

# Package ‘readabs’

October 14, 2020

**Type** Package

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

**Version** 0.4.5.1

**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** 2020-09-16

**License** MIT + file LICENSE

**Encoding** UTF-8

**LazyData** true

**Depends** R (>= 3.5)

**Imports** readxl (>= 1.2.0), tibble (>= 1.4.99), dplyr (>= 0.8.0),  
hutils (>= 1.5.0), fst, curl, purrr, tidyr (>= 1.0.0), stringr,  
stringi, rsdmx, tools, glue, httr, rvest, xml2, rlang

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

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

**RoxygenNote** 7.1.1

**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** 2020-10-14 05:20:02 UTC

R topics documented:

download_abs_data_cube . . . . .	2
extract_abs_sheets . . . . .	3
get_available_files . . . . .	4
read_abs . . . . .	5
read_abs_data . . . . .	6
read_abs_local . . . . .	7
read_abs_metadata . . . . .	8
read_abs_sdmx . . . . .	9
read_cpi . . . . .	9
scrape_abs_catalogues . . . . .	10
separate_series . . . . .	11
show_available_catalogues . . . . .	12
tidy_abs . . . . .	12
tidy_abs_list . . . . .	13
<b>Index</b>	<b>14</b>

---

download\_abs\_data\_cube

*Experimental helper function to download ABS data cubes that are not compatible with read\_abs.*

---

Description

download\_abs\_data\_cube() downloads the latest ABS data cubes based on the catalogue name (from the new website url) and cube. The function downloads the file to disk. In comparison to read\_abs() this function doesn't tidy the data.

Usage

```
download_abs_data_cube(  
  catalogue_string,  
  cube,  
  path = Sys.getenv("R_READABS_PATH", unset = tempdir())  
)
```

Arguments

- catalogue\_string

ABS catalogue name as a string from the new website. For example, Labour Force, Australia, Detailed is "labour-force-australia-detailed". The possible catalogues can be obtained using the helper function show\_available\_catalogues()
- cube

character. A character string that is either the complete filename or (uniquely) in the filename of the data cube you want to download, e.g. "EQ09". # The available filenames can be obtained using the helper function get\_available\_filenames()

**path** Local directory in which downloaded files 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.

## Details

‘download\_abs\_data\_cube()’ downloads a file from the ABS containing a data cube. 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()`.

The filepath is returned invisibly which enables piping to `unzip()` or `readxl::read_excel`.

## Examples

```
## Not run:
download_abs_data_cube(
  catalogue_string = "labour-force-australia-detailed",
  cube = "EQ09"
)

## End(Not run)
```

---

extract_abs_sheets	<i>Extract data sheets from an ABS timeseries workbook saved locally as an Excel 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)
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_available_files	<i>Helper function for download_abs_data_cube to show the available catalogues.</i>
---------------------	---

---

**Description**

This function lists the possible files that are available in a catalogue. The filename (or an unambiguous part of the filename) must be specified as a string as an argument to download\_abs\_data\_cube.

**Usage**

```
get_available_files(catalogue_string, refresh = FALSE)
```

**Arguments**

catalogue_string	character string specifying the catalogue, e.g. "labour-force-australia-detailed". You can use show_available_catalogues to find this out.
refresh	logical; 'FALSE' by default. If 'FALSE', an internal table of the available ABS catalogues is used. If 'TRUE', this table is refreshed from the ABS website.

**Value**

A tibble containing the title of the file, the filename and the complete url.

**Examples**

```
## Not run:
get_available_files("labour-force-australia-detailed")

## End(Not run)
```

read\_abs

*Download, extract, and tidy ABS time series spreadsheets***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,
  check_local = TRUE
)
```

**Arguments**

<code>cat_no</code>	ABS catalogue number, as a string, including the extension. For example, "6202.0".
<code>tables</code>	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)'.
<code>series_id</code>	(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'.
<code>path</code>	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 <code>read_abs()</code> will be stored in a temporary directory ( <code>tempdir()</code> ). See Details below for more information.
<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.
<code>show_progressBars</code>	TRUE by default. If set to FALSE, progress bars will not be shown when ABS spreadsheets are downloading.
<code>retain_files</code>	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.
<code>check_local</code>	If 'TRUE', the default, local 'fst' files are used, if present.

## 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)
## Not run:
wpi <- read_abs("6345.0")

## End(Not run)

# Get two specific time series, based on their time series IDs
## Not run:
cpi <- read_abs(series_id = c("A2325806K", "A2325807L"))

## End(Not run)
```

---

<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()),
  use_fst = TRUE,
  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'.
use_fst	logical. If 'TRUE' (the default) then, if an 'fst' file of the tidy data frame has already been saved in 'path', it is read immediately.
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
## Not run:
lfs <- read_abs_local(c("6202001.xls", "6202005.xls"))

## End(Not run)
```

---

<code>read_abs_metadata</code>	<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

<code>path</code>	Filepath to Excel spreadsheet.
<code>sheet</code>	Sheet name or number.

## Value

Long-form dataframe



read\_abs\_sdmx

*Extracts ABS XML-formatted data using the SDMX API***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**

```
## Not run:
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

## End(Not run)
```

read\_cpi

*Download a tidy tibble containing the Consumer Price Index from the ABS***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,
  check_local = FALSE,
  retain_files = FALSE
)
```

**Arguments**

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

**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.
```

---

scrape\_abs\_catalogues *Helper function for download\_abs\_data\_cube to scrape the available catalogues from the ABS website.*

---

**Description**

This function downloads a new version of the lookup table used by show\_available\_catalogues.

**Usage**

```
scrape_abs_catalogues()
```

**Value**

A tibble containing the catalogues and how they are organised on the ABS website.

---

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

```
## Not run:  
motor_vehicles <- read_abs("9314.0") %>%  
  separate_series()  
  
## End(Not run)
```

---

```
show_available_catalogues
```

*Helper function for download\_abs\_data\_cube to show the available catalogues.*

---

### Description

This function lists the possible catalogues that are available on the ABS website. These catalogues must be specified as a string as an argument to download\_abs\_data\_cube.

### Usage

```
show_available_catalogues(selected_heading = NULL, refresh = FALSE)
```

### Arguments

selected_heading	optional character string specifying the heading on the <a href="#">ABS statistics webpage</a> . e.g. "Earnings and work hours"
refresh	logical; 'FALSE' by default. If 'FALSE', an internal table of the available ABS catalogues is used. If 'TRUE', this table is refreshed from the ABS website.

### Value

a character vector of catalogues.

### Examples

```
show_available_catalogues("Earnings and work hours")
```

---

```
tidy_abs
```

*Tidy ABS time series data.*

---

### 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 extract_abs_sheets.
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")

## End(Not run)

# 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.
## Not run:
tidy_wpi <- tidy_abs(wpi[[1]])

## End(Not run)
```

---

tidy\_abs\_list

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


---

**Description**

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

**Usage**

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

**Arguments**

list_of_dfs	A list of dataframes containing extracted ABS time series data.
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.

# Index

`download_abs_data_cube`, [2](#)

`extract_abs_sheets`, [3](#)

`get_available_files`, [4](#)

`read_abs`, [5](#)

`read_abs_data`, [6](#)

`read_abs_local`, [7](#)

`read_abs_metadata`, [8](#)

`read_abs_sdmx`, [9](#)

`read_cpi`, [9](#)

`scrape_abs_catalogues`, [10](#)

`separate_series`, [11](#)

`show_available_catalogues`, [12](#)

`tidy_abs`, [12](#)

`tidy_abs_list`, [13](#)