

Índice general

1. Introducción	3
2. Conociendo datos y términos financieros	7
2.1. Paqueterías de R	8
2.2. Precios de Acciones	9
2.2.1. Tickers	9
2.2.2. OHLC	11
2.2.3. Liquidez	15
2.3. Cambio relativo y rendimiento de acciones	19
2.3.1. Cambio relativo y rendimiento en el periodo \mathbb{T}	19
2.3.1.1. Rendimiento en diferentes tipos de cambio	20
2.3.2. Cambios relativos diarios	21
2.3.3. Rendimiento acumulado	23
2.4. Precios Ajustados	25
2.4.1. Dividendos	26
2.5. Valor de un portafolio	28
2.5.1. Posiciones Largas y Posiciones Cortas	31
2.6. Rebalanceo y costos de transacción	32
2.7. Índices	35
2.7.1. Proveedores de Índices	36
2.7.1.1. Capitalización de mercado	37
2.8. ETFs	40

Capítulo 1

Introducción

El objetivo de este trabajo tiene tres frentes:

1. *Ser una propuesta didáctica de manejo de portafolios.* Esto es, presentamos de forma simple y con ejemplos las bases, en términos de lenguaje y manejo de datos, de los conceptos que son parte del día a día de los manejadores de portafolios.
2. *Ofrecer herramientas que permitan al lector hacer manejo de portafolios por su cuenta.* En ese sentido, dejamos disponible en github una paquetería en **R** que desarrollamos a lo largo del trabajo. Esta paquetería facilita extraer en cualquier momento datos de precios y generar de forma reproducible los análisis que se presentan en este trabajo.
3. *Proponer una estrategia de inversión basada en paridad de riesgo.* Asumiremos que somos manejadores de portafolios y que nos rige un mandato de inversión al que nos tenemos que apegar. Describiremos todas las partes de un mandato hipotético y explicaremos como traducir estas partes a una estrategia educada de inversión. Se exploran los conceptos detrás de paridad de riesgo presentados por Thierry Roncalli (Roncalli, 2013), se modifica la formulación para incorporar componentes principales y se prueban los resultados históricos de la estrategia.

Harry Markowitz acuñó hace más de 60 años los principios de la Teoría Moderna de Portafolios a partir de la cual se construyen portafolios en base a combinaciones óptimas de retornos esperados y apetitos de riesgo. Éstas combinaciones son soluciones al problema de optimización de media-varianza el cual descansa en el supuesto que los manejadores de portafolios basan sus decisiones de inversión primordialmente en valores esperados y covarianzas de rendimientos.

Las dificultades que ha presentado la estimación de los retornos esperados ha hecho que la implementación de la Teoría Moderna de Portafolios sea difícil de aplicar a los portafolios en la realidad. Por otro lado, la optimización media-varianza es muy sensible

a las entradas del modelo de tal manera que las posiciones óptimas resultantes suelen ser soluciones esquina, es decir tienen altas concentraciones y son inestables de un período a otro.

La teoría de Markovitz regularmente se asocia con la idea de diversificación; sin embargo, **la optimización de media-varianza no se basa en ninguna medida de diversificación**. De hecho las soluciones esquina en general conllevan a poca diversificación. En nuestro contexto, la diversificación será medida de acuerdo al porcentaje de riesgo asignado a distintas fuentes de riesgo (risk budgeting).

Atendiendo el tercer objetivo de este trabajo, supondremos que somos manejadores de portafolios y que nos tenemos que apegar a un mandato de inversión. El marco de inversión de nuestro mandato hipotético estará basado en diversificación. En ese sentido tanto el portafolio de referencia del mandato como la estrategia de inversión dependerán del concepto de paridad de riesgo.

Los métodos de construcción de portafolios basados en paridad de riesgo han adquirido popularidad en la última década pues no requieren estimaciones de retornos esperados (Kolm et al., 2014). El principio detrás es obtener portafolios cuya asignación de riesgo se encuentre perfectamente equilibrada. En particular, en este trabajo se presenta la noción de paridad de contribución al riesgo que ofrece Thierry Roncalli (Maillard et al., 2010).

Nosotros exploramos cómo incorporar componentes principales a la formulación de contribución al riesgo y a partir de esto proponemos una estrategia de inversión que busca superar satisfactoriamente al portafolio de referencia estipulado en nuestro mandato hipotético. Esta estrategia combina la optimización de media-varianza y los pesos de paridad de riesgo. De aquí es de donde viene el título de este trabajo: **Optimización de Portafolios con Retornos Implícitos de Paridad de Riesgo**.

Aquellos que no estén familiarizados con los términos que se utilizaron en la primer parte de esta introducción no deben desanimarse. Este trabajo está especialmente dirigido a alumnos de licenciatura que tengan conocimientos de álgebra y estadística (en particular que tengan entendimiento de componentes principales) pero que no conocen los conceptos y términos financieros que se utilizan en la literatura y práctica de manejo de portafolios. Basta que el lector esté familiarizado con el concepto de acciones y de casas de bolsa (exchanges); fuera de eso, se explicarán a detalle todos los términos que usemos.

Adicionalmente el público ideal tiene conocimiento básico de **R** y está interesado en aprender como usar **R** para manipular portafolios pues presentaremos todas las partes de una paquetería que permitirá al lector replicar todos los cálculos. Esta paquetería se encuentra en <https://github.com/audiracmichelle/optimist> y lo único que se requiere para descargarla es usar el siguiente código desde **R**.

```
devtools::install_github("audiracmichelle/optimist")
```

Con esta paquetería se invita a los lectores a seguir el contenido de este trabajo para posteriormente construir sus propios portafolios y, ¡salir al mercado a invertir!

Capítulo 2

Conociendo datos y términos financieros

Nuestro destino final: la optimización de la diversificación de portafolios, para llegar ahí debemos primero hacer un viaje por el mundo de términos que se utilizan en la industria de manejo de portafolios. Por esto, en este capítulo no solo se va familiarizar al lector con los conceptos utilizados frecuentemente por los manejadores de portafolios. El objetivo principal será construir las funciones básicas de una paquetería de optimización de portafolios. Estas funciones nos permitirán manipular rápidamente los precios de activos.

En la sección 2.1 de este capítulo, empezamos preparando la caja de herramientas que necesitamos para armar nuestra paquetería de optimización de portafolios en **R**. En particular resulta práctico conocer como se contruyen los objetos **xts**, éstos objetos contienen series de tiempo.

En la sección 2.2 construimos la función `get_prices_yahoo()` con la que vamos a extraer los datos que utilizaremos en el resto del trabajo. Esta función obtiene precios de acciones de Yahoo! Finance y produce objetos **xts** fácilmente. Familiarizarnos con los *tickers* y el término *OHLC* nos permitirá entender los parámetros de `get_prices_yahoo()`. Adicionalmente, en esta sección creamos la función `plot_xts()` que utilizaremos regularmente para graficar series de tiempo.

Más adelante, en la sección 2.3 presentaremos varios conceptos y funciones relacionados con el cálculo de *rendimientos*. Hacemos énfasis en cómo obtener rendimientos en distintos tipos de cambio pues resulta indispensable utilizar rendimientos en la misma moneda al momento de construir portafolios.

El cálculo del *rendimiento acumulado* (usando la función `get_cumRet_from_dlyChg()`) nos permitirá entender los *precios de cierre ajustados* y el *valor de un portafolio*, cómo

se cubre en las secciones 2.4 y 2.5. En los cursos básicos de matemáticas financieras regularmente se muestran ejemplos donde se fijan los pesos iniciales de las componentes de los portafolios; sin embargo, los pesos de los portafolios cambian conforme se mueven los precios de sus componentes y es necesario volver a fijarlos a través de *rebalanceos* como se muestra en la sección 2.6

En particular, los índices de acciones y bonos son portafolios con distintas configuraciones de frecuencia y pesos de rebalanceo. En la sección 2.7 vamos a construir un índice con rebalanceos trimestrales y pesos de *capitalización de mercado*. Para esto usamos las funciones que construimos a lo largo del capítulo dentro de las que destacan: `get_dlyChg_from_price()` y `get_rebalance()`.

Finalmente, en la sección 2.8 se presenta un tipo de activo llamado *ETF*. Hablaremos de las cualidades que hacen que los ETF sean los instrumentos más operados en el mundo. Consecuentemente, en los siguientes capítulo ya no usaremos acciones y únicamente trabajaremos con este tipo de instrumentos.

Notemos que adicionalmente, a lo largo de este primer capítulo se presentan los conceptos *liquidez*, *precios ajustados*, *costos de operación*, *posiciones largas y cortas*; también se mencionan algunos jugadores importantes en la industria como son los *proveedores de índices*, los *proveedores de tickers*, las *casas de bolsa* y los *custodios*.

2.1. Paqueterías de R

A continuación describimos todas las paqueterías de **R** (R Core Team, 2018) que necesitamos para cargar las funciones que usaremos a lo largo de todo este trabajo.

- **xts** (Ryan and Ulrich, 2018b): es una paquetería en **R** para manejo de objetos que contienen series de tiempo. El nombre **xts** se refiere a *extensible time series* porque esta paquetería *extiende* las propiedades de **ts** y **zoo**, paqueterías de **R** más simples para manejar series de tiempo).

A lo largo de este trabajo nos iremos familiarizando con las funciones de **xts** o **zoo** (Zeileis and Grothendieck, 2018). Cuando se use por primera vez un método de una paquetería se intentará hacer explícito (e.g., `zoo::index()`, `lag.xts()`).

- **quantmod** (Ryan and Ulrich, 2018a): es una paquetería que proporciona una plantilla que nos va a servir para manejar y visualizar de forma sencilla series de datos financieros.
- Para los problemas de optimización usaremos **nloptr** (Johnson, 2018), esta librería envuelve una serie de rutinas de código abierto de optimización cuadrática.
- **tidyverse** (Wickham, 2017), **mltools** (Gorman, 2018) y **Rcpp** (Eddelbuettel, 2018): estas paqueterías nos facilitan la manipulación de los objetos en **R** y nos ayudan a mejorar la eficiencia de algoritmos.

El código abajo es para instalar y cargar las paqueterías.

```
pkgs <- c('xts', 'quantmod', 'nloptr', 'tidyverse', 'mltools', 'Rcpp')

# para instalar paquetes usar:
#install.packages(pkgs,
#                  dependencies = TRUE,
#                  repos = "http://cran.itam.mx")

# para cargar paquetes usar:
lapply(pkgs, require, character.only = TRUE)
```

2.2. Precios de Acciones

Vamos a extraer precios reales de acciones de Yahoo! Finance. Para esto, usaremos la función `getSymbols.yahoo()` de **quantmod**.

Empecemos por elegir una empresa y un periodo, digamos:

- empresa - International Business Machines Corporation (IBM)
- período - 2017 (del 31 de Diciembre de 2016 al 31 de Diciembre de 2017)

Con esto, vamos a crear el objeto `ibm`.

```
# `getSymbols(IBM)`, `getSymbols(IBM, src = "yahoo")` o
# `getSymbols.yahoo(IBM)` son equivalentes

ibm <- getSymbols.yahoo('IBM', auto.assign=FALSE,
                        from = '2016-12-31', to = '2017-12-31')
```

Veamos que la clase de objeto que nos regresa `getSymbols()` es "xts" "zoo".

```
class(ibm)
```

```
## [1] "xts" "zoo"
```

De aquí en adelante nos referiremos a un objeto de este tipo como un `xts`.

2.2.1. Tickers

A las acciones que están listadas en bolsas de valores se les asignan identificadores llamados **tickers**. Por ejemplo, IBM es un ticker para acciones de IBM listadas en el NYSE (*New York Stock Exchange*). AAPL y GOOG son tickers de emisiones de Apple y Google listadas en el Nasdaq (*National Association of Securities Dealers*).

Es importante saber que además de los tickers de las casas de bolsa existen otros identificadores que han sido creados por diferentes entidades. Por ejemplo:

- El *National Numbering Agency* (NNA) creo el ISIN (*International Securities Identification Number*).
- El *CUSIP Service Bureau*, operado por *Standard & Poor's for the American Bankers Association* creo el CUSIP (*Committee on Uniform Securities Identification Procedures*).
- También empresas como Yahoo! Finance y Bloomberg crean sus propios identificadores (aunque muchas veces son iguales a los tickers de las casas de bolsa).

Veamos algunos ejemplos en la siguiente tabla:

Ticker	Yahoo! Finance	ISIN	CUSIP
IBM	IBM	US4592001014	459200101
AAPL	AAPL	US0378331005	037833100

Aunque existen bases de datos que mapean Tickers, ISINs, CUSIPs y otros identificadores, **estas bases son propietarias y se paga el servicio para tener acceso a ellas.**

En este trabajo estaremos usando la función `getSymbols.yahoo()`. Para usarla necesitamos conocer los identificadores de Yahoo! Finance. Desafortunadamente, no contamos con una base de datos que liste los identificadores de Yahoo!. Sin embargo, para nuestro propósito, basta con hacer búsquedas de estos identificadores en la página <https://finance.yahoo.com/lookup>.

Además de identificadores para acciones, **Yahoo! Finance también cuenta con identificadores para otros tipos de instrumentos, tipos de cambio e indicadores financieros.** Por ejemplo, el identificador de yahoo para el tipo de cambio pesos por dólar es `MXN=X`.

Observemos el contenido en `ibm`.

```
head(ibm)
```

##		IBM.Open	IBM.High	IBM.Low	IBM.Close	IBM.Volume	IBM.Adjusted
##	2017-01-03	167.00	167.87	166.01	167.19	2934300	145.4166
##	2017-01-04	167.77	169.87	167.36	169.26	3381400	147.2170
##	2017-01-05	169.25	169.39	167.26	168.70	2682300	146.7300
##	2017-01-06	168.69	169.92	167.52	169.53	2945500	147.4518
##	2017-01-09	169.47	169.80	167.62	167.65	3189900	145.8167
##	2017-01-10	167.98	168.09	165.34	165.52	4118700	143.9641

Como vemos, `getSymbols.yahoo()` regresa seis series:

- Open - precio de apertura
- High - precio más alto del día
- Low - precio más bajo del día
- Close - precio de cierre
- Volume - volumen o número de transacciones en el día
- Adjusted - precio de cierre ajustado (o precio ajustado)

Más adelante se explica qué es OHLC (Open, High, Low, Close), porqué es importante el volumen y cómo obtener precios de cierre ajustados.

Los índices de los renglones de un objeto `xts` son fechas ordenadas. Para extraer las fechas de un objeto `xts` se usa la función `xts::index`.

```
dates_ibm <- index(ibm)
head(dates_ibm, 15)
```

```
## [1] "2017-01-03" "2017-01-04" "2017-01-05" "2017-01-06" "2017-01-09"
## [6] "2017-01-10" "2017-01-11" "2017-01-12" "2017-01-13" "2017-01-17"
## [11] "2017-01-18" "2017-01-19" "2017-01-20" "2017-01-23" "2017-01-24"
```

Podemos ver en las fechas que imprimimos que las series que extrajimos con `getSymbols.yahoo()` son diarias. Notemos que **no tenemos observaciones los fines de semana ni los días feriados**.

2.2.2. OHLC

OHLC, que se refiere a *Open High Low Close*, es un *standard* popular en la forma de presentar precios porque ofrece un resumen de los movimientos más relevantes de las transacciones de un instrumento financiero pactadas en un día.

El NYSE, donde se hacen las transacciones de las acciones de IBM, funciona días hábiles entre 9:30am y 4:00pm EST (Eastern Standard Time). Se dice que el **precio de apertura** (open) es el primer precio después de las 9:30am al que se pacta una transacción de IBM en el día. El **precio de cierre** (close) es el último precio antes de las 4:00pm al que se pacta una transacción de IBM en el día.

En realidad, los precios de cierre y apertura no son exactamente el primer y último precio al que se cierra una transacción. Dependiendo de la casa de bolsa, los precios de cierre y apertura que se publican pueden ser **el promedio de los precios pactados en los últimos/primeros minutos del día de operación**.

Para familiarizarnos con los precios de apertura y cierre, veamos una captura de pantalla de una gráfica con precios intradía de IBM que se obtuvo de la



Figura 2.1: Serie de precios de IBM

página <https://finance.yahoo.com/chart/IBM>.

El punto verde en la figura 2.1 es el precio de cierre de IBM del 15 de diciembre, el punto naranja es el precio de apertura del 18 de diciembre.

Para imprimir los precios de apertura y cierre que tenemos guardados en `ibm` entre el 14 y 19 de Diciembre usamos `/` en `'2017-12-14/2017-12-19'` de la siguiente manera:

```
ibm['2017-12-14/2017-12-19', c('IBM.Open', 'IBM.Close')]
```

##		IBM.Open	IBM.Close
##	2017-12-14	154.60	154.00
##	2017-12-15	153.61	152.50
##	2017-12-18	153.59	153.33
##	2017-12-19	154.05	153.23

Entre el cierre del 15 y la apertura del 18 de Diciembre no hubo operaciones en el NYSE, entonces ¿porqué tienen una diferencia de más de un dólar el cierre del 15 y la apertura del 18 mientras el resto de los días no ocurre esto? Esto ocurre porque aunque existe un horario de operación de acciones, las noticias que afectan a IBM (o a la industria Tech o a USA) no están acotadas a ningún horario. **Aquellos eventos que ocurren fuera del horario de operación se reflejan en el siguiente precio de apertura.**

En la imagen también podemos ver que el 18 de Diciembre hubo poca volatilidad (poco movimiento) en el precio de IBM, pero el 12 de Diciembre hubo mucha volatilidad. Esto lo corroboramos comparando el `'High'` que es el **precio más alto del día**, contra el `'Low'` que el **precio más bajo del día**.

```
ibm[c('2017-12-18', '2017-12-12'), c('IBM.High', 'IBM.Low')] %>%
  as.data.frame() %>%
  tibble::rownames_to_column("date") %>%
  mutate(diff = IBM.High - IBM.Low)
```

```
##           date IBM.High IBM.Low    diff
## 1 2017-12-12   157.85  155.16 2.690002
## 2 2017-12-18   154.18  153.21 0.969986
```

Ya vimos que `getSymbols.yahoo()` nos proporciona series diarias pero también vimos que a veces hay movimientos intradía interesantes, ¿no necesitamos una función que extraiga series con todos los precios intradía de las acciones? **Los modelos de optimización que vamos a analizar sólo requieren series diarias de precios de cierre.** De hecho, las posiciones de trading intradía por lo regular no se mantienen al cierre mientras que los modelos que usaremos se basan en los cambios de cierre a cierre.

Extraigamos de Yahoo! Finance una serie que contenga el tipo de cambio pesos por dólar.

```
mxn_usd <- getSymbols.yahoo('MXN=X', auto.assign=FALSE)['2017']
head(mxn_usd)
```

```
##           MXN=X.Open MXN=X.High MXN=X.Low MXN=X.Close MXN=X.Volume
## 2017-01-02    20.7183    20.7590    20.6348    20.7405             0
## 2017-01-03    20.7397    21.1332    20.6654    21.1202             0
## 2017-01-04    21.1187    21.6175    21.0082    21.4797             0
## 2017-01-05    21.4753    21.6047    21.1390    21.3855             0
## 2017-01-06    21.3779    21.4656    21.1946    21.2165             0
## 2017-01-09    21.2456    21.3839    21.1896    21.3591             0
##           MXN=X.Adjusted
## 2017-01-02    20.7405
## 2017-01-03    21.1202
## 2017-01-04    21.4797
## 2017-01-05    21.3855
## 2017-01-06    21.2165
## 2017-01-09    21.3591
```

Podemos ver que cuando extraemos tipos de cambio, `getSymbols.yahoo()` regresa ceros en la columna `Volume` y que la columna de precios de cierre es igual a la de precios de cierre ajustados.

Aún no hemos explicado que son los precios de cierre ajustados. Antes de que se explique a detalle el concepto, supongamos simplemente que son precios de cierre.

Ahora, comparemos los precios de acciones de dos empresas mexicanas: Herdez y Gigante. Para eso, extraigamos con la función `xts::Ad()` los precios de cierre ajustados

de cada uno en 2017. Guardemos en herdez y gigante estos precios.

```
herdez <- getSymbols('herdez.mx', auto.assign=FALSE,
                    from = '2016-12-31', to = '2017-12-31')

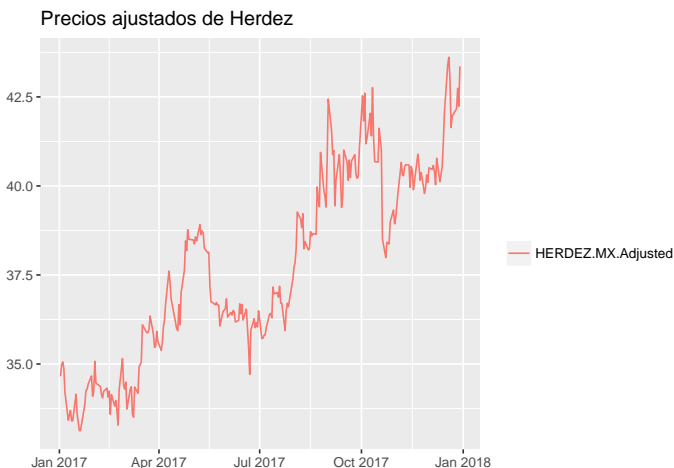
herdez <- Ad(herdez)
gigante <- getSymbols('gigante.mx', auto.assign=FALSE,
                    from = '2016-12-31', to = '2017-12-31')
gigante <- Ad(gigante)
```

Vamos a crear una función que llamaremos `plot_xts` que recibe un objeto `xts` y que utiliza la paquetería `ggplot2` para graficar series de tiempo. Esta será la primer función de nuestra paquetería de optimización de portafolios.

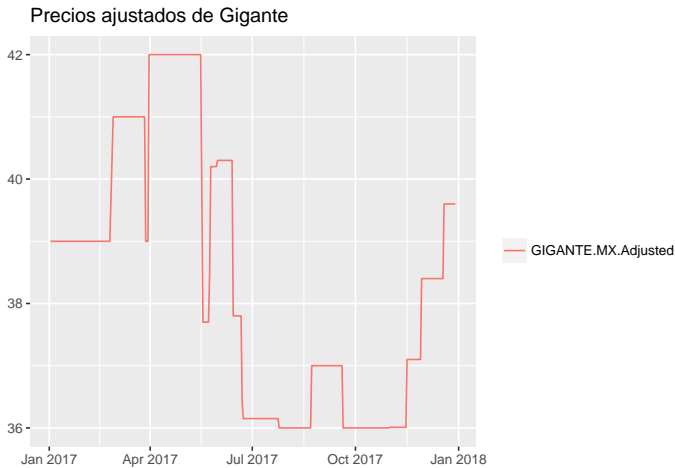
```
plot_xts <- function(x){
  x %>%
    as.data.frame() %>%
    tibble::rownames_to_column("date") %>%
    mutate(date = as.Date(date)) %>%
    gather(key, value, -date) %>%
    ggplot(aes(x = date, y = value, color = key)) +
    geom_line() +
    labs(x = "", y = "", color = "")
}
```

Veamos las gráficas de los precios ajustados diarios en los objetos `herdez` y `gigante`!

```
plot_xts(herdez) +
  labs(title = 'Precios ajustados de Herdez')
```



```
plot_xts(gigante) +  
  labs(title = 'Precios ajustados de Gigante')
```



Podemos ver que las acciones de Herdez son **líquidas** mientras que las acciones de Gigante son **ilíquidas**.

2.2.3. Liquidez

¿Porqué el precio de Gigante parece que se estaciona en distintos periodos en comparación con el precio de Herdez? Los precios de los instrumentos financieros se determinan al momento en el que se pacta una transacción. En este ejemplo podemos ver que todos los días se pactan transacciones de Herdez pero no es el caso de Gigante.

```
head(herdez)
```

```
##          HERDEZ.MX.Adjusted  
## 2017-01-02          34.65604  
## 2017-01-03          34.97813  
## 2017-01-04          35.06094  
## 2017-01-05          34.87690  
## 2017-01-06          34.20513  
## 2017-01-09          33.42293
```

```
head(gigante)
```

```
##          GIGANTE.MX.Adjusted  
## 2017-01-02                  39  
## 2017-01-03                  39
```

## 2017-01-04	39
## 2017-01-05	39
## 2017-01-06	39
## 2017-01-09	39

Las bolsas de valores determinan los sistemas y medios electrónicos en los que se hacen las operaciones de compra-venta de instrumentos financieros. Supongamos que manejamos un portafolio en una casa de bolsa que tiene acciones de Gigante que queremos vender. El 3 de Enero de 2017 colocamos un precio de venta de 42 pesos en los sistemas de la casa de bolsa, pero no hay ninguna postura de compra en todo el día. Entonces el precio de cierre del 3 de Enero se queda igual al precio de cierre del 2 de Enero (39 pesos). Esto se repite día con día, no hay posturas de compra y no logramos vender esas acciones. Entonces el precio de Gigante se mantiene en 39 pesos.

El 27 de Febrero decidimos bajar nuestra postura a 41 pesos y finalmente alguien accede a colocar una postura de compra a ese precio. Se cierra la operación y el precio de cierre de Gigante de ese día se fija en 41 pesos como se ve en la gráfica.

Un instrumento financiero es ilíquido cuando es difícil de comprar o vender. Existen medidas como *share turnover* o *rotación de títulos* que nos dicen qué tan líquido es un instrumento. Para calcular la rotación de títulos se utiliza el volumen, que es el número de posturas de una acción que se cierran en un día.

Como vimos, `getSymbols.yahoo()` nos proporciona la serie diaria de volumen para una acción. En este trabajo no nos detendremos en cómo obtener la rotación de títulos, tampoco veremos cuánta liquidez es deseable en un instrumento. Para nuestro propósito nos basta entender que si **el precio de cierre de una acción se mantiene estática por varios días, entonces es una acción poco líquida.**

Liquidez es un concepto que debe entenderse y tomarse en cuenta al seleccionar el universo de activos que se usarán al momento de definir un problema de optimización de portafolios.

Ahora, guardemos en un mismo objeto que llamaremos `price` los precios ajustados en 2017 de `ibm`, `herdez` y `gigante`. Una manera de hacerlo es *pegando* los objetos `Ad(ibm)`, `herdez` y `gigante` con la función `cbind.xts`.

```
price <- cbind.xts(Ad(ibm), herdez, gigante)
head(price)
```

##	IBM.Adjusted	HERDEZ.MX.Adjusted	GIGANTE.MX.Adjusted
## 2017-01-02	NA	34.65604	39
## 2017-01-03	145.4166	34.97813	39
## 2017-01-04	147.2170	35.06094	39

## 2017-01-05	146.7300	34.87690	39
## 2017-01-06	147.4518	34.20513	39
## 2017-01-09	145.8167	33.42293	39

Observemos que `ibm` no cuenta con observaciones para el día 2017-01-02; `herdez` y `gigante` sí cuentan con observaciones para ese día. Al crear `price` la entrada `['2017-01-02', 'IBM.Adjusted']` se rellena con `NA`. De esta manera, el método `cbind.xts` nos facilita pegar objetos `xts` que tienen índices distintos.

Para trabajar con series extraídas con `getSymbols.yahoo()` y pegarlas en un mismo objeto `xts` vamos a crear la función `get_prices_yahoo()`.

Como queremos que el objeto que regrese `get_prices_yahoo()` no contenga `NA`'s, vamos a imputar los `NA`s usando `xts::na.locf()`, un método de `xts` que reemplaza cada valor vacío con el siguiente valor en la serie que sea distinto de `NA`.

Ponemos “Adjusted” como valor predeterminado de un parámetro `column` de `get_prices_yahoo()`. El valor del parámetro se puede cambiar por “Open”, “High”, “Low”, “Close” y “Volume”.

```
get_prices_yahoo <- function(
  yahoo_id,
  column = 'Adjusted',
  from = "2007-01-01",
  to = Sys.Date(),
  ...,
  periodicity = "daily"
) {
  series <- list()
  i <- 0

  for(id in yahoo_id) {
    col <- paste(toupper(id), '.', column, sep = '')

    ss <- getSymbols.yahoo(id, auto.assign = FALSE,
                          from = from, to = to,
                          periodicity = periodicity, ...)[, col]

    i <- i + 1
    series[[i]] <- ss
  }

  series <- do.call(cbind, series)
  series <- na.locf(series, na.rm = FALSE, fromLast = TRUE)
  series
}
```

Volvamos a generar price pero esta vez con la función que acabamos de crear y añadiendo el tipo de cambio como el que guardamos en `mxn`. Recordemos que las columnas `Close` y `Adjusted` son idénticas para el tipo de cambio.

```
price <- get_prices_yahoo(c('IBM',
                           'herdez.mx',
                           'gigante.mx',
                           'MXN=X'),
                        from = '2016-12-31',
                        to = '2017-12-31')

head(price)
```

```
##          IBM.Adjusted HERDEZ.MX.Adjusted GIGANTE.MX.Adjusted
## 2017-01-02      145.4166           34.65604                39
## 2017-01-03      145.4166           34.97813                39
## 2017-01-04      147.2170           35.06094                39
## 2017-01-05      146.7300           34.87690                39
## 2017-01-06      147.4518           34.20513                39
## 2017-01-09      145.8167           33.42293                39
##          MXN.X.Adjusted
## 2017-01-02       20.7405
## 2017-01-03       21.1202
## 2017-01-04       21.4797
## 2017-01-05       21.3855
## 2017-01-06       21.2165
## 2017-01-09       21.3591
```

Con la función `plot_xts()` y el objeto `price` que acabamos de crear, hagamos una gráfica para comparar los precios de nuestros instrumentos.

```
plot_xts(price)
```

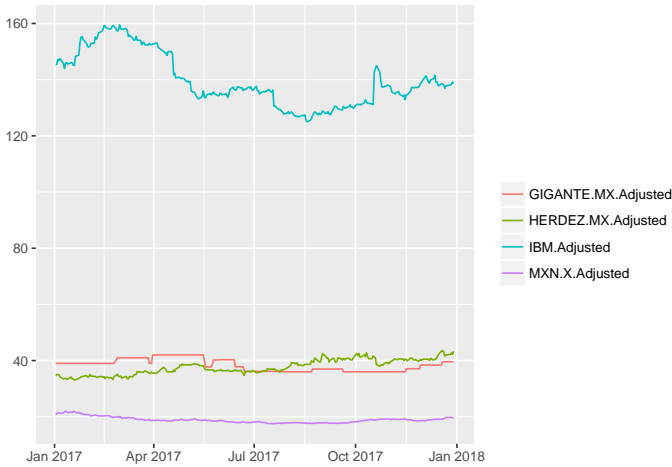


Figura 2.2: Precios de Acciones

2.3. Cambio relativo y rendimiento de acciones

Ya sabemos cómo extraer precios, combinarlos en un mismo objeto y visualizarlos en una gráfica. Ahora, nos resulta interesante comparar cuánto cambiaron nuestras acciones por cada unidad invertida al inicio del 2017. Para esto, definiremos el **cambio relativo** y el **rendimiento**.

2.3.1. Cambio relativo y rendimiento en el periodo \mathbb{T}

Digamos que tenemos un periodo \mathbb{T} que empieza en el tiempo 0 y termina al tiempo T . Al **cambio relativo de 0 a T** lo definimos como

$$\delta^{\mathbb{T}} = \delta^{0,T} = p^T / p^0 \quad (2.1)$$

donde p^T es el precio en T y p^0 es el precio en el tiempo 0. Para nuestro caso de uso, los “tiempos” siempre serán días.

Calculemos el cambio relativo en el 2017 de nuestras acciones.

```
pT <- as.matrix(price[nrow(price), ])
p0 <- as.matrix(price[1, ])
pT / p0
```

```
##          IBM.Adjusted HERDEZ.MX.Adjusted GIGANTE.MX.Adjusted
## 2017-12-29      0.9532303              1.251087              1.015385
```

```
##                MXN.X.Adjusted
## 2017-12-29      0.9476821
```

Como el precio en IBM que extrajimos con `getSymbols` está en dólares y los precios de Herdez están en pesos mexicanos, entonces tenemos que en 2017:

- un peso en Herdez dejó 1.25 pesos, y
- un dólar en IBM dejó 0.95 dólares.

Ya que un peso dejó 0.95 dólares, **los cambios de precio en distintos tipos de cambio no son comparables**. Más adelante hablaremos a detalle sobre esto.

El **rendimiento en el periodo** \mathbb{T} simplemente es el cambio porcentual que tuvo el precio de una acción en ese periodo. Esto es exactamente igual al cambio relativo de 0 a T menos uno.

$$r^{\mathbb{T}} = r^{0,T} = (p^T - p^0)/p^0 = p^T/p^0 - 1 = \delta^{0,T} - 1$$

Para obtener el rendimiento en el 2017 de nuestras acciones simplemente haremos

$p^T / p^0 - 1$

```
##                IBM.Adjusted  HERDEZ.MX.Adjusted  GIGANTE.MX.Adjusted
## 2017-12-29    -0.04676966          0.2510874          0.01538456
##                MXN.X.Adjusted
## 2017-12-29    -0.05231793
```

Esto equivale a decir que en este periodo:

- las acciones Herdez dieron un rendimiento en pesos de 25.11 %, y que
- las IBM dieron un rendimiento en dólares de -4.68 %.

2.3.1.1. Rendimiento en diferentes tipos de cambio

Supongamos que un inversionista americano decide invertir por un año en xyz, una acción mexicana listada en la BMV (Bolsa Mexicana de Valores). Para poder comprar xyz el inversionista debe cambiar sus dólares a pesos.

Al final del año el rendimiento en pesos de xyz es 0 % y el peso se fortalece contra el dolar. El inversionista vende xyz y recibe pesos. Al cambiar los pesos por dólares el inversionista recibe más pesos que los que invirtió inicialmente pues el valor del peso contra el dolar aumentó.

Esto significa que, aunque el rendimiento en pesos de xyz fue 0 %, el rendimiento en dólares que recibió el inversionista fue mayor a cero.

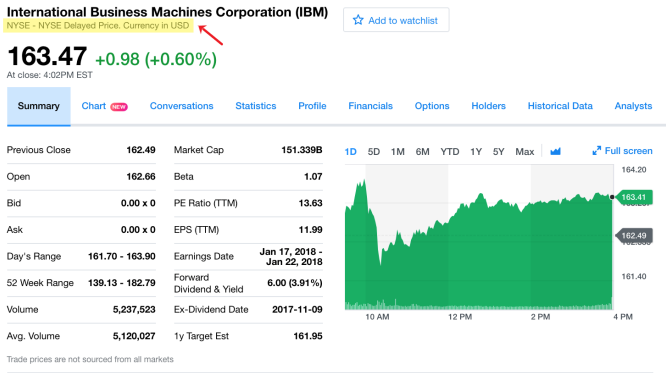


Figura 2.3: Información de IBM en Yahoo! Finance

El precio de un instrumento financiero está en la moneda que maneja la casa de valores en el que está listado. Desafortunadamente, **bases de datos con información sobre la denominación de los instrumentos financieros suelen ser propietarias y el acceso a esa información es pagado.**

Una manera de consultar la denominación de un instrumento financiero es hacer una búsqueda en la página <https://finance.yahoo.com/> usando el identificador de yahoo del instrumento. El resultado mostrará un resumen del instrumento financiero. Por ejemplo, la figura 2.3 es una captura de pantalla del resumen de IBM. **En la imagen podemos ver marcado en amarillo el nombre la casa de valores donde están listadas las acciones de IBM (NYSE) y la denominación de la emisión (USD).**

2.3.2. Cambios relativos diarios

Los cambios (relativos) diarios son simplemente el cambio relativo de una acción de $t - 1$ a t para cada día t en cierto periodo. Al **cambio diario en t** lo denotaremos por δ^t .

$$\delta^t = p^t / p^{t-1}$$

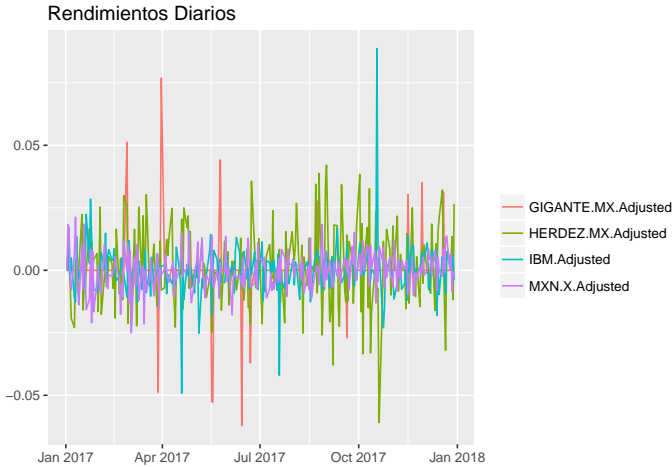
Obtengamos los cambios relativos diarios de las acciones y del tipo de cambio en price usando lag.xts().

```
# calculamos cambios relativos diarios
dlyChg <- price / lag.xts(price)
```

```
# por conveniencia asignamos 1's al primer renglon de `dlyChg`
dlyChg[1, ] <- 1
```

Los **rendimientos diarios** son los cambios relativos diarios menos 1. Para ver una gráfica con los rendimientos diarios de nuestras acciones en 2017 restamos 1 a dlyChg.

```
ret = dlyChg - 1
plot_xts(ret) +
  labs(title = "Rendimientos Diarios")
```



Al multiplicar o *acumular geométicamente* todos los cambios diarios dentro del periodo \mathbb{T} , obtenemos el cambio relativo de 0 a T .

$$\delta^{\mathbb{T}} = \delta^{0,t} = \delta^0 \delta^1 \dots \delta^t = (1 + r^0)(1 + r^1) \dots (1 + r^t) \quad (2.2)$$

Esto significa que tenemos dos maneras de calcular cambios relativos en un periodo:

- directamente dividiendo el precio final entre el precio inicial del periodo como en (2.1)
- multiplicando o acumulando los rendimientos diarios en el periodo como en (2.2)

Ahora, calculemos el rendimiento de IBM en pesos usando cambios relativos:

- 1) Sean δ_{ibm}^t el cambio diario de IBM en dólares y $\delta_{mxn/usd}^t$ el cambio diario del peso contra el dólar, entonces $\delta_{ibm.mxn}^t = \delta_{ibm}^t * \delta_{mxn/usd}^t$ es el cambio diario de IBM en pesos.
- 2) Usando esto, guardemos en `dlyChg$IBM_MXN.Adjusted` la serie de cambios diarios de IBM en pesos.
- 3) Finalmente, acumulemos usando `prod` los cambios diarios para obtener el rendimiento en pesos que buscamos.

```
dlyChg$IBM_MXN.Adjusted <- dlyChg$IBM.Adjusted * dlyChg$MXN.X.Adjusted
apply(dlyChg, 2, prod)[c('IBM.Adjusted',
                        'IBM_MXN.Adjusted',
                        'MXN.X.Adjusted')] - 1
```

```
##      IBM.Adjusted  IBM_MXN.Adjusted  MXN.X.Adjusted
##      -0.04676966      -0.09664070      -0.05231793
```

Comparemos los rendimientos en pesos y dólares de IBM en 2017:

- El rendimiento de los dólares en pesos en 2017 fue de -5.23 %
- Hemos obtenido un rendimiento en pesos de IBM en 2017 de -9.66 %
- En contraste tenemos que el rendimiento en dólares de IBM fue -4.68 %.

¡Ojo! Un error común es pensar que si en el periodo \mathbb{T} vemos que IBM tuvo un rendimiento en dólares de $r_{ibm}^{\mathbb{T}}$ (-4.68 %) y el dólar tuvo un rendimiento en pesos de $r_{mxn/usd}^{\mathbb{T}}$ (-5.23 %), entonces el rendimiento en pesos de IBM es la suma de los dos $r_{ibm.mxn}^{\mathbb{T}} = r_{ibm}^{\mathbb{T}} + r_{mxn/usd}^{\mathbb{T}}$ (-9.91 %). Como ya vimos, hacer esto no es correcto.

Agregaremos a nuestra paquetería la función `get_dlyChg_from_price()`. El código de la función se encuentra en el anexo ???. De aquí en adelante usaremos esta función para obtener series de cambios diarios a partir de precios.

2.3.3. Rendimiento acumulado

Una serie de tiempo cuya entrada t es $\delta^{0,t}$ o $r^{0,t}$, es una serie de **rendimiento acumulado**.

Construyamos `cumRet`, un objeto `xts` con los rendimientos acumulados de los activos en `price`. Para esto, acumulamos los cambios diarios de `dlyChg` usando `cumprod()`.

```
cumRet <- dlyChg
for (c in 1:ncol(cumRet)) {
  cumRet[,c] <- cumprod(dlyChg[,c])
}
```

Ahora, grafiquemos los rendimientos acumulados en `cumRet`.

```
plot_xts(cumRet)
```

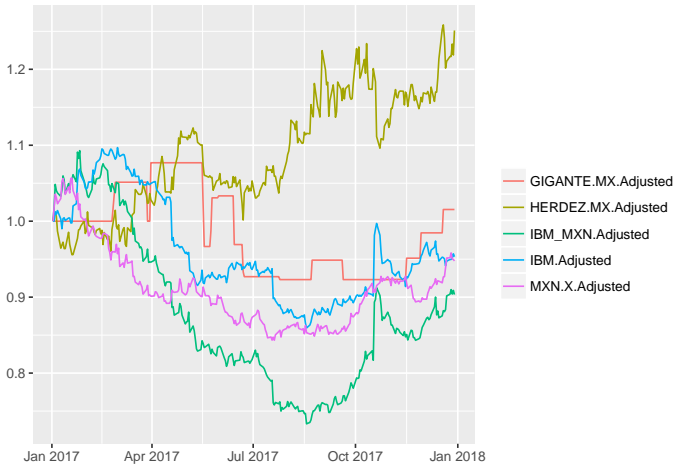


Figura 2.4: Precios de Acciones

¡La gráfica en la figura 2.4 es mucho más útil que la de la figura 2.2! Los rendimientos acumulados, a diferencia de los precios, nos permite comparar fácilmente cuánto cambió el valor de nuestros activos en el tiempo.

El equivalente a multiplicar cambios diarios es sumar el logaritmo de los cambios diarios y luego exponenciar.

$$\begin{aligned}\delta^{0,t} &= \exp(\ln(\delta^0) + \ln(\delta^1) + \cdots + \ln(\delta^t)) \\ &= \exp(\ln(1 + r^0) + \ln(1 + r^1) + \cdots + \ln(1 + r^t))\end{aligned}$$

Típicamente esta es la manera en la que se obtiene el rendimiento acumulado porque tiene ventajas de estabilidad numérica.

En el siguiente bloque de código, volvemos a generar `cumRet` pero esta vez construyendo el rendimiento acumulado de nuestros activos usando logaritmos.

```
cumRet <- dlyChg
for (c in 1:ncol(cumRet)) {
  cumRet[,c] <- exp(cumsum(log(cumRet[,c])))
}
```

Aunque hemos sido muy estrictos en la diferencia entre cambios relativos y rendimiento, como el rendimiento siempre es el cambio relativo menos uno, en la práctica no se hace una distinción clara y muchas veces se usan indistintamente.

Por eso, **a partir de aquí usaremos indistintamente todos los siguientes términos:** cambio relativo en el periodo \mathbb{T} , cambio relativo de 0 a T , rendimiento en el periodo \mathbb{T} , rendimiento de 0 a T y rendimiento acumulado en T .

Para obtener rendimientos acumulados a partir de cambios diarios, agregaremos a nuestra paquetería la función `get_cumRet_from_dlyChg()`. Esta función se encuentra en el anexo ??.

2.4. Precios Ajustados

Aunque hasta ahora hemos trabajado con precios de cierre ajustados, aún no hemos visto cuál es la diferencia entre los precios de cierre y los precios de cierre ajustados. Finalmente, ¡en esta sección mostraremos su diferencia!

Grafiquemos los rendimientos acumulados de los precios de cierre y ajustados de IBM.

```
# obtenemos los cambios diarios de ibm
ibm_dlyChg <- get_dlyChg_from_price(ibm)

# acumulamos los cambios diarios
ibm_cumRet <- get_cumRet_from_dlyChg(ibm_dlyChg)

plot_xts(ibm_cumRet[, c('IBM.Close', 'IBM.Adjusted')])
```

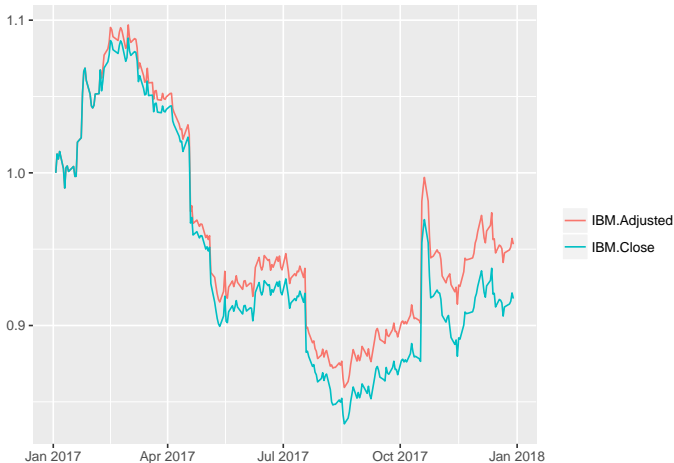


Figura 2.5: Precios de cierre y ajustados

En la figura 2.5 vemos que los rendimientos acumulados de los precios de cierre y ajustados se van separando en el tiempo.

Los **precios ajustados eliminan el efecto que tienen los dividendos en los precios de cierre de los instrumentos financieros**. En esta sección explicamos que significa esto.

Usemos la función `getDividends()` de **quantmod** para extraer de Yahoo! Finance los pagos de dividendos que tuvo la acción IBM en el 2017.

```
ibm_dividends <- getDividends('IBM',
                              from = '2016-12-31',
                              to = '2017-12-31')

ibm_dividends
```

```
##           IBM.div
## 2017-02-08      1.4
## 2017-05-08      1.5
## 2017-08-08      1.5
## 2017-11-09      1.5
```

`getDividends()` regresa para un instrumento financiero:

- las fechas en las que hubo pagos de dividendos
- el monto pagado por concepto de dividendos por cada título del instrumento

Como vemos, IBM hizo un pago de dividendos de 1.4 dólares por acción el 8 de febrero de 2017. Adicionalmente, en 2017 IBM hizo otros tres pagos de dividendos.

2.4.1. Dividendos

Veamos los precios de cierre y los pagos de dividendo de IBM entre el 7 y el 9 de febrero de 2017.

```
cbind(ibm[, 'IBM.Close'], ibm_dividends)['2017-02-07/2017-02-09', ]
```

```
##           IBM.Close IBM.div
## 2017-02-07      178.46      NA
## 2017-02-08      176.17      1.4
## 2017-02-09      177.21      NA
```

Si solo vemos el precio debajo de `IBM.Close`, podríamos pensar que un inversionista que tenía una acción de IBM tuvo una pérdida de -2.29 (176.17 - 178.46) dólares del 7 al 8 de febrero de 2017. Sin embargo, esto no es cierto porque el inversionista recibió 1.4 dólares como pago en dividendos.

Existen instituciones encargadas del resguardo central de la información de las transacciones de los instrumentos financieros. A este tipo de instituciones se les conoce como **custodios o depósitos de valores** (custodian agents). Los custodios también son los encargados de entregarle a los tenedores de instrumentos financieros los pagos de dividendos de sus instrumentos. En México, el Indeval (Instituto para el Depósito de Valores) es el depósito central de valores.

Después de cada pago de dividendos, los precios de cierre se ven afectados por los pagos de dividendos. Para evitar que el pago de dividendos se malinterprete como una pérdida de rendimiento, una **serie de precios ajustados es una serie de precios ‘hipotéticos’** cuyo valor representa una inversión:

- en la que se invierte en una sola acción, y
- en la que se reinvierten inmediatamente los dividendos pagados por esa misma acción (suponiendo que se pueden comprar fracciones de títulos de acciones).

Construcción de pesos ajustados

Supongamos que tenemos una serie de precios de cierre p^0, p^1, \dots, p^T de una acción xyz y que en d esta acción tuvo un pago de dividendos por un monto div^d .

Queremos construir una serie de precios $p_{\text{prima}}^0, p_{\text{prima}}^1, \dots, p_{\text{prima}}^d$ de una acción hipotética $\text{xyz}_{\text{prima}}$ que replique a xyz pero sin pagos de dividendos. Esto significa que deben cumplirse los siguientes tres supuestos:

- (1) $p_{\text{prima}}^d = p^d$
- (2) $p_{\text{prima}}^{d-1} = p^{d-1} - \text{div}^d$
- (3) $\delta_{\text{prima}}^t = \delta^t$ para toda t distinta a d

Combinando (2) y (3) llegamos a

$$\delta^{d-1} = \delta_{\text{prima}}^{d-1} \frac{p^{d-1}}{p^{d-2}} = \frac{p_{\text{prima}}^{d-1}}{p_{\text{prima}}^{d-2}} = \frac{p^{d-1} - \text{div}^d}{p_{\text{prima}}^{d-2}}$$

de donde obtenemos el valor de p_{prima}^{d-2}

$$p_{\text{prima}}^{d-2} = \left(\frac{p^{d-1} - \text{div}^d}{p^{d-1}} \right) p^{d-2}$$

Usando (3) y el resultado anterior, obtenemos el valor de p_{prima}^{d-3}

$$\delta^{d-2} = \delta_{\text{prima}}^{d-2} \frac{p^{d-2}}{p^{d-3}} = \frac{p_{\text{prima}}^{d-2}}{p_{\text{prima}}^{d-3}} = \frac{\left(\frac{p^{d-1} - \text{div}^d}{p^{d-1}} \right) p^{d-2}}{p_{\text{prima}}^{d-3}} = \left(\frac{p^{d-1} - \text{div}^d}{p^{d-1}} \right) p^{d-3}$$

Si continuamos con este proceso (con inducción), entonces se cumple para toda t distinta de d que

$$p_{\text{prima}}^t = \left(\frac{p^{d-1} - \text{div}^d}{p^{d-1}} \right) p^t$$

Así $\left(\frac{p^{d-1} - \text{div}^d}{p^{d-1}} \right)$ es el **factor de ajuste** del precio al tiempo d , y p_{prima}^t es el precio ajustado de p^t .

Ya conocemos la diferencia de precios de cierre y precios de cierre ajustados. De aquí en adelante solo usaremos precios ajustados.

2.5. Valor de un portafolio

Para la construcción de portafolios en este trabajo **vamos a suponer que podemos comprar fracciones de títulos de acciones**. Para ejemplificar, supongamos que tenemos 100 pesos y que hay una acción que cuesta 20 pesos. Si queremos invertir 25 % de nuestro capital en esa acción, entonces compraríamos 1.25 títulos.

Normalmente, no son fáciles de conseguir en el mercado fracciones de títulos de instrumentos listados en casas de bolsa. En contraste, es fácil invertir en fracciones de tipos de cambio.

Imaginemos que al inicio del 2017 invertimos en un portafolio con la siguiente composición:

- 30 % en Herdez,
- 20 % en Gigante,
- 40 % en dólares, y
- 10 % en IBM

Estos pesos los guardamos en H_0 (initial holdings).

```
H0 <- c(
  HERDEZ.MX.Adjusted = 0.3,
  GIGANTE.MX.Adjusted = 0.2,
  MXN.X.Adjusted = 0.4,
  IBM_MXN.Adjusted = 0.1
)
```

Quisiéramos conocer el rendimiento acumulado en 2017 de este portafolio.

Construcción de rendimiento acumulado de un portafolio

Supongamos que tenemos un portafolio con n activos y el **peso inicial de apertura de las componentes del portafolio** está dado por $H^0 = (H_1^0, H_2^0, \dots, H_n^0)$ con $\sum H_i^0 = 1$. Con esto, tenemos que

- El **rendimiento acumulado del portafolio al tiempo t** es

$$\sum_j H_j^0 \delta_j^{0,t}$$

donde $\delta_j^{0,t}$ es el rendimiento acumulado del activo j al tiempo t .

- La **contribución del activo i al rendimiento acumulado del portafolio al tiempo t** está dada por

$$H_i^0 \delta_i^{0,t} + (1 - H_i^0)$$

- El **peso de cierre del portafolio invertido en el activo i al tiempo t** denotado por h_i^t es:

$$h_i^t = \frac{H_i^0 \delta_i^{0,t}}{\sum_j H_j^0 \delta_j^{0,t}}$$

Regularmente nos referiremos a los pesos de cierre de un portafolio simplemente como pesos del portafolio.

Calculemos el rendimiento acumulado de nuestro portafolio, así como las contribuciones al rendimiento acumulado y los pesos en cada t de los componentes.

```
cumRet <- cumRet[, c('HERDEZ.MX.Adjusted',
                     'GIGANTE.MX.Adjusted',
                     'MXN.X.Adjusted',
                     'IBM_MXN.Adjusted')]

portfolio_cumterm <- xts(order.by = index(cumRet))

portfolio_cumRet <- xts(order.by = index(cumRet))
portfolio_cumRet$portfolio <- 0

portfolio_contrib <- xts(order.by = index(cumRet))

for(c in 1:ncol(cumRet)) {
  portfolio_cumterm <- cbind(portfolio_cumterm,
                             H0[c] * cumRet[, c])
}
```

```

portfolio_cumRet$portfolio <-
  portfolio_cumRet$portfolio + portfolio_cumterm[,c]

portfolio_contrib <- cbind(portfolio_contrib,
                           portfolio_cumterm[,c] + (1 - H0[c]))
}

portfolio_weights <-
  portfolio_cumterm / as.numeric(portfolio_cumRet$portfolio)

```

Ahora, veamos las contribuciones y el rendimiento acumulado que acabamos de calcular.

```
plot_xts(cbind(portfolio_cumRet, portfolio_contrib))
```

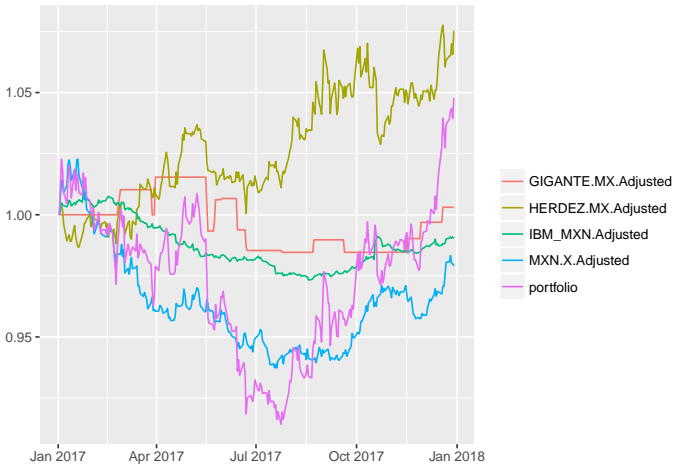


Figura 2.6: Contribuciones al rendimiento de un portafolio

En la figura 2.6 podemos ver que al final del año el portafolio tuvo un rendimiento de 4.78 % y que la contribución de cada componente a este rendimiento es:

- 7.53 % de Herdez
- 0.31 % de Gigante
- -2.09 % de dólares
- -0.97 % de IBM (en pesos)

Aunque asignamos un peso inicial (de apertura) a cada activo dentro del portafolio, **en la medida en la que se mueven los precios de los componentes los pesos de cierre se van moviendo.**

Grafiquemos los pesos de cierre de cada componente del portafolio en el tiempo.

```
plot_xts(portfolio_weights)
```

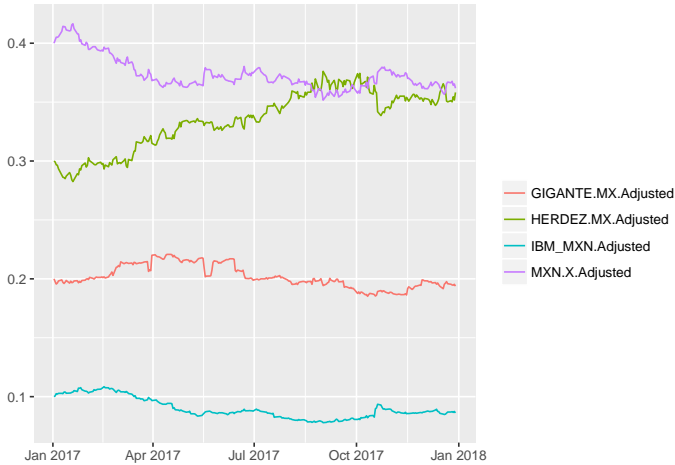


Figura 2.7: Pesos de cierre de los componentes de un portafolio

Veamos en la figura 2.7 las diferencias entre los pesos iniciales de las componentes del portafolio y sus pesos al 30 de junio de 2017:

- Herdez pasó de 30.00 % a 33.88 %
- Gigante pasó de 20.00 % a 19.88 %
- Los dólares pasaron de 40.00 % a 37.47 %
- IBM (en pesos) pasó de 10.00 % a 8.77 %

2.5.1. Posiciones Largas y Posiciones Cortas

En este trabajo únicamente consideraremos pesos iniciales positivos, que por construcción nos llevan a pesos de cierre positivos. A las posiciones $H_i^0 > 0$ se les llama **posiciones largas**.

Existen casas de bolsa que permiten colocar órdenes llamadas **ventas en corto**. Supongamos que a principio de 2017 vamos a registrar una venta en corto de Herdez a precio de mercado. Esto significa que si el rendimiento de Herdez al final de 2017 fue de 25.11 % y si decidimos cerrar la posición corta en ese momento, entonces al final del periodo obtendríamos un rendimiento de -25.11 %.

A las posiciones $H_i^0 < 0$ se les llama **posiciones cortas**. Tener una posición corta en el componente i de un portafolio significa colocar una venta en

corto en el activo i . Si ese activo da un rendimiento $\delta_i^{\mathbb{T}}$ en el periodo \mathbb{T} , entonces la contribución del activo i al tiempo T será $(-H_i^0 \delta_i^{\mathbb{T}})(1 - H_i^0)$

2.6. Rebalanceo y costos de transacción

Se dice que se está **rebalanceado** un portafolio cuando en una fecha se decide cambiar los pesos de los componentes del portafolio a un nivel específico. Para esto, se tienen que realizar transacciones de compra venta. Las casas de bolsa cobran **costos de transacción** que son comisiones por realizar estas operaciones.

En la vida real se tienen que considerar los costos de transacción para determinar el desempeño de una inversión. Además las operaciones se cierran a precios de mercado (intradía) y no necesariamente a precio de cierre. Es posible que haya operaciones de compra venta que no logran ejecutarse por temas operativos o de liquidez.

En este trabajo vamos a suponer que siempre que deseamos modificar los porcentajes de los componente de un portafolio:

- No hay costos de transacción.
- Se logran cerrar todas las operaciones de compra venta.
- Estas operaciones se efectúan a precio de cierre y al día siguiente se amanece con la nueva composición.

El 30 de junio del 2017 decidimos hacer un rebalanceo de nuestro portafolio para llevar los pesos de los componentes a los siguientes niveles:

- 25 % en Herdez,
- 20 % en Gigante,
- 55 % en dólares, y
- 0 % en IBM

Estos pesos los guardamos en Hb.

```
Hb <- c(
  HERDEZ.MX.Adjusted = 0.25,
  GIGANTE.MX.Adjusted = 0.2,
  MXN.X.Adjusted = 0.55,
  IBM_MXN.Adjusted = 0.0
)
```

Con este cambio, ¿cuál es el rendimiento acumulado en 2017?

Construcción de rendimiento acumulado de un portafolio después de un rebalanceo

Si el día b se realiza un rebalanceo, asumimos que ese día se vende a precio de cierre la posición con pesos $h^b = (h_1^b, h_2^b, \dots, h_n^b)$ y se compra a ese mismo precio la posición con **pesos de rebalanceo** $H^b = (H_1^b, H_2^b, \dots, H_n^b)$ tales que $\sum H_i^b = 1$. De esta manera el día $b + 1$ los pesos de apertura de los componentes del portafolio quedan en H^b .

Dado este rebalanceo y bajo estos supuestos tenemos que, para $t \leq b$ el rendimiento acumulado, las contribuciones y pesos se calculan igual que antes. En el caso de $t > b$ se cumple que:

- El rendimiento acumulado esta dado por

$$\sum_j H_j^0 \delta_j^{0,b} \sum_k H_k^b \delta_j^{b,t}$$

- La contribución del activo i al rendimiento acumulado del portafolio es

$$\left(H_i^0 \delta_i^{0,b} + (1 - H_i^0) \right) \left(H_i^b \delta_i^{b,t} + (1 - H_i^b) \right)$$

- El peso de cierre h_i^t del activo i del portafolio está dado por

$$h_i^t = \frac{\left(\sum_j H_j^0 \delta_j^{0,b} \right) \left(H_i^b \delta_i^{b,t} \right)}{\left(\sum_j H_j^0 \delta_j^{0,b} \right) \left(\sum_j H_j^b \delta_j^{b,t} \right)} = \frac{H_i^b \delta_i^{b,t}}{\sum_j H_j^b \delta_j^{b,t}}$$

Para obtener estos valores nos será útil tomar en cuenta que

$$\delta_j^{b,t} = \frac{\delta_j^{0,t}}{\delta_j^{0,b}}$$

El anexo ?? incluye en la función `get_rebalance()` que usaremos a continuación, este procedimiento se describe una lógica secuencial para obtener valores de portafolios con múltiples fechas de rebalanceo.

Considerando los pesos de rebalanceo del 30 de junio, vamos a obtener el nuevo rendimiento acumulado de nuestro portafolio y las contribuciones de los componentes.

```
dlyChg <- dlyChg[, c('HERDEZ.MX.Adjusted',
                     'GIGANTE.MX.Adjusted',
                     'MXN.X.Adjusted',
                     'IBM_MXN.Adjusted')]

rebWeight <- xts(rbind(H0, Hb),
                 order.by = c(index(cumRet)[1], as.Date('2017-06-30')))
```

```
rebalance <- get_rebalance(dlyChg, rebWeight)
plot_xts(cbind(rebalance$portValue, rebalance$portContrib))
```

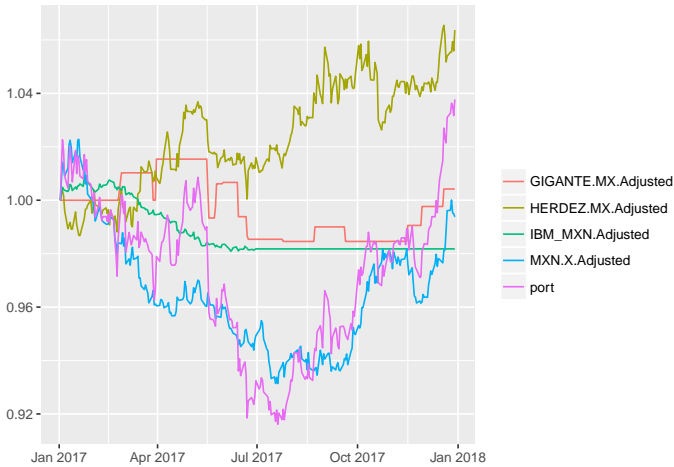


Figura 2.8: Contribuciones al rendimiento después del rebalanceo

De acuerdo a la figura 2.8:

- El rebalanceo que se hizo el 30 de junio provocó que disminuyera el rendimiento final del portafolio. Este rendimiento fue de 3.77 %.
- Resalta que después del rebalanceo, al fijar el peso de IBM en cero, a partir del 30 de junio su contribución se mantiene fija en -1.82 %.
- Por otro lado, los pesos de cierre del portafolio antes y después del rebalanceo del 30 de junio también muestran cambios como vemos en la siguiente figura 2.9.

```
plot_xts(rebalance$portWeight)
```

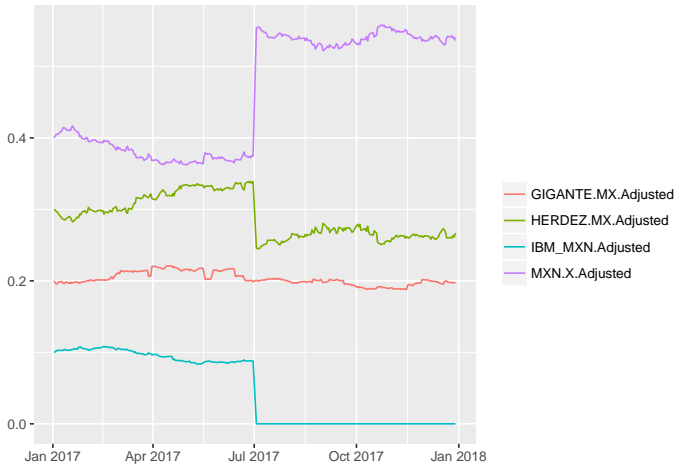


Figura 2.9: Pesos de cierre despues del rebalanceo

Para finalizar esta sección, definiremos el **valor de un portafolio en t** . Este es simplemente el rendimiento acumulado en t por un valor inicial V_0 que puede ser un precio, una cantidad de dinero o un nivel de puntos arbitrario.

Así, si el valor inicial de nuestro portafolio rebalanceado es 100 pesos, entonces el valor de este portafolio al 29 de diciembre de 2017 es

```
V_0 = 100
V_0 * rebalance$portValue['2017-12-29']
```

```
## port
## 2017-12-29 103.7705
```

2.7. Índices

Imaginemos que queremos un indicador que promedie el precio de muchas acciones de un sector de la industria, o un indicador que represente los bonos de un país; también, podemos imaginar un indicador de los movimientos de todos los tipos de cambio de países latino americanos. Los **índices de mercado** son portafolios de instrumentos financieros que permiten entender el comportamiento agregado de sus componentes.

2.7.1. Proveedores de Índices

Los primeros índices fueron creados y publicados por periódicos financieros para resumir las fluctuaciones diarias de los precios de las acciones. El precursor del *Wall Street Journal* llamado *Dow Jones and Co's Customer's Afternoon letter* creó en 1884 al **Dow Jones**, este índice sigue siendo una de las principales referencias para los inversionistas americanos junto con el **S&P500**.

Hoy existen empresas como *Standard & Poor's Financial Services LLC* (S&P) y *Morgan Stanley Capital International* (MSCI) que son **proveedores de índices**. Estas empresas desarrollan metodologías propietarias para definir sus índices.

Algunos índices de acciones famosos son:

- S&P500 - índice con 500 acciones de Estados Unidos creado por Standard and Poors (S&P)
- Dow Jones - uno de los primeros índices del mundo constituido por acciones de Estados Unidos
- All County World Index - índice de referencia compuesto por acciones de todo el mundo creado por MSCI
- IPC (Índice de Precios y Cotizaciones) - índice de acciones mexicanas creado por la Bolsa Mexicana de Valores

También existen índices de bonos y de commodities:

- Barclays Capital U.S. Aggregate Bond Index - índice compuesto por bonos del tesoro americanos
- S&P GSCI Crude Oil Index - índice cuyo valor se basa en la producción de petróleo

Por lo regular las **metodologías para construir índices** de acciones y bonos consisten en:

1. Definir una regla de selección de instrumentos líquidos. Éstos determinan la composición del índice (index composition)
2. Asignar pesos de rebalanceo a cada componente del índice. Los tipos de índices más comunes son:
 - **índices con pesos de capitalización de mercado** (market-cap weighted indices) - al rebalancearse este tipo de índices, el peso que se asigna a cada componente es su *capitalización de mercado*
 - **índices con pesos iguales** (equally weighted indices) - índices que al rebalancearse asignan un mismo peso a todos sus componentes
3. Determinar la frecuencia de los rebalanceos del índice. Regularmente los índices se rebalancean semestralmente o trimestralmente

2.7.1.1. Capitalización de mercado

La **capitalización de mercado al tiempo t** (market capitalization o market cap) es simplemente el precio de una acción emitida por el número de títulos de la emisión al tiempo t . De esta manera, el market cap representa el valor total del mercado en t de una acción.

En la figura 2.10 tenemos un mapa que muestra la capitalización de mercado en miles de millones de dólares para cada país en el mundo en 2015.



Figura 2.10: Mapa del mundo por capitalización de mercado

Estados Unidos, con un market cap de 19.8 billones de dólares, es el más grande y representa el 52 % del valor del mercado en el mundo. Japón es el segundo lugar con \$3 billones, seguido por UK en \$2.7 billones.

Para obtener la capitalización de mercado de una emisión necesitamos su precio y número de títulos. Ya sabemos como extraer precios. Una manera de consultar el número de títulos de un instrumento financiero es usar las estadísticas de Yahoo! Finance.

La figura 2.11 muestra una captura de pantalla de las estadísticas de IBM en <https://finance.yahoo.com/quote/ibm/key-statistics>. **El número total de títulos de IBM está en amarillo.**

Share Statistics	
Avg Vol (3 month) ³	5.22M
Avg Vol (10 day) ³	4.63M
Shares Outstanding ⁵	925.79M
Float	888.06M
% Held by Insiders ¹	0.09%
% Held by Institutions ¹	59.71%
Shares Short ³	16.84M
Short Ratio ³	2.71
Short % of Float ³	1.79%
Shares Short (prior month) ³	15.32M

Figura 2.11: Estadísticas de IBM

Con Herdez y Gigante construyamos un *market-cap weighted index*. Para eso buscamos el número total de títulos de las dos acciones que se muestran aquí.

Títulos	
Herdez	428.91 mill
Gigante	817.49 mill

Con esto, vamos a obtener para cada inicio de trimestre en 2017 la capitalización de mercado de nuestras acciones y el valor total del mercado (la suma de las capitalizaciones de estas acciones).

```
t <- c(1, floor(nrow(price) * c(1/4, 1/2, 3/4)))
marketCap <- xts(order.by = index(price)[t])

marketCap$HERDEZ.MX <- price$HERDEZ.MX.Adjusted[t] * 428910000
marketCap$GIGANTE.MX <- price$GIGANTE.MX.Adjusted[t] * 817490000
marketCap$total_mkt <- marketCap$HERDEZ.MX + marketCap$GIGANTE.MX

marketCap
```

```
##                HERDEZ.MX  GIGANTE.MX  total_mkt
## 2017-01-02 14864320401 31882110000 46746430401
## 2017-03-31 15282701947 34334580000 49617281947
## 2017-06-30 15652913040 29552265135 45205178175
## 2017-09-29 17568131726 29429640000 46997771726
```

Ahora, obtengamos para cada inicio de trimestre los pesos de rebalanceo de las acciones en nuestro índice.

```
rebWeight <- xts(order.by = index(price)[t])
rebWeight$HERDEZ.MX <- marketCap$HERDEZ.MX / marketCap$total_mkt
rebWeight$GIGANTE.MX <- marketCap$GIGANTE.MX / marketCap$total_mkt

rebWeight
```

```
##              HERDEZ.MX GIGANTE.MX
## 2017-01-02 0.3179777 0.6820223
## 2017-03-31 0.3080117 0.6919883
## 2017-06-30 0.3462637 0.6537363
## 2017-09-29 0.3738078 0.6261922
```

Calcular el valor diario de un índice se hace de la misma manera que calcular el valor de un portafolio. De esta manera usamos los pesos de rebalanceo en `rebWeight`, los cambios diarios en `dlyChg` y un valor inicial de 1000 para obtener el valor diario de nuestro índice.

De esta manera el valor al final del 2017 del índice de capitalización de mercado de Herdez y Gigante está dado por

```
dlyChg <- dlyChg[, c('HERDEZ.MX.Adjusted',
                     'GIGANTE.MX.Adjusted')]

rebalance <- get_rebalance(dlyChg, rebWeight)

1000 * rebalance$portValue['2017-12-29']
```

```
##              port
## 2017-12-29 1090.333
```

Los pesos de capitalización de mercado de las acciones en nuestro índice se ven así:

```
plot_xts(rebalance$portWeight)
```

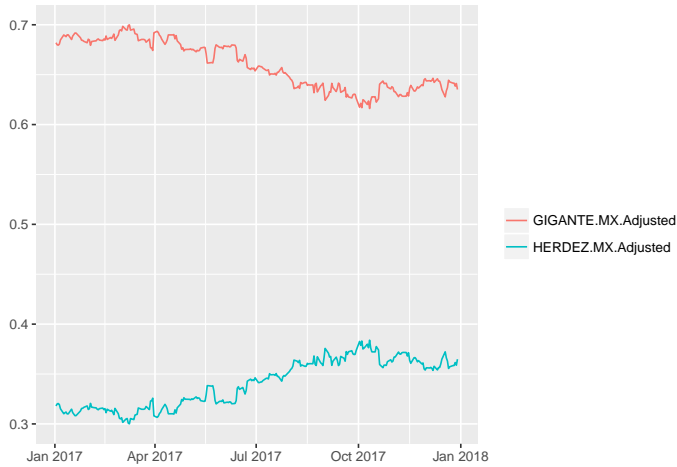


Figura 2.12: Pesos de capitalización de mercado de un índice hipotético de dos acciones

De la figura 2.12 concluimos que Gigante perdió valor de mercado contra Herdez durante 2017.

2.8. ETFs

Se le llama **Exchange Traded Fund (ETF)** a un fondo que replica el valor de algún índice o activo (puede no ser operado en bolsa, como el oro). Una de sus cualidades es que los ETF's son listados en bolsa y pueden ser operados como cualquier acción.

Recordemos que un índice es un conjunto de acciones. Si quisiéramos replicar el valor de un índice necesitaríamos el capital necesario para comprar todos los activos en el índice con su peso correspondiente. Para eso, necesitaríamos mucho capital. De igual manera, si quisiéramos aumentar o disminuir nuestra posición en el índice sería muy complicado porque tendríamos que comprar o vender todos los componentes con consecuencias de costos de transacción y operativos.

Por otro lado, si quisiéramos invertir en el valor del oro tendríamos que conseguir lingotes de oro y no podríamos comprar fracciones de lingotes. Además al momento de vender, podría ser complicado encontrar alguien interesado en esos lingotes.

Los ETFs se han vuelto muy populares porque son muy fáciles de operar y permiten tener exposición a distintas fuentes de valor. **A partir del 2013, éstos son los instrumentos más operados globalmente; de hecho, el instrumento más operado en el mundo**

es el **SPY**. Este ETF lo ofrece *State Street Global Advisors* y replica el valor del S&P 500. Guardemos en `spy` los precios de este ETF.

```
spy <- get_prices_yahoo(c('spy'),
                        from = '2012-12-31',
                        to = '2017-12-31')
```

Para conocer los valores en los que se encuentra invertido un ETF es necesario encontrar el **prospecto de información del ETF**. Para el caso de ETFs de índices, el prospecto indica el nombre del índice que busca replicar. También resulta muy útil visitar la página <https://www.etf.com/> para encontrar los específicos de un ETF particular.

En los siguientes capítulos vamos a utilizar varios ETFs para nuestros ejemplos. A continuación listamos los ETFs que usaremos y los índices que replican.

ETFs de Índices Sectoriales

Existen metodologías propietarias que se usan para clasificar acciones en distintos grupos. En particular es común que se busque clasificar las empresas de acuerdo al sector industrial al que pertenecen. Este problema no es necesariamente sencillo pues existen empresas cuyos productos y servicios se encuentran muy diversificados. Un **índice sectorial** está compuesto por acciones que pertenecen a un sector industrial específico.

- IYC - ETF que replica al Dow Jones US Consumer Services Index. Este índice contiene emisiones en US del sector de servicios al consumidor
- IYE - Dow Jones US Energy Index - sector energético
- IYH - Dow Jones US Health Care Index - sector de salud
- IYR - Dow Jones US Real State Index - sector de bienes raíces
- IYW - Dow Jones US Technology Index - sector de tecnología

Guardamos en `sectors` los precios de los ETFs de Índices Sectoriales.

```
sectors <- get_prices_yahoo(c('IYC',
                              'IYE',
                              'IYH',
                              'IYR',
                              'IYW'),
                           from = '2012-12-31',
                           to = '2017-12-31')
```

ETFs de Bonos del Tesoro Americanos

- IEF - Barclays U.S. 7-10 Year Treasury Bond Index - bonos del Tesoro de US con vencimiento entre 7 y 10 años.
- TLT - Barclays 20+ Yr Treasury Bond Index - bonos del Tesoro de US con vencimiento entre 20 y 30 años.

- TLH - Barclays 10-20 Yr Treasury Bond Index - bonos del Tesoro de US con vencimiento entre 10 y 20 años.
- SHY - Barclays U.S. 7-10 Year Treasury Bond Index - bonos del Tesoro de US con vencimiento entre 1 y 3 años.

```
bonds <- get_prices_yahoo(c('TLT',
                             'TLH',
                             'IEF',
                             'SHY'),
                          from = '2012-12-31',
                          to = '2017-12-31')
```

ETFs para estrategias Multi-Activas

La comunidad financiera se refiere por **multi-activos** a aquellas estrategias de inversión o portafolios compuestos por activos que tienen distintos tipos de fuentes de valor, digamos:

- 1) bonos - Los bonos determinan su valor de acuerdo a movimientos de tasa, que reflejan las expectativas de la calidad del emisor del bono y del valor de una moneda en el tiempo.
 - 2) acciones - Las acciones determinan su valor de acuerdo a las expectativas de flujos futuros que generará una empresa.
 - 3) commodities - El valor de un commodity proviene de la demanda y oferta de ese bien material.
 - 4) divisas- Las fortaleza de una moneda proviene de su demanda y su demanda depende de las expectativas de confianza y crecimiento de la economía que utiliza esa moneda.
- ACWI - MSCI's All Country World Index - índice de capitalización de mercado de acciones listadas en todo el mundo
 - EEM - Emerging Markets Index - índice de capitalización de mercado de acciones listadas en países emergentes
 - USO - Front Month Light Sweet Crude Oil Index - fondo compuesto por futuros de precios de petróleo
 - GLD - Precio del oro - fondo invertido directamente en oro físico
 - UUP - Deutsche Bank Long USD Currency Portfolio Index - índice que representa la fortaleza del dólar americano relativo a una canasta de seis monedas: el euro, el yen, la libra, el dólar canadiense, la corona sueca y el franco suizo.

```
multi <- get_prices_yahoo(c('acwi',
                             'eem',
                             'uso',
                             'tlt',
                             'ief',
                             'gld',
```

```
        'uup'),  
        from = '2008-12-31',  
        to = '2018-06-30')
```

Extrajimos los precios de los ETFs listados arriba para constuir los objetos `spy`, `sectors`, `bonds` y `multi`. Los precios en estos objetos `xts` los utilizamos en los siguientes capítulos.

En muchos artículos de manejo de portafolios se utilizan índices para probar resultados. Nosotros usaremos ETF's en nuestros ejemplos. Le recordamos al lector que es equivalente desde el punto de vista teórico utilizar índices o ETFs. Sin embargo, el precio de los ETFs está en una moneda mientras que el valor de los índices se encuentra en un valor arbitrario; y como se explicó antes, es posible invertir en ETFs mientras que en índices no es posible.

Bibliografía

- Eddelbuettel, D. (2018). *Rcpp: Seamless R and C++ Integration*. R package version 1.0.0.
- Gorman, B. (2018). *mltools: Machine Learning Tools*. R package version 0.3.5.
- Johnson, S. G. (2018). *nloptr: The NLOpt nonlinear-optimization package*. R package version 0.3.14.
- Kolm, P., Tutuncu, R., and Fabozzi, F. (2014). 60 years of portfolio optimization: Practical challenges and current trends. *European Journal of Operational Research*, 234.
- Maillard, S., Roncalli, T., and Teiletche, J. (2010). On the properties of equally-weighted risk contributions portfolios. *The Journal of Portfolio Management*, 36.
- R Core Team (2018). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.
- Roncalli, T. (2013). *Introduction to Risk Parity and Budgeting*. CRC Press.
- Ryan, J. A. and Ulrich, J. M. (2018a). *quantmod: Quantitative Financial Modelling Framework*. R package version 0.4-13.
- Ryan, J. A. and Ulrich, J. M. (2018b). *xts: eXtensible Time Series*. R package version 0.11-0.
- Wickham, H. (2017). *tidyverse: Easily Install and Load the 'Tidyverse'*. R package version 1.2.1.
- Zeileis, A. and Grothendieck, G. (2018). *zoo: S3 Infrastructure for Regular and Irregular Time Series*. R package version 1.8-4.