APM466 Assignment 1

Nick Zhu hong Jia
APM466 - Mathematical Finance

February 14, 2024

Contents

1 Question 1:

1.1 Part a)

The government issues bonds to borrow money from the public or other financial institutions; excessively printing money can cause inflation and bonds are a good form of credibility, debt management alongside being subject to market discipline.

1.2 Part b)

There could be many reasons but this could be explained by the liquidity premium theorem; the flattening could be due to expectations that long-term interest rates are similar to short-term rates; this suggests that investors expect short-term rates to remain relatively stable which could be driven by factors such as expectations of economic slowdown or low inflation expectations.

1.3 Part c)

Quantitative easing (QE) is a monetary policy used by central banks though purchasing long-term securities from the market to increase the central bank's balance sheet and injects money into the financial system, aiming to lower long-term interest rates and stimulate the economy when standard monetary policy measures aren't effective. The US Fed employed QE through purchasing mortgage-backed securities, corporate bonds and providing liquidity to state and local governments facing funding pressures due to the pandemic.

2 Question 2:

The bonds that I selected are the following: [CAN 2.5 Jun 24, CAN 2.25 Jun 25, CAN 1.5 Jun 26, CAN 1.25 Mar 25, CAN 0.5 Sept 25, CAN 8 Jun 27, CAN 2.25 Jun 29, CAN 2 Jun 32, CAN 1.5 Dec 31, CAN 3.25 Sept 28]

I selected a mixture of both bonds that mature between 0-3 years along side bonds that mature between 3 - 10 years because I wanted to see if there will be a significant difference in yield to maturity between these categories. Through a wise spread of coupon rates and maturity dates, I tried to get a grasp of the general trend of ytm.

3 Question 3:

Principal Component Analysis (PCA) allows us to understand higher dimension data through projection into a lower dimension. the eigenvalues of the covariance matrix tell us the variance of the data along the corresponding eigenvector (principal component)(PC).Larger eigenvalues represent the respective PC explains more variance in the data. Eigenvalues are also used to determine the importance of each PC in capturing the variability of the original data. PC with highest variance are the main focus, but PC's with low variances allows identification of near-linear dependencies between variables.

Eigenvectors represent the directions (or axes) in the original feature space along which the data vary the most. Each eigenvector is associated with an eigenvalue, and together, they form a pair that represents a principal component.The PC loadings are the eigenvectors in the sample covariance matrix and are orthogonal to each other and capture different directions of variability in the data.

4 Question 4

4.1 Part a)

The plots for part a) b) and c) for space considerations. The full plots will be available in the RMD file in github I started with calculating the dirty price of the bonds and then calculated the bond yield for each of the closing prices of the bonds. I used the jrvfinance package to calcuate bond yield for each of the bonds that I selected.

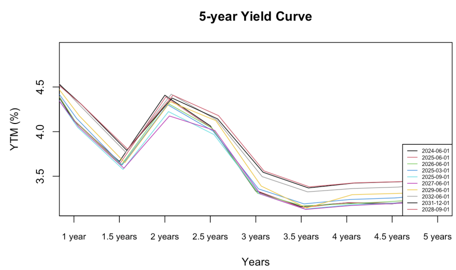

Figure 1: 5 year to maturity plot

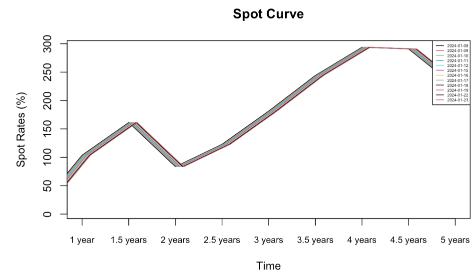

Figure 2: Spot rate plot

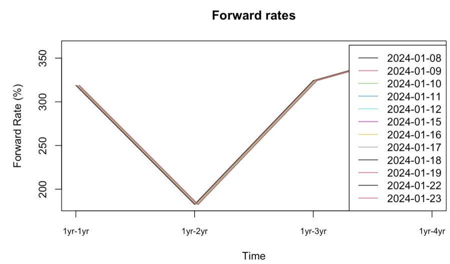

Figure 3: Forward Rate plot

4.2 Part b)

In pesudo code terms, I outlined what the code is doing below:

```
For each bond in bonds_data:
    For each settlement date in settlementdates (12 days):
        Calculate the time until payment as the absolute difference between maturity date and
             settlement date, divided by 365:
        Store the result in the datesmat matrix at the corresponding position
    Append the datesmat matrix to the list of matrices
Convert the list of matrices into a data frame called ratiomatdat(the ratio of differences)
Rename the columns of ratiomatdat with the settlement dates

For each bond in bonds_data:
    For each settlement date in settlementdates (12 days):
        If it's the first bond:
            Calculate spot rate using the formula:
                -log10(dirtyprice / (100 + Coupon * 100)) / ratiomatdat
        Else:
            Using the data from the previous row in the spot rate matrix to calculate part1:
                dirtyprice - (Coupon * sum(exp(-previous_spot_rates * ratiomatdat)))
            Calculate part2:
                100 + (Coupon * 100)
            Calculate spot rate using the formula:
                -log(part1 / part2) / ratiomatdat
```

4.3 Part c)

```
Define a vector of forward times, fwdtimes, containing the values 1 to 5.
```

```
Create an empty matrix called fwdrates with 4 rows and 12 columns.

For each row i in the range 1 to 4:
    Calculate the forward rates for each column j in the range 1 to 12 using the formula:
        - Retrieve the spot rate at index (2*i + 1) from spotratesdat and store it as spotrate1
        - Retrieve the spot rate at time 1 from spotratesdat and store it as spotrate0
        - Calculate forwardrates at the (i,j) index as ((i + 1) * spotrate1 - spotrate0) /
             (fwdtimes[i + 1] - fwdtimes[1])

Create a data frame called fwdratedat from the fwdrates matrix.
```

5 Question 5

	vec1	vec2	vec3	vec4	vec5
vec1	0.0002607810	0.0002449683	0.0002393193	0.0002517985	0.0002766392
vec2	0.0002449683	0.0003317221	0.0002293315	0.0002480340	0.0002661596
vec3	0.0002393193	0.0002293315	0.0002438066	0.0002440069	0.0002618195
vec4	0.0002517985	0.0002480340	0.0002440069	0.0002791941	0.0002783935
vec5	0.0002766392	0.0002661596	0.0002618195	0.0002783935	0.0003035146

Figure 4: Covariance matrix of time series of daily log returns to yield of bonds

	vec1	vec2	vec3	vec4
vec1	6.840458e-09	5.970745e-09	2.241082e-09	1.503766e-09
vec2	5.970745e-09	5.211610e-09	1.956145e-09	1.312573e-09
vec3	2.241082e-09	1.956145e-09	7.342267e-10	4.926661e-10
vec4	1.503766e-09	1.312573e-09	4.926661e-10	3.305790e-10

Figure 5: Covariance matrix of time series of daily log forward rate

6 Question 6

```
eigen() decomposition
$values
[1] 1.302362e-03 7.965044e-05 1.985700e-05 1.346472e-05 3.683828e-06

$vectors
          [,1]       [,2]        [,3]       [,4]         [,5]
[1,] 0.4376413 -0.1901598  0.46588242  0.4638096  0.583218298
[2,] 0.4542510  0.8889327 -0.01449453 -0.0566823  0.005628671
[3,] 0.4183182 -0.2645056  0.20529969 -0.8381045  0.102369455
[4,] 0.4472477 -0.2360923 -0.83868672  0.1127810  0.167675010
[5,] 0.4765558 -0.2189409  0.19287328  0.2579322 -0.788181655

eigen() decomposition
$values
[1]  2.473996e-09  3.286390e-18  1.070528e-26 -2.544924e-25

$vectors
          [,1]        [,2]        [,3]       [,4]
[1,] 0.7221500  0.41354554  0.00000000  0.5545083
[2,] 0.6303341 -0.06469528  0.03878259 -0.7726509
[3,] 0.2365919 -0.70947649 -0.62596059  0.2209994
[4,] 0.1587532 -0.56695478  0.77888976  0.2160797
```

Figure 6: Eigen value and Eigen vectors of both covariance matrices

The first eigen value and eigen vector pair represents the first principle component which accounts for the over all performance of the bonds along side the variability.