APM466A1

Shu Wang

2022/2/13

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.8
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(readr)
library(readxl)
selected_bonds = read_xlsx("bond_close price.xlsx")
view(selected_bonds)
```

```
selected_bonds <- as.data.frame(selected_bonds)
```

```
current_day = c(10,11,12,13,14,17,18,19,20,21)

t = 1
# Calculate each column DP_i for a given t.
while (t <= 10) {
  colname_DP = paste("DP", toString(t), sep="")

  # Calculate column DP_i at day t
  AI = (4*30 + (current_day[t]))/360 * selected_bonds[["Coupon"]] * 100
  DP = AI + selected_bonds[[paste("CPd", toString(t), sep="")]]

  selected_bonds[[colname_DP]] = DP
  t = t + 1
}
```

```
install.packages("jrvFinance")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.1'
## (as 'lib' is unspecified)
```

```
view(n)
```

```
library(jrvFinance)
# Calculate each column YTM_i for a given day t.
t = 1
while(t<=10){
```

```
  colname_YTM = paste("YTM",toString(t),sep="")
  n = (selected_bonds[["Term(M)"]]+2)/6 + 1

  # Calculate each element YTM_i,t for bond i at day t.
  i = 1
  YTM_t = c()
  while (i<=10) {
    cash_flow = c(-selected_bonds[[paste("DP", toString(t), sep="")]][i])
    pmt_time = c(0)
    # Create the cash flow and time for bond i at day t.
    m = 1
    while (m<=n[i]-1) {
      cash_flow = append(cash_flow, selected_bonds[["Coupon"]][i]*100/2)
      pmt_time = append(pmt_time, (60-(current_day[t]-1))/360+1/2)
      m = m + 1
    }
    cash_flow = append(cash_flow, selected_bonds[["Coupon"]][i]*100/2 + 100)
    pmt_time = append(pmt_time, (60-(current_day[t]-1))/360+(n[i]-1)/2)

    YTM_it = irr(cf = cash_flow, cf.freq = 2, comp.freq = Inf, cf.t = pmt_time)
    YTM_t = append(YTM_t, YTM_it)
    i = i + 1
 }

  selected_bonds[[colname_YTM]] = YTM_t
  t = t + 1
}
```

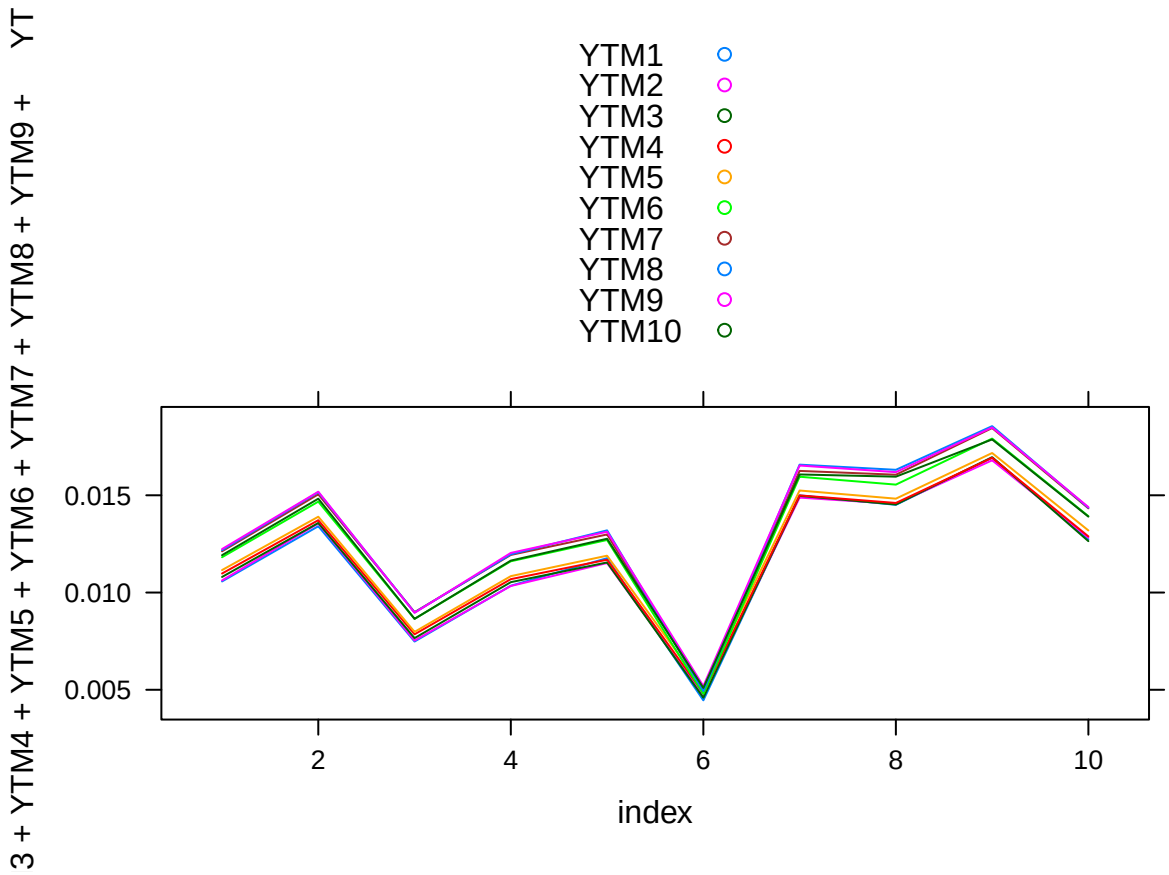
```
selected_bonds <- tibble::rowid_to_column(selected_bonds, "index")
```

```
require(tigerstats)
```

```
## Loading required package: tigerstats
## Loading required package: abd
## Loading required package: nlme
##
## Attaching package: 'nlme'
## The following object is masked from 'package:dplyr':
##
##     collapse
## Loading required package: lattice
## Loading required package: grid
## Loading required package: mosaic
## Registered S3 method overwritten by 'mosaic':
##   method                           from
##   fortify.SpatialPolygonsDataFrame ggplot2
##
## The 'mosaic' package masks several functions from core packages in order to add
## additional features.  The original behavior of these functions should not be affected by this.
```

```
##
## Attaching package: 'mosaic'

## The following object is masked from 'package:Matrix':
##
##     mean

## The following objects are masked from 'package:dplyr':
##
##     count, do, tally

## The following object is masked from 'package:purrr':
##
##     cross

## The following object is masked from 'package:ggplot2':
##
##     stat

## The following objects are masked from 'package:stats':
##
##     binom.test, cor, cor.test, cov, fivenum, IQR, median, prop.test,
##     quantile, sd, t.test, var

## The following objects are masked from 'package:base':
##
##     max, mean, min, prod, range, sample, sum

## Welcome to tigerstats!
## To learn more about this package, consult its website:
##  http://homerhanumat.github.io/tigerstats
```

```
xyplot(YTM1+YTM2+YTM3+YTM4+YTM5+YTM6+YTM7+YTM8+YTM9+YTM10~index,
       data=selected_bonds,type='l',auto.key=T)
```




```
# Calculate the column r_i at each day t.
```

```
# Time series for yield
```

```r
# Calculate the time series X_i for bond i.
i = 1
X_ytm = c()
while (i <= 10) {
  yield_i_j_1 = as.numeric(selected_bonds[i,28:36])
  yield_i_j = as.numeric(selected_bonds[i,27:35])
  X_ytm_i = log(yield_i_j_1/yield_i_j)
```

```
  X_ytm = append(X_ytm, X_ytm_i)
  i = i + 2
}
X_ytm = matrix(X_ytm, ncol=5, byrow=FALSE)
# View(X_ytm)
M_ytm = cov(X_ytm)
eigen_space_ytm = eigen(M_ytm)
eigen_space_ytm$values
```

```
## [1] 2.983782e-03 1.286982e-04 5.117518e-05 1.044970e-05 3.158246e-06
```

```
eigen_space_ytm$vectors
```

```
##            [,1]       [,2]        [,3]        [,4]       [,5]
## [1,] -0.3985625  0.2767223  0.04198979 -0.47361214  0.7338264
## [2,] -0.5672920  0.6364920 -0.19199810  0.09323351 -0.4769716
## [3,] -0.4847468 -0.5253156 -0.59372385  0.32392341  0.1778473
## [4,] -0.3623432 -0.4799719  0.25097752 -0.62320823 -0.4323842
## [5,] -0.3912234 -0.1094217  0.73883412  0.52314834  0.1241416
```

6 Eigenvalues and Eigenvectors

```
M_ytm = cov(X_ytm)
M_ytm
```

```
##              [,1]         [,2]         [,3]         [,4]         [,5]
## [1,] 0.0004879699 0.0006953254 0.0005552972 0.0004164351 0.0004606414
## [2,] 0.0006953254 0.0010150757 0.0007833690 0.0005715906 0.0006463142
## [3,] 0.0005552972 0.0007833690 0.0007558786 0.0005465571 0.0005526467
## [4,] 0.0004164351 0.0005715906 0.0005465571 0.0004292696 0.0004356445
## [5,] 0.0004606414 0.0006463142 0.0005526467 0.0004356445 0.0004890697
```