# Accessing financial reports and corporate events with GetDFPData

## (Acessando relatórios financeiros e eventos corporativos com GetDFPData)

**Marcelo S. Perlin**[†]
**Guilherme Kirch**[‡]
**Daniel Vancin**[*]

***Abstract***

This paper presents and discusses the contributions and usage of GetDFPData, which is an open and free software for accessing corporate data from the Brazilian financial exchange, B3. The distribution and popularization of an open-source algorithm for gathering and managing financial data can improve finance research and practice in two ways. First, it increases the number and quality of research in accounting and corporate finance. Secondly, it provides retail investors with reliable data that may help their allocation decisions. Initially, we analyze the use of this kind of data in a list of recent publications to show the relevance of financial reports and corporate events data for research in the fields of accounting and finance. Finally, we illustrate the use of GetDFPData in large-scale research, an empirical and reproducible example of a corporate finance study.

***Keywords*:** B3; GetDFPData; R; financial reports; corporate events
**JEL Code**: G30, G10, N26.

## 1. Introduction

Financial statements portray the current financial situation of a company and are at the heart of corporate finance studies and investors' allocation decisions. Easy access to information is a mandatory quality of a well-functioning market. In Brazil, financial reports of companies traded on B3 (formerly known as BM&FBovespa), are available on its website[1] or on the CVM (Comissão de Valores Mobiliários) website[2]. The documents are organized into four systems: ITR/DFP for quarterly/annual financial reports, FRE for corporate events and FCA for a company's registry information. On the B3 website, one can find a simple web interface for accessing this large dataset for a single company.

However, gathering and organizing data for large-scale research, with many companies and many years, requires a significant amount of manual and tedious work. Financial reports and corporate events such as the balance sheet, dividend payments, and others must be downloaded or copied individually for each time period and later aggregated. Changes in the format over time can make this process slow, unreliable, and unreproducible.

Many commercial data vendors offer solutions for accessing clean financial statements and corporate events. Our bibliometric research demonstrates that, in a group of recent publications, approximately 70% of these studies used data from Economatica[3]. Using data from commercial vendors requires the purchase of a license and may not be within reach of some research institutions and retail investors. Moreover, most data vendors do not offer a data format suitable for large-scale research or an API for easier access. Likewise, even when the data is available, it still requires reorganization, adding an unnecessary cost. These types of barriers to data acquisition may impede the development of local research in corporate finance and accounting.

Submitted on March 23, 2019. Revised on June 14, 2019. Accepted on July 22, 2019. Published online in December, 2019. Editor in charge: Cristiano Costa.

[†] (Corresponding author) Escola de Administração, UFRGS, Brasil. E-mail: marcelo.perlin@ufrgs.br

[‡] Escola de Administração, UFRGS, Brasil. E-mail: guilherme.kirch@ufrgs.br

[*] Universidade do Vale do Rio dos Sinos, Brasil. E-mail: daniel_vancin@hotmail.com

[1] http://www.b3.com.br/ (accessed on 25/03/2019).

[2] http://www.cvm.gov.br/ (accessed on 25/03/2019).

[3] https://economatica.com/ (accessed on 25/03/2019).

*Revista Brasileira de Finanças (Online), Rio de Janeiro, Vol. 17, No. 3, September 2019, pp. 85–108*
*ISSN 1679-0731, ISSN online 1984-5146*

GetDFPData provides an open and free interface to all financial statements distributed by B3 and CVM (*Comissão de Valores Mobiliários*). It not only downloads the data but also cleans it, adjusts for inflation, and prepares it for research within a tabular format. Users only need to select companies and a time period to download all of the available data. Up-to-date information about companies traded on B3, including situation (active or inactive), sector, CVM code, and related tickers of traded assets, are available to the user. This information allows researchers to maintain a continuous workflow, from the selection of companies to hypothesis testing, facilitating and increasing the reproducibility of studies.

The software was created by researchers and designed for researchers. The dataset resulting from the application was formatted in line with modern methodologies for data organization and processing, the so-called *tidy data* (Wickham, 2014). Its main principle is to keep data in a row-oriented tabular format. This facilitates visualization and modeling of large financial datasets, without the need for further reorganization.

As with any provider of financial data, quality is key. However, having said that, it is important to understand that all data imported with GetDFPData has been taken directly from the source, B3's DFP and FRE system. Moreover, as is well-known, companies must report and be responsible for the information they provide. Financial markets and institution set strong and direct incentives for companies to release truthful information.

The package works as a mirror for all financial documents available in B3. Unlike other data vendors, the information is downloaded and read directly from the exchange. There is no intermediating compiled database. Therefore, the package benefits from the strong incentives that companies have to provide quality data. However, it is best to be aware that, in the current version, data from GetDFPData is offered as-is, without any error checking.

The main contribution of the software is to make it easy to access financial statements in large-scale studies by setting an open standard of data acquisition and organization (Gandrud, 2013). Researchers and investors unfamiliar with R can use the web version[4] of GetDFPData to download an Excel spreadsheet or CSV files with all the data. The software not only contributes to easier, unrestricted access but also provides more information than most commercial data vendors. History of corporate events such as dividend payments, stock splits and reverse splits, changes in stockholder composition, auditing reports, a company's remuneration policy, family relationships in the company, debt composition, history of governance listings and more are included in the output. The available content should satisfy a large variety of corporate finance and accounting studies.

This paper is organized as follows. First, we discuss the use of financial reports and software in a list of recent publications. There follow instructions for the web interface and a tutorial for installation and usage of GetDFPData in R. We next illustrate the use of our package with a reproducible example of a corporate finance study. We conclude the paper in the last section.

## 2. Literature Review

We surveyed the most important Brazilian journals to show the relevance of financial reports and corporate events data for research in the fields of accounting and finance: *Contabilidade Vista & Revista*, *Revista Contabilidade & Finanças*, *Revista de Contabilidade e Organizações*, *Revista Contemporânea de Contabilidade*, and *Revista Brasileira de Finanças*.

The list was built based on a simple rule: we select accounting and finance journals classified as "A2" in the "Qualis-CAPES" classification system for the period between 2013 and 2016, in the area of Business (*Administração, Contabilidade e Turismo*). We believe that the accounting and finance fields of research would benefit the most from the data provided by GetDFPData. Therefore, we only consider journals which are strictly related to the Finance/Accounting fields, excluding others with more broad topics such as management. The "A2" classification is the highest tier for local journals. The only exception is *Revista Brasileira de Finanças*, classified as "B1" (one rank below "A2"), which is included because it is the official publication of the Brazilian Society of Finance (SBFin) and because of its relevance as an outlet for financial research (Perlin & Portela Santos, 2015). To keep things manageable, we also restrict our survey to five years: 2013-2017.

[4] https://www.msperlin.com/shiny/GetDFPData/ (accessed on 25/03/2019).

In our survey, we look for papers that use financial reports of Brazilian firms as the main input for the data analysis procedure. We collect the following information for each paper that fits this requirement: authors, year of publication, journal title, subject, sample period, number of firms, number of observations, and databases used (data sources).

Table 1 presents the number of papers selected (surveyed) and the total number of papers published in each journal. Research based on financial reports and corporate events data of Brazilian listed companies is quite common. Of the published papers surveyed (477), 105 (22.01%) share this characteristic., *Revista Contabilidade & Finanças* is the most important outlet for this type of research, both in absolute and relative terms. This journal published 34 papers based on financial reports and corporate events data of Brazilian listed companies and this number represents 28.81% of the papers published in the journal from 2013 to 2017. At the other extreme is *Revista Brasileira de Finanças*, with 9 papers out of 85 (10.59%), showing a heterogeneous content in Finance.

**Table 1**
**Papers by Journal**

| journal | surveyed | total | percentage |
|---|---|---|---|
| Contabilidade Vista & Revista | 19 | 80 | 23.7% |
| Revista Brasileira de Finanças | 9 | 85 | 10.6% |
| Revista Contabilidade & Finanças | 34 | 118 | 28.8% |
| Revista de Contabilidade e Organizações | 22 | 82 | 26.8% |
| Revista Contemporânea de Contabilidade | 21 | 112 | 18.8% |

Data sources used in the surveyed papers are listed in Table 2. As we can see, *Economatica* is the most used software: 77 out of 105 papers (73.33%) built their data sets from this commercial source. Far behind in this list are public data sources: CVM (23 papers, 21.90%) and B3[5] (19 papers, 18.10%). The preference for *Economatica* can be likely explained by its data exporting tools. Although not in a suitable format for large scale research, *Economatica* allows the user to export data from multiple firms and several time periods at once. Whereas, on the CVM and B3 websites, data can only be retrieved for one company/period at a time.

**Table 2**
**Database Summary**

| data-base | *N* | percentage |
|---|---|---|
| Economática | 77 | 73.3% |
| CVM | 23 | 21.9% |
| B3 | 19 | 18.1% |
| Thomson | 2 | 1.9% |
| Comdinheiro | 1 | 1.0% |
| EMIS | 1 | 1.0% |
| EmpresasNet | 1 | 1.0% |

In Table 3 we present descriptive statistics of three variables that characterize the samples used in the surveyed papers: number of observations, number of firms, and number of years. On average (at the median) the sample consists of 2,039 (722) observations, 159 (116) firms, and 7 (5) years. These variables vary widely, as indicated by the standard deviation, minimum, and maximum statistics. The characteristics of the samples indicate that researchers can greatly benefit from the proposed software, since it delivers data from several firms and time periods in formats ready to be analyzed by data analysis tools, including R, Stata, SPSS, and others.

Finally, in Table 4 we list the main subjects explored in the surveyed papers. Earnings management is the most common subject: 19 out of 105 papers (18.10%) deal with this research topic. Disclosure (14 papers,

[5] BM&F (Bolsa de Mercadorias e Futuros) was the data source cited in these 19 papers surveyed. Since BM&F no longer exists and is now part of B3, we opted to use the current name of the Brazilian Exchange.

### Table 3
### Descriptive Statistics for Variables Related to the Samples used in the Surveyed Papers

We collected data about Observations (the number of observations in the sample), Firms (the number of firms in the sample), and Years (the number of years in the sample) of the samples used in the surveyed papers. Not all papers reported these three characteristics of their samples. In this table we present descriptive statistics of these three variables that characterize the samples used in the surveyed papers: Observations, Firms, and Years. Mean shows the average value of each column, SD the standard deviation, Min the minimum value, Perc 25 the value in the 25th percentile, Median the value in the 50th percentile, Perc 75 the value in the 75th percentile, and Max the maximum value.

| statistic | observations | firms | years |
|---|---|---|---|
| mean | 2,039 | 159 | 7 |
| standard deviation | 2,691 | 128 | 5 |
| minimum | 64 | 10 | 1 |
| percentile 25 | 340 | 56 | 3 |
| median | 722 | 116 | 5 |
| percentile 75 | 2,444 | 250 | 10 |
| maximum | 11,147 | 655 | 24 |

13.33%), Performance (12 papers, 11.43%), IFRS (11 papers, 10.48%), and Corporate Governance (10 papers, 9.52%) are also very common subjects. Beyond the interest of the researcher and the importance of the subject, the choice of research topic is likely to be influenced by data availability. In this sense, our package could contribute to a greater diversity of research topics, since it provides access to data unavailable in other data sources for a large number of firms, such as executive compensation, family members in the company, transactions with related parties, auditing information, debt structure, and board composition. As shown in Table 4, papers dealing with these data are relatively rare in the Brazilian accounting and finance literature.

### Table 4
### Papers' Main Subjects

For each surveyed paper we collected the main subjects explored by the authors. A paper may deal with more than one subject, so the sum of the Percent (Papers) column is greater than one.

| subject | *N* | percent (papers) |
|---|---|---|
| Earnings Management | 19 | 18.1% |
| Disclosure | 14 | 13.3% |
| Performance | 12 | 11.4% |
| IFRS | 11 | 10.5% |
| Corporate Governance | 10 | 9.5% |
| Market Value | 8 | 7.6% |
| Capital Structure | 7 | 6.7% |
| Dividend Policy | 6 | 5.7% |
| Intangible Assets | 5 | 4.8% |
| Ownership Structure | 5 | 4.8% |
| Value Relevance | 5 | 4.8% |
| Quality of Accounting Information | 4 | 3.8% |
| Executive Compensation | 4 | 3.8% |
| Information Asymmetry | 3 | 2.9% |
| Board of Directors | 3 | 2.9% |
| Financial Crises | 3 | 2.9% |
| Liquidity Demand | 3 | 2.9% |
| Assets Valuation Methods | 3 | 2.9% |
| Bankruptcy Forecasting Methods | 3 | 2.9% |
| Others | 57 | 54.3% |

## 3. Using GetDFPData

The software is distributed as an R package. However, it is important to justify the platform choice, before introducing examples of usage. R is a programming language especially designed for solving data-related problems. The choice of R for distributing `GetDFPData` is justified by its large user base, absence of user or corporate license fees, compatibility with different operating systems and easy distribution of modules (packages) through CRAN, and a user-contributed repository of packages (R Core Team, 2017). Nonetheless, creating and deploying web applications with R is straightforward with the Shiny technology (Chang, Cheng, Allaire, Xie & McPherson, 2017). This makes it easy to create and maintain the web interface of `GetDFPData`. More details about how to install R are available on its webpage[6].

### 3.1 How GetDFPData works

Package `GetDFPData` functions as an interface to all corporate data stored in .xml files on B3's website. For example, see Petrobras at this link[7]. Users should know that the package only mirrors the data available from B3. There are no consistency tests beyond a basic structural check. If an error exists in the data on DFP or FRE, the package will replicate it.

The location of links and characteristics of XML files (type of system, year, . . . ) is available in a GitHub table[8] that is built manually.

When a user of GetDFPData asks for data, this process starts:

1) The GitHub table is read using the internet. The code checks the availability of files regarding the desired datasets (companies/years).

2) Assuming there is a match of data files from the previous step, the code proceeds to download and read all files. Each system (DFP/FRE/FCA) has a different format and a custom function that will parse the XML information. Likewise, each table within the system also has a particular format and function used to read the information.

3) After reading all files, the code structures all datasets in a tabular format and outputs to the user.

The data from GetDFPData is updated annually. In March each year, after the submission of the last DFP document (4T), the Github table file is updated so the new files/datasets are available to all users.

### 3.2 The Web Interface

A web version of GetDFPData was developed and published on the Internet as a shiny application (Chang et al., 2017). It provides a direct and simple interface to the main features of the package. Users can select available companies and date range, and download the data as an Excel spreadsheet or a zipped file with several .csv archives. The code underlying both interfaces is the same, so their output is identical. However, the command line interface offers far more functionality. The web application is available at `https://www.msperlin.com/shiny/GetDFPData/` (accessed on 25/03/2019).

### 3.3 The R interface

In this section, we will merge the R code with text. The code can include comments in single hashtags. The output is identified with double hashtags. Please see the next example, where we run command `print(1:10)` and see its output in the document:

[6] https://www.r-project.org/ (accessed on 25/03/2019).

[7] http://www.rad.cvm.gov.br/enetconsulta/frmDownloadDocumento.aspx?CodigoInstituicao=2&NumeroSequencialDocumento=80929 (accessed on 25/03/2019)

[8] https://github.com/msperlin/GetITRData_auxiliary

```
# this is a comment
print(1:10)
```

```
## [1]  1  2  3  4  5  6  7  8  9 10
```

We will follow the same standard in all future code chunks. All code in this document is reproducible and can be executed on any computer with an R installation. A standalone script with all code executed in this document is available on the author’s homepage.

### 3.4 Installation of GetDFPData

Package `GetDFPData` is available in CRAN in its release (stable) version. After installing R on your computer, the new module (package) is installed with the following command in R’s prompt:

```
# Release version in CRAN
install.packages(’GetDFPData’)
```

This operation may take a while, as several dependent packages are also installed. Once the installation is complete, the package is ready for use. Windows platform users may require the installation of Java 64 bits. Instructions are available at this link: `https://java.com/en/download/faq/java_win64bit.xml`.

### 3.5 Using GetDFPData

The starting point of GetDFPData is a search for information on current companies. We can download the table and check its contents using function `gdfpd.get.info.companies` to demonstrate its use. We set the argument `type.data = 'companies'` so that it only returns information about companies.

```
library(GetDFPData)
library(tibble)

df.info <- gdfpd.get.info.companies(type.data = ’companies’)
```

```
## Found cache file. Loading data..
```

```
glimpse(df.info)
```

```
## Observations: 528
## Variables: 16
## $ name.company <chr> "521 PARTICIPAÇOES S.A. - ...
## $ id.company <int> 16330, 16284, 21725, 18970...
## $ cnpj <dbl> 1.547749e+12, 1.851771e+12...
## $ date.registration <date> 1997-07-11, 1997-05-30, 2...
## $ date.constitution <date> 1996-07-30, 1997-04-02, 2...
## $ city <chr> "RIO DE JANEIRO", "RIO DE ...
## $ estate <chr> "RJ", "RJ", "SP", "SP", "R...
## $ situation <chr> "ATIVO", "ATIVO", "ATIVO",...
## $ situation.operations <chr> "LIQUIDAÇÃO EXTRAJUDICIAL"...
## $ listing.segment <chr> NA, "Tradicional", "Tradic...
## $ main.sector <chr> NA, "Outros", "Saúde", "Ut...
## $ sub.sector <chr> NA, "Outros", "Serv.Méd.Ho...
## $ segment <chr> NA, "Outros", "Serv.Méd.Ho...
## $ tickers <chr> NA, "QVQP3B", "ADHM3", "TI...
## $ first.date <date> 1998-12-31, 2001-12-31, 2...
## $ last.date <date> 2018-12-31, 2018-12-31, 2...
```

We find the IDs of companies, their official names, dates of registration and constitution, sector and subsectors of activities, current governance listing, traded tickers and current operating situation (active or not). Moreover, this makes it an excellent source of information for the data selection stage of research, since it is easy to filter companies based on factors like dates, sectors, tickers or governance listings.

The current number of active and inactive companies, as of 2019-07-15, is available in column `situation`. We can see that there are 526 active companies, 2 canceled and NA with an administrative decision to suspend activities.

We can also look at the distribution sectors of companies trading on B3:

```
t.sector <- table(df.info$main.sector)
print(as.data.frame(sort(t.sector, decreasing = TRUE)))
```

```
## Var1 Freq
## 1 Financeiro 86
## 2 Consumo Cíclico 81
## 3 Bens Industriais 77
## 4 Utilidade Pública 67
## 5 Materiais Básicos 32
## 6 Consumo não Cíclico 25
## 7 Outros 25
## 8 Saúde 19
## 9 Financeiro e Outros 12
## 10 Não Classificados 10
## 11 Petróleo. Gás e Biocombustíveis 10
## 12 Tecnologia da Informação 7
## 13 Telecomunicações 5
```

We find a large proportion of companies in the financial sector, followed by cyclical consumption. Going further, the distribution of current governance listings are also available:

```
t.listings <- table(df.info$listing.segment)
print(as.data.frame(sort(t.listings, decreasing = TRUE)))
```

```
## Var1 Freq
## 1 Tradicional 252
## 2 Novo Mercado 140
## 3 Corporate Governance - Level 1 27
## 4 Corporate Governance - Level 2 19
## 5 Bovespa Mais 16
## 6 Bovespa Mais - Level 2 2
```

As expected, most companies are not included in governance listings (`'Tradicional'`). The level of `'Novo Mercado'` is the governance listing with the highest number of companies.

Tickers of companies are also available in column `tickers`. As an example, we can check all currently traded stocks of Gerdau:

```
idx <- which(df.info$name.company == 'GERDAU S.A.')
tickers.Gerdau <- df.info$tickers[idx]
print(tickers.Gerdau)
```

```
## [1] "GGBR3;GGBR4"
```

Tickers are separated by a semi-colon. This information can be used to match stock prices and companies in external databases.

### 3.5.1 Finding names of companies

In the database, every company can be identified by its official name or CVM ID. We decided to select companies by their name, as it was more convenient. Function `gdfp.search.company` allows users to search for the official name of a company. Given a search text, it will look for a partial match in the names with all available companies. In its use, Latin characters and upper/lower cases are ignored. As an example, we can find the official name of Ambev, one of the largest companies in Brazil:

```
library(GetDFPData)

my.names <- gdfpd.search.company('ambev')
```

```
## Found cache file. Loading data..
##
## Found 1 companies:
## AMBEV S.A. | situation = ATIVO | first date = 2012-12-31 | last date - 2018-12-31
```

The official name in Bovespa records is AMBEV S.A.. Also, the data for annual statements are available from 2012-12-31 to 2018-12-31. Notice how the situation of the company (active or canceled) is also provided.

### 3.5.2 Downloading financial information

We use the main function of the package, `gdfpd.GetDFPData`, to download information. In its basic use, we set the official name as input name.companies and the time period as inputs `first.date` and `last.date`. Let us try it for Ambev:

```
name.companies <- 'AMBEV S.A.'
first.date <- '2014-12-31'
last.date <- '2017-12-31'

df.reports <- gdfpd.GetDFPData(name.companies = name.companies,
first.date = first.date,
last.date = last.date)
```

```
## Found cache file. Loading data..
##
## Downloading data for 1 companies
## First Date: 2014-12-31
## Laste Date: 2017-12-31
## Inflation index: dollar
##
## Downloading inflation data
## Caching inflation RDATA into tempdir() Done
##
## Inputs looking good! Starting download of files:
##
## AMBEV S.A.
## Available periods: 2017-12-31 2016-12-31 2015-12-31 2014-12-31
##
##
## Processing 23264 - AMBEV S.A.
## Finding info from Bovespa
## Found BOV cache file
## Processing 23264 - AMBEV S.A. | date 2017-12-31
## Acessing DFP data | Found DFP cache file
## Acessing FRE data | Found FRE cache file
## Acessing FCA data | Found FCA cache file
## Processing 23264 - AMBEV S.A. | date 2016-12-31
## Acessing DFP data | Found DFP cache file
## Acessing FRE data | Found FRE cache file
## Acessing FCA data | Found FCA cache file
## Processing 23264 - AMBEV S.A. | date 2015-12-31
## Acessing DFP data | Found DFP cache file
## Acessing FRE data | Found FRE cache file
## Acessing FCA data | Found FCA cache file
## Processing 23264 - AMBEV S.A. | date 2014-12-31
## Acessing DFP data | Found DFP cache file
## Acessing FRE data | Found FRE cache file
## Acessing FCA data | Found FCA cache file
```

Messages from using `gdfpd.GetDFPData` report the stages of the software, from early data acquisition of the reference table from Github to the download and parsing of B3 files. Notice that files from three systems are accessed: DFP (*Demonstrações Financeiras Padronizadas*), FRE (*Formulário de Referência*) and FCA (*Formulário Cadastral*). Also, notice how using a local caching system significantly improves the speed of the software, at a low memory cost. The caching system, when active, ensures that all data queries are saved locally and, if repeated, the information is read from local files, rather than the Internet.

The resulting object from `gdfpd.GetDFPData` is a `tibble`, a tabular object, which allows for list columns (Wickham & Grolemund, 2016). When using it, we can save any kind of financial information in a tabular structure. Next, we will be presenting a description of the most interesting columns available at object `df.reports`. We omit the description of other columns for brevity.

**company.name** The official name of the company (e.g. `'AMBEV S.A.'` ).
**company.code** The CVM code of company (e.g. 9512).
**cnpj** The CNPJ of the company (official government registration ID).
**date.company.constitution** Date of company constitution.
**date.cvm.registration** Date of company registration in CVM.
**company.tickers** Tickers of stocks traded in B3.
**min.date** The earliest date of available financial reports, given user choice of the time period (e.g. `'2011-12-31'`).
**max.date** The most recent date of the available financial report, given user choice of the time period (e.g. `'2016-12-31'`).
**n.periods** Number of years between min.date and max.date (e.g `'4'`).

**company.segment** The corporate governance listing of the company (e.g. `'Novo Mercado'`).

**current.stockholders** The names and percentage holdings of the current stockholders.

**current.stockcomposition** The current composition of stocks, a number of ordinary and preferred shares.

**fr.assets** Financial report for accounting assets of firms (*ativo*), for all time periods. An additional column featuring consolidated values is also available.

**fr.liabilities** Financial report for liabilities of firms (*passivo*), for all time periods. The result is an object of class data.frame with same columns as **fr.assets**.

**fr.income** Financial reports for income statements (*demonstração do resultado do exercício*), for all time periods.

**fr.cashflow** Financial report for cashflow statements (*demonstração do fluxo de caixa*), for all time periods.

**history.dividends** The history of cash payouts including dividends and JSCP (*juros sobre capital próprio*).

**history.stockholders** The history of stockholders for the selected time periods.

**history.capital.issues** The history of capital issues for the company, for all time periods.

**history.mkt.value** The history of the market value of the company.

**history.capital.increases** Historical records of capital increases.

**history.capital.reductions** Historical records of capital reductions.

**history.stock.repurchases** Contains historical records of equity repurchases.

**history.compensation** History of corporate compensation at different hierarchy levels.

**history.compensation.summary** A summary of compensation at different corporate levels.

**history.debt.composition** History of debt composition of the company.

**history.board.composition** History of board composition.

**history.family.relations** History of family relationships within the company.

All tables are indexed by time with column `ref.date` and by firm with `company.name`. We can track all of the different corporate events, for different companies by using it, thus, making it a rich source of information for research. Also, it is worth pointing out that object `df.reports` has only one row because we asked for data of only one company. The number of rows increases with the number of companies, as we will soon learn with the next example.

All financial statements for the different years are available within `df.reports`. For example, the income statements for all desired years of *AMBEV* are:

```r
df.income.long <- df.reports
$
fr.income[[1]]

glimpse(df.income.long)
```

```
## Observations: 104
## Variables: 6
## $ name.company <chr> "AMBEV S.A.", "AMBEV S.A.", ...
## $ ref.date <date> 2017-12-31, 2017-12-31, 201...
## $ acc.number <chr> "3.01", "3.02", "3.03", "3.0...
## $ acc.desc <chr> "Receita de Venda de Bens e/...
## $ acc.value <dbl> 21730363, -11483719, 1024664...
## $ acc.value.infl.adj <dbl> 6570225.25, -3472128.86, 309...
```

The resulting data frame is in the long format, ready for processing. In the long format, financial statements of different years are stacked row-wise. In the wide format, we have years as columns, where every new year appends a column to the table. If desired, the wide format can be forced in the resulting financial reports with function `gdfpd.convert.to.wide`. See the next example:

```
df.income.wide <- gdfpd.convert.to.wide(df.income.long)

glimpse(df.income.wide)
```

```
## Observations: 26
## Variables: 7
## $ acc.number <chr> "3.01", "3.02", "3.03", "3.04", "3...
## $ acc.desc <chr> "Receita de Venda de Bens e/ou Ser...
## $ name.company <chr> "AMBEV S.A.", "AMBEV S.A.", "AMBEV...
## $ ‘2014-12-31‘ <dbl> 20014913, -9704361, 10310552, 3729...
## $ ‘2015-12-31‘ <dbl> 22106965, -11431627, 10675338, 337...
## $ ‘2016-12-31‘ <dbl> 20634193, -11263620, 9370573, 4006...
## $ ‘2017-12-31‘ <dbl> 21730363, -11483719, 10246644, 113...
```

However, beware that, given the limitations of the wide format, it only works for financial reports, that is, columns in `df.reports` with names starting with a `'fr.'`, such as `fr.assets` and `fr.assets.consolidated`.

### 3.5.3 Downloading financial information for several companies

`GetDFPData` is specifically designed for handling large-scale download of data. We now build a case with three randomly-selected active companies:

```
set.seed(1)
active.companies <- df.info$name.company[df.info$situation == ’ATIVO’]
my.companies <- sample(active.companies, 3)

first.date <- ’2012-01-31’
last.date <- ’2018-01-01’

df.reports <- gdfpd.GetDFPData(name.companies = my.companies,
first.date = first.date,
last.date = last.date)
```

So, we can now check the resulting `tibble`:

```
glimpse(df.reports)
```

```
## Observations: 3
## Variables: 46
## $ company.name <chr> "COMPANHIA BRA...
## $ company.code <int> 8672, 8672, 8672
## $ cnpj <chr> "4750841100015...
## $ date.company.constitution <date> 1981-11-11, 1...
## $ date.cvm.registration <date> 1995-04-04, 1...
## $ company.tickers <chr> "JPSA3", "JPSA...
## $ min.date <date> 2012-12-31, 2...
## $ max.date <date> 2017-12-31, 2...
## $ n.periods <int> 6, 6, 6
## $ company.segment <chr> "Corporate Gov...
## $ current.stockholders <list> [<data.frame[...
## $ current.stock.composition <list> [<data.frame[...
## $ history.files <list> [<data.frame[...
## $ fr.assets <list> [<data.frame[...
## $ fr.liabilities <list> [<data.frame[...
## $ fr.income <list> [<data.frame[...
## $ fr.cashflow <list> [<data.frame[...
## $ fr.value <list> [<data.frame[...
## $ fr.assets.consolidated <list> [<data.frame[...
## $ fr.liabilities.consolidated <list> [<data.frame[...
## $ fr.income.consolidated <list> [<data.frame[...
## $ fr.cashflow.consolidated <list> [<data.frame[...
## $ fr.value.consolidated <list> [<data.frame[...
## $ fr.auditing.report <list> [<data.frame[...
## $ history.dividends <list> [<data.frame[...
## $ history.stockholders <list> [<data.frame[...
## $ history.capital.issues <list> [<data.frame[...
## $ history.mkt.value <list> [<data.frame[...
## $ history.capital.increases <list> [<data.frame[...
## $ history.capital.reductions <list> [<data.frame[...
## $ history.stock.repurchases <list> [<data.frame[...
## $ history.other.stock.events <list> [<data.frame[...
## $ history.compensation <list> [<data.frame[...
## $ history.compensation.summary <list> [<data.frame[...
## $ history.transactions.related <list> [<data.frame[...
## $ history.debt.composition <list> [<data.frame[...
## $ history.governance.listings <list> [<data.frame[...
## $ history.board.composition <list> [<data.frame[...
## $ history.committee.composition <list> [<data.frame[...
## $ history.family.relations <list> [<data.frame[...
## $ history.family.related.companies <list> [<data.frame[...
## $ history.auditing <list> [<data.frame[...
## $ history.responsible.docs <list> [<data.frame[...
## $ history.stocks.details <list> [<data.frame[...
## $ history.dividends.details <list> [<data.frame[...
## $ history.intangible <list> [NULL, NULL, ...
```

Every row of df.reports provides information for one company. Metadata about the corresponding data frames such as min/max dates is available in the first columns. Keeping a tabular structure facilitates the organization and future processing of all financial data.

Each company has its own tables in df.reports. We can merge the tables into a single object by asking R to *bind* the rows of all elements available within a column of df.reports. See the following example, in which we bind the rows of fr.assets data frame for all three companies:

```r
library(dplyr)
# bind rows of assets for different companies
df.assets <- bind_rows(df.reports$fr.assets)

# check result
glimpse(df.assets)
```

```
## Observations: 288
## Variables: 6
## $ name.company <chr> "COMPANHIA BRASILEIRA DE DIS...
## $ ref.date <date> 2017-12-31, 2017-12-31, 201...
## $ acc.number <chr> "1", "1.01", "1.01.01", "1.0...
## $ acc.desc <chr> "Ativo Total", "Ativo Circul...
## $ acc.value <dbl> 22978000, 9175000, 2868000, ...
## $ acc.value.infl.adj <dbl> 6947451.2, 2774082.4, 867146...
```

If we go further we can use `df.assets` to build a simple table with the total value for each company’s assets, for each year:

```
library(dplyr)

my.tab <- df.assets %>%
 filter(acc.desc == ’Ativo Total’) %>%
 group_by(name.company, ref.date) %>%
 summarise(Total.Assets = acc.value)

print(my.tab)
```

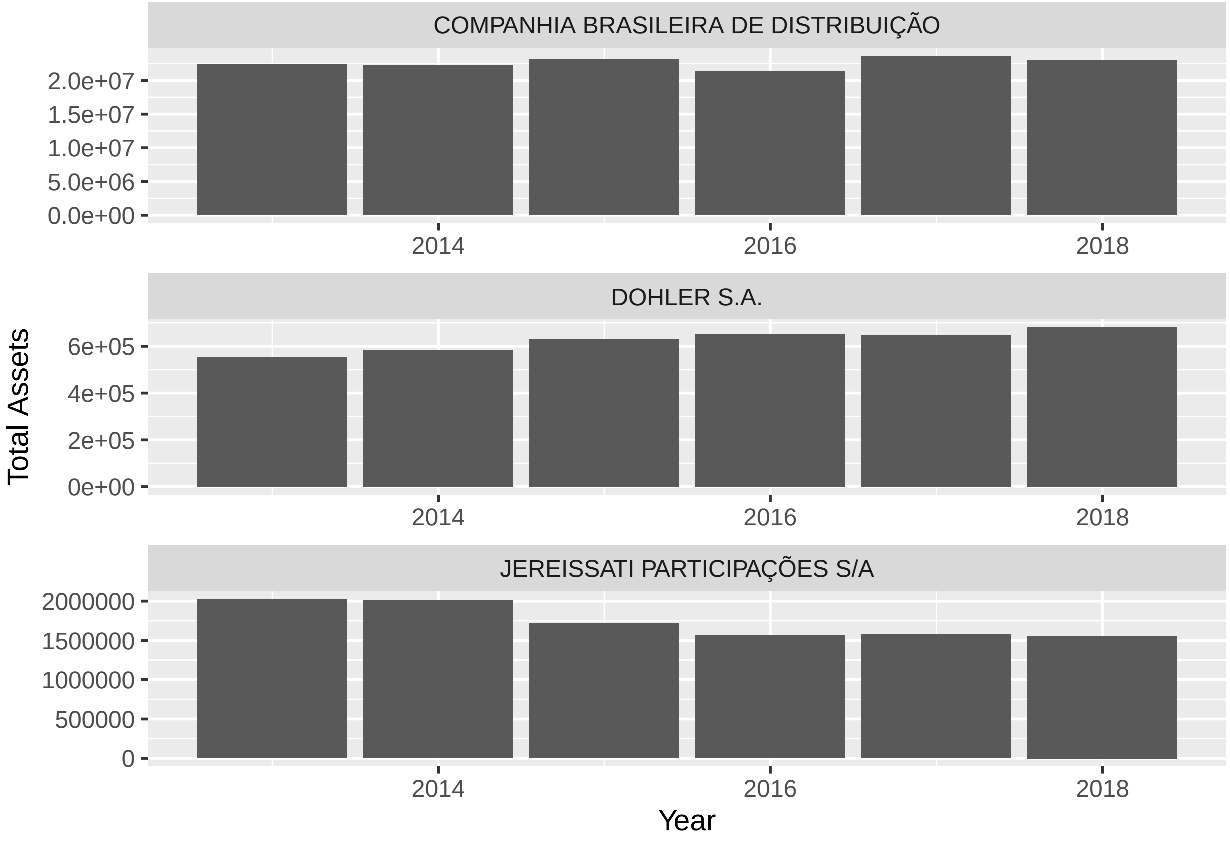
```
## # A tibble: 18 x 3
## # Groups: name.company [3]
## name.company ref.date Total.Assets
## <chr> <date> <dbl>
## 1 COMPANHIA BRASILEIRA DE DISTRIBU~ 2012-12-31 22444808
## 2 COMPANHIA BRASILEIRA DE DISTRIBU~ 2013-12-31 22214075
## 3 COMPANHIA BRASILEIRA DE DISTRIBU~ 2014-12-31 23226000
## 4 COMPANHIA BRASILEIRA DE DISTRIBU~ 2015-12-31 21399000
## 5 COMPANHIA BRASILEIRA DE DISTRIBU~ 2016-12-31 23660000
## 6 COMPANHIA BRASILEIRA DE DISTRIBU~ 2017-12-31 22978000
## 7 DOHLER S.A. 2012-12-31 554890
## 8 DOHLER S.A. 2013-12-31 581937
## 9 DOHLER S.A. 2014-12-31 629084
## 10 DOHLER S.A. 2015-12-31 649793
## 11 DOHLER S.A. 2016-12-31 647460
## 12 DOHLER S.A. 2017-12-31 680821
## 13 JEREISSATI PARTICIPAÇÕES S/A 2012-12-31 2028504
## 14 JEREISSATI PARTICIPAÇÕES S/A 2013-12-31 2013354
## 15 JEREISSATI PARTICIPAÇÕES S/A 2014-12-31 1715483
## 16 JEREISSATI PARTICIPAÇÕES S/A 2015-12-31 1561651
## 17 JEREISSATI PARTICIPAÇÕES S/A 2016-12-31 1573714
## 18 JEREISSATI PARTICIPAÇÕES S/A 2017-12-31 1552707
```

We can now visualize the dataset with a figure:

```
library(ggplot2)

p <- ggplot(my.tab, aes(x = ref.date, y=Total.Assets)) +
 geom_col() + facet_wrap(~name.company, nrow = 3, scales = ’free’) +
```

**Figure 1**
**Total Assets for Three Randomly Selected Companies**



```
 labs(y = ’Total Assets’, x = ’Year’)

print(p)
```

### 3.6 Exporting financial data

The package includes function `gdfpd.export.DFP.data` to export the financial data in *Excel* or *csv* format, which can help a user who needs to import the data in other software. For the Excel file, each table available in the output of function `gdfpd.GetDFPData` is stored as a sheet in the Excel document. As for the *csv* format, each table becomes a single *.csv* file and all CSV files are stored in a single zipped archive.

Next, we present an example of saving the output as an Excel file.

```
my.basename <- ’MyExcelData’
my.type.export <- ’xlsx’

gdfpd.export.DFP.data(df.reports,
base.file.name = my.basename,
type.export = my.type.export)
```

## 4. Example of usage: Family companies and financial performance

We select a relatively simple and timely topic: **the relationship between family companies and financial performance**. We point out that raw data regarding family ties in companies is unavailable in other popular software programs. Our main goal is to illustrate the use of the package, keeping matters related to the method as simple and direct as possible. In other words, we prefer simplicity over scientific rigor. We provide most code used in the study in the text, making it reproducible. Examples of other academic uses of our package, such as the exercises and tests involving firms' financial and economic indices, for example, are available upon request.

### 4.1 Motivation

One of the greatest concerns of corporate law is the protection of minority interests in large public companies. Often, controlling shareholders will seek private benefits, to the detriment of minority shareholders. Large shareholders often have substantial control and influence over firm matters, and agency theory suggests they have powerful incentives to consume the firm's resources, since they bear only a fraction of the total cost (Anderson & Reeb, 2004).

Specifically, founding-family ownership and control of public firms are commonly perceived as a less efficient, or at the very least, a less profitable, ownership structure than dispersed ownership (Anderson & Reeb, 2003). According to the authors, families may pursue actions that maximize their personal utility, and many of these same actions may cause poor firm performance. Additionally, families also often limit executive management positions to family members, suggesting a restricted labor pool from which to obtain qualified professionals.

On the other hand, families have strong incentives to monitor and mitigate the free rider problem that characterizes well-diversified corporations, since their wealth is closely linked to the company value (Anderson & Reeb, 2003). According to these authors, families maintain a long-term presence in their firms. Thus, firm survival is of great concern to them. This long-term perspective suggests that families are long-term value maximization advocates, which is known to raise the value of a firm.

However, it remains an open question (Villalonga & Amit, 2006) whether family firms are more or less valuable than non-family firms. As pointed out by Andres (2008), recent empirical evidence suggests that founding-family ownership is associated with superior firm performance when compared to widely-held companies, both in terms of accounting performance and market valuation.

### 4.2 Methodology

We analyze a large sample of boards of directors in Brazil to test if a founding-family company has a better/worse performance. The boards of directors are governance instruments, which serve important functions for companies, including management monitoring on behalf of shareholders. Specifically, we test if family presence on the board affects firm financial performance. To do this we estimate the following (fixed effects panel data) model:

$$\text{Performance}_{i,t} = \phi_1 \times \text{Family}_{i,t} + \phi_2 \times \text{Family}_{i,t}^2 + \beta \times \text{Controls} + \alpha_i + \mu_t + \varepsilon_{i,t}$$

where the main variables of interest are

$\text{Performance}_{i,t}$: return on assets (ROA) and return on equity (ROE) for company $i$, year $t$.

$\text{Family}_{i,t}$: the percentage of participation of family members in boards for company $i$, year $t$.

and the controls variables are

$\text{Size}_{i,t}$: log of total assets, company $i$, year $t$.

$\text{Age}_{i,t}$: log of companies age, company $i$, year $t$.

$\text{Risk}_{i,t}$: standard deviation of daily returns of most liquid stock, calculated by year and by company.

$\text{Leverage}_{i,t}$: ratio of total debt to total assets.

Note that our main variable, Family$_{i,t}$, enters the model in level and in its square. This last term is included to deal with non-linearities in the relationship between family and firm performance – see Anderson & Reeb (2003) and Andres (2008). As family presence on the board changes little over time, we also estimate the model by OLS with Industry (Sector) dummies as additional covariates, as is usual in this literature (Anderson & Reeb, 2003; Villalonga & Amit, 2006; Andres, 2008).

Our definitions of ROA and ROE follow the literature. ROA is the ratio between EBITDA and total assets, while ROE is the ratio between net income and book equity. We opt to avoid loss of observations to calculate risk using daily share returns during the current year, instead of monthly share returns for the prior 60 months – see, for example, Anderson & Reeb (2003), Andres (2008) and Villalonga & Amit (2006). Since these three variables exhibit large outliers, we winsorize them at both tails at the 2.5% level. The definitions of all other variables follow the literature and are specified above.

### 4.3 The Code

The research proposed in this section requires three datasets: 1) financial reports, 2) information about the participation of family members in companies and 3) stock prices of companies. For 1) and 2) we can use `GetDFPData` to retrieve the datasets from B3. For 3), we can use the package `BatchGetSymbols` to download stock data directly from the internet.

Our sample selection procedure follows the aforementioned studies. First, we filter the data for companies that only have tickers, that is, are tradeable on the equity market, and those not belonging to the financial and utility sectors. This is a common filter in corporate finance studies, as these sectors have specific dynamics related to their performance measured by ROE and ROA.

```
library(GetDFPData)
library(dplyr)
library(stringr)

# get info table
df.info <- gdfpd.get.info.companies(type.data = 'companies')
```

```
## Found cache file. Loading data..
```

```
# filter companies
df.info <- df.info %>%
 filter(!is.na(tickers),
!(main.sector %in% c('Financeiro e Outros', 'Utilidade Pública')))
```

We find 305 companies with this simple filter. Now, we download data for the selected companies. Beware that this is a lengthy download and may take several hours to accomplish.

```
# set options
first.date <- '2012-01-31'
last.date <- '2018-01-01'
my.companies <- df.info$name.company

# get data
df.reports <- gdfpd.GetDFPData(name.companies = my.companies,
first.date = first.date,
last.date = last.date)
```

The next step is the download of stock data with package `BatchGetSymbols` and the calculation of our volatility measure, $risk_{i,t}$. Before downloading the data, we need to find all the tickers of all companies. In the case of more than one stock for a company, we use the asset with highest financial volume.

```
library(BatchGetSymbols)
library(lubridate)

get.ticker.df <- function(df.in) {
# Gets ticker string and organizes it in another data_frame
require(stringr)
require(dplyr)

 temp.split <- str_split(df.in$tickers, ’;’)[[1]]

 temp.df <- tibble(name.company = df.in$name.company,
ticker = temp.split)

return(temp.df)
}

# get dataframe with tickers
my.l <- by(data = df.info,
INDICES = df.info$name.company,
FUN = get.ticker.df)

# bind into one dataframe
df.tickers <- do.call(bind_rows, my.l)

# add .SA for yahoo finance
my.tickers <- paste0(df.tickers$ticker, ’.SA’)

# get data
df.stocks <- BatchGetSymbols(tickers = my.tickers,
first.date = ’2010-01-01’,
last.date = ’2017-01-01’,
bench.ticker = ’^BVSP’,
thresh.bad.data = 0.5)[[2]]

# replace .SA (keep same notation)
df.stocks$ticker <- str_replace_all(df.stocks$ticker,
fixed(’.SA’), ’’)

# join with ticker df
df.stocks <- inner_join(df.tickers, df.stocks)

# find selected tickers by volume
tab.volume <- df.stocks %>%
 group_by(name.company, ticker) %>%
 summarise(mean.volume = mean(volume, na.rm = TRUE)) %>%
 group_by(name.company) %>%
 summarise(selected.ticker = ticker[which.max(mean.volume)])

df.stocks <- df.stocks[df.stocks$ticker %in% tab.volume$selected.ticker, ]

# calculate volatility measure
df.volat <- df.stocks %>%
 mutate(ref.date = as.Date(paste0(year(ref.date),’-12-31’))) %>%
 group_by(name.company, ref.date) %>%
 summarise(risk = sd(ret.adjusted.prices, na.rm = TRUE))
```

The last step in the data acquisition stage is to bind all financial reports in a single data frame.

```
get.col <- function(df.in, my.col) {
# function to bind all rows (dataframes) of a tibble
return(do.call(rbind, df.in[[my.col]]))
}

# financial reports
df.fr <- bind_rows(get.col(df.reports, ’fr.income’),
get.col(df.reports, ’fr.assets’),
get.col(df.reports, ’fr.cashflow’),
get.col(df.reports, ’fr.liabilities’))

# board composition and family participation
df.board <- get.col(df.reports, ’history.board.composition’)
df.family <- get.col(df.reports, ’history.family.relations’)
```

Now that all of the raw data is available, we can proceed to calculate our variables of interest. The next code will process `df.family` and `df.board`, which contains information about board composition and family members of companies and output a reference table that shows, for each company and year, the percentage of board seats held by family members.

```
ref.tab <- df.family %>%
 group_by(name.company, ref.date) %>%
 summarise(persons = str_c(unique(person.name),
collapse = ’ ; ’)) %>%
 merge(df.board) %>%
 mutate(dummy = person.name %in% str_split(persons,
fixed(’ ; ’),
simplify = TRUE)) %>%
 group_by(name.company, ref.date) %>%
 summarise(family.idx = mean(dummy)) %>%
 glimpse()
```

```
## Observations: 1,063
## Variables: 3
## Groups: name.company [171]
## $ name.company <chr> "ADVANCED DIGITAL HEALTH MEDICINA ...
## $ ref.date <date> 2014-12-31, 2009-12-31, 2010-12-3...
## $ family.idx <dbl> 0.20000000, 0.12500000, 0.16000000...
```

Finally, we build our final data frame containing all variables necessary for our panel and OLS model:

```
get.acc <- function(acc.value, search.vec, my.str) {
# get account value from fr.dataframe

 out.value <- acc.value[search.vec == my.str]

# test if it exists
if (length(out.value) == 0) out.value <- NA

return(out.value)

}

# build final table
model.table <- df.fr %>%
 group_by(name.company, ref.date) %>%
 summarise(profit.loss = get.acc(acc.value, acc.number, '3.09'),
total.assets = get.acc(acc.value, acc.number, '1'),
capital = get.acc(acc.value, acc.number, '2.03'),
total.sales = get.acc(acc.value, acc.number, '3.01'),
EBIT = get.acc(acc.value, acc.number, '3.05'),
deprec.amort = get.acc(acc.value, acc.number, '6.01.01.02'),
EBITDA = EBIT + deprec.amort,
short.term.debt = get.acc(acc.value, acc.number, '2.01.04'),
long.term.debt = get.acc(acc.value, acc.number, '2.02.01'),
total.debt = short.term.debt + long.term.debt,
size = log(total.assets),
leverage = total.debt/total.assets,
ROA = EBITDA/total.assets,
ROE = profit.loss/capital) %>%
 inner_join(df.volat) %>%
 inner_join(ref.tab) %>%
 inner_join(df.info) %>%
 filter(total.assets > 0,
 total.sales > 0,
 capital > 0) %>%
 inner_join(select(df.reports, company.name,
 date.company.constitution,
 date.cvm.registration),
by = c('name.company' = 'company.name' )) %>%
 mutate(age = log(as.numeric(ref.date - date.company.constitution)/365)) %>%
 select(name.company, ref.date, ROA,
 ROE, size, age, risk, leverage, family.idx, main.sector) %>%
 glimpse()
```

```
## Observations: 610
## Variables: 10
## Groups: name.company [109]
## $ name.company <chr> "ALPARGATAS SA", "ALPARGATAS SA", ...
## $ ref.date <date> 2010-12-31, 2011-12-31, 2012-12-3...
## $ ROA <dbl> 0.17146869, 0.15552614, 0.13210314...
## $ ROE <dbl> 0.2312902, 0.2080412, 0.1691103, 0...
## $ size <dbl> 14.47189, 14.52877, 14.66921, 14.8...
## $ age <dbl> 4.642624, 4.652211, 4.661732, 4.67...
## $ risk <dbl> 0.01516229, 0.01972431, 0.01816124...
## $ leverage <dbl> 0.11401661, 0.05965369, 0.05713984...
## $ family.idx <dbl> 0.16000000, 0.15384615, 0.15384615...
## $ main.sector <chr> "Consumo Cíclico", "Consumo Cíclic...
```

As one last step, we remove outliers from the data using a simple *winsorization* procedure with a 2.5% cut-off in both sides of the distribution:

```
winsorize <- function(x) {
 lim <- quantile(x, probs = c(0.025,0.975), na.rm = T)
 x[x<lim[1]] <- lim[1]
 x[x>lim[2]] <- lim[2]
return(x)
}

model.table[,c('ROA','ROE','risk')] <- sapply(model.table[,c('ROA','ROE','risk')],
 winsorize)
```

A summary of the resulting dataset is given next:

```
summary(model.table)
```

```
## name.company ref.date ROA
## Length:610 Min. :2010-12-31 Min. :-0.08604
## Class :character 1st Qu.:2011-12-31 1st Qu.: 0.03436
## Mode :character Median :2013-12-31 Median : 0.07900
## Mean :2013-11-27 Mean : 0.08167
## 3rd Qu.:2015-12-31 3rd Qu.: 0.12998
## Max. :2016-12-31 Max. : 0.24659
## NA's :35
## ROE size age
## Min. :-0.8558183 Min. : 9.639 Min. :0.3384
## 1st Qu.: 0.0009763 1st Qu.:13.441 1st Qu.:3.4110
## Median : 0.0815090 Median :14.462 Median :3.9027
## Mean : 0.0578097 Mean :14.541 Mean :3.7861
## 3rd Qu.: 0.1635195 3rd Qu.:15.426 3rd Qu.:4.2447
## Max. : 0.5658081 Max. :21.133 Max. :7.6095
##
## risk leverage family.idx
## Min. :0.001007 Min. :0.00000 Min. :0.00000
## 1st Qu.:0.019455 1st Qu.:0.08347 1st Qu.:0.09091
## Median :0.023876 Median :0.24107 Median :0.13043
## Mean :0.031707 Mean :0.24292 Mean :0.16311
## 3rd Qu.:0.032687 3rd Qu.:0.36836 3rd Qu.:0.20000
## Max. :0.155380 Max. :0.78926 Max. :0.71429
## NA's :8
## main.sector
## Length:610
## Class :character
## Mode :character
##
##
##
##
```

### 4.4 Model estimation and results

With the required dataset cleaned and structured, we proceed with the estimation of the models. First, we set the model specification as a formula object:

```
control.vars <- '+ size + age + risk + leverage'
my.index <- c('name.company', 'ref.date')

my.form.ROA.plm <- formula(paste0('ROA~family.idx + I(family.idx^2)',
 control.vars,
' + factor(year(ref.date))'))

my.form.ROE.plm <- formula(paste0('ROE~family.idx + I(family.idx^2)',
 control.vars,
' + factor(year(ref.date))'))

my.form.ROE.lm <- formula(paste0('ROE~family.idx + I(family.idx^2)',
 control.vars,
' + factor(year(ref.date))',
' + factor(main.sector)') )

my.form.ROA.lm <- formula(paste0('ROA~family.idx + I(family.idx^2)',
 control.vars,
' + factor(year(ref.date))',
' + factor(main.sector)') )
```

We have four models: two panel data and two ordinary least squares. We can now estimate them using `plm` and `lm`:

```
library(plm)

model.table <- as.data.frame(model.table)

my.plm.ROA <- plm(data = as.data.frame(model.table),
formula = my.form.ROA.plm,
index = my.index,
model = 'within')
```

```
my.lm.ROA <- lm(data = model.table,
formula = my.form.ROA.lm)

my.plm.ROE <- plm(data = model.table,
formula = my.form.ROE.plm,
index = my.index,
model = ’within’)

my.lm.ROE <- lm(data = model.table,
formula = my.form.ROE.lm)
```

Finally, we report our results using `stargazer`:

```
library(stargazer)

stargazer(my.plm.ROA, my.lm.ROA, my.plm.ROE, my.lm.ROE,
header = FALSE,
type = ’latex’,
font.size = ’small’,
title = ’Estimation Results of OLS and FE Models’,
model.names = F,
column.labels = c(’FE’, ’OLS’, ’FE’, ’OLS’),
style = ’aer’,
omit = ’factor*’,
omit.stat = c(’f’, ’ser’),
covariate.labels = c(’Family’, ’$Family^2$’, ’Size’,
’Age’, ’Risk’, ’Leverage’),
add.lines = list(c(’Year Dummies’, ’Yes’, ’Yes’,
’Yes’, ’Yes’),
c(’Sector Dummies’, ’No’, ’Yes’,
’No’, ’Yes’)))
```

According to our results, ROA is not significantly related to the presence of a family member on the board of directors, in both fixed effect (FE) and ordinary least squares (OLS) estimations. This evidence is in contrast to those presented by Anderson & Reeb (2003) and Andres (2008), in which family ownership first increases firm performance and then, at higher values, decreases firm performance (positive coefficient in the level and negative coefficient in the square). In the FE estimation, Size has a positive and statistically significant effect on ROA, while Risk and Leverage have a negative and statistically significant effect on ROA and ROE. These results suggest that big, low-risk and low-leveraged firms have higher performance.

**Table 5**
**Estimation Results of OLS and FE Models**

| | ROA | | ROE | |
|---|---|---|---|---|
| | FE | OLS | FE | OLS |
| | (1) | (2) | (3) | (4) |
| Family | 0.053 | 0.039 | −0.005 | 0.258 |
| | (0.105) | (0.076) | (0.414) | (0.224) |
| Family$^2$ | −0.033 | −0.161 | −0.584 | $-0.947^{**}$ |
| | (0.200) | (0.135) | (0.746) | (0.368) |
| Size | $0.052^{***}$ | −0.002 | 0.014 | $0.012^{**}$ |
| | (0.009) | (0.002) | (0.020) | (0.005) |
| Age | 0.040 | 0.000 | 0.091 | −0.003 |
| | (0.026) | (0.003) | (0.106) | (0.010) |
| Risk | $-0.328^{***}$ | $-0.772^{***}$ | $-0.718^{*}$ | $-1.732^{***}$ |
| | (0.110) | (0.121) | (0.420) | (0.355) |
| Leverage | $-0.074^{***}$ | −0.004 | −0.088 | $-0.202^{***}$ |
| | (0.028) | (0.017) | (0.113) | (0.054) |
| Constant | | $0.176^{***}$ | | 0.098 |
| | | (0.030) | | (0.094) |
| Year Dummies | Yes | Yes | Yes | Yes |
| Sector Dummies | No | Yes | No | Yes |
| Observations | 568 | 568 | 602 | 602 |
| R$^2$ | 0.262 | 0.218 | 0.111 | 0.174 |
| Adjusted R$^2$ | 0.075 | 0.191 | −0.111 | 0.147 |

Notes: $^{***}$Significant at the 1 percent level. $^{**}$Significant at the 5 percent level. $^{*}$Significant at the 10 percent level.

## 5. Conclusions

In this paper, we introduce and demonstrate the use of `GetDFPData`, open software for accessing data from B3. The contribution of the software is clear: it facilitates the acquisition and organization of a large and interesting dataset of financial reports and corporate events from the financial exchange. We hope the software becomes popular for investors and academics, facilitating the contribution and reproducibility of studies.

Our literature review section shows that most research conducted in Brazil uses proprietary software for data acquisition. A closer look at a history of published articles indicates that large-scale studies are the rule and not the exception. The proposed software will facilitate current and future researchers' access to a large amount of information available from B3's systems. A reproducible example of usage of `GetDFPData` is provided. Particularly, we investigate the performance of family firms on the Brazilian exchange. This example shows how easy it is to access, manipulate and model the resulting datasets using `GetDFPData` and R's capabilities.

The package is actively maintained and developed over time. Since its release in 2016, many issues have been fixed and features added. The project is not dependent on external funding. The author is committed to building open-source software for importing financial data and uses the package extensively in teaching and research material. Therefore, users should be aware that there is a strong commitment to keeping the project alive and maintained. Nonetheless, it worth remembering that all code is distributed with a generous license. If by chance, the author cannot maintain it, the community can take up the work.

## References

Anderson, R. C. & Reeb, D. M. (2003). Founding-family ownership and firm performance: evidence from the S&P 500, *The Journal of Finance* **58**(3): 1301–1328.

Anderson, R. C. & Reeb, D. M. (2004). Board composition: Balancing family influence in S&P 500 firms, *Administrative Science Quarterly* **49**(2): 209–237.

Andres, C. (2008). Large shareholders and firm performance – an empirical examination of founding-family ownership, *Journal of Corporate Finance* **14**(4): 431–445.

Chang, W., Cheng, J., Allaire, J., Xie, Y. & McPherson, J. (2017). *Shiny: Web Application Framework for R*. R package version 1.0.5.
**URL:** *https://CRAN.R-project.org/package=shiny*

Gandrud, C. (2013). *Reproducible research with R and RStudio*, CRC Press.

Perlin, M. & Portela Santos, A. (2015). Os pesquisadores, as publicações e os períodicos da área de finanças no Brasil: Uma análise com base em currículos da plataforma Lattes, *Revista Brasileira de Finanças* **13**(2).

Perlin, M. & Ramos, H. (2016). Gethfdata: A R package for downloading and aggregating high frequency trading data from bovespa.

R Core Team (2017). *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria.
**URL:** *https://www.R-project.org/*

Villalonga, B. & Amit, R. (2006). How do family ownership, control and management affect firm value?, *Journal of Financial Economics* **80**(2): 385–417.

Wickham, H. (2014). Tidy data, *Journal of Statistical Software* **59**(10): 1–23.

Wickham, H. & Grolemund, G. (2016). R for data science.