# MAT1856/APM466 Assignment 1

Bingrui Wu, Student #: 1005726699

February, 2020

## Fundamental Questions - 25 points

1.

   (a) Governments often issue bonds rather than simply print more money when there is a need to raise money because bonds serve as a way of adjusting the nation's money supply in both directions, either to increase or to reduce, while simply printing money is an uni-dimensional action and causes inflation.

   (b) Suppose there is a bond that matures 100 years from now, then people are not so willing to buy it since they are uncertain about will happen in future, especially in such long time, hence the long-term parts of the yield curve might be flat.

   (c) Quantitative easing is the action that a central bank of a nation purchases securities from open market, the US Fed employed this since the beginning of the COVID-19 pandemic because the economy was experiencing a recession and the Fed wanted to increase the money supply to stimulate the economy.

2. The 10 selected bonds are "CAN 1.75 Mar 1","CAN 1.50 Jun 1","CAN 2.25 Mar 1","CAN 1.50 Sep 1","CAN 1.25 Mar 1"(2025),"CAN 0.50 Sep 1","CAN 0.25 Mar 1","CAN 1.00 Sep 1","CAN 1.25 Mar 1"(2027),"CAN 2.75 Sep 1". Since the coupon payment is on a semi-annual basis, we want the selected data to be of every six months. The above 10 bonds have maturity every six years, expect "CAN 1.50 Jun 1", and is the best that we can have. Notice that we have two bonds that mature at Mar. 1, 2023, and we chose the one with nearer issue date.

3. The principal component analysis is used to interpret the variance-covariance structure of the given data. Statistically, PCA performs a job of dimension reduction, i.e., find those directions that explains most of the variation. By the Rayleigh-Ritz theorem, the eigenvector that corresponds to the largest eigenvalue is the direction of variation that can explain the most variances, the eigenvector that corresponds to the second largest eigenvalue is the direction of variation that can explain the second most variances, and so on [1].

## Empirical Questions - 75 points

4.

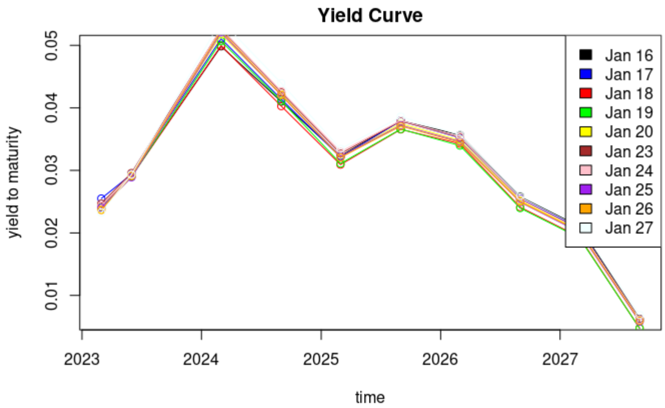
   (a) The first two bonds have maturities less than six months, hence they are considered zero coupon bonds, their yield to maturity curve is calculated using the formula

$$r(T) = -\frac{\log(P/N)}{T}$$

where $P$ is the dirty price, $N$ is the notional, and $T$ is the time to maturity. For the rest bonds with maturities more than six months, we apply the formula

$$P = \sum_{i=1}^{n} p_i e^{-r(t_i)t_i}$$

where $n$ is the total number of payments left, $p_i$ is the money one receives at $i$-th payment. Solve the equation we obtain the yield when the bond matures. With all the data calculated, the yield curve is as follows:




We shall see that the plot indicate that the yield to maturity curve is generally on a decline trend, though also fluctuated. It is also worth mentioning that the yield curve based on different date of data are similar to each other, which indicates the variation of yield curve is small if the data used to calculate it is on a day to day basis and is within a short amount of time.

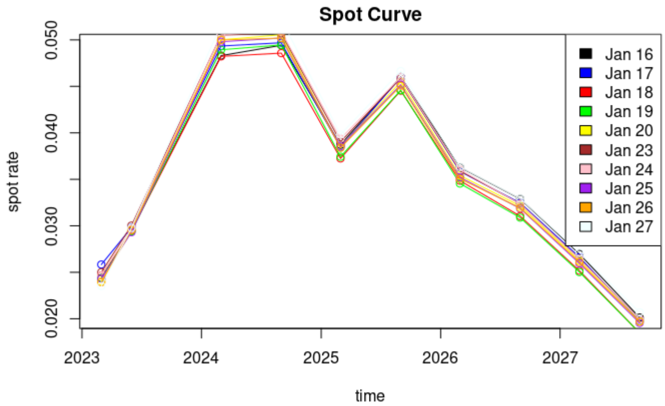
(b) For bonds that matures within six months, one have

$$P = \frac{N}{(1+s)^T}$$

where $s$ is the spot rate and $T$ is the time to maturity measured in year.
For other bonds, one have

$$P = \sum_{i=1}^{n} \frac{p_i}{(1+s_i)^{t_i}}$$

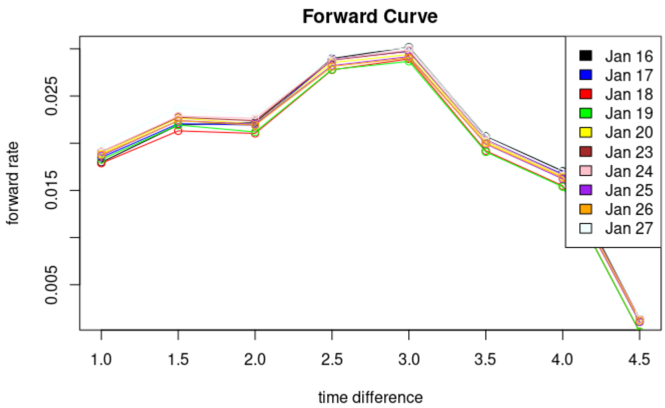
where $n$ is the number of payments left, $s_i$ is the spot rate at time point $i$, and $t_i$ is the time from now to the time point $i$. The idea of the pseudo-code is that we first calculate the spot rate at Mar.1.2023 using bond 1 which matures at Mar.1.2023, after we get the first spot rate, we can calculate the spot rate six months after using the formula mentioned above. Repeat the process and calculate recursively, and we will have all the spot rates needed. The pseudo-code and the spot curve are as follows. We shall see from the graph that the spot curve is different from the yield curve, but the two curves have similar trends in general.




(c) To calculate the forward yield, we use the formula

$$r(t, T_1, T_2) = -\frac{\log P(t, T_2) - \log P(t, T_1)}{T_2 - T_1}$$

where $P(t, T_1)$ is the price we need to pay at time $t$ in order to get a payment of the notional at time $T_1$, and $P(t, T_2)$ is the price we need to pay at time $t$ in order to get a payment of the notional at time $T_2$, here we use the dirty prices.
The forward curve is as follows.




5. We first construct $X_i$ as required using the previously derived yield in year $i$, and then combine them into a $5 \times 9$ matrix and calculates its covariance. We do the same thing to the forward rates. The R-codes used are as follows:

```
X_1 <-log(ytm_1[-c(1)]/ytm_1[-c(10)])
X_2 <-log(ytm_2[-c(1)]/ytm_1[-c(10)])
X_3 <-log(ytm_3[-c(1)]/ytm_1[-c(10)])
X_4 <-log(ytm_4[-c(1)]/ytm_1[-c(10)])
X_5 <-log(ytm_5[-c(1)]/ytm_1[-c(10)])
X <- cbind(X_1,X_2,X_3,X_4,X_5)
cov(X)

Y_1 <- -log(forward_rate_1[-c(1)]/forward_rate_1[-c(10)])
Y_2 <- -log(forward_rate_2[-c(1)]/forward_rate_2[-c(10)])
Y_3 <- -log(forward_rate_3[-c(1)]/forward_rate_3[-c(10)])
```

```
Y_4 <- -log(forward_rate_4[-c(1)]/forward_rate_4[-c(10)])
Y_5 <- -log(forward_rate_5[-c(1)]/forward_rate_5[-c(10)])
Y <- cbind(Y_1,Y_2,Y_3,Y_4,Y_5)
cov(Y)
```

The results obtained is

$$Cov(X) = \begin{pmatrix} 0.0009885950 & 0.0005502381 & 0.0004715777 & 0.000481121 & 0.0006126836 \\ 0.0005502381 & 0.0007664971 & 0.0010674149 & 0.001109914 & 0.0010990068 \\ 0.0004715777 & 0.0010674149 & 0.0017948659 & 0.001854526 & 0.0018132114 \\ 0.0004811210 & 0.0011099140 & 0.0018545265 & 0.001936769 & 0.0018856701 \\ 0.0006126836 & 0.0010990068 & 0.0018132114 & 0.001885670 & 0.0018999233 \end{pmatrix}$$

$$Cov(Y) = \begin{pmatrix} 0.0006862532 & 0.0007083762 & 0.0006554829 & 0.0005808759 & 0.0004366424 \\ 0.0007083762 & 0.0008272449 & 0.0007801711 & 0.0006627302 & 0.0005238777 \\ 0.0006554829 & 0.0007801711 & 0.0008614990 & 0.0006999022 & 0.0006128760 \\ 0.0005808759 & 0.0006627302 & 0.0006999022 & 0.0005910175 & 0.0004986625 \\ 0.0004366424 & 0.0005238777 & 0.0006128760 & 0.0004986625 & 0.0004607474 \end{pmatrix}$$

6. The eigenvalues and corresponding eigenvectors of X's covariance matrix are

| eigenvalues | 6.432185e-03 | 8.702797e-04 | 6.561247e-05 | 1.194670e-05 | 6.625797e-06 |
|---|---|---|---|---|---|
| eigenvectors | -0.1869800 | 0.9354417 | -0.15716103 | -0.1524863 | -0.2050261 |
| | -0.3274265 | 0.2032734 | 0.76804611 | 0.3116615 | 0.4055171 |
| | -0.5221985 | -0.1926887 | 0.08699746 | -0.8148590 | 0.1364405 |
| | -0.5424944 | -0.2097118 | 0.10835571 | 0.2823716 | -0.7551464 |
| | -0.5392948 | -0.0502076 | -0.60505873 | 0.3686274 | 0.4523915 |

The eigenvalues and corresponding eigenvectors of Y (forward rates)'s covariance matrix are

| eigenvalues | 3.196490e-03 | 1.786019e-04 | 3.879288e-05 | 8.732360e-06 | 4.144951e-06 |
|---|---|---|---|---|---|
| eigenvectors | -0.4331368 | 0.6402757 | -0.5783553 | -0.1413608 | 0.21900176 |
| | -0.4950166 | 0.3492780 | 0.7460575 | 0.2448504 | 0.12810096 |
| | -0.5083630 | -0.4104900 | 0.1036099 | -0.7497238 | -0.01562651 |
| | -0.4271092 | -0.1210747 | -0.2021186 | 0.3446843 | -0.80203459 |
| | -0.3556625 | -0.5337515 | -0.2394086 | 0.4890525 | 0.54048562 |

The eigenvector corresponds to the eigenvalue with the largest absolute value is the direction in which the data varies the most, i.e., is the principal direction that the data points being distributed; and the ratio of an eigenvalue to the sum of all eigenvalues demonstrates how many variations can be explained in the direction of its corresponding eigenvector.

# Appendix: pseudo codes

**Algorithm 1** An algorithm to calculate spot rates

```
Ensure:
  S ← []                                        ▷ S is the list of spot rates
  i ← 1                                         ▷ loop start with the first bond
  while i ≤ 10 do
      N ← n_i                 ▷ n_i is the total payments left before maturity for bond i
      P ← P_i                                   ▷ P_i is the dirty price for bond i
      CP ← cp_i                                 ▷ cp_i is the coupon payment for bond i
      F ← 100                                   ▷ F is the notional
      T_i ← [t_1, ··· , t_N]                    ▷ t_j is the time to j-th payment, where 1 ≤ j ≤ N
      while N > 0 do
          if N =1 then
              s_i ← (F/P)^(1/t_i) − 1           ▷ The first two bonds matures within six months
              S ← S + [s_i]
              N ← N − 1
          else
              y ← P − Σ_{j=1}^{N−1} CP/(1+s_j)^(t_j)
              s_i = (F/y)^(1/t_N)
              S = S + [s_i]
              N ← N − 1
          end if
      end while
      i ← i + 1
  end while
```

**Algorithm 2** An algorithm to calculate forward rates

```
  i ← 1
  F ← []
  P_j ← p_j                 ▷ where p_j is the price of the bond that matures at time j
  while i ≤ 5 do
      f = −(logP_{1+i} − logP_1)/i
      F = F + [f]
      i = i + 0.5
  end while
```

# References and GitHub Link to Code


[1] Gallier, J. H.,& Quaintance, J. (2020). *Linear algebra and optimization with applications to machine learning.* World Scientific.


https://github.com/Ruii-W/Math-Finance-A1