A Simple Bootstrapping Algrithm For Calulating Projected Bond Yield Rates

This some mathematica code I used to implement a simple bootstrapping algorithm in an effort to compute projected bond yield rates (along with some other information related to the resulting time series) for the next 5 years. The data ragarding bond prices, coupon payments, and advertized yield rates all came from http://www.pfin.ca/canadianfixedincome/Default.aspx, and was collected over the span of 10 consecutive business days, starting on January 13, 2016.

A note on our basic assumptions, to compute the desired yield/forward rates, we assumed that the yeild rate is constant for the first year, where the “first year” means the date the data was collected to that date in the subsequent year (e.g. January 13, 2016 to January 13, 2017), as a matter of convention we take the first year rate to be the projected rate on the date one year after the date the data was collected, (e.g. if the data was collected on January 13, 2016, then the one year rate corresponds to January 13, 2017). After the first year, we use simple linear interpolation to compute the yield rates for the following years, again taking the n^{th} year rate to always fall 365 days (or 366 for leap years) after what was taken to be the $(n-1)^{\text{th}}$ year rate.

```
s = Import["Documents/1850-1.xlsx"] (*  Importing the data from excel  *)
```

```
P = {{"", "Canada", 4.`, "2016-Jun-01", 101.37`, 0.35`, 101.35`,
   0.34`, 101.35`, 0.3`, 101.34`, 0.32`, 101.33`, 0.31`, 101.29`, 0.39`,
   101.24`, 0.43`, 101.23`, 0.44`, 101.22`, 0.44`, 101.21`, 0.44`},
  {"", "Canada", 4.`, "2017-Jun-01", 105.04`, 0.32`, 105.`, 0.34`,
   105.04`, 0.3`, 104.99`, 0.33`, 105.02`, 0.3`, 104.88`, 0.39`,
   104.78`, 0.44`, 104.77`, 0.45`, 104.8`, 0.42`, 104.77`, 0.43`},
  {"", "Canada", 4.25`, "2018-Jun-01", 109.3`, 0.32`, 109.24`, 0.33`,
   109.31`, 0.3`, 109.22`, 0.33`, 109.26`, 0.31`, 109.02`, 0.4`,
   108.86`, 0.46`, 108.82`, 0.47`, 108.89`, 0.43`, 108.85`, 0.45`},
  {"", "Canada", 3.75`, "2019-Jun-01", 111.3`, 0.37`, 111.27`, 0.38`,
   111.4`, 0.34`, 111.28`, 0.37`, 111.3`, 0.36`, 111.01`, 0.43`,
   110.8`, 0.49`, 110.75`, 0.51`, 110.87`, 0.47`, 110.82`, 0.48`},
  {"", "Canada", 3.5`, "2020-Jun-01", 113.17`, 0.45`, 113.14`, 0.46`,
   113.41`, 0.4`, 113.27`, 0.43`, 113.25`, 0.43`, 112.99`, 0.48`,
   112.66`, 0.55`, 112.42`, 0.6`, 112.65`, 0.55`, 112.62`, 0.55`}}
```

```
{{, Canada, 4., 2016-Jun-01, 101.37, 0.35, 101.35, 0.34, 101.35, 0.3, 101.34, 0.32,
  101.33, 0.31, 101.29, 0.39, 101.24, 0.43, 101.23, 0.44, 101.22, 0.44, 101.21, 0.44},
 {, Canada, 4., 2017-Jun-01, 105.04, 0.32, 105., 0.34, 105.04, 0.3, 104.99, 0.33,
  105.02, 0.3, 104.88, 0.39, 104.78, 0.44, 104.77, 0.45, 104.8, 0.42, 104.77, 0.43},
 {, Canada, 4.25, 2018-Jun-01, 109.3, 0.32, 109.24, 0.33, 109.31, 0.3, 109.22, 0.33,
  109.26, 0.31, 109.02, 0.4, 108.86, 0.46, 108.82, 0.47, 108.89, 0.43, 108.85, 0.45},
 {, Canada, 3.75, 2019-Jun-01, 111.3, 0.37, 111.27, 0.38, 111.4, 0.34, 111.28, 0.37,
  111.3, 0.36, 111.01, 0.43, 110.8, 0.49, 110.75, 0.51, 110.87, 0.47, 110.82, 0.48},
 {, Canada, 3.5, 2020-Jun-01, 113.17, 0.45, 113.14, 0.46, 113.41, 0.4, 113.27, 0.43,
  113.25, 0.43, 112.99, 0.48, 112.66, 0.55, 112.42, 0.6, 112.65, 0.55, 112.62, 0.55}}
```

```
(*  the data can be cleaned up a little ...  *)
i = 1;
While[i ≤ 5,
 P[[i]] = Delete[P[[i]],
   {{1}, {2}, {6}, {8}, {10}, {12}, {14}, {16}, {18}, {20}, {22}, {24}}];
 i++
]
```

```
P
{{4., 2016-Jun-01, 101.37, 101.35, 101.35,
  101.34, 101.33, 101.29, 101.24, 101.23, 101.22, 101.21},
 {4., 2017-Jun-01, 105.04, 105., 105.04, 104.99, 105.02, 104.88, 104.78,
  104.77, 104.8, 104.77}, {4.25, 2018-Jun-01, 109.3, 109.24,
  109.31, 109.22, 109.26, 109.02, 108.86, 108.82, 108.89, 108.85},
 {3.75, 2019-Jun-01, 111.3, 111.27, 111.4, 111.28, 111.3, 111.01,
  110.8, 110.75, 110.87, 110.82}, {3.5, 2020-Jun-01, 113.17, 113.14,
  113.41, 113.27, 113.25, 112.99, 112.66, 112.42, 112.65, 112.62}}

(*
  Here we point out that P[[i]]] is a list of
 data correponding to a single Government of Canada bond
 P[[i,1]] is the annual coupon payment for that bond
 P[[i,2]] is the effective maturity date of the bond
 P[[i, 3 ;; 12]] are the prices of the bond recorded on the 10 subsequent days
*)
```

```
(*  this is the function we will use to
 compute all of the desired yield/forward rates  *)
Rates[P_] := Module[{i, M, R, r1,
   midr2, m2, b2, y2, r2,
   midr3, m3, b3, y3, r3,
   midr4, m4, b4, y4, r4,
   midr5, m5, b5, y5, r5},
  i = 0; (*  initializing our counter  *)
  M = {}; (*  This will be a list of lists, where each nested list contains
   the forward rates for the fives subsequent years following 2016  *)
  R = {};  (*  This will be a list of lists, where each nested list contains
   the yield rates for the fives subsequent years following 2016  *)
  While[i ≤ 9,
   (*
   We start off by calculating the first year yield rate. Here
     we assume that the rate is constant over the first year -
    so to linearly interpolate for this year is trivial.
   *)
   (*  r1 is the (constant) first year rate  *)
   r1 = -365 / DateDifference[{2016, 1, 13 + i}, {2016, 6, 1}]
     Log[(P[[1, 3 + i]] + P[[1, 1]] (1/365 DateDifference[
            {2015, 12, 1}, {2016, 1, 13 + i}])) / (100 + P[[1, 1]]/2)];
```

```
(*
Next we compute the second year rate. Since from the second year
 onward we are linearly interpolating between years. We first use
 our data to calculate the yield rate on June 6th, 2017 and then
 linearly interpolate to get the yield rate for January (13+i)th, 2018.
*)
```

$$\texttt{midr2} = \frac{-365}{\texttt{DateDifference[\{2016, 1, 13 + i\}, \{2017, 6, 1\}]}} \texttt{Log}\left[\left(\left(P[[2, 3 + i]] + P[[2, 1]] \left(\frac{1}{365}\texttt{DateDifference[\{2015, 12, 1\}, \{2016, 1, 13 + i\}]}\right)\right) - \frac{P[[2, 1]]}{2} \left(E^{-r1\,\texttt{DateDifference[\{2016, 1, 13 + i\},\{2016, 6, 1\}]}/365} + E^{-r1\,\texttt{DateDifference[\{2016, 1, 13 + i\},\{2016, 12, 1\}]}/365}\right)\right) \Big/ \left(100 + \frac{P[[2, 1]]}{2}\right)\right];$$

```
(*
the following two computations calculate the slope and y-intercept of
  the line we will use in the interpolation of the second year rate

  namely, we want    y2 = m2*x + b2     knowing the rate for the dates,
January (13+i)th, 2017 and June 6th, 2017
*)
```

$$\texttt{m2} = \frac{\texttt{midr2 - r1}}{\texttt{DateDifference[\{2017, 1, 13 + i\}, \{2017, 6, 1\}]}};$$

```
b2 = r1 - m2 * DateDifference[{2016, 1, 13}, {2017, 1, 13}];
(*
Next is the equation for the interpolation line
*)
y2[y_, m_, d_] := m2 * DateDifference[{2016, 1, 13 + i}, {y, m, d}] + b2;
(*
and finally the second year rate!
*)
r2 = y2[2018, 1, 13 + i];
```

```
(*
From here on we are more or less repeating a
 structurally similar computation to compute the rates of subsequent
 years using the rates calculated for the previous years.
```

```
*)

(*  THIRD YEAR RATE  *)
midr3 = -365 / DateDifference[{2016, 1, 13 + i}, {2018, 6, 1}]
  Log[((P[[3, 3 + i]] + P[[3, 1]] (1/365 DateDifference[{2015, 12, 1}, {2016, 1,
          13 + i}])) - P[[3, 1]]/2 (E^(-r1 DateDifference[{2016, 1, 13 + i}, {2016, 6, 1}]/365) +
        E^(-r1 DateDifference[{2016, 1, 13 + i}, {2016, 12, 1}]/365) +
        E^(-y2[2017, 6, 1] DateDifference[{2016, 1, 13 + i}, {2017, 6, 1}]/365) +
        E^(-y2[2017, 12, 1] DateDifference[{2016, 1, 13 + i}, {2017, 12, 1}]/365))) /
    (100 + P[[3, 1]]/2)];
m3 = (midr3 - r2) / DateDifference[{2018, 1, 13 + i}, {2018, 6, 1}];
b3 = r2 - m3 * DateDifference[{2016, 1, 13}, {2018, 1, 13}];
y3[y_, m_, d_] := m3 * DateDifference[{2016, 1, 13 + i}, {y, m, d}] + b3;
r3 = y3[2019, 1, 13 + i];


(*  FOURTH YEAR RATE  *)

midr4 = -365 / DateDifference[{2016, 1, 13 + i}, {2019, 6, 1}]
  Log[((P[[4, 3 + i]] + P[[4, 1]] (1/365 DateDifference[{2015, 12, 1}, {2016, 1,
          13 + i}])) - P[[4, 1]]/2 (E^(-r1 DateDifference[{2016, 1, 13 + i}, {2016, 6, 1}]/365) +
        E^(-r1 DateDifference[{2016, 1, 13 + i}, {2016, 12, 1}]/365) +
        E^(-y2[2017, 6, 1] DateDifference[{2016, 1, 13 + i}, {2017, 6, 1}]/365) +
        E^(-y2[2017, 12, 1] DateDifference[{2016, 1, 13 + i}, {2017, 12, 1}]/365) +
        E^(-y3[2018, 6, 1] DateDifference[{2016, 1, 13 + i}, {2018, 6, 1}]/365) +
        E^(-y3[2018, 12, 1] DateDifference[{2016, 1, 13 + i}, {2018, 12, 1}]/365))) /
    (100 + P[[4, 1]]/2)];
m4 = (midr4 - r3) / DateDifference[{2019, 1, 13 + i}, {2019, 6, 1}];
b4 = r3 - m4 * DateDifference[{2016, 1, 13}, {2019, 1, 13}];
y4[y_, m_, d_] := m4 * DateDifference[{2016, 1, 13 + i}, {y, m, d}] + b4;
r4 = y4[2020, 1, 13 + i];
```

```
(*  FIFTH YEAR RATE  *)

midr5 = -365 / DateDifference[{2016, 1, 13 + i}, {2020, 6, 1}]
  Log[((P[[5, 3 + i]] + P[[5, 1]] (1/365 DateDifference[{2015, 12, 1}, {2016, 1,
          13 + i}])) - P[[5, 1]]/2 (E^(-r1 DateDifference[{2016, 1, 13 + i}, {2016, 6, 1}]/365) +
        E^(-r1 DateDifference[{2016, 1, 13 + i}, {2016, 12, 1}]/365) +
        E^(-y2[2017, 6, 1] DateDifference[{2016, 1, 13 + i}, {2017, 6, 1}]/365) +
        E^(-y2[2017, 12, 1] DateDifference[{2016, 1, 13 + i}, {2017, 12, 1}]/365) +
        E^(-y3[2018, 6, 1] DateDifference[{2016, 1, 13 + i}, {2018, 6, 1}]/365) +
        E^(-y3[2018, 12, 1] DateDifference[{2016, 1, 13 + i}, {2018, 12, 1}]/365) +
        E^(-y4[2019, 6, 1] DateDifference[{2016, 1, 13 + i}, {2019, 6, 1}]/365) +
        E^(-y4[2019, 12, 1] DateDifference[{2016, 1, 13 + i}, {2019, 12, 1}]/365))) /
    (100 + P[[5, 1]]/2)];
m5 = (midr5 - r4) / DateDifference[{2020, 1, 13 + i}, {2020, 6, 1}];
b5 = r4 - m5 * DateDifference[{2016, 1, 13}, {2020, 1, 13}];
y5[y_, m_, d_] := m5 * DateDifference[{2016, 1, 13 + i}, {y, m, d}] + b5;
r5 = y5[2021, 1, 13 + i];


(*
And so before we move on to computing the rates for the next
 days worth of data, we append the data to the appropriate lists.
*)
AppendTo[M, {(r2 - r1)/2, (r3 - r1)/2, (r4 - r2)/2,
   (r5 - r3)/2, m5 * DateDifference[{2020, 1, 13}, {2021, 1, 13}]}]
(*  a list of forward rates  *);
AppendTo[R, {r1, r2, r3, r4, r5}] (*  a list of yield rates  *);
i++;
 ];
 {R, M}
]
```

```
info = Rates[P] (*  storing our computed information  *)
```

```
{{{0.00406128, 0.00232083, 0.00481792, 0.00226473, 0.00856975},
  {0.00432363, 0.0024309, 0.00517586, 0.00178727, 0.00951595},
  {0.00407034, 0.00190028, 0.00522825, 0.00062255, 0.00989108},
  {0.00407496, 0.00264498, 0.00484462, 0.00208564, 0.00823739},
  {0.00407966, 0.00185147, 0.00568354, 0.000409365, 0.0112526},
  {0.00488095, 0.0029617, 0.00629977, 0.00144214, 0.0109714},
  {0.00596204, 0.00279596, 0.00827695, -0.000535475, 0.0164393},
  {0.00598101, 0.00274638, 0.00875193, -0.00112259, 0.0190696},
  {0.00600027, 0.0018877, 0.00943751, -0.00346877, 0.0219679},
  {0.00601982, 0.00223482, 0.0091985, -0.0028014, 0.0210716}},
 {{-0.000870228, 0.000378318, -0.0000280484, 0.00187591, 0.00630502},
  {-0.000946367, 0.000426114, -0.000321815, 0.00217004, 0.00772868},
  {-0.00108503, 0.000578958, -0.000638863, 0.00233141, 0.00926853},
  {-0.000714992, 0.00038483, -0.000279672, 0.00169638, 0.00615175},
  {-0.0011141, 0.000801944, -0.00072105, 0.00278454, 0.0108433},
  {-0.000959621, 0.000709414, -0.000759784, 0.00233581, 0.00952925},
  {-0.00158304, 0.00115746, -0.00166572, 0.00408116, 0.0169747},
  {-0.00161731, 0.00138546, -0.00193448, 0.00515882, 0.0201922},
  {-0.00205628, 0.00171862, -0.00267823, 0.0062652, 0.0254367},
  {-0.0018925, 0.00158934, -0.00251811, 0.00593653, 0.023873}}}
```

```
(*  We now calculate a time series of daily log returns -
 in this cell it is for the yield rates,
and two cells below it is for the forward rates  *)
LRr = {};
i = 1;
While[i ≤ 9,
 AppendTo[LRr, Log[Abs[info[[1]][[i + 1]] / info[[1]][[i]]]]];
 i++
]
```

```
LRr
```

```
{{0.0625961, 0.0463357, 0.0716629, -0.236769, 0.104731},
 {-0.0603698, -0.246261, 0.010072, -1.05462, 0.0386638},
 {0.00113575, 0.330664, -0.076208, 1.20901, -0.18295},
 {0.0011512, -0.356685, 0.159706, -1.62822, 0.311918},
 {0.179326, 0.469788, 0.102939, 1.25928, -0.0253104},
 {0.200073, -0.0575902, 0.272961, -0.990727, 0.404382},
 {0.00317694, -0.0178917, 0.0558002, 0.74024, 0.148421},
 {0.00321482, -0.374926, 0.075418, 1.12816, 0.141489},
 {0.00325341, 0.168801, -0.0256521, -0.213679, -0.0416584}}
```

```
(*  Daily log returns for the forward rates   *)
LFr = {};
i = 1;
While[i ≤ 9,
 AppendTo[LFr, Log[Abs[info[[2]][[i + 1]] / info[[2]][[i]]]]];
 i++
]
LFr
```

```
{{0.0838755, 0.118972, 2.44004, 0.145651, 0.203592},
 {0.136733, 0.306524, 0.685715, 0.071726, 0.181686},
 {-0.417092, -0.408429, -0.826074, -0.317976, -0.409887},
 {0.443527, 0.734237, 0.947093, 0.495584, 0.566806},
 {-0.14926, -0.122599, 0.0523249, -0.175725, -0.129177},
 {0.500564, 0.489541, 0.784977, 0.558023, 0.57736},
 {0.0214206, 0.179809, 0.149585, 0.234327, 0.173568},
 {0.240134, 0.21549, 0.325316, 0.194303, 0.230898},
 {-0.0830013, -0.0782046, -0.0616486, -0.0538853, -0.0634455}}
```

```
(*  Finally we calculate the eigenvalues and eigenvectors
 corresponding to the covariance matrices of the time series
 daily log returns for the yield and forward rates  *)
Eigensystem[Covariance[LRr]]
```

```
{{1.25562, 0.0662008, 0.0284214, 0.00102104, 0.0000841264},
 {{-0.00782172, -0.151119, 0.0423905, -0.982281, 0.102117},
  {0.116473, 0.923181, -0.0835613, -0.178649, -0.308663},
  {-0.474252, -0.22254, -0.520259, -0.054327, -0.672265},
  {-0.645497, 0.264292, -0.34063, 0.0153008, 0.630254},
  {0.58719, -0.0743344, -0.77751, -0.00470608, 0.212459}}}
```

```
Eigensystem[Covariance[LFr]]
```

```
{{0.971992, 0.213876, 0.00669829, 0.00318292, 0.0000777616},
 {{-0.212832, -0.243819, -0.889385, -0.205724, -0.248853},
  {0.416707, 0.500609, -0.456269, 0.424362, 0.432991},
  {0.139456, -0.772171, -0.00641424, 0.597082, 0.166604},
  {0.811294, -0.255298, -0.00836482, -0.525482, 0.0205786},
  {-0.321577, -0.168873, -0.0265015, -0.380732, 0.849948}}}
```