

Pacotes para a Importação de Dados da Internet

Análise de Dados Financeiros e Econômicos com o R

Marcelo S. Perlin

EA-UFRGS

03/09/2026

Sumário

- R e a Internet
- Pacote `yfR` (Perlin 2023b)
- Pacote `GetTDDData` (Perlin 2023a)
- Pacote `rb3` (Freitas and Perlin 2023)
- Pacote `GetBCBData` (Perlin 2022)
- Pacote `GetDFPData2` (Perlin and Kirch 2023)
- Pacote `GetFREData` (Perlin and Kirch 2022)
- Tarefa de final de aula

Re a Internet

Por que importar dados da internet?

- acessibilidade, quantidade e qualidade!
- reprodutibilidade de pesquisa
- flexibilidade da janela de análise
- fácil compartilhamento

Quais dados podemos importar?

- Mercado Financeiro
 - Preços de fechamento e ajustados de preços de ações e contratos derivativos
 - Preços e yields de títulos públicos
 - Curvas de juros
 - Dados corporativos de empresas negociadas em bolsa:
 - Sistema DFP (Balanços financeiros Anuais: BP, DRE, FC)
 - Sistema FRE (diversos dados corporativos)
- Economia
 - diversos dados do IBGE, SUS e BCB

Pacote yfR (Perlin 2023b)

Introdução

Pacote `yfR` faz a comunicação do R com os dados financeiros disponíveis no *Yahoo Finance*.

Os diferenciais do `yfR` são:

- Limpeza e organização
- Controle de erros de importação
- Comparação de datas a um *benchmark*
- Uso de sistema de cache
- Acesso a *tickers* em índices de mercado
- Processamento paralelo
- Flexibilidade de formato (mudança de frequência dos dados, e tipo de tabela)

Ibovespa desde 1995

```
1 # set tickers
2 my_tickers ← "^BVSP"
3 first_date ← '2000-01-01'
4 last_date ← Sys.Date()
5 cache_folder ← "~/.cache/yfR/"
6
7 df_yf ← yfR::yf_get(tickers = my_tickers,
8                     first_date = first_date,
9                     last_date = last_date,
10                     cache_folder = cache_folder)
11
12 dplyr::glimpse(df_yf)
```

Rows: 6,608

Columns: 11

\$ ticker <chr> "^BVSP", "^BVSP", "^BVSP",
"^BVSP", "^BVSP", "^...

```
$ ref_date          <date> 2000-01-03, 2000-01-04,  
2000-01-05, 2000-01-06...  
$ price_open        <dbl> 17098, 16908, 15871,  
16237, 16125, 16325, 17045...  
$ price_high        <dbl> 17408, 16908, 16302,  
16499, 16449, 17057, 17197...
```

Gráfico Ibovespa



Exemplo para várias empresas

```
1 # set tickers
2 my_tickers ← c('PETR4.SA', 'CIEL3.SA',
3               'GGBR4.SA', 'GOAU4.SA')
4 first_date ← "2015-01-01"
5 last_date ← Sys.Date()
6
7 df_yf ← yfR::yf_get(tickers = my_tickers,
8                    first_date = first_date,
9                    last_date = last_date,
10                   cache_folder = cache_folder)
11
12 dplyr::glimpse(df_yf)
```

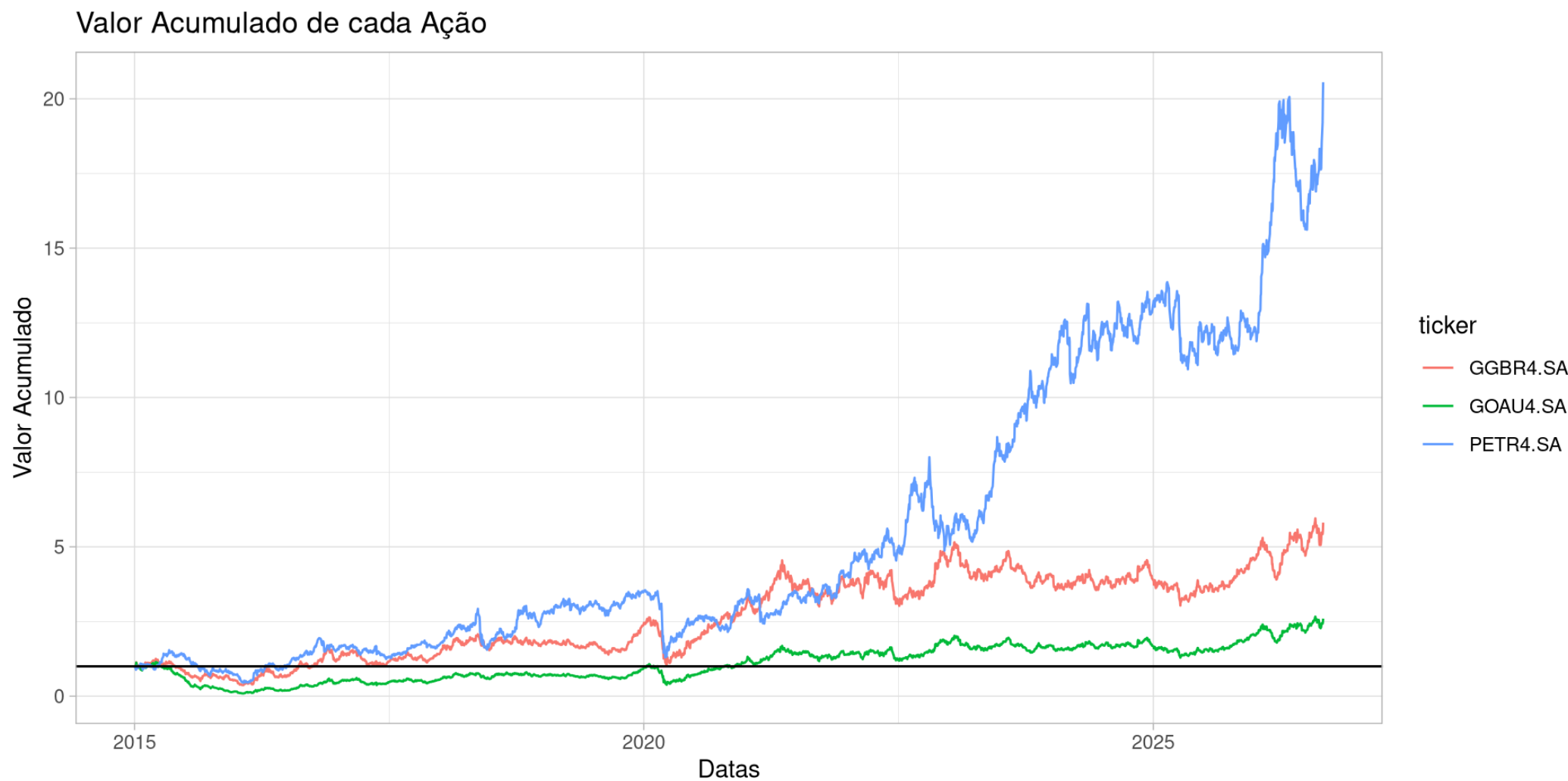
Rows: 8,715

Columns: 11

\$ ticker <chr> "GGBR4.SA", "GGBR4.SA",
"GGBR4.SA", "GGBR4.SA", ...

```
$ ref_date          <date> 2015-01-02, 2015-01-05,  
2015-01-06, 2015-01-07...  
$ price_open        <dbl> 7.571428, 7.174603,  
6.920634, 7.896825, 8.36507...  
$ price_high        <dbl> 7.571428, 7.198412,  
7.698412, 8.277777, 8.46031...
```

Grafico



Baixando Dados da Composição do Ibovespa

```
1 # set tickers
2 df_ibov ← yfR::yf_index_composition("IBOV")
3
4 dplyr::glimpse(df_ibov)
```

Rows: 88

Columns: 6

```
$ ticker      <chr> "ALPA4", "ABEV3", "AMER3", "AZUL4",  
"B3SA3", "BBAS3", "BB...
```

```
$ company      <chr> "Alpargatas", "Ambev", "Americanas",  
"Azul", "B3", "Banco..."
```

```
$ type_stock      <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,
NA, NA, NA, NA, NA, N...
```

```
$ ref_date      <date> 2026-09-03, 2026-09-03, 2026-09-03,
2026-09-03, 2026-09-...
```

[illegible]

Pacote GetTDDData (Perlin 2023a)

Introdução

Pacote `GetTDDData` importa os dados de preços e *yields* de títulos do Tesouro Direto do site do Tesouro Nacional e os organiza para facilitar a análise.

- resultado é sempre um `dataframe` com dados empilhados.
- disponibiliza dados para diferentes tipos de títulos de dívida, para diferentes maturidades

```
1 asset_codes ← 'LTN'      # Identifier of assets
2 maturity ← '010121'      # Maturity date as string (ddmmy
3 first_year ← 2015
4 last_year ← 2024
5 dl_folder ← "~/cache/GetTDDData/"
6
7 # download
8 df_TD ← GetTDDData::td_get(asset_codes,
```

```

9
10
11
12
13 dn1vr::dlmnse(df TD)
first_year = first_year,
last_year = last_year,
dl_folder = dl_folder)

```

Rows: 28,024

Columns: 5

\$ ref_date <date> 2005-01-03, 2005-01-03, 2005-01-04,
2005-01-05, 2005-01-06...

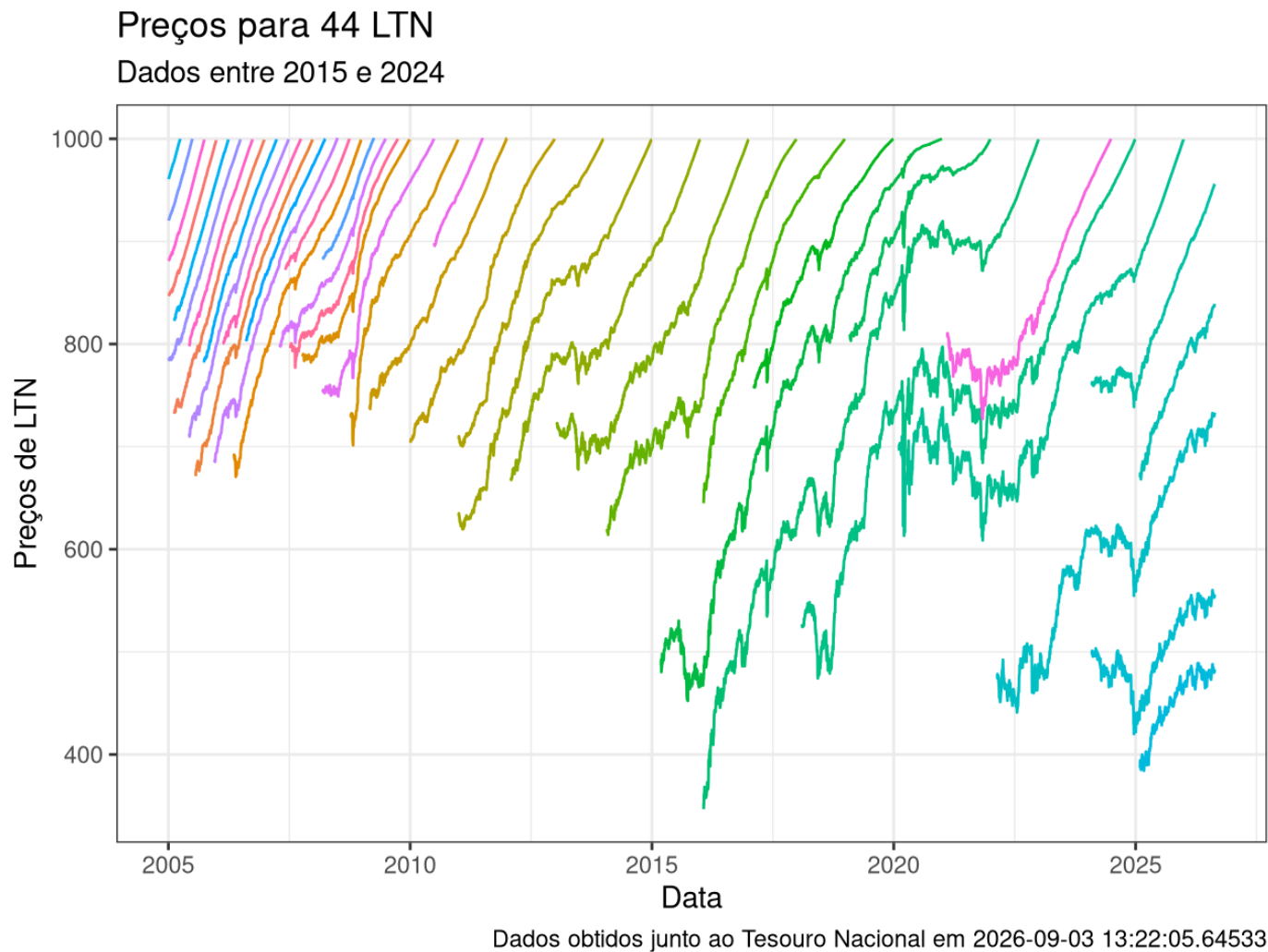
\$ yield_bid <dbl> 0.1774, 0.1836, 0.1839, 0.1841,
0.1842, 0.1847, 0.1844, 0.1...

\$ price_bid <dbl> 1000.00, 960.66, 961.24, 961.85,
962.48, 963.03, 963.73, 96...

\$ asset_code <chr> "LTN 040105", "LTN 010405", "LTN
010405", "LTN 010405", "LT...

\$ ref_date <date> 2005-01-03, 2005-01-04, 2005-01-05, 2005-01-06

Preços de LTNs

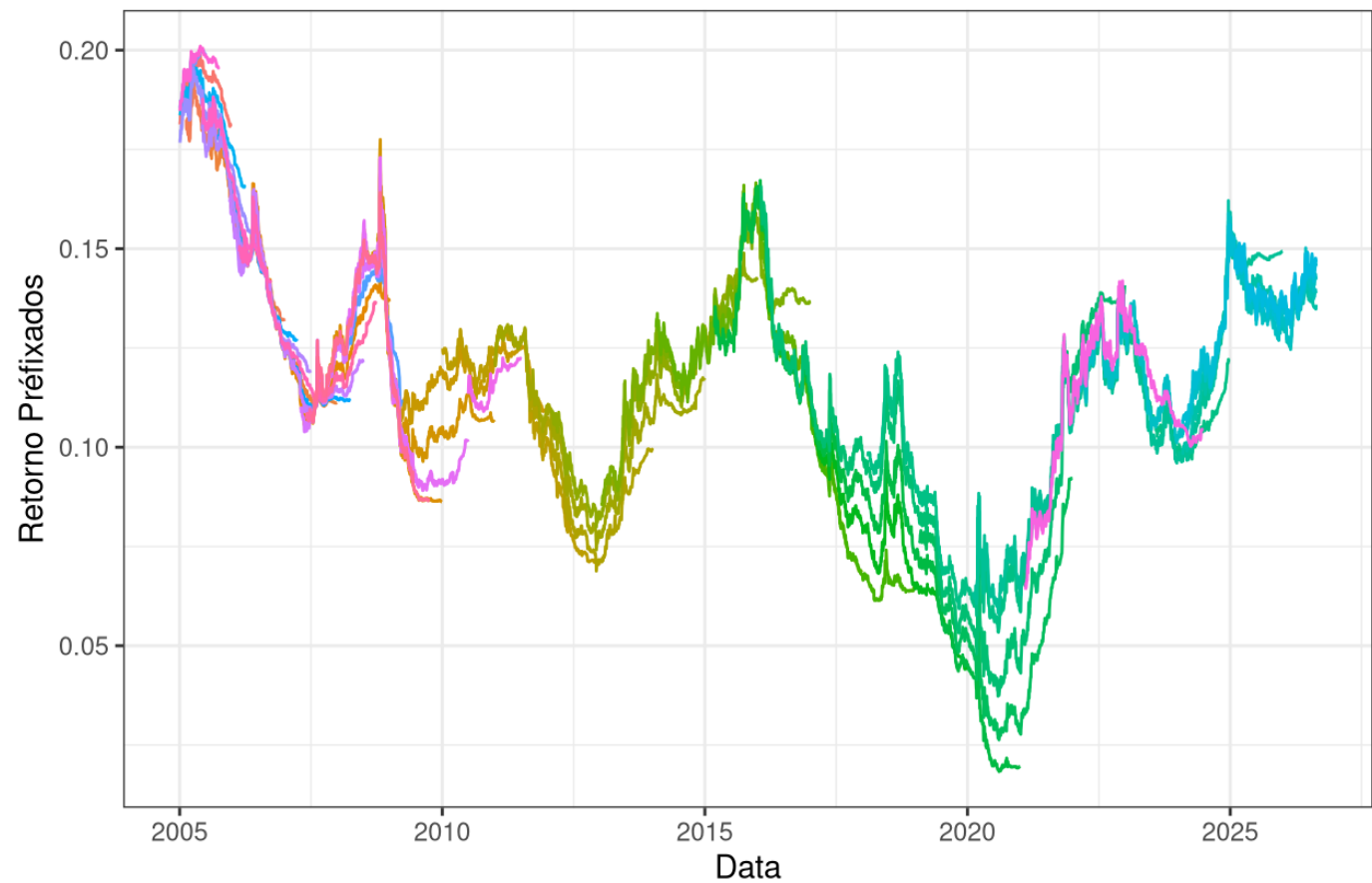


Título TD

LTN 010106	LTN 010121	LTN 010705
LTN 010107	LTN 010122	LTN 010706
LTN 010108	LTN 010123	LTN 010707
LTN 010109	LTN 010125	LTN 010708
LTN 010110	LTN 010126	LTN 010709
LTN 010111	LTN 010127	LTN 010710
LTN 010112	LTN 010128	LTN 010711
LTN 010113	LTN 010129	LTN 010724
LTN 010114	LTN 010131	LTN 011005
LTN 010115	LTN 010132	LTN 011006
LTN 010116	LTN 010405	LTN 011007
LTN 010117	LTN 010406	LTN 011008
LTN 010118	LTN 010407	LTN 011009
LTN 010119	LTN 010408	LTN 040105
LTN 010120	LTN 010409	

Yields LTN

Retornos Pré-fixados para 44 LTN
Dados entre 2015 e 2024



Dados obtidos junto ao Tesouro Nacional em 2026-09-03 13:22:06.937301

Título TD		
LTN 010106	LTN 010121	LTN 010705
LTN 010107	LTN 010122	LTN 010706
LTN 010108	LTN 010123	LTN 010707
LTN 010109	LTN 010125	LTN 010708
LTN 010110	LTN 010126	LTN 010709
LTN 010111	LTN 010127	LTN 010710
LTN 010112	LTN 010128	LTN 010711
LTN 010113	LTN 010129	LTN 010724
LTN 010114	LTN 010131	LTN 011005
LTN 010115	LTN 010132	LTN 011006
LTN 010116	LTN 010405	LTN 011007
LTN 010117	LTN 010406	LTN 011008
LTN 010118	LTN 010407	LTN 011009
LTN 010119	LTN 010408	LTN 040105
LTN 010120	LTN 010409	

Curva de Juros

```
1 library(GetTDDData)
2
3 # get yield curve
4 df_yield ← get.yield.curve()
5
6 # check result
7 dplyr::glimpse(df_yield)
```

Rows: 103

Columns: 5

\$ n_biz_days <dbl> 252, 252, 252, 378, 378, 378, 504,
504, 504, 630, 630, 63...

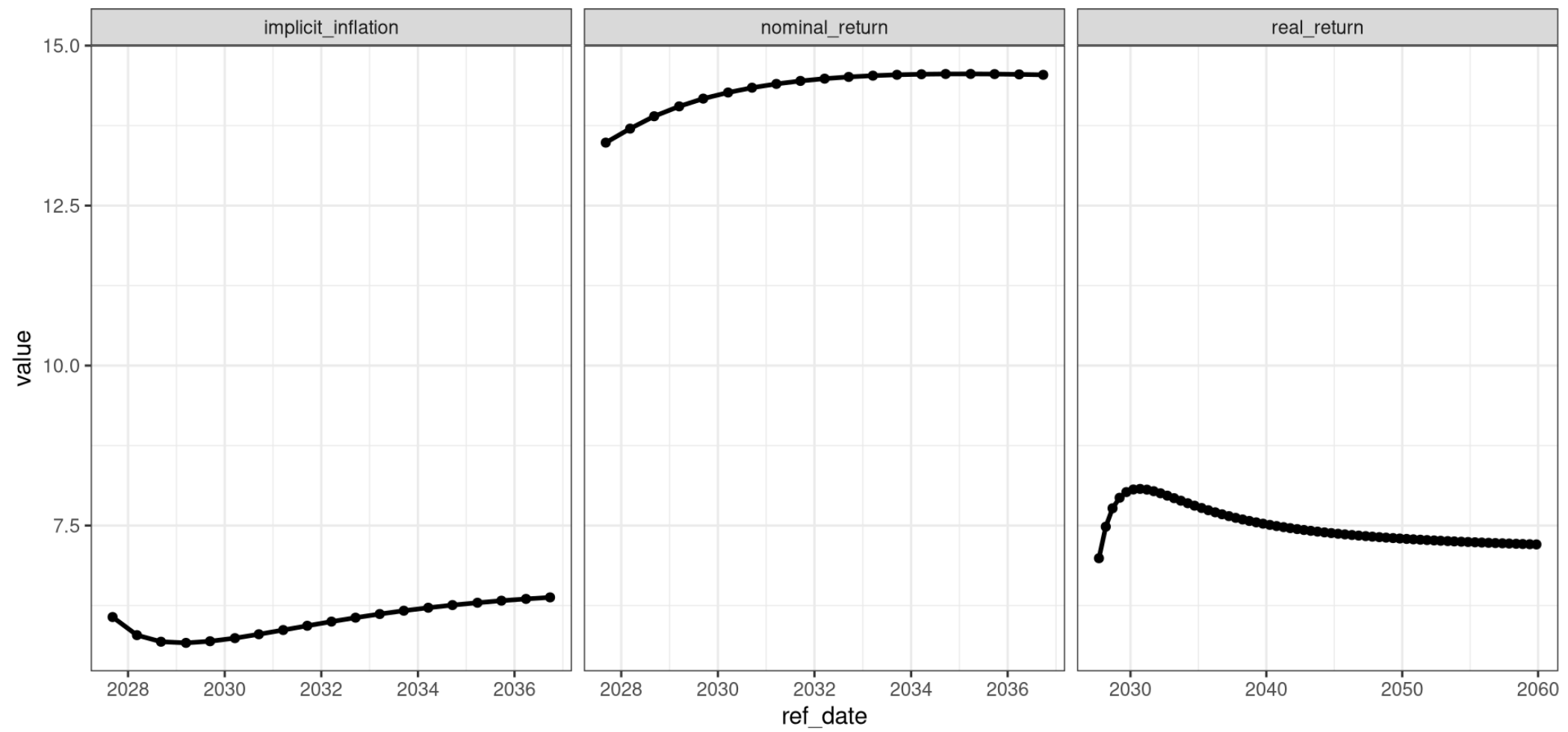
\$ type <chr> "real_return", "nominal_return",
"implicit_inflation", "r...

\$ value <dbl> 6.9894, 13.4839, 6.0702, 7.4821,
13.7028, 5.7876, 7.7699,...

\$ ref_date <date> 2027-09-06, 2027-09-06, 2027-09-06,

Gráfico

The current Brazilian Yield Curve -



Pacote `rb3` (Freitas and Perlin 2023)

Introdução

Pacote desenvolvido por Wilson Freitas, especializado na captura de dados da B3, a bolsa brasileira

Possui dados para:

- equities/stocks
- futures
- FII (Reits)
- options
- BDRs
- historical yield curves (calculated from futures contracts)
- B3 indexes composition

Exemplo para curva de juros

```
1 library(rb3)
2 library(glue)
3 library(stringr)
4
5 # TODO: Get more examples
6 date_vec ← bizdays::bizseq(Sys.Date() - 365 * 5, Sys.
7 fetch_marketdata("b3-reference-rates",
8                   refdate = date_vec,
9                   curve_name = c("DIC", "DOC", "PRE"),
10                  throttle = TRUE)
11
12 df_yc ← rb3::yc_br1_get() ▷
13 dplyr::collect() ▷
```

Gráfico da Curva

Pacote GetBCBData (Perlin 2022)

Introdução

Pacote que permite acesso ao [Sistema de Séries Temporais \(SGS\)](#) do Banco Central Brasileiro (BCB)

- Uso de um sistema de *cache* local para acelerar solicitações repetidas de dados;
- Escolha do formato da saída: longo (orientado a linhas, dados empilhados) ou amplo (orientado a colunas)
- Tratamento interno de erros. Mesmo que a série solicitada não exista, a função ainda retornará todos os resultados.
- identificadores de séries disponíveis no [sistema de séries temporais do BCB](#)
- Com o `id` e período de tempo, usamos função `GetBCBData::gcbcd_get_series()` para baixar os dados

Exemplo para série de inadimplência de crédito pessoal no brasil

```
1 library(GetBCBData)
2 library(dplyr)
3
4 # set ids and dates
5 id_series <- c(perc_default = 21082)
6 first_date = '2010-01-01'
7
8 # get series from bcb
9 df_cred <- gcbcd_get_series(id = id_series,
10                             first.date = first_date,
11                             last.date = Sys.Date(),
12                             use.memoise = FALSE)
13
```

Rows: 185

Columns: 4

```
$ ref.date      <date> 2011-03-01, 2011-04-01, 2011-05-01,  
2011-06-01, 2011-07-01...  
$ value        <dbl> 3.17, 3.24, 3.37, 3.32, 3.42, 3.45,  
3.46, 3.58, 3.58, 3.54...  
$ id.num       <dbl> 21082, 21082, 21082, 21082, 21082,  
21082, 21082, 21082, 21...  
$ series.name  <chr> "perc_default", "perc_default",  
"perc_default", "perc_defa..."
```

Gráfico



Source: SGS - BCB (by GetBCBData)

A Taxa Selic

```
1 my_id ← c('Taxa SELIC' = 432)
2 first_date ← '2010-01-01'
3 last_date ← Sys.Date()
4
5 df_bcb ← GetBCBData::gbcdb_get_series(
6   my_id,
7   first.date = first_date,
8   last.date = last_date
9 )
10
11 dplyr::glimpse(df_bcb)
```

Rows: 6,090

Columns: 4

\$ ref.date <date> 2010-01-01, 2010-01-02, 2010-01-03,
2010-01-04, 2010-01-0...

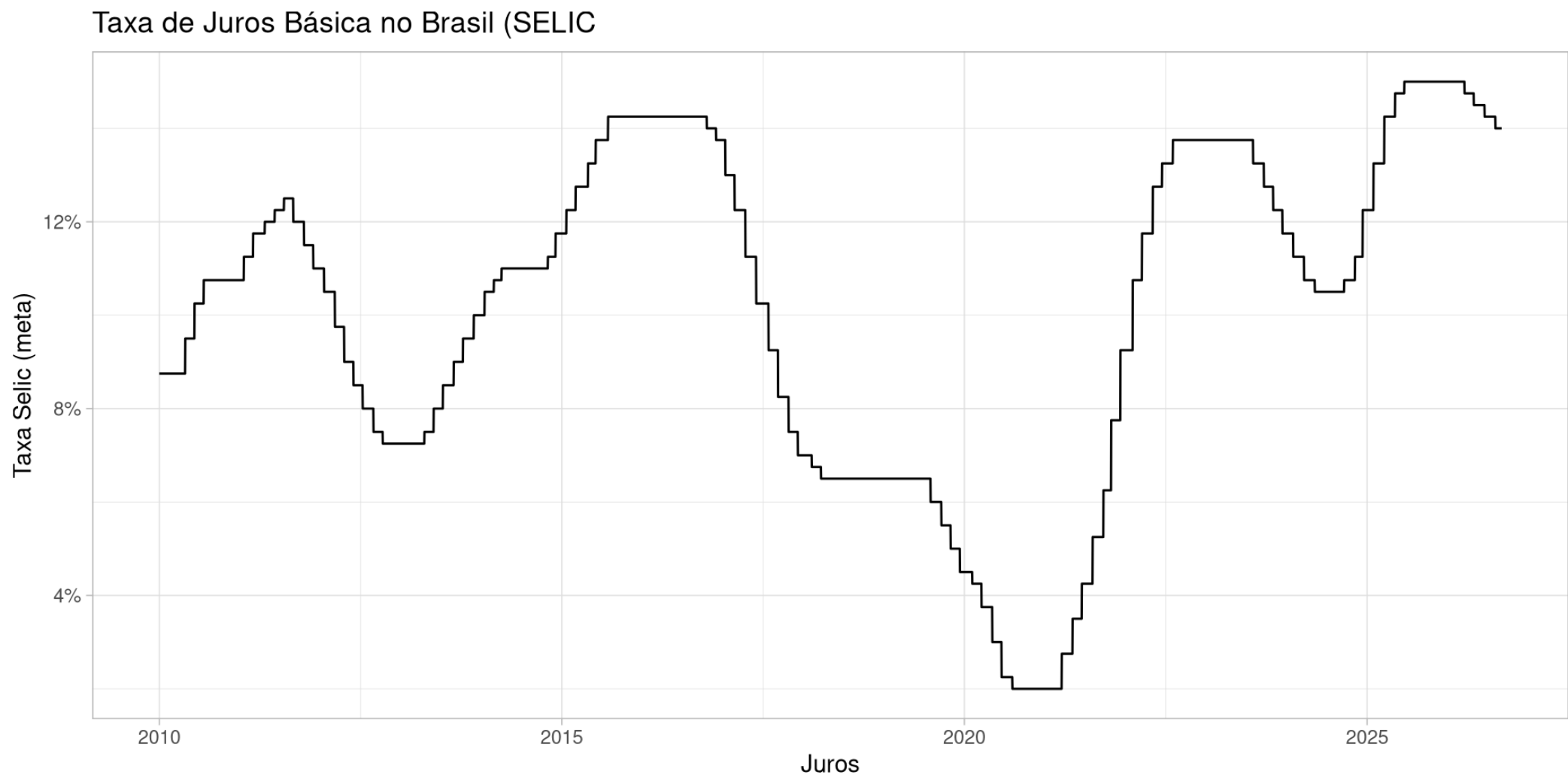
\$ value <dbl> 8.75, 8.75, 8.75, 8.75, 8.75, 8.75,

8.75, 8.75, 8.75, 8.75...

\$ id.num <dbl> 432, 432, 432, 432, 432, 432, 432,
432, 432, 432, 432, 432...

\$ series.name <chr> "Taxa SELIC", "Taxa SELIC", "Taxa
SELIC", "Taxa SELIC", "T...

Gráfico da SELIC



Pacote GetDFPData2 (Perlin and Kirch 2023)

Introdução

Pacote `GetDFPData2` (Perlin and Kirch 2023) é uma evolução do pacote `GetDFPData` (Perlin 2021) e fornece uma interface aberta para todas as demonstrações financeiras distribuídas pela B3 e pela CVM nos sistemas DFP (dados anuais)

- Permite acesso a dados anuais do:
 - DRE (demonstrativo de resultado do exercício)
 - BP (balanço patrimonial)
 - DFC (demonstrativo de fluxo de caixa)

Exemplo de uso

```
1 # get info for companies in B3
2 df_info ← GetDFPData2::get_info_companies()
3
4 # check it
5 dplyr::glimpse(df_info)
```

Rows: 2,677

Columns: 47

```
$ CNPJ          <chr> "08.773.135/0001-00",
"11.396.633/0001-87", "01.547...
$ DENOM_SOCIAL  <chr> "2W ECOBANK S.A. - EM
RECUPERAÇÃO JUDICIAL", "3A CO...
$ DENOM_COMERC <chr> "2W ECOBANK S.A.", "TRIPLA A
COMPANHIA SECURITIZAD...
$ DT_REG       <date> 2020-10-29, 2010-03-08, 1997-
07-11, 1997-05-30, 19...
$ DT_CONCT     <date> 2007-07-07, 2000-11-07, 1997-
```

Dados Financeiros da GRENDENE SA

```
1 id_companies ← 19615
2 first_year ← 2010
3 last_year ← 2022
4
5 # download data
6 tbl_dfp ← GetDFPData2::get_dfp_data(
7   companies_cvm_codes = id_companies,
8   type_docs = '*', # get all docs
9   type_format = 'con', # consolidated
10  first_year = first_year,
11  last_year = last_year
12 )
13
```

List of 6

```
$ DF Consolidado - Balanço Patrimonial Ativo
: tibble [794 × 16] (S3: tbl_df/tbl/data.frame)
```

```
..$ CNPJ_CIA      : chr [1:794] "89.850.341/0001-60"  
"89.850.341/0001-60" "89.850.341/0001-60"  
"89.850.341/0001-60" ...  
..$ CD_CVM        : num [1:794] 19615 19615 19615 19615  
19615 ...  
..$ DT_REFER      : Date[1:794], format: "2010-12-31"  
"2010-12-31"
```

O lucro da GRENDENE SA (código)

O lucro da Grendene (tabela)

Lucro da GRENDENE SA			
DENOM_CIA	DT_REFER	CD_CONTA	VL_CONTA
GRENDENE S.A.	2010-12-31	3.11	R\$312,467,000.00
GRENDENE S.A.	2011-12-31	3.11	R\$305,451,000.00
GRENDENE S.A.	2012-12-31	3.11	R\$429,891,000.00
GRENDENE S.A.	2013-12-31	3.11	R\$434,005,000.00
GRENDENE S.A.	2014-12-31	3.11	R\$485,259,000.00
GRENDENE S.A.	2015-12-31	3.11	R\$539,311,000.00
GRENDENE S.A.	2016-12-31	3.11	R\$633,955,000.00
GRENDENE S.A.	2017-12-31	3.11	R\$660,903,000.00
GRENDENE S.A.	2018-12-31	3.11	R\$585,530,000.00
GRENDENE S.A.	2019-12-31	3.11	R\$494,954,000.00
GRENDENE S.A.	2020-12-31	3.11	R\$405,206,000.00
GRENDENE S.A.	2021-12-31	3.11	R\$601,005,000.00
GRENDENE S.A.	2022-12-31	3.11	R\$568,027,000.00

O EBITDA da Grendene SA (código)

O EBITDA da Grendene SA (tabela)

EBITDA da GRENDENE SA			
DENOM_CIA	DT_REFER	CD_CONTA	VL_CONTA
GRENDENE S.A.	2010-12-31	3.05	R\$208,413,000.00
GRENDENE S.A.	2011-12-31	3.05	R\$187,293,000.00
GRENDENE S.A.	2012-12-31	3.05	R\$362,813,000.00
GRENDENE S.A.	2013-12-31	3.05	R\$399,233,000.00
GRENDENE S.A.	2014-12-31	3.05	R\$389,413,000.00
GRENDENE S.A.	2015-12-31	3.05	R\$400,732,000.00
GRENDENE S.A.	2016-12-31	3.05	R\$399,594,000.00
GRENDENE S.A.	2017-12-31	3.05	R\$465,590,000.00
GRENDENE S.A.	2018-12-31	3.05	R\$456,963,000.00
GRENDENE S.A.	2019-12-31	3.05	R\$353,528,000.00
GRENDENE S.A.	2020-12-31	3.05	R\$301,199,000.00
GRENDENE S.A.	2021-12-31	3.05	R\$393,522,000.00
GRENDENE S.A.	2022-12-31	3.05	R\$239,120,000.00

Pacote GetFREData (Perlin and Kirch 2022)

Introdução

O pacote `GetFREData` importa dados do sistema FRE (Formulário de Referência) da B3

- inclui dados para eventos e informações corporativas
 - composição do conselho e diretoria
 - remuneração dos conselhos
 - composição da dívida
 - vários outros.

Exemplo de uso

```
1 id_companies ← 23264
2 first_year ← 2017
3 last_year ← 2018
4
5 # download data
6 l_fre ← GetFREData::get_fre_data2(companies_cvm_codes
7                                   first_year = first_year,
8                                   last_year = last_year)
9
10 dplyr::glimpse(l_fre)
```

List of 43

```
$ cia_aberta : tibble [10 ×  
10] (S3: tbl_df/tbl/data.frame)  
.. $ cnpj_cia : chr [1:10] "07.526.557/0001-00"  
"07.526.557/0001-00" "07.526.557/0001-00"  
"07.526.557/0001-00" ...
```

```
..$ dt_refer      : Date[1:10], format: "2017-01-01"  
"2017-01-01" ...  
..$ versao        : num [1:10] 1 2 3 4 5 6 7 8 9 10  
..$ denom_cia     : chr [1:10] "AMBEV S.A." "AMBEV S.A."
```

Tarefa de final de aula

Tarefa 02

1. Utilizando pacote `yfR`, baixe os dados da ação MDIA3 (ticker MDIA3.SA) no *Yahoo Finance* para os últimos 30 dias.

- Qual o preço ajustado mais baixo no período analisado?
- Qual a data com o preço mais baixo?
- Faça um gráfico simples com função `plot()`

2. Utilize pacote `GetBCBData` para baixar dados do IPCA mensal (em % anual) nos últimos 5 anos.

- É possível observar algum período com deflação, inflação negativa, mensal?

3. Visite o [site](#) de sistema de séries temporais do Banco Central do Brasil e escolha uma série para análise de acordo com seus próprios interesses de pesquisa. Importe a mesma usando pacote `GetBCBData`.

4. Utilizando função `GetDFPData2::get_info_companies`, baixe informações sobre as ações negociadas na B3.

- Quantas empresas no total estão atualmente disponíveis no banco de dados? - Qual a proporção de empresas ativas (veja coluna `SIT_REG`)?

- Qual a empresa mais antiga (veja coluna `DT_REG`)?

Referências

- Freitas, Wilson, and Marcelo Perlin. 2023. *Download and Parse Public Data Released by B3 Exchange*. <https://github.com/ropensci/rb3>.
- Perlin, Marcelo. 2021. *GetDFPData: Reading Annual Financial Reports from Bovespa's DFP, FRE and FCA System*. <https://CRAN.R-project.org/package=GetDFPData>.
- Perlin, Marcelo. 2022. *GetBCBData: Imports Datasets from BCB (Central Bank of Brazil) Using Its Official API*. <https://CRAN.R-project.org/package=GetBCBData>.
- Perlin, Marcelo. 2023a. *GetTDDData: Get Data for Brazilian Bonds (Tesouro Direto)*. <https://CRAN.R-project.org/package=GetTDDData>.
- Perlin, Marcelo. 2023b. *yfR: Downloads and Organizes Financial Data from Yahoo Finance*. <https://github.com/ropensci/yfR>.
- Perlin, Marcelo, and Guilherme Kirch. 2022. *GetFREDData: Reading FRE Corporate Data of Public Traded Companies from B3*. <https://CRAN.R-project.org/package=GetFREDData>.
- Perlin, Marcelo, and Guilherme Kirch. 2023. *GetDFPData2: Reading Annual and Quarterly Financial Reports from B3*. <https://CRAN.R-project.org/package=GetDFPData2>.