

# Package ‘GetDFPData’

May 18, 2020

**Title** Reading Annual Financial Reports from Bovespa's DFP, FRE and FCA System

**Version** 1.5.3

**Date** 2020-05-16

**Description** Reads annual financial reports including assets, liabilities, dividends history, stockholder composition and much more from Bovespa's DFP, FRE and FCA systems <[http://www.bmfbovespa.com.br/en\\_us/products/listed-equities-and-derivatives/equities/listed-companies.htm](http://www.bmfbovespa.com.br/en_us/products/listed-equities-and-derivatives/equities/listed-companies.htm)>.

These are web based interfaces for all financial reports of companies traded at Bovespa. The package is specially designed for large scale data importation, keeping a tabular (long) structure for easier processing.

**Depends** R (>= 3.3.0)

**Imports** stringr, XML, dplyr, readr, reshape2, tibble, xlsx, stats, curl, lubridate

**ByteCompile** true

**License** GPL-2

**BugReports** <https://github.com/msperlin/GetDFPData/issues>

**URL** <https://github.com/msperlin/GetDFPData/>

**LazyData** true

**RoxygenNote** 7.1.0

**Suggests** knitr, rmarkdown, testthat, ggplot2

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** Marcelo Perlin [aut, cre]

**Maintainer** Marcelo Perlin <[marceloperlin@gmail.com](mailto:marceloperlin@gmail.com)>

**Repository** CRAN

**Date/Publication** 2020-05-18 16:10:02 UTC

**R topics documented:**

fix.fct . . . . .	3
gdfpd.convert.to.wide . . . . .	3
gdfpd.download.file . . . . .	4
gdfpd.export.DFP.data . . . . .	5
gdfpd.fix.DFP.dataframes . . . . .	6
gdfpd.get.bovespa.data . . . . .	7
gdfpd.get.files.from.bovespa . . . . .	7
gdfpd.get.inflation.data . . . . .	8
gdfpd.get.info.companies . . . . .	9
gdfpd.GetDFPData . . . . .	10
gdfpd.read.dfp.zip.file . . . . .	11
gdfpd.read.dfp.zip.file.type.1 . . . . .	12
gdfpd.read.dfp.zip.file.type.2 . . . . .	13
gdfpd.read.fca.zip.file . . . . .	13
gdfpd.read.fre.zip.file . . . . .	14
gdfpd.read.fwf.file . . . . .	15
gdfpd.read.zip.file.type.fca . . . . .	15
gdfpd.read.zip.file.type.fre . . . . .	16
gdfpd.search.company . . . . .	17
get_files . . . . .	17
my.copy.fct . . . . .	18
my.merge.dfs.lists . . . . .	19
xml.fct.auditing . . . . .	19
xml.fct.board.composition . . . . .	20
xml.fct.capital . . . . .	20
xml.fct.capital.reduction . . . . .	21
xml.fct.committee.composition . . . . .	21
xml.fct.compensation . . . . .	22
xml.fct.compensation.summary . . . . .	23
xml.fct.debt . . . . .	23
xml.fct.div.details . . . . .	24
xml.fct.family.related.parts . . . . .	24
xml.fct.family.relations . . . . .	25
xml.fct.intangible.details . . . . .	25
xml.fct.repurchases . . . . .	26
xml.fct.responsible . . . . .	26
xml.fct.splits.inplits . . . . .	27
xml.fct.stock.values . . . . .	27
xml.fct.stockholder . . . . .	28
xml.fct.stocks.details . . . . .	28
xml.fct.transactions.related . . . . .	29

---

fix.fct	<i>Fix NULL values in dataframe</i>
---------	-------------------------------------

---

**Description**

Fix NULL values in dataframe

**Usage**

```
fix.fct(x, type.info = "character", format.date = "%Y-%m-%d")
```

**Arguments**

x	Am object, possibly NULL
type.info	Type of object
format.date	Format of data, as string

**Value**

A single object

**Examples**

```
x <- NULL  
x2 <- fix.fct(x)
```

---

gdfpd.convert.to.wide	<i>Converts a dataframe from gdfpd_GetDFPData to the wide format</i>
-----------------------	--

---

**Description**

Converts a dataframe from gdfpd\_GetDFPData to the wide format

**Usage**

```
gdfpd.convert.to.wide(data.in, data.in.cols = "original")
```

**Arguments**

data.in	Data frame with financial information
data.in.cols	Which data to go in rows values ('original' or 'inflation adjusted')

**Value**

A dataframe in the wide format

## Examples

```
# get example data from RData file
my.f <- system.file('extdata/Example_DFP_Report_Petrobras.RData', package = 'GetDFPData')
load(my.f)

df.assets <- df.reports$fr.assets[[1]]
df.assets.wide <- gdfpd.convert.to.wide(df.assets)
```

---

gdfpd.download.file     *Downloads files from the internet*

---

## Description

Downloads files from the internet

## Usage

```
gdfpd.download.file(dl.link, dest.file, max.dl.tries)
```

## Arguments

dl.link	Link to file
dest.file	= Destination, as local file
max.dl.tries	Maximum number of attempts for downloading files

## Value

Nothing

## Examples

```
my.url <- paste0('http://www.rad.cvm.gov.br/enetconsulta/',
                'frmDownloadDocumento.aspx?CodigoInstituicao=2',
                '&NumeroSequencialDocumento=46133')

## Not run: # keep CHECK fast
dl.status <- gdfpd.download.file(my.url, 'tempfile.zip', 10)

## End(Not run)
```

---

gdfpd.export.DFP.data *Export tibble to an excel or csv (zipped) file*

---

## Description

Export information from `gdfpd_GetDFPData()` to an excel file or csv. In the csv case, all tables are exported as csv files and zipped in a single zip file.

## Usage

```
gdfpd.export.DFP.data(  
  df.reports,  
  base.file.name = paste0("GetDFPData_Export_", Sys.Date()),  
  type.export = "xlsx"  
)
```

## Arguments

`df.reports`      Tibble with financial information (output of `gdfpd.GetDFPData`)  
`base.file.name`    The basename of excel file (make sure you dont include the file extension)  
`type.export`      The extension of the desired format: 'xlsx' (default) or 'csv'

## Value

TRUE, if successfull (invisible)

## Examples

```
# get example data from RData file  
my.f <- system.file('extdata/Example_DFP_Report_Petrobras.RData', package = 'GetDFPData')  
load(my.f)  
  
## Not run: # dontrun: keep cran check time short  
gdfpd.export.DFP.data(df.reports, base.file.name = 'MyExcelFile', format.data = 'wide')  
  
## End(Not run)
```



---

`gdfpd.get.bovespa.data`

*Reads information for a company from B3 site*

---

### **Description**

Given a CVM code, this function scrapes information from the company page.

### **Usage**

```
gdfpd.get.bovespa.data(my.id)
```

### **Arguments**

`my.id`            A CVM id

### **Value**

A list with several dataframes

### **Examples**

```
## Not run: # keep cran check fast
l.info.PETR <- gdfpd.get.dovespa.data(my.id = 9512)
str(l.info.PETR)

## End(Not run)
```

---

`gdfpd.get.files.from.bovespa`

*Fetches ALL new files from Bovespa*

---

### **Description**

Fetches ALL new files from Bovespa

### **Usage**

```
gdfpd.get.files.from.bovespa(my.id)
```

### **Arguments**

`my.id`            Company's ID

**Value**

A dataframe with several information about files

**Examples**

```
## Not run:  
df.files <- gdfpd.get.files.from.bovespa(9512)  
  
## End(Not run)
```

---

```
gdfpd.get.inflation.data
```

*Downloads and read inflation data from github*

---

**Description**

Inflation data is available at git repo 'msperlin/GetITRData\_auxiliary'

**Usage**

```
gdfpd.get.inflation.data(inflation.index, do.cache)
```

**Arguments**

inflation.index	Sets the inflation index to use for finding inflation adjusted values of all reports. Possible values: 'dollar' (default) or 'IPCA', the brazilian main inflation index. When using 'IPCA', the base date is set as the last date found in the DFP dataset.
do.cache	Logical for controlling to whether to use a cache system or not. Default = TRUE

**Value**

A dataframe with inflation data

**Examples**

```
## Not run: # keep cran check fast  
df.inflation <- gdfpd.get.inflation.data('IPCA')  
str(df.inflation)  
  
## End(Not run)
```



---

`gdfpd.get.info.companies`

*Reads up to date information about Bovespa companies from a github file*

---

## Description

A csv file with information about available companies, file links and time periods is read from github. This file is manually updated by the author. When run for the first time in a R session, a .RDATA file containing the output of the function is saved for caching.

## Usage

```
gdfpd.get.info.companies(  
  type.data = "companies_files",  
  cache.folder = "DFP Cache Folder"  
)
```

## Arguments

<code>type.data</code>	A string that sets the type of information to be returned ('companies' or 'companies_files'). If 'companies', it will return a dataframe with several information about companies, but without download links.
<code>cache.folder</code>	Folder to cache (save) all processed information. Default = <code>file.path(getwd(), 'DFP Cache Folder')</code>

## Value

A dataframe with several information about Bovespa companies

## Examples

```
## Not run: # keep cran check fast  
df.info <- gdfpd.get.info.companies()  
str(df.info)  
  
## End(Not run)
```

---

gdfpd.GetDFPDData	<i>Downloads and reads financial reports from B3's DFP/FRE/FCA system</i>
-------------------	---

---

### Description

Annual data for financial reports and corporate events are downloaded from B3 for a combination of companies and time period. This function gathers data into a single tibble object and organizes it in a tabular/long format.

### Usage

```
gdfpd.GetDFPDData(
  name.companies,
  first.date = Sys.Date() - 12 * 30,
  last.date = Sys.Date(),
  selected.data = "DFP|FRE|FCA",
  inflation.index = "dollar",
  max.levels = 3,
  folder.out = tempdir(),
  do.cache = TRUE,
  cache.folder = "DFP Cache Folder",
  fetch.new.files = FALSE,
  max.dl.tries = 10
)
```

### Arguments

name.companies	Official names of companies to get financial reports (e.g. 'ELETROPAULO METROPOLITANA EL.S.PAULO S.A'). Names of companies can be found using function <code>gdfpd.search.company('nametolookfor')</code> or <code>gdfpd.get.info.companies('companies')</code>
first.date	First date (YYYY-MM-DD) to get data. Character or Date. E.g. <code>first.date = '2010-01-01'</code> .
last.date	Last date (YYYY-MM-DD) to get data. Character or Date. E.g. <code>last.date = '2017-01-01'</code> .
selected.data	Symbols for the selection of datasets: 'DFP FRE FCA', 'DFP FRE', 'FRE FCA', 'DFP FCA', 'DFP', 'FRE', 'FCA'. Default = 'DFP FRE FCA'
inflation.index	Sets the inflation index to use for finding inflation adjusted values of all reports. Possible values: 'dollar' (default) or 'IPCA', the brazilian main inflation index. When using 'IPCA', the base date is set as the last date found in the DFP dataset.
max.levels	Sets the maximum number of levels of accounting items in financial reports (default = 3)
folder.out	Folder where to download and manipulate the zip files. Default = <code>tempdir()</code>
do.cache	Logical for controlling to whether to use a cache system or not. Default = TRUE

`cache.folder` Folder to cache (save) all processed information. Default = `file.path(getwd(), 'DFP Cache Folder')`  
`fetch.new.files` Logical. Should the function search for new files/data in Bovespa? (default = FALSE)  
`max.dl.tries` Maximum number of attempts for downloading files

### Details

The easiest way to get started with `gdfpd.GetDFPData` is looking for the official name of traded companies using function `gdfpd.search.company('nametolookfor')`. Alternatively, you can use function `gdfpd.get.info.companies('companies')` to import a dataframe with information for all available companies and time periods.

### Value

A tibble object with all gathered financial statements, with each company as a row

### Examples

```
## Not run: #dontrun: keep cran check time short
name.companies <- 'ELETROPAULO METROPOLITANA EL.S.PAULO S.A'
first.date <- '2005-01-01'
last.date <- '2006-01-01'

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

## End(Not run)
```

---

`gdfpd.read.dfp.zip.file`

*Reads a single zip file downloaded from Bovespa*

---

### Description

Reads a single zip file downloaded from Bovespa

### Usage

```
gdfpd.read.dfp.zip.file(my.zip.file, folder.to.unzip = tempdir(), id.type)
```

### Arguments

`my.zip.file` Full path to zip file  
`folder.to.unzip` Folder to unzip files (default = `tempdir()`)  
`id.type` The type of file structure ('after 2011' or 'before 2011')

**Value**

A list with several dataframes containing financial statements

**Examples**

```
my.f <- system.file('extdata/9512_PETR_2002-12-31.zip', package = 'GetDFPData')  
  
#my.l <- gdfpd.read.dfp.zip.file(my.f, id.type = 'before 2011')  
#print(my.l)
```

---

```
gdfpd.read.dfp.zip.file.type.1  
  Reads folder for zip file post 2011 (internal)
```

---

**Description**

Reads folder for zip file post 2011 (internal)

**Usage**

```
gdfpd.read.dfp.zip.file.type.1(rnd.folder.name, folder.to.unzip = tempdir())
```

**Arguments**

```
rnd.folder.name  
  Folder where unzipped files are available  
folder.to.unzip  
  Folder to unzip files (default = tempdir())
```

**Value**

A list with financial statements

**Examples**

```
# no example (this functions not used directly)
```

---

```
gdfpd.read.dfp.zip.file.type.2
    Reads folder for zip file pre 2011 (internal)
```

---

**Description**

Reads folder for zip file pre 2011 (internal)

**Usage**

```
gdfpd.read.dfp.zip.file.type.2(rnd.folder.name, folder.to.unzip = tempdir())
```

**Arguments**

```
rnd.folder.name      Folder where unzipped files are available
folder.to.unzip      Folder to unzip files (default = tempdir())
```

**Value**

A list with financial statements

**Examples**

```
# no example (this functions not used directly)
```

---

```
gdfpd.read.fca.zip.file
    Reads a single FCA zip file downloaded from Bovespa
```

---

**Description**

Reads a single FCA zip file downloaded from Bovespa

**Usage**

```
gdfpd.read.fca.zip.file(my.zip.file, folder.to.unzip = tempdir())
```

**Arguments**

```
my.zip.file          Full path to zip file
folder.to.unzip      Folder to unzip files, default = tempdir()
```

**Value**

A list with several dataframes containing financial statements

**Examples**

```
my.f <- system.file('extdata/FCA_9512_PETR_2015-12-31.zip', package = 'GetDFPData')  
  
my.l <- gdfpd.read.fca.zip.file(my.f)  
print(my.l)
```

---

gdfpd.read.fre.zip.file

*Reads a single FRE zip file downloaded from Bovespa*

---

**Description**

Reads a single FRE zip file downloaded from Bovespa

**Usage**

```
gdfpd.read.fre.zip.file(my.zip.file, folder.to.unzip = tempdir())
```

**Arguments**

my.zip.file      Full path to zip file  
folder.to.unzip      Folder to unzip files (default = tempdir())

**Value**

A list with several dataframes containing financial statements

**Examples**

```
my.f <- system.file('extdata/FRE_6629_HERC_2010-12-31.zip', package = 'GetDFPData')  
  
my.l <- gdfpd.read.fre.zip.file(my.f)  
print(my.l)
```

---

`gdfpd.read.fwf.file` *Reads FWF file from bovespa (internal)*

---

### Description

Reads FWF file from bovespa (internal)

### Usage

```
gdfpd.read.fwf.file(my.f, flag.thousands)
```

### Arguments

`my.f` File to be read  
`flag.thousands` A flag for thousands values

### Value

A dataframe with data

### Examples

```
my.f <- system.file('extdata/DFPBPAE.001', package = 'GetDFPData')  
df.assets <- gdfpd.read.fwf.file(my.f, flag.thousands = FALSE)
```

---

`gdfpd.read.zip.file.type.fca`  
*Reads folder for FCA zip file contents (internal)*

---

### Description

Reads folder for FCA zip file contents (internal)

### Usage

```
gdfpd.read.zip.file.type.fca(rnd.folder.name, folder.to.unzip = tempdir())
```

### Arguments

`rnd.folder.name`  
Folder where unzipped files are available  
`folder.to.unzip`  
Folder to unzip files, default = tempdir()

**Value**

A list with FCA data

**Examples**

```
# no example (this functions is not used directly)
```

---

```
gdfpd.read.zip.file.type.fre  
    Reads folder for zip file post 2011 (internal)
```

---

**Description**

Reads folder for zip file post 2011 (internal)

**Usage**

```
gdfpd.read.zip.file.type.fre(rnd.folder.name, folder.to.unzip = tempdir())
```

**Arguments**

```
rnd.folder.name  
    Folder where unzipped files are available  
folder.to.unzip  
    Folder to unzip files (default = tempdir())
```

**Value**

A list with financial statements

**Examples**

```
# no example (this functions not used directly)
```



---

gdfpd.search.company *Helps users search for a company name*

---

**Description**

Helps users search for a company name

**Usage**

```
gdfpd.search.company(char.to.search, cache.folder = "DFP Cache Folder")
```

**Arguments**

char.to.search Character for partial matching  
cache.folder Folder to cache (save) all processed information. Default = file.path(getwd(), 'DFP Cache Folder')

**Value**

Names of found companies

**Examples**

```
## Not run: # dontrun: keep cran check fast  
gdfpd.search.company('GERDAU')  
  
## End(Not run)
```

---

get\_files *Fetches files for different systems (INTERNAL)*

---

**Description**

Fetches files for different systems (INTERNAL)

**Usage**

```
get_files(my.id, type.fin.report)
```

**Arguments**

my.id Company id  
type.fin.report type of financial report (dfp/itr/fre/fca)

**Value**

A dataframe

**Examples**

```
## Not run:  
df.fre.files <- get_files(9512, type.fin.report = 'dfp')  
  
## End(Not run)
```

---

my.copy.fct

*Copies data to external file*

---

**Description**

Copies data to external file

**Usage**

```
my.copy.fct(  
  df.in,  
  name.df,  
  base.file.name,  
  type.export = "xlsx",  
  csv.dir = tempdir()  
)
```

**Arguments**

df.in	Dataframe to be copied
name.df	Name of dataframe to be copied
base.file.name	The basename of excel file (make sure you dont include the file extension)
type.export	The extension of the desired format: 'xlsx' (default) or 'csv'
csv.dir	Location where to save csv files prior to zipping (default = tempdir())

**Value**

TRUE (invisible), if successfull

**Examples**

```
test.data <- data.frame(test.data = runif(100))  
name.df <- 'TestData'  
base.file.name <- 'TestData'  
type.export <- 'csv'  
  
my.copy.fct(df.in = test.data, name.df, base.file.name, type.export)
```

---

my.merge.dfs.lists	<i>Merges (row wise) dataframes from different list, using names of dataframes as index</i>
--------------------	---

---

**Description**

Merges (row wise) dataframes from different list, using names of dataframes as index

**Usage**

```
my.merge.dfs.lists(l.1, l.2)
```

**Arguments**

l.1	First dataframe
l.2	Second dataframe

**Value**

A list with binded dataframes (same names as l.1)

**Examples**

```
l.1 <- list(x = data.frame(runif(10)) )  
l.2 <- list(x = data.frame(runif(10)) )  
  
l <- my.merge.dfs.lists(l.1, l.2)
```

---

xml.fct.auditing	<i>Reads XML data for auditing</i>
------------------	------------------------------------

---

**Description**

Reads XML data for auditing

**Usage**

```
xml.fct.auditing(x)
```

**Arguments**

x	A list with data
---	------------------

**Value**

A dataframe

**Examples**

```
# No example (INTERNAL)
```

---

```
xml.fct.board.composition  
Reads XML data for board composition
```

---

**Description**

Reads XML data for board composition

**Usage**

```
xml.fct.board.composition(x)
```

**Arguments**

x                    A list with data

**Value**

A dataframe

**Examples**

```
# No example (INTERNAL)
```

---

```
xml.fct.capital            Reads XML data for capita
```

---

**Description**

Reads XML data for capita

**Usage**

```
xml.fct.capital(x)
```

**Arguments**

x                    A list with capital summary data

**Value**

A dataframe

**Examples**

```
# No example (INTERNAL)
```

---

```
xml.fct.capital.reduction  
Reads XML data for capital reduction data
```

---

**Description**

Reads XML data for capital reduction data

**Usage**

```
xml.fct.capital.reduction(x)
```

**Arguments**

x                    A list with data

**Value**

A dataframe

**Examples**

```
# No example (INTERNAL)
```

---

```
xml.fct.committee.composition  
Reads XML data for commitee composition
```

---

**Description**

Reads XML data for commitee composition

**Usage**

```
xml.fct.committee.composition(x)
```

**Arguments**

x                    A list with data

**Value**

A dataframe

**Examples**

```
# No example (INTERNAL)
```

---

xml.fct.compensation    *Reads XML data for compensation*

---

**Description**

Reads XML data for compensation

**Usage**

```
xml.fct.compensation(x)
```

**Arguments**

x                    A list with compensation data

**Value**

A dataframe

**Examples**

```
# No example (INTERNAL)
```

---

xml.fct.compensation.summary

*Reads XML data for compensation summary data*

---

### **Description**

Reads XML data for compensation summary data

### **Usage**

xml.fct.compensation.summary(x)

### **Arguments**

x                    A list with compensation summary data

### **Value**

A dataframe

### **Examples**

```
# No example (INTERNAL)
```

---

xml.fct.debt

*Reads XML data for debt*

---

### **Description**

Reads XML data for debt

### **Usage**

xml.fct.debt(x)

### **Arguments**

x                    A list with data

### **Value**

A dataframe

### **Examples**

```
# No example (INTERNAL)
```

xml.fct.div.details    *Reads XML data for div details*

---

**Description**

Reads XML data for div details

**Usage**

```
xml.fct.div.details(x)
```

**Arguments**

x                    A list with data

**Value**

A dataframe

**Examples**

```
# No example (INTERNAL)
```

---

xml.fct.family.related.parts  
                          *Reads XML data for family related parts*

---

**Description**

Reads XML data for family related parts

**Usage**

```
xml.fct.family.related.parts(x)
```

**Arguments**

x                    A list with data

**Value**

A dataframe

**Examples**

```
# No example (INTERNAL)
```



---

xml.fct.family.relations  
*Reads XML data for family relations*

---

**Description**

Reads XML data for family relations

**Usage**

xml.fct.family.relations(x)

**Arguments**

x                    A list with data

**Value**

A dataframe

**Examples**

# No example (INTERNAL)

---

xml.fct.intangible.details  
*Reads XML data for patents details*

---

**Description**

Reads XML data for patents details

**Usage**

xml.fct.intangible.details(x)

**Arguments**

x                    A list with data

**Value**

A dataframe

**Examples**

```
# No example (INTERNAL)
```

---

```
xml.fct.repurchases    Reads XML data for repurchases
```

---

**Description**

Reads XML data for repurchases

**Usage**

```
xml.fct.repurchases(x)
```

**Arguments**

x                    A list with data

**Value**

A dataframe

**Examples**

```
# No example (INTERNAL)
```

---

```
xml.fct.responsible    Reads XML data for responsables documents
```

---

**Description**

Reads XML data for responsables documents

**Usage**

```
xml.fct.responsible(x)
```

**Arguments**

x                    A list with data

**Value**

A dataframe

### Examples

```
# No example (INTERNAL)
```

---

```
xml.fct.splits.inplits
```

*Reads XML data for splits/inplits data*

---

### Description

Reads XML data for splits/inplits data

### Usage

```
xml.fct.splits.inplits(x)
```

### Arguments

x                    A list with data

### Value

A dataframe

### Examples

```
# No example (INTERNAL)
```

---

```
xml.fct.stock.values
```

*Reads XML data for stock value*

---

### Description

Reads XML data for stock value

### Usage

```
xml.fct.stock.values(x)
```

### Arguments

x                    A list with stock value data

### Value

A dataframe

**Examples**

```
# No example (INTERNAL)
```

---

```
xml.fct.stockholder    Reads XML data for stockholder data
```

---

**Description**

Reads XML data for stockholder data

**Usage**

```
xml.fct.stockholder(x)
```

**Arguments**

x                    A list with stockholder data

**Value**

A dataframe

**Examples**

```
# No example (INTERNAL)
```

---

```
xml.fct.stocks.details  
                          Reads XML data for stock details
```

---

**Description**

Reads XML data for stock details

**Usage**

```
xml.fct.stocks.details(x)
```

**Arguments**

x                    A list with data

**Value**

A dataframe

### **Examples**

```
# No example (INTERNAL)
```

---

```
xml.fct.transactions.related  
Reads XML data for transaction data
```

---

### **Description**

Reads XML data for transaction data

### **Usage**

```
xml.fct.transactions.related(x)
```

### **Arguments**

x                    A list with transaction data

### **Value**

A dataframe

### **Examples**

```
# No example (INTERNAL)
```

# Index

[fix.fct](#), 3

[gdfpd.convert.to.wide](#), 3  
[gdfpd.download.file](#), 4  
[gdfpd.export.DFP.data](#), 5  
[gdfpd.fix.DFP.dataframes](#), 6  
[gdfpd.get.bovespa.data](#), 7  
[gdfpd.get.files.from.bovespa](#), 7  
[gdfpd.get.inflation.data](#), 8  
[gdfpd.get.info.companies](#), 9  
[gdfpd.GetDFPDData](#), 10  
[gdfpd.read.dfp.zip.file](#), 11  
[gdfpd.read.dfp.zip.file.type.1](#), 12  
[gdfpd.read.dfp.zip.file.type.2](#), 13  
[gdfpd.read.fca.zip.file](#), 13  
[gdfpd.read.fre.zip.file](#), 14  
[gdfpd.read.fwf.file](#), 15  
[gdfpd.read.zip.file.type.fca](#), 15  
[gdfpd.read.zip.file.type.fre](#), 16  
[gdfpd.search.company](#), 17  
[get\\_files](#), 17

[my.copy.fct](#), 18

[my.merge.dfs.lists](#), 19

[xml.fct.auditing](#), 19  
[xml.fct.board.composition](#), 20  
[xml.fct.capital](#), 20  
[xml.fct.capital.reduction](#), 21  
[xml.fct.committee.composition](#), 21  
[xml.fct.compensation](#), 22  
[xml.fct.compensation.summary](#), 23  
[xml.fct.debt](#), 23  
[xml.fct.div.details](#), 24  
[xml.fct.family.related.parts](#), 24  
[xml.fct.family.relations](#), 25  
[xml.fct.intangible.details](#), 25  
[xml.fct.repurchases](#), 26  
[xml.fct.responsible](#), 26  
[xml.fct.splits.inplits](#), 27

[xml.fct.stock.values](#), 27

[xml.fct.stockholder](#), 28

[xml.fct.stocks.details](#), 28

[xml.fct.transactions.related](#), 29