

Package ‘getCRUCLdata’

October 26, 2020

Type Package

Title 'CRU' 'CL' v. 2.0 Climatology Client

Version 0.3.2

Description Provides functions that automate downloading and importing University of East Anglia Climate Research Unit ('CRU') 'CL' v. 2.0 climatology data, facilitates the calculation of minimum temperature and maximum temperature and formats the data into a tidy data frame as a 'tibble' or a list of 'raster' 'stack' objects for use. 'CRU' 'CL' v. 2.0 data are a gridded climatology of 1961-1990 monthly means released in 2002 and cover all land areas (excluding Antarctica) at 10 arcminutes (0.1666667 degree) resolution. For more information see the description of the data provided by the University of East Anglia Climate Research Unit, <<https://crudata.uea.ac.uk/cru/data/hrg/tmc/readme.txt>>.

License MIT + file LICENSE

Depends R (>= 3.2.0)

Imports curl, data.table, hoardr, raster, tibble, utils

LazyData TRUE

RoxygenNote 7.1.1

ByteCompile TRUE

Suggests covr, ggplot2, knitr, markdown, rmarkdown, rappdirs, roxygen2, testthat, viridis

Encoding UTF-8

URL <https://docs.ropensci.org/getCRUCLdata/>

BugReports <https://github.com/ropensci/getCRUCLdata/issues>

VignetteBuilder knitr

X-schema.org-applicationCategory Tools

X-schema.org-keywords anglia-cru, climate-data, cru-cl2, temperature, rainfall, elevation, data-access, wind, relative-humidity, solar-radiation, diurnal-temperature, frost

X-schema.org-isPartOf <https://ropensci.org>

Language en-GB**NeedsCompilation** no**Author** Adam H. Sparks [aut, cre] (<<https://orcid.org/0000-0002-0061-8359>>)**Maintainer** Adam H. Sparks <adamhsparks@gmail.com>**Repository** CRAN**Date/Publication** 2020-10-26 07:40:02 UTC

R topics documented:

create_CRU_df	2
create_CRU_stack	4
get_CRU_df	7
get_CRU_stack	9
manage_cache	11

Index	13
--------------	-----------

create_CRU_df	<i>Create a Tidy Data Frame From CRU CL v.2.0 Climatology Variables on Local Disk</i>
---------------	---

Description

Automates importing CRU CL v.2.0 climatology data and creates a tidy data frame of the data. If requested, minimum and maximum temperature may also be automatically calculated as described in the data readme.txt file. This function can be useful if you have network connection issues that mean automated downloading of the files using R does not work properly. In this instance it is recommended to use an FTP client (*e.g.*, FileZilla), web browser or command line command (*e.g.*, wget or curl) to download the files, save locally and use this function to import the data into R.

Nomenclature and units from readme.txt:

- pre** precipitation (millimetres/month)
- cv** cv of precipitation (percent)
- rd0** wet-days (number days with >0.1mm rain per month)
- tmp** mean temperature (degrees Celsius)
- dtr** mean diurnal temperature range (degrees Celsius)
- reh** relative humidity (percent)
- sunp** sunshine (percent of maximum possible (percent of day length))
- frs** ground-frost (number of days with ground-frost per month)
- wnd** 10 metre windspeed (metres/second)
- elv** elevation (automatically converted to metres)

For more information see the description of the data provided by CRU, <https://crudata.uea.ac.uk/cru/data/hrg/tmc/readme.txt>

Usage

```

create_CRU_df(
  pre = FALSE,
  pre_cv = FALSE,
  rd0 = FALSE,
  tmp = FALSE,
  dtr = FALSE,
  reh = FALSE,
  tmn = FALSE,
  tmx = FALSE,
  sunp = FALSE,
  frs = FALSE,
  wnd = FALSE,
  elv = FALSE,
  dsn = ""
)

```

Arguments

pre	Logical. Fetch precipitation (millimetres/month) from server and return in the data frame? Defaults to FALSE.
pre_cv	Logical. Fetch cv of precipitation (percent) from server and return in the data frame? Defaults to FALSE. NOTE. Setting this to TRUE will always results in pre being set to TRUE and returned as well.
rd0	Logical. Fetch wet-days (number days with >0.1millimetres rain per month) and return in the data frame? Defaults to FALSE.
tmp	Logical. Fetch temperature (degrees Celsius) and return it in the data frame? Defaults to FALSE.
dtr	Logical. Fetch mean diurnal temperature range (degrees Celsius) and return it in the data frame? Defaults to FALSE.
reh	Logical. Fetch relative humidity and return it in the data frame? Defaults to FALSE.
tmn	Logical. Calculate minimum temperature values (degrees Celsius) and return it in the data frame? Defaults to FALSE.
tmx	Logical. Calculate maximum temperature (degrees Celsius) and return it in the data frame? Defaults to FALSE.
sunp	Logical. Fetch sunshine, percent of maximum possible (percent of day length) and return it in data frame? Defaults to FALSE.
frs	Logical. Fetch ground-frost records (number of days with ground- frost per month) and return it in data frame? Defaults to FALSE.
wnd	Logical. Fetch 10m wind speed (metres/second) and return it in the data frame? Defaults to FALSE.
elv	Logical. Fetch elevation (converted to metres) and return it in the data frame? Defaults to FALSE.
dsn	Local file path where CRU CL v.2.0 .dat.gz files are located.

Value

A tidy data frame of CRU CL v. 2.0 climatology elements as a [tibble](#) object

Note

This package automatically converts elevation values from kilometres to metres.

Author(s)

Adam H Sparks, <adamhsparks@gmail.com>

See Also

[get_CRU_df](#)

Examples

```
# Create a data frame of temperature from locally available files in the
# tempdir() directory.

download.file(
  url = "https://crudata.uea.ac.uk/cru/data/hrg/tmc/grid_10min_tmp.dat.gz",
  destfile = file.path(tempdir(), "grid_10min_tmp.dat.gz")
)

CRU_tmp <- create_CRU_df(tmp = TRUE, dsn = tempdir())

CRU_tmp
```

create_CRU_stack	<i>Create a List of Raster Stack Objects From CRU CL v. 2.0 Climatology Variables on Local Disk</i>
------------------	---

Description

Automates importing CRU CL v.2.0 climatology data and creates a [stack](#) of the data. If requested, minimum and maximum temperature may also be automatically calculated as described in the data readme.txt file. This function can be useful if you have network connection issues that mean automated downloading of the files using R does not work properly. In this instance it is recommended to use an FTP client (*e.g.*, FileZilla), web browser or command line command (*e.g.*, wget or curl) to download the files, save locally and use this function to import the data into R.

Nomenclature and units from readme.txt:

pre precipitation (millimetres/month)

cv cv of precipitation (percent)

rd0 wet-days (number days with >0.1mm rain per month)
tmp mean temperature (degrees Celsius)
dtr mean diurnal temperature range (degrees Celsius)
reh relative humidity (percent)
sunp sunshine (percent of maximum possible (percent of day length))
frs ground-frost (number of days with ground-frost per month)
wnd 10 metre windspeed (metres/second)
elv elevation (automatically converted to metres)

For more information see the description of the data provided by CRU, <https://crudata.uea.ac.uk/cru/data/hrg/tmc/readme.txt>

Usage

```
create_CRU_stack(  
  pre = FALSE,  
  pre_cv = FALSE,  
  rd0 = FALSE,  
  tmp = FALSE,  
  dtr = FALSE,  
  reh = FALSE,  
  tmn = FALSE,  
  tmx = FALSE,  
  sunp = FALSE,  
  frs = FALSE,  
  wnd = FALSE,  
  elv = FALSE,  
  dsn = ""  
)
```

Arguments

pre	Logical. Fetch precipitation (millimetres/month) from server and return in the data frame? Defaults to FALSE.
pre_cv	Logical. Fetch cv of precipitation (percent) from server and return in the data frame? Defaults to FALSE. NOTE. Setting this to TRUE will always results in pre being set to TRUE and returned as well.
rd0	Logical. Fetch wet-days (number days with >0.1millimetres rain per month) and return in the data frame? Defaults to FALSE.
tmp	Logical. Fetch temperature (degrees Celsius) and return it in the data frame? Defaults to FALSE.
dtr	Logical. Fetch mean diurnal temperature range (degrees Celsius) and return it in the data frame? Defaults to FALSE.
reh	Logical. Fetch relative humidity and return it in the data frame? Defaults to FALSE.

tmn	Logical. Calculate minimum temperature values (degrees Celsius) and return it in the data frame? Defaults to FALSE.
tmx	Logical. Calculate maximum temperature (degrees Celsius) and return it in the data frame? Defaults to FALSE.
sunp	Logical. Fetch sunshine, percent of maximum possible (percent of day length) and return it in data frame? Defaults to FALSE.
frs	Logical. Fetch ground-frost records (number of days with ground- frost per month) and return it in data frame? Defaults to FALSE.
wnd	Logical. Fetch 10m wind speed (metres/second) and return it in the data frame? Defaults to FALSE.
elv	Logical. Fetch elevation (converted to metres) and return it in the data frame? Defaults to FALSE.
dsn	Local file path where CRU CL v.2.0 .dat.gz files are located.

Value

A [list](#) of [raster stack](#) objects of CRU CL v. 2.0 climatology elements

Note

This package automatically converts elevation values from kilometres to metres.

This package crops all spatial outputs to an extent of ymin = -60, ymax = 85, xmin = -180, xmax = 180. Note that the original wind data include land area for parts of Antarctica, these data are excluded in the raster stacks generated by this function.

Author(s)

Adam H. Sparks, <adamhsparks@gmail.com>

See Also

[get_CRU_stack](#)

Examples

```
# Create a raster stack of temperature from tmp
# files in the tempdir() directory.

download.file(
  url = "https://crudata.uea.ac.uk/cru/data/hrg/tmc/grid_10min_tmp.dat.gz",
  destfile = file.path(tempdir(), "grid_10min_tmp.dat.gz")
)

CRU_tmp <- create_CRU_stack(tmp = TRUE, dsn = tempdir())

CRU_tmp
```

`get_CRU_df`*Download and Create a Tidy Data Frame of CRU CL v. 2.0 Climatology Variables*

Description

This function automates downloading and importing CRU CL v. 2.0 climatology data and creates a tidy data frame of the data. If requested, minimum and maximum temperature may also be automatically calculated as described in the data readme.txt file. Data may be cached for later use by this function, saving time downloading files in future use of the function.

Nomenclature and units from readme.txt:

pre precipitation (millimetres/month)

cv cv of precipitation (percent)

rd0 wet-days (number days with >0.1mm rain per month)

tmp mean temperature (degrees Celsius)

dtr mean diurnal temperature range (degrees Celsius)

reh relative humidity (percent)

sunp sunshine (percent of maximum possible (percent of day length))

frs ground-frost (number of days with ground-frost per month)

wnd 10 metre windspeed (metres/second)

elv elevation (automatically converted to metres)

For more information see the description of the data provided by CRU, <https://crudata.uea.ac.uk/cru/data/hrg/tmc/readme.txt>

Usage

```
get_CRU_df(  
  pre = FALSE,  
  pre_cv = FALSE,  
  rd0 = FALSE,  
  tmp = FALSE,  
  dtr = FALSE,  
  reh = FALSE,  
  tmn = FALSE,  
  tmx = FALSE,  
  sunp = FALSE,  
  frs = FALSE,  
  wnd = FALSE,  
  elv = FALSE,  
  cache = FALSE  
)
```

Arguments

pre	Logical. Fetch precipitation (millimetres/month) from server and return in the data frame? Defaults to FALSE.
pre_cv	Logical. Fetch cv of precipitation (percent) from server and return in the data frame? Defaults to FALSE. NOTE. Setting this to TRUE will always results in pre being set to TRUE and returned as well.
rd0	Logical. Fetch wet-days (number days with >0.1millimetres rain per month) and return in the data frame? Defaults to FALSE.
tmp	Logical. Fetch temperature (degrees Celsius) and return it in the data frame? Defaults to FALSE.
dtr	Logical. Fetch mean diurnal temperature range (degrees Celsius) and return it in the data frame? Defaults to FALSE.
reh	Logical. Fetch relative humidity and return it in the data frame? Defaults to FALSE.
tmn	Logical. Calculate minimum temperature values (degrees Celsius) and return it in the data frame? Defaults to FALSE.
tmx	Logical. Calculate maximum temperature (degrees Celsius) and return it in the data frame? Defaults to FALSE.
sunp	Logical. Fetch sunshine, percent of maximum possible (percent of day length) and return it in data frame? Defaults to FALSE.
frs	Logical. Fetch ground-frost records (number of days with ground- frost per month) and return it in data frame? Defaults to FALSE.
wnd	Logical. Fetch 10m wind speed (metres/second) and return it in the data frame? Defaults to FALSE.
elv	Logical. Fetch elevation (converted to metres) and return it in the data frame? Defaults to FALSE.
cache	Logical. Store CRU CL v. 2.0 data files locally for later use? If FALSE, the downloaded files are removed when R session is closed. To take advantage of cached files in future sessions, use cache = TRUE after the initial download and caching. Defaults to FALSE.

Value

A tidy data frame of CRU CL v. 2.0 climatology elements as a [tibble](#) object

Note

This package automatically converts elevation values from kilometres to metres.

Author(s)

Adam H. Sparks, <adamhsparks@gmail.com>

See Also

[create_CRU_stack](#) [manage_cache](#)

Examples

```
# Download data and create a data frame of precipitation and temperature
# without caching the data files
CRU_pre_tmp <- get_CRU_df(pre = TRUE, tmp = TRUE)

head(CRU_pre_tmp)

library(tibble)
CRU_pre_tmp
```

get_CRU_stack	<i>Download and Create a List of Raster Stack Objects From CRU CL v. 2.0 Climatology Variables</i>
---------------	--

Description

This function automates downloading and importing CRU CL v. 2.0 climatology data into R and creates a list of raster stacks of the data. If requested, minimum and maximum temperature may also be automatically calculated as described in the data readme.txt file. Data may be cached for later use by this function, saving time downloading files in future use of the function.

Nomenclature and units from readme.txt:

pre precipitation (millimetres/month)

cv cv of precipitation (percent)

rd0 wet-days (number days with >0.1mm rain per month)

tmp mean temperature (degrees Celsius)

dtr mean diurnal temperature range (degrees Celsius)

reh relative humidity (percent)

sunp sunshine (percent of maximum possible (percent of day length))

frs ground-frost (number of days with ground-frost per month)

wnd 10 metre windspeed (metres/second)

elv elevation (automatically converted to metres)

For more information see the description of the data provided by CRU, <https://crudata.uea.ac.uk/cru/data/hrg/tmc/readme.txt>

Usage

```

get_CRU_stack(
  pre = FALSE,
  pre_cv = FALSE,
  rd0 = FALSE,
  tmp = FALSE,
  dtr = FALSE,
  reh = FALSE,
  tmn = FALSE,
  tmx = FALSE,
  sunp = FALSE,
  frs = FALSE,
  wnd = FALSE,
  elv = FALSE,
  cache = FALSE
)

```

Arguments

pre	Logical. Fetch precipitation (millimetres/month) from server and return in the data frame? Defaults to FALSE.
pre_cv	Logical. Fetch cv of precipitation (percent) from server and return in the data frame? Defaults to FALSE. NOTE. Setting this to TRUE will always results in pre being set to TRUE and returned as well.
rd0	Logical. Fetch wet-days (number days with >0.1millimetres rain per month) and return in the data frame? Defaults to FALSE.
tmp	Logical. Fetch temperature (degrees Celsius) and return it in the data frame? Defaults to FALSE.
dtr	Logical. Fetch mean diurnal temperature range (degrees Celsius) and return it in the data frame? Defaults to FALSE.
reh	Logical. Fetch relative humidity and return it in the data frame? Defaults to FALSE.
tmn	Logical. Calculate minimum temperature values (degrees Celsius) and return it in the data frame? Defaults to FALSE.
tmx	Logical. Calculate maximum temperature (degrees Celsius) and return it in the data frame? Defaults to FALSE.
sunp	Logical. Fetch sunshine, percent of maximum possible (percent of day length) and return it in data frame? Defaults to FALSE.
frs	Logical. Fetch ground-frost records (number of days with ground- frost per month) and return it in data frame? Defaults to FALSE.
wnd	Logical. Fetch 10m wind speed (metres/second) and return it in the data frame? Defaults to FALSE.
elv	Logical. Fetch elevation (converted to metres) and return it in the data frame? Defaults to FALSE.

cache Logical. Store CRU CL v. 2.0 data files locally for later use? If FALSE, the downloaded files are removed when R session is closed. To take advantage of cached files in future sessions, use cache = TRUE after the initial download and caching. Defaults to FALSE.

Value

A list of raster stack objects of CRU CL v. 2.0 climatology elements

Note

This package automatically converts elevation values from kilometres to metres.

This package crops all spatial outputs to an extent of ymin = -60, ymax = 85, xmin = -180, xmax = 180. Note that the original wind data include land area for parts of Antarctica, these data are excluded in the raster stacks generated by this function.

Author(s)

Adam H. Sparks, <adamhsparks@gmail.com>

See Also

[create_CRU_stack](#) [manage_cache](#)

Examples

```
# Download data and create a raster stack of precipitation and temperature
# without caching the data files
CRU_pre_tmp <- get_CRU_stack(pre = TRUE, tmp = TRUE)

CRU_pre_tmp
```

manage_cache

Manage locally cached CRU CL v. 2.0 files

Description

Manage cached **getCRUCLdata** files with **hoardr**

Details

The default cache directory is `file.path(rappdirs::user_cache_dir(), "R/getCRUCLdata")`, but you can set your own path using `manage_cache$cache_path_set()`

`manage_cache$cache_delete` only accepts one file name, while `manage_cache$cache_delete_all` does not accept any names, but deletes all files. For deleting many specific files, use `manage_cache$cache_delete` in an `lapply()` type call.

Useful user functions

- `manage_cache$cache_path_get()` - get cache path
- `manage_cache$cache_path_set()` - set cache path
- `manage_cache$list()` - returns a character vector of full path file names
- `manage_cache$files()` - returns file objects with metadata
- `manage_cache$details()` - returns files with details
- `manage_cache$delete()` - delete specific files
- `manage_cache$delete_all()` - delete all files, returns nothing

Author(s)

Adam H. Sparks, <adamhsparks@gmail.com>

Examples

```
## Not run:  
  
# list files in cache  
manage_cache$list()  
  
# delete certain database files  
manage_cache$delete("file path")  
manage_cache$list()  
  
# delete all files in cache  
manage_cache$delete_all()  
manage_cache$list()  
  
# set a different cache path from the default  
manage_cache$cache_path_set("~/tmp")  
  
## End(Not run)
```

Index

`create_CRU_df`, [2](#)
`create_CRU_stack`, [4](#), [8](#), [11](#)

`get_CRU_df`, [4](#), [7](#)
`get_CRU_stack`, [6](#), [9](#)

`lapply`, [11](#)
`list`, [6](#), [11](#)

`manage_cache`, [8](#), [11](#), [11](#)

`raster`, [6](#), [11](#)

`stack`, [4](#), [6](#), [11](#)

`tibble`, [4](#), [8](#)