Principal Component Analysis in Excel ~ PART III

Posted on August 5, 2015January 7, 2016 by bquanttrading
In this post we would like to expand on previous PCA post and show you how to build a very useful tool for scenario analysis of a yield curve. The method presented is an implementation of the main results of a paper by Leonardo M. Nogueira "Updating the Yield Curve to Analyst's Views". The SSRN link is at the end of the post.

If we find ourselves in a situation where we have a view on a particular point on a yield curve, for example the 2 year swap rate in Australia, we will probably need to know what the rest of the curve would look like if our view is correct. So what we need is a method to update the rest of the curve points based on where we think the two year yield is. There are many ways that we can draw a line through a single point so in order to have a sensible estimate for the full curve we need to impose some constraints. In the method that is presented below the constraint that we impose is that the curve is consistent with the main principal components.

The main formula that we will be working with is found on page 5 of the paper that we mentioned.

$$E[y_{t+1}] = y_t + \mu + DA(\Delta q - V\mu)$$ (https://asmquantmacro.com/wp-content/uploads/2015/08/eq1.jpg)

In case of a yield curve mu is set to zero so we are left with:

$$E[y_{t+1}] = y_t + DA(\Delta q)$$ (https://asmquantmacro.com/wp-content/uploads/2015/08/eq2.jpg)

where

$$A = \tilde{W}(VD\tilde{W})^{-1}$$ (https://asmquantmacro.com/wp-content/uploads/2015/08/eq3.jpg)

To help with the interpretation of the formula let's put some flesh on our example. Lets work with the Australian yield curve. We have a time series matrix for the following tenors (2y, 3y, 4y, 5y, 7y, 10y, 12y, 15y, 20y, 25y, 30y). We will be working with daily observations for each tenor dating back 3 years. Further, let's assume we have a view on where 2y and 5y yields are going to be. We want to know what the rest of the curve will look like if we are correct in our assumptions for the two swap rates. To do so we will need to use the first two principal components. If we had a view only on one swap tenor then we can only use the first principal component to derive the full yield curve. The key point is that if we have m tenors (in our case m =11) but only n views (in our case n= 2, ie 2y and 5y swap rates) we can use the first n principal components to derive where all m swap rates are likely to be.

Now we can use our example to help with the interpretation of the arguments of the above formula.

$E[y_{t+1}]$ is our derived yield curve that includes our expectations for a particular tenor. This is our expected swap rates for all 11 swap points.

(https://asmquantmacro.com/wp-content/uploads/2015/08/eq4.jpg)

y_t is the current yield curve   (https://asmquantmacro.com/wp-content/uploads/2015/08/eq5.jpg)

D is a matrix of standard deviations of each yield curve tenor   (https://asmquantmacro.com/wp-content/uploads/2015/08/eq6.jpg)

Δq is a matrix of our expected change for particular tenors. In our case this is a nx1 matrix of the expected change in the 2y and 5y tenor.

(https://asmquantmacro.com/wp-content/uploads/2015/08/eq7.jpg)

$\tilde{W}$ is the eigenvalue matrix of our PCA. As mentioned before we will use the first n principal components. So we are taking a subsection of the mxm matrix of the PCA and use only mxn matrix of the eigenvalues. If you need a refresher on PCA please see our first post on PCA.

(https://asmquantmacro.com/wp-content/uploads/2015/08/eq8.jpg)

V is a nxm matrix that holds 1 in place for any tenor that we have a view on and 0 otherwise. Again in our example we have eleven tenors with views on only the 2y and 5y swap rate. Therefore in the first row of V we will have the first column set to 1 (because 2y swap rate is the first tenor in our yield curve) and 0 in the rest of the columns. In the second row V will have 1 in the 4th row because that is the placement of our 5y tenor in the yield curve and 0 otherwise.

We can then use excel's native matrix multiplication and inverse function to derive the expected change in each yield curve tenor. By adding the expected changes to the current yield curve we are left with the expected yield curve.

Excel Implementation:

We first need to calculate the variance/covariance matrix of our swap data. We discussed this in our first PCA post with some user defined functions. We do this in range C3:M13

	D4				f_x {=CovarM(SwapData!D6:N944)}							
A	B	C	D	E	F	G	H	I	J	K	L	M
2	Var/Covar	2y	3y	4y	5y	7y	10y	12y	15y	20y	25y	30y
3	2y	0.24	0.23	0.23	0.23	0.22	0.21	0.20	0.20	0.18	0.16	0.15
4	3y	0.23	0.24	0.24	0.24	0.24	0.24	0.23	0.23	0.21	0.19	0.18
5	4y	0.23	0.24	0.25	0.25	0.26	0.25	0.25	0.25	0.23	0.21	0.20
6	5y	0.23	0.24	0.25	0.26	0.27	0.27	0.27	0.26	0.25	0.23	0.22
7	7y	0.22	0.24	0.26	0.27	0.28	0.29	0.29	0.29	0.27	0.25	0.24
8	10y	0.21	0.24	0.25	0.27	0.29	0.30	0.30	0.30	0.29	0.27	0.26
9	12y	0.20	0.23	0.25	0.27	0.29	0.30	0.31	0.31	0.29	0.28	0.27
10	15y	0.20	0.23	0.25	0.26	0.29	0.30	0.31	0.31	0.30	0.28	0.28
11	20y	0.18	0.21	0.23	0.25	0.27	0.29	0.29	0.30	0.29	0.28	0.28
12	25y	0.16	0.19	0.21	0.23	0.25	0.27	0.28	0.28	0.28	0.27	0.27
13	30y	0.15	0.18	0.20	0.22	0.24	0.26	0.27	0.28	0.28	0.27	0.27

(https://asmquantmacro.com/wp-content/uploads/2015/08/1.jpg)

Now we can calculate matrix D by taking the square root of the main diagonal. This is done in range C16:M26

| C16 | | fx | =IF(C$15=$B16,SQRT(C3),0) | | | | | | | | | |

	A	B	C	D	E	F	G	H	I	J	K	L	M
2		Var/Covar	2y	3y	4y	5y	7y	10y	12y	15y	20y	25y	30y
3		2y	0.24	0.23	0.23	0.23	0.22	0.21	0.20	0.20	0.18	0.16	0.15
4		3y	0.23	0.24	0.24	0.24	0.24	0.24	0.23	0.23	0.21	0.19	0.18
5		4y	0.23	0.24	0.25	0.25	0.26	0.25	0.25	0.25	0.23	0.21	0.20
6		5y	0.23	0.24	0.25	0.26	0.27	0.27	0.27	0.26	0.25	0.23	0.22
7		7y	0.22	0.24	0.26	0.27	0.28	0.29	0.29	0.29	0.27	0.25	0.24
8		10y	0.21	0.24	0.25	0.27	0.29	0.30	0.30	0.30	0.29	0.27	0.26
9		12y	0.20	0.23	0.25	0.27	0.29	0.30	0.31	0.31	0.29	0.28	0.27
10		15y	0.20	0.23	0.25	0.26	0.29	0.30	0.31	0.31	0.30	0.28	0.28
11		20y	0.18	0.21	0.23	0.25	0.27	0.29	0.29	0.30	0.29	0.28	0.28
12		25y	0.16	0.19	0.21	0.23	0.25	0.27	0.28	0.28	0.28	0.27	0.27
13		30y	0.15	0.18	0.20	0.22	0.24	0.26	0.27	0.28	0.28	0.27	0.27
14													
15		StDev	2y	3y	4y	5y	7y	10y	12y	15y	20y	25y	30y
16		2y	48%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
17		3y	0%	49%	0%	0%	0%	0%	0%	0%	0%	0%	0%
18		4y	0%	0%	50%	0%	0%	0%	0%	0%	0%	0%	0%
19		5y	0%	0%	0%	51%	0%	0%	0%	0%	0%	0%	0%
20		7y	0%	0%	0%	0%	53%	0%	0%	0%	0%	0%	0%
21		10y	0%	0%	0%	0%	0%	54%	0%	0%	0%	0%	0%
22		12y	0%	0%	0%	0%	0%	0%	55%	0%	0%	0%	0%
23		15y	0%	0%	0%	0%	0%	0%	0%	56%	0%	0%	0%
24		20y	0%	0%	0%	0%	0%	0%	0%	0%	54%	0%	0%
25		25y	0%	0%	0%	0%	0%	0%	0%	0%	0%	52%	0%
26		30y	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	52%
27													

(https://asmquantmacro.com/wp-content/uploads/2015/08/2.jpg)

Next we can carry out PCA using matrix.xla add-in that was created by Leonardo Volpi and the Foxes Team (this add-in is no longer supported on the foxes team's webpage but is available at http://excellaneous.com/ (http://excellaneous.com/). We discussed implementation of PCA using this add-in in our first post on PCA and you can refer to it to see the functions and their uses. The eigenvector is in range P5:Z15

| P5 | | fx | {=MatEigenvector(C3:M13,TRANSPOSE(P4:Z4))} | | | | | | | | |

	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
2		CumVar	91.8%	99.4%	99.9%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%
3		Var	91.8%	7.6%	0.5%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
4		EigVal	2.760	0.228	0.015	0.001	0.001	0.000	0.000	0.000	0.000	0.000	0.000
5		2y	0.24	0.55	0.40	0.42	0.44	0.21	0.01	0.24	0.08	0.04	0.00
6		3y	0.27	0.39	0.19	-0.02	-0.21	-0.56	0.09	-0.49	-0.32	0.11	0.17
7		4y	0.29	0.32	-0.00	-0.32	-0.29	0.01	-0.11	0.07	0.22	-0.50	-0.57
8		5y	0.30	0.22	-0.13	-0.38	-0.24	0.20	-0.16	0.18	0.36	0.23	0.60
9		7y	0.32	0.07	-0.29	-0.27	0.12	0.32	0.10	0.08	-0.55	0.45	-0.31
10		10y	0.33	-0.07	-0.36	0.02	0.33	0.16	0.33	-0.23	-0.11	-0.59	0.33
11		12y	0.33	-0.13	-0.35	0.19	0.22	-0.31	0.14	-0.17	0.57	0.35	-0.27
12		15y	0.33	-0.17	-0.23	0.33	-0.02	-0.28	-0.65	0.32	-0.26	-0.12	0.07
13		20y	0.32	-0.24	0.14	0.41	-0.59	0.13	0.47	0.24	-0.04	0.02	0.01
14		25y	0.30	-0.34	0.37	0.05	-0.03	0.43	-0.37	-0.57	0.08	0.05	-0.06
15		30y	0.29	-0.41	0.49	-0.44	0.31	-0.32	0.15	0.31	-0.02	-0.03	0.03

(https://asmquantmacro.com/wp-content/uploads/2015/08/3.jpg)

Now we can set up our current yield curve and a range that will hold our forecast for 2y and 5y tenors. In column P we input the current swap rates. In column Q we input our expected value for the 2y tenor. Let's assume we assume 2y swap rate is going to 1.95%. In column R we input our estimate for the 5y swap rate. Again let's assume we think the 5y rate is

going to 2.30%. It is important to note that with this set up we must have one input in each column.

 (https://asmquantmacro.com/wp-content/uploads/2015/08/4.jpg)

To review, so far we have our column matrix y, we have our matrix D and we have our eigenvalues which is W hat. We still need matrix V and delta q.

In range C29:M30 we set up matrix V. In range M33:M34 we set up our delta q matrix. Delta q is simply a matrix that holds our expected changes of each tenor (for example, our expected 2y yield of 1.95 less 2.165 current 2y yield yields -.215)

 (https://asmquantmacro.com/wp-content/uploads/2015/08/5.jpg)

So now we have all the components needed to derive the full yield curve. In range S18:S28 input the formula
=MMULT(MMULT(C16:M26,MMULT(P5:Q15,MINVERSE(MMULT(MMULT($C$29:$M$30,C16:M26),P5:Q15)))),M33:M34)

This will calculate

$$DA(\Delta q)$$ (https://asmquantmacro.com/wp-content/uploads/2015/08/eq9.jpg)

We can then add the results to our current yield curve and we are finished

(https://asmquantmacro.com/wp-content/uploads/2015/08/6.jpg)

We can see that the derived curve matches exactly our expected values for the 2y and 5y swap rate and also gives up the rest of the tenors that are consistent with the first two principal components which account for over 99% of the variance in the data.

(https://asmquantmacro.com/wp-content/uploads/2015/08/7.jpg)

VBA Implementation:

Above implementation is a touch tedious. We took the long road to help with the exposition of the main results. This same method can be implemented neatly in VBA without all the intermediate calculations in excel. Below is the main function:

```
(General)                                                    ▼  PCAScenarioAnalysis

    Option Base 1
    Option Explicit

    Public Function PCAScenarioAnalysis(LatestRates As Range, AnalystViews As Range, SwapsTimeSeries As Range) As Variant
        Dim N As Integer:   N = SwapsTimeSeries.Rows.Count
        Dim M As Integer:   M = SwapsTimeSeries.Columns.Count
        Dim ii As Integer:  ii = 1
        Dim jj As Integer:  jj = 1
        Dim cc As Integer:  cc = 1

        'Calculate VarCovar and StDev Matrix
        ReDim VarCovarM(M, M) As Double
        ReDim StDevM(M, M) As Double
        For ii = 1 To M
            For jj = 1 To M
                VarCovarM(ii, jj) = _
                    Application.WorksheetFunction.Covar(SwapsTimeSeries.Columns(ii), SwapsTimeSeries.Columns(jj))
                If ii = jj Then
                    StDevM(ii, jj) = Sqr(VarCovarM(ii, jj))
                Else
                    StDevM(ii, jj) = 0
                End If
            Next jj
        Next ii
        'Calculate eigenvectors using matrix add-in
        ReDim EigenValM(M, M) As Double
        EigenValM = Application.Run("matrix.xla!MatEigenvector_pow", VarCovarM, True)

        'Create V, W_hat and deltaQ matrix
        Dim NumViews As Integer:        NumViews = Application.WorksheetFunction.CountA(AnalystViews)
        ReDim V(NumViews, M) As Double
        ReDim DeltaQ(NumViews, 1) As Double
        ReDim W_hat(M, NumViews) As Double

        ii = 1: cc = 1: jj = 1
        Dim TempCount As Integer
        For ii = 1 To M
            If AnalystViews(ii).Value <> 0 Then
                DeltaQ(jj, 1) = AnalystViews(ii).Value - LatestRates(ii).Value
                For cc = 1 To M
                    W_hat(cc, jj) = EigenValM(cc, jj)
                    If cc = ii Then
                        V(jj, cc) = 1
                    Else
                        V(jj, cc) = 0
                    End If
                Next cc
                jj = jj + 1
            End If
        Next ii
```
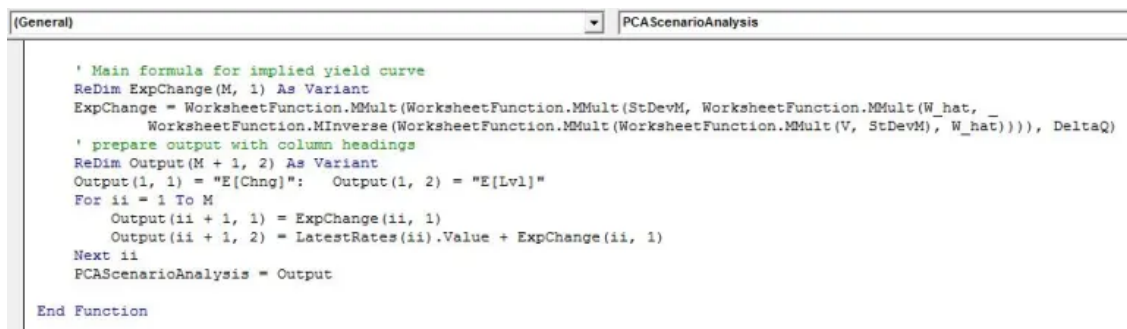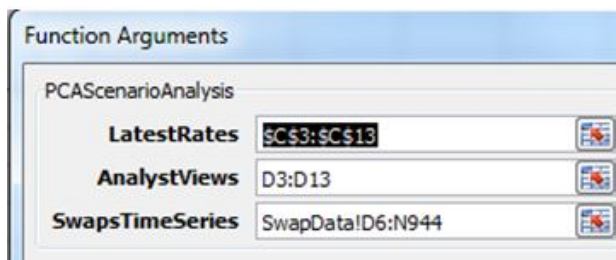
(https://asmquantmacro.com/wp-content/uploads/2015/08/8.jpg)

```
(General)                                                    ▼  PCAScenarioAnalysis

        ' Main formula for implied yield curve
        ReDim ExpChange(M, 1) As Variant
        ExpChange = WorksheetFunction.MMult(WorksheetFunction.MMult(StDevM, WorksheetFunction.MMult(W_hat, _
                WorksheetFunction.MInverse(WorksheetFunction.MMult(WorksheetFunction.MMult(V, StDevM), W_hat)))), DeltaQ)
        ' prepare output with column headings
        ReDim Output(M + 1, 2) As Variant
        Output(1, 1) = "E[Chng]":   Output(1, 2) = "E[Lvl]"
        For ii = 1 To M
            Output(ii + 1, 1) = ExpChange(ii, 1)
            Output(ii + 1, 2) = LatestRates(ii).Value + ExpChange(ii, 1)
        Next ii
        PCAScenarioAnalysis = Output

    End Function
```

(https://asmquantmacro.com/wp-content/uploads/2015/08/9.jpg)
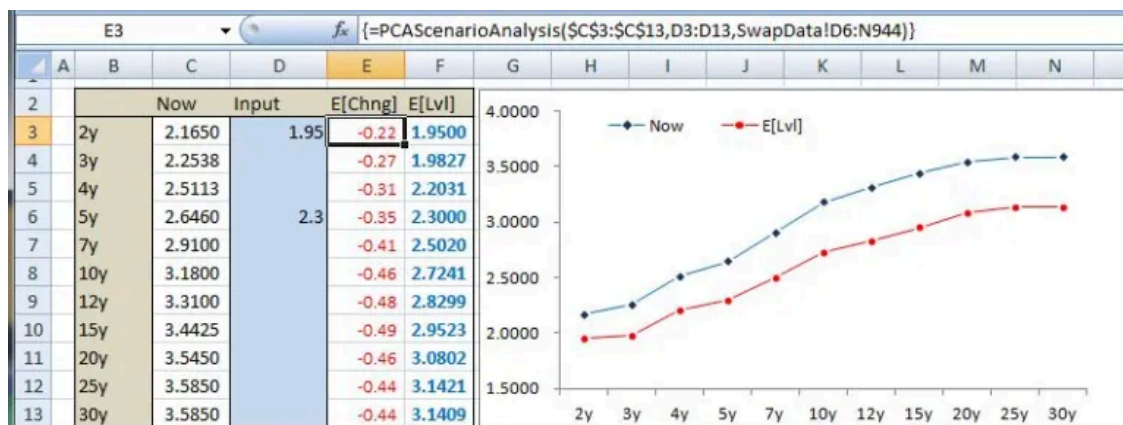
To use the function we need to enter it as an array function by selecting a range 2 columns wide and M+1 rows and then hitting Shift+Ctrl+Enter.

In a spreadsheet we input the function is range E2:F13

Function Arguments

PCAScenarioAnalysis

LatestRates	$C$3:$C$13
AnalystViews	D3:D13
SwapsTimeSeries	SwapData!D6:N944

(https://asmquantmacro.com/wp-content/uploads/2015/08/11.jpg)

(https://asmquantmacro.com/wp-content/uploads/2015/08/10.jpg)

The results agree with our excel implementation earlier.

Final Note:

This method can be used with all yield curves, volatility surfaces and futures curves. A word of caution for macro traders, this approach will not work across asset classes. For instance if you are looking to see what the expected change in USDJPY is given your expectations for SPX and US rates by running this function, you will be disappointed with the results. Main reason this approach does not work is because the variance/covariance matrix is unstable and hence your PCA results will be dubious. We have carried out extensive backtests on this approach across asset classes and without exception results are poor. This approach to scenario analysis works incredibly well on highly correlated time series such as yield curves but may breakdown when there is a regime change. For example, modelling US yield curve with data that only includes history of when Fed was on hold should be made with caution given the likely changes in Fed policy in the near future.

Some useful resources:

1. This is a link to the main paper ~ http://papers.ssrn.com/sol3/papers.cfm?abstract_id=1107653 (http://papers.ssrn.com/sol3/papers.cfm?abstract_id=1107653)
2. https://quantmacro.wordpress.com/2015/06/28/principal-component-analysis-in-excel-part-i/ (https://quantmacro.wordpress.com/2015/06/28/principal-component-analysis-in-excel-part-i/)
3. https://quantmacro.wordpress.com/2015/06/29/principal-component-analysis-part-ii/ (https://quantmacro.wordpress.com/2015/06/29/principal-component-analysis-part-ii/)
4. Market Risk Analysis Volume 1 by Carol Alexander (an excellent source about PCA): http://www.amazon.com/Market-Analysis-Quantitative-Methods-Finance/dp/0470998008/ref=sr_1_2?s=books&ie=UTF8&qid=1435483909&sr=1-2&keywords=market+risk+analysis (http://www.amazon.com/Market-Analysis-Quantitative-Methods-Finance/dp/0470998008/ref=sr_1_2?s=books&ie=UTF8&qid=1435483909&sr=1-2&keywords=market+risk+analysis)

  ↳   Posted in Excel, VBA

---

5 thoughts on "Principal Component Analysis in Excel ~ PART III"

1. *Peter Urbani*   says: May 11, 2016 at 4:20 am
   Hi Bquant

   Great post. My comments seem to be going astray. I am unable to replicate the model I think due to my having a different version of Matrix.xla. Would you be able to send me a copy of the spreadsheet ?

   Many thanks,

   Peter

Reply

> **bquanttrading**   says: May 28, 2016 at 11:35 am
> hey PU, thanks for reading. unfortunately I dont have the sheet anymore. I carried on with the project and it doesnt look anything like what I wrote about in the post. Perhaps you can use prcomp in R to run PCA to compare.
>
> gluck
>
> Reply
>
> 1. **peterurbani1**   says: May 28, 2016 at 11:44 am
>   Thanks anyway. I did manage to get it to work. Kind regards,
>
>   PU

2. *Regis*   says: September 14, 2019 at 1:59 pm
   Hello BPquant

   Would it be possible to share an R implementation?

   Many thanks in advance

   Reply

3. *Andreea Oprea*   says: June 28, 2021 at 10:55 pm
   Hello, I have performed this analysis for the Romanian yield curve and I have some dubious results when incorporating 2 views and hence 2 PCs. If I only select 1 view, then the scenario I get for the entire yield curve is plausible. Do you still maintain this blog, so that I can provide you with a screen shot, and maybe you can provide some advice? thank you very much in advance,

   Reply