

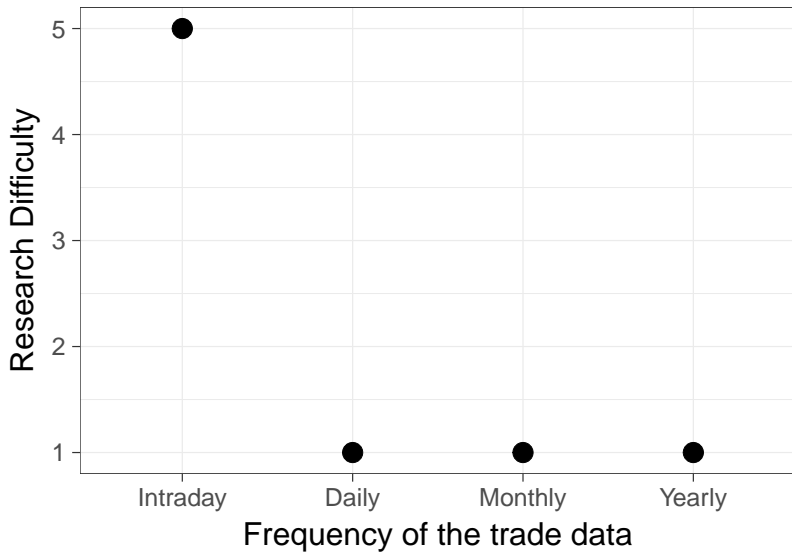
Package GetHFData

R/Finance 2017 - Chicago (Preliminary version)

Marcelo S. Perlin (marcelo.perlin@ufrgs.br)
Universidade Federal do Rio Grande do Sul (UFRGS)
Porto Alegre, Brazil

2017-05-19

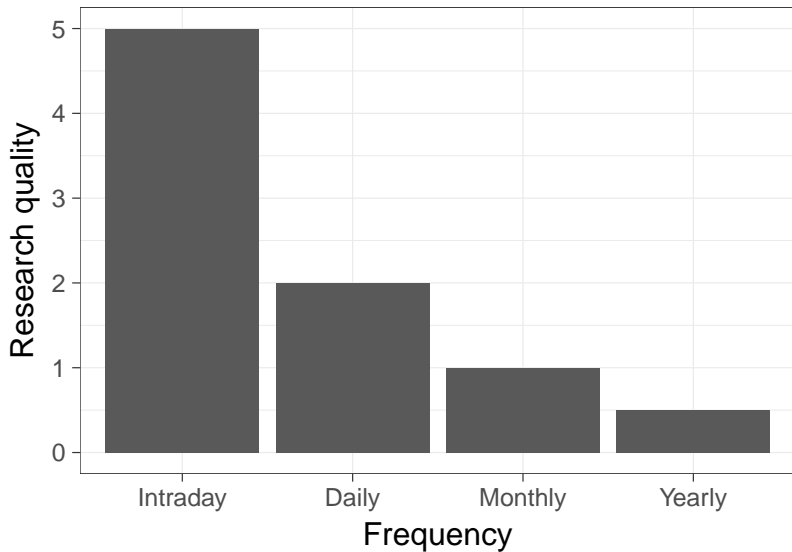
The Problem (1)



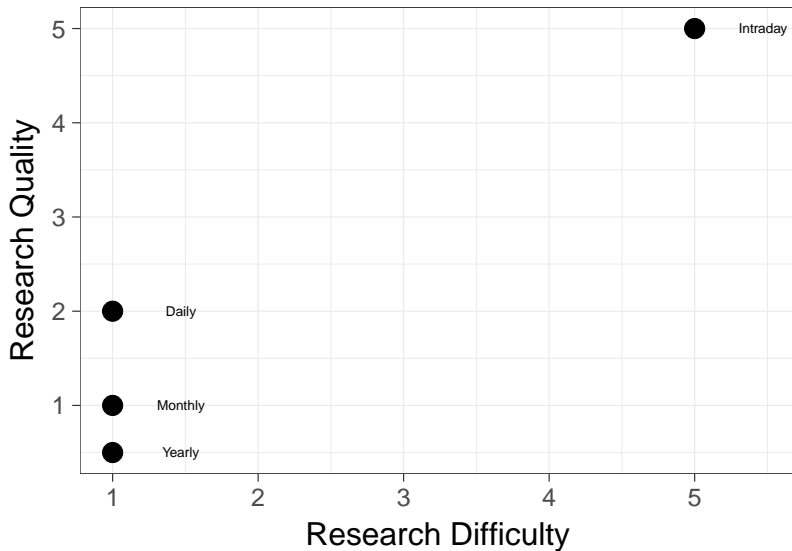
The Problem (2)

- ▶ Intraday data
 - ▶ Large text files, usually separated by trading day
 - ▶ Must know and use proper packages for dealing with these files
- ▶ Daily, monthly, yearly data
 - ▶ Many packages for accessing datasets from Yahoo/Google:
 - ▶ `quantmod` (Ryan 2017)
 - ▶ `BatchGetSymbols` (Perlin 2016)
 - ▶ `tidyquant` (Dancho and Vaughan 2017)
 - ▶ Change of frequency is straightforward with `xts` (Ryan and Ulrich 2014), or `dplyr/data.table` (Wickham and Francois 2016, Dowle and Srinivasan (2017))

The Problem (3)



```
if (!pain) {gain = NULL}
```



About GetHFData

- ▶ CRAN package for importing and aggregating high frequency trade data for the Brazilian stock exchange
- ▶ The package works by creating an interface to Bovespa's ftp site

Main points

- ▶ Users can download raw or aggregated trade data in one line of code
- ▶ Choices of tickers and dates are possible
- ▶ Available datasets:
 - ▶ Equity (spot) trade data and odds-lots
 - ▶ **Derivatives trade data (options, futures)**

Example of usage - Derivative Data

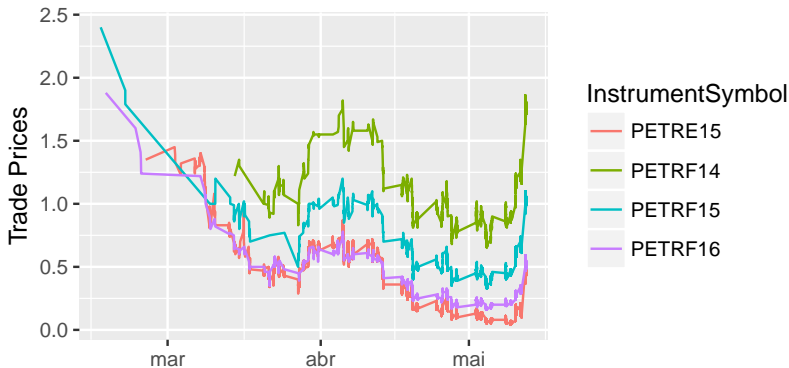
```
library(GetHFData)

last.date <- as.Date('2017-05-14')
first.date <- last.date - 90
type.output <- 'raw'
my.assets <- c('PETRE15', 'PETRF16', 'PETRF15', 'PETRF14')
type.market <- 'options'

df.out <- ghfd_get_HF_data(my.assets = my.assets,
                           type.market = type.market,
                           first.date = first.date,
                           last.date = last.date,
                           type.output = type.output,
                           agg.diff = agg.diff)
```

Example of usage - plotting prices

```
library(ggplot2)
print(ggplot(df.out, aes(x = TradeDateTime,
                        y = TradePrice,
                        color=InstrumentSymbol))+
      geom_line()+labs(y='Trade Prices', x = ''))
```



Example of usage - Calculating Realized volatility

```
first.date <- as.Date("2016-05-25")
last.date <- as.Date("2016-09-30")
type.output <- 'raw'
my.assets <- c("ITSA4", "PETR4", "ITUB4",
               "BBDC4", "ABEV3", "BBSE3")
type.market <- 'equity'

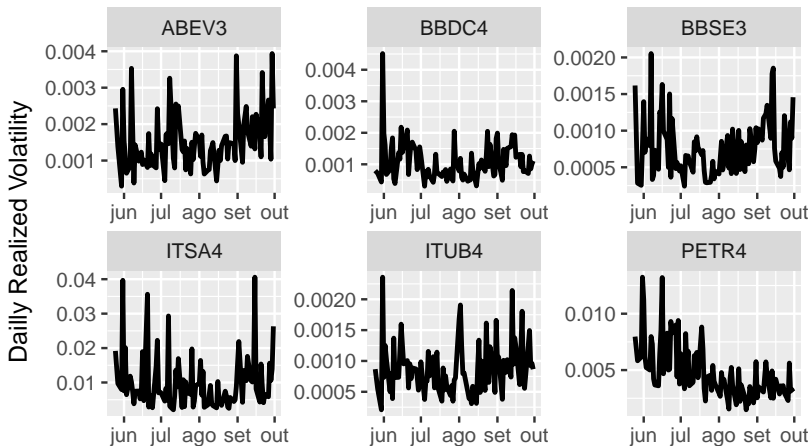
df.out <- ghfd_get_HF_data(my.assets = my.assets,
                           type.market = type.market,
                           first.date = first.date,
                           last.date = last.date,
                           type.output = type.output)
```

Calculating Realized Volatility with highfrequency

```
my.RV.fct <- function(TradePrice, TradeDateTime){  
  require(highfrequency)  
  
  temp.x <- xts(TradePrice, order.by = TradeDateTime)  
  RV <- medRV(temp.x, makeReturns = T)  
  
  return(as.numeric(RV))  
}  
  
library(dplyr)  
RV.tab <- df.out %>%  
  group_by(InstrumentSymbol, SessionDate) %>%  
  summarise(RV = my.RV.fct(TradePrice, TradeDateTime))
```

Results

```
print(ggplot(RV.tab, aes(x=SessionDate, y=RV)) +  
  geom_line(size=1) +  
  facet_wrap(~InstrumentSymbol, scales = 'free') +  
  labs(x='', y='Dailly Realized Volatility'))
```



Next steps

- ▶ Implement access to the order data (very soon)
 - ▶ Allows creating estimates of order book states
 - ▶ quoting strategies and intensity
- ▶ Interface with `highfrequency` (Boudt, Cornelissen, and Payseur 2017)
- ▶ Distribute whole datasets at 5min, 10min and 15min

More details about `GetHFData` are available in Perlin (2017)

Thank you! :)

Processing and Analyzing Financial Data with R

Public online version with full content available here

PROCESSING AND ANALYZING FINANCIAL DATA WITH R

This book introduces the reader to the use of R and RStudio as a platform for processing and analyzing financial data. The book covers all necessary knowledge for using R, from its installation in your computer to the organization and development of scripts. For every chapter, the book presents practical and replicable examples of R code, providing context and facilitating the learning process.

Based on the material, the reader will learn how to download financial data from local files or the Internet, represent and process it using native objects in R, and create tables and figures to report the results in a technical document. The book is organized based on the author's practical experience in scientific research and includes instructions for using the best R packages for each purpose, such as `xtable` and `teeming` for reporting tables, `dplyr` in data processing, and `ggplot2` in creating figures. After presenting the capabilities of R in processing financial data, the last chapter presents three complete and reproducible examples of research in Finance.

This book is recommended for postgraduate researchers and students interested in learning how to use R. No prior knowledge of programming or finance is required to take advantage of this book. After finishing, the reader will have enough knowledge to develop their own scripts autonomously, producing academic documents or data analysis for public and private institutions.



MARCELO S. PERLIN holds a PhD in Finance from Reading University (ICMA Centre - UK) and is a teacher and researcher in the post-graduate program of Business Administration from Universidade Federal do Rio Grande do Sul, south of Brazil. With more than a decade of experience in consulting and production of scientific research, Marcelo acquired vast experience in writing and distributing software, related to the analysis of financial data.

MARCELO S. PERLIN

PROCESSING AND ANALYZING FINANCIAL DATA WITH R



References

Boudt, Kris, Jonathan Cornelissen, and Scott Payseur. 2017. *Highfrequency: Tools for Highfrequency Data Analysis*. <https://CRAN.R-project.org/package=highfrequency>.

Dancho, Matt, and Davis Vaughan. 2017. *Tidyquant: Tidy Quantitative Financial Analysis*. <https://CRAN.R-project.org/package=tidyquant>.

Dowle, Matt, and Arun Srinivasan. 2017. *Data.table: Extension of 'Data.frame'*. <https://CRAN.R-project.org/package=data.table>.

Perlin, Marcelo. 2016. *BatchGetSymbols: Downloads and Organizes Financial Data for Multiple Tickers*. <https://CRAN.R-project.org/package=BatchGetSymbols>.

———. 2017. *GetHFData: Download and Aggregate High Frequency Trading Data from Bovespa*. <https://CRAN.R-project.org/package=GetHFData>.

Ryan, Jeffrey A. 2017. *Quantmod: Quantitative Financial Modelling Framework*. <https://CRAN.R-project.org/package=quantmod>.