Adil\_Gokturk\_HW03.R

HAG

2020-03-30

# Adil Gokturk  
# FIN 659  
  
# HW3: INTEREST RATES  
# Textbook Reference: Section 4.4, pp. 81-83; Section 4.6, pp. 84-85;  
# Section 4.8, pp. 89-92; Section 4.9, pp. 92-94; Section 4.10, pp. 94-97   
  
# set working directory  
setwd("~/Desktop/Spring2020/FIN659/Assignments/hw3")  
getwd()

## [1] "/Users/HAG/Desktop/Spring2020/FIN659/Assignments/hw3"

# Load the libraries  
library(tidyverse)

## ── Attaching packages ────────────────────────────────────────────────────────── tidyverse 1.3.0 ──

## ✓ ggplot2 3.3.0 ✓ purrr 0.3.3  
## ✓ tibble 2.1.3 ✓ dplyr 0.8.5  
## ✓ tidyr 1.0.2 ✓ stringr 1.4.0  
## ✓ readr 1.3.1 ✓ forcats 0.5.0

## ── Conflicts ───────────────────────────────────────────────────────────── tidyverse\_conflicts() ──  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

library(quantmod)

## Loading required package: xts

## Loading required package: zoo

##   
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':  
##   
## as.Date, as.Date.numeric

##   
## Attaching package: 'xts'

## The following objects are masked from 'package:dplyr':  
##   
## first, last

## Loading required package: TTR

## Registered S3 method overwritten by 'quantmod':  
## method from  
## as.zoo.data.frame zoo

## Version 0.4-0 included new data defaults. See ?getSymbols.

library(optiRum)  
library(jrvFinance)  
library(knitr)  
  
## Problem 1   
# Go to theU.S. Department of the Treasury website,   
# and search for "Daily Treasury Yield Curve Rates" on Jan 2, 2019   
# I https://fred.stlouisfed.org/series/DGS1used   
# 1-Year Treasury Constant Maturity Rate (DGS1)  
  
## What is the quoted interest rate per annum for 1 Yr?  
# source: https://fred.stlouisfed.org/series/DGS1  
  
# I used quantmode to get the data form St. Louis FED  
getSymbols("DGS1",src="FRED")

## 'getSymbols' currently uses auto.assign=TRUE by default, but will  
## use auto.assign=FALSE in 0.5-0. You will still be able to use  
## 'loadSymbols' to automatically load data. getOption("getSymbols.env")  
## and getOption("getSymbols.auto.assign") will still be checked for  
## alternate defaults.  
##   
## This message is shown once per session and may be disabled by setting   
## options("getSymbols.warning4.0"=FALSE). See ?getSymbols for details.

## [1] "DGS1"

str(DGS1)

## An 'xts' object on 1962-01-02/2020-03-26 containing:  
## Data: num [1:15193, 1] 3.22 3.24 3.24 3.26 3.31 3.32 3.33 3.33 3.3 3.32 ...  
## - attr(\*, "dimnames")=List of 2  
## ..$ : NULL  
## ..$ : chr "DGS1"  
## Indexed by objects of class: [Date] TZ: UTC  
## xts Attributes:   
## List of 2  
## $ src : chr "FRED"  
## $ updated: POSIXct[1:1], format: "2020-03-30 06:09:31"

head(DGS1)

## DGS1  
## 1962-01-02 3.22  
## 1962-01-03 3.24  
## 1962-01-04 3.24  
## 1962-01-05 3.26  
## 1962-01-08 3.31  
## 1962-01-09 3.32

tail(DGS1)

## DGS1  
## 2020-03-19 0.20  
## 2020-03-20 0.15  
## 2020-03-23 0.17  
## 2020-03-24 0.25  
## 2020-03-25 0.19  
## 2020-03-26 0.13

# Let's get the Daily Treasury Yield Curve Rates" on Jan 2, 2019  
window(DGS1, start="2019-01-02", end="2019-01-02")

## DGS1  
## 2019-01-02 2.6

# Assign the Daily Treasury Yield Curve Rates" on Jan 2, 2019 as "yield"  
(yield.2019.1.2 <- coredata(window(DGS1, start="2019-01-02", end="2019-01-02")))

## DGS1  
## [1,] 2.6

# (date <- DGS1[14872,]) # another way of getting the "Daily Treasury Yield Curve Rates" on Jan 2, 2019  
  
# Answer: as of 2019-01-02, daily Treasury Yield Curve Rates is 2.6 %  
  
## Assume the interest rate is quoted with semiannual compounding.  
# What is the equivalent rate with   
# (a) annual compounding   
  
(annual.compounding <- ((((1 + (yield.2019.1.2/100)/2))^2-1) \* 100) %>% round(2))

## DGS1  
## [1,] 2.62

# Answer: Annual compounding rate =2.62 %  
  
# (b) continuous compounding?  
(continuous.compounding <-(2 \* log( 1+((yield.2019.1.2/100))/2)\* 100) %>% round(2))

## DGS1  
## [1,] 2.58

# Answer: continuous compounding rate =2.58 %  
  
## What is the quoted interest rate per annum for 30 Yr?   
# Source: https://fred.stlouisfed.org/series/DGS30  
  
# Let's get data form St. Louis FED  
getSymbols("DGS30",src="FRED") # 30-Year Treasury Constant Maturity Rate (DGS30)

## [1] "DGS30"

# let's check the structure of the data  
str(DGS30)

## An 'xts' object on 1977-02-15/2020-03-26 containing:  
## Data: num [1:11248, 1] 7.7 7.67 7.67 7.76 NA 7.77 7.81 7.82 7.79 7.8 ...  
## - attr(\*, "dimnames")=List of 2  
## ..$ : NULL  
## ..$ : chr "DGS30"  
## Indexed by objects of class: [Date] TZ: UTC  
## xts Attributes:   
## List of 2  
## $ src : chr "FRED"  
## $ updated: POSIXct[1:1], format: "2020-03-30 06:09:31"

# Sample value of the data  
head(DGS30, 1)

## DGS30  
## 1977-02-15 7.7

# The quoted interest rate per annum for 30 Yr on Jan 2, 2019  
window(DGS30, start="2019-01-02", end="2019-01-02")

## DGS30  
## 2019-01-02 2.97

# let's assign The quoted interest rate per annum for 30 Yr on Jan 2, 2019  
(date30years <- coredata(window(DGS30, start="2019-01-02", end="2019-01-02"))) # Let's just use the core value of the data

## DGS30  
## [1,] 2.97

# Answer: The quoted interest rate per annum for 30 Yr on Jan 2, 2019 = 2.97 %  
  
## Assume the interest rate is quoted with continuous compounding (although, in actuality it is not).  
##What is the equivalent rate with   
## (a) annual compounding?   
(annual.compounding2 <- ((exp(date30years/100)-1) \* 100) %>% round(2))

## DGS30  
## [1,] 3.01

# Answer: Annual compounding rate = 3.01 %  
  
## (b) quarterly compounding?  
(quarterly.compounding2 <- ((4 \*(exp(date30years/100/4)-1))\* 100) %>% round(2))

## DGS30  
## [1,] 2.98

# Answer: Quarterly compounding rate = 2.98 %  
  
## Problem 2  
##The following table gives Treasury zero rates and cash flows on a Treasury bond.   
## Assume the zero rates are continuously compounded.  
(principal <- 1000) # USD

## [1] 1000

(maturity <- c(.5, 1, 1.5, 2)) # years

## [1] 0.5 1.0 1.5 2.0

(zero.rates <- c(0.02, 0.023, 0.027, 0.032)) # %

## [1] 0.020 0.023 0.027 0.032

(coupon.payment <- c(20, 20, 20, 1020)) # USD

## [1] 20 20 20 1020

## Find the present value of the cash flows (coupon payment and principal) for each period.  
  
# Let's find PV of cash flows   
(coupon.payment \* exp((-zero.rates\*maturity))) %>% round(2)

## [1] 19.80 19.55 19.21 956.77

# Answer: PV of cash flows = $19.80 $19.55 $19.21 $956.77  
  
# What is the bond's theoretical price?  
(pv.bond <- sum((coupon.payment \* exp((-zero.rates\*maturity))))) %>% round(2)

## [1] 1015.32

# Answer: The bond's theoretical price = USD1,015.32  
  
## What is the bond's yield-to-maturity (expressed with semiannual compounding)?  
   
# Let's assing as a data frame  
(df <- data.frame(nper=4, pmt=c(-20, -20, -20, -1020), pv=c(19.8, 19.55, 19.21, 956.77)))

## nper pmt pv  
## 1 4 -20 19.80  
## 2 4 -20 19.55  
## 3 4 -20 19.21  
## 4 4 -1020 956.77

# Calculate the YTM by using optirum package  
(bond.YTM <- sum(RATE(nper = df$nper, pmt = df$pmt, pv = df$pv, fv = 0))) %>% round(2)

## [1] 3.86

# Answer: the bond's yield-to-maturity = 3.86%, which is differnt than the original answer!  
  
  
# Alternative package "jrvFinance": NEED to work on it!  
  
# https://cran.r-project.org/web/packages/jrvFinance/jrvFinance.pdf  
# bond.duration(settle, mature, coupon, freq = 2, yield,   
# convention = c("30/360", "ACT/ACT", "ACT/360", "30/360E"), modified = FALSE,   
# comp.freq = freq, redemption\_value = 100)  
# example NPV from jrvFinance package  
# npv(cf = df$pmt, rate = zero.rates, cf.freq = 2, comp.freq = 2)  
  
  
  
## Problem 3  
## Suppose that risk-free zero interest rates with semiannual compounding are as follows:  
  
(maturity <- c(.5, 1, 1.5, 2)) # years

## [1] 0.5 1.0 1.5 2.0

(rates.semi.ann.comp <- c(0.04, 0.045, 0.0475, 0.05))

## [1] 0.0400 0.0450 0.0475 0.0500

# Let's find the continuous compounding Rates   
(rates.cont.comp <- (2 \*log(1+(rates.semi.ann.comp/2))) %>% round(4))

## [1] 0.0396 0.0445 0.0469 0.0494

# Answer: continuous compounding Rates = 3.96%, 4.45% 4.69%, 4.94%  
  
# Let's Calculate the continuously compounded forward rates for the periods: 6 months to 12 months, 12 months to 18 months,   
# and 18 months to 24 months.  
(forward.rates.cont.comp <- c("", # No calculation for the first row (0 to 6 months)  
 (rates.cont.comp[2] \* maturity[2] - rates.cont.comp[1] \* maturity[1])/(maturity[2] - maturity[1]),  
 ((rates.cont.comp[3] \* maturity[3] - rates.cont.comp[2] \* maturity[2])/(maturity[3] - maturity[2])),  
 ((rates.cont.comp[4] \* maturity[4] - rates.cont.comp[3] \* maturity[3])/(maturity[4] - maturity[3]))))

## [1] "" "0.0494" "0.0517" "0.0569"

# convert forward.rates.cont.comp's structure factor to numeric   
forward.rates.cont.comp <- as.numeric(forward.rates.cont.comp)   
# Let's put the all data in a dataframe  
analysis2.df <- data.frame(maturity, rates.semi.ann.comp, rates.cont.comp, forward.rates.cont.comp)  
  
  
# Let's take a look at the table  
kable(analysis2.df, align = "c")

|  |  |  |  |
| --- | --- | --- | --- |
| maturity | rates.semi.ann.comp | rates.cont.comp | forward.rates.cont.comp |
| 0.5 | 0.0400 | 0.0396 | NA |
| 1.0 | 0.0450 | 0.0445 | 0.0494 |
| 1.5 | 0.0475 | 0.0469 | 0.0517 |
| 2.0 | 0.0500 | 0.0494 | 0.0569 |

# Answer: Continuously compounded forward rates for the periods:  
# 6 months to 12 months = 4.94%  
# 12 months to 18 months = 5.17%  
# 18 months to 24 months = 5.69%   
  
## Calculate the semiannually compounded forward rate for the six-month period beginning in 18 months.  
((2 \* (exp(forward.rates.cont.comp[4]/2) - 1)) \* 100) %>% round(2)

## [1] 5.77

# Answer: The semiannually compounded forward rate for the six-month period beginning in 18 months = 5.77%  
  
## What is the value today of an FRA where the holder pays LIBOR and   
## receives 7% (semiannually compounded) for the six-month period beginning in 18 months?  
  
(rate.of.interest.agreed.to.in.FRA <- 0.07) # 7%

## [1] 0.07

(forward.LIBOR.interest.rate.for.period <- 0.0575)# 5.75%

## [1] 0.0575

(principal.2 <- 10000000) # $10,000,000

## [1] 1e+07

(principal.2 \* (rate.of.interest.agreed.to.in.FRA - forward.LIBOR.interest.rate.for.period) \* (maturity[4] - maturity[3]) \* exp(-forward.rates.cont.comp[4] \* maturity[4])) %>% round(0)

## [1] 55777

# Value of the FRA today = $55,777  
  
## Problem 4   
## A five-year bond with a yield of 11% (continuously compounded) pays an 8% coupon at the end of each year.   
  
(bond.YTM <- 0.11) # 11%

## [1] 0.11

(time.to.maturity.of.contract <- 5) # years

## [1] 5

(annual.coupon.rate <- 0.08) # 8%

## [1] 0.08

(face.value.of.the.bond <- 1000) # $1,000

## [1] 1000

(time <- c(1:5))

## [1] 1 2 3 4 5

(cash.flows <- c(rep(80, 4), (face.value.of.the.bond+80)))

## [1] 80 80 80 80 1080

# Calculate the discount factor for each year, the present value of each cash flow,  
# the corresponding weight, and the Time x Weight.   
(discount.factor <- (exp(-bond.YTM \* time)) %>% round(3))

## [1] 0.896 0.803 0.719 0.644 0.577

(PV.of.cash.flows <- cash.flows \* discount.factor)

## [1] 71.68 64.24 57.52 51.52 623.16

(weight <- (PV.of.cash.flows/(sum(PV.of.cash.flows))) %>% round(3))

## [1] 0.083 0.074 0.066 0.059 0.718

(timeXweight <- time \* weight)

## [1] 0.083 0.148 0.198 0.236 3.590

# Let's create and calculate to totals' ROW  
  
(total <- c("","", "", sum(PV.of.cash.flows), sum(weight), sum(timeXweight)))

## [1] "" "" "" "868.12" "1" "4.255"

# create a table   
  
(analysis3 <- cbind(time, cash.flows, discount.factor, PV.of.cash.flows, weight, timeXweight))

## time cash.flows discount.factor PV.of.cash.flows weight timeXweight  
## [1,] 1 80 0.896 71.68 0.083 0.083  
## [2,] 2 80 0.803 64.24 0.074 0.148  
## [3,] 3 80 0.719 57.52 0.066 0.198  
## [4,] 4 80 0.644 51.52 0.059 0.236  
## [5,] 5 1080 0.577 623.16 0.718 3.590

analysis3 <- rbind(analysis3, total)  
kable(analysis3, align = "c")

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | time | cash.flows | discount.factor | PV.of.cash.flows | weight | timeXweight |
|  | 1 | 80 | 0.896 | 71.68 | 0.083 | 0.083 |
|  | 2 | 80 | 0.803 | 64.24 | 0.074 | 0.148 |
|  | 3 | 80 | 0.719 | 57.52 | 0.066 | 0.198 |
|  | 4 | 80 | 0.644 | 51.52 | 0.059 | 0.236 |
|  | 5 | 1080 | 0.577 | 623.16 | 0.718 | 3.59 |
| total |  |  |  | 868.12 | 1 | 4.255 |

# str(analysis3)  
## What is the bond's price?  
(bond.price <- total[4])

## [1] "868.12"

bond.price <- as.numeric(bond.price)  
# total[4] same result: bond price  
  
# Answer: Bond price = $868.12  
  
  
## What is the bond's duration?  
(duration <- total[6])

## [1] "4.255"

duration <- as.numeric(duration)  
  
# Answer: Bond duration = 4.255 years  
  
  
## Assume that interest rates in the market change,   
## and the bond's yield decreases by 0.2%.  
(change.in.bond.yield <- -0.002) # %

## [1] -0.002

change.in.bond.yield <- as.numeric(change.in.bond.yield)  
## What is the effect on the bond's price of the decrease in its yield   
## if you use the duration approximation for the change in price?   
  
(effect.onprice <- (-bond.price \* duration \* change.in.bond.yield) %>% round(2))

## [1] 7.39

effect.onprice <- as.numeric(effect.onprice)  
  
# Answer: the effect on the bond's price = $7.39  
  
  
# put values into a data frame   
df.change <- data.frame(bond.price, duration, change.in.bond.yield, effect.onprice)  
  
# Let's take a look at the change of the bond price  
kable(df.change, align = 'c')

|  |  |  |  |
| --- | --- | --- | --- |
| bond.price | duration | change.in.bond.yield | effect.onprice |
| 868.12 | 4.255 | -0.002 | 7.39 |