



# Finanzas en R

## Notebook 03

Sebastián Egaña Santibáñez

Nicolás Leiva Díaz

---

### Enlaces del profesor

<https://sevana.netlify.app>

<https://github.com/sebaegana>

<https://www.linkedin.com/in/sebastian-egana-santibanez/>

---

### Precios de activos financieros utilizando Quantmod

Quantmod es una de las librerías financieras más utilizadas para la obtención de precios.

La manera más simple de obtener precios es la siguiente:

```
library(tidyverse)
library(tidyquant)
library(quantmod)

getSymbols("AAPL",
           src = "yahoo",
           from = "2010-01-01",
```

```
to = "2010-07-30",  
periodicity = "daily")
```

```
[1] "AAPL"
```

Solo debemos declarar el simbolo (instrumento financiero), el período y la periodicidad del instrumento.

La librería también posee una forma simple de graficar las series obtenidas:

```
chartSeries(AAPL, TA=NULL)
```



Otro ejemplo:

```
chartSeries(AAPL, subset = "last 3 months")
```



Para llevar esto a ggplot, deberíamos aplicar algunas funciones convirtiendo el set de datos obtenido en un formato adecuado.

```
df_aapl_dates <- as_tibble(index(AAPL))
df_aapl_data <- as_tibble(AAPL)

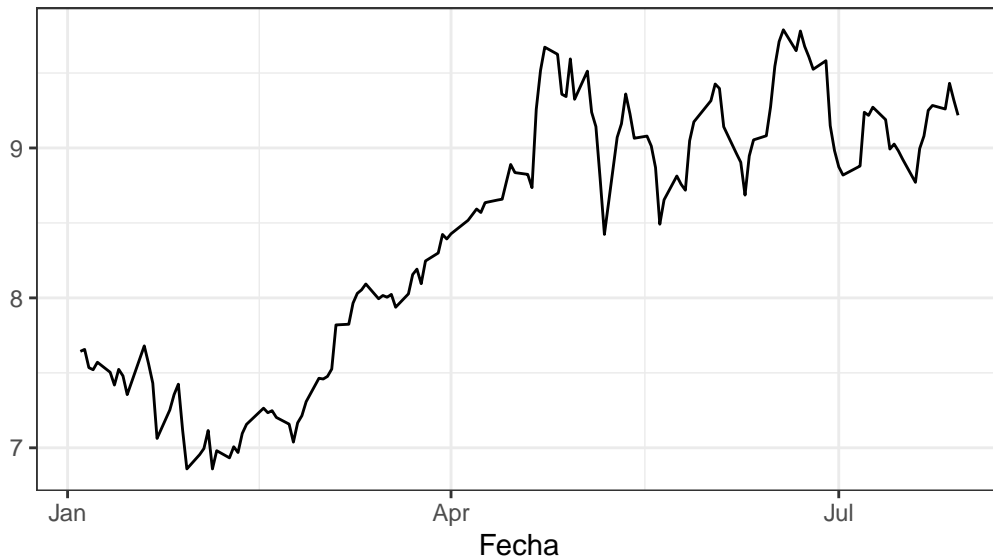
df_aapl <- df_aapl_dates %>%
  bind_cols(df_aapl_data)
```

Graficamos:

```
g1 <- ggplot(df_aapl) + geom_line(mapping = aes(value, AAPL.Close))
g1 <- g1 + labs(title = "AAPLE", subtitle = "Año 2010") +
  xlab("Fecha") + ylab("")
g1 <- g1 + theme_bw()
g1
```

## AAPLE

Año 2010



## Múltiples símbolos

```
tickers <- c("ORCL","AMD","IBM","NVDA")  
  
getSymbols(tickers, src = "yahoo",  
           from = "2010-01-01",  
           to = "2018-07-30", periodicity = "daily")
```

```
[1] "ORCL" "AMD"  "IBM"  "NVDA"
```

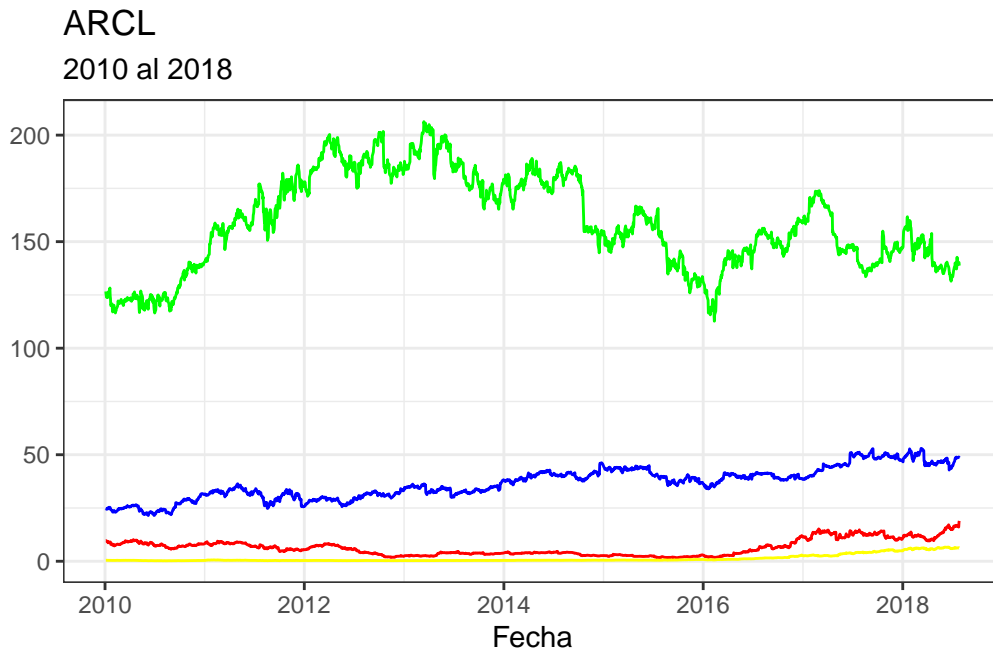
```
list <- lapply(tickers, function(x) Cl(get(x)))  
precio.cierre <- do.call(merge,list)  
  
df_all_dates <- as_tibble(index(precio.cierre))  
df_all_data <- as_tibble(precio.cierre)  
  
df_all <- df_all_dates %>%  
  bind_cols(df_all_data)
```

Graficamos:

```

g1 <- ggplot(df_all) +
  geom_line(mapping = aes(value, ORCL.Close), color = 'blue') +
  geom_line(mapping = aes(value, AMD.Close), color = 'red') +
  geom_line(mapping = aes(value, IBM.Close), color = 'green') +
  geom_line(mapping = aes(value, NVDA.Close), color = 'yellow')
g1 <- g1 + labs(title = "ARCL", subtitle = "2010 al 2018") + xlab("Fecha") + ylab("")
g1 <- g1 + theme_bw()
g1

```



Pero esto no tiene mucho sentido, debido a las diferencias de escala de cada una de las acciones. La medida básica de comparación para acciones corresponde a los retornos.

## Calcular retornos

La fórmula básica para obtener retorno, corresponde a la variación porcentual:

$$r_t = \frac{p_t - p_{t-1}}{p_{t-1}}$$

en Finanzas también se puede trabajar con una expresión logaritmica:

Table 1: Summary Statistics

Variable	N	Mean	Std. Dev.	Min	Pctl. 25	Pctl. 75	Max
orcl	2156	0.00031	0.015	-0.12	-0.0066	0.0082	0.097
amd	2156	0.00031	0.035	-0.28	-0.017	0.017	0.42
ibm	2156	0.000042	0.012	-0.086	-0.0059	0.0062	0.085
nvda	2156	0.0012	0.024	-0.12	-0.011	0.013	0.26

$$r_t = \ln\left(\frac{p_t}{p_{t-1}}\right)$$

Que converge al mismo valor cuando el intervalo tiende a cero.

Calculamos de las dos maneras:

```
retornos_2 <- data.frame(apply(precio.cierre, 2,
                             function(x) Delt(x, type = "arithmetic")),
                      fecha = index(precio.cierre)) %>%
  rename(orcl = ORCL.Close,
         amd = AMD.Close,
         ibm = IBM.Close,
         nvda = NVDA.Close) %>%
  na.omit()

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()
```

Veamos algunas diferencias entre cada data set (cargamos la librería vtable para esto):

```
library(vtable)
st(retornos)

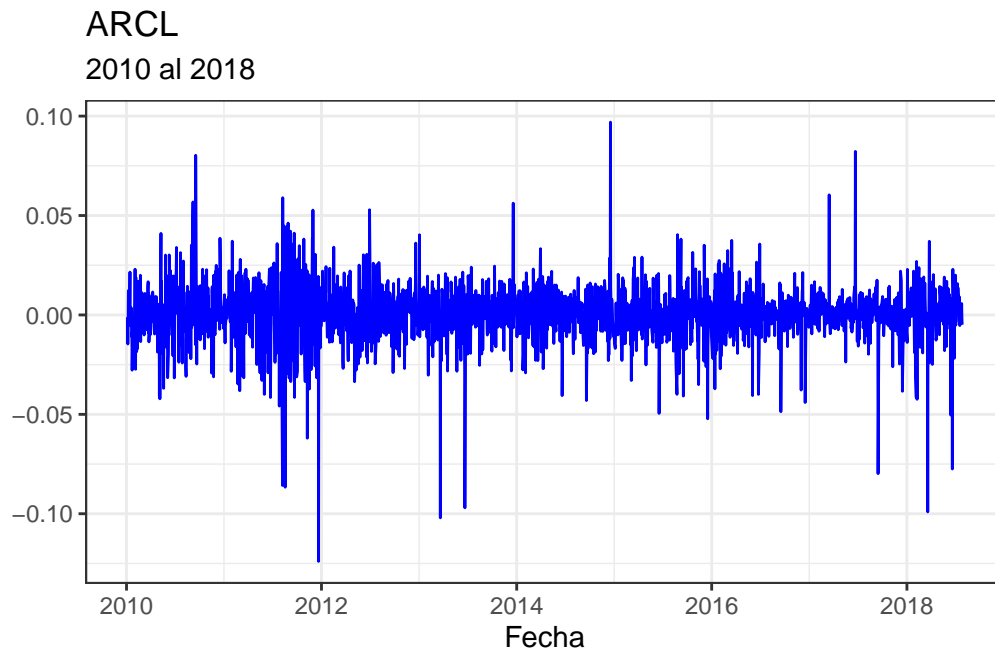
st(retornos_2)
```

Graficamos:

Table 2: Summary Statistics

Variable	N	Mean	Std. Dev.	Min	Pctl. 25	Pctl. 75	Max
orcl	2156	0.00042	0.015	-0.12	-0.0066	0.0082	0.1
amd	2156	0.00093	0.036	-0.24	-0.017	0.018	0.52
ibm	2156	0.00012	0.012	-0.083	-0.0059	0.0062	0.089
nvda	2156	0.0015	0.025	-0.12	-0.011	0.013	0.3

```
g1 <- ggplot(retornos) +
  geom_line(mapping = aes(fecha, orcl), color = 'blue')
g1 <- g1 + labs(title = "ARCL", subtitle = "2010 al 2018") + xlab("Fecha") + ylab("")
g1 <- g1 + theme_bw()
g1
```

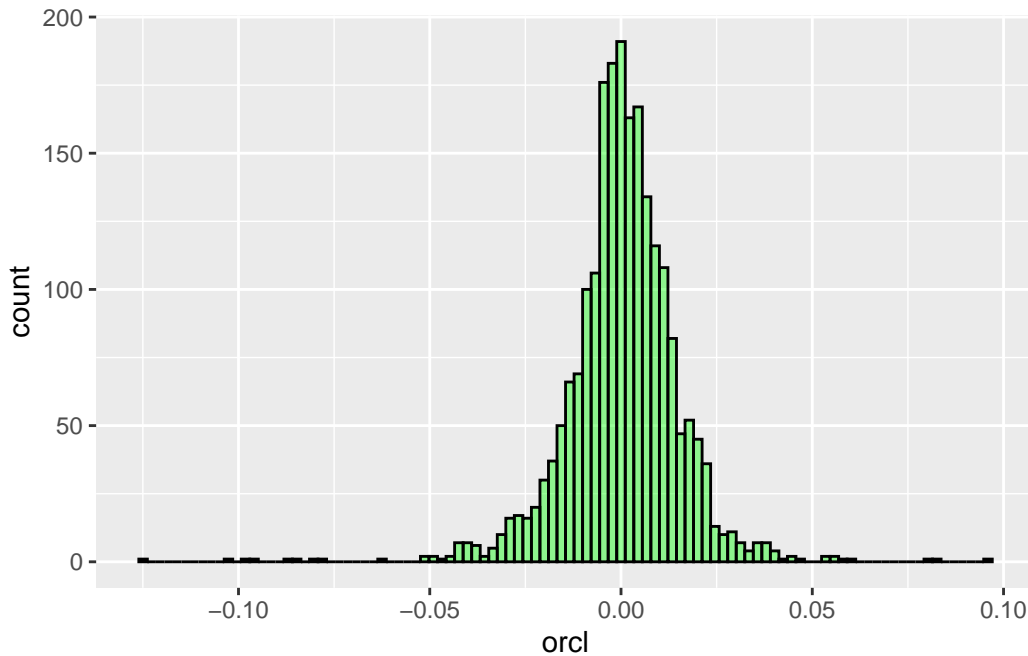


Podemos también construir un histograma de los retornos:

```
p1 <- ggplot(retornos, aes(x=orcl)) +
  geom_histogram(bins = 100, col='black', fill='green', alpha=0.4)
p1
```

Table 3: Summary Statistics

Variable	N	Mean	Std. Dev.	Min	Pctl. 25	Pctl. 75	Max
orcl	2156	0.00031	0.015	-0.12	-0.0066	0.0082	0.097
amd	2156	0.00031	0.035	-0.28	-0.017	0.017	0.42
ibm	2156	0.000042	0.012	-0.086	-0.0059	0.0062	0.085
nvda	2156	0.0012	0.024	-0.12	-0.011	0.013	0.26



¿Algún comentario sobre la distribución? Veamos denuevo las medidas de resumen:

```
library(vtable)
st(retornos)
```

### Retorno acumulados

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



```
library("reshape2")
```

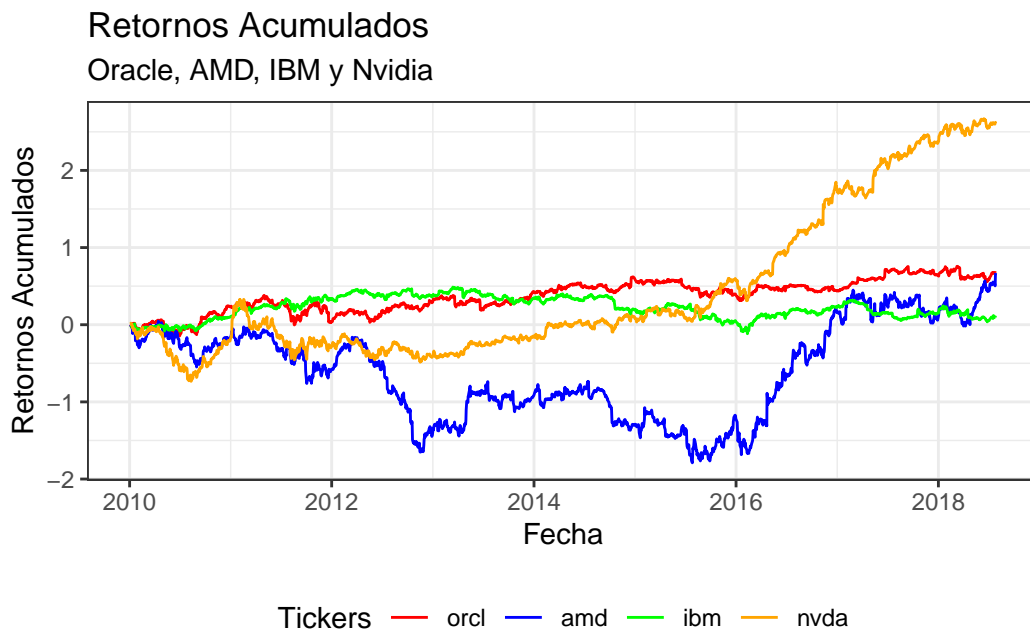
Attaching package: 'reshape2'

The following object is masked from 'package:tidyr':

smiths

```
reshape <- melt(acumulados, id.vars = "fecha")
```

```
g2 <- ggplot(reshape) + geom_line(mapping = aes(fecha,value, color = variable))  
g2 <- g2 + labs(title = "Retornos Acumulados", subtitle = "Oracle, AMD, IBM y Nvidia")  
g2 <- g2 + theme_bw() + xlab("Fecha") + ylab("Retornos Acumulados")  
g2 <- g2 + scale_color_manual("Tickers", values = c("red","blue","green","orange"))  
g2 <- g2 + theme(legend.position = "bottom")  
g2
```



## Ejemplo aplicado

Una de las principales medidas de riesgo es el Value at risk o VaR. Se define como “cuál sería la pérdida que solo será excedida en  $p \cdot 100\%$  veces en los próximos  $k$  días”. En este sentido, nos preguntamos por la mayor pérdida que se genera con un  $x\%$  de probabilidad.

Veamos un ejemplo:

```
library(PerformanceAnalytics)
library(tidyquant)
library(tseries)
library(tidyverse)
```

Determinamos las acciones:

```
symbol_name <- c("AAPL", "GOOG", "AMZN", "F", "A", "TQQQ") # try to put Your tickers!!
vahy <- c(0.2,0.2,0.2,0.2,0.1,0.1)
FROM <- "2020-01-01" # change to Your dates!!!
TO <- "2022-12-31"
```

Generamos una función para obtener los precios:

```
# preparing one table in common for all the downloaded tickers - You can change
for (i in 1:length(symbol_name)) {
  prac <- Ad(getSymbols(symbol_name[i], from = FROM, to = TO,auto.assign=FALSE))
  if (i==1) {
    price <- prac
  } else{
    price <- merge(price,prac)
  }
}
rm(prac) # prac is just temporary variable to remove
colnames(price) <- symbol_name #puting the names of the shares
```

Calculamos los retornos en base a log:

```
return_a <- CalculateReturns(price, method="log")
#hist(return_a$AAPL)

# next cycle imputes the missing data by the variable medians
for(i in 1:dim(return_a)[2]){
  return_a[,i][is.na(return_a[,i])] <- median(return_a[,i],na.rm = TRUE)
```

```
}  
return2_a <- return_a[-1,]
```

Obtenemos el VaR:

```
Qh <- VaR(return2_a[,1], p=0.95, method="gaussian")  
Qh
```

```
AAPL  
VaR -0.03747267
```

Cambiamos el método:

```
Qv <- VaR(return2_a[,1], p=0.95, method="historical")  
Qv
```

```
AAPL  
VaR -0.03546483
```

Para obtener el monto monetario aplicamos la siguiente formula:

$$VaR(\text{en pesos}) = V_{PF}(1 - \exp(-VaR))$$

Asumamos que tenemos una posición de 2.000.000:

```
2000000 * (1 - exp(-Qh))
```

```
AAPL  
VaR -76367.25
```

## Referencias:

- Gabriel Cabrera G., Introducción a las finanzas cuantitativas, <https://finance-r.netlify.app>
- <https://rpubs.com/gazdavlاديمir/1039600>
- <https://www.r-bloggers.com/2022/02/nonparametric-value-at-risk-var/>
- <https://www.youtube.com/watch?v=0N7XBk7YA5A&list=PLJqpMAAwcLuus0FqGByq8FC3ukwYcJz7A>