

Package ‘fImport’

July 21, 2022

Title Rmetrics - Importing Economic and Financial Data

Date 2022-07-21

Version 4021.86

Description Provides a collection of utility functions
to download and manage data sets from the Internet or from other
sources.

Depends R (>= 2.15.1), timeDate, timeSeries

Imports methods, utils

Suggests RUnit

LazyData yes

License GPL (>= 2)

URL <https://r-forge.r-project.org/scm/viewvc.php/pkg/fImport/?root=rmetrics>
(devel), <https://www.rmetrics.org>

BugReports <https://r-forge.r-project.org/projects/rmetrics>

NeedsCompilation no

Author Diethelm Wuertz [aut] (original code),
Tobias Setz [aut],
Yohan Chalabi [aut],
Georgi N. Boshnakov [cre]

Maintainer Georgi N. Boshnakov <georgi.boshnakov@manchester.ac.uk>

Repository CRAN

Date/Publication 2022-07-21 08:40:06 UTC

R topics documented:

fImport-package	2
class-fWEBDATA	3
import-fred	4
provider-Listings	6
read-lines	7

read-links	7
read-lynx	8
read-w3m	9
show-methods	10
utils-download	11
utils-split	11
Index	13

fImport-package	<i>Import data from the web</i>
-----------------	---------------------------------

Description

The Rmetrics "fImport" package is a collection of utility functions to download and manage data sets from the Internet or from other sources.

1 Introduction

The major content of this package is to provide download functions for financial market data from the Internet. For this we have implemented the web text browsers "Lynx", "Links", and "W3M" for an easy and straightforward download of data from the Internet.

Furthermore helpful utility functions are included to split numerical data matrices, to split date character vectors, and to split strings from downloads. This allows to create in a very easy way `timeSeries` objects.

Examples are provided for downloading data from the Federal Reserve data base in St. Louis web portal.

The data part contains instruments listings from the American Stock Exchange, from the FED H15 Report, from the NASDAQ Stock Market, from the New York Stock Exchange, of OANDAs Foreign Exchange Rates, of STOXX Indices, and from the Swiss Stock Exchange.

For the download of spread sheets from the Internet we refer to the functions `gdata::read.xls` and `xlsx::read.xlsx` for the contributed R packages `gdata` and `xlsx` respectively.

2 Download Functions

The package makes functions available to download financial market data from the internet. Currently functions are available for the following web sites. The functions are:

`fredSeries` downloads data from `research.stlouisfed.org`

The economic and financial time series data are extractes as objects of class "timeSeries".

3 Readers and Web downloaders

The package comes with the following tailored readers and web downloaders:

read.lines	a synonym function call to readLines
read.links	uses the links browser to read from a web page
read.lynx	uses the lynx browser to read from a web page
read.w3m	uses the w3m browser to read from a web page

4 Split Function Utilities

This section provides functions to split numerical data matrices, to split date character vectors, and to split strings from downloads:

dataSplit	splits a data matrix from a downloaded file
charvecSplit	splits a charvec vector from a downloaded file
stringSplit	splits a string vector from a downloaded file

About Rmetrics

The fImport Rmetrics package is written for educational support in teaching "Computational Finance and Financial Engineering" and licensed under the GPL.

class-fWEBDATA	<i>Class "fWEBDATA"</i>
----------------	-------------------------

Description

The class fWEBDATA represents a download from the internet.

Objects from the Class

Objects can be created by calls of the import or series functions.

Slots

call: Object of class "call": the call of the applied function.

data: Object of class "data.frame": the data as downloaded formatted as a data.frame.

param: Object of class "character": a character vector whose elements contain the values of selected parameters of the argument list.

title: Object of class "character": a character string with the name of the download. This can be overwritten specifying a user defined input argument.

description: Object of class "character": a character string with an optional user defined description. By default just the current date and user when the test was applied will be returned.

Methods

`show signature(object = "fWEBDATA")`: prints an object of class 'fWEBDATA'.

Note

The import and series functions like `fredImport` and `fredSeries` are typical examples which show how to implement download functions as simple `timeSeries` objects or as more complicated `fWEBDATA` S4 objects. Inspect the R code and feel free to create your own download functions and objects.

References

Diethelm Wuertz, Yohan Chalabi, and Andrew Ellis, (2010); *Financial Market Data for R/Rmetrics*, Rmetrics eBook, Rmetrics Association and Finance Online, Zurich, www.rmetrics.org.

import-fred

Import Market Data from the Federal Reserve Database

Description

Imports financial time series data from research.stlouisfed.org.

Usage

```
fredSeries(symbols, from = NULL, to = Sys.timeDate(),
           nDaysBack = 366, ...)
```

```
fredImport(query, file = "tempfile", source = NULL, frequency = "daily",
           from = NULL, to = Sys.timeDate(), nDaysBack = NULL,
           save = FALSE, sep = ";", try = TRUE)
```

Arguments

<code>file</code>	a character string with filename, usually having extension ".csv", where to save the downloaded data.
<code>frequency</code>	a character string, one of "auto", "quarterly", "monthly", or "daily", defining the frequency of the data records. Note, the import function tries autodetect the frequency of the time series to be downloaded. This may fail, in such case specify the frequency explicitly.
<code>from</code>	the date from when to extract the time series.
<code>nDaysBack</code>	the number of days back.
<code>query</code>	a character string, denoting the location of the data at the web site.
<code>save</code>	a logical value, if set to TRUE the downloaded data file will be stored under the path and file name specified by the string <code>file</code> . By default FALSE.
<code>sep</code>	a character value specifying the column separator.

source	a character string setting the URL of the source. If NULL, then the URL will be set automatically to its default value.
symbols	a character string with the symbols to be downloaded.
to	the end date of the data download, by default the current date.
try	a logical value, if set to TRUE the Internet access will be checked.
...	optional arguments to be passed.

Value

The function `fredImport` returns an S4 object of class `fWEBDATA` with the following slots:

@call	the function call.
@data	the data as downloaded formatted as a <code>data.frame</code> .
@param	a character vector whose elements contain the values of selected parameters of the argument list.
@title	a character string with the name of the download. This can be overwritten specifying a user defined input argument.
@description	a character string with an optional user defined description. By default just the current date when the test was applied will be returned.

The function `fredSeries` returns an S4 object of class `timeSeries` or alternatively an object specified by the function argument `returnClass`.

Note**Internet Download Functions:**

IMPORTANT NOTE: If the service provider changes the data file format it may become necessary to modify and update the functions.

Feel free to inspect the code of the functions and to create your own download function from other Internet web sites and Portals.

Author(s)

Diethelm Wuertz for the Rmetrics R-port.

References

Diethelm Wuertz, Yohan Chalabi, and Andrew Ellis, (2010); *Financial Market Data for R/Rmetrics*, Rmetrics eBook, Rmetrics Association and Finance Online, Zurich, www.rmetrics.org.

Examples

```

a = fredImport("DEXSZUS")
head(a@data) # a@data is a data frame

b = fredSeries("DEXSZUS")
head(b) # timeSeries object

```

provider-Listings *Provider Listing of Symbols and Descriptions*

Description

CSV files with provider listings of symbols, descriptions and related information.

The listings include those from:

amexListing	Listing from the American Stock Exchange
h15Listing	Listing from the FED H15 Report
nasdaqListing	Listing from the NASDAQ Stock Market
nyseListing	Listing from the New York Stock Exchange
oandaListing	Listing of OANDAs Foreign Exchange Rates
stoxxListing	Listing of STOXX Indices
swxListing	Listing from the Swiss Stock Exchange

Format

All files are given in CSV Excel spreadsheet format. The delimiter is a semicolon.

References

Diethelm Wuertz, Yohan Chalabi, and Andrew Ellis, (2010); *Financial Market Data for R/Rmetrics*, Rmetrics eBook, Rmetrics Association and Finance Online, Zurich, www.rmetrics.org.

Examples

```

data(package = "fImport")

head(h15Listing)
head(nyseListing)

```

read-lines	<i>Read from a text file line by line</i>
------------	---

Description

Reads from a text file line by line. Wrapper to readLines() function.

Usage

```
read.lines(con=stdin(), n=-1, ok=TRUE, warn=FALSE, encoding="unknown")
```

Arguments

con	a connection object or a character string.
n	an integer, the (maximal) number of lines to read. Negative values indicate that one should read up to the end of input on the connection.
ok	a logical, is it OK to reach the end of the connection before n > 0 lines are read? If not, an error will be generated.
warn	a logical, warn if a text file is missing a final EOL. The default is FALSE, note different from function readLines.
encoding	a character string, the encoding to be assumed for input strings.

Value

the downloaded text. Same output as readLines() function.

References

Diethelm Wuertz, Yohan Chalabi, and Andrew Ellis, (2010); *Financial Market Data for R/Rmetrics*, Rmetrics eBook, Rmetrics Association and Finance Online, Zurich, www.rmetrics.org.

read-links	<i>Links Browser interface</i>
------------	--------------------------------

Description

Uses the Links Text Browser to read a web page.

Usage

```
read.links(url, intern = TRUE, bin = NULL, pipe = FALSE, ...)
```

Arguments

<code>url</code>	a character string specifying the URL of the web page.
<code>intern</code>	a logical which indicates whether to make the output of the command an R object.
<code>bin</code>	a string with the path of your lynx binary or NULL if lynx binary is available in the operating system path.
<code>pipe</code>	a logical which indicates whether the result should be returned as a pipe() command.
<code>...</code>	optional arguments passed to links binary.

Value

the downloaded text.

References

Diethelm Wuertz, Yohan Chalabi, and Andrew Ellis, (2010); *Financial Market Data for R/Rmetrics*, Rmetrics eBook, Rmetrics Association and Finance Online, Zurich, www.rmetrics.org.

See Also

Alternative text browser functions are the Rmetrics functions [read.lynx](#) and [read.w3m](#).

To download xls and xlsx spread sheets use the functions `gdata::read.xls` and `xlsx::read.xlsx` from the contributed packages `gdata` and `xlsx`, respectively.

To download text files line by line use the Rmetrics function [read.lines](#) which wraps the function `readLines` from R's base environment.

To postprocess downloaded files use the Rmetrics functions [indexGrep](#), [dataSplit](#), [charvecSplit](#), and [stringSplit](#).

`read-lynx`

Lynx Browser interface

Description

Uses the Lynx Browser to read a web page.

Usage

```
read.lynx(url, intern = TRUE, bin = NULL, pipe = FALSE, ...)
```

Arguments

url	a character string specifying the URL of the web page.
intern	a logical which indicates whether to make the output of the command an R object.
bin	a string with the path of your lynx binary or NULL if lynx binary is available in the operating system path.
pipe	a logical which indicates whether the result should be returned as a pipe() command.
...	optional arguments passed to lynx binary. For example <code>accept_all_cookies = TRUE</code> or <code>cookie_file = "~/lynx_cookies"</code> . For a list of options, see the lynx manual page.

Value

the downloaded text.

References

Diethelm Wuertz, Yohan Chalabi, and Andrew Ellis, (2010); *Financial Market Data for R/Rmetrics*, Rmetrics eBook, Rmetrics Association and Finance Online, Zurich, www.rmetrics.org.

See Also

Alternative text browser functions are the Rmetrics functions [read.links](#) and [read.w3m](#).

To download xls and xlsx spread sheets use the functions `gdata::read.xls` and `xlsx::read.xlsx` from the contributed packages `gdata` and `xlsx`, respectively.

To download text files line by line use the Rmetrics function [read.lines](#) which wraps the function `readLines` from R's base environment.

To postprocess downloaded files use the Rmetrics functions [indexGrep](#), [dataSplit](#), [charvecSplit](#), and [stringSplit](#).

read-w3m

w3m Browser interface

Description

Uses the w3m Browser to read a web page.

Usage

```
read.w3m(url, intern = TRUE, bin = NULL, pipe = FALSE, ...)
```

Arguments

<code>url</code>	a character string specifying the URL of the web page.
<code>intern</code>	a logical which indicates whether to make the output of the command an R object.
<code>bin</code>	a string with the path of your w3m binary or NULL if w3m binary is available in the operating system path.
<code>pipe</code>	a logical which indicates whether the result should be returned as a pipe() command.
<code>...</code>	optional arguments passed to w3m binary. For a list of options, see the w3m manual page.

Value

the downloaded text.

References

Diethelm Wuertz, Yohan Chalabi, and Andrew Ellis, (2010); *Financial Market Data for R/Rmetrics*, Rmetrics eBook, Rmetrics Association and Finance Online, Zurich, www.rmetrics.org.

See Also

Alternative text browser functions are the Rmetrics functions [read.links](#) and [read lynx](#).

To download xls and.xlsx spread sheets use the functions `gdata::read.xls` and `xlsx::read.xlsx` from the contributed packages `gdata` and `xlsx`, respectively.

To download text files line by line use the Rmetrics function [read.lines](#) which wraps the function `readLines` from R's base environment.

To postprocess downloaded files use the Rmetrics functions [indexGrep](#), [dataSplit](#), [charvecSplit](#), and [stringSplit](#).

show-methods

WEBDATA Download Show Methods

Description

Show methods for WEBDATA downloads.

Methods

object = "ANY" Generic function.

object = "fWEBDATA" Print function for objects of class "fWEBDATA".

References

Diethelm Wuertz, Yohan Chalabi, and Andrew Ellis, (2010); *Financial Market Data for R/Rmetrics*, Rmetrics eBook, Rmetrics Association and Finance Online, Zurich, www.rmetrics.org.

utils-download	<i>Utilities for composing URL's</i>
----------------	--------------------------------------

Description

Two helpful utilities for assembling URL's.

Usage

```
composeURL(..., prefix="http://")
indexGrep(pattern, x, ...)
```

Arguments

...	for composeURL, character strings from which the URL will be composed; for indexGrep, optional arguments to be passed to the function grep.
prefix	a character string specifying the prefix of the URL.
pattern	a character string containing a regular expression to be matched in the given character vector.
x	a character vector where matches are sought.

References

Diethelm Wuertz, Yohan Chalabi, and Andrew Ellis, (2010); *Financial Market Data for R/Rmetrics*, Rmetrics eBook, Rmetrics Association and Finance Online, Zurich, www.rmetrics.org.

utils-split	<i>Split downloaded data sets</i>
-------------	-----------------------------------

Description

Helpful dataset and charvec splitting utilities.

Usage

```
dataSplit(x, split=" ", col=-1)
charvecSplit(x, split=" ", col=1, format="%F")
stringSplit(x, split=" ", col=NULL)
```

Arguments

x	character vector to be splitted.
split	the split character, by default a blank.
col	an integer value or vector, the column(s) to be selected.
format	the date format of the character vector, by default the ISO-8601 date format.

References

Diethelm Wuertz, Yohan Chalabi, and Andrew Ellis, (2010); *Financial Market Data for R/Rmetrics*, Rmetrics eBook, Rmetrics Association and Finance Online, Zurich, www.rmetrics.org.

Index

- * **datasets**
 - provider-Listings, 6
- * **data**
 - class-fWEBDATA, 3
 - import-fred, 4
 - show-methods, 10
- * **package**
 - fImport-package, 2
- * **programming**
 - read-lines, 7
 - read-links, 7
 - read-lynx, 8
 - read-w3m, 9
 - utils-download, 11
 - utils-split, 11
- * **ts**
 - import-fred, 4
- amexListing (provider-Listings), 6
- charvecSplit, 8–10
- charvecSplit (utils-split), 11
- class-fWEBDATA, 3
- composeURL (utils-download), 11
- dataSplit, 8–10
- dataSplit (utils-split), 11
- fImport (fImport-package), 2
- fImport-package, 2
- fredImport, 4
- fredImport (import-fred), 4
- fredSeries, 4
- fredSeries (import-fred), 4
- fWEBDATA-class (class-fWEBDATA), 3
- h15Listing (provider-Listings), 6
- import-fred, 4
- indexGrep, 8–10
- indexGrep (utils-download), 11
- nasdaqListing (provider-Listings), 6
- nyseListing (provider-Listings), 6
- oandaListing (provider-Listings), 6
- provider-Listings, 6
- providerListings (provider-Listings), 6
- read-lines, 7
- read-links, 7
- read-lynx, 8
- read-w3m, 9
- read.lines, 8–10
- read.lines (read-lines), 7
- read.links, 9, 10
- read.links (read-links), 7
- read.lynx, 8, 10
- read.lynx (read-lynx), 8
- read.w3m, 8, 9
- read.w3m (read-w3m), 9
- show, ANY-method (show-methods), 10
- show, fWEBDATA-method (show-methods), 10
- show-methods, 10
- stoxxListing (provider-Listings), 6
- stringSplit, 8–10
- stringSplit (utils-split), 11
- swxListing (provider-Listings), 6
- utils-download, 11
- utils-split, 11