

Introducción a bonos


Aplicaciones con R


Gabriel E. Cabrera


Universidad de Chile
Facultad de Economía y Negocios


12 de Mayo del 2019



 gcabrerag@fen.uchile.cl

 gcabrerag.rbind.io

 GaboC_g

 GaboCg

 Facultad de Economía & Negocios, Universidad de Chile

Tabla de contenido



- 1 Precio de un Bono
- 2 Otra cosa es con funciones
- 3 Relación precio del Bono y Yield
- 4 Graficar nuestros datos
- 5 Trabajando con yields reales
- 6 Duración y Convexidad de un Bono
- 7 Efecto Dolar
- 8 Ejercicio



✉ gcabrerag@fen.uchile.cl

🔗 gcabrerag.rbind.io

🐦 @GaboC_g

👤 @GaboCg

📍 Facultad de Economía & Negocios, Universidad de Chile

Precio de un Bono



Como se vio en clases el precio de un bono se calcula como:

$$P_B = \sum_{t=1}^T \frac{C}{(1+r)^t} + \frac{ValorNominal}{(1+r)^t}$$

Donde:

- P_B : Precio del Bono
- C_t : Pago interéses o cupones
- T : Números de períodos o madurez
- r : Tasa de descuento o yield-to-maturity semi-anual



Considere el siguiente ejercicio: Calcular el precio de un bono con pago de cupón semestral, Madurez 25 años, Tasa cupón 6.5%, Yield semi-anual de 6.9% y Valor nominal de 100. Para desarrollar el ejercicio, vamos a construir por parte los componentes de nuestro bono.

1. Construimos tanto la tasa cupón como la *Yield semi-anual*.

```
1 tc <- 0.065  
2 y  <- 0.069
```

2. Construimos un vector con los valores de los cupones más el principal

```
1 pago <- c(rep(tc*100/2, 49), (100 + tc*100/2))
```

tanto `tc`, `r` y `pago` se encontraran en *Values* del *global environment*, en tipo `numeric`.

3. Para poder trabajar con nuestra base de datos, transformamos nuestro vector `pago` que está en forma `numeric` a `data frame`.

```
1 pago <- as.data.frame(pago)
```

Ahora existe un objeto con estructuración de datos `data frame` en nuestro global environment. Se podría haber llamado de cualquier forma a nuestro nuevo objeto `pago`.



Ya construido nuestro objeto pago veremos que en **R** existen muchas formas de hacer lo mismo:

1. Al principio de la sesión cargamos la libreria `tidyverse`, está nos permitirá trabajar con un “megapaquete” que incluye otros paquetes en su interior (`ggplot2`, `dplyr`, `magittr`, entre otros). Todos los paquetes que conforman “el Tidyverse” comparten la misma visión sobre el trabajo con datos y la escritura de código. Si va a la pestaña `packages` y escribe `dplyr` verá que está activa, pero nunca la “llamamos”, esto se debe a `tidyverse` lo hizo por nosotros.

```
1 pago1 <- pago %>%
2   mutate(t1 = as.numeric(index(pago)), factor_desc = 1/(1+y/2)^(t1),
3          val_present = pago*factor_desc) %>%
4   summarise(sum(val_present))
5 pago1
```

```
## sum(val_present)
## 1 95.26627
```

2. La otra forma es:

```
1 # replicamos el objeto
2 pago2 <- pago
3
4 pago2$t2 <- as.numeric(rownames(pago2))
5
6 # Calculamos el factor de descuento
7 pago2$factor_desc <- 1 / (1 + y/2)^(pago2$t2)
8 # Calculamos el valor presente
9 pago2$val_present <- pago2$factor_desc*pago2$pago
10 # Calculamos el precio
11 sum(pago2$val_present)
```

```
## [1] 95.26627
```

Otra cosa es con funciones

Otra cosa es con funciones



Ahora se contruirá una función para valorizar cualquier bono que pague cupones iguales:

```
1 # p: valor nominal; tc: tasa cupón; t: madurez; y: yield to maturity
2 precio.bono <- function(p,tc,t,y){
3   # rep returns a vector with value = p * r and times = ttm -1
4   pago    <- c(rep(tc*p, t - 1),p*(1 + tc))
5   pago    <- as.data.frame(pago)
6   pago$t  <- as.numeric(rownames(pago))
7   pago$factor_desc <- 1 / (1 + y)^(pago$t)
8   pago$valor_prese <- pago$factor_desc*pago$pago
9   sum(pago$valor_prese)
10 }
```

```
1 precio.bono(100,0.065/2,50,0.069/2)
```

```
## [1] 95.26627
```

Relación precio del Bono y Yield



Ahora utilizando la función `precio.bono` valorizaremos un bono con las siguientes características:

- Principal : 100
- Tasa Cupón: 5%
- Madurez: 10 años
- Yield: 4.29%

```
1 # Valoramos el siguiente Bono
2 precio.bono(p = 100, tc = 0.05, t = 10, y = 0.0429)
```

```
## [1] 105.6765
```



Se contruirá una secuencia de yields:

```
1 # Cosntruimos yields
2 yields <- seq(0.02, 0.4, 0.01)
```

La función seq generará una secuencia. En este caso parte del 0.02 hasta el 0.4 pero con intervalos de 0.01.

```
1 # Convertimos yields a data frame como antes
2 yields <- as.data.frame(yields)
```



Explicación en clases.

```
1 # Calaculamos el precio del bono para distintas yields
2 for (i in 1:nrow(yields)) {
3   yields$precio[i] <- precio.bono(100, 0.10, 20, yields$yields[i])
4 }
```


Graficar nuestros datos

Graficar nuestros datos: Con ggplot2

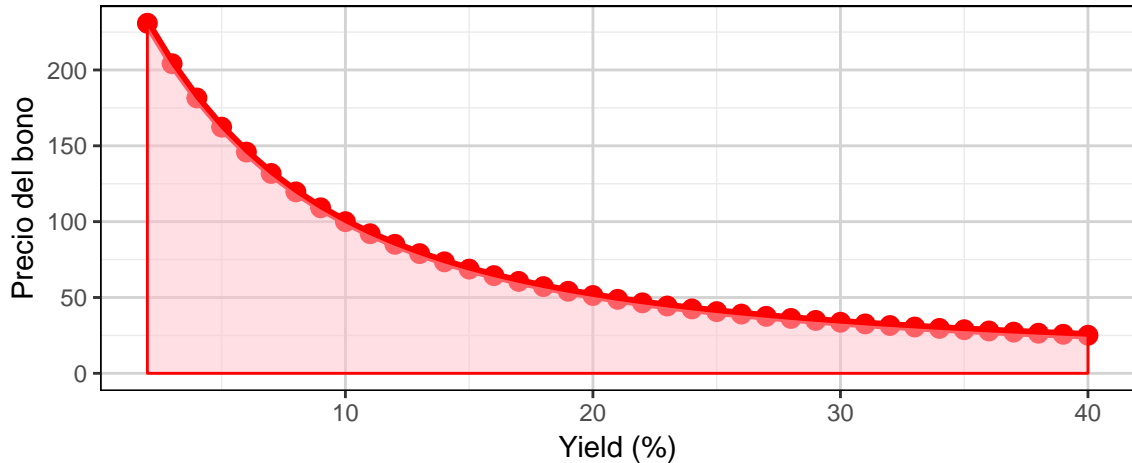


Una manera de visualizar datos es usar ggplot2, se recomienda que añadan por parte lo que desean en su gráfico.

```
1 # Graficamos con ggplot2
2 g1 <- ggplot(data = yields, aes(x = yields*100, y = precio)) + geom_line(size = 1.5,
3                               color = "red") +
4 g1 <- g1 + geom_point(size = 3, color = "red")
5 g1 <- g1 + ggtitle("Relación inversa:", subtitle = "Precio del Bono vs Yield")
6 g1 <- g1 + xlab("Yield (%)") + ylab("Precio del bono")
7 g1 <- g1 + geom_ribbon(aes(ymin = 0, ymax = pmax(precio, 0)), fill="pink",
8                       col="red", alpha=0.5)
9 g1 <- g1 + theme_bw()
10 g1 <- g1 + theme(panel.border = element_rect(colour = "black", fill = NA, size = .5),
11                 panel.grid.major = element_line(colour = "#d3d3d3"))
12 g1
```

```
1 # Guardamos gráfico
2 ggsave("retorno-yield.png", width = 8.5, height = 4.5, dpi = 300)
```

Relación inversa: Precio del Bono vs Yield

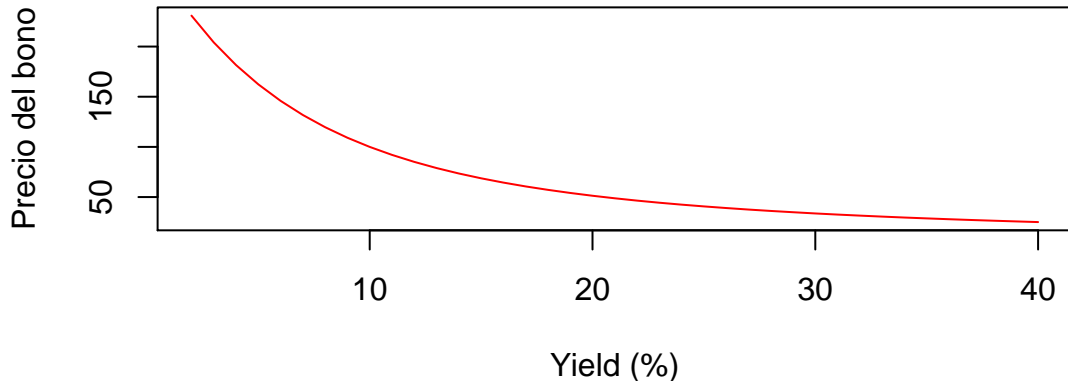




otra manera de visualizar datos es usar plot, esta opción es valida pero es más “arcaica” y más limitada que ggplot2.

```
1 # Con plot
2 g2 <- plot(yields$yields*100,yields$precio,type = "l",col = "red",
3           main = "Relación inversa: Precio del Bono vs Yield",
4           xlab="Yield (%)", ylab="Precio del bono")
```

Relación inversa: Precio del Bono vs Yield



Trabajando con yields reales



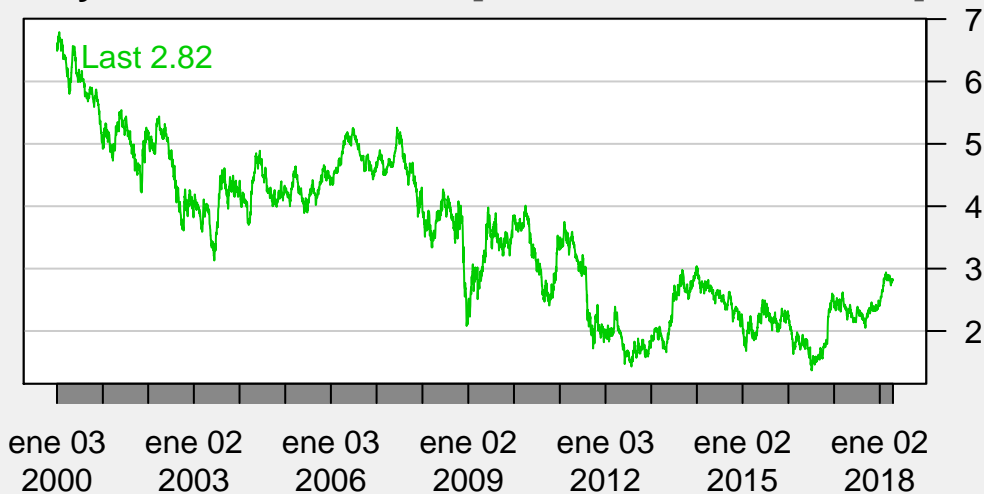
quantmod es uno de las librerías más ocupadas en R para extraer datos financieros, te permite graficar, realizar análisis técnico, calcular retornos ($\text{Delt}(x)$), etc. Aunque las series son descargadas con estructura xts, la podemos transformar a data frame. A continuación descargaremos la yield de los bonos del tesoro de Estados Unidos a 10 años:

```
1 t10yr <- getSymbols(Symbols = "DGS10", src = "FRED", auto.assign = FALSE)
2
3 t10yr <- subset(t10yr["2000-01-01/2018-04-17"])
```

```
1 # Grafico con chartSeries de quantmod solo funciona con xts
2 chartSeries(t10yr)
```

t10yr

[2000-01-03/2018-04-17]




```

1 t10yr.df <- as.data.frame(t10yr)
2
3 t10yr.df <- t10yr.df %>%
4     mutate(fecha = as.Date(rownames(t10yr.df))) %>%
5     na.omit()
6
7 g3 <- ggplot(data = t10yr.df, aes(x = fecha , y = DGS10)) + geom_line(size = 1, color =
  ↳ "green")
8 g3 <- g3 + ggtitle("10-Year US Treasury Yields", subtitle = "Desde 2000-01-01 hasta
  ↳ 2018-04-17")
9 g3 <- g3 + ylab("Fecha") + xlab("Yield(%)")
10 g3 <- g3 + theme_bw() + theme(panel.border = element_rect(colour = "black", fill = NA,
  ↳ size = .5),
11
12                               panel.grid.major = element_line(colour = "#d3d3d3"))

```

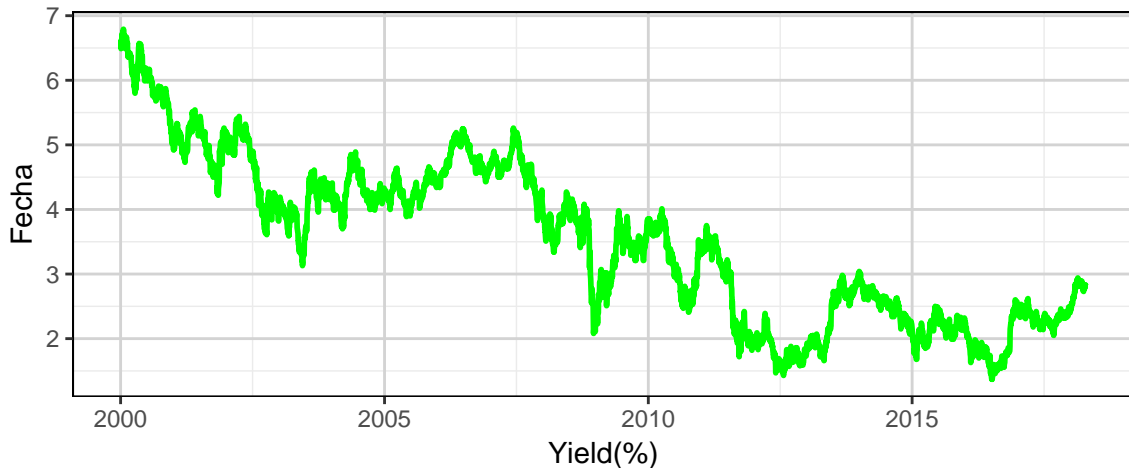
```

1 # Guarmados gráfico
2 ggsave("treasury-yields.png", width = 8.5, height = 4.5, dpi = 300)

```

10-Year US Treasury Yields

Desde 2000-01-01 hasta 2018-04-17



Duración y Convexidad de un Bono



La función `subset` se explica sola, permite extraer una parte de tu base según un criterio.

```
1 # Extraemos un valor en específico
2 t10yr_yield <- t10yr.df %>%
3   subset(fecha == "2017-03-03")
4
5 t10yr_yield <- as.numeric(t10yr_yield$DGS10)*0.01
```



Existen dos Duraciones, la de Macaulay y modificada (o de Hicks), las que miden sensibilidad del precio ante cambios de la yield.

Macaulay:

$$\text{Duración de Macaulay} = \left[\frac{1+y}{y} - \frac{1+y + [n \cdot (c-y)]}{[c \cdot ((1+y)^n - 1)] + y} \right]$$

Modificada:

$$\text{Duración Modificada} = \text{Duración de Macaulay} / (1+y)$$

Aprox. Duración Modificada:

$$\text{Aprox. Dur. Mod.} = \frac{MV_- - MV_+}{2 \cdot \Delta y \cdot MV_0}$$



Duración Macaulay:

```
1 # duracion de Macaulay
2 macaulay <- function(y,n,c,t,T){
3     mac <- (1 + y)/y - (1+y+(n*(c-y)))/(c*((1+y)^n - 1) + y)
4     print(mac)
5 }
```

```
1 macaulay <- macaulay(tl0yr_yield,10,0.03)
```

```
## [1] 8.817423
```



Duración Modificada:

```
1 # duración modificada
2 modificada <- macaulay/(1+t10yr_yield)
3 modificada
```

```
## [1] 8.603204
```



Aproximación Duración Modificada:

```
1 # Para la aproximación de la duración modificada
2 precio.arriba <- precio.bono(p = 100, tc = 0.03, t = 10, y = t10yr_yield + 0.01)
3 precio        <- precio.bono(p = 100, tc = 0.03, t = 10, y = t10yr_yield)
4 precio.abajo  <- precio.bono(p = 100, tc = 0.03, t = 10, y = t10yr_yield - 0.01)
```

```
1 # Calculo de aproximación duración modificada
2 aprox.dur.mod <- (precio.abajo - precio.arriba) / (2 * precio * 0.01)
3 aprox.dur.mod
```

```
## [1] 8.620075
```


Duración y Convexidad de un Bono: Duración con librería



Una librería útil para trabajar con bono es `derivmkt`s. Si no lo tenemos instalado:

```
1 install.packages("derivmkt")  
2 library("derivmkt")
```

```
1 # Duración moficada  
2 duration(precio, 3, 10, 100, 1, modified = TRUE)
```

```
## [1] 8.603204
```

```
1 # Duración Macaulay  
2 duration(precio, 3, 10, 100, 1, modified = FALSE)
```

```
## [1] 8.817423
```



Cuando las tasa de interés (yield) varían en demasiados puntos base, deja de ser la duración (cualquier tipo) una buena medida de sensibilidad y se recurre a la convexidad.

Convexidad:

$$\frac{1}{P \times (1 + y)^2} \sum_{t=1}^T \left[\frac{CF_t}{(1 + y)^t} (t^2 + t) \right]$$

Aprox. Convexidad:

$$\frac{MV_- + MV_+ - 2 * MV_0}{MV_0 * \Delta y^2}$$

Convexidad:

```
1 convexidad <- function(p,tc,t,y,precio){
2   # rep returns a vector with value = p * r and times = ttm -1
3   pago    <- c(rep(tc*p, t - 1),p*(1 + tc))
4   pago    <- as.data.frame(pago)
5   pago$t  <- as.numeric(rownames(pago))
6   pago$factor_desc <- 1 / (1 + y)^(pago$t)
7   pago$valor_prese <- pago$factor_desc*pago$pago*((pago$t)^2 + pago$t)
8   print(sum(pago$valor_prese)*(1/(precio*(1+y)^2)))
9 }
10
11 convexidad(100, 0.03, 10, t10yr_yield, precio.bono(p = 100, tc = 0.03, t = 10,
12                                                    y = t10yr_yield))
```

```
## [1] 88.34131
```

Aproximación Convexidad:

```
1 # Calculamos medida de convexidad
2 convexidad <- (precio.arriba + precio.abajo - 2 * precio) / (precio * (0.01)^2)
3 convexidad
```

```
## [1] 88.44685
```

Convexidad con librería:

```
1 convexity(precio, 3, 10, 100, 1)
```

```
## [1] 88.34131
```

Efecto Dolar



Efecto Dolar:

$$\Delta P_{duration} + \Delta P_{convexity}$$

Duration Dollar Change:

$$-D \times \Delta y \times P$$

Convexity Dollar Change:

$$0.5 \times C \times (\Delta y)^2 \times P$$

Ejercicio

Ejercicio: Bonos, Convexidad y Duración



La empresa Finanzas I S.A. emitió deuda en forma de bono bullet, con valor nominal de \$100, tasa cupón 3% y una madurez de 8 año. Este bono tiene clasificación Aaa por Moody (“DAAA”) y fue emitido el 30 de septiembre del 2016. Se determinó que la yield del bono es comparable con la yield de un bono con clasificación Aaa.

- a. Calcule el precio del Bono.
- b. Calcule ambas duraciones y convexidad.
- c. Calcule el efecto dolar de la duración y convexidad en el precio ante un cambio del 1%.
- d. Estime el cambio en el precio y el nuevo precio.

a. Calcule el precio del Bono

```
1 if(!require("pacman")) install.packages("pacman")
2 p_load("tidyverse", "quantmod", "derivmks")
```

```
1 yield <- getSymbols(Symbols = "DAAA", src = "FRED", auto.assign = FALSE)
2 yield <- as.numeric(subset(yield["2016-09-30"])) / 1200
```

Yield
0.0029

Función Precio Bono:

```
1 precio_bono <- function(p,tc,t,y){
2   pago <- c(rep(tc*p, t - 1),p*(1 + tc))
3   pago <- as.data.frame(pago)
4   pago$t <- as.numeric(rownames(pago))
5   pago$factor_desc <- 1 / (1 + y)^(pago$t)
6   pago$valor_prese <- pago$factor_desc*pago$pago
7   sum(pago$valor_prese)
8 }
```

Función Duración:

```
1 # duracion de Macaulay función
2 macaulay <- function(y,n,c,t,T){
3   mac <- (1 + y)/y - (1+y+(n*(c-y)))/(c*((1+y)^n -1) + y)
4 }
```

Función Convexidad:

```
1 convexidad <- function(p,tc,t,y){
2   pago <- c(rep(tc*p, t - 1),p*(1 + tc))
3   pago <- as.data.frame(pago)
4   pago$t <- as.numeric(rownames(pago))
5   pago$factor_desc <- 1 / (1 + y)^(pago$t)
6   pago$valor_prese <- pago$factor_desc*pago$pago
7   pago$valor_prese_t <- (pago$factor_desc*pago$pago)*pago$t*(pago$t + 1)
8   (sum(pago$valor_prese_t)/sum(pago$valor_prese))/(1 + y)^2
9 }
```

```
1 precio <- precio_bono(100, 0.03, 8, yield)
```

Precio

121.43

b. Calcule ambas duraciones y convexidad.

```
1 # Con función
2 duracion_macaulay <- macaulay(yield, 8, 0.03)
3 duracion_macaulay_mod <- duracion_macaulay / (1+yield)
4 convexidad <- convexidad(precio, 0.03, 8, yield)
```

Duración	Duracion Mod.	Convexidad
7.31	7.29	63.41

c. Calcule el efecto dolar de la duración y convexidad en el precio ante un cambio del 1%.

```
1 duracion_pct_cambio <- -duracion_macaulay_mod*0.01
2 duracion_dolar_cambio <- duracion_pct_cambio*precio
3
4 convexidad_pct_cambio <- convexidad*0.5*(0.01)^2
5 convexidad_dolar_cambio <- convexidad_pct_cambio*precio
```

Duración	Convexidad
-8.86	0.39

d. Estime el cambio en el precio y el nuevo precio.

```
1 cambioPrecio <- duracion_dolar_cambio + convexidad_dolar_cambio
```

Cambio Precio

-8.47

```
1 # Estimamos nuevo precio  
2 nuevoPrecio <- cambioPrecio + precio
```

Nuevo Precio

112.96



- Apunte curso Finanzas I: <https://finance-r.netlify.com/>



Introducción a bonos

Aplicaciones con R

Gabriel E. Cabrera

Universidad de Chile
Facultad de Economía y Negocios

12 de Mayo del 2019