

Laborator 3

Elemente de statistică descriptivă

Obiectivul acestui laborator este de a prezenta câteva elemente (numerice și grafice) de statistică descriptivă/exploratorie pentru studiul rentabilității zilnice (*daily returns*) a unui număr de active reprezentative (assets).

1 Importarea datelor

Înainte de a analiza datele trebuie să le descărcăm și să le punem într-un format ușor de utilizat pentru analiză. Datele pot fi obținute de pe diferite platforme online precum Yahoo!Finance sau Google Finance. Mai jos este ilustrat codul funcției `google_stocks` care permite extragerea datelor de pe platforma Google Finance:

```
google_stocks <- function(sym, current = TRUE, sy = 2007, sm = 1, sd = 1, ey, em, ed)
{
  # sy, sm, sd, ey, em, ed corespund la
  # start year, start month, start day, end year, end month, si end day

  # Daca este TRUE folosim data curenta
  if(current){
    system_time <- as.character(Sys.time())
    ey <- as.numeric(substr(system_time, start = 1, stop = 4))
    em <- as.numeric(substr(system_time, start = 6, stop = 7))
    ed <- as.numeric(substr(system_time, start = 9, stop = 10))
  }

  tmp <- tempfile()

  # cat("downloading ", sym, ".....\n\n")
  google.URL = "http://finance.google.com/finance/historical?"
  download.file(paste(google.URL, "q=", sym, "&startdate=",
    month.abb[sm], "+", sprintf("%.2d", sd),
    "+", sy, "&enddate=", month.abb[em], "+",
    sprintf("%.2d", ed), "+", ey, "&output=csv",
    sep = ""), destfile = tmp, quiet = FALSE)
  google_out <- read.csv(tmp)

  # Redenumim prima coloana
  if(!is.null(google_out)){
    names(google_out)[1] = "Date"
  }

  # transformam coloana timp
  google_out$Date = as.Date(strptime(google_out$Date, "%d-%B-%y"),
    origin = "1970-01-01")

  google_out = google_out[order(google_out$Date), ]

  return(google_out)
}
```

```
}
```

Pachetul R, *quantmod* pune la dispoziție o serie de funcții care permit atât accesul la datele din Yahoo!Finance sau Google Finance, dar și din alte surse, cât și utilizarea unor tehnici specifice modelării financiare. De exemplu, funcția de mai sus are echivalentul (mult mai complex) `getSymbols()` din pachetul *quantmod*.

Să presupunem că suntem interesați de stoc-urile (*stock*) firmelor Apple, Microsoft și Google din perioada 01-01-2000 până în prezent. Pentru a extrage aceste date va trebui să folosim simbolul/abrevierea (ticker symbol - search pe Goole după *ticker symbol*) fiecărei firme separat. Vom accesa aceste date cu ajutorul funcției `google_stocks`:

```
# data de start
sy = 2005
sm = 1
sd = 1

# datele
apple_data = google_stocks("AAPL", sy = sy, sm = sm, sd = sd)
msft_data = google_stocks("MSFT", sy = sy, sm = sm, sd = sd)
google_data = google_stocks("GOOG", sy = sy, sm = sm, sd = sd)
```

sau citind din fisier

```
# Google data
google_data = read.csv("dataIn/google_data.csv",
                      col.names = c("Date", "Open", "High", "Low",
                                    "Close", "Volume", "Adjusted"))

google_data$Date = as.Date(strptime(as.character(google_data$Date), "%Y-%m-%d"),
                          origin = "1970-01-01")

google_data = google_data[order(google_data$Date), ]

# Apple data
apple_data = read.csv("dataIn/apple_data.csv",
                     col.names = c("Date", "Open", "High", "Low",
                                   "Close", "Volume", "Adjusted"))
apple_data$Date = as.Date(strptime(as.character(apple_data$Date), "%Y-%m-%d"),
                          origin = "1970-01-01")

apple_data = apple_data[order(apple_data$Date), ]

# Microsoft data
msft_data = read.csv("dataIn/msft_data.csv",
                    col.names = c("Date", "Open", "High", "Low",
                                  "Close", "Volume", "Adjusted"))

msft_data$Date = as.Date(strptime(as.character(msft_data$Date), "%Y-%m-%d"),
                          origin = "1970-01-01")

msft_data = msft_data[order(msft_data$Date), ]
```

Același lucru poate fi obținut cu ajutorul funcției `getSymbols` (descărcate de pe Yahoo!Finance):

```
# pt a utiliza pachetul trebuie incarcat
if (!require("quantmod")) {
```

```
install.packages("quantmod")
library(quantmod)
}

# data de start
start <- as.Date("2005-01-01")

# datele din Yahoo finance
apple_data_yh = data.frame(getSymbols("AAPL", from = start, auto.assign = F))
msft_data_yh = data.frame(getSymbols("MSFT", from = start, auto.assign = F))
google_data_yh = data.frame(getSymbols("GOOG", from = start, auto.assign = F))
```

Vom folosi în cele ce urmează datele obținute cu funcția `google_stocks`. Google Finance oferă 5 serii pentru fiecare bun/activ. *Open* corespunde prețului stoc-ului la începutul zilei de tranzacționare și nu trebuie să fie neapărat egal cu prețul cu care s-a închis ziua precedentă, *High* și respectiv *Low* este prețul cel mai mare și respectiv cel mai mic din ziua de tranzacționare, iar *Close* este prețul stoc-ului la închiderea zilei de tranzacționare. Coloana *Volume* arată câte stoc-uri au fost tranzacționate în ziua respectivă.

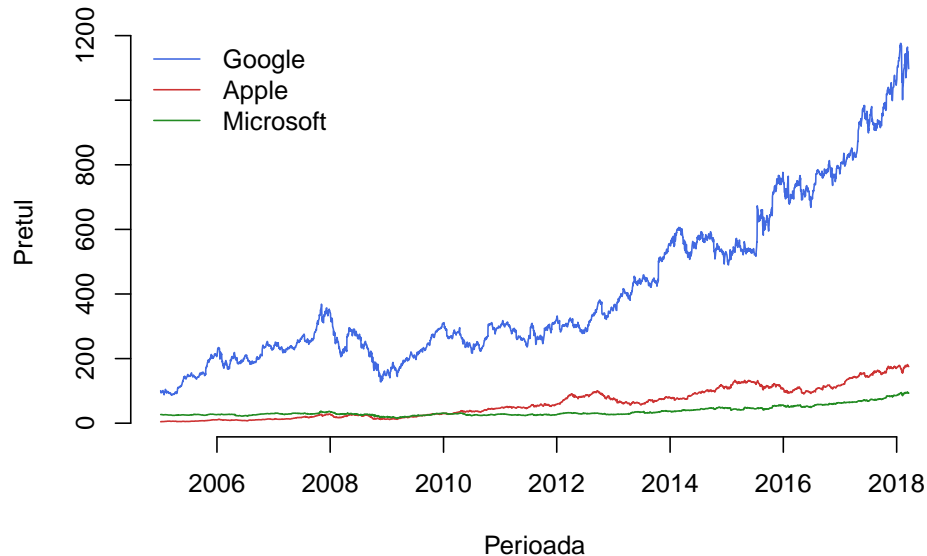
```
head(apple_data, n = 5)
  Date      Open      High      Low      Close      Volume Adjusted
1 2005-01-03 4.627143 4.650714 4.471428 4.520714 172998000 3.060327
2 2005-01-04 4.556428 4.676429 4.497857 4.567143 274202600 3.091757
3 2005-01-05 4.604286 4.660714 4.575000 4.607143 170108400 3.118836
4 2005-01-06 4.619286 4.636428 4.523571 4.610714 176388800 3.121253
5 2005-01-07 4.642857 4.973571 4.625000 4.946429 556862600 3.348517
head(msft_data, n = 5)
  Date      Open      High      Low      Close      Volume Adjusted
1 2005-01-03 26.80 26.95 26.65 26.74 65002900 19.93452
2 2005-01-04 26.87 27.10 26.66 26.84 109442100 20.00907
3 2005-01-05 26.84 27.10 26.76 26.78 72463500 19.96435
4 2005-01-06 26.85 27.06 26.64 26.75 76890500 19.94198
5 2005-01-07 26.82 26.89 26.62 26.67 68723300 19.88235
head(google_data, n = 5)
  Date      Open      High      Low      Close      Volume Adjusted
1 2005-01-03 98.06220 101.16204 97.09847 100.70004 31894300 100.70004
2 2005-01-04 100.04928 100.80933 96.11487 96.62157 27690700 96.62157
3 2005-01-05 96.09996 97.81381 95.49390 96.12977 16580200 96.12977
4 2005-01-06 96.90970 97.31705 93.25348 93.66579 20909100 93.66579
5 2005-01-07 94.70404 96.49738 93.78005 96.29867 19451500 96.29867
```

Pentru a vizualiza evoluția prețului stoc-ului la închidere avem următoarele grafice:

```
plot(google_data$Date, google_data$Close,
     type = "l",
     col = "royalblue",
     bty = "n",
     ylim = c(min(c(google_data$Close, apple_data$Close, msft_data$Close), na.rm = TRUE),
               max(c(google_data$Close, apple_data$Close, msft_data$Close), na.rm = TRUE)),
     xlab = "Perioada",
     ylab = "Pretul")

lines(apple_data$Date, apple_data$Close,
      col = "brown3")
lines(msft_data$Date, msft_data$Close,
      col = "forestgreen")
```

```
legend("topleft",  
      legend = c("Google", "Apple", "Microsoft"),  
      col = c("royalblue", "brown3", "forestgreen"),  
      lty = 1,  
      bty = "n")
```



2 Rentabilitate (Returns)

Scopul unei investiții este acela de a face profit, prin urmare investitorii sunt interesați în a face investiții care produc venituri mari relativ la mărimea investiției. Rentabilitatea / Rata de rentabilitate (returns) măsoară modificarea prețului unui bun exprimat ca o fracție din prețul inițial.

2.1 Rentabilitate netă și brută (net and gross return)

Să presupunem că achiziționăm un bun (activ, stock, etc.) la momentul t_0 cu prețul P_{t_0} și îl vindem la momentul t_1 cu prețul P_{t_1} . Dacă între t_0 și t_1 nu avem schimbări de preț atunci rata de rentabilitate pe perioada $t_0 - t_1$ este

$$R(t_0, t_1) = \frac{P_{t_1} - P_{t_0}}{P_{t_0}}.$$

Perioada dintre t_0 și t_1 se numește perioada de retenție a bunului (*holding period*), perioada dintre achiziția și vânzarea unui bun (activ, etc.), și poate fi măsurată în secunde, minute, ore, zile, luni, etc. Dacă P_t este prețul unui activ la momentul t (să zicem la sfârșitul lunii t) și P_{t-1} este prețul activului la momentul $t - 1$ atunci rentabilitatea netă (*net return*) în perioada de la $t - 1$ la t este

$$R_t = \frac{P_t - P_{t-1}}{P_{t-1}}$$

și putem spune că

$$\text{venitul} = \text{investitia initiala} \times \text{rentabilitatea neta.}$$

Rentabilitatea brută (*gross return*) este definită prin

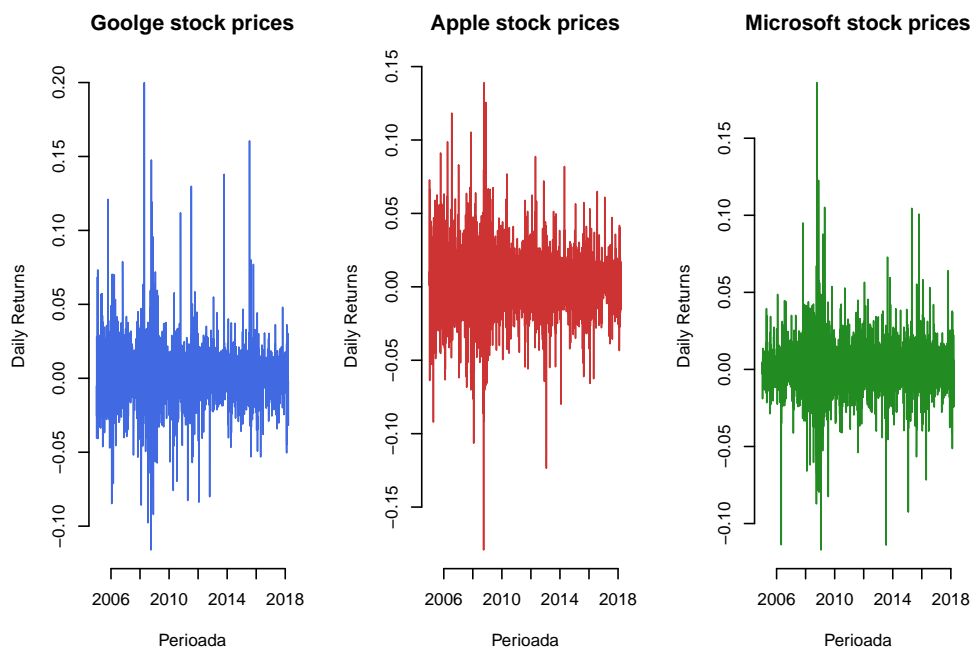
$$\frac{P_t}{P_{t-1}} = 1 + R_t.$$

Spre exemplu să considerăm o investiție de o lună într-un stock Microsoft. Să presupunem că achiziționăm stock-ul în luna $t - 1$ cu prețul $P_{t-1} = 85 \text{ u.m.}$ și îl vindem în luna următoare cu $P_t = 90 \text{ u.m.}$. Atunci rentabilitatea netă și brută pe perioada de 1 lună sunt: $R_t = \frac{90-85}{85} = 0.058$ iar $1 + R_t = 1.058$ ceea ce înseamnă că investiția a condus la o rentabilitate de 5.8% sau altfel spus 1 u.m. investită în stock-ul Microsoft în luna $t - 1$ a crescut la 1.058 u.m. în luna t (creșterea a fost de 105.8%).

Să presupunem că vrem să calculăm rentabilitatea zilnică (netă și brută) a stock-urilor Apple, Google și Microsoft:

```
# Rentabilitatea zilnică
google_ret_s_daily = google_data$Close[-1] / google_data$Close[-length(google_data$Close)] - 1
apple_ret_s_daily = apple_data$Close[-1] / apple_data$Close[-length(apple_data$Close)] - 1
msft_ret_s_daily = msft_data$Close[-1] / msft_data$Close[-length(msft_data$Close)] - 1

head(cbind(google_ret_s_daily, apple_ret_s_daily, msft_ret_s_daily))
      google_ret_s_daily apple_ret_s_daily msft_ret_s_daily
[1,]      -0.040501234      0.0102702803      0.003739716
[2,]      -0.005089951      0.0087582105     -0.002235432
[3,]      -0.025631748      0.0007751008     -0.001120276
[4,]       0.028109237      0.0728119332     -0.002990654
[5,]       0.006241924     -0.0041878697      0.004874353
[6,]      -0.007792465     -0.0638049631     -0.002611903
```



și rentabilitatea lunară:

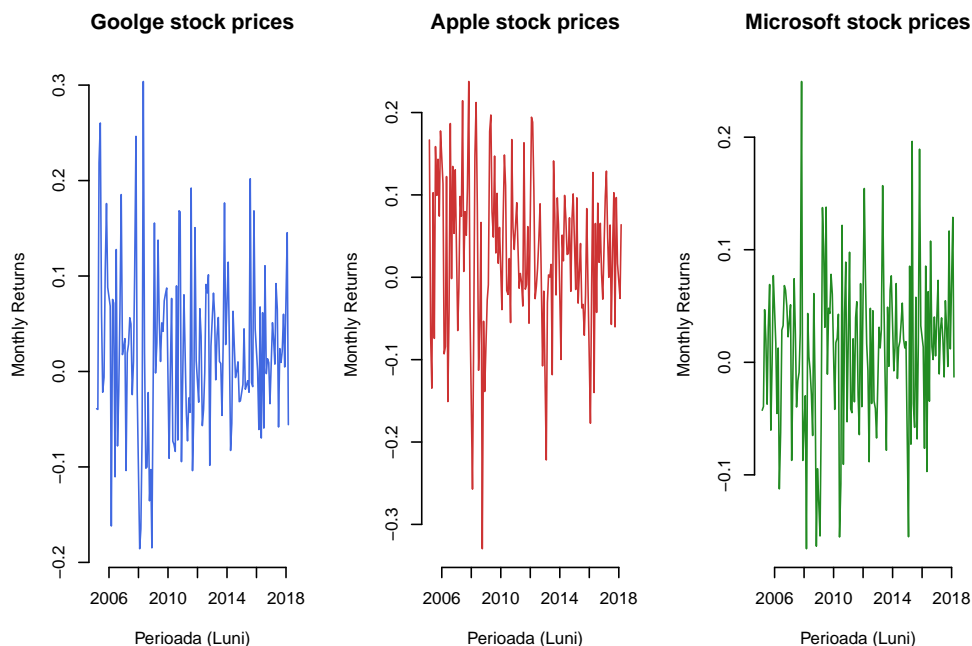
```
# Rentabilitatea lunara
returnMonth = function(dat){
  dat$Day = as.numeric(strftime(dat$Date, "%d"))
  dat$Month = as.numeric(strftime(dat$Date, "%m"))
  dat$Year = as.numeric(strftime(dat$Date, "%Y"))

  dat_month_diff = diff(dat$Month)
  dat_month = dat[dat_month_diff == 1, ]

  month_ret = dat_month$Close[-1] / dat_month$Close[-length(dat_month$Close)] - 1
  return(data.frame(date = dat_month$Date[-1], ret = month_ret))
}

google_ret_s_monthly = returnMonth(google_data)
apple_ret_s_monthly = returnMonth(apple_data)
msft_ret_s_monthly = returnMonth(msft_data)

head(cbind(google_ret_s_monthly$ret, apple_ret_s_monthly$ret,
            msft_ret_s_monthly$ret))
      [,1]      [,2]      [,3]
[1,] -0.03900416  0.16670997 -0.04261800
[2,] -0.03978939 -0.07111008 -0.03934817
[3,]  0.21876907 -0.13462914  0.04675213
[4,]  0.26031817  0.10260667  0.01976285
[5,]  0.06087933 -0.07419507 -0.03720927
[6,] -0.02172367  0.15865239  0.03099843
```



Rentabilitatea brută pe o k -perioadă (dacă $k = 2$ și perioada este o lună atunci avem rentabilitatea brută pentru 2 luni) este dată de formula

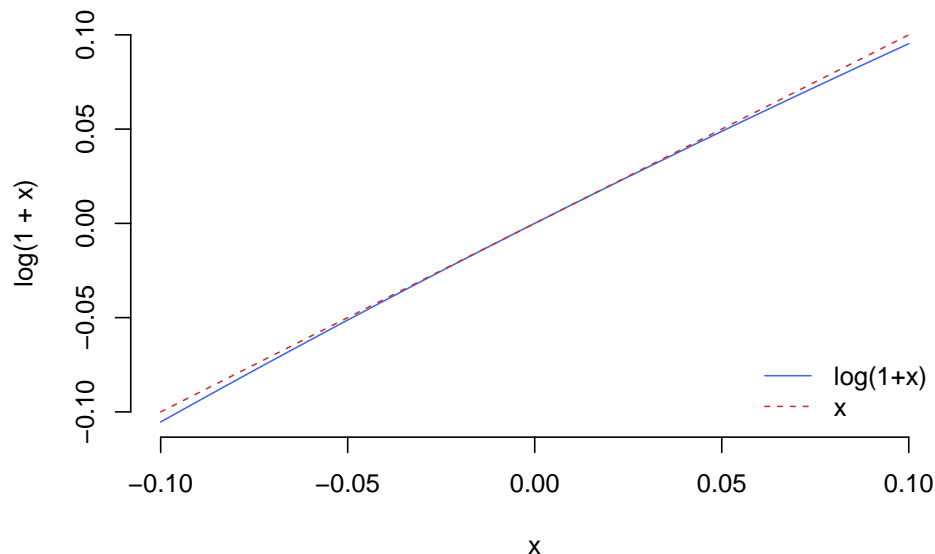
$$1 + R_t(k) = \frac{P_t}{P_{t-k}} = \frac{P_t}{P_{t-1}} \frac{P_{t-1}}{P_{t-2}} \cdots \frac{P_{t-k+1}}{P_{t-k}} = \prod_{j=0}^{k-1} (1 + R_{t-j}).$$

2.2 Rentabilitate compusă / logaritmică (log returns)

Dacă luăm prețurile în scară logaritmică, $p_t = \log(P_t)$, atunci definim logaritmul rentabilității - *log returns* (sau *continuously compounded returns*) prin

$$r_t = \log(1 + R_t) = p_t - p_{t-1}.$$

Deoarece, pentru x suficient de mic putem folosi aproximarea $\log(1 + x) \approx x$, putem spune că $r_t \approx R_t$ în special pentru rentabilități calculate pe perioade scurte de timp (e.g. zilnice).



De asemenea avem că

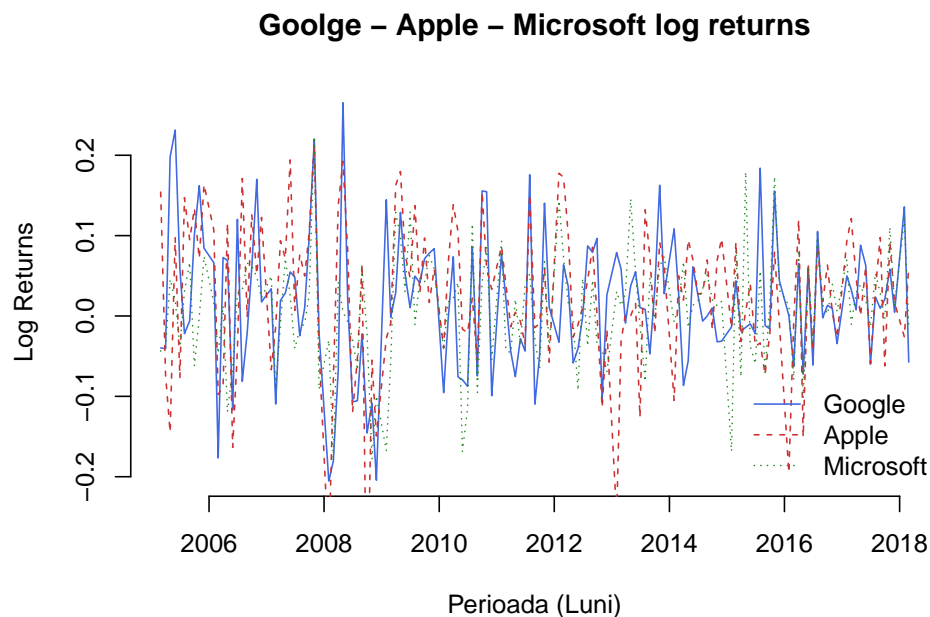
$$r_t(k) = \log(1 + R_t(k)) = \log \left(\prod_{j=0}^{k-1} (1 + R_{t-j}) \right) = \sum_{j=0}^{k-1} r_{t-j}.$$

Pentru datele noastre avem

```
# log returns
google_ret_s_monthly$cret = log(1+google_ret_s_monthly$ret)
apple_ret_s_monthly$cret = log(1+apple_ret_s_monthly$ret)
msft_ret_s_monthly$cret = log(1+msft_ret_s_monthly$ret)

# ilustrare grafica
plot(google_ret_s_monthly$date, google_ret_s_monthly$cret,
     type = "l",
```

```
col = "royalblue",  
main = "Goolge - Apple - Microsoft log returns",  
xlab = "Perioada (Luni)",  
ylab = "Log Returns",  
bty = "n")  
  
lines(apple_ret_s_monthly$date, apple_ret_s_monthly$cret,  
      col = "brown3",  
      lty = 2)  
  
lines(msft_ret_s_monthly$date, msft_ret_s_monthly$cret,  
      col = "forestgreen",  
      lty = 3)  
  
legend("bottomright",  
      legend = c("Google", "Apple", "Microsoft"),  
      col = c("royalblue", "brown3", "forestgreen"),  
      lty = c(1,2,3),  
      bty = "n")
```



2.3 Ajustarea pentru dividende

Dacă un activ plătește un dividend D_t undeva între momentul de timp $t - 1$ și t atunci rentabilitatea brută la momentul t se calculează după formula

$$1 + R_t = \frac{P_t + D_t}{P_{t-1}}$$

iar rentabilitatea netă este $R_t = \frac{P_t + D_t - P_{t-1}}{P_{t-1}} = \frac{P_t - P_{t-1}}{P_{t-1}} + \frac{D_t}{P_{t-1}}$ unde $\frac{P_t - P_{t-1}}{P_{t-1}}$ se numește câștigul de capital (*capital gain*) iar $\frac{D_t}{P_{t-1}}$ se numește randamentul dividendului (*dividend yield*). Avem astfel că

$$1 + R_t(k) = \prod_{j=0}^{k-1} \frac{P_{t-j} + D_{t-j}}{P_{t-j-1}} = \prod_{j=0}^{k-1} (1 + R_{t-j})$$

iar

$$r_t(k) = \log(1 + R_t(k)) = \sum_{j=0}^{k-1} \log\left(\frac{P_{t-j} + D_{t-j}}{P_{t-j-1}}\right).$$

Pentru a calcula rentabilitățile ajustate vom folosi coloana `Adjusted` (care apare doar la datele de pe Yahoo Finance !):

```
# Rentabilitatea lunara ajustata pentru dividende
returnMonthAdjusted = function(dat){
  dat$Day = as.numeric(strftime(dat$Date, "%d"))
  dat$Month = as.numeric(strftime(dat$Date, "%m"))
  dat$Year = as.numeric(strftime(dat$Date, "%Y"))

  dat_month_diff = diff(dat$Month)
  dat_month = dat[dat_month_diff == 1, ]

  month_ret = dat_month$Adjusted[-1] / dat_month$Adjusted[-length(dat_month$Adjusted)] - 1
  return(data.frame(date = dat_month$Date[-1], ret = month_ret))
}

google_ret_adj_monthly = returnMonthAdjusted(google_data)
apple_ret_adj_monthly = returnMonthAdjusted(apple_data)
msft_ret_adj_monthly = returnMonthAdjusted(msft_data)

head(cbind(google_ret_adj_monthly$ret, apple_ret_adj_monthly$ret,
            msft_ret_adj_monthly$ret))
      [,1]      [,2]      [,3]
[1,] -0.03900416  0.16670979 -0.03966420
[2,] -0.03978939 -0.07110995 -0.03934825
[3,]  0.21876907 -0.13462939  0.04675180
[4,]  0.26031817  0.10260652  0.02299797
[5,]  0.06087933 -0.07419498 -0.03720931
[6,] -0.02172367  0.15865307  0.03099854

# log returns
google_ret_adj_monthly$cret = log(1+google_ret_adj_monthly$ret)
apple_ret_adj_monthly$cret = log(1+apple_ret_adj_monthly$ret)
msft_ret_adj_monthly$cret = log(1+msft_ret_adj_monthly$ret)
```

