Coletando dados

Lucca Simeoni Pavan *João Carlos de Carvalho*

18 de outubro de 2016

```
knitr::opts_chunk$set(echo = TRUE, cache = TRUE, warning = FALSE, message = FALSE,
    error = FALSE, tidy = TRUE, tidy.opts = list(width.cutoff = 70))
```

Os dados podem ser coletados usando o pacote 'GetHFData' desenvolvido por Perlin (2016). Para maiores detalhes sobre o pacote veja também Perlin and Ramos (2016). Primeiramente baixaremos os *layouts* da base de dados usando o comando `gthf_download_file`.

```
library(GetHFData)
layout_negocios <- "ftp://ftp.bmf.com.br/MarketData/NEG_LAYOUT_portuguese.txt"
ghfd_download_file(layout_negocios, out.file = "layout_negocios")
```

```
##  Attempt 1 - File exists, skipping dl
```

```
layout_oferta_compra <- "ftp://ftp.bmf.com.br/MarketData/OFER_CPA_LAYOUT_portuguese.txt"
ghfd_download_file(layout_oferta_compra, out.file = "layout_oferta_compra")
```

```
##  Attempt 1 - File exists, skipping dl
```

```
layout_oferta_venda <- "ftp://ftp.bmf.com.br/MarketData/OFER_VDA_LAYOUT_portuguese.txt"
ghfd_download_file(layout_oferta_venda, out.file = "layout_oferta_venda")
```

```
##  Attempt 1 - File exists, skipping dl
```

`Attempt 1` e `TRUE` significam que o download na primeira tentativa foi realizado com sucesso. A mensagem `File exists, skipping dl` aparece quando o comando for acionado pela segunda vez e portanto o documento já foi baixado. Os arquivos de *layout* podem ser abertos pelo bloco de notas.

O comando `ghfd_get_ftp_contents` acessa o ftp da Bovespa e retorna um vetor com todos os arquivos relacionadosm à negócios (todos os outros são ignorados).

```
library("GetHFData")
contents_equity <- ghfd_get_ftp_contents(type.market = "equity")
```

```
##
## Reading ftp contents for equity (attempt = 1|10)
```

```
contents_options <- ghfd_get_ftp_contents(type.market = "options")
```

```
##
## Reading ftp contents for options (attempt = 1|10)
```

```
contents_bmf <- ghfd_get_ftp_contents(type.market = "BMF")
```

```
##
## Reading ftp contents for BMF (attempt = 1|10)
```

Usando os comandos `head`e `tail` podemos ver os 6 primiros e 6 últimos elementos dos arquivos baixados anteriormente.

```
head(contents_equity)
```

```
##              files      dates
## 1 NEG_20141103.zip 2014-11-03
## 2 NEG_20141104.zip 2014-11-04
## 3 NEG_20141105.zip 2014-11-05
## 4 NEG_20141106.zip 2014-11-06
## 5 NEG_20141107.zip 2014-11-07
## 6 NEG_20141110.zip 2014-11-10
##                                                                  link
## 1 ftp://ftp.bmf.com.br/marketdata/Bovespa-Vista/NEG_20141103.zip
## 2 ftp://ftp.bmf.com.br/marketdata/Bovespa-Vista/NEG_20141104.zip
## 3 ftp://ftp.bmf.com.br/marketdata/Bovespa-Vista/NEG_20141105.zip
## 4 ftp://ftp.bmf.com.br/marketdata/Bovespa-Vista/NEG_20141106.zip
## 5 ftp://ftp.bmf.com.br/marketdata/Bovespa-Vista/NEG_20141107.zip
## 6 ftp://ftp.bmf.com.br/marketdata/Bovespa-Vista/NEG_20141110.zip
```

```
tail(contents_equity)
```

```
##                files      dates
## 462 NEG_20160823.zip 2016-08-23
## 463 NEG_20160824.zip 2016-08-24
## 464 NEG_20160825.zip 2016-08-25
## 465 NEG_20160826.zip 2016-08-26
## 466 NEG_20160829.zip 2016-08-29
## 467 NEG_20160830.zip 2016-08-30
##                                                                    link
## 462 ftp://ftp.bmf.com.br/marketdata/Bovespa-Vista/NEG_20160823.zip
## 463 ftp://ftp.bmf.com.br/marketdata/Bovespa-Vista/NEG_20160824.zip
## 464 ftp://ftp.bmf.com.br/marketdata/Bovespa-Vista/NEG_20160825.zip
## 465 ftp://ftp.bmf.com.br/marketdata/Bovespa-Vista/NEG_20160826.zip
## 466 ftp://ftp.bmf.com.br/marketdata/Bovespa-Vista/NEG_20160829.zip
## 467 ftp://ftp.bmf.com.br/marketdata/Bovespa-Vista/NEG_20160830.zip
```

O primeiro dia disponível para o mercado de ações (*equity*) é 2014-11-03 e o último é 2016-08-30. Os arquivos `.zip` armazenam dados das transações diárias e obviamente somente de segunda à sexta-feira.

Para sabermos os *tickers* (nomes dos ativos transacionados, ex. para o mercado de ações `PETR4`, é um *ticker* para ações da PETROBRAS) podemos usar o comando `ghfd_get_available_tickers_from_file` que obtém os *tickers* disponíveis de um arquivo baixado do ftp da Bovespa ou podemos usar o comando `ghfd_get_available_tickers_from_ftp` que obtém os *tickers* disponíveis em um mercado e uma data específicos. Os dois comandos apresentam como resultado um vetor numérico com os tickers e outro com o número de transações de cada *ticker*.

```
tickers_equity <- ghfd_get_available_tickers_from_ftp(my.date = "2015-11-03",
    type.market = "equity", max.dl.tries = 10)
```

```
##
## Reading ftp contents for equity (attempt = 1|10) Attempt 1 - File exists, skipping dl
```

```
head(tickers_equity)
```

```
##   tickers n.trades                      f.name
## 1   PETR4    52231 ftp files/NEG_20151103.zip
## 2   ITUB4    50437 ftp files/NEG_20151103.zip
## 3   BVMF3    47214 ftp files/NEG_20151103.zip
## 4   VALE5    41959 ftp files/NEG_20151103.zip
## 5   BBDC4    39403 ftp files/NEG_20151103.zip
## 6   ITSA4    37993 ftp files/NEG_20151103.zip
```

Existem 419 *tickers* para o mercado de ações na data especificada.

Para baixar os dados de transações de alta frequência e agregá-los para análise usamos o comando `ghfd_get_HF_data`. Para exemplo usarei os três *tickers* mais comercializados no mercado de ações em 03/11/2015, coletados no período de 30/06/2016 a 30/08/2016.

```
dados_top3 <- ghfd_get_HF_data(c("PETR4", "ITUB4", "BVMF3"), type.market = "equity",
    first.date = as.Date("2016-06-30"), last.date = as.Date("2016-08-30"),
    first.time = "9:00:00", last.time = "18:00:00", type.output = "agg",
    agg.diff = "1 hour", dl.dir = "ftp files", max.dl.tries = 10, clean.files = FALSE)
```

```
load("dados_top3.Rda")
head(dados_top3, n = 3)
```

```
##   InstrumentSymbol SessionDate       TradeDateTime n.trades last.price
## 1            BVMF3  2016-06-30 2016-06-30 10:00:00     2992      17.63
## 2            BVMF3  2016-06-30 2016-06-30 11:00:00     3642      17.67
## 3            BVMF3  2016-06-30 2016-06-30 12:00:00     2289      17.72
##   weighted.price  period.ret period.ret.volat sum.qtd  sum.vol n.buys
## 1       17.53706 0.021436848     0.0003225179 1523500 26716617   1238
## 2       17.62966 0.001700680     0.0003044433 1200900 21171287   1395
## 3       17.68812 0.002829655     0.0003512668 1156900 20463311   1079
##   n.sells Tradetime
## 1    1754  10:00:00
## 2    2247  11:00:00
## 3    1210  12:00:00
```

```
tail(dados_top3, n = 3)
```

```
##      InstrumentSymbol SessionDate       TradeDateTime n.trades last.price
## 1054            PETR4  2016-08-30 2016-08-30 15:00:00     4943      13.02
## 1055            PETR4  2016-08-30 2016-08-30 16:00:00     5006      13.06
## 1056            PETR4  2016-08-30 2016-08-30 17:00:00      489      13.15
##      weighted.price   period.ret period.ret.volat sum.qtd   sum.vol n.buys
## 1054       13.02425 -0.003062787     0.0003166287 4252300  55382934   1635
```

```
## 1055       13.02341  0.003072197     0.0003043510 5535600  72092146   2506
## 1056       13.09081  0.004583652     0.0003054307 9056300 118554268    184
##      n.sells Tradetime
## 1054    3308  15:00:00
## 1055    2500  16:00:00
## 1056     305  17:00:00
```

Por fim o comando `ghfd_read_file` baixa os dados na sua forma bruta, ou seja apenas lê o arquivo `.zip` baixado do ftp da Bovespa.

```
library("GetHFData")
path <- path.expand("~/artigo_macroeconometria_lucca_joao/ftp files/NEG_20160830.zip")
dados_bruto <- ghfd_read_file(out.file = path, my.assets = NULL, first.time = "10:00:00",
    last.time = "17:00:00", type.output = "raw")
```

```
##  - Imported  713224 lines, 475 unique tickers
##    -> Processing file - Found 713224 lines, 475 unique tickers
```

```
head(dados_bruto)
```

```
## # A tibble: 6 x 10
##   SessionDate InstrumentSymbol TradePrice TradedQuantity    Tradetime
##         <date>            <chr>      <dbl>          <dbl>        <chr>
## 1  2016-08-30           AALC34      32.81            800 16:10:39.669
## 2  2016-08-30           AAPL34      34.50           3600 16:05:22.618
## 3  2016-08-30           AAPL34      34.15           8700 16:10:39.669
## 4  2016-08-30           ABCB10      14.21            500 10:00:57.694
## 5  2016-08-30           ABCB10      14.00           1000 15:01:20.909
## 6  2016-08-30           ABCB10      14.00            400 15:15:49.496
## # ... with 5 more variables: CrossTradeIndicator <int>, BuyMember <dbl>,
## #   SellMember <dbl>, TradeDateTime <time>, TradeSign <dbl>
```

```
tail(dados_bruto)
```

```
## # A tibble: 6 x 10
##   SessionDate InstrumentSymbol TradePrice TradedQuantity    Tradetime
##         <date>            <chr>      <dbl>          <dbl>        <chr>
## 1  2016-08-30           XTED11      22.56             30 16:42:14.335
## 2  2016-08-30           XTED11      22.52             85 16:42:14.335
## 3  2016-08-30           XTED11      22.57            500 16:42:14.335
## 4  2016-08-30           XTED11      22.52              3 16:42:14.335
## 5  2016-08-30           XTED11      22.55              6 16:42:14.335
## 6  2016-08-30           XTED11      22.52            172 16:44:59.661
## # ... with 5 more variables: CrossTradeIndicator <int>, BuyMember <dbl>,
## #   SellMember <dbl>, TradeDateTime <time>, TradeSign <dbl>
```

```
head(dados_bruto[, 5:8])
```

```
## # A tibble: 6 x 4
##      Tradetime CrossTradeIndicator BuyMember SellMember
##          <chr>               <int>     <dbl>      <dbl>
```

```
## 1 16:10:39.669                   0        40         40
## 2 16:05:22.618                   1       238        238
## 3 16:10:39.669                   0        40         40
## 4 10:00:57.694                   0        58        174
## 5 15:01:20.909                   0       735        114
## 6 15:15:49.496                   0        15        114
```

```
tail(dados_bruto[, 9:10])
```

```
## # A tibble: 6 x 2
##         TradeDateTime TradeSign
##                 <time>     <dbl>
## 1 2016-08-30 16:42:14        -1
## 2 2016-08-30 16:42:14        -1
## 3 2016-08-30 16:42:14        -1
## 4 2016-08-30 16:42:14        -1
## 5 2016-08-30 16:42:14        -1
## 6 2016-08-30 16:44:59        -1
```

Referências

Perlin, Marcelo. 2016. *GetHFData: Download and Aggregate High Frequency Trading Data from Bovespa.* https://CRAN.R-project.org/package=GetHFData.

Perlin, Marcelo, and Henrique Ramos. 2016. "GetHFData: A R Package for Downloading and Aggregating High Frequency Trading Data from Bovespa." SSRN Scholarly Paper ID 2824058. Rochester, NY: Social Science Research Network. https://papers.ssrn.com/abstract=2824058.