

Package ‘readabs’

October 8, 2019

Type Package

Title Download and Tidy Time Series Data from the Australian Bureau of Statistics

Version 0.4.2

Maintainer Matt Cowgill <mattcowgill@gmail.com>

Description Downloads, imports, and tidies time series data from the Australian Bureau of Statistics <<https://www.abs.gov.au/>>.

Date 2019-10-08

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Depends R (>= 3.3)

Imports readxl (>= 1.2.0), tibble (>= 1.4.99), dplyr (>= 0.8.0), XML, curl, purrr, tidyr, stringr, rsdmx, readr

URL <https://github.com/mattcowgill/readabs>

BugReports <https://github.com/mattcowgill/readabs/issues>

RoxygenNote 6.1.1

VignetteBuilder knitr

Suggests knitr, rmarkdown, testthat (>= 2.1.0), RCurl, ggplot2

NeedsCompilation no

Author Matt Cowgill [aut, cre],
Zoe Meers [aut],
Jaron Lee [aut],
David Diviny [ctb],
Hugh Parsonage [ctb]

Repository CRAN

Date/Publication 2019-10-08 06:20:10 UTC

R topics documented:

extract_abs_sheets	2
get_abs	3
read_abs	3
read_abs_data	4
read_abs_local	5
read_abs_metadata	6
read_abs_sdmx	6
read_cpi	7
separate_series	8
tidy_abs	9
tidy_abs_list	9

Index	11
--------------	-----------

extract_abs_sheets	<i>Extract data sheets from an ABS timeseries workbook saved locally as an .xls file.</i>
--------------------	---

Description

Note that this function will not tidy the data for you. Use ‘read_abs_local()’ to import and tidy data from local ABS time series spreadsheets or ‘read_abs()’ to download, import and tidy ABS time series.

Usage

```
extract_abs_sheets(filename, table_title = NULL,
  path = Sys.getenv("R_READABS_PATH", unset = tempdir()))
```

Arguments

filename	Filename for an ABS time series spreadsheet (as string), such as "6202002.xls".
table_title	String giving the full title of the ABS table, such as "Table 1. Employed persons, Australia"
path	Local directory in which an ABS time series is stored. Default is ‘Sys.getenv("R_READABS_PATH", unset = tempdir())’.

get_abs	<i>Download, extract, and tidy ABS time series spreadsheets (deprecated)</i>
---------	--

Description

get_abs() is deprecated. Please use read_abs() instead. It has identical functionality.

Usage

```
get_abs()
```

read_abs	<i>Download, extract, and tidy ABS time series spreadsheets</i>
----------	---

Description

read_abs() downloads ABS time series spreadsheets, then extracts the data from those spreadsheets, then tidies the data. The result is a single data frame (tibble) containing tidied data.

Usage

```
read_abs(cat_no = NULL, tables = "all", series_id = NULL,
         path = Sys.getenv("R_READABS_PATH", unset = tempdir()),
         metadata = TRUE, show_progressBars = TRUE, retain_files = TRUE)
```

Arguments

cat_no	ABS catalogue number, as a string, including the extension. For example, "6202.0".
tables	numeric. Time series tables in 'cat_no' to download and extract. Default is "all", which will read all time series in 'cat_no'. Specify 'tables' to download and import specific tables(s) - eg. 'tables = 1' or 'tables = c(1, 5)'.
series_id	(optional) character. Supply an ABS unique time series identifier (such as "A2325807L") to get only that series. This is an alternative to specifying 'cat_no'.
path	Local directory in which downloaded ABS time series spreadsheets should be stored. By default, 'path' takes the value set in the environment variable "R_READABS_PATH". If this variable is not set, any files downloaded by read_abs() will be stored in a temporary directory (tempdir()). See Details below for more information.
metadata	logical. If 'TRUE' (the default), a tidy data frame including ABS metadata (series name, table name, etc.) is included in the output. If 'FALSE', metadata is dropped.
show_progressBars	TRUE by default. If set to FALSE, progress bars will not be shown when ABS spreadsheets are downloading.

`retain_files` when TRUE (the default), the spreadsheets downloaded from the ABS website will be saved in the directory specified with `'path'`. If set to `'FALSE'`, the files will be stored in a temporary directory.

Details

`'read_abs()'` downloads spreadsheet(s) from the ABS containing time series data. These files need to be saved somewhere on your disk. This local directory can be controlled using the `'path'` argument to `'read_abs()'`. If the `'path'` argument is not set, `'read_abs()'` will store the files in a directory set in the `"R_READABS_PATH"` environment variable. If this variable isn't set, files will be saved in a temporary directory.

To check the value of the `"R_READABS_PATH"` variable, run `Sys.getenv("R_READABS_PATH")`. You can set the value of this variable for a single session using `Sys.setenv(R_READABS_PATH = <path>)`. If you would like to change this variable for all future R sessions, edit your `'.Renviron'` file and add `R_READABS_PATH = <path>` line. The easiest way to edit this file is using `usethis::edit_r_environ()`.

Value

A data frame (tibble) containing the tidied data from the ABS time series table(s).

Examples

```
# Download and tidy all time series spreadsheets from the Wage Price Index (6345.0)
wpi <- read_abs("6345.0")

# Get two specific time series, based on their time series IDs
cpi <- read_abs(series_id = c("A2325806K", "A2325807L"))
```

<code>read_abs_data</code>	<i>Extracts ABS time series data from local Excel spreadsheets and converts to long format.</i>
----------------------------	---

Description

`'read_abs_data()'` is soft deprecated and will be removed in a future version. Please use `'read_abs_local()'` to import and tidy locally-stored ABS time series spreadsheets, or `'read_abs()'` to download, import, and tidy time series spreadsheets from the ABS website.

Usage

```
read_abs_data(path, sheet)
```

Arguments

path	Filepath to Excel spreadsheet.
sheet	Sheet name or number.

Value

Long-format dataframe

read_abs_local	<i>Read and tidy locally-saved ABS time series spreadsheet(s)</i>
----------------	---

Description

If you need to download and tidy time series data from the ABS, use `read_abs()`. `read_abs_local()` imports and tidies data from ABS time series spreadsheets that are already saved to your local drive.

Usage

```
read_abs_local(cat_no = NULL, filenames = NULL,
              path = Sys.getenv("R_READABS_PATH", unset = tempdir()),
              metadata = TRUE)
```

Arguments

cat_no	character; a single catalogue number such as "6202.0". When 'cat_no' is specified, all local files in 'path' corresponding to the specified catalogue number will be imported. For example, if you run 'read_abs_local("6202.0")', it will look in "data/ABS/6202.0" and attempt to load any .xls files in that location. If 'cat_no' is specified, 'filenames' will be ignored.
filenames	character vector of at least one filename of a locally-stored ABS time series spreadsheet. For example, "6202001.xls" or c("6202001.xls", "6202005.xls"). Ignored if a value is supplied to 'cat_no'. If 'filenames' is blank and 'cat_no' is blank, 'read_abs_local()' will attempt to read all .xls files in the directory specified with 'path'.
path	path to local directory containing ABS time series file(s). Default is 'Sys.getenv("R_READABS_PATH", unset = tempdir())'. If nothing is specified in 'filenames' or 'cat_no', 'read_abs_local()' will attempt to read all .xls files in the directory specified with 'path'.
metadata	logical. If 'TRUE' (the default), a tidy data frame including ABS metadata (series name, table name, etc.) is included in the output. If 'FALSE', metadata is dropped.

Details

Unlike `read_abs()`, the 'table_title' column in the data frame returned by `read_abs_local()` is blank. If you require 'table_title', please use `read_abs()` instead.

Examples

```
# Load and tidy two specified files from the "data/ABS" subdirectory
# of your working directory

lfs <- read_abs_local(c("6202001.xls", "6202005.xls"))
```

read_abs_metadata	<i>Extracts ABS series metadata directly from Excel spreadsheets and converts to long-form.</i>
-------------------	---

Description

Extracts ABS series metadata directly from Excel spreadsheets and converts to long-form.

Usage

```
read_abs_metadata(path, sheet)
```

Arguments

path	Filepath to Excel spreadsheet.
sheet	Sheet name or number.

Value

Long-form dataframe

read_abs_sdmx	<i>Extracts ABS XML-formatted data using the exported XML link at http://www.abs.gov.au/</i>
---------------	---

Description

Access the sdmx URLs at 'http://www.abs.gov.au/ausstats/abs@.nsf/Lookup/1407.0.55.002Main+Features4User+Guide'

Usage

```
read_abs_sdmx(url)
```

Arguments

url	URL weblink.
-----	--------------

Value

data frame

Examples

```
url <- paste0("http://stat.data.abs.gov.au/restsdmx/sdmx.ashx/GetData/LF/",
             "0.2+3+4+11+13+6+15+14+10.3+1+2.1519+1599.10+20+30.M/",
             "all?startTime=2017-12&endTime=2018-11")
lfs <- read_abs_sdmx(url)
lfs
```

read_cpi	<i>Download a tidy tibble containing the Consumer Price Index from the ABS</i>
----------	--

Description

`read_cpi()` uses the `read_abs()` function to download, import, and tidy the Consumer Price Index from the ABS. It returns a tibble containing two columns: the date and the CPI index value that corresponds to that date. This makes joining the CPI to another dataframe easy. `read_cpi()` returns the original (ie. not seasonally adjusted) all groups CPI for Australia. If you want the analytical series (eg. seasonally adjusted CPI, or trimmed mean CPI), you can use `read_abs()`.

Usage

```
read_cpi(path = Sys.getenv("R_READABS_PATH", unset = tempdir()),
         show_progressBars = TRUE, retain_files = FALSE)
```

Arguments

<code>path</code>	character; default is "data/ABS". Only used if <code>retain_files</code> is set to <code>TRUE</code> . Local directory in which to save downloaded ABS time series spreadsheets.
<code>show_progressBars</code>	logical; <code>TRUE</code> by default. If set to <code>FALSE</code> , progress bars will not be shown when ABS spreadsheets are downloading.
<code>retain_files</code>	logical; <code>FALSE</code> by default. When <code>TRUE</code> , the spreadsheets downloaded from the ABS website will be saved in the directory specified with <code>'path'</code> .

Examples

```
# Create a tibble called 'cpi' that contains the CPI index numbers for each quarter
cpi <- read_cpi()

# This tibble can now be joined to another to help streamline the process of
```

```
# deflating nominal values.
```

separate_series	<i>Separate the series column in a tidy ABS time series data frame</i>
-----------------	--

Description

Separate the 'series' column in a data frame (tibble) downloaded using `read_abs()` into multiple columns using the ";" separator.

Usage

```
separate_series(data, column_names = NULL, remove_totals = FALSE,
  remove_nas = FALSE)
```

Arguments

<code>data</code>	A data frame (tibble) containing tidied data from the ABS time series table(s).
<code>column_names</code>	(optional) character vector. Supply a vector of column names, such as <code>c("group_name", "variable", "geography")</code> . If not supplied, columns will be named "series_1" etc.
<code>remove_totals</code>	logical. FALSE by default. If set to TRUE, any series rows that contain the word "total" will be removed.
<code>remove_nas</code>	logical. FALSE by default. If set to TRUE, any rows containing an NA in at least one of the separated series columns will be removed.

Value

A data frame (tibble) containing the tidied data from the ABS time series table(s).

Examples

```
motor_vehicles <- read_abs("9314.0") %>% separate_series()
```

tidy_abs	<i>Tidy ABS time series data.</i>
----------	-----------------------------------

Description

Tidy ABS time series data.

Usage

```
tidy_abs(df, metadata = TRUE)
```

Arguments

df	A data frame containing ABS time series data that has been extracted using <code>extract_abs_sheets</code> .
metadata	logical. If 'TRUE' (the default), a tidy data frame including ABS metadata (series name, table name, etc.) is included in the output. If 'FALSE', metadata is dropped.

Value

data frame (tibble) in long format.

Examples

```
# First extract the data from the local spreadsheet
## Not run: wpi <- extract_abs_sheets("634501.xls")

# Then tidy the data extracted from the spreadsheet. Note that
# \code{extract_abs_sheets()} returns a list of data frames, so we need to
# subset the list.

tidy_wpi <- tidy_abs(wpi[[1]])
```

tidy_abs_list	<i>Tidy multiple dataframes of ABS time series data contained in a list.</i>
---------------	--

Description

Tidy multiple dataframes of ABS time series data contained in a list.

Usage

```
tidy_abs_list(list_of_dfs, metadata = TRUE)
```

Arguments

<code>list_of_dfs</code>	A list of dataframes containing extracted ABS time series data.
<code>metadata</code>	logical. If 'TRUE' (the default), a tidy data frame including ABS metadata (series name, table name, etc.) is included in the output. If 'FALSE', metadata is dropped.

Index

`extract_abs_sheets`, 2

`get_abs`, 3

`read_abs`, 3

`read_abs_data`, 4

`read_abs_local`, 5

`read_abs_metadata`, 6

`read_abs_sdmx`, 6

`read_cpi`, 7

`separate_series`, 8

`tidy_abs`, 9

`tidy_abs_list`, 9