Bond Valuation Using R–Further Exploration of Vectors and Functions

Gracie Yao

October 31, 2017

```
library(ggplot2)
library(Quandl)
```

Introduction

We all know how to create r objects such as data frame and functions, but how can r help us solve real-world problems? This post focuses on analyzing and evaluating bonds using r. Bonds, which represent debt obligations, have become a popular investment option. Therefore, it is helpful to understand the value of bonds when we are making investment decisions. The motivation for this post is to familiarize the audience with bonds, to present the application of basic r commands, as well as to explore several useful functions and packages.
The post starts with the introduction of numerous financial concepts necessary to understand the models built in the following sections (part I and II). Then, it explores four main topics: bond's cash flow, bond valuation, price-yield relationship, and the estimation of a bond's yield. The post mainly involves manipulating vectors and data frame, creating functions, and making plots using ggplot2.

I. Dates in R

When analyzing a financial portfolio, it is inevitable for us to deal with different formats of dates.

- Two main classes of this type of data:
  - Type 1: Date
    - Used for calendar date object
    - "2017-10-28"
  - Type 2: POSIX-Portable Operating System Interface
    - Date + time + timezone
    - "2017-10-28 10:00:21 EDT"

```
# Current Date
today<-Sys.Date()
today
```

```
## [1] "2017-10-31"
```

```
class(today)
```

```
## [1] "Date"
```

```
#Current date and time
today2<-Sys.time()
today2
```

```
## [1] "2017-10-31 08:54:45 PDT"
```

```
class(today2)
```

```
## [1] "POSIXct" "POSIXt"
```

- Create date from character strings

```
dates <- as.Date(c("1929-11-29","2017-10-28"))
dates
```

```
## [1] "1929-11-29" "2017-10-28"
```

```
class(dates)
```

```
## [1] "Date"
```

- Date Formats
  Usually, we won't have dates in "YYYY-MM-DD" format. In order to use `as.Date` to convert this type of character, we need to specify the format using the `format` argument.

```
as.Date("09/28/2008", format = "%m / %d / %Y")
```

```
## [1] "2008-09-28"
```

Here are a list of formats we can use to specify:

```
%Y: 4-digit year (0-9999)
%y: 2-digit year (00-99)
%m: 2-digit month (01-12)
%d: 2-digit day of the month (01-31)
%A: weekday (Wednesday)
%a: abbreviated weekday (Wed)
%B: month (January)
%b: abbreviated month (Jan)
"/" "-" ",": common separators
```

```
# convert "10,28,17"
as.Date("10,28,2017", format = "%m,%d,%Y")
```

```
## [1] "2017-10-28"
```

```
# convert "Oct 28, 2017"
as.Date("Oct 28, 2017", format = "%b %d,%Y")
```

```
## [1] "2017-10-28"
```

```
# convert "28oct2017"
as.Date("28oct2017", format = "%d%b%Y")
```

```
## [1] "2017-10-28"
```

We can also use `format()` to change between any two formats.

```
# change "1jan17" to "Jan 01, 17"
cha_date <- "1jan17"
date <- as.Date(cha_date,format="%d%b%y")
format(date, format="%b %d, %y")
```

```
## [1] "Jan 01, 17"
```

- Find the difference between two days

```
# Dates
dates <- as.Date("2017-04-01")

# Create the origin
origin <- as.Date("2017-01-05")

# Find the difference between dates and origin
dates - origin
```

```
## Time difference of 86 days
```

II. A couple of financial concepts related to bonds

- *Time Value of Money (TVM)*: the idea that money available today is more valuable than the same amount tomorrow because you can invest the money you have today so that you gain more money in the future.
- *Future value*: the value of a current asset at a specified point in the future. Since we would prefer receiving money at the present, we need some interest rate r that would make us agree to get the money in the future.
  Therefore, fv (the future value) is calculated as follows, with r the interest rate and pv the present value:
  - The future value one year from now: `fv1 <- pv * (1+r)`
  - The future value five years from now: `fv5 <- pv * (1+r)^5`
  - The future value n years from now: `fvn <- pv * (1+r)^n`
- *Present value*: the current value of a future asset. Since we would like to receive money today, we will be willing to receive less money today to avoid having to waiting for a period of time to receive the same amount of money in the future.
  pv (the present value) is calculated as follows:
  - `Pv <- fv1 / (1+r)`
  - `Pv <- fv5 / (1+r)^5`
  - `Pv <- fvn / (1+r)^n`

example: calculate the present value of fv2 and fv5 assuming an interest rate of 0.15. fv1 is a future value one year from now and fv2 is a future value two years from now.

```
# Calculate pv2
pv2 <- 100 / (1 + 0.15)^2
# Calculate pv5
pv5 <- 100 / (1 + 0.15)^5
# Print pv2 and pv5
pv2
```

```
## [1] 75.61437
```

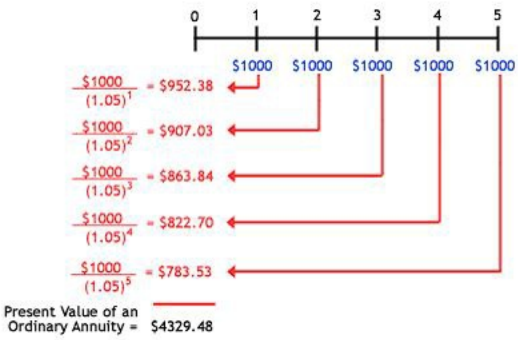
```
pv5
```

```
## [1] 49.71767
```

- *Bond issuer*: the entity that borrows the money, which can include corporations, governments, and municipalities.
- *Principal*: the amount of money borrowed; also called **par value** or **face value** of the bond.
- *Coupon rate*: expressed as a percentage of the face value, coupon rate is the rate of interest paid by bond issuers. Coupon rate can be paid at different frequencies, such as annually, semi-annually, or quarterly. It can be fixed for the life of the bond or it can change at each payment period.
- *Maturity Date*: the date when the principal is paid back.
  In the following discussion, we use bond with fixed annual coupons and fixed maturity date.
- *Value of an asset*: present value of expected future cash flows.
- *Discounted cash flow formula*, where:
  - CF: cash flow in the period
  - r: the discount rate; also called "yield" for bonds
  - n: the period number

$$DCF = (CF/(1+r)^1) + (CF/(1+r)^2) + (CF/(1+r)^3) + \ldots + (CF/(1+r)^n)$$

The following image shows the discounted cash flow for an annuity, which demonstrates this formula:



III. Bond's Cash flow

Before the maturity date, the investor receives coupon payments; at maturity, the investor receives the last coupon payment and the principal.

Suppose we want to invest in a bond with a $200 par value, 4% coupon rate which is the interest that we get by holding the bond, and 5 years to maturity. In this case, prior to the maturity date, we can get $8 coupon payments each year. At maturity, we will also receive the $200 par value back. The bond has a yield of 6%, which acts as the discount rate.

```
# step 1: create a cash flow vector and then convert it to data frame so that we can add additional columns
cf <- c(8, 8, 8, 8, 208)
cf <- data.frame(cf)
# step 2: add a time index--each cash flow corresponds to a certain period of time (in this case, the unit is year)
cf$t <- as.numeric(rownames(cf))
# step 3: add a present value factor used for discounting
cf$pv_fac <- 1/(1+0.06)^cf$t
# step 4: calculate each cash flow's present value by multiplying cash flow by the corresponding pv_factor
cf$pv <- cf$cf * cf$pv_fac
# step 5: calculate the bond's value by summing up the present value of cash flows
sum(cf$pv)
```

```
## [1] 183.1505
```

IV. Bond Valuation Function

In order to make the calculation more efficient, creating a bond valuation function is helpful.
Parameters used:

- par for principal / par value
- r for coupon rate
- ttm for time to maturity
- y for yield

step 1: Create cash flow vector: use rep(x, y) function to repeat x (coupon payment) y (time to maturity - 1) times, which presents the cash flow before the maturity date; then, add the cash (par value + final coupon) we receive in the last year; finally, we convert the vector into a data frame.
`cf <- c(rep(par*r, ttm - 1), par*(1 + r))`
`cf <- data.frame(cf)`
step 2: Add time index (rownames of cf is 1,2,3…until the "ttm" of the bond)
`cf$t <- as.numeric(rownames(cf))`
step 3: Create PV factor
`cf$pv_factor <- 1 / (1 + y)^cf$t`
step 4: Calculate present value of each cash flow
`cf$pv <- cf$cf * cf$pv_factor`

step 5: Get the bond's value by summing up PVs
`sum(cf$pv)`

```
# taking all these together, we write the bond_value() function:
bond_value <- function(par,r,ttm, y){
 cf <- c(rep(par * r, ttm - 1), par * (1 + r))
 cf <- data.frame(cf)
 cf$t <- as.numeric(rownames(cf))
 cf$pv_factor <- 1 / (1 + y)^cf$t
 cf$pv <- cf$cf * cf$pv_factor
 sum(cf$pv)
}
bond_value(p=200, r=0.04, ttm=5, y=0.06)
```

```
## [1] 183.1505
```

V. Price-Yield Relationship

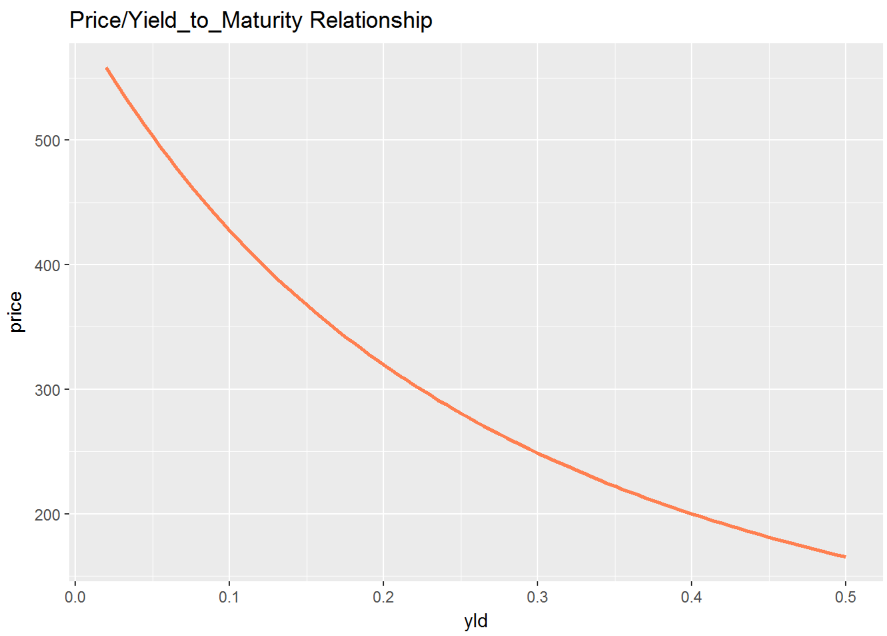
- *Price-Yield Relationship*
  In order to explore the relationship between bond price and yield, we can first make a plot using the `bond_value` function we just created.

```
bond_value <- function(par, r, ttm, y) {
  cf <- c(rep(par * r, ttm - 1), par * (1 + r))
  cf <- data.frame(cf)
  cf$t <- as.numeric(rownames(cf))
  cf$pv_fac <- 1 / (1 + y)^cf$t
  cf$pv <- cf$cf * cf$pv_fac
  sum(cf$pv)
}
# Generate bond yield vector
yld <- seq(0.02, 0.5, 0.005)

# Convert yld to data frame
yld <- data.frame(yld)

# Calculate bond price given different yields using for loop
for (i in 1:nrow(yld)) {
     yld$price[i] <- bond_value(200, 0.40, 5, yld$yld[i])
}

# Plot P/YTM relationship
ggplot(yld,aes(x=yld,y=price))+
  geom_line(col="coral",size=1)+
  ggtitle("Price/Yield_to_Maturity Relationship")
```



The plot shows the non-linear inverse relationship between bond's price and yield. For bonds, yield acts as a discount rate. When the yield goes up, an investor would gain profits only if the bond price is low. Otherwise, the investor would choose to make other kinds of investment because the bond offers below market returns.

- *Credit rating*: Bond yields are associated with the risk of bonds, which can be determined by looking at the bond's credit rating. Credit rating is made by firms called rating agencies; The three most prominent ones are S&P, Fitch, and Moody's. The probability of default increases as the credit rating goes from A to C.
  The tables below show the rating system by the three agencies:

Investment Grade Bonds (low default risk)

```
SP<-c("AAA","AA","A","BBB")
Fitch<-c("AAA","AA","A","BBB")
Moodys<-c("Aaa","Aa","A","Baa")
data.frame(SP,Fitch,Moodys)
```

```
##    SP Fitch Moodys
## 1 AAA   AAA    Aaa
## 2  AA    AA     Aa
## 3   A     A      A
## 4 BBB   BBB    Baa
```

High Yield Bonds (high default risk)

```
SP<-c("BB","B","CCC-Lower")
Fitch<-c("BB","B","CCC-Lower")
Moodys<-c("Ba","B","Caa-Lower")
data.frame(SP,Fitch,Moodys)
```

```
##          SP     Fitch    Moodys
## 1        BB        BB        Ba
## 2         B         B         B
## 3 CCC-Lower CCC-Lower Caa-Lower
```

The role of credit rating in bond valuation is essential. For instance, bonds with a BB rating can be seen as having a similar risk profile. Therefore, in order to identify the unknown yield of a particular bond, we could use the yield of bonds with the same credit rating.

- `Quandl` package: collects the financial and economic data.

```
#obtain Moody's Baa index data
baa <- Quandl("FED/RIMLPBAAR_N_M")
# Find 9/30/16 yield
baa_yield <- subset(baa, baa$Date == "2016-09-30")
# Convert yield to decimals
baa_yield <- baa_yield$Value / 100
baa_yield
```

```
## [1] 0.0431
```

Once we obtain the yield, we can use it to value the bond.

```
bond_value(par=200,r=0.04,ttm=5,y=0.0431)
```

```
## [1] 197.2637
```

VI. The Estimation of a Bond's Yield

The basic idea to estimate a bond's yield: if the price of the bond and its cash flow is given, the yield we get should equate the bond price with the present value of the bond's cash flows.
Example: suppose someone wants to pay $95 to purchase a bond you own with a $100 par value, 5% coupon rate, and 5 years to maturity. You want to know the yield of this bond.

Trial-and-error method

```
# Bond value using 5% yield
bond_value(p=100,r=0.05,ttm=5,y=0.05)
```

```
## [1] 100
```

```
#a bond value of $100 which is higher than $95 means that the yield is lower than the real value, so try 7%
bond_value(p=100,r=0.05,ttm=5,y=0.07)
```

```
## [1] 91.79961
```

```
#a bond value of $91 which is lower than $95 means that the yield is higher than the real value, so try 6%
bond_value(p=100,r=0.05,ttm=5,y=0.06)
```

```
## [1] 95.78764
```

```
#the bond value we get is pretty close to $95, so the yield is approximately 6%
```

Use the function `uniroot` to automating the guessing process
`uniroot()` takes a modified cash flow vector cf and a modified bond valuation function (bond_val) as inputs and uses them to determine the yield. Since the yield is a number between 0.00 to 1.00, we use C(0,1) to limit the interval for the search.

```r
# the cash flow vector used here includes the bond price as the first element. The price is negative because it's a cash outflow.
cf <- c(-95, 5, 5, 5, 5, 105)

# Create bond valuation function bond_val using modified cf
bond_val <- function(i, cf, t=seq(along = cf)) {
     sum(cf / (1 + i)^t)
  }

# Create ytm() function using uniroot; uniroot function searches the interval from 0 to 1 for a root which is the yield to maturity we are looking for
ytm <- function(cf) {
    uniroot(bond_val, c(0, 1), cf = cf)$root
}

# Use ytm() function to find yield
ytm(cf)
```

```
## [1] 0.06193422
```

Conclusion

In the process of analyzing bonds and developing financial models, we explored functions and packages that are commonly used in finance and got more familiar with data manipulation. The major take-home message is that the power of r is not limited to general data processing; rather, we are able to use basic commands to convert the knowledge we know about a specific field, in this case finance, into models that would make our work more efficient.

Reference

https://www.statmethods.net/input/dates.html
http://www.investopedia.com/terms/t/timevalueofmoney.asp
http://www.investopedia.com/video/play/coupon-rate/
https://cran.r-project.org/web/packages/Quandl/Quandl.pdf
https://s3.amazonaws.com/assets.datacamp.com/production/course_1698/slides/Ch2.pdf
https://www.thebalance.com/why-do-bond-prices-and-yields-move-in-opposite-directions-417082
https://www.youtube.com/watch?v=CYXAzTwZr30
http://news.morningstar.com/classroom2/course.asp?docId=5375&page=3
http://www.stat.missouri.edu/~dsun/8100/Fig3.1.R
http://www.getobjects.com/Components/Finance/TVM/concepts.html
https://en.wikipedia.org/wiki/Bond_(finance)
http://i.investopedia.com/inv/articles/site/Corpfin27.jpg