Introducción a quantmod

Aplicaciones y usos en R

Gabriel Cabrera G.

Universidad de Chile
Facultad de Economía y Negocios

6 de Abril del 2019

Información de contacto

gcabrerag@fen.uchile.cl
gcabrerag.rbind.io
GaboC_g
GaboCg
Facultad de Economía & Negocios, Universidad de Chile

Tabla de contenido

Introducción a Quantmod

¿Qué es **Quantmod**?

1. Es un paquete diseñado para desarrollar, testear e implementar modelos estadísticos financieros.
2. A través de la función `getsymbols` podemos extraer datos financieros desde varias fuentes: Google Finance, Yahoo Finance, Federal Reserve Bank of St. Louis FRED (más de 11,000 series !!!) y Oanda. Incluso desde fuentes propias: MySQL , R (Rdata) y Comma Separated Value files (csv).
3. Tiene herramientas para realizar análisis técnico.
4. Con `chartSeries` se puede graficar, al más puro estilo de los terminales de Bloomberg y/o Reuters, genial no. No obstante... nunca está demás `ggplot2` o `plotly` (Tufte's Principles) (Wickham et al. 2014).

Manos a la Obra: Obtención de Datos

Comencemos: Preambulo

Como todo paquete se debe instalar:

```
# Instalación package
install.packages("quantmod")
```

y posteriormente lo agregamos a nuestro script:

```
# Cargamos "quantmod"
library("quantmod")
```

HINT

Con ctrl + R en windows/linux y cmd + R en MAC OS agregamos más rapido comentarios (sección) en Rstudio.

Función *getSymbols*

La función se compone principalmente de 5 elementos:

- Nombre del la serie (ticker o nemotecnico).
- Fuente/source (`src`), e.g: `src="google"`, `src="yahoo"`, `src="FRED"`
- Inicio de la serie (`from`), e.g: `as.Date("1990-01-01")`.
- Fin de la serie (`to`), e.g: `as.Date("1990-01-01")`.
- periodicity, esta puede ser daily, monthly o yearly.

```
1 # Estructura de la función getSymbols
2 getSymbols(" ", src = , from = as.Date(" "), to = as.Date(" "), periodicity = )
```

Obtención de Datos

A continuación obtendremos los datos del S&P 500 (Standard & Poor 500), aquí necesitamos saber el ticker o nemotécnico de la acción (stock) que vamos a trabajar, para *Yahoo* es ^GSPC. Si se desea buscar otra acción basta con ir a `https://finance.yahoo.com` y extraerlo.

```
1 getSymbols("^GSPC", src = "yahoo", from = "2010-01-01", to = "2018-07-30", periodicity =
  ↪  "daily")
```

```
## [1] "GSPC"
```

¿Como son los datos?

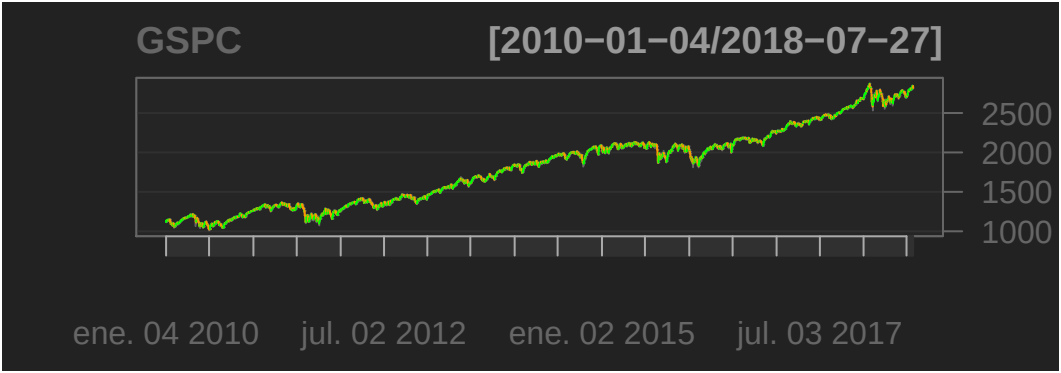
```
1 # Primera 1 observaciones con las 5 primeras columnas
2 head(GSPC[,1:5],5)
```

GSPC.Open	GSPC.High	GSPC.Low	GSPC.Close	GSPC.Volume
1116.56	1133.87	1116.56	1132.99	3991400000
1132.66	1136.63	1129.66	1136.52	2491020000
1135.71	1139.19	1133.95	1137.14	4972660000
1136.27	1142.46	1131.32	1141.69	5270680000
1140.52	1145.39	1136.22	1144.98	4389590000

Gráficos usando Quantmod

Función chartSeries

```
1 # Graficamos usando chartSeries sin análisis técnico
2 chartSeries(GSPC, TA=NULL)
```



Función chartSeries

Como se ve, en el eje de las x muestra el periodo y en el eje de las ordenadas el precio. La opción `TA` implica que no hay ningún análisis técnico. sin `TA` aparecen el volumen.

```
# Graficamos usando chartSeries con volume
chartSeries(GSPC)
```

Pero cuando las series son muy largas, podemos ver tendencias pero dificulta ver cambios importantes a nivel de análisis técnico.

```
# Graficando S&P 500 con Valume y los tres últimos meses
chartSeries(GSPC, subset = "last 3 months")
```

Con el código anterior nos enfocamos solo en los tres meses anteriores.

Gráfico con ggplot2

Debemos cargar ggplot2, pero para esto usamos tidyverse.

```
1 library("tidyverse")
```

luego graficamos:

```
1 # Usando ggplot2
2 g1 <- ggplot(gspc) + geom_line(mapping = aes(index(gspc),GSPC.Adjusted))
3 g1 <- g1 + labs(title = "S&P 500", subtitle = "Desde Enero 2010 a 2018") + xlab("Fecha")
  ↪  + ylab("")
4 g1 <- g1 + theme_bw()
5 g1
```

¿Observan algo que está mal?

Trabajando con múltiples datos

Oracle, Nvidia, IBM y AMD I

A continuación trabajaremos con las acciones de Oracle, Nvidia, IBM y AMD, comenzamos con crear un objeto con los nombres de los tickers

```
# Nuevos tickers
tickers <- c("ORCL","AMD","IBM","NVDA")
```

descargamos los datos con las características requeridas, que son las mismas que usamos anteriormente con S&P 500

```
# descargamos los tickers
getSymbols(tickers, src = "yahoo", from = "2010-01-01", to = "2018-07-30", periodicity =
↪  "daily")
```

Acá deben tener mucha atención (Wickham 2014):

Oracle, Nvidia, IBM y AMD II

```
# Precio de cierre
list <- lapply(tickers, function(x) Cl(get(x)))
precio.cierre <- do.call(merge,list)
```

Cálculo de los retornos I

La ecuación (1) es la utilizada para calcular (log) retornos:

$$r_t = log(1 + R_t) = log(\frac{P_t}{P_{t-1}}) = p_t - p_{t-1} \quad (1)$$

donde $p_t = log(P_t)$ es llamado "log price" (Ruppert 2011).
A veces nos puede molestar tener tanta objetos que no vamos a utilizar:

```
1 # removemos los objetos que no vamos a usar
2 rm(tickers, AMD, IBM, NVDA, ORCL, list)
```

Ahora pasamos a construir el retorno

Cálculo de los retornos II

```
# calculamos los  retornos
retornos <- data.frame(apply(precio.cierre, 2, function(x) Delt(x, type = "log")),
                       fecha = index(precio.cierre)) %>%
            rename(orcl = ORCL.Close, amd = AMD.Close, ibm = IBM.Close, nvda =
↪  NVDA.Close) %>%
            na.omit()
```

Cálculo de los retornos acumulados

Si graficamos los retornos no será muy descriptivo, una forma es trabajar con su acumulado. Con la misma lógica usamos la función `cumsum()`.

```
1 # calculamos los retornos acumulados
2 acumulados <- data.frame(apply(retornos[1:4], 2, function(x) cumsum(x)), fecha =
  ↪  index(precio.cierre[-1]))
```

Gráfico retornos acumulados

La librería ggplot2 trabaja por "capas":

1. Base de datos
2. Tipo de gráfico: geom_line, geom_point, entre otros.
3. Todo lo extra, que sería título, subtítulo, nombre de los ejes, etc.

```
# Cambiamos la forma de los datos
reshape <- melt(acumulados, id.vars = "fecha")
```

```
# graficamos los retornos acumulados forma 2
g3 <- ggplot(reshape) + geom_line(mapping = aes(fecha,value, color = variable))
g3 <- g3 + labs(title = "Retornos Acumulados", subtitle = "Oracle, AMD, IBM y Nvidia")
g3 <- g3 + theme_bw() + xlab("Fecha") + ylab("Retornos Acumulados")
g3 <- g3 + scale_color_manual("Tickers", values = c("red","red","green","orange"))
g3 <- g3 + theme(legend.position = "bottom")
g3
```

Estadística Descriptiva

Estadística Descriptiva

Existe muchas formas de obtener la estadística descriptiva en R, un librería es `fBasics`, la que a su vez contiene test de normalidad.

```
# cargamos la librería fBasics
library("fBasics")

# construímos un objeto con las estadística descriptiva
summary <- basicStats(retornos[1:4])[c("Mean", "Stdev", "Median", "Minimum",
                                       "Maximum", "nobs","Skewness","Kurtosis"),]
```

Recursos de la ayudantía: apunte del curso y videos

Este semestre para complementar su camino en aprender R es que podran acceder a un apunte en construcción:

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

Algunas sesiones tiene incluido videos tutoriales:

- Parte 1 Introducción a quantmod: `https://youtu.be/TwBKLTq3mfY`
- Parte 2 Retornos y retornos acumulados: `https://youtu.be/-3RvuhtfNGU`
- Parte 3 Graficar los retornos acumulados y estadística descriptiva: `https://youtu.be/qlLFjBMSUlQ`

Referencia I

David Ruppert. *Statistics and data analysis for financial engineering*. Vol. 13. Springer, 2011.

Hadley Wickham. *Advanced r*. Chapman and Hall/CRC, 2014.

Hadley Wickham et al. "Tidy data". In: *Journal of Statistical Software* 59.10 (2014), pp. 1–23.