MAT1856/APM466 Assignment 1

Ziqi Zhang, Student #: 1003537336

February 2, 2020

Fundamental Questions - 25 points

1.

   (a) Because the government needs funds for expenditures including large projects that may cost a lot of money.

   (b) It can help predict how the economy of a country goes.

   (c) The money supply in the market decreases when people spend money to buy bonds from the government.

2. Here are the 11 bonds I collected: "CAN 1.5 Mar 1,2020", "CAN 0.75 Sep 1,2020", "CAN 0.75 Mar 1, 2021", "CAN 0.75 Sep 1,2021", "CAN 0.5 Mar 1,2022", "CAN 2.75 Jun 1,2022", "CAN 1.75 Mar 1,2023", "CAN 1.5 Jun 1,2023" "CAN 2.25 Mar 1,2024", "CAN 1.5 Sep 1,2024", "CAN 1.25 Mar 1,2025". I chose these bonds based on the time interval of six months because the coupons are issued semi-annually and it would be easy for bootstrapping. However, there are no bonds that mature on Sep 1, 2022 and Sep 1, 2023 so the two bonds which mature on Jun 1 were selected to get a better estimation of the curves.

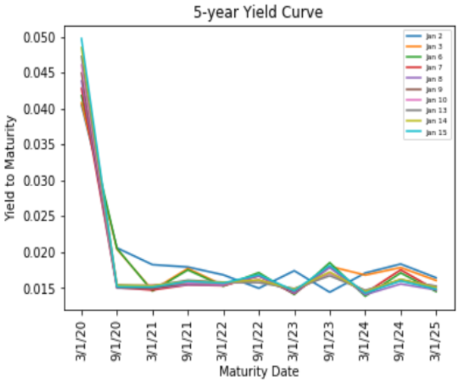
3. The largest eigenvalue of the covariance matrix represents the largest variance, second largest eigenvalue the second largest variance etc. When studying the dynamics of the market, it is usually the case that the largest eigenvalue usually associates with an eigenvector which has the same sign through different maturities, indicating parallel shifts of the bonds; the second largest eigenvalue usually associates with an eigenvector that has first-half of the component positive and the second-half of the component negative, indicating slope-tilts; the third largest eigenvalue usually associates with an eigenvector that has short-term and long-term rates positive and medium-term rates negative, indicating the flexing and related to the convexity of the curve.

Empirical Questions - 75 points

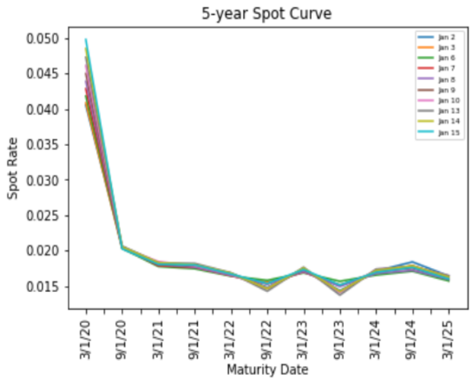
4.

   (a) The graph is provided below and the code that generated this graph can be found in the GitHub link at the end of this report. I first calculated the dirty price and then use the function cal-ytm to get all the ytms and stored in a list of lists called ytm-day-list. Then I applied the interpolation technique to estimate the ytm on Sep 1, 2022 and Sep 1, 2023 using the two bonds that mature on Jun 1, 2022 and Jun 1, 2023. By linear estimation, I assumed that the ytm on Jun 1 is approximately the average of Mar 1 and Sep 1. So I can get the estimation of the ytm on Sep 1 from the ytm on Mar 1 and Jun 1. **Notice normally the ytm curve should go up. This curve has an unusual big drop in the beginning may due to the reason**

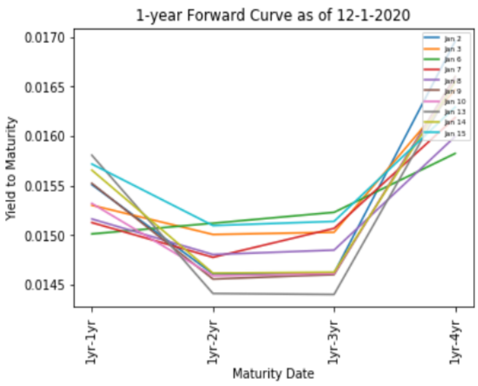
that the error was amplified when the maturity date is close.



(b) I used the bootstrapping method. Let p be the dirty price, c be the coupon, N be the number of coupon payments remaining, t_n be the days as a fraction of a year until nth coupon, r_{tn} be the spot rate. The formula is $p = (100 + c)e^{-r_{t_N} t_N} + \sum_{n=1}^{N-1} ce^{-r_{t_n} t_n} for N \geq 2$ When N = 1, it is a zero-coupon bond and the spot rate is equal to the ytm. I then solve for the spot rate r_{tn} when N = 2. When As the only unknown to solve is r_{t2}, it is easy to solve. Similarly, I can get all the spot rate for N > 3 respectively. Practically, I calculated the spot rate every quarter of year and used similar interpolation method as in part (a).



(c) I used the formula $r(T_0; T_1, T_N) = (r_N T_N - r_1 T_1)/(T_N - T_1) for N = 2, 3, 4 or 5$ to get the forward rate. T_0 is the present date, T_1 is 1 year to present date (practically it is Dec 1, 2020 as it closer to a year than Mar 1, 2021 and I have recorded that estimation of spot rate in part(b)), T_N is N years to present, and r_N is the spot rate at N years from present. Solve for $r(T_0; T_1, T_N)$ when $N = 2, 3, 4, 5$ to get all forward rates.



5. In Python, import numpy and use it to build the matrix time series of daily log-returns and then find the covariance matrix.

   Covariance matrices for the time series of daily log-returns of yield rate

```
[[3.13397854e-05 4.05120057e-04 1.70249686e-04 3.71408091e-04 1.38159644e-05]

 [4.05120057e-04 5.54791444e-03 2.27090340e-03 5.00806612e-03 2.75982431e-04]

 [1.70249686e-04 2.27090340e-03 1.00870836e-03 2.14692323e-03 2.79601861e-04]

 [3.71408091e-04 5.00806612e-03 2.14692323e-03 4.65423888e-03 4.30186274e-04]

 [1.38159644e-05 2.75982431e-04 2.79601861e-04 4.30186274e-04 4.54119337e-03]]
```

   Covariance matrices for the time series of daily log-returns of forward rate

```
[[ 0.00029953 -0.00019318 -0.0002086   0.00027027]
 [-0.00019318  0.00035977  0.00034168 -0.0003264 ]
 [-0.0002086   0.00034168  0.00038288 -0.00030116]
 [ 0.00027027 -0.0003264  -0.00030116  0.00056911]]
```

6. I have already got the covariance matrix in question 5, now I just need to apply linalg from numpy to calculate the eigenvalues and eigenvectors.

   Eigenvalues and eigenvectors of the yield covariance matrix

```
[1.11731515e-02 4.50198303e-03 1.01576690e-04 1.44180316e-06 5.24178722e-06]

[[-0.05149537 0.00761853 0.00462438 -0.99734348 -0.05074226]

 [-0.69963818 0.08447081 -0.67562848 0.04442029 -0.21195305]

 [-0.29531992 -0.00149381 0.55311718 0.05732752 -0.7768905]

 [-0.64318455 0.0372899 0.48577011 0.00569354 0.59069289]

 [-0.08339324 -0.99569768 -0.03991933 -0.00373547 0.00491814]]
```

   Eigenvalues and eigenvectors of the forward covariance matrix

```
[1.25167426e-03 1.99838094e-04 1.34506139e-04 2.52747242e-05]
[[ 0.38045181 -0.26870657 -0.87764845  0.11307694]
 [-0.49362231 -0.42972595 -0.17711738 -0.73505241]
 [-0.49527994 -0.56922101  0.04394163  0.65479337]
 [ 0.60521997 -0.64739464  0.44320623 -0.13474866]]
```

   Explanation: similarly in question 3, the eigenvector that is associated with the largest eigenvalue tells how much variance can be explained by its associated eigenvector, so it represents the largest variation after orthogonal decomposition in the direction of the eigenvector.

References and GitHub Link to Code

https://quant.stackexchange.com/questions/36844/principal-component-analysis-of-yield-curve-change

https://stats.stackexchange.com/questions/31908/ what-is-percentage-of-variance-in-pca

GitHub Link to Code:
`https://github.com/zziqizhang/APM466_A1`