Package 'futureheatwaves'

December 31, 2016

Type Package

Title Find, Characterize, and Explore Extreme Events in Climate Projections

Version 1.0.3

Date 2016-12-30

Description Inputs a directory of climate projection files and, for each, identifies and characterizes heat waves for specified study locations. The definition used to identify heat waves can be customized. Heat wave characterizations include several metrics of heat wave length, intensity, and timing in the year. The heat waves that are identified can be explored using a function to apply user-created functions across all generated heat wave files.This work was supported in part by grants from the National Institute of Environmental Health Sciences (R00ES022631), the National Science Foundation (1331399), and the Colorado State University Vice President for Research.

License GPL-2

LazyData TRUE

Depends R (>= 2.10)

Imports data.table (>= 1.10.0), dplyr (>= 0.4.3), ggplot2 (>= 2.0.0), ggthemes (>= 3.0.0), leaflet (>= 1.0.1), Rcpp (>= 0.12.5), stringr (>= 1.1.0), tidyr (>= 0.6.0)

Suggests gridExtra (>= 2.0.0), knitr (>= 1.10.5), mapproj (>= 1.2.4), maps (>= 3.0.2), rmarkdown (>= 0.7.0), testthat (>= 0.11.0), weathermetrics (>= 1.2.2)

VignetteBuilder knitr

RoxygenNote 5.0.1

LinkingTo Rcpp

NeedsCompilation yes

Author Brooke Anderson [aut, cre], Colin Eason [aut], Elizabeth Barnes [aut]

Maintainer Brooke Anderson <brooke.anderson@colostate.edu>

Repository CRAN

Date/Publication 2016-12-31 08:44:48

R topics documented:

`acquireDirectoryStructure` *Acquire structure of input directory*

Description

This function parses the structure of the user-provided directory of climate projection files to create a list of the climate models and ensemble members included.

Usage

```
acquireDirectoryStructure(dataFolder, coordinateFilenames, tasFilenames,
  timeFilenames, models_to_run, dataDirectories, threshold_ensemble,
  thresholdBoundaries)
```

Arguments

`dataFolder`	Character string with pathway to a directory with climate projection data. This directory must have a specific structure– see the `futureheatwaves` vignette for guidance on setting up this directory.
`coordinateFilenames`	Character string the with filename of each grid point location file. This filename should be identical for all ensemble member subdirectories included in the `dataFolder` directory. See the package vignette for an example of the required structure for this file.
`tasFilenames`	Character string the with filename of each climate projection file. This filename should be identical for all ensemble member subdirectories included in the `dataFolder` directory. See the package vignette for an example of the required structure for this file.
`timeFilenames`	Character string the with filename of each projection dates file. This filename should be identical for all ensemble member subdirectories included in the `dataFolder` directory. See the package vignette for an example of the required structure for this file.
`models_to_run`	A character vector with either "all" (the default), in which case the function runs through all models in the `dataFolder` directory, or the names of the models to run, using the names of each model's subdirectory within the data directory (e.g., `c("bcc1", "ccsm")`).
`dataDirectories`	A list object, with two elements, one for each of the two subdirectories included in the main directory. Typically, these will be separate directories of historical and projection experiments from climate models. Each element of the list should be named with the name of the subdirectory and should provide a numeric vector

with the starting and ending years of the data within each of the two subdirectories (e.g., `list("historical" = c(1990, 1999), "rcp85" = c(2060, 2079))` for a `dataFolder` with historical experiment data for 1990 to 1999 and RCP8.5 projections for 2060 to 2079).

`threshold_ensemble`
: A character vector giving the name of the ensemble member that should be used when determining the city-specific threshold temperatures for each climate model (e.g., `"r1i1p1"`). This threshold is used for relative extreme event definitions. See the `futureheatwaves` vignette for more on extreme event definitions. If any climate model lacks that ensemble member for the specified dates for calculating the threshold, it will be excluded from the processing.

`thresholdBoundaries`
: A numeric vector with the custom time boundaries to be used to determine the threshold variable values for the extreme event definition. The required format for this vector is c(start year, end year), with the restriction that bounds must be contained within the time boundaries of one of the two experiment subdirectories specified by the `dataDirectories` argument in `gen_hw_set`. The default value is 1990 to 1999.

Value

The function returns a list object outlining the file structure of the `dataFolder` directory.

Note

Projection, grid locations, and projection time files within the `dataFolder` directory must be comma-separated (.csv) files and must be named using the names specified by the arguments `coordinateFilenames`, `tasFilenames`, and `timeFilenames`. See the `futureheatwaves` vignette for more information about setting up the `dataFolder` directory.

`apply_all_models` *Apply a function across heat waves from all projections*

Description

This function will take a user-specified function and apply it across all the extreme event files created by `gen_hw_set`. It will generate either a single value for every ensemble member within every climate model, if `city_specific` is set to `FALSE`, or a value for every city for every ensemble member, if `city_specific` is set to `TRUE`.

Usage

```
apply_all_models(out, FUN, city_specific = FALSE, ...)
```

Arguments

out	Character string with pathname to which extreme event files were written by gen_hw_set. Typically, this will be the same pathname as that specified with out when running gen_hw_set.
FUN	A character string with the name of a function to apply to the extreme event dataframes located in the directory specified by out. This function must take the argument, hw_datafr, a dataframe of identified and characterized extreme events, as generated by gen_hw_set. For an example of one of these dataframes, load the hw_datafr data. The function should output a single value when applied to the full extreme event dataframe. The function can also have other arguments, which are passed through the ... argument of apply_all_models.
city_specific	TRUE or FALSE, specifying whether the function should be applied separately for each study location.
...	Optional arguments to FUN.

Value

A dataframe with the value output by the FUN function, as applied to all the extreme event dataframes generated by gen_hw_set.

Note

The function input as FUN must follow a very specific structure. It most have as an argument a dataframe with characterized extreme events, as generated by the gen_hw_set function. See the futureheatwaves vignette for more guidance on creating and applying a custom function to explore the extreme events identified and characterized by gen_hw_set.

Examples

```
ex_results <- system.file("extdata/example_results",
                          package = "futureheatwaves")
apply_all_models(ex_results, FUN = average_mean_temp)
apply_all_models(ex_results, FUN = average_mean_temp,
                 city_specific = TRUE)
```

---

apply_hw_projections *Apply a function to projected extreme events*

Description

This function takes a user-specified function and applies is to a single file of extreme event projections, as specified by hwPath. It will generate either a single value for every ensemble member, if city_specific is set to FALSE, or a value for every city, if city_specific is set to TRUE.

Usage

```
apply_hw_projections(hwPath, FUN, city_specific = FALSE, ...)
```

Arguments

hwPath	A filepath to a comma-separated (.csv) file with a dataset of extreme events and their characteristics, as generated by gen_hw_set. The file at the specified filepath must conform exactly to the format of the extreme event files created by gen_hw_set.
FUN	A character string giving the name of a function to apply to the extreme event dataframe in the file specified by hwPath. This function must only take one argument, hw_datafr, which identifies a dataframe as generated by gen_hw_set. The function should output a single value (e.g., average heat wave length) when applied to the full dataframe.
city_specific	TRUE or FALSE, specifying whether the function should be applied separately for each study location.
...	Optional arguments to FUN.

Note

The function input as FUN must follow a very specific structure. It most have only one argument, and that argument must be a dataframe with extreme events and their characteristics, as generated by the gen_hw_set function. See the futureheatwaves vignette for more guidance on creating and applying a custom function to explore the extreme events identified and characterized by gen_hw_set.

---

average_length *Calculate average length of extreme events*

Description

This function takes a dataframe of extreme events, as created by gen_hw_set, and calculates the average length (in days) of events in the dataframe.

Usage

```
average_length(hw_datafr)
```

Arguments

hw_datafr	A dataframe of extreme events and their characteristics, as created by gen_hw_set

Value

A numeric value with the average length of extreme events in a dataframe of events, as generated by gen_hw_set.

Note

This function is an example of a function that can be created and used to explore extreme events using the apply_all_models function.

Examples

```
data(hw_datafr)
average_length(hw_datafr)
```

`average_mean_temp` *Calculate average variable of extreme events*

Description

This function takes a dataframe of extreme events, as created by gen_hw_set, and calculates the average of the variable (e.g., temperature0 during each extreme event across all events in the dataframe.

Usage

```
average_mean_temp(hw_datafr)
```

Arguments

hw_datafr	A dataframe of extreme events and their characteristics, as created by gen_hw_set

Value

A numeric value with the average variable during all heat waves for a dataset of heat waves, as generated by gen_hw_set. This value is the mean of the average variable within each event, not the average variable across all event days (to calculate that, you would need to calculate a weighted mean, weighted by the number of days in each heat wave).

Note

This function is an example of a function that can be created and used to explore extreme events using the apply_all_models function.

Examples

```
data(hw_datafr)
average_mean_temp(hw_datafr)
```

`buildStructureEnsembles` *List files for a single ensemble member*

Description

This function reads through the user-specified `dataFolder` directory and creates a list with pathnames to all three files (projection times, grid points, and projections) for a single ensemble member.

Usage

```
buildStructureEnsembles(ensemblePath, coordinateFilenames, tasFilenames,
  timeFilenames, dataDirectories)
```

Arguments

`ensemblePath`	A character string that gives the absolute file path for files for an ensemble member within the user-specified projection directory (`dataFolder`).
`coordinateFilenames`	Character string the with filename of each grid point location file. This filename should be identical for all ensemble member subdirectories included in the `dataFolder` directory. See the package vignette for an example of the required structure for this file.
`tasFilenames`	Character string the with filename of each climate projection file. This filename should be identical for all ensemble member subdirectories included in the `dataFolder` directory. See the package vignette for an example of the required structure for this file.
`timeFilenames`	Character string the with filename of each projection dates file. This filename should be identical for all ensemble member subdirectories included in the `dataFolder` directory. See the package vignette for an example of the required structure for this file.
`dataDirectories`	A list object, with two elements, one for each of the two subdirectories included in the main directory. Typically, these will be separate directories of historical and projection experiments from climate models. Each element of the list should be named with the name of the subdirectory and should provide a numeric vector with the starting and ending years of the data within each of the two subdirectories (e.g., `list("historical" = c(1990, 1999), "rcp85" = c(2060, 2079))` for a `dataFolder` with historical experiment data for 1990 to 1999 and RCP8.5 projections for 2060 to 2079).

Value

A list of length 2. The first element is the name of the ensemble that was processed. The second element is a vector with the filepaths of the three files.

buildStructureExperiments *Generate file structure for an experiment*

Description

This function generates a list object with the file structure of files in the dataFolder directory for a single experiment (e.g., "historical" or "rcp85").

Usage

```
buildStructureExperiments(modelName, experiment, dataPath, coordinateFilenames,
  tasFilenames, timeFilenames, dataDirectories)
```

Arguments

modelName	Character string of climate model name (e.g., "bcc1"). This name is generated from the subdirectory name for the climate model within dataFolder.
experiment	Character string of the experiment. Possible variables are the names of elements in the list object specified by the dataDirectories argument in gen_hw_set.
dataPath	Character string of the file path to dataFolder. Must include the final '/'.
coordinateFilenames	Character string the with filename of each grid point location file. This filename should be identical for all ensemble member subdirectories included in the dataFolder directory. See the package vignette for an example of the required structure for this file.
tasFilenames	Character string the with filename of each climate projection file. This filename should be identical for all ensemble member subdirectories included in the dataFolder directory. See the package vignette for an example of the required structure for this file.
timeFilenames	Character string the with filename of each projection dates file. This filename should be identical for all ensemble member subdirectories included in the dataFolder directory. See the package vignette for an example of the required structure for this file.
dataDirectories	A list object, with two elements, one for each of the two subdirectories included in the main directory. Typically, these will be separate directories of historical and projection experiments from climate models. Each element of the list should be named with the name of the subdirectory and should provide a numeric vector with the starting and ending years of the data within each of the two subdirectories (e.g., list("historical" = c(1990, 1999), "rcp85" = c(2060, 2079)) for a dataFolder with historical experiment data for 1990 to 1999 and RCP8.5 projections for 2060 to 2079).

Value

A list with one element for each ensemble member. Each element is a return value of the buildStructureEnsembles function for one ensemble member in the experiment and climate model.

---

buildStructureModels *Generate list of file structure*

Description

This function takes input from acquireDirectoryStructure and uses it to generate a list object with the projection directory file structure. This parsed file structure is later used to lead other code through all climate models and ensemble members in the input projection directory.

Usage

```
buildStructureModels(modelName, experiments, dataFolder, coordinateFilenames,
  tasFilenames, timeFilenames, dataDirectories)
```

Arguments

modelName	Character string of climate model name (e.g., "bcc1"). This name is generated from the subdirectory name for the climate model within dataFolder.
experiments	Character string of the experiment(s). Possible variables are the names of elements in the list object specified by the dataDirectories argument in gen_hw_set.
dataFolder	Character string with pathway to a directory with climate projection data. This directory must have a specific structure– see the futureheatwaves vignette for guidance on setting up this directory.
coordinateFilenames	Character string the with filename of each grid point location file. This filename should be identical for all ensemble member subdirectories included in the dataFolder directory. See the package vignette for an example of the required structure for this file.
tasFilenames	Character string the with filename of each climate projection file. This filename should be identical for all ensemble member subdirectories included in the dataFolder directory. See the package vignette for an example of the required structure for this file.
timeFilenames	Character string the with filename of each projection dates file. This filename should be identical for all ensemble member subdirectories included in the dataFolder directory. See the package vignette for an example of the required structure for this file.
dataDirectories	A list object, with two elements, one for each of the two subdirectories included in the main directory. Typically, these will be separate directories of historical and projection experiments from climate models. Each element of the list should be named with the name of the subdirectory and should provide a numeric vector

with the starting and ending years of the data within each of the two subdirectories (e.g., `list("historical" = c(1990, 1999), "rcp85" = c(2060, 2079))` for a `dataFolder` with historical experiment data for 1990 to 1999 and RCP8.5 projections for 2060 to 2079).

Value

A list of length 3. The first element is the name of the model whose structure was being built. The second element is, for this climate model, the hierarchy of the first subdirectory specified by `dataDirectories`. The third element is the hierarchy of the second subdirectory specified by `dataDirectories`. The second and third elements are return values of `buildStructureExperiments`.

`checkCustomBounds` *Check year boundaries for errors*

Description

This function inputs the boundary lists specified in `gen_hw_set`, `thresholdBoundaries`, `projectionBoundaries`, and `referenceBoundaries`, and checks them for errors in structure of the input or in the years selected.

Usage

```
checkCustomBounds(boundList, dataDirectories)
```

Arguments

`boundList`	A set of boundary years in the format c(start year, end year).
`dataDirectories`	A list object, with two elements, one for each of the two subdirectories included in the main directory. Typically, these will be separate directories of historical and projection experiments from climate models. Each element of the list should be named with the name of the subdirectory and should provide a numeric vector with the starting and ending years of the data within each of the two subdirectories (e.g., `list("historical" = c(1990, 1999), "rcp85" = c(2060, 2079))` for a `dataFolder` with historical experiment data for 1990 to 1999 and RCP8.5 projections for 2060 to 2079).

check_params *Check for input parameter errors*

Description

This function goes through all parameter inputs for gen_hw_set and makes sure all parameter entries are in the appropriate format. If any parameters are in an incorrect format, the function stops and returns an error describing the problem.

Usage

```
check_params(out, dataFolder, dataDirectories, citycsv, coordinateFilenames,
  tasFilenames, timeFilenames, IDheatwavesFunction, thresholdBoundaries,
  projectionBoundaries, referenceBoundaries, numDays, seasonal_months,
  absolute_thresholds)
```

Arguments

out	Character string with pathway to directory to which extreme event files will be written. This should be a pathname to a directory on the user's local computer. If the directory already exists, it will be overwritten by this function, so the user should either specify a pathname for a directory that does not yet exist or be willing to overwrite the existing directory. The parent directory of the specified directory must exist.
dataFolder	Character string with pathway to a directory with climate projection data. This directory must have a specific structure– see the futureheatwaves vignette for guidance on setting up this directory.
dataDirectories	A list object, with two elements, one for each of the two subdirectories included in the main directory. Typically, these will be separate directories of historical and projection experiments from climate models. Each element of the list should be named with the name of the subdirectory and should provide a numeric vector with the starting and ending years of the data within each of the two subdirectories (e.g., list("historical" = c(1990, 1999), "rcp85" = c(2060, 2079)) for a dataFolder with historical experiment data for 1990 to 1999 and RCP8.5 projections for 2060 to 2079).
citycsv	Character string giving the filepath to a comma-separated (.csv) file with, for each study city, a unique city identifier, latitude, and longitude. These values must be specified with the column names city, lat, and lon. See the futureheatwaves vignette for guidance on setting up this file.
coordinateFilenames	Character string the with filename of each grid point location file. This filename should be identical for all ensemble member subdirectories included in the dataFolder directory. See the package vignette for an example of the required structure for this file.

tasFilenames
: Character string the with filename of each climate projection file. This filename should be identical for all ensemble member subdirectories included in the dataFolder directory. See the package vignette for an example of the required structure for this file.

timeFilenames
: Character string the with filename of each projection dates file. This filename should be identical for all ensemble member subdirectories included in the dataFolder directory. See the package vignette for an example of the required structure for this file.

IDheatwavesFunction
: A character string with the name of the R function to use to identify extreme events. This function may be a user-specified custom function, but it must be loaded into the current R session. The function name must be put in quotation marks. For more guidance on how to write and use a custom function to identify extreme events, see the package vignette for futureheatwaves.

thresholdBoundaries
: A numeric vector with the custom time boundaries to be used to determine the threshold variable values for the extreme event definition. The required format for this vector is c(start year, end year), with the restriction that bounds must be contained within the time boundaries of one of the two experiment subdirectories specified by the dataDirectories argument in gen_hw_set. The default value is 1990 to 1999.

projectionBoundaries
: A numeric vector with the custom time boundaries for which the user wants to create extreme event projections. The required format for this vector is c(start year, end year), with the restriction that bounds must be contained within the time boundaries of one of the two experiment subdirectories specified by the dataDirectories argument in gen_hw_set. The default value is 2070 to 2079.

referenceBoundaries
: A numeric vector with the custom time boundaries to use in calculating relative characteristics for extreme events (i.e., to use when exploring the role of adaptation in projections). For more information on how reference temperatures are used, see the package vignette for futureheatwaves. The required format for this vector is c(start year, end year), with the restriction that bounds must be contained within the time boundaries of one of the two experiment subdirectories specified by the dataDirectories argument in gen_hw_set. The default value is 2070 to 2079. If the time bounds used differ from those used for projections, these reference variables will be pulled from the ensemble member for each climate model specified by threshold_ensemble.

numDays
: Integer greater than 0 giving the number of days to use in the extreme event definition (e.g., numDays = 2 would define a heat wave as two or more days above the threshold temperature).

seasonal_months
: A numeric vector giving the months to use to calculate the "mean.seasonal.var" column in the extreme event characteristics output file. For example, if seasonal_months is set to 6:8, daily temperatures from June through August each day during the reference years would be used to calculate this mean seasonal value. The default is 5:9 (May–September).

absolute_thresholds
: A numeric vector with four values giving the absolute thresholds to use when calculating the "days.above.abs.thershold" columns in the heatwave characteristic output files. These values must be given in the units in which temperature is expressed in the input data (typically Kelvin for climate model output data). The default values are the values in Kelvin corresponding to 80, 85, 90, and 95 degrees Fahrenheit. If this parameter is customized, it must include four values.

Value

Only stops and returns an error if any parameters are incorrect.

Note

This function does not check if the data is organized in the proper structure or if any data exists within the directory at all, so a call to gen_hw_set could still pass through this check and make it further through the function code with those mistakes.This function also does not check if the three ensemble final .csv data files exist, only if they have the .csv extension if they do exist.

---

closest_point *Find closest grid point to a city location*

Description

This function identifies the closest grid points in a climate model to a city based on the city's latitude and longitude using the Euclidean distance.

Usage

```
closest_point(city, latlong)
```

Arguments

city
: A numeric vector containing the city's ID, latitude, and longitude, in that order.

latlong
: A dataframe of latitude and longitude of each climate grid cell, with latitudes in the first column and longitudes in the second column. The readLatLong function re-orders the columns of longitude and latitude to comply with this format if necessary.

Value

An index corresponding to the column in the climate projections dataframe for that climate model that is closest to the city's coordinates.

consolidate *Consolidate heat wave dataframes*

Description

This function combines all identified city-specific heat wave dataframes together into a single dataframe. This function is used to create a single dataframe with all heat waves from all study cities for an ensemble member.

Usage

```
consolidate(hwDataframeList)
```

Arguments

hwDataframeList	A list object where each element is the dataframe of heat waves, created by the closure created by `createCityProcessor`, for a single city.

Value

A combined dataframe version of the list object that was passed as an argument.

createAccumulators *Create accumulator closure*

Description

This function creates a closure that holds, adds to, and returns data structures that the user wishes to grow at various points in the execution of the package (e.g., location and model information dataframes).

Usage

```
createAccumulators()
```

Details

As an example, when the generated closure is used with the command "append location list", it will add information on the cities and closest grid point locations based on the climate model it has just completed analyzing to a growing dataframe with this information for all climate models. After the function run to generate the heat wave projections is completed, this closure can be used with the command "return locations" to output the completed dataframe of this location information.The closure can be used in a similar manner to aggregate and then return meta-data on the models analyzed based on their inclusion in the user-specified projections directory.

Value

A closure that accepts commands to access and append new data onto data structures as the program executes. The closure created by this function accepts two arguments: (1) the command and (2) an element to be appended to the end of the data structure of the command. These two arguments must be entered in this exact order. The first argument (the command) can be one of the following options: `return model information`, `append model information`, `return locations`, and `append location list`. The second argument for the created closure should only be used in conjunction with the two "append" commands for the closure.

`createCityProcessor` *Create closure to identify and aggregate heat waves*

Description

This function creates a closure that returns a dataframe with the identified heat waves and heat wave characteristics for a given city for the specified projection period, as generated by the `createHwDataframe` function.

Usage

```
createCityProcessor(global)
```

Arguments

`global`	An list object created by `gen_hw_set` that includes user specifications (e.g., the path to the output directory, the path to the input climate projections, the dataframe with city locations).

Value

This function creates a closure that takes inputs of `threshold`, `ensembleSeries`, and `custom` and will find and characterize all heat waves in all cities for a given ensemble. See the help file for `formHwFrame` for more information on the format of the dataframe created by this closure.

Note

The closure encapsulates an incrementer varaible and advances it with every call. This variable is used to index into the `cities` vector from the `global` object passed into this function.

createEnsembleWriter *Ensemble writer factory function*

Description

This function creates a closure that writes a single heat wave list to a comma-separated file in the directory specified by the user in the out argument of gen_hw_set.

Usage

```
createEnsembleWriter(modelName, global, custom)
```

Arguments

modelName	Character string of climate model name (e.g., "bcc1"). This name is generated from the subdirectory name for the climate model within dataFolder.
global	An list object created by gen_hw_set that includes user specifications (e.g., the path to the output directory, the path to the input climate projections, the dataframe with city locations).
custom	An list object created by gen_hw_set that includes user specifications (e.g., the name of the R function to use to identify heat waves, alternative upper and lower year boundaries for the data used to determine threshold temperatures for the heat wave definition, alternative upper and lower year boundaries for the projection period of the heat wave datasets being generated).

Details

The closure created by this function closes over an incrementer variable for ensembles that advances each time the closure is called.

Value

A closure that inputs hwFrame, a combined heat wave dataframe with all heat wave information for all cities for the ensemble and writes out a heat wave dataframe to the output directory specified by the out argument in gen_hw_set.

createHwDataframe *Characterize heat waves*

Description

This function takes a dataframe with identified heat waves and returns a dataframe that lists and characterizes all of the heat waves. If no heat waves were identified in a city, it returns a data frame with the same columns but no observations, to allow the empty dataframe to be joined without error to the dataframes for cities that do have heat waves under the definition.

Usage

```
createHwDataframe(city, threshold, heatwaves, ensembleSeries, i, global, custom)
```

Arguments

city	A character vector with the identification of the city being processed.
threshold	Numeric string with threshold temperature used in the heat wave definition, in degrees Fahrenheit.
heatwaves	A dataframe with the following columns: • date: Date of each observation, in class "Date"; • tmpd: Temperature in degrees Fahrenheit; • hw: A binary variable designating whether a day is in a heat wave (0: not in a heat wave; 1: in a heat wave); and • hw.number: A numeric value, 0 if the day was not part of a heat wave, otherwise the number of the heat wave to which the day belonged. This is the format of the output of IDheatwaves.
ensembleSeries	A list object giving the projection time series as well as a variety of other information for a single ensemble member. This is the output of processEnsemble.
i	An index specifying which city is being processed. This corresponds to the order of the cities in the citycsv file specified in gen_hw_set.
global	An list object created by gen_hw_set that includes user specifications (e.g., the path to the output directory, the path to the input climate projections, the dataframe with city locations).
custom	An list object created by gen_hw_set that includes user specifications (e.g., the name of the R function to use to identify heat waves, alternative upper and lower year boundaries for the data used to determine threshold temperatures for the heat wave definition, alternative upper and lower year boundaries for the projection period of the heat wave datasets being generated).

Value

A dataframe of identified and characterized heat waves for a single city and single ensemble member. Each row of this dataframe represents a heat wave, with the following columns:

- hw.number: A sequential number identifying each heat wave in a city;
- mean.temp: Average daily temperature across all days in the heat wave, in degrees Fahrenheit;
- max.temp: Highest daily temperature across days in the heat wave, in degrees Fahrenheit;
- min.temp: Lowest daily temperature across days in the heat wave, in degrees Fahrenheit
- length: Number of days in the heat wave;
- start.date: Date of the first day of the heat wave;
- end.date: Date of the last day of the heat wave;
- start.doy: Day of the year of the first day of the heat wave (1 = Jan. 1, etc.);
- start.month: Month in which the heat wave started (1 = January, etc.);

- days.above.80: Number of days in the heat wave above 80 degrees Fahrenheit;
- days.above.85: Number of days in the heat wave above 85 degrees Fahrenheit;
- days.above.90: Number of days in the heat wave above 90 degrees Fahrenheit;
- days.above.95: Number of days in the heat wave above 90 degrees Fahrenheit;
- days.above.99th: Number of days in the heat wave above the 99th percentile temperature for the city, using the period specified by the user with the `referenceBoundaries` argument in `gen_hw_set` as a reference for determining these percentiles;
- days.above.99.5th: Number of days in the heat wave above the 99.5th percentile temperature for the city, using the period specified by the user with the `referenceBoundaries` argument in `gen_hw_set` as a reference for determining these percentiles;
- first.in.season: Whether the heat wave was the first to occur in its calendar year (Note: this characteristic is likely not useful in southern hemisphere studies.);
- threshold.temp: The temperature used as the threshold for the heat wave definition in the city;
- mean.temp.quantile: The percentile of the average daily mean temperature during the heat wave compared to the city's year-round temperature distribution, based on the temperatures for the city during the period specified by the `referenceBoundaries` argument in `gen_hw_set`;
- max.temp.quantile: The percentile of the highest daily mean temperature during the heat wave compared to the city's year-round temperature distribution;
- min.temp.quantile: The percentile of the lowest daily mean temperature during the heat wave compared to the city's year-round temperature distribution;
- mean.temp.1: The city's average year-round temperature, based on the temperatures for the city during the period specified by the `referenceBoundaries` argument in `gen_hw_set`;
- mean.summer.temp: The city's average May–September temperature, based on the temperatures for the city during the period specified by the `referenceBoundaries` argument in `gen_hw_set`; and
- city: The identifier for the city, as given in the file specified in the `citycsv` argument of `gen_hw_set`.

Note

When calculating relative characteristics of heat waves, like the relative value of the heat wave's mean temperature, this function uses a time series from the date ranges specified by the user using the `referenceBoundaries` option in `gen_hw_set`. By default, these references are based on projection data from 2070 to 2079.

`datafr` *Example data to input to extreme event identifier functions*

Description

This dataset provides an example of the dataframe to be input into a custom extreme event identification function. This dataset can be used to test a custom extreme event identification function before using the function in `gen_hw_set`.

Usage

```
datafr
```

Format

A dataframe with 7,300 rows and 2 columns. The variables are:

- date: A Date vector with the date of the projection
- tmpd: A numeric vector with the temperature of the projection, in degrees Fahrenheit

---

formDates *Create date vector for requested time period*

Description

This function creates a vector that includes all the dates spanning the requested date range.

Usage

```
formDates(times, bounds)
```

Arguments

times	A dataframe containing the time data for one ensemble member's projections. Each row in this dataframe corresponds to the row with the same row number in the climate projection dataframe.
bounds	A numeric vector with the desired time boundaries of the ensemble, in terms of indices referring to the first and last row in the climate projection data frame within the requested date range.

Value

A vector of the desired time boundaries, formatted as year-month-day.

formHwFrame *Create heat wave dataframe for an ensemble*

Description

This function takes inputs, from processProjections, on the projection data for an ensemble member, the thresholds for cities for the ensemble, and global and custom objects with user specifications. Using these inputs, the function creates a dataframe with heat waves identified and characterized for the ensemble member.

Usage

```
formHwFrame(ensembleSeries, thresholds, global, custom)
```

Arguments

ensembleSeries	A list object giving the projection time series as well as a variety of other information for a single ensemble member. This is the output of processEnsemble.
thresholds	A vector with the thresholds to use within each city in that city's heat wave definition.These are typically automatically determine during the run of the gen_hw_set function.
global	An list object created by gen_hw_set that includes user specifications (e.g., the path to the output directory, the path to the input climate projections, the dataframe with city locations).
custom	An list object created by gen_hw_set that includes user specifications (e.g., the name of the R function to use to identify heat waves, alternative upper and lower year boundaries for the data used to determine threshold temperatures for the heat wave definition, alternative upper and lower year boundaries for the projection period of the heat wave datasets being generated).

Value

The combined dataframe of identified and characterized heat waves for selected projection date range for all cities specified by the user. This dataframe includes the following columns:

- hw.number: A sequential number identifying each heat wave in a city;
- mean.temp: Average daily temperature across all days in the heat wave, in degrees Fahrenheit;
- max.temp: Highest daily temperature across days in the heat wave, in degrees Fahrenheit;
- min.temp: Lowest daily temperature across days in the heat wave, in degrees Fahrenheit
- length: Number of days in the heat wave;
- start.date: Date of the first day of the heat wave;
- end.date: Date of the last day of the heat wave;
- start.doy: Day of the year of the first day of the heat wave (1 = Jan. 1, etc.);
- start.month: Month in which the heat wave started (1 = January, etc.);

- days.above.80: Number of days in the heat wave above 80 degrees Fahrenheit;
- days.above.85: Number of days in the heat wave above 85 degrees Fahrenheit;
- days.above.90: Number of days in the heat wave above 90 degrees Fahrenheit;
- days.above.95: Number of days in the heat wave above 90 degrees Fahrenheit;
- days.above.99th: Number of days in the heat wave above the 99th percentile temperature for the city, using the period specified by the user with the `referenceBoundaries` argument in `gen_hw_set` as a reference for determining these percentiles;
- days.above.99.5th: Number of days in the heat wave above the 99.5th percentile temperature for the city, using the period specified by the user with the `referenceBoundaries` argument in `gen_hw_set` as a reference for determining these percentiles;
- first.in.season: Whether the heat wave was the first to occur in its calendar year (Note: this characteristic is likely not useful in southern hemisphere studies.);
- threshold.temp: The temperature used as the threshold for the heat wave definition in the city;
- mean.temp.quantile: The percentile of the average daily mean temperature during the heat wave compared to the city's year-round temperature distribution, based on the temperatures for the city during the period specified by the `referenceBoundaries` argument in `gen_hw_set`;
- max.temp.quantile: The percentile of the highest daily mean temperature during the heat wave compared to the city's year-round temperature distribution;
- min.temp.quantile: The percentile of the lowest daily mean temperature during the heat wave compared to the city's year-round temperature distribution;
- mean.temp.1: The city's average year-round temperature, based on the temperatures for the city during the period specified by the `referenceBoundaries` argument in `gen_hw_set`;
- mean.summer.temp: The city's average May–September temperature, based on the temperatures for the city during the period specified by the `referenceBoundaries` argument in `gen_hw_set`; and
- city: The identifier for the city, as given in the file specified in the `citycsv` argument of `gen_hw_set`.

An example of the output of this function is available as the `hw_datafr` dataset and can be accessed using `data(hw_datafr)`.

`futureheatwaves` *Find, Characterize, and Explore Heat Waves in Climate Projections*

Description

`futureheatwaves` takes a directory of climate projection files and, for each, identifies and characterizes all examples of a specified type of extreme event. The definition used to identify extreme events can be customized. Characterizations include several metrics of event length, intensity, and timing in the year. Extreme events can be explored by applying custom functions across all generated heat wave files. This work was supported in part by grants from the National Institute of Environmental Health Sciences (R00ES022631), the National Science Foundation (1331399), and the Colorado State University Vice President for Research.

gen_hw_set *Create and write extreme event projections*

Description

This function creates datasets of identified and characterized extreme events for all ensemble members in all climate models in a directory of climate projections for a user-specified set of locations.The resulting extreme event projections are written out to a specified directory on the user's local computer. For more details on downloading CMIP5 climate model output data and setting it up to use with this function, see this package's "starting_from_netcdf" vignette.

Usage

```
gen_hw_set(out, dataFolder, dataDirectories = list(historical = c(1980, 2004),
  rcp85 = c(2006, 2099)), citycsv, coordinateFilenames, tasFilenames,
  timeFilenames, IDheatwavesFunction = "IDHeatwavesCPPwrapper",
  thresholdBoundaries = c(1990, 1999), projectionBoundaries = c(2070, 2079),
  referenceBoundaries = c(2070, 2079), models_to_run = "all",
  probThreshold = 0.98, numDays = 2, printWarning = TRUE,
  threshold_ensemble = "r1i1p1", lat_lon_colnames = c("lat", "lon"),
  above_threshold = TRUE, absolute_thresholds = c(299.82, 302.6, 305.37,
  308.15), seasonal_months = c(5:9))
```

Arguments

`out`	Character string with pathway to directory to which extreme event files will be written. This should be a pathname to a directory on the user's local computer. If the directory already exists, it will be overwritten by this function, so the user should either specify a pathname for a directory that does not yet exist or be willing to overwrite the existing directory. The parent directory of the specified directory must exist.
`dataFolder`	Character string with pathway to a directory with climate projection data. This directory must have a specific structure– see the `futureheatwaves` vignette for guidance on setting up this directory.
`dataDirectories`	A list object, with two elements, one for each of the two subdirectories included in the main directory. Typically, these will be separate directories of historical and projection experiments from climate models. Each element of the list should be named with the name of the subdirectory and should provide a numeric vector with the starting and ending years of the data within each of the two subdirectories (e.g., `list("historical" = c(1990, 1999), "rcp85" = c(2060, 2079))` for a `dataFolder` with historical experiment data for 1990 to 1999 and RCP8.5 projections for 2060 to 2079).
`citycsv`	Character string giving the filepath to a comma-separated (.csv) file with, for each study city, a unique city identifier, latitude, and longitude. These values must be specified with the column names `city`, `lat`, and `lon`. See the `futureheatwaves` vignette for guidance on setting up this file.

coordinateFilenames
: Character string the with filename of each grid point location file. This filename should be identical for all ensemble member subdirectories included in the dataFolder directory. See the package vignette for an example of the required structure for this file.

tasFilenames
: Character string the with filename of each climate projection file. This filename should be identical for all ensemble member subdirectories included in the dataFolder directory. See the package vignette for an example of the required structure for this file.

timeFilenames
: Character string the with filename of each projection dates file. This filename should be identical for all ensemble member subdirectories included in the dataFolder directory. See the package vignette for an example of the required structure for this file.

IDheatwavesFunction
: A character string with the name of the R function to use to identify extreme events. This function may be a user-specified custom function, but it must be loaded into the current R session. The function name must be put in quotation marks. For more guidance on how to write and use a custom function to identify extreme events, see the package vignette for futureheatwaves.

thresholdBoundaries
: A numeric vector with the custom time boundaries to be used to determine the threshold variable values for the extreme event definition. The required format for this vector is c(start year, end year), with the restriction that bounds must be contained within the time boundaries of one of the two experiment subdirectories specified by the dataDirectories argument in gen_hw_set. The default value is 1990 to 1999.

projectionBoundaries
: A numeric vector with the custom time boundaries for which the user wants to create extreme event projections. The required format for this vector is c(start year, end year), with the restriction that bounds must be contained within the time boundaries of one of the two experiment subdirectories specified by the dataDirectories argument in gen_hw_set. The default value is 2070 to 2079.

referenceBoundaries
: A numeric vector with the custom time boundaries to use in calculating relative characteristics for extreme events (i.e., to use when exploring the role of adaptation in projections). For more information on how reference temperatures are used, see the package vignette for futureheatwaves. The required format for this vector is c(start year, end year), with the restriction that bounds must be contained within the time boundaries of one of the two experiment subdirectories specified by the dataDirectories argument in gen_hw_set. The default value is 2070 to 2079. If the time bounds used differ from those used for projections, these reference variables will be pulled from the ensemble member for each climate model specified by threshold_ensemble.

models_to_run
: A character vector with either "all" (the default), in which case the function runs through all models in the dataFolder directory, or the names of the models to run, using the names of each model's subdirectory within the data directory (e.g., c("bcc1", "ccsm")).

probThreshold Numerical value between 0 and 1 specifying the percentile to be used for the threshold variable used to define extreme events. The default value is 0.98 (i.e., a heat wave is a certain number of days above the city's 98th percentile temperature).

numDays Integer greater than 0 giving the number of days to use in the extreme event definition (e.g., `numDays = 2` would define a heat wave as two or more days above the threshold temperature).

printWarning TRUE / FALSE, specifies whether to print out a warning informing the user that the function will write out results to the local directory specified by the user with `out`. This warning prints out by default; the user must opt out of this warning by specifying FALSE for this argument, for example if running this function within a loop.

threshold_ensemble
A character vector giving the name of the ensemble member that should be used when determining the city-specific threshold temperatures for each climate model (e.g., `"r1i1p1"`). This threshold is used for relative extreme event definitions. See the `futureheatwaves` vignette for more on extreme event definitions. If any climate model lacks that ensemble member for the specified dates for calculating the threshold, it will be excluded from the processing.

lat_lon_colnames
A character vector of length two with the column names in the `citycsv` dataframe for latitude (first vector element) and longitude (second vector element)

above_threshold
A logical value specifying whether events should be identified by finding days above a threshold (`TRUE`, the default, e.g., for finding heat waves or extreme air pollution events) or below a threshold (`FALSE`, e.g., for finding cold waves or droughts).

absolute_thresholds
A numeric vector with four values giving the absolute thresholds to use when calculating the "days.above.abs.thershold" columns in the heatwave characteristic output files. These values must be given in the units in which temperature is expressed in the input data (typically Kelvin for climate model output data). The default values are the values in Kelvin corresponding to 80, 85, 90, and 95 degrees Fahrenheit. If this parameter is customized, it must include four values.

seasonal_months
A numeric vector giving the months to use to calculate the "mean.seasonal.var" column in the extreme event characteristics output file. For example, if `seasonal_months` is set to `6:8`, daily temperatures from June through August each day during the reference years would be used to calculate this mean seasonal value. The default is `5:9` (May–September).

Value

This function creates, and writes to the user's computer, files with the extreme events and their characteristics for the specified climate projections and dates. For more information on customizing this function, see the `futureheatwaves` vignette. This function also returns a dataframe listing the name of each climate model processed, as well as the number of historical and future projection

ensemble members for each model. This output can be used as a check that the function processed through the directory of input files specified using the dataFolder argument.

Examples

```
## Not run:
projection_dir_location <- system.file("extdata/cmip5",
                                       package = "futureheatwaves")
city_file_location <- system.file("extdata/cities.csv",
                                  package = "futureheatwaves")
gen_hw_set(out = "example_results",
           dataFolder = projection_dir_location ,
           dataDirectories = list("historical" = c(1990, 1999),
                                  "rcp85" = c(2060, 2079)),
           citycsv = city_file_location,
           coordinateFilenames = "latitude_longitude_NorthAmerica_12mo.csv",
           tasFilenames = "tas_NorthAmerica_12mo.csv",
           timeFilenames = "time_NorthAmerica_12mo.csv")

## End(Not run)
```

---

getBounds *Acquire boundaries of time series data*

Description

This function acquires the boundaries of time series data.

Usage

```
getBounds(times, custom, type)
```

Arguments

times	A dataframe containing the time data for one ensemble member's projections. Each row in this dataframe corresponds to the row with the same row number in the climate projection dataframe.
custom	An list object created by gen_hw_set that includes user specifications (e.g., the name of the R function to use to identify heat waves, alternative upper and lower year boundaries for the data used to determine threshold temperatures for the heat wave definition, alternative upper and lower year boundaries for the projection period of the heat wave datasets being generated).
type	A character vector giving the type of boundaries that are being checked. Possible values are "threshold", "projections", and "reference".

Value

A numeric vector containing the number of days spanned by the time period, the lower bound of the experiment time period as an index specifying the relevant row of the climate projection data; and an upper bound of the experiment time period as an index specifying the relevant row of the climate projection data.

`heatwave_days` *Calculate total event days*

Description

This function takes a dataframe of heat waves, as created by `gen_hw_set`, and calculates the total number of extreme event days in the dataframe (sum of the number of days in each event for all events in the dataset).

Usage

```
heatwave_days(hw_datafr)
```

Arguments

`hw_datafr`	A dataframe of extreme events and their characteristics, as created by `gen_hw_set`

Value

A numeric value with the number of extreme event days in a dataframe of heat waves, as generated by `gen_hw_set`.

Note

This function is an example of a function that can be created and used to explore extreme events using the `apply_all_models` function.

Examples

```
data(hw_datafr)
heatwave_days(hw_datafr)
```

hw_datafr *Example of extreme event characteristics dataset*

Description

This dataset provides an example of the structure of the dataframe of extreme events and their characteristics, as created by gen_hw_set. This example dataset can be used in developing custom functions to use with apply_all_models.

Usage

```
hw_datafr
```

Format

A dataframe with 258 rows and 23 columns. The variables are:

- hw.number: A sequential number identifying each heat wave in a city;
- mean.var: Average daily temperature across all days in the heat wave, in degrees Fahrenheit;
- max.var: Highest daily temperature across days in the heat wave, in degrees Fahrenheit;
- min.var: Lowest daily temperature across days in the heat wave, in degrees Fahrenheit
- length: Number of days in the heat wave;
- start.date: Date of the first day of the heat wave;
- end.date: Date of the last day of the heat wave;
- start.doy: Day of the year of the first day of the heat wave (1 = Jan. 1, etc.);
- start.month: Month in which the heat wave started (1 = January, etc.);
- days.above.abs.threshold.1: Number of days in the heat wave above 80 degrees Fahrenheit;
- days.above.abs.threshold.2: Number of days in the heat wave above 85 degrees Fahrenheit;
- days.above.abs.threshold.3: Number of days in the heat wave above 90 degrees Fahrenheit;
- days.above.abs.threshold.4: Number of days in the heat wave above 90 degrees Fahrenheit;
- days.above.99th: Number of days in the heat wave above the 99th percentile temperature for the city, using the period specified by the user with the referenceBoundaries argument in gen_hw_set as a reference for determining these percentiles;
- days.above.99.5th: Number of days in the heat wave above the 99.5th percentile temperature for the city, using the period specified by the user with the referenceBoundaries argument in gen_hw_set as a reference for determining these percentiles;
- first.in.year: Whether the heat wave was the first to occur in its calendar year (Note: this characteristic is likely not useful in southern hemisphere studies.);
- threshold: The temperature used as the threshold for the heat wave definition in the city;
- mean.var.quantile: The percentile of the average daily mean temperature during the heat wave compared to the city's year-round temperature distribution, based on the temperatures for the city during the period specified by the referenceBoundaries argument in gen_hw_set;

- max.var.quantile: The percentile of the highest daily mean temperature during the heat wave compared to the city's year-round temperature distribution;
- min.var.quantile: The percentile of the lowest daily mean temperature during the heat wave compared to the city's year-round temperature distribution;
- mean.yearround.var: The city's average year-round temperature, based on the temperatures for the city during the period specified by the `referenceBoundaries` argument in `gen_hw_set`;
- mean.seasonal.var: The city's average May–September temperature, based on the temperatures for the city during the period specified by the `referenceBoundaries` argument in `gen_hw_set`; and
- city: The identifier for the city, as given in the file specified in the `citycsv` argument of `gen_hw_set`.

`IDheatwaves` *Identify all heat waves in a time series*

Description

This function takes a dataframe with columns for date and projected temperature and adds columns identifying which days belong to a heat wave and giving separate numbers to identify each discrete heat wave.

Usage

```
IDheatwaves(threshold, datafr, global, custom)
```

Arguments

`threshold`	Numeric string with threshold temperature used in the heat wave definition, in degrees Fahrenheit.
`datafr`	A dataframe with daily temperature projections in the the city being processed. This dataframe must have two columns: (1) the first column must have the date of each observation, with class "Date" and; (2) the second column must have temperatures in degrees Fahrenheit. In the normal running of this package, this dataframe will be generated by the closure created by `createCityProcessor`.
`global`	An list object created by `gen_hw_set` that includes user specifications (e.g., the path to the output directory, the path to the input climate projections, the dataframe with city locations).
`custom`	An list object created by `gen_hw_set` that includes user specifications (e.g., the name of the R function to use to identify heat waves, alternative upper and lower year boundaries for the data used to determine threshold temperatures for the heat wave definition, alternative upper and lower year boundaries for the projection period of the heat wave datasets being generated).

Value

Returns the dataframe entered as `datafr`, but with new columns providing heat wave identifiers. The returned dataframe will have new columns for: `hw`: whether a day was part of a heat wave (0 : not part of a heat wave / 1: part of a heat wave); and `hw.number`: if it was part of a heat wave, the number of the heat wave (1, 2, etc.).

Note

The function actually used to identify heat waves in the time series is specified in the 'IDheatwaves' slot of the 'custom' object passed into this function. The default is the function `IDHeatwavesCPPwrapper`. The user can specify a different function using the argument `IDheatwavesFunction` in `gen_hw_set`.

`IDHeatwavesAlternative`

Identify heat waves in a time series

Description

This function identifies heat waves in a time series of temperature data using a heat wave definition that a heat wave must be a certain number of days above the greater of either a given threshold or 80 degrees Fahrenheit.

Usage

```
IDHeatwavesAlternative(threshold, datafr, numDays)
```

Arguments

`threshold`	Numeric string with threshold temperature used in the heat wave definition, in degrees Fahrenheit.
`datafr`	A dataframe with daily temperature projections in the the city being processed. This dataframe must have two columns: (1) the first column must have the date of each observation, with class "Date" and; (2) the second column must have temperatures in degrees Fahrenheit. In the normal running of this package, this dataframe will be generated by the closure created by `createCityProcessor`.
`numDays`	Integer greater than 0 giving the number of days to use in the extreme event definition (e.g., `numDays = 2` would define a heat wave as two or more days above the threshold temperature).

Value

Returns the dataframe entered as `datafr`, but with new columns providing heat wave identifiers. The returned dataframe will have new columns for whether a day was part of a heat wave (`hw`, 0 / 1), and if it was part of a heat wave, the number of the heat wave (`hw.number`).

Note

There are a few cases near the edges of data frames when this function would return that a day was not a heat wave when it was. First, if the first day of the dataset is a heat wave because preceeding days exceeded the threshold, but the second day in the dataframe is not above the threshold, this function would not capture that the first day was a heat wave. A similar caveat applies to the last day in the dataframe. In northern hemisphere studies, this should not be a concern, as it is unlikely that Jan. 1 or Dec. 31 would qualify as a heat wave. However, care should be taken when using this function either with Southern Hemisphere cities or when exploring exposures that, unlike heat waves, may occur very early or late in the calendar year.

Examples

```
data(datafr)
hw_ids <- IDHeatwavesAlternative(threshold = 80, datafr = datafr,
                                 numDays = 3)
```

IDHeatwavesCPP *Identify heat waves that are a certain number of days long*

Description

This function identifies heat waves that are a certain number of days long.

Usage

```
IDHeatwavesCPP(heatwaveLength, tempsExceedingCutoff)
```

Arguments

heatwaveLength	Number of days that a heat wave lasts under the user's definition
tempsExceedingCutoff	A vector of 1s and 0s the length of the number of days in the time series. 1 means the temperature for that day exceeds the calculated threshold; 0 means it does not exceed the threshold.

Value

A dataframe containing information about the heat waves for this series. It contains two columns: hw and hw.number.

`IDHeatwavesCPPwrapper` *Identify heat waves in a time series*

Description

This function identifies heat waves in a time series of temperature data using a heat wave definition that a heat wave must be a certain number of days with temperatures equal to or above some threshold temperature. This function uses a compiled C++ function for part of the process, making it faster than the R analogue, `IDHeatwavesR`, although the two functions give identical results.

Usage

```
IDHeatwavesCPPwrapper(datafr, threshold, numDays)
```

Arguments

`datafr`	A dataframe with daily temperature projections in the the city being processed. This dataframe must have two columns: (1) the first column must have the date of each observation, with class "Date" and; (2) the second column must have temperatures in degrees Fahrenheit. In the normal running of this package, this dataframe will be generated by the closure created by `createCityProcessor`.
`threshold`	Numeric string with threshold temperature used in the heat wave definition, in degrees Fahrenheit.
`numDays`	Integer greater than 0 giving the number of days to use in the extreme event definition (e.g., `numDays = 2` would define a heat wave as two or more days above the threshold temperature).

Details

This function is the default function used to identify heat waves in `gen_hw_set`.

Value

Returns the dataframe entered as `datafr`, but with new columns providing heat wave identifiers. The returned dataframe will have new columns for whether a day was part of a heat wave (`hw`, 0 / 1), and, if it was part of a heat wave, the number of the heat wave (`hw.number`).

Note

There are a few cases near the edges of data frames when this function would return that a day was not a heat wave when it was. First, if the first day of the dataset is a heat wave because preceeding days exceeded the threshold, but the second day in the dataframe is not above the threshold, this function would not capture that the first day was a heat wave. A similar caveat applies to the last day in the dataframe. In northern hemisphere studies, this should not be a concern, as it is unlikely that Jan. 1 or Dec. 31 would qualify as a heat wave. However, care should be taken when using this function either with Southern Hemisphere cities or when exploring exposures that, unlike heat waves, may occur very early or late in the calendar year.

Examples

```
data(datafr)
hw_ids <- IDHeatwavesCPPwrapper(threshold = 80, datafr = datafr,
                                numDays = 3)
```

`IDHeatwavesR` *Identify heat waves in a time series*

Description

This function identifies heat waves in a time series of temperature data using a heat wave definition that a heat wave must be a certain number of days with temperatures equal to or above some threshold temperature.

Usage

```
IDHeatwavesR(threshold, datafr, numDays)
```

Arguments

`threshold`	Numeric string with threshold temperature used in the heat wave definition, in degrees Fahrenheit.
`datafr`	A dataframe with daily temperature projections in the the city being processed. This dataframe must have two columns: (1) the first column must have the date of each observation, with class "Date" and; (2) the second column must have temperatures in degrees Fahrenheit. In the normal running of this package, this dataframe will be generated by the closure created by `createCityProcessor`.
`numDays`	Integer greater than 0 giving the number of days to use in the extreme event definition (e.g., `numDays = 2` would define a heat wave as two or more days above the threshold temperature).

Value

Returns the dataframe entered as `datafr`, but with new columns providing heat wave identifiers. The returned dataframe will have new columns for whether a day was part of a heat wave (`hw`, 0 / 1), and , if it was part of a heat wave, the number of the heat wave (`hw.number`).

Note

There are a few cases near the edges of data frames when this function would return that a day was not a heat wave when it was. First, if the first day of the dataset is a heat wave because preceeding days exceeded the threshold, but the second day in the dataframe is not above the threshold, this function would not capture that the first day was a heat wave. A similar caveat applies to the last day in the dataframe. In northern hemisphere studies, this should not be a concern, as it is unlikely that Jan. 1 or Dec. 31 would qualify as a heat wave. However, care should be taken when using this function either with Southern Hemisphere cities or when exploring exposures that, unlike heat waves, may occur very early or late in the calendar year.

Examples

```
data(datafr)
hw_ids <- IDHeatwavesR(threshold = 80, datafr = datafr,
                       numDays = 3)
```

`map_grid` *Create a map of model grid*

Description

This function creates a map of the grid points of a climate model used for the study locations and draws lines connecting each study city to its climate model grid point. It currently can only be used for studies within the United States.

Usage

```
map_grid(plot_model, out)
```

Arguments

`plot_model`	A character string with the name of the model to plot
`out`	Character string with pathname to which extreme event files were written by `gen_hw_set`. Typically, this will be the same pathname as that specified with `out` when running `gen_hw_set`.

Value

A `ggplot2` object with a map of grid points for the climate model that were used in processing extreme events for the study locations, with a line drawn from each study locations to the grid point used for it.

Note

This function creates a `ggplot2` object, so the output can be edited using `ggplot2` functions.

Examples

```
out <- system.file("extdata/example_results", package = "futureheatwaves")
map_grid(plot_model = "bcc1", out = out)
```

map_grid_leaflet *Create an interactive map of model grid*

Description

This function creates an interactive map of the grid points of a climate model used for the study locations and draws lines connecting each study city to its climate model grid point. This map is made using the `leaflet` package.

Usage

```
map_grid_leaflet(plot_model, out)
```

Arguments

`plot_model`	A character string with the name of the model to plot
`out`	Character string with pathname to which extreme event files were written by `gen_hw_set`. Typically, this will be the same pathname as that specified with `out` when running `gen_hw_set`.

Value

A `leaflet` object with a map of grid points for the climate model that were used in processing extreme events for the study locations, with a line drawn from each study locations to the grid point used for it. This map can be explored interactively.

Examples

```
out <- system.file("extdata/example_results", package = "futureheatwaves")
map_grid_leaflet(plot_model = "bcc1", out = out)
```

number_of_heatwaves *Calculate number of extreme_events*

Description

This function takes a dataframe of extreme events, as created by `gen_hw_set`, and calculates the number of events in the dataframe.

Usage

```
number_of_heatwaves(hw_datafr)
```

Arguments

hw_datafr	A dataframe of extreme events and their characteristics, as created by gen_hw_set

Details

To calculate the number of extreme events, this function determines the number of rows in the dataframe, since gen_hw_set outputs a dataframe with one and only one row per event.

Value

A numeric value with the number of extreme events in a dataframe of events, as generated by gen_hw_set.

Note

This function is an example of a function that can be created and used to explore extreme events using the apply_all_models function.

Examples

```
data(hw_datafr)
number_of_heatwaves(hw_datafr)
```

---

processEnsemble *Extract projections from ensemble member*

Description

This function extracts the desired climate data from a single ensemble member, from one of the two experiment subdirectories.

Usage

```
processEnsemble(ensemble, modelName, global, custom, type)
```

Arguments

ensemble	List with parsed information about the directory structure for a specific ensemble member from the user-specified projections directory. This list is a subset of the list generated by acquireDirectoryStructure.
modelName	Character string of climate model name (e.g., "bcc1"). This name is generated from the subdirectory name for the climate model within dataFolder.
global	An list object created by gen_hw_set that includes user specifications (e.g., the path to the output directory, the path to the input climate projections, the dataframe with city locations).

custom — An list object created by gen_hw_set that includes user specifications (e.g., the name of the R function to use to identify heat waves, alternative upper and lower year boundaries for the data used to determine threshold temperatures for the heat wave definition, alternative upper and lower year boundaries for the projection period of the heat wave datasets being generated).

type — A character vector giving the type of boundaries that are being checked. Possible values are "threshold", "projections", and "reference".

Value

A list object with processed projections, as well as various other elements needed for further processing to identify heat waves.

Note

This function calls another function that uses Euclidean distance for the distance between each city given in the citycsv file and the nearest grid point for the climate model for the specified ensemble member.

---

processModel *Process valid models*

Description

This function takes any models that are valid, identifies and characterizes heat waves within projections for each of its ensemble members, writes those dataframes of heat wave projections out to the user-specified output directory, and stores information about the models and grid locations.

Usage

```
processModel(model, global, custom, accumulators, dataDirectories)
```

Arguments

model — List with parsed information about the directory structure for a specific climate model from the user-specified projections directory. This list is a subset of the list generated by acquireDirectoryStructure.

global — An list object created by gen_hw_set that includes user specifications (e.g., the path to the output directory, the path to the input climate projections, the dataframe with city locations).

custom — An list object created by gen_hw_set that includes user specifications (e.g., the name of the R function to use to identify heat waves, alternative upper and lower year boundaries for the data used to determine threshold temperatures for the heat wave definition, alternative upper and lower year boundaries for the projection period of the heat wave datasets being generated).

accumulators — The closure generated by createAccumulators that allows you to append model information and grid location data as you process through models to a growing list.

dataDirectories — A list object, with two elements, one for each of the two subdirectories included in the main directory. Typically, these will be separate directories of historical and projection experiments from climate models. Each element of the list should be named with the name of the subdirectory and should provide a numeric vector with the starting and ending years of the data within each of the two subdirectories (e.g., `list("historical" = c(1990, 1999), "rcp85" = c(2060, 2079))` for a dataFolder with historical experiment data for 1990 to 1999 and RCP8.5 projections for 2060 to 2079).

Value

A list object with processed projections, as well as various other elements needed for further processing to identify heat waves.

---

processProjections *Create heat wave dataframe for climate projection*

Description

This function, for each ensemble member, creates the heat wave dataframe and writes it to file. It also stores the locations for each ensemble.

Usage

```
processProjections(ensemble, modelName, ensembleWriter, thresholds, global,
  custom, accumulators, reference, reference_dates)
```

Arguments

ensemble — List with parsed information about the directory structure for a specific ensemble member from the user-specified projections directory. This list is a subset of the list generated by acquireDirectoryStructure.

modelName — Character string of climate model name (e.g., "bcc1"). This name is generated from the subdirectory name for the climate model within dataFolder.

ensembleWriter — A closure created by createEnsembleWriter to write output from this process to a file within the output directory specified by the user with the out argument in gen_hw_set.

thresholds — A vector with the thresholds to use within each city in that city's heat wave definition.These are typically automatically determine during the run of the gen_hw_set function.

global — An list object created by gen_hw_set that includes user specifications (e.g., the path to the output directory, the path to the input climate projections, the dataframe with city locations).

custom — An list object created by gen_hw_set that includes user specifications (e.g., the name of the R function to use to identify heat waves, alternative upper and lower year boundaries for the data used to determine threshold temperatures for the heat wave definition, alternative upper and lower year boundaries for the projection period of the heat wave datasets being generated).

accumulators — The closure generated by createAccumulators that allows you to append model information and grid location data as you process through models to a growing list.

reference — FALSE, if the user has not specified custom reference boundaries through the referenceBoundaries argument in gen_hw_set, otherwise a dataframe with the time series of projected temperatures for the custom reference period for each study city.

reference_dates — A numeric vector with the start and end years to use for the reference period

Value

This function writes every heat wave dataframe to a .csv and returns the ensemble member used as the reference.

---

processReference *Get projection data for reference period*

Description

This function is only run if the reference period is different from the projection period. In that case, this function will acquire the time series of projected temperatures during the indicated reference period and pass that through to be used in a later function to characterize heat waves.

Usage

```
processReference(model, global, custom)
```

Arguments

model — List with parsed information about the directory structure for a specific climate model from the user-specified projections directory. This list is a subset of the list generated by acquireDirectoryStructure.

global — An list object created by gen_hw_set that includes user specifications (e.g., the path to the output directory, the path to the input climate projections, the dataframe with city locations).

custom — An list object created by gen_hw_set that includes user specifications (e.g., the name of the R function to use to identify heat waves, alternative upper and lower year boundaries for the data used to determine threshold temperatures for the heat wave definition, alternative upper and lower year boundaries for the projection period of the heat wave datasets being generated).

Value

A list with, among other elements, a `series` element with the time series of projected temperatures for the reference period for each study city.

`processThresholds` *Calculate threshold temperatures*

Description

This function calculates the threshold temperatures required to identify heat waves in the climate projection data using the ensemble member specified by `threshold_ensemble` in `gen_hw_set`. This threshold is used in later functions to identify heat waves.

Usage

```
processThresholds(model, global, custom)
```

Arguments

`model`	List with parsed information about the directory structure for a specific climate model from the user-specified projections directory. This list is a subset of the list generated by `acquireDirectoryStructure`.
`global`	An list object created by `gen_hw_set` that includes user specifications (e.g., the path to the output directory, the path to the input climate projections, the dataframe with city locations).
`custom`	An list object created by `gen_hw_set` that includes user specifications (e.g., the name of the R function to use to identify heat waves, alternative upper and lower year boundaries for the data used to determine threshold temperatures for the heat wave definition, alternative upper and lower year boundaries for the projection period of the heat wave datasets being generated).

Value

A list with two elements: (1) a vector with one element for each city included in the user-specified city location file, with each element value giving the threshold temperature for the heat wave definition for a city, in the same order that cities are listed in the user-specified `citycsv` file and (2) the components of the dataframe of climate model grid locations that will ultimately be output as the climate model grid locations file.

process_cities_file *Process city file*

Description

This function takes the dataframe read in from the file specified with `citycsv` in `gen_hw_set` and renames the columns for latitude and longitude `lat` and `lon`, based on the user's selections in `lat_lon_colnames` for `gen_hw_set`. If there are extra columns besides those and the `city` column, this function removes them.

Usage

```
process_cities_file(cities, lat_lon_colnames)
```

Arguments

`cities`	Dataframe with study cities and their latitudes and longitudes.The dataframe must have a column named `city` with a unique identifier for each city in the study, as well as columns for latitude and longitude. Other columns may be included in the dataset, but will not be passed through to later code.
`lat_lon_colnames`	A character vector of length two with the column names in the `citycsv` dataframe for latitude (first vector element) and longitude (second vector element)

Value

A processed version of the latitude and longitude dataframe.

readLatLong *Read latitude and longitude data*

Description

This function reads in data on the longitudes and latitudes of a climate model's grid pooints from a locally-stored comma-separated file located within the directory specified by the `dataFolder` argument of `gen_hw_set`.

Usage

```
readLatLong(ensemble, global)
```

Arguments

ensemble	Character vector that includes the file paths to (1) the latitude and longitude file; (2) the climate projection file; and (3) the projection dates file for the selected climate model.
global	An list object created by gen_hw_set that includes user specifications (e.g., the path to the output directory, the path to the input climate projections, the dataframe with city locations).

Value

A dataframe of the latitude and longitude data of the ensemble with columns for the latitude (first column) and longitude (second column) of each grid point in the climate model.

---

readtas *Read climate projection data*

Description

This function reads climate projection data for a climate model into R from a locally-stored comma-separated file located within the directory specified by the dataFolder argument of gen_hw_set.

Usage

```
readtas(ensemble, global, locations)
```

Arguments

ensemble	Character vector that includes the file paths to (1) the latitude and longitude file; (2) the climate projection file; and (3) the projection dates file for the selected climate model.
global	An list object created by gen_hw_set that includes user specifications (e.g., the path to the output directory, the path to the input climate projections, the dataframe with city locations).
locations	A numerical vector giving the column numbers that correspond to the closest grid point location for each study location in the temperature input data.

Value

A dataframe of climate projection data where each column corresponds to a climate model grid point and each row corresponds to a date of observation.

readTimes *Read projection dates data*

Description

This function reads the dates corresponding to climate projection data for a climate model into R from a locally-stored comma-separated file located within the directory specified by the `dataFolder` argument of `gen_hw_set`.

Usage

```
readTimes(ensemble, global)
```

Arguments

`ensemble`	Character vector that includes the file paths to (1) the latitude and longitude file; (2) the climate projection file; and (3) the projection dates file for the selected climate model.
`global`	An list object created by `gen_hw_set` that includes user specifications (e.g., the path to the output directory, the path to the input climate projections, the dataframe with city locations).

Value

A dataframe of dates corresponding to climate projection data, where each row gives a date, split into columns of day in the dataframe, four-digit year, numeric month, and numeric day.

storeHeatwaveEntry *Helper function that adds entries*

Description

This is a helper function for IDHeatwavesCPP that adds entries to the data structures that track heatwave information

Usage

```
storeHeatwaveEntry(index, hwSize, hwCounter, hw, hwNumber)
```

Arguments

`index`	A running index.
`hwSize`	Size of the heat wave to be added.
`hwCounter`	Current number of heat waves.
`hw`	A reference of the vector that contains the heat waves
`hwNumber`	A reference of the vector that contains the heat wave numbers.

storeZeroes *Helper function that adds zeroes*

Description

This function is a helper function that adds zeros to the data structures that track heat wave information

Usage

```
storeZeroes(index, potentialHeatwave, hw, hwNumber)
```

Arguments

index	A running index.
potentialHeatwave	Size of the potential heat wave that turned out not to be a heat wave.
hw	A reference of the vector that contains the heat waves
hwNumber	A reference of the vector that contains the heat wave numbers.

writeAccumulators *Write model information to file*

Description

This function writes out a dataframe that accumulates information on all climate models included in the analysis, including the number of ensemble members in each of the two experiments.

Usage

```
writeAccumulators(modelInfoAccumulator, locationList, global)
```

Arguments

modelInfoAccumulator	A dataframe with information about the number of ensembles for each climate model.
locationList	A dataframe with information about city locations and the closest grid point from each climate model to each city.
global	An list object created by gen_hw_set that includes user specifications (e.g., the path to the output directory, the path to the input climate projections, the dataframe with city locations).

Value

This function writes out a dataframe with information on all climate models included in the analysis, including the number of ensemble members in each of the two experiments.

Index