCYC[,3], breaks = 50)

qqnorm(PC_PCYC[,1],xlab = 'Theoritical Quantiles', ylab = 'Sample Quantiles', main = 'QQPlot of YCPC
PCA 1')
qqline(PC_PCYC[,1], distribution = qnorm,probs = c(0.25, 0.75), qtype = 7)

qqnorm(PC_PCYC[,2],xlab = 'Theoritical Quantiles', ylab = 'Sample Quantiles', main = 'QQPlot of YCPC
PCA 2')
qqline(PC_PCYC[,2], distribution = qnorm,probs = c(0.25, 0.75), qtype = 7)

qqnorm(PC_PCYC[,3],xlab = 'Theoritical Quantiles', ylab = 'Sample Quantiles', main = 'QQPlot of YCPC
PCA 3')
qqline(PC_PCYC[,3], distribution = qnorm,probs = c(0.25, 0.75), qtype = 7)

h = hist(PC_PCYC[,1], breaks = 50)
lines(seq(min(PC_PCYC[,1]),max(PC_PCYC[,1]),length = 400),
    dnorm(seq(min(PC_PCYC[,1]),max(PC_PCYC[,1]),length = 400),
        mean = mean(PC_PCYC[,1]),
        sd = sd(PC_PCYC[,1]))*diff(h$mids[1:2])*length(PC_PCYC[,1]), col = 'red')

h = hist(PC_PCYC[,2], breaks = 50)
lines(seq(min(PC_PCYC[,2]),max(PC_PCYC[,2]),length = 400),
    dnorm(seq(min(PC_PCYC[,2]),max(PC_PCYC[,2]),length = 400),
        mean = mean(PC_PCYC[,2]),
        sd = sd(PC_PCYC[,2]))*diff(h$mids[1:2])*length(PC_PCYC[,2]), col = 'red')

h = hist(PC_PCYC[,3], breaks = 50)
lines(seq(min(PC_PCYC[,3]),max(PC_PCYC[,3]),length = 400),
    dnorm(seq(min(PC_PCYC[,3]),max(PC_PCYC[,3]),length = 400),
        mean = mean(PC_PCYC[,3]),
        sd = sd(PC_PCYC[,3]))*diff(h$mids[1:2])*length(PC_PCYC[,3]), col = 'red')

print(paste("PCYC PC 1 : Mean=",round(mean(PC_PCYC[,1]),2),", St Dev=",round(sd(PC_PCYC[,1]),2),",
Skewness=", round(skewness(PC_PCYC[,1]),2),
        "Kurtosis= ", round(kurtosis(PC_PCYC[,1],method = 'excess'),2)))

print(paste("PCYC PC 2 : Mean=",round(mean(PC_PCYC[,2]),2),", St Dev=",round(sd(PC_PCYC[,2]),2),",
Skewness=", round(skewness(PC_PCYC[,2]),2),
        "Kurtosis= ", round(kurtosis(PC_PCYC[,2],method = 'excess'),2)))

print(paste("PCYC PC 3 : Mean=",round(mean(PC_PCYC[,3]),2),", St Dev=",round(sd(PC_PCYC[,2]),2),",
Skewness=", round(skewness(PC_PCYC[,3]),2),
        "Kurtosis= ", round(kurtosis(PC_PCYC[,3],method = 'excess'),2)))

#Simulate using Normal
PCA1_Norm = rnorm(1000, mean = mean(PC_PCYC[,1]), sd = sd(PC_PCYC[,1]))
Q_PC1 = quantile(PC_PCYC[,1], probs = c(1:9)/10)
Q_PC1 = rbind(Q_PC1,quantile(PCA1_Norm, probs = c(1:9)/10))

PCA2_Norm = rnorm(1000, mean = mean(PC_PCYC[,2]), sd = sd(PC_PCYC[,2]))
Q_PC2 = quantile(PC_PCYC[,2], probs = c(1:9)/10)
Q_PC2 = rbind(Q_PC2,quantile(PCA2_Norm, probs = c(1:9)/10))

PCA3_Norm = rnorm(1000, mean = mean(PC_PCYC[,3]), sd = sd(PC_PCYC[,3]))
```

```
Q_PC3 = quantile(PC_PCYC[,3], probs = c(1:9)/10)
Q_PC3 = rbind(Q_PC3,quantile(PCA3_Norm, probs = c(1:9)/10))

PCA_Norm = cbind(PCA1_Norm,PCA2_Norm,PCA3_Norm)
PCYC_Norm = PCA_Norm%*%t(factor_loadings_PCYC)
PCYC_Norm = apply(PCYC_Norm,1,function(x) x*scale_PCYC+center_PCYC)
PCYC_Norm = t(PCYC_NOrm)

#Simulating yield curves using last date actual yield curve
YC_Norm = t(apply(PCYC_Norm + 1, 2, function(x) x*USA_YC_SPline[5002,]))

#Laplace distribution

bestfit = function(x) {
 loc_array = seq(from = mean(x)-2,to = mean(x)+2, length.out = 100)
 sca_array = seq(from = .001,to = 10, length.out = 100)
 sumsq_min = 1000000
 sca_opt = 0
 loc_opt = 0
 for (i in 1:100) {
  for (j in 1:100) {
   sumsq = sum((quantile(rlaplace(1000, location = loc_array[i], scale = sca_array[j]),probs =
c(1:99)/100) - quantile(x,probs = c(1:99)/100))^2)
   if (sumsq<sumsq_min) {
    sumsq_min = sumsq
    sca_opt = sca_array[j]
    loc_opt = loc_array[i]
   }
  }
 }
 return(c(loc_opt,sca_opt))
}

param1 = bestfit(PC_PCYC[,1])
h = hist(PC_PCYC[,1], breaks = 50)
lines(seq(min(PC_PCYC[,1]),max(PC_PCYC[,1]),length = 400),
   dlaplace(seq(min(PC_PCYC[,1]),max(PC_PCYC[,1]),length = 400),
        location = param1[1], scale = param1[2])*diff(h$mids[1:2])*length(PC_PCYC[,1]), col = 'red')

qqplot(qlaplace(ppoints(1000),scale = param1[2], location = param1[1]), PC_PCYC[,1],
    main = "Laplace Q-Q plot",xlab = "Theoretical quantiles", ylab = "Sample quantiles")
abline(c(0,1), col = "red", lwd = 2)

param2 = bestfit(PC_PCYC[,2])
h = hist(PC_PCYC[,2], breaks = 50)
lines(seq(min(PC_PCYC[,2]),max(PC_PCYC[,2]),length = 400),
   dlaplace(seq(min(PC_PCYC[,2]),max(PC_PCYC[,2]),length = 400),
        location = param2[1], scale = param2[2])*diff(h$mids[1:2])*length(PC_PCYC[,2]), col = 'red')
qqplot(qlaplace(ppoints(1000),scale = param2[2], location = param2[1]), PC_PCYC[,2],
    main = "Laplace Q-Q plot",xlab = "Theoretical quantiles", ylab = "Sample quantiles")
abline(c(0,1), col = "red", lwd = 2)

param3 = bestfit(PC_PCYC[,3])
h = hist(PC_PCYC[,3], breaks = 50)
lines(seq(min(PC_PCYC[,3]),max(PC_PCYC[,3]),length = 400),
   dlaplace(seq(min(PC_PCYC[,3]),max(PC_PCYC[,3]),length = 400),
        location = param3[1], scale = param3[2])*diff(h$mids[1:2])*length(PC_PCYC[,3]), col = 'red')
qqplot(qlaplace(ppoints(1000),scale = param3[2], location = param3[1]), PC_PCYC[,3],
    main = "Laplace Q-Q plot",xlab = "Theoretical quantiles", ylab = "Sample quantiles")
abline(c(0,1), col = "red", lwd = 2)

PCA_Laplace = cbind(rlaplace(1000,location = param1[1],scale = param1[2]),
         rlaplace(1000,location = param2[1],scale = param2[2]),
         rlaplace(1000,location = param3[1],scale = param3[2]))
PCYC_Laplace = PCA_Laplace%*%t(factor_loadings_PCYC)
PCYC_Laplace = apply(PCYC_Laplace,1,function(x) x*scale_PCYC+center_PCYC)
PCYC_Laplace = t(PCYC_Laplace)

YC_Laplace = t(apply(PCYC_Laplace + 1, 1, function(x) x*USA_YC_SPline[5002,]))

Lap_Plot = YC_Laplace
Lap_Plot = as.data.frame(Lap_Plot)
colnames(Lap_Plot) = c(1:30)
Lap_Plot$Sims = rownames(Lap_Plot)
Lap_Plot = melt(Lap_Plot, id.vars="Sims")
Lap_Plot$Maturity = as.numeric(gsub("time", "", Lap_Plot$variable))

ggplot(Lap_Plot, aes(x=Maturity, y=value, group=Sims)) +
 theme_bw() +
 theme(panel.grid=element_blank()) +
 geom_line(size=0.2, alpha=0.1)
######################################################################
#####################################################
######## Create single factor
portfolios##########################################################
########

Date_USA = USA_YC$Date
y.pca = prcomp(USA_YC_SPline,scale = T,center = T)
cumsum(y.pca$sdev^2)/sum(y.pca$sdev^2)
factor_loadings_USA = y.pca$rotation[,1:3]
PC_USA = y.pca$x[,1:3]
scale = y.pca$scale
center = y.pca$center
# PCA_YC_USA = PC_USA%*%t(factor_loadings_USA)
# PCA_YC_USA = apply(PCA_YC_USA,1,function(x) x*scale+center)
# PCA_YC_USA = t(PCA_YC_USA)

#Regressing against maturities
#PCA1
PCA1_Train = PC_USA[1:4752,1]
PCA1_Test = PC_USA[4753:5002,1]
Maturities = cbind(USA_YC$`3-Year`,USA_YC$`5-Year`,USA_YC$`10-Year`)
Mat_Train = Maturities[1:4752,]
Mat_Test = Maturities[4753:5002,]
Factor_Model1 = lm(PCA1_Train ~ Mat_Train)
summary(Factor_Model1)

plot(Date_USA[1:4752],Factor_Model1$residuals,
    #ylim = c(90,100),
    type = 'l', xlab = 'Date',ylab = '', main = 'Regression Residuals',col = 'black')

PCA1_Model = Factor_Model1$coefficients[1] + Factor_Model1$coefficients[2]*Mat_Train[,1] +
Factor_Model1$coefficients[3]*Mat_Train[,2] + Factor_Model1$coefficients[4]*Mat_Train[,3]

ggplot() +
 geom_point(data = data.frame(Date = date_USA[1:4752],PC1 = PCA1_Train),aes(x = Date, y = PC1),col
= 'blue',shape = 1) +
 geom_line(data = data.frame(Date = date_USA[1:4752],PC1_Model = PCA1_Model),aes(x = Date, y =
PC1_Model),col = 'red',size = 0.05)

PCA1_Model_OOS = Factor_Model1$coefficients[1] + Factor_Model1$coefficients[2]*Mat_Test[,1] +
Factor_Model1$coefficients[3]*Mat_Test[,2] + Factor_Model1$coefficients[4]*Mat_Test[,3]

ggplot() +
 geom_point(data = data.frame(Date = date_USA[4753:5002],PC1 = PCA1_Test),aes(x = Date, y =
PC1),col = 'blue',shape = 1) +
 geom_line(data = data.frame(Date = date_USA[4753:5002],PC1_Model_OOS = PCA1_Model_OOS),aes(x
= Date, y = PC1_Model_OOS),col = 'red',size = 0.05)

plot(Date_USA[4753:5002],PCA1_Test - PCA1_Model_OOS,
    #ylim = c(90,100),
    type = 'l', xlab = 'Date',ylab = '', main = 'Regression Residuals',col = 'black')

#PCA2
PCA2_Train = PC_USA[1:4752,2]
PCA2_Test = PC_USA[4753:5002,2]
Maturities = cbind(USA_YC$`3-Year`,USA_YC$`5-Year`,USA_YC$`10-Year`)
Mat_Train = Maturities[1:4752,]
Mat_Test = Maturities[4753:5002,]
Factor_Model2 = lm(PCA2_Train ~ Mat_Train)
summary(Factor_Model2)

plot(Date_USA[1:4752],Factor_Model2$residuals,
    #ylim = c(90,100),
    type = 'l', xlab = 'Date',ylab = '', main = 'Regression Residuals',col = 'black')

PCA2_Model = Factor_Model2$coefficients[1] + Factor_Model2$coefficients[2]*Mat_Train[,1] +
Factor_Model2$coefficients[3]*Mat_Train[,2] + Factor_Model2$coefficients[4]*Mat_Train[,3]

ggplot() +
 geom_point(data = data.frame(Date = date_USA[1:4752],PC2 = PCA2_Train),aes(x = Date, y = PC2),col
= 'blue',shape = 1) +
 geom_line(data = data.frame(Date = date_USA[1:4752],PC2_Model = PCA2_Model),aes(x = Date, y =
PC2_Model),col = 'red',size = 0.05)

PCA2_Model_OOS = Factor_Model2$coefficients[1] + Factor_Model2$coefficients[2]*Mat_Test[,1] +
Factor_Model2$coefficients[3]*Mat_Test[,2] + Factor_Model2$coefficients[4]*Mat_Test[,3]

ggplot() +
 geom_point(data = data.frame(Date = date_USA[4753:5002],PC2 = PCA2_Test),aes(x = Date, y =
PC2),col = 'blue',shape = 1) +
 geom_line(data = data.frame(Date = date_USA[4753:5002],PC2_Model_OOS = PCA2_Model_OOS),aes(x
= Date, y = PC2_Model_OOS),col = 'red',size = 0.05)

plot(Date_USA[4753:5002],PCA2_Test - PCA2_Model_OOS,
    #ylim = c(90,100),
    type = 'l', xlab = 'Date',ylab = '', main = 'Regression Residuals',col = 'black')


# PCA1_Model_onPCA2 = Factor_Model1$coefficients[1] + Factor_Model1$coefficients[2]*PC_USA[,2] +
Factor_Model1$coefficients[3]*PC_USA[,2] + Factor_Model1$coefficients[4]*PC_USA[,2]
# ggplot() +
#    geom_point(data = data.frame(Date = date_USA,PC2 = PC_USA[,2]),aes(x = Date, y = PC2),col =
'blue',shape = 1) +
#    geom_line(data = data.frame(Date = date_USA,PC1_Model_onPC2 = PCA1_Model_onPCA2),aes(x =
Date, y = PC1_Model_onPC2),col = 'red',size = 0.05)


#Increasing number of maturitues doesnt increase Rsquare
PCA1_Train = PC_USA[1:4752,1]
Maturities = cbind(USA_YC$`3-Year`,USA_YC$`5-Year`,USA_YC$`7-Year`,USA_YC$`10-Year`)
Mat_Train = Maturities[1:4752,]
Factor_Model1 = lm(PCA1_Train ~ Mat_Train)
summary(Factor_Model1)
#Decreasing number of maturitues doesnt decreases Rsquare
PCA1_Train = PC_USA[1:4752,1]
Maturities = cbind(USA_YC$`3-Year`,USA_YC$`5-Year`)
Mat_Train = Maturities[1:4752,]
Factor_Model1 = lm(PCA1_Train ~ Mat_Train)
summary(Factor_Model1)

#######################################################################
############################
########################    Regress factors against macro-economic variables
########################
#Data Preprocessing
US_CAP_UTIL = US_CAP_UTIL[215:1,]
```

```
rownames(US_CAP_UTIL) = 1:nrow(US_CAP_UTIL)

US_CPI = US_CPI[1295:1,]
rownames(US_CPI) = 1:nrow(US_CPI)

US_GDP = US_GDP[295:1,]
rownames(US_GDP) = 1:nrow(US_GDP)

US_IND_PRO = US_IND_PRO[1223:1,]
rownames(US_IND_PRO) = 1:nrow(US_IND_PRO)

US_Oil_WTI = US_Oil_WTI[898:1,]
rownames(US_Oil_WTI) = 1:nrow(US_Oil_WTI)

US_SP500 = US_SP500[10087:1,]
rownames(US_SP500) = 1:nrow(US_SP500)

US_UNEM = US_UNEM[875:1,]
rownames(US_UNEM) = 1:nrow(US_UNEM)

US_Gold = US_Gold[13396:1,1:2]
rownames(US_Gold) = 1:nrow(US_Gold)

US_YC_PCA = data.frame(Date = date_USA, PC_USA)

#Cutoff Date - 1st Jan 2000 - 31 Dec 2019
US_CAP_UTIL = US_CAP_UTIL[133:213,]
rownames(US_CAP_UTIL) = 1:nrow(US_CAP_UTIL)

US_CPI = US_CPI[1045:1285,]
rownames(US_CPI) = 1:nrow(US_CPI)

US_GDP = US_GDP[213:293,]
rownames(US_GDP) = 1:nrow(US_GDP)

US_IND_PRO = US_IND_PRO[973:1213,]
rownames(US_IND_PRO) = 1:nrow(US_IND_PRO)

US_Oil_WTI = US_Oil_WTI[649:889,]
rownames(US_Oil_WTI) = 1:nrow(US_Oil_WTI)

US_SP500 = US_SP500[4815:9846,]
rownames(US_SP500) = 1:nrow(US_SP500)
US_SP500 = US_SP500[,c(1,6)]
US_SP500$Date = as.Date(US_SP500$Date)

US_UNEM = US_UNEM[625:865,]
rownames(US_UNEM) = 1:nrow(US_UNEM)

US_Gold = US_Gold[8090:13144,]
rownames(US_Gold) = 1:nrow(US_Gold)

########### Quaterly Returns %############################
QtrChange = function(x) {
 a = xts(x[,2],x[,1])
 return(quarterlyReturn(a))
}

CAP_UTIL = as.data.frame(QtrChange(US_CAP_UTIL))*100
CAP_UTIL$Date = rownames(CAP_UTIL)
CAP_UTIL = CAP_UTIL[-1,]

CPI = as.data.frame(QtrChange(US_CPI))*100
CPI$Date = rownames(CPI)
CPI = CPI[-nrow(CPI),]
```

```
GDP = as.data.frame(QtrChange(US_GDP))*100
GDP$Date = rownames(GDP)
GDP = GDP[-1,]

IND_PRO = as.data.frame(QtrChange(US_IND_PRO))*100
IND_PRO$Date = rownames(IND_PRO)
IND_PRO = IND_PRO[-nrow(IND_PRO),]

Oil = as.data.frame(QtrChange(US_Oil_WTI))*100
Oil$Date = rownames(Oil)
Oil = Oil[-nrow(Oil),]

SP500 = as.data.frame(QtrChange(US_SP500))*100
SP500$Date = rownames(SP500)
SP500 = SP500[-nrow(SP500),]

UNEM = as.data.frame(QtrChange(US_UNEM))*100
UNEM$Date = rownames(UNEM)
UNEM = UNEM[-nrow(UNEM),]

Gold = as.data.frame(QtrChange(US_Gold))*100
Gold$Date = rownames(Gold)
Gold = Gold[-nrow(Gold),]

US_YC_PCA =
cbind(as.data.frame(QtrChange(US_YC_PCA[,c(1,2)]))*100,as.data.frame(QtrChange(US_YC_PCA[,c(1,3
)]))*100,
      as.data.frame(QtrChange(US_YC_PCA[,c(1,4)]))*100)
colnames(US_YC_PCA) = c("PCA1","PCA2","PCA3")
US_YC_PCA$Date = rownames(US_YC_PCA)

#PCA 1 correlations
paste("Correlations b/w PC 1 and GDP (Quaterly %age change)=
",round(cor(US_YC_PCA$PCA1,GDP$quarterly.returns),2))
paste("Correlations b/w PC 1 and Gold (Quaterly %age change)=
",round(cor(US_YC_PCA$PCA1,Gold$quarterly.returns),2))
paste("Correlations b/w PC 1 and CPI (Quaterly %age change)=
",round(cor(US_YC_PCA$PCA1,CPI$quarterly.returns),2))
paste("Correlations b/w PC 1 and WTI Oil (Quaterly %age change)=
",round(cor(US_YC_PCA$PCA1,Oil$quarterly.returns),2))
paste("Correlations b/w PC 1 and S&P 500 (Quaterly %age change)=
",round(cor(US_YC_PCA$PCA1,SP500$quarterly.returns),2))
paste("Correlations b/w PC 1 and Unemployment (Quaterly %age change)=
",round(cor(US_YC_PCA$PCA1,UNEM$quarterly.returns),2))
paste("Correlations b/w PC 1 and Industrial Production (Quaterly %age change)=
",round(cor(US_YC_PCA$PCA1,IND_PRO$quarterly.returns),2))
paste("Correlations b/w PC 1 and Capacity Utilisation (Quaterly %age change)=
",round(cor(US_YC_PCA$PCA1,CAP_UTIL$quarterly.returns),2))

#PCA 2 correlations
paste("Correlations b/w PC 2 and GDP (Quaterly %age change)=
",round(cor(US_YC_PCA$PCA2,GDP$quarterly.returns),2))
paste("Correlations b/w PC 2 and Gold (Quaterly %age change)=
",round(cor(US_YC_PCA$PCA2,Gold$quarterly.returns),2))
paste("Correlations b/w PC 2 and CPI (Quaterly %age change)=
",round(cor(US_YC_PCA$PCA2,CPI$quarterly.returns),2))
paste("Correlations b/w PC 2 and WTI Oil (Quaterly %age change)=
",round(cor(US_YC_PCA$PCA2,Oil$quarterly.returns),2))
paste("Correlations b/w PC 2 and S&P 500 (Quaterly %age change)=
",round(cor(US_YC_PCA$PCA2,SP500$quarterly.returns),2))
paste("Correlations b/w PC 2 and Unemployment (Quaterly %age change)=
",round(cor(US_YC_PCA$PCA2,UNEM$quarterly.returns),2))
paste("Correlations b/w PC 2 and Industrial Production (Quaterly %age change)=
",round(cor(US_YC_PCA$PCA2,IND_PRO$quarterly.returns),2))
paste("Correlations b/w PC 2 and Capacity Utilisation (Quaterly %age change)=
",round(cor(US_YC_PCA$PCA2,CAP_UTIL$quarterly.returns),2))
```

```
#Plots PC 1
plot(US_YC_PCA$PCA1,GDP$quarterly.returns,type = 'p',xlab = 'PC 1',ylab = 'GDP', xlim = c(-100,100))
lines(US_YC_PCA$PCA1, predict.lm(lm(GDP$quarterly.returns ~ poly(US_YC_PCA$PCA1, 1)), newdata =
list(x = US_YC_PCA$PCA1)), col = 2)  ## add regression curve (colour: red)
legend("bottomleft",legend = paste("Adjusted R-squared =
",round(summary(lm(GDP$quarterly.returns ~ poly(US_YC_PCA$PCA1, 1)))$adj.r.squared,2),sep = ''),
    pt.cex = 1, cex = 1,text.col = "black")

plot(US_YC_PCA$PCA1,Gold$quarterly.returns,type = 'p',xlab = 'PC 1',ylab = 'Gold', xlim = c(-100,100))
lines(US_YC_PCA$PCA1, predict.lm(lm(Gold$quarterly.returns ~ poly(US_YC_PCA$PCA1, 1)), newdata =
list(x = US_YC_PCA$PCA1)), col = 2)  ## add regression curve (colour: red)
legend("bottomleft",legend = paste("Adjusted R-squared =
",round(summary(lm(Gold$quarterly.returns ~ poly(US_YC_PCA$PCA1, 1)))$adj.r.squared,2),sep = ''),
    pt.cex = 1, cex = 1,text.col = "black")

plot(US_YC_PCA$PCA1,CPI$quarterly.returns,type = 'p',xlab = 'PC 1',ylab = 'CPI',xlim = c(-100,100))
lines(US_YC_PCA$PCA1, predict.lm(lm(CPI$quarterly.returns ~ poly(US_YC_PCA$PCA1, 1)), newdata =
list(x = US_YC_PCA$PCA1)), col = 2)  ## add regression curve (colour: red)
legend("bottomleft",legend = paste("Adjusted R-squared =
",round(summary(lm(CPI$quarterly.returns ~ poly(US_YC_PCA$PCA1, 1)))$adj.r.squared,2),sep = ''),
    pt.cex = 1, cex = 1,text.col = "black")

plot(US_YC_PCA$PCA1,Oil$quarterly.returns,type = 'p',xlab = 'PC 1',ylab = 'WTI Oil',xlim = c(-100,100))
lines(US_YC_PCA$PCA1, predict.lm(lm(Oil$quarterly.returns ~ poly(US_YC_PCA$PCA1, 1)), newdata =
list(x = US_YC_PCA$PCA1)), col = 2)  ## add regression curve (colour: red)
legend("bottomleft",legend = paste("Adjusted R-squared = ",round(summary(lm(Oil$quarterly.returns
~ poly(US_YC_PCA$PCA1, 1)))$adj.r.squared,2),sep = ''),
    pt.cex = 1, cex = 1,text.col = "black")

plot(US_YC_PCA$PCA1,SP500$quarterly.returns,type = 'p',xlab = 'PC 1',ylab = 'S&P 500',xlim = c(-
100,100))
lines(US_YC_PCA$PCA1, predict.lm(lm(SP500$quarterly.returns ~ poly(US_YC_PCA$PCA1, 1)), newdata
= list(x = US_YC_PCA$PCA1)), col = 2)  ## add regression curve (colour: red)
legend("bottomleft",legend = paste("Adjusted R-squared =
",round(summary(lm(SP500$quarterly.returns ~ poly(US_YC_PCA$PCA1, 1)))$adj.r.squared,2),sep =
''),
    pt.cex = 1, cex = 1,text.col = "black")

plot(US_YC_PCA$PCA1,UNEM$quarterly.returns,type = 'p',xlab = 'PC 1',ylab = 'Unemployment
Rate',xlim = c(-100,100))
lines(US_YC_PCA$PCA1, predict.lm(lm(UNEM$quarterly.returns ~ poly(US_YC_PCA$PCA1, 1)), newdata
= list(x = US_YC_PCA$PCA1)), col = 2)  ## add regression curve (colour: red)
legend("bottomleft",legend = paste("Adjusted R-squared =
",round(summary(lm(UNEM$quarterly.returns ~ poly(US_YC_PCA$PCA1, 1)))$adj.r.squared,2),sep = ''),
    pt.cex = 1, cex = 1,text.col = "black")

plot(US_YC_PCA$PCA1,IND_PRO$quarterly.returns,type = 'p',xlab = 'PC 1',ylab = 'Industrial
Production',xlim = c(-100,100))
lines(US_YC_PCA$PCA1, predict.lm(lm(IND_PRO$quarterly.returns ~ poly(US_YC_PCA$PCA1, 1)),
newdata = list(x = US_YC_PCA$PCA1)), col = 2)  ## add regression curve (colour: red)
legend("bottomleft",legend = paste("Adjusted R-squared =
",round(summary(lm(IND_PRO$quarterly.returns ~ poly(US_YC_PCA$PCA1, 1)))$adj.r.squared,2),sep
= ''),
    pt.cex = 1, cex = 1,text.col = "black")

plot(US_YC_PCA$PCA1,CAP_UTIL$quarterly.returns,type = 'p',xlab = 'PC 1',ylab = 'Capacity
Utilisation',xlim = c(-100,100))
lines(US_YC_PCA$PCA1, predict.lm(lm(CAP_UTIL$quarterly.returns ~ poly(US_YC_PCA$PCA1, 1)),
newdata = list(x = US_YC_PCA$PCA1)), col = 2)  ## add regression curve (colour: red)
legend("bottomleft",legend = paste("Adjusted R-squared =
",round(summary(lm(CAP_UTIL$quarterly.returns ~ poly(US_YC_PCA$PCA1, 1)))$adj.r.squared,2),sep
= ''),
    pt.cex = 1, cex = 1,text.col = "black")

#Plots PC 2
```

```
plot(US_YC_PCA$PCA2,GDP$quarterly.returns,type = 'p',xlab = 'PC 2',ylab = 'GDP', xlim = c(-100,100))
lines(US_YC_PCA$PCA2, predict.lm(lm(GDP$quarterly.returns ~ poly(US_YC_PCA$PCA2, 1)), newdata =
list(x = US_YC_PCA$PCA2)), col = 2)  ## add regression curve (colour: red)
legend("bottomleft",legend = paste("Adjusted R-squared =
",round(summary(lm(GDP$quarterly.returns ~ poly(US_YC_PCA$PCA2, 1)))$adj.r.squared,2),sep = ''),
    pt.cex = 1, cex = 1,text.col = "black")

plot(US_YC_PCA$PCA2,Gold$quarterly.returns,type = 'p',xlab = 'PC 2',ylab = 'Gold', xlim = c(-100,100))
lines(US_YC_PCA$PCA2, predict.lm(lm(Gold$quarterly.returns ~ poly(US_YC_PCA$PCA2, 1)), newdata =
list(x = US_YC_PCA$PCA2)), col = 2)  ## add regression curve (colour: red)
legend("bottomleft",legend = paste("Adjusted R-squared =
",round(summary(lm(Gold$quarterly.returns ~ poly(US_YC_PCA$PCA2, 1)))$adj.r.squared,2),sep = ''),
    pt.cex = 1, cex = 1,text.col = "black")

plot(US_YC_PCA$PCA2,CPI$quarterly.returns,type = 'p',xlab = 'PC 2',ylab = 'CPI',xlim = c(-100,100))
lines(US_YC_PCA$PCA2, predict.lm(lm(CPI$quarterly.returns ~ poly(US_YC_PCA$PCA2, 1)), newdata =
list(x = US_YC_PCA$PCA2)), col = 2)  ## add regression curve (colour: red)
legend("bottomleft",legend = paste("Adjusted R-squared =
",round(summary(lm(CPI$quarterly.returns ~ poly(US_YC_PCA$PCA2, 1)))$adj.r.squared,2),sep = ''),
    pt.cex = 1, cex = 1,text.col = "black")

plot(US_YC_PCA$PCA2,Oil$quarterly.returns,type = 'p',xlab = 'PC 2',ylab = 'WTI Oil',xlim = c(-100,100))
lines(US_YC_PCA$PCA2, predict.lm(lm(Oil$quarterly.returns ~ poly(US_YC_PCA$PCA2, 1)), newdata =
list(x = US_YC_PCA$PCA2)), col = 2)  ## add regression curve (colour: red)
legend("bottomleft",legend = paste("Adjusted R-squared = ",round(summary(lm(Oil$quarterly.returns
~ poly(US_YC_PCA$PCA2, 1)))$adj.r.squared,2),sep = ''),
    pt.cex = 1, cex = 1,text.col = "black")

plot(US_YC_PCA$PCA2,SP500$quarterly.returns,type = 'p',xlab = 'PC 2',ylab = 'S&P 500',xlim = c(-
100,100))
lines(US_YC_PCA$PCA2, predict.lm(lm(SP500$quarterly.returns ~ poly(US_YC_PCA$PCA2, 1)), newdata
= list(x = US_YC_PCA$PCA2)), col = 2)  ## add regression curve (colour: red)
legend("bottomleft",legend = paste("Adjusted R-squared =
",round(summary(lm(SP500$quarterly.returns ~ poly(US_YC_PCA$PCA2, 1)))$adj.r.squared,2),sep =
''),
    pt.cex = 1, cex = 1,text.col = "black")

plot(US_YC_PCA$PCA2,UNEM$quarterly.returns,type = 'p',xlab = 'PC 2',ylab = 'Unemployment
Rate',xlim = c(-100,100))
lines(US_YC_PCA$PCA2, predict.lm(lm(UNEM$quarterly.returns ~ poly(US_YC_PCA$PCA2, 1)), newdata
= list(x = US_YC_PCA$PCA2)), col = 2)  ## add regression curve (colour: red)
legend("bottomleft",legend = paste("Adjusted R-squared =
",round(summary(lm(UNEM$quarterly.returns ~ poly(US_YC_PCA$PCA2, 1)))$adj.r.squared,2),sep = ''),
    pt.cex = 1, cex = 1,text.col = "black")

plot(US_YC_PCA$PCA2,IND_PRO$quarterly.returns,type = 'p',xlab = 'PC 2',ylab = 'Industrial
Production',xlim = c(-100,100))
lines(US_YC_PCA$PCA2, predict.lm(lm(IND_PRO$quarterly.returns ~ poly(US_YC_PCA$PCA2, 1)),
newdata = list(x = US_YC_PCA$PCA2)), col = 2)  ## add regression curve (colour: red)
legend("bottomleft",legend = paste("Adjusted R-squared =
",round(summary(lm(IND_PRO$quarterly.returns ~ poly(US_YC_PCA$PCA2, 1)))$adj.r.squared,2),sep
= ''),
    pt.cex = 1, cex = 1,text.col = "black")

plot(US_YC_PCA$PCA2,CAP_UTIL$quarterly.returns,type = 'p',xlab = 'PC 2',ylab = 'Capacity
Utilisation',xlim = c(-100,100))
lines(US_YC_PCA$PCA2, predict.lm(lm(CAP_UTIL$quarterly.returns ~ poly(US_YC_PCA$PCA2, 1)),
newdata = list(x = US_YC_PCA$PCA2)), col = 2)  ## add regression curve (colour: red)
legend("bottomleft",legend = paste("Adjusted R-squared =
",round(summary(lm(CAP_UTIL$quarterly.returns ~ poly(US_YC_PCA$PCA2, 1)))$adj.r.squared,2),sep
= ''),
    pt.cex = 1, cex = 1,text.col = "black")

#Regression
Reg_Data_PC1 = data.frame(PC = US_YC_PCA$PCA1,
                 GDP = GDP$quarterly.returns,
                 Gold = Gold$quarterly.returns,
                 CPI = CPI$quarterly.returns,
                 Oil = Oil$quarterly.returns,
                 SP500 = SP500$quarterly.returns,
                 Unem = UNEM$quarterly.returns,
                 IndPro = IND_PRO$quarterly.returns,
                 CapUtil = CAP_UTIL$quarterly.returns)
#Econ_Model_1 = lm(PC~. ,Reg_Data_PC1)
Econ_Model_1 = lm(PC~. - SP500 - Gold - IndPro - Oil - Unem,Reg_Data_PC1)
summary(Econ_Model_1)

Reg_Data_PC2 = data.frame(PC = US_YC_PCA$PCA2,
                 GDP = GDP$quarterly.returns,
                 Gold = Gold$quarterly.returns,
                 CPI = CPI$quarterly.returns,
                 Oil = Oil$quarterly.returns,
                 SP500 = SP500$quarterly.returns,
                 Unem = UNEM$quarterly.returns,
                 IndPro = IND_PRO$quarterly.returns,
                 CapUtil = CAP_UTIL$quarterly.returns)
#Econ_Model_2 = lm(PC~.,Reg_Data_PC2)
Econ_Model_2 = lm(PC~Unem ,Reg_Data_PC2)
summary(Econ_Model_2)

#########################################################################
############
######## Find PCs of different countries ##########################################

#USA, CAN, JPN, SWISS, FRANCE
load('YieldCurve.RData')
#USA
USA_YC = USA_YC[3993:241,c(1,5:12)]
USA_YC[,2:9] = USA_YC[,2:9]/100
rownames(USA_YC) = 1:nrow(USA_YC)
USA_YC[is.na(USA_YC[,9]),9] = USA_YC[is.na(USA_YC[,9]),8]
x = c(1,2,3,5,7,10,20,30)
USA_YC_SPline = matrix(rep(0,30*nrow(USA_YC)),nrow = nrow(USA_YC),ncol = 30)
for (i in 1:nrow(USA_YC)) {
 y = USA_YC[i,-1]
 USA_YC_SPline[i,] = spline(x,y,n=30)$y
}
date_USA = USA_YC$Date
y.pca = prcomp(USA_YC_SPline,scale = T,center = T)
factor_loadings_USA = y.pca$rotation[,1:3]
PC_USA = y.pca$x[,1:3]
scale = y.pca$scale
center = y.pca$center
PCA_YC_USA = PC_USA%*%t(factor_loadings_USA)
PCA_YC_USA = apply(PCA_YC_USA,1,function(x) x*scale+center)
PCA_YC_USA = t(PCA_YC_USA)
plot(c(1:10),(cumsum(y.pca$sdev^2)/sum(y.pca$sdev^2)*100)[1:10],ylim = c(90,100),type = 'b',xlab
= 'Principal Components',
    ylab = 'Percentage of cumulative varinace explained', main = 'Cumlative variance explained by
Pricipal Components',col = 'red',
    xaxt = 'n')
axis(side = 1, at = c(1:10))
abline(h = 100,col = 'gray',lty=2)
Expl_Var = data.frame(USA=(cumsum(y.pca$sdev^2)/sum(y.pca$sdev^2)*100)[1:10])
a = data.frame(Date = rep(date_USA,3),Values = c(PC_USA[,1],PC_USA[,2],PC_USA[,3]),PC = c(rep('PC
1',length(date_USA)),rep('PC 2',length(date_USA)),rep('PC 3',length(date_USA))))
ggplot(data = a) +
 geom_line(aes(x = Date, y = Values, colour = PC)) +
 geom_hline(yintercept = 0,linetype = 'dashed', color = 'grey') +
 scale_x_date(date_breaks = '2 years') +
 labs(title = "Evolution of Principal Components over time (USA)", x = "Date", y = "", color = "Principal
Components") +
 scale_color_manual(labels = c("PC 1", "PC 2", 'PC 3'), values = c("red", "green",'blue')) +
 theme_bw() +
 theme(legend.position = "top",plot.title = element_text(hjust = 0.5))
# rand_rows = c(1035,2467,3182)
# rand_dates = date_USA[rand_rows]
# b = data.frame(Maturity = c(1:30))
# for (i in 1:3) {
# b = cbind(b,data.frame(a = USA_YC_SPline[rand_rows[i],]),data.frame(b =
PCA_YC_USA[rand_rows[i],]))
# names(b)[(i*2):(i*2+1)] <- c(paste('Observed-',rand_dates[i],sep = ''),paste(rand_dates[i],'-PCA',sep
= ''))
# }
# ggplot(data = b) +
#  geom_line(aes(x=b[,1], y=b[,2]), color = 'red') +
#  geom_point(aes(x=b[,1], y=b[,3]), color = 'red' ) +
#  geom_line(aes(x=b[,1], y=b[,4]), color = 'green') +
#  geom_point(aes(x=b[,1], y=b[,5]), color = 'green') +
#  geom_line(aes(x=b[,1], y=b[,6]), color = 'blue') +
#  geom_point(aes(x=b[,1], y=b[,7]),  color = 'blue') +
#  # geom_line(aes(x=b[,1], y=b[,8]), color = 'orange') +
#  # geom_point(aes(x=b[,1], y=b[,9]),  color = 'orange') +
#  geom_label(label='24-02-2004', x=10.1,y=0.045,label.size = 0.2,color = "black",fill=NA) +
#  geom_label(label='10-11-2009', x=20.1,y=0.045,label.size = 0.2,color = "black",fill=NA) +
#  geom_label(label='17-09-2012', x=8.1,y=0.02,label.size = 0.2,color = "black",fill=NA) +
#  # geom_label(label='07-01-2019', x=15.1,y=0.03,label.size = 0.2,color = "black",fill=NA) +
#  labs(title = "Randomly selected yield curves and PC fit", x = "Maturity", y = "Rate") +
#  theme_bw() +
#  theme(plot.title = element_text(hjust = 0.5))

#Japan
JPN_YC = JPN_YC[3906:233,c(1:15)]
JPN_YC[,2:15] = JPN_YC[,2:15]/100
rownames(JPN_YC) = 1:nrow(JPN_YC)
JPN_YC[is.na(JPN_YC[,14]),14] = JPN_YC[is.na(JPN_YC[,14]),13]
x = c(1:10,15,20,25,30)
JPN_YC_SPline = matrix(rep(0,30*nrow(JPN_YC)),nrow = nrow(JPN_YC),ncol = 30)
for (i in 1:nrow(JPN_YC)) {
 y = JPN_YC[i,-1]
 JPN_YC_SPline[i,] = spline(x,y,n=30)$y
}
date_JPN = JPN_YC$Date
y.pca = prcomp(JPN_YC_SPline,scale = T,center = T)
factor_loadings_JPN = y.pca$rotation[,1:3]
PC_JPN = y.pca$x[,1:3]
scale = y.pca$scale
center = y.pca$center
PCA_YC_JPN = PC_JPN%*%t(factor_loadings_JPN)
PCA_YC_JPN = apply(PCA_YC_JPN,1,function(x) x*scale+center)
PCA_YC_JPN = t(PCA_YC_JPN)
plot(c(1:10),(cumsum(y.pca$sdev^2)/sum(y.pca$sdev^2)*100)[1:10],ylim = c(90,100),type = 'b',xlab
= 'Principal Components',
   ylab = 'Percentage of cumulative varinace explained', main = 'Cumlative variance explained by
Pricipal Components',col = 'red',
   xaxt = 'n')
axis(side = 1, at = c(1:10))
abline(h = 100,col = 'gray',lty=2)
Expl_Var$JPN = (cumsum(y.pca$sdev^2)/sum(y.pca$sdev^2)*100)[1:10]
a = data.frame(Date = rep(date_JPN,3),Values = c(-PC_JPN[,1],PC_JPN[,2],PC_JPN[,3]),PC = c(rep('PC
1',length(date_JPN)),rep('PC 2',length(date_JPN)),rep('PC 3',length(date_JPN))))
ggplot(data = a) +
 geom_line(aes(x = Date, y = Values, colour = PC)) +
 geom_hline(yintercept = 0,linetype = 'dashed', color = 'grey') +
 scale_x_date(date_breaks = '2 years') +
 labs(title = "Evolution of Principal Components over time (Japan)", x = "Date", y = "", color = "Principal
Components") +
 scale_color_manual(labels = c("PC 1", "PC 2", 'PC 3'), values = c("red", "green",'blue')) +
 theme_bw() +
 theme(legend.position = "top",plot.title = element_text(hjust = 0.5))


#CANADA
CAN_YC = CAN_YC[3992:241,c(1,5:11)]
CAN_YC[,2:8] = CAN_YC[,2:8]/100
rownames(CAN_YC) = 1:nrow(CAN_YC)
CAN_YC[is.na(CAN_YC[,2]),2] = 0
CAN_YC[is.na(CAN_YC[,3]),3] = CAN_YC[is.na(CAN_YC[,3]),2]
CAN_YC[is.na(CAN_YC[,4]),4] = CAN_YC[is.na(CAN_YC[,4]),3]
CAN_YC[is.na(CAN_YC[,5]),5] = CAN_YC[is.na(CAN_YC[,5]),4]
CAN_YC[is.na(CAN_YC[,6]),6] = CAN_YC[is.na(CAN_YC[,6]),5]
CAN_YC[is.na(CAN_YC[,7]),7] = CAN_YC[is.na(CAN_YC[,7]),6]
CAN_YC[is.na(CAN_YC[,8]),8] = CAN_YC[is.na(CAN_YC[,8]),7]
x = c(1,2,3,5,7,10,30)
CAN_YC_SPline = matrix(rep(0,30*nrow(CAN_YC)),nrow = nrow(CAN_YC),ncol = 30)
for (i in 1:nrow(CAN_YC)) {
 y = CAN_YC[i,-1]
 CAN_YC_SPline[i,] = spline(x,y,n=30)$y
}
date_CAN = CAN_YC$Date
y.pca = prcomp(CAN_YC_SPline,scale = T,center = T)
factor_loadings_CAN = y.pca$rotation[,1:3]
PC_CAN = y.pca$x[,1:3]
scale = y.pca$scale
center = y.pca$center
PCA_YC_CAN = PC_CAN%*%t(factor_loadings_CAN)
PCA_YC_CAN = apply(PCA_YC_CAN,1,function(x) x*scale+center)
PCA_YC_CAN = t(PCA_YC_CAN)
plot(c(1:10),(cumsum(y.pca$sdev^2)/sum(y.pca$sdev^2)*100)[1:10],ylim = c(90,100),type = 'b',xlab
= 'Principal Components',
    ylab = 'Percentage of cumulative varinace explained', main = 'Cumlative variance explained by
Pricipal Components',col = 'red',
    xaxt = 'n')
axis(side = 1, at = c(1:10))
abline(h = 100,col = 'gray',lty=2)
Expl_Var$CAN = (cumsum(y.pca$sdev^2)/sum(y.pca$sdev^2)*100)[1:10]
a = data.frame(Date = rep(date_CAN,3),Values = c(PC_CAN[,1],PC_CAN[,2],PC_CAN[,3]),PC = c(rep('PC
1',length(date_CAN)),rep('PC 2',length(date_CAN)),rep('PC 3',length(date_CAN))))
ggplot(data = a) +
 geom_line(aes(x = Date, y = Values, colour = PC)) +
 geom_hline(yintercept = 0,linetype = 'dashed', color = 'grey') +
 scale_x_date(date_breaks = '2 years') +
 labs(title = "Evolution of Principal Components over time (Canada)", x = "Date", y = "", color =
"Principal Components") +
 scale_color_manual(labels = c("PC 1", "PC 2", 'PC 3'), values = c("red", "green",'blue')) +
 theme_bw() +
 theme(legend.position = "top",plot.title = element_text(hjust = 0.5)) +
 ylim(-15,15)

#SWISS
SWISS_YC = SWISS_YC[4071:244,c(1,6:14)]
SWISS_YC[,2:10] = SWISS_YC[,2:10]/100
rownames(SWISS_YC) = 1:nrow(SWISS_YC)
SWISS_YC[is.na(SWISS_YC[,2]),2] = SWISS_YC[is.na(SWISS_YC[,2]),3]
SWISS_YC[is.na(SWISS_YC[,2]),2] = 0
SWISS_YC[is.na(SWISS_YC[,3]),3] = SWISS_YC[is.na(SWISS_YC[,3]),2]
SWISS_YC[is.na(SWISS_YC[,4]),4] = SWISS_YC[is.na(SWISS_YC[,4]),3]
SWISS_YC[is.na(SWISS_YC[,5]),5] = SWISS_YC[is.na(SWISS_YC[,5]),4]
SWISS_YC[is.na(SWISS_YC[,6]),6] = SWISS_YC[is.na(SWISS_YC[,6]),5]
SWISS_YC[is.na(SWISS_YC[,7]),7] = SWISS_YC[is.na(SWISS_YC[,7]),6]
SWISS_YC[is.na(SWISS_YC[,8]),8] = SWISS_YC[is.na(SWISS_YC[,8]),7]
SWISS_YC[is.na(SWISS_YC[,9]),9] = SWISS_YC[is.na(SWISS_YC[,9]),8]
SWISS_YC[is.na(SWISS_YC[,10]),10] = SWISS_YC[is.na(SWISS_YC[,10]),9]
x = c(1,2,3,4,5,7,10,20,30)
```

```
SWISS_YC_SPline = matrix(rep(0,30*nrow(SWISS_YC)),nrow = nrow(SWISS_YC),ncol = 30)
for (i in 1:nrow(SWISS_YC)) {
 y = SWISS_YC[i,-1]
 SWISS_YC_SPline[i,] = spline(x,y,n=30)$y
}
date_SWISS = SWISS_YC$Date
y.pca = prcomp(SWISS_YC_SPline,scale = T,center = T)
factor_loadings_SWISS = y.pca$rotation[,1:3]
PC_SWISS = y.pca$x[,1:3]
scale = y.pca$scale
center = y.pca$center
PCA_YC_SWISS = PC_SWISS%*%t(factor_loadings_SWISS)
PCA_YC_SWISS = apply(PCA_YC_SWISS,1,function(x) x*scale+center)
PCA_YC_SWISS = t(PCA_YC_SWISS)
plot(c(1:10),(cumsum(y.pca$sdev^2)/sum(y.pca$sdev^2)*100)[1:10],ylim = c(90,100),type = 'b',xlab
= 'Principal Components',
    ylab = 'Percentage of cumulative varinace explained', main = 'Cumlative variance explained by
Pricipal Components',col = 'red',
    xaxt = 'n')
axis(side = 1, at = c(1:10))
abline(h = 100,col = 'gray',lty=2)
Expl_Var$SWISS = (cumsum(y.pca$sdev^2)/sum(y.pca$sdev^2)*100)[1:10]
a = data.frame(Date = rep(date_SWISS,3),Values = c(PC_SWISS[,1],PC_SWISS[,2],PC_SWISS[,3]),PC =
c(rep('PC 1',length(date_SWISS)),rep('PC 2',length(date_SWISS)),rep('PC 3',length(date_SWISS))))
ggplot(data = a) +
 geom_line(aes(x = Date, y = Values, colour = PC)) +
 geom_hline(yintercept = 0,linetype = 'dashed', color = 'grey') +
 scale_x_date(date_breaks = '2 years') +
 labs(title = "Evolution of Principal Components over time (Switzerland)", x = "Date", y = "", color =
"Principal Components") +
 scale_color_manual(labels = c("PC 1", "PC 2", 'PC 3'), values = c("red", "green",'blue')) +
 theme_bw() +
 theme(legend.position = "top",plot.title = element_text(hjust = 0.5)) +
 ylim(-15,15)

#Explained Variance Chart
Expl_Var = gather(Expl_Var, Country, ExplVar, USA:SWISS)
Expl_Var$PC = rep(c(1:10),4)
ggplot(data = Expl_Var) +
 geom_line(aes(x = PC,y=ExplVar, color = Country)) +
 scale_x_continuous(breaks = c(1:10)) +
 theme_bw() +
 theme(legend.position = 'top',plot.title = element_text(hjust = 0.5))+
 labs(title = "Explained Variance of PCs for different countries", x = "PC #", y = "Explained Variance")

#Taking common dates
date_PC = Reduce(intersect, list(date_USA,date_CAN,date_JPN,date_SWISS))

row_USA = sapply(date_PC, function(x) which(date_USA==x))
PC_USA = PC_USA[row_USA,]
date_USA = date_USA[row_USA]

row_CAN = sapply(date_PC, function(x) which(date_CAN==x))
PC_CAN = PC_CAN[row_CAN,]
date_CAN = date_CAN[row_CAN]

row_JPN = sapply(date_PC, function(x) which(date_JPN==x))
PC_JPN = PC_JPN[row_JPN,]
date_JPN = date_JPN[row_JPN]

row_SWISS = sapply(date_PC, function(x) which(date_SWISS==x))
PC_SWISS = PC_SWISS[row_SWISS,]
date_SWISS = date_SWISS[row_SWISS]

#Checking Correlation
PC1 = data.frame(USA = PC_USA[,1],CAN = PC_CAN[,1], JPN = -PC_JPN[,1], SWISS = PC_SWISS[,1])
```

```
corrplot(cor(PC1), type = 'upper', method = 'number')

PC2 = data.frame(USA = PC_USA[,2],CAN = PC_CAN[,2], JPN = PC_JPN[,2], SWISS = PC_SWISS[,2])
corrplot(cor(PC2), type = 'upper', method = 'number')

PC3 = data.frame(USA = PC_USA[,3],CAN = PC_CAN[,3], JPN = PC_JPN[,3], SWISS = PC_SWISS[,3])
corrplot(cor(PC3), type = 'upper', method = 'number')

PCs = data.frame(USA_1 = PC_USA[,1],CAN_1 = PC_CAN[,1], JPN_1 = PC_JPN[,1], SWISS_1 =
PC_SWISS[,1],
            USA_2 = PC_USA[,2],CAN_2 = PC_CAN[,2], JPN_2 = PC_JPN[,2], SWISS_2 = PC_SWISS[,2],
            USA_3 = PC_USA[,3],CAN_3 = PC_CAN[,3], JPN_3 = PC_JPN[,3], SWISS_3 = PC_SWISS[,3])
corrplot(cor(PCs), type = 'full', method = 'number')

#Creating global PC
y.pca = prcomp(PCs,scale = T,center = T)
factor_loadings_Global = y.pca$rotation[,1:6]
PC_Global = y.pca$x[,1:6]
scale = y.pca$scale
center = y.pca$center
PCA_YC_Global = PC_Global%*%t(factor_loadings_Global)
PCA_YC_Global = apply(PCA_YC_Global,1,function(x) x*scale+center)
PCA_YC_Global = t(PCA_YC_Global)
plot(c(1:10),(cumsum(y.pca$sdev^2)/sum(y.pca$sdev^2)*100)[1:10],ylim = c(0,100),type = 'b',xlab =
'Principal Components',
    ylab = 'Percentage of cumulative varinace explained', main = 'Cumlative variance explained by global
Pricipal Components',col = 'red',
    xaxt = 'n')
axis(side = 1, at = c(1:10))
abline(h = 100,col = 'gray',lty=2)
a = data.frame(Date = rep(date_PC,6),
           Values = c(PC_Global[,1],PC_Global[,2],PC_Global[,3],PC_Global[,4],PC_Global[,5],PC_Global[,6]),
           PC = c(rep('PC 1',length(date_PC)),rep('PC 2',length(date_PC)),rep('PC
3',length(date_PC)),rep('PC 4',length(date_PC)),rep('PC 5',length(date_PC)),rep('PC
6',length(date_PC))))
a$Date = as.Date(date_PC)
ggplot(data = a) +
  geom_line(aes(x = Date, y = Values, colour = PC)) +
  geom_hline(yintercept = 0,linetype = 'dashed', color = 'grey') +
  scale_x_date(date_breaks = '2 years') +
  labs(title = "Evolution of Principal Components over time", x = "Date", y = "", color = "Principal
Components") +
  scale_color_manual(labels = c("PC 1", "PC 2", 'PC 3', 'PC 4', 'PC 5', 'PC 6'), values = c("red",
"green",'blue','violet','orange','grey')) +
  theme_bw() +
  theme(legend.position = "top",plot.title = element_text(hjust = 0.5)) +
  ylim(-5,5)
```

References

[1] Ang, A. and Piazzesi, M., 2003. A no-arbitrage vector autoregression of term structure dynamics with macroeconomic and latent variables. Journal of Monetary economics, 50(4), pp.745-787.

[2] Barber, J.R. and Copper, M.L., 2012. Principal component analysis of yield curve movements. Journal of Economics and Finance, 36(3), pp.750-765.

[3] Bliss, R.R., 1997. Testing term structure estimation methods. Advances in Futures and Options research, 9, pp.197-232.

[4] Carcano, N., 2009. Yield curve risk management: adjusting principal component analysis for model errors. The Journal of Risk, 12(1), p.3.

[5] Carcano, N. and Hakim, D.O., 2011. Alternative models for hedging yield curve risk: An empirical comparison. Journal of Banking & Finance, 35(11), pp.2991-3000.

[6] Falkenstein, E. and Hanweck, J., 1997. Minimizing basis risk from non-parallel shifts in the yield curve Part II: Principal Components. Journal of fixed income, 7, pp.85-90.

[7] Golub, B.W. and Tilman, L.M., 1997. Measuring yield curve risk using principal components analysis, value at risk, and key rate durations. Journal of Portfolio Management, 23(4), p.72.

[8] Gürkaynak, R.S., Sack, B. and Swanson, E., 2005. The sensitivity of long-term interest rates to economic news: Evidence and implications for macroeconomic models. American economic review, 95(1), pp.425-436.

[9] Jamshidian, F. and Zhu, Y., 1996. Scenario simulation: Theory and methodology. Finance and stochastics, 1(1), pp.43-67.

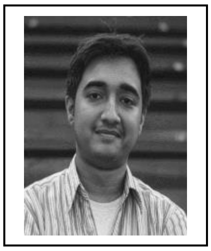
[10] Litterman, R. and Scheinkman, J., 1991. Common factors affecting bond returns. Journal of fixed income, 1(1), pp.54-61.

[11] Novosyolov, A. and Satchkov, D., 2008. Global term structure modelling using principal component analysis. Journal of Asset Management, 9(1), pp.49-60.

[12] Reisman, H. and Zohar, G., 2004. Short-term predictability of the term structure. The Journal of Fixed Income, 14(3), pp.7-14.

[13] Redfern, D. and McLean, D., 2014. Principal Component Analysis for Yield Curve Modelling. *Enterprise Risk Solutions*.

Authors



Dipan Biswas, Master of Technology (Mechanical Engineering) from Indian Institute of Technology, Kharagpur.