

Package ‘crn’

February 19, 2015

Type Package

Title Downloads and Builds datasets for Climate Reference Network

Version 1.1

Date 2011-09-20

Author Steven Mosher

Maintainer Steven Mosher <moshersteven@gmail.com>

Depends R (>= 2.11.0), chron, RCurl

Description The crn package provides the core functions required to download and format data from the Climate Reference Network. Both daily and hourly data are downloaded from the ftp, a consolidated file of all stations is created, station metadata is extracted. In addition functions for selecting individual variables and creating R friendly datasets for them is provided.

License GPL (>= 2)

URL <http://stevemosher.wordpress.com/>

LazyLoad yes

LazyData FALSE

Repository CRAN

Date/Publication 2012-08-02 04:40:52

NeedsCompilation no

R topics documented:

crn-package	2
collateDaily	3
collateHourly	4
colnamesDaily	5
colnamesHourly	5
CRN.DAILY.URL	6
CRN.HOURLY.URL	7

DAILY_DIR	8
downloadCRN	8
HOURLY_DIR	9
HOURS	10
writeDataset	10

Index	12
--------------	-----------

crn-package	<i>Climate Reference Network datatools</i>
-------------	--

Description

The crn package provides the core functions required to download and format data from the Climate Reference Network. Both daily and hourly data are downloaded from the ftp, a consolidated file of all stations is created, station metadata is extracted. In addition functions for selecting individual variables and creating R friendly datasets for them is provided.

Details

Package:	crn
Type:	Package
Version:	1.1
License:	GPL (>=2)
LazyLoad:	yes
LazyData:	FALSE

basic operations

The package provides the basic functions to download and organize CRN data into R friendly datastructures. The first function that should be used is `downloadCRN`, That function downloads either daily or hourly data depending on the variables you pass it. Next, `collateDaily` and `collateHourly` should be run. Finally, `writeDataset` is used to create files with individual variables in them.

Author(s)

Steven Mosher

Maintainer: Steven Mosher <moshersteven@gmail.com>

References

<ftp://ftp.ncdc.noaa.gov/pub/data/uscrn/> <ftp://ftp.ncdc.noaa.gov/pub/data/uscrn/products/>

Examples

```
## Not run:  
downloadCRN()  
collateDaily()  
collateHourly()  
  
## End(Not run)
```

collateDaily	<i>collates individual daily files for multiple stations</i>
--------------	--

Description

CRN daily data is hosted on the ftp as a collection of individual files for every station for every year. Once that data has been downloaded with `downloadCRN`, the individual files can be collated into one monolithic file with this function. In addition, a metadata file is created.

Usage

```
collateDaily(directory = DAILY_DIR)
```

Arguments

directory	The default directory is the directory where the source datafiles are downloaded to by the function <code>downloadCRN</code> . This should not be changed.
-----------	--

Details

The function will examine the directory and read the datafiles in that directory. It will create two output files and write them into your working directory: A data file and a metadata file. A date code is added to the file name of the data file.

Value

The side effect is writing two datafiles for the daily data: a file of data for all the stations and variable. And a metadata file.

Author(s)

Steven Mosher

References

The following urls give the necessary background and variable names. The readme is essential.
<http://www.ncdc.noaa.gov/crn/> <ftp://ftp.ncdc.noaa.gov/pub/data/uscrn/> <ftp://ftp.ncdc.noaa.gov/pub/data/uscrn/products/daily01/README.txt>

collateHourly	<i>collates individual hourly files for multiple stations</i>
---------------	---

Description

CRN daily data is hosted on the ftp as a collection of individual files for every station for every year. Once that data has been downloaded downloadCRN the individual files can be collated into one monolithic file with this function. In addition, a metadata file is created.

Usage

```
collateHourly(directory = HOURLY_DIR)
```

Arguments

directory	The default directory is the directory where the source datafiles are downloaded to by the function downloadCRN. This should not be changed.
-----------	--

Details

The function will examine the directory and read the datafiles in that directory. It will create two output files and write them into your working directory: A data file and a metadata file. A date code is added to the file name of the data file.

Value

The side effect is writing two datafiles for the hourly data: a file of data for all the stations and variable. And a metadata file.

Author(s)

Steven Mosher

References

The following urls give the necessary background and variable names. The readme is essential.
<http://www.ncdc.noaa.gov/crn/> <ftp://ftp.ncdc.noaa.gov/pub/data/uscrn/> <ftp://ftp.ncdc.noaa.gov/pub/data/uscrn/products/hourly02/README.txt>

colnamesDaily	<i>column names for the daily data files</i>
---------------	--

Description

These are the column names that are required for the daily data.

Usage

```
colnamesDaily
```

Format

The format is: chr [1:27] "WBANNO" "LST_DATE" "CRX_VN" " LONGITUDE" "LATITUDE"
"T_DAILY_MAX" ...

Details

The files have columns defined in the readme linked below. They are: "WBANNO,LST_DATE, CRX_VN, LONGITUDE,LATITUDE, T_DAILY_MAX,T_DAILY_MIN,T_DAILY_MEAN,T_DAILY_AVE,P_DAILY_C, P_DAILY_CALC,SOLARAD_DAILY, SUR_TEMP_DAILY_MAX,SUR_TEMP_DAILY_MIN,SUR_TEMP_DAILY_AVG, RH_DAILY_MAX,RH_DAILY_MIN, RH_DAILY_AVE,SOIL_MOISTURE_5_DAILY, SOIL_MOISTURE_10_DAILY,SOIL_MOISTURE_50_DAILY,SOIL_MOISTURE_100_DAILY, SOIL_TEMP_5_DAILY,SOIL_TEMP_10_DAILY,SOIL_TEMP_50_DAILY,SOIL_TEMP_100_DAILY

Source

<ftp://ftp.ncdc.noaa.gov/pub/data/uscrn/products/daily01/README.txt>

Examples

```
print(colnamesDaily)
```

colnamesHourly	<i>column names for the hourly data files</i>
----------------	---

Description

These are the column names that are required for the hourly data.

Usage

```
colnamesHourly
```

Format

The format is: chr [1:37] "WBANNO" "UTC_DATE" "UTC_TIME" "LST_DATE" "LST_TIME"
"CRX_VN" " LONGITUDE" ...

Details

These are the hourly column names. Note these supercede the readme WBANNO,UTC_DATE,UTC_TIME,LST_DATE,LST_DATE,CRX_VN, LONGITUDE,LATITUDE,T_CALC,T_HR_AVG, T_MAX,T_MIN,P_CALC,SOLARAD,SOLARAD_FLAG,SOLARAD_MAX,SOLARAD_MAX_FLAG,SOLARAD_MIN,SOLARAD_MIN_FLAG, SUR_TEMP,SUR_TEMP_FLAG,SUR_TEMP_MIN,SUR_TEMP_MIN_FLAG,RH_HR_AVG,RH_HR_AVG_FLAG, SOIL_MOISTURE_5,SOIL_MOISTURE_10,SOIL_MOISTURE_50,SOIL_MOISTURE_100, SOIL_TEMP_5,SOIL_TEMP_10,SOIL_TEMP_20,SOIL_TEMP_50,SOIL_TEMP_100

Source

<ftp://ftp.ncdc.noaa.gov/pub/data/uscrn/products/hourly02/README.txt>

Examples

```
print(colnamesHourly)
```

CRN.DAILY.URL

URL of the daily data collection

Description

URL to the parent directory

Usage

CRN.DAILY.URL

Format

The format is: chr "ftp://ftp.ncdc.noaa.gov/pub/data/uscrn/products/daily01/"

Details

This is the base url. Subdirectories below this are read and files in those directories are downloaded

Source

<ftp://ftp.ncdc.noaa.gov/pub/data/uscrn/products/daily01/>

References

<ftp://ftp.ncdc.noaa.gov/pub/data/uscrn/products/daily01/README.txt>

Examples

```
print(CRN.DAILY.URL)
```

CRN.HOURLY.URL	<i>URL of the hourly data collection</i>
----------------	--

Description

URL to the parent directory

Usage

```
CRN.HOURLY.URL
```

Format

The format is: `chr "ftp://ftp.ncdc.noaa.gov/pub/data/uscrn/products/hourly02/"`

Details

This is the base url. Subdirectories below this are read and files in those directories are downloaded

Source

<ftp://ftp.ncdc.noaa.gov/pub/data/uscrn/products/hourly02/>

References

<ftp://ftp.ncdc.noaa.gov/pub/data/uscrn/products/hourly02/README.txt>

Examples

```
print(CRN.HOURLY.URL)
```

DAILY_DIR	<i>Directory for downloaded files of daily data</i>
-----------	---

Description

The daily files are downloaded from the various subdirectories on the ftp and they all are downloaded to the DAILY directory. These files are then collated into a monolithic file

Usage

```
DAILY_DIR
```

Format

The format is: chr "DailySourceData"

Examples

```
print(DAILY_DIR)
file.exists(DAILY_DIR)
```

downloadCRN	<i>A function to download all the CRN files</i>
-------------	---

Description

The function is passed a url for daily or hourly data, a matching directory, and sequence of years to download. The function will then download all the files and place them in the right directory.

Usage

```
downloadCRN(url = CRN.DAILY.URL, directory = DAILY_DIR, years = seq(from = 2000, to = 2011, by = 1))
```

Arguments

url	Must be either CRN.DAILY.URL or CRN.HOURLY.URL
directory	Must match the url: DAILY_DIR or HOURLY_DIR
years	a sequence of years to download. On initial download you should download all files. For updates just use the last year

Details

The function iterates through the years in the sequence, fetches all the urls for the files in those subdirectories and downloads them all

Value

Side effect is downloaded files

Author(s)

Steven Mosher

Examples

```
## Not run:  
downloadCRN()  
downloadCRN( url = CRN.HOURLY.URL, directory = HOURLY_DIR )  
  
## End(Not run)
```

HOURLY_DIR

Directory for Hourly data

Description

The hourly files are downloaded from the various subdirectories on the ftp and they all are downloaded to the HOURLY directory. These files are then collated into a monolithic file using `collateHourly`

Usage

```
HOURLY_DIR
```

Format

The format is: `chr "HourlySourceData"`

Examples

```
print(HOURLY_DIR)
```

 HOURS

Collection of hours in a day. A chron object

Description

For use with the hourly data to format the time component according to the specifications of the chron package.

Usage

```
HOURS
```

Format

The format is: Class 'times' atomic [1:24] 0 0.0417 0.0833 0.125 0.1667- attr(*, "format")= chr "h:m:s"

Details

The hours in the hourly files are of a format 0,100,200,..2300 This data object will be used in transforming the time codes of the data into a standard time form

Examples

```
print(HOURS)
```

 writeDataset

Create and save to disk a single variable dataset

Description

The daily and hourly files contain many data elements Typically one works with a single variable at time. This function allows you to create subsets of data that have one and only one variable in the file. The monolithic file, of course, can be read with read.table. This function uses read.table, but skips the columns not requested and writes out the subset of data

Usage

```
writeDataset(filename, cnames = colnamesDaily, varname = "T_DAILY_MEAN")
```

Arguments

filename	Filename of the monolithic file. for example: CRN_Hourly_2011-09-19.dat or CRN_Daily_2011-09-19.dat. The function collateDaily for example will create a file of all stations and all variables and append a date stamp. this function reads that data and selects a single variable from the 35+ variables in the datasets
cnames	The column names either colnamesDaily or colnamesHourly It must match the file you are reading: hourly for hourly; daily for daily
varname	the variable you want to extract. It must be quoted exactly as it appears in the colnames constant

Details

The function masks off all unnecessary columns and writes a dataset. The station ID (WBANNO) and date and time are also saved. one variable and one variable only is added to these base variables and output to the file

Value

Side effect is a file containing all stations and the variable requested.

Note

This only works for numeric variables and not flag data

Author(s)

Steven Mosher

Examples

```
## Not run:  
writeDataset(filename=CRN_Daily_2011-09-19.dat,  
cnames = colnamesDaily, varname = "T_DAILY_MEAN")  
  
## End(Not run)
```

Index

- *Topic **constants**
 - colnamesDaily, 5
 - colnamesHourly, 5
 - HOURS, 10
- *Topic **directories**
 - DAILY_DIR, 8
 - HOURLY_DIR, 9
- *Topic **filetools**
 - collateDaily, 3
 - collateHourly, 4
 - downloadCRN, 8
 - writeDataset, 10
- *Topic **package**
 - crn-package, 2
- *Topic **urls**
 - CRN.DAILY.URL, 6
 - CRN.HOURLY.URL, 7

- collateDaily, 3
- collateHourly, 4
- colnamesDaily, 5
- colnamesHourly, 5
- crn (crn-package), 2
- crn-package, 2
- CRN.DAILY.URL, 6
- CRN.HOURLY.URL, 7

- DAILY_DIR, 8
- downloadCRN, 8

- HOURLY_DIR, 9
- HOURS, 10

- writeDataset, 10