

Tidyquant

Trasladando el análisis financiero al Tidyverse

Gabriel Cabrera
Universidad de Chile



GaboCg



@GaboC_g



gcabrerag@fen.uchile.cl

"The most valuable commodity I know of is information."

Gordon Gekko, *Wall Street* (1987).

Breve historia

Paquete Tidyquant



1. Un **megapaquete** creado por Matt Dancho (@mdancho84).
2. Es la base del libro *"Reproducible Finance with R: Code Flows and Shiny Apps for Portfolio Analysis"* de Jonathan K. Regenstein Jr. (@jkregenstein).
3. **tidyquant** integra los mejores recursos/paquetes para coleccionar y analizar datos financieros: zoo, xts, quantmod, TTR y PerformanceAnalytics, con la infraestructura **tidy data**¹ del tidyverse, permitiendo la interacción entre ambos.

? Actividad. Instale y descargue **tidyquant**.

```
install.packages("tidyquant")  
library("tidyquant")
```

[1] Para mayor detalle ver el paper "Tidy Data (2014)" de Hadley Wickham.

Resumen de los paquetes

1. **xts** o **eXtensible time series**: Es una estructura de datos y a la vez un paquete para manipular series de tiempo. Detrás se encuentra la estructura zoo.

2. **quantmod** o **Quantitative Financial Modelling & Trading Framework**: Es un paquete diseñado para recuperar, manipular y modelar datos cuantitativos financieros.

3. **TTR** o **Technical Trading Rules**: Paquete que incluye varias funciones para computar análisis técnico.

4. **PerformanceAnalytics**: Incluye una colección de funciones econométricas para desempeño y análisis de riesgo. Se necesita los retornos y no los precios.

Lo

Tres Universos

xts

llena
las fechas
en
las
files.

Time Series

TIBBLE

En el mundo `xts` debemos considerar:

1. `xts` es un paquete y un tipo de objeto.
2. Contiene series de tiempo, ubicando la fecha en el `index` de cada observación.
3. Los paquetes cruciales para trabajar en el mundo del `xts` son: `quantmod` y `PerformanceAnalytics`. `quantmod` permite acceder a internet y extraer los precios en un objeto `xts`. `PerformanceAnalytics`, contiene funciones útiles para analizar el *performance* de portafolios: `StdDev()`, `SharpeRatio()`, `SortinoRatio()` y `CAPM.beta()`.

TAREA

? Actividad. Descargue el Standard & Poor's 500 (`^GSPC`) utilizando `quantmod` desde 2018-01-01 hasta 2018-12-01.

```
library("quantmod")

getSymbols("^GSPC",
           from = "2018-01-01",
           to = "2018-12-01",
           periodicity = "daily")
```

En la consola:

`tx-get-options()`

tidyverse

En el mundo `tidyverse` debemos considerar:

1. No es específico para análisis financiero ni su propósito esta pensado en trabajar con series de tiempo.
2. Usar `tidyverse` implicaría un uso intensivo de `dplyr` para transformar y organizar nuestros datos.
3. En este mundo, nuestros datos estarán almacenados en **data frames** o **tibble**.

tidyquant

En el mundo `tidyquant` debemos considerar:

1. Tendremos lo mejor de ambos mundos, integrando las opciones financieras de `PerformanceAnalytics` con la transformación/manipulación del `tidyverse`.
2. Se suman dos paquetes: `timetk` y `tibbletime`.

? Actividad. Descargue el Standard & Poor's 500 (^GSPC) desde 2018-01-01 hasta 2018-02-01 utilizando `tidyquant`.

```
sp500_tbl <- tq_get("^GSPC",  
  from = "2018-01-01",  
  to = "2018-12-01",  
  periodicity = "daily")
```

Workflow con tidyquant

Importación

La extracción de datos depende del **tipo** y de la **fuentes**. Las funciones relevantes son:

1. Obtener **Stock Index**, `tq_index()` y **Stock Exchange**, `tq_exchange()`: como resultado tendremos los **tickers** o *stock symbols* con sus atributos. Existen 18 índices y 3 *exchanges*.

- `tq_index_options()`
- `tq_exchange_options()`

2. Obtener datos cuantitativos, `tq_get()`: Obtenemos datos desde distintas fuentes online.

- `tq_get_options()`

El más usado es `stock.prices`, que descarga directamente desde **yahoo finance** y `economic.data` desde **FRED** Federal Reserve of Saint Louis.

.

? Actividad. Seleccione las acciones diarias de Facebook (FB), Apple (AAPL), Netflix (NFLX) y Google (GOOG), desde Enero del 2013 hasta Diciembre del 2018

```
symbols <- c("FB", "AAPL", "NFLX", "GOOG")  
  
stocks_daily_tbl <- tq_get(symbols,  
  get = "stock.prices",  
  from = "2013-01-01",  
  to = "2019-01-01",  
  periodicity = "daily")
```

? Actividad. Descargue la tasa libre de riesgo mensual del bono de Estados Unidos a 10 años (DGS10), desde Enero del 2010 hasta Diciembre del 2018

```
tbill_daily_tbl <- tq_get("DGS10",  
  get = "economic.data",  
  from = "2013-01-01",  
  to = "2019-01-01",  
  periodicity = "daily")
```

get_symbols → XTS.

tq_get → tibble


mutate

vs

transmute

Transformación

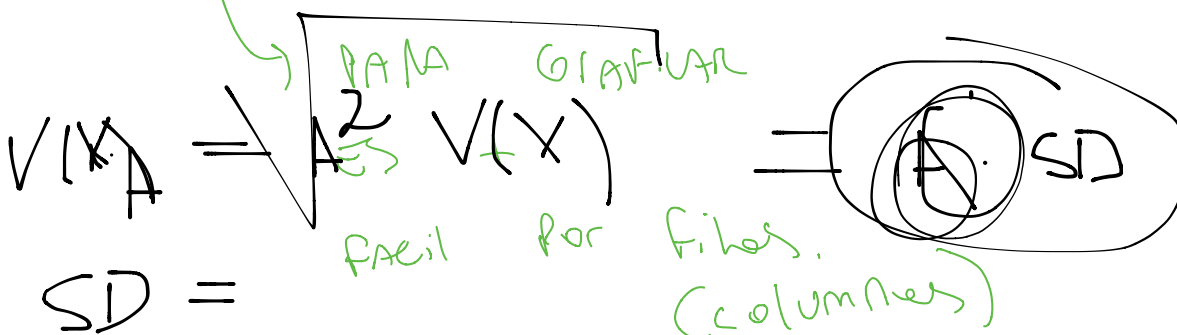
1. Para efecto del curso **la transformación** será pasar datos diarios a mensual/anual.
2. Luego calcular el retorno y retorno acumulado.

 **Ejemplo.** Deberíamos obtener el mismo resultado, si pasamos nuestros datos diarios a mensual/anual o si los descargamos directamente mensual/anual, como veremos a continuación.

Utilizando el objeto `stocks_tbl` que contiene en un `tibble` los precios de FANG, seleccionamos solo la observación final dentro de cada mes (`lastof`).

```
stocks_daily_to_monthly_tbl <- stocks_daily_tbl %>%  
  select(date, symbol, close) %>%  
  group_by(symbol) %>%  
  tq_transmute(mutate_fun = to.monthly,  
                indexAt = "lastof") %>%  
  mutate(date = as.Date(as.yearmon(date))) %>%  
  select(date, symbol, close) %>%  
  spread(symbol, close)
```

↳ wide a longer → longer a wide.



Obteniendo el siguiente resultado.

Show entries

Search:

	Fecha	FB	AAPL	NFLX	GOOG
1	2013-01-01	65.07	30.98	376.43	23.61
2	2013-02-01	63.06	27.25	399.10	26.87
3	2013-03-01	63.24	25.58	395.61	27.04
4	2013-04-01	63.25	27.77	410.75	30.87
5	2013-05-01	64.25	24.35	433.98	32.32

Showing 1 to 5 of 72 entries

Previous

1

2

3

4

5

...

15

Next

mutate_each (funs(), A:Z)

Summarise_each ()

¡lo mismo!

TARLA → DPLYR:
- select
- mutate
- arrange
- filter
- arrange
→ rename
mutate
each

Utilizando la opción **monthly** y realizando alguna manipulación obtenemos el mismo resultado.

```
stocks_monthly_tbl <- tq_get(symbols,
  get = "stock.prices",
  from = "2013-01-01",
  to = "2019-01-01",
  periodicity = "monthly") %>%
  select(date, symbol, close) %>%
  spread(symbol, close)
```

Show entries

Search:

	Fecha	FB	AAPL	NFLX	GOOG
1	2013-01-01	65.07	30.98	376.43	23.61
2	2013-02-01	63.06	27.25	399.10	26.87
3	2013-03-01	63.24	25.58	395.61	27.04
4	2013-04-01	63.25	27.77	410.75	30.87

Showing 1 to 4 of 72 entries

Previous

1

2

3

4

5

...

18

Next

Transformación: Retornos

Para calcular el retorno logarítmico se debe aplicar la formula:

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

? Actividad. Calcule el retorno logarítmico mensual para FANG

```
stock_returns_monthly_tbl <- stocks_monthly_tbl %>%  
  gather(symbols, close, -date) %>%  
  group_by(symbols) %>%  
  tq_transmute(mutate_fun = periodReturn,  
               period = "monthly",  
               type = "log",  
               col_rename = "returns_monthly",  
               ) %>%  
  slice(-1) %>%  
  mutate(date = as.Date(as.yearmon(date))) %>%  
  spread(symbols, returns_monthly)
```

Handwritten notes:
- A green arrow points from "slice(-1)" to "no. omit", indicating that the last row is not omitted.
- The word "filter" is written in green next to "no. omit".

Transformación: Retornos acumulados



En el caso de querer el retorno acumulado, usamos la función `cumsum()`. *✓ Gather vs spread*

? Actividad. Calcule el retorno logarítmico acumulado de FANG

```
stock_returns_acum_monthly_tbl <- stock_returns_monthly_tbl %  
gather(symbol, returns, -date) %>%  
group_by(symbol) %>%  
mutate(returns_acum = cumsum(returns)) %>%  
select(-returns) %>%  
spread(symbol, returns_acum)
```

Show entries

Search:

	Fecha	FB	AAPL	NFLX	GOOG
1	2013-02-01	-0.0314	-0.1283	0.0585	0.1295

Showing 1 to 1 of 71 entries

Previous

1

2

3

4

5

...

71

Next

Visualización

1. Para graficar datos financieros:

- **highcharter**: Cuando los datos sean **xts** y queramos que sean interactivo. Desde tibble a **xts** se usa `tk_xts()` del paquete `timetk`.
- `ggplot2`: Cuando los datos sean almacenado en **tibble** o **data frame**.

2. Una manera de graficar los retornos es usar `geom_density()`, que es la versión suavizada de un histograma y se suele usar para datos que son continuos.

Recomendado para retornos



3. Para graficar los retornos acumulados, se recomienda usar `geom_line()`.

Visualización: Retornos

? Actividad. Grafique los retornos logarítmicos de FANG

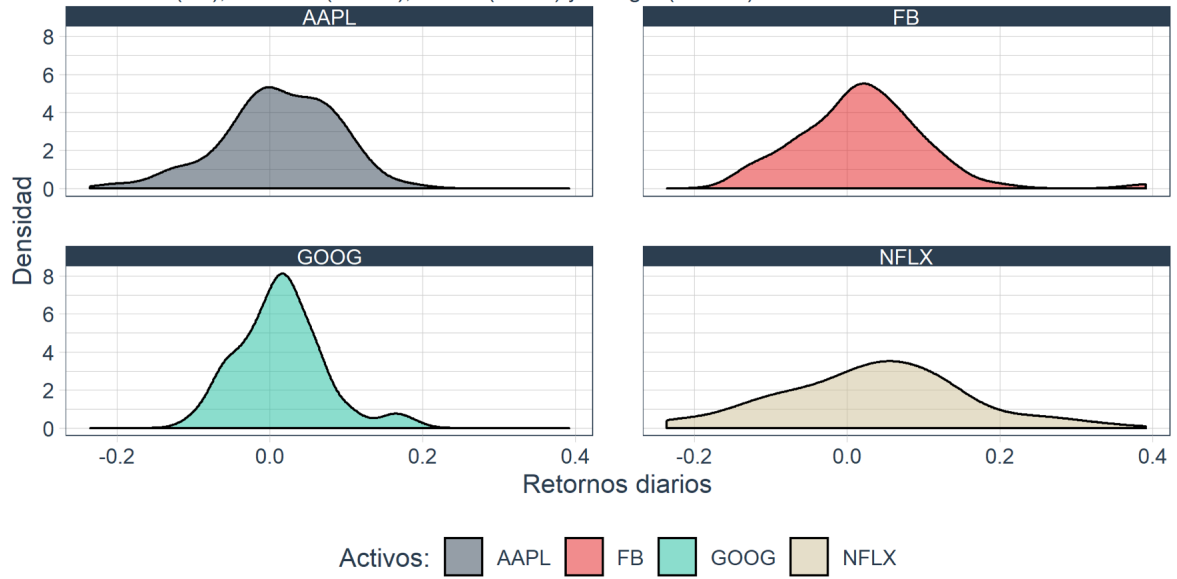
```
stock_returns_monthly_tbl %>%  
  gather(symbol, returns, -date) %>%  
  ggplot(mapping = aes(x = returns, fill = symbol)) +  
  geom_density(alpha = 0.5) +  
  labs(title = "Retornos Activos", subtitle = "Facebook (FB),  
        x = "Retornos diarios", y = "Densidad") +  
  theme_tq() +  
  scale_fill_tq() +  
  facet_wrap(~ symbol, ncol = 2) +  
  guides(fill=guide_legend(title="Activos:"))
```

tema
tigris

Legendas

Retornos

Facebook (FB), Amazon (AMZN), Netflix (NFLX) y Google (GOOG)



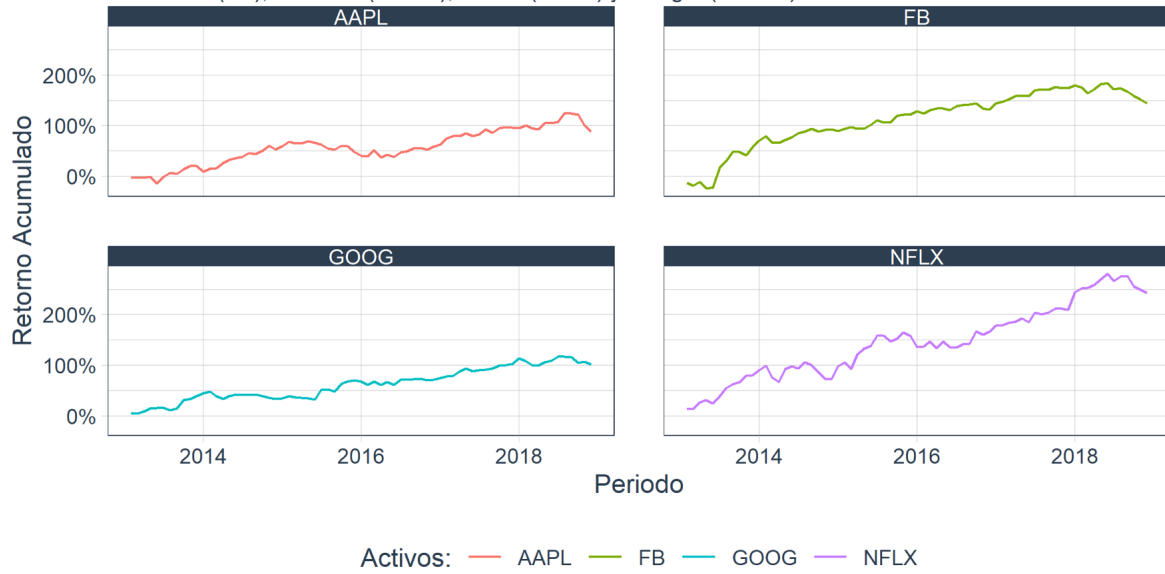
Visualización: Retornos acumulados

? Actividad. Grafique los retornos logarítmicos acumulados de FANG

```
stock_returns_acum_monthly_tbl %>%  
  gather(symbol, returns_acum, -date) %>%  
  ggplot(mapping = aes(x = date, y = returns_acum, color = sy  
  geom_line() +  
  labs(title = "Retornos Acumulado", subtitle = "Facebook (FE  
        x = "Periodo", y = "Retorno Acumulado") +  
  theme_tq() +  
  scale_fill_tq() +  
  facet_wrap(~ symbol, ncol = 2) +  
  guides(color = guide_legend(title="Activos:")) +  
  scale_y_continuous(labels = scales::percent)
```

Retornos Acumulado

Facebook (FB), Amazon (AMZN), Netflix (NFLX) y Google (GOOG)



Análisis Técnico

¿Qué es el Análisis Técnico?

1. El análisis técnico consiste en detectar determinados patrones de comportamiento de los precios en el pasado, con la esperanza de que dicho patrones vuelvan a repetirse y poder así aprovecharnos de ello.

2. Las bases provienen de la **Teoría de Dow.**

3. En esta sesiones veremos:

a. *Candlestick Chart*

b. *Bandas de Bollinger*

Análisis Candle Stick

Técnico:

```
end <- as_date("2018-07-31")
```

```
sp500_tbl %>%  
  ggplot(aes(x = date, y = close)) +  
  geom_candlestick(aes(open = open, high = high, low = low,  
    labs(title = "Standard & Poor 500 Candle Stick",  
      subtitle = "CStick, últimos 6 meses",  
      y = "Closing Price", x = "") +  
  coord_x_date(xlim = c(end - weeks(12), end),  
    ylim = c(2500, 3000)) +  
  theme_tq()
```

close = close
:
(falta)

Standard & Poor 500 Candle Stick

CStick, últimos 6 meses



Análisis

Técnico:

Bandas de Bollinger

→ media móvil

Comparar la volatilidad de la cotización de cualquier activo y el valor relativo de su precio a lo largo de un período de tiempo.

- La línea central (*Middle Line*, ML) es una media móvil simple.

$$SMA = \frac{P_M + P_{M-1} + \dots + P_{M-(n-1)}}{n} = ML$$

- La línea superior (*Top Line*, TL) es la misma línea central pero desplazada hacia arriba a un número determinado de desviaciones estándares (D).

$$TL = ML + (D * \sigma)$$

- La línea inferior (*Bottom Line*, BL) es la línea central desplazada hacia abajo al mismo número de desviaciones estándares.

$$BL = ML - (D * \sigma)$$

Ejemplo. Construcción Bandas de Bollinger

Solo seleccionaremos las últimas 6 semanas del Standard and Poor's a partir de la fecha 2018-07-31.

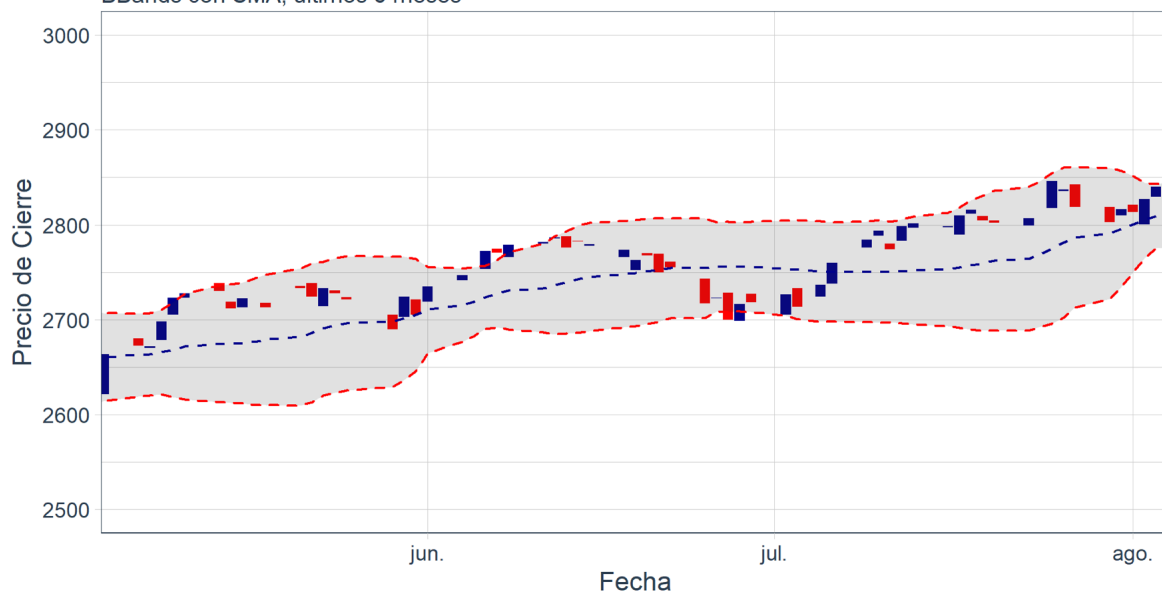
```
end <- as_date("2018-07-31")
```

El **tibble** debe contener las variables: open, high, low y close (OHLC).

```
sp500_tbl %>%  
  ggplot(aes(x=date, y=close, open = open,  
             high = high, low = low, close = close)) +  
  geom_candlestick() +  
  geom_bbands(ma_fun = SMA, sd = 2, n = 20) +  
  labs(title = "Standard & Poor 500",  
       subtitle = "BBands con SMA",  
       y = "Precio de Cierre", x = "Fecha") +  
  coord_x_date(xlim = c(end - weeks(12), end),  
              ylim = c(2500, 3000)) +  
  theme_tq()
```

Standard & Poor 500

BBands con SMA, últimos 6 meses



Análisis Tendencia

Técnicos:

INDICATOR	TTR NAME	QUANTMOD NAME
Welles Wilders Directional Movement Indicator	ADX	addADX
Double Exponential Moving Average	DEMA	addDEMA
Exponential Moving Average	EMA	addEMA
Simple Moving Average	SMA	addSMA
Parabolic Stop and Reverse	SAR	addSAR
Exponential Volume Weighted Moving Average	EVWMA	addEVWMA
Moving Average Convergence Divergence	MACD	addMACD
Triple Smoothed Exponential Oscillator	TRIX	addTRIX
Weighted Moving Average	WMA	addWMA

Análisis Volatility

Técnicos:

INDICATOR	TTR NAME	QUANTMOD NAME
Average True Range	ATR	addATR
Bollinger Bands	BBands	addBBands
Price Envelope	N/A	addEnvelope

Análisis Volume

Técnicos:

INDICATOR	TTR NAME	QUANTMOD NAME
Chaiken Money Flow	CMF	addCMF
Volume	N/A	addVo

Análisis Momentum

Técnicos:

INDICATOR	TTR NAME	QUANTMOD NAME
Commodity Channel Index	CCI	addCCI
Chande Momentum Oscillator	CMO	addCMO
Detrended Price Oscillator	DPO	addDPO
momentum	addMomentum	
Rate of Change	ROC	addROC
Relative Strength Indicator	RSI	addRSI
Stochastic Momentum Index	SMI	addSMI
Williams %R	WPR	addWPR

Construyendo un portafolio

Asignando los pesos

Como sabemos de nuestros cursos de finanzas, para construir un portafolio necesitamos los pesos. Estos pueden provenir de una procesos de optimización o bien asignarlos bajo algún criterio.

```
w <- c(0.25, 0.25, 0.25, 0.25)
```

Luego lo asignamos a cada `symbols`.

```
tibble(symbols, w)
```

Debemos verificar que sumen 1.

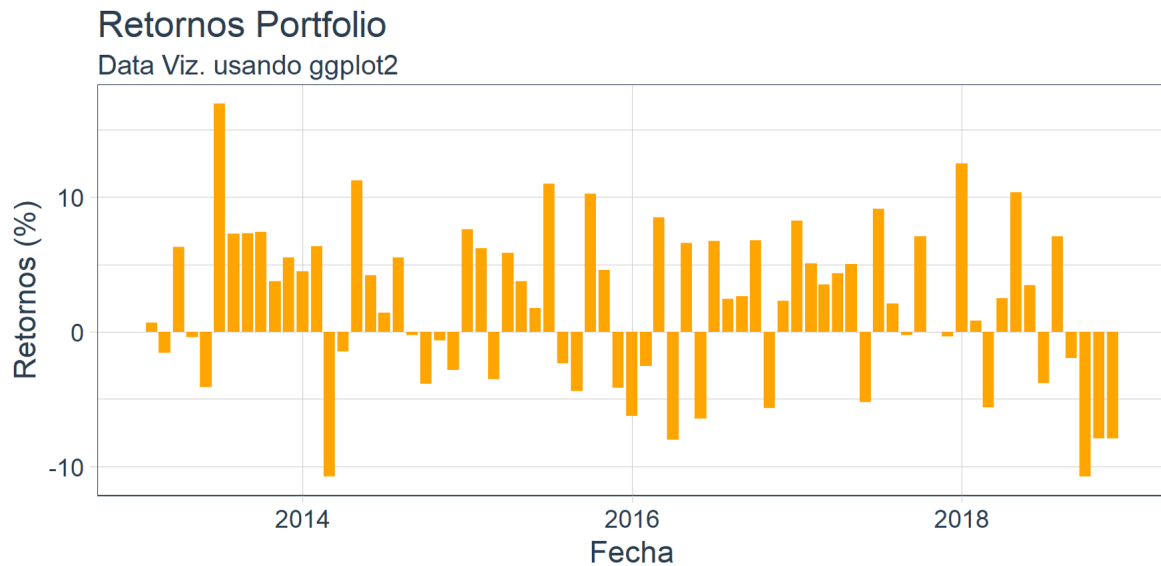
```
tibble(symbols, w) %>%  
  summarise(total_weight = sum(w))
```

Retornos del portafolio

? Actividad. Calcule el retorno del portafolio rebalanceandolo cada mes

```
portfolio_returns_monthly <- stock_returns_monthly_tbl %>%  
  gather(asset, returns, -date) %>%  
  tq_portfolio(assets_col = asset,  
    returns_col = returns,  
    weights = w,  
    col_rename = "returns",  
    rebalance_on = "months")
```

Visualización de los retornos: ggplot2



Visualización de los retornos: highcharter



Visualización de los retornos: plotly

Existe el paquete plotly el que nos permite construir gráficos dinámicos. A continuación les presento lo útil que es para realizar análisis técnico.

- [Ejemplo plotly](#)

Material de Apoyo

Recomendaciones

Libro basado en tidyquant de [Jonathan K. Regenstein Jr.](#):

1. [Reproducible Finance with R: Code Flows and Shiny Apps for Portfolio Analysis](#)

Muchas Gracias



Slides creadas por el paquete de R [xaringan](#).

El chakra proviene [remark.js](#), [knitr](#), y R Markdown.