

I gathered data on bond prices (issued by the Government of Canada) over a period of two weeks. I tracked five bonds, which mature on June 1, 2016, 2017, 2018, 2019, and 2020, respectively. I wrote a program to compute the yield rates for each day, for each of those five years.

Here is my data entry. Each list includes the price of that bond on that day, the coupon payment in dollars, and the months to maturity (which was always 4, since these bonds all happen to mature on June 1).

```
jan13year1 = {100.62, 2, 4}
jan14year1 = {100.61, 2, 4}
jan15year1 = {100.62, 2, 4}
jan18year1 = {100.61, 2, 4}
jan19year1 = {100.61, 2, 4}
jan20year1 = {100.58, 2, 4}
jan21year1 = {100.55, 2, 4}
jan22year1 = {100.54, 2, 4}
jan25year1 = {100.53, 2, 4}
jan26year1 = {100.53, 2, 4}
```

```
jan13year2 = {105.04, 4, 4}
jan14year2 = {105.00, 4, 4}
jan15year2 = {105.04, 4, 4}
jan18year2 = {104.99, 4, 4}
jan19year2 = {105.02, 4, 4}
jan20year2 = {104.88, 4, 4}
jan21year2 = {104.78, 4, 4}
jan22year2 = {104.77, 4, 4}
jan25year2 = {104.80, 4, 4}
jan26year2 = {104.77, 4, 4}
```

```
jan13year3 = {109.30, 4.25, 4}
jan14year3 = {109.24, 4.25, 4}
jan15year3 = {109.31, 4.25, 4}
jan18year3 = {109.22, 4.25, 4}
jan19year3 = {109.26, 4.25, 4}
jan20year3 = {109.02, 4.25, 4}
jan21year3 = {108.86, 4.25, 4}
jan22year3 = {108.82, 4.25, 4}
jan25year3 = {108.89, 4.25, 4}
jan26year3 = {108.85, 4.25, 4}
```

jan13year4 = {111.30, 3.75, 4}
jan14year4 = {111.27, 3.75, 4}
jan15year4 = {111.40, 3.75, 4}
jan18year4 = {111.28, 3.75, 4}
jan19year4 = {111.30, 3.75, 4}
jan20year4 = {111.01, 3.75, 4}
jan21year4 = {110.80, 3.75, 4}
jan22year4 = {110.75, 3.75, 4}
jan25year4 = {110.87, 3.75, 4}
jan26year4 = {110.82, 3.75, 4}

jan13year5 = {113.17, 3.5, 4}
jan14year5 = {113.14, 3.5, 4}
jan15year5 = {113.41, 3.5, 4}
jan18year5 = {113.27, 3.5, 4}
jan19year5 = {113.25, 3.5, 4}
jan20year5 = {112.99, 3.5, 4}
jan21year5 = {112.66, 3.5, 4}
jan22year5 = {112.42, 3.5, 4}
jan25year5 = {112.65, 3.5, 4}
jan26year5 = {112.62, 3.5, 4}

{100.62, 2, 4}

{100.61, 2, 4}

{100.62, 2, 4}

{100.61, 2, 4}

{100.61, 2, 4}

{100.58, 2, 4}

{100.55, 2, 4}

{100.54, 2, 4}

{100.53, 2, 4}

{100.53, 2, 4}

{105.04, 4, 4}

{105., 4, 4}

{105.04, 4, 4}

{104.99, 4, 4}

{105.02, 4, 4}

- {104.88, 4, 4}
- {104.78, 4, 4}
- {104.77, 4, 4}
- {104.8, 4, 4}
- {104.77, 4, 4}
- {109.3, 4.25, 4}
- {109.24, 4.25, 4}
- {109.31, 4.25, 4}
- {109.22, 4.25, 4}
- {109.26, 4.25, 4}
- {109.02, 4.25, 4}
- {108.86, 4.25, 4}
- {108.82, 4.25, 4}
- {108.89, 4.25, 4}
- {108.85, 4.25, 4}
- {111.3, 3.75, 4}
- {111.27, 3.75, 4}
- {111.4, 3.75, 4}
- {111.28, 3.75, 4}
- {111.3, 3.75, 4}
- {111.01, 3.75, 4}
- {110.8, 3.75, 4}
- {110.75, 3.75, 4}
- {110.87, 3.75, 4}
- {110.82, 3.75, 4}
- {113.17, 3.5, 4}
- {113.14, 3.5, 4}
- {113.41, 3.5, 4}
- {113.27, 3.5, 4}
- {113.25, 3.5, 4}
- {112.99, 3.5, 4}

```
{112.66, 3.5, 4}
{112.42, 3.5, 4}
{112.65, 3.5, 4}
{112.62, 3.5, 4}
```

```
jan13 = {jan13year1, jan13year2, jan13year3, jan13year4, jan13year5}
jan14 = {jan14year1, jan14year2, jan14year3, jan14year4, jan14year5}
jan15 = {jan15year1, jan15year2, jan15year3, jan15year4, jan15year5}
jan18 = {jan18year1, jan18year2, jan18year3, jan18year4, jan18year5}
jan19 = {jan19year1, jan19year2, jan19year3, jan19year4, jan19year5}
jan20 = {jan20year1, jan20year2, jan20year3, jan20year4, jan20year5}
jan21 = {jan21year1, jan21year2, jan21year3, jan21year4, jan21year5}
jan22 = {jan22year1, jan22year2, jan22year3, jan22year4, jan22year5}
jan25 = {jan25year1, jan25year2, jan25year3, jan25year4, jan25year5}
jan26 = {jan26year1, jan26year2, jan26year3, jan26year4, jan26year5}
```

```
{{100.62, 2, 4}, {105.04, 4, 4}, {109.3, 4.25, 4}, {111.3, 3.75, 4}, {113.17, 3.5, 4}}
{{100.61, 2, 4}, {105., 4, 4}, {109.24, 4.25, 4}, {111.27, 3.75, 4}, {113.14, 3.5, 4}}
{{100.62, 2, 4}, {105.04, 4, 4}, {109.31, 4.25, 4}, {111.4, 3.75, 4}, {113.41, 3.5, 4}}
{{100.61, 2, 4}, {104.99, 4, 4}, {109.22, 4.25, 4}, {111.28, 3.75, 4}, {113.27, 3.5, 4}}
{{100.61, 2, 4}, {105.02, 4, 4}, {109.26, 4.25, 4}, {111.3, 3.75, 4}, {113.25, 3.5, 4}}
{{100.58, 2, 4}, {104.88, 4, 4}, {109.02, 4.25, 4}, {111.01, 3.75, 4}, {112.99, 3.5, 4}}
{{100.55, 2, 4}, {104.78, 4, 4}, {108.86, 4.25, 4}, {110.8, 3.75, 4}, {112.66, 3.5, 4}}
{{100.54, 2, 4}, {104.77, 4, 4}, {108.82, 4.25, 4}, {110.75, 3.75, 4}, {112.42, 3.5, 4}}
{{100.53, 2, 4}, {104.8, 4, 4}, {108.89, 4.25, 4}, {110.87, 3.75, 4}, {112.65, 3.5, 4}}
{{100.53, 2, 4}, {104.77, 4, 4}, {108.85, 4.25, 4}, {110.82, 3.75, 4}, {112.62, 3.5, 4}}
```

```
solveoneyearrate[date_] := Solve[
  (100 + (date[[1]][[2]]/2)) (1 + r)^-date[[1]][[3]]/12 ==
  date[[1]][[1]] + date[[1]][[2]] * ((6 - date[[1]][[3]])/12), r
]
```

```
oneyearrate[date_] := (r /. solveoneyearrate[date])[[1]]
```

```
solvetwoyearrate[date_] :=
```

```
Solve[(date[[2]][[2]]/2) (1 + oneyearrate[date])^-date[[2]][[3]]/12 +
  (date[[2]][[2]]/2) (1 + oneyearrate[date])^-(date[[2]][[3]]+6)/12 +
  (100 + date[[2]][[2]]/2) ((1 + ((12 - date[[2]][[3]])/12) oneyearrate[date] +
  (date[[2]][[3]]/12) r))^-date[[2]][[3]]/12 ==
  date[[2]][[1]] + date[[2]][[2]] * ((6 - date[[2]][[3]])/12), r]
```

```

twoyearrate[date_] := (r /. solvetwoyearrate[date])[1]

solvetwoyearrate[date_] := Solve[
  (date[[3]][[2]]/2) (1 + oneyearrate[date])-date[[3]][[3]]/12
  + (date[[3]][[2]]/2) (1 + oneyearrate[date])-(date[[3]][[3]]+6)/12
  + (date[[3]][[2]]/2) (1 + (
    ((12 - date[[3]][[3]])/12) oneyearrate[date]
    + (date[[3]][[3]]/12) twoyearrate[date]))-(date[[3]][[3]]+12)/12
  + (date[[3]][[2]]/2) (1 + (
    ((6 - date[[3]][[3]])/12) oneyearrate[date]
    + ((date[[3]][[3]] + 6)/12) twoyearrate[date]))-(date[[3]][[3]]+18)/12
  + (100 + date[[3]][[2]]/2) ((1 + ((12 - date[[3]][[3]])/12) twoyearrate[date] +
    r (date[[3]][[3]]/12))-(date[[3]][[3]]+24)/12 ==
  date[[3]][[1]] + date[[3]][[2]] * ((6 - date[[3]][[3]])/12)
  ,
  r]

threeyearrate[date_] := (r /. solvetwoyearrate[date])[1]

solvetwoyearrate[date_] := Solve[
  (date[[4]][[2]]/2) (1 + oneyearrate[date])-date[[4]][[3]]/12
  + (date[[4]][[2]]/2) (1 + oneyearrate[date])-(date[[4]][[3]]+6)/12
  + (date[[4]][[2]]/2) (1 + (
    ((12 - date[[4]][[3]])/12) oneyearrate[date]
    + (date[[4]][[3]]/12) twoyearrate[date]))-(date[[4]][[3]]+12)/12
  + (date[[4]][[2]]/2) (1 + (
    ((6 - date[[4]][[3]])/12) oneyearrate[date]
    + ((date[[4]][[3]] + 6)/12) twoyearrate[date]))-(date[[4]][[3]]+18)/12
  + (date[[4]][[2]]/2) ((1 + ((12 - date[[4]][[3]])/12) twoyearrate[date] +
    (date[[4]][[3]]/12) threeyearrate[date]))-(date[[4]][[3]]+24)/12
  + (date[[4]][[2]]/2) (1 + (
    ((6 - date[[4]][[3]])/12) twoyearrate[date]
    + ((date[[4]][[3]] + 6)/12) threeyearrate[date]))-(date[[4]][[3]]+30)/12
  + (100 + (date[[4]][[2]]/2) (1 + ((12 - date[[4]][[3]])/12) threeyearrate[date] +
    (date[[4]][[3]]/12) r)-(date[[4]][[3]]+36)/12 ==
  date[[4]][[1]] + date[[4]][[2]] * ((6 - date[[4]][[3]])/12)
  ,
  r]

fouryearrate[date_] := (r /. solvetwoyearrate[date])[1]

```

```

solvefiveyearrate[date_] := Solve[
  (date[[5]][[2]]/2) (1 + oneyearrate[date])-date[[5]][[3]]/12
  + (date[[5]][[2]]/2) (1 + oneyearrate[date])-(date[[5]][[3]]+6)/12
  + (date[[5]][[2]]/2) (1 + (
    ((12 - date[[5]][[3]])/12) oneyearrate[date]
    + (date[[5]][[3]]/12) twoyearrate[date]))-(date[[5]][[3]]+12)/12
  + (date[[5]][[2]]/2) (1 + (
    ((6 - date[[5]][[3]])/12) oneyearrate[date]
    + ((date[[5]][[3]] + 6)/12) twoyearrate[date]))-(date[[5]][[3]]+18)/12
  + (date[[5]][[2]]/2) ((1 + ((12 - date[[5]][[3]])/12) twoyearrate[date] +
    (date[[5]][[3]]/12) threeyearrate[date]))-(date[[5]][[3]]+24)/12
  + (date[[5]][[2]]/2) (1 + (
    ((6 - date[[5]][[3]])/12) twoyearrate[date]
    + ((date[[5]][[3]] + 6)/12) threeyearrate[date]))-(date[[5]][[3]]+30)/12
  + (date[[5]][[2]]/2) (1 + ((12 - date[[5]][[3]])/12) threeyearrate[date] +
    (date[[5]][[3]]/12) fouryearrate[date])-(date[[5]][[3]]+36)/12
  + (date[[5]][[2]]/2) (1 + (
    ((6 - date[[5]][[3]])/12) threeyearrate[date]
    + ((date[[5]][[3]] + 6)/12) fouryearrate[date]))-(date[[5]][[3]]+42)/12
  + (100 + (date[[5]][[2]]/2)) (1 + ((12 - date[[5]][[3]])/12) fouryearrate[date] +
    (date[[5]][[3]]/12) r)-(date[[5]][[3]]+48)/12
  == date[[5]][[1]] + date[[5]][[2]] * ((6 - date[[5]][[3]])/12)
  , r]

fiveyearrate[date_] := (r /. solvefiveyearrate[date])[1]

```

As you can see, I wrote the functions to find the rates using the Solve function, and then I defined a new function below (which returns a number, not a list) so that I could use it in the successive functions (and below).

I will now set up the table I will use to display the data. In the formulae for the table, I multiplied by 100 in some places so that the numbers would be displayed as percentages.

```

daylist = {jan13, jan14, jan15, jan18, jan19, jan20, jan21, jan22, jan25, jan26}
{{{100.62, 2, 4}, {105.04, 4, 4}, {109.3, 4.25, 4}, {111.3, 3.75, 4}, {113.17, 3.5, 4}},
 {{100.61, 2, 4}, {105., 4, 4}, {109.24, 4.25, 4}, {111.27, 3.75, 4}, {113.14, 3.5, 4}},
 {{100.62, 2, 4}, {105.04, 4, 4}, {109.31, 4.25, 4}, {111.4, 3.75, 4},
 {113.41, 3.5, 4}}, {{100.61, 2, 4}, {104.99, 4, 4}, {109.22, 4.25, 4},
 {111.28, 3.75, 4}, {113.27, 3.5, 4}}, {{100.61, 2, 4}, {105.02, 4, 4},
 {109.26, 4.25, 4}, {111.3, 3.75, 4}, {113.25, 3.5, 4}}, {{100.58, 2, 4},
 {104.88, 4, 4}, {109.02, 4.25, 4}, {111.01, 3.75, 4}, {112.99, 3.5, 4}},
 {{100.55, 2, 4}, {104.78, 4, 4}, {108.86, 4.25, 4}, {110.8, 3.75, 4},
 {112.66, 3.5, 4}}, {{100.54, 2, 4}, {104.77, 4, 4}, {108.82, 4.25, 4},
 {110.75, 3.75, 4}, {112.42, 3.5, 4}}, {{100.53, 2, 4}, {104.8, 4, 4},
 {108.89, 4.25, 4}, {110.87, 3.75, 4}, {112.65, 3.5, 4}}, {{100.53, 2, 4},
 {104.77, 4, 4}, {108.85, 4.25, 4}, {110.82, 3.75, 4}, {112.62, 3.5, 4}}

```

```

daystringlist = {"Jan 13", "Jan 14", "Jan 15", "Jan 18",
  "Jan 19", "Jan 20", "Jan 21", "Jan 22", "Jan 25", "Jan 26"}
{Jan 13, Jan 14, Jan 15, Jan 18, Jan 19, Jan 20, Jan 21, Jan 22, Jan 25, Jan 26}

```

```

gridrow[n] := {daystringlist[[n]], 100 * oneyearrate[daylist[[n]]],
  100 * twoyearrate[daylist[[n]]], 100 * threeyearrate[daylist[[n]]],
  100 * fouryearrate[daylist[[n]]], 100 * fiveyearrate[daylist[[n]]]}

```

```

grid = Grid[{
  {"Date", "1-Year Rate", "2-Year Rate",
  "3-Year Rate", "4-Year Rate", "5-Year Rate"},
  gridrow[1], gridrow[2], gridrow[3], gridrow[4], gridrow[5],
  gridrow[6], gridrow[7], gridrow[8], gridrow[9], gridrow[10]
}]

```

Date	1-Year Rate	2-Year Rate	3-Year Rate	4-Year Rate	5-Year Rate
Jan 13	0.138742	0.364039	0.0235357	0.990109	-0.679539
Jan 14	0.168506	0.391681	0.0411054	0.977933	-0.637134
Jan 15	0.138742	0.364039	0.0108415	0.926547	-0.715141
Jan 18	0.168506	0.413867	0.0209769	1.0088	-0.79052
Jan 19	0.168506	0.347321	0.10676	0.81923	-0.390823
Jan 20	0.257868	0.474822	0.144698	0.989826	-0.571018
Jan 21	0.347337	0.513743	0.26323	0.929615	-0.229869
Jan 22	0.377183	0.474777	0.392424	0.711001	0.37858
Jan 25	0.407042	0.346726	0.564001	0.258023	1.13953
Jan 26	0.407042	0.413519	0.477957	0.475215	0.717901

After computing the yield curves for each day and for each year of maturity, I calculated the (rough) forward curves. The forward curve is essentially the derivative of the yield rate curve, but since I only have discrete values for the rates, I approximated the derivative by, at each corner, taking the “tangent” line to be the line that produces an equal angle on each side of the corner. I also had to approximate the derivative at the endpoints. This explains the somewhat strange formulae for the derivatives below.

```

year1forwardcurve[date_] := 100 * (twoyearrate[date] - oneyearrate[date]) / 2

```

```

year2forwardcurve[date_] := 100 * (threeyearrate[date] - oneyearrate[date]) / 2

```

```

year3forwardcurve[date_] := 100 * (fouryearrate[date] - twoyearrate[date]) / 2
year4forwardcurve[date_] := 100 * (fiveyearrate[date] - threeyearrate[date]) / 2
year5forwardcurve[date_] := 100 * (fiveyearrate[date] - fouryearrate[date])

forwardgridrow[n_] := {daystringlist[[n]], year1forwardcurve[daylist[[n]]],
  year2forwardcurve[daylist[[n]]], year3forwardcurve[daylist[[n]]],
  year4forwardcurve[daylist[[n]]], year5forwardcurve[daylist[[n]]]}

forwardgrid = Grid[{
  {"Date", "1-Year FRate", "2-Year FRate",
   "3-Year FRate", "4-Year FRate", "5-Year FRate"},
  forwardgridrow[1], forwardgridrow[2], forwardgridrow[3], forwardgridrow[4],
  forwardgridrow[5], forwardgridrow[6], forwardgridrow[7],
  forwardgridrow[8], forwardgridrow[9], forwardgridrow[10]
}]

Date 1-Year FRate 2-Year FRate 3-Year FRate 4-Year FRate 5-Year FRate
Jan 13 0.112648 -0.0576032 0.313035 -0.351537 -1.66965
Jan 14 0.111588 -0.0637003 0.293126 -0.33912 -1.61507
Jan 15 0.112648 -0.0639503 0.281254 -0.362991 -1.64169
Jan 18 0.122681 -0.0737645 0.297465 -0.405748 -1.79932
Jan 19 0.0894073 -0.0308727 0.235955 -0.248792 -1.21005
Jan 20 0.108477 -0.0565853 0.257502 -0.357858 -1.56084
Jan 21 0.0832031 -0.0420533 0.207936 -0.246549 -1.15948
Jan 22 0.0487968 0.00762035 0.118112 -0.00692179 -0.33242
Jan 25 -0.0301577 0.0784796 -0.0443515 0.287766 0.881509
Jan 26 0.00323864 0.0354577 0.030848 0.119972 0.242686

```

This is easily the most complex program I've written to date. I found *Mathematica* extremely useful in this case, as it allowed me to make huge computations quickly, whereas they would have taken hours by hand.