

Package ‘noaastormevents’

October 13, 2022

Type Package

Title Explore NOAA Storm Events Database

Version 0.2.0

Description Allows users to explore and plot data from the National Oceanic and Atmospheric Administration (NOAA) Storm Events database through R for United States counties. Functionality includes matching storm event listings by time and location to hurricane best tracks data. This work was supported by grants from the Colorado Water Center, the National Institute of Environmental Health Sciences (R00ES022631) and the National Science Foundation (1331399).

URL <https://github.com/geanders/noaastormevents>

BugReports <https://github.com/geanders/noaastormevents/issues>

License GPL (>= 2)

LazyData TRUE

Imports choroplethr (>= 3.7.0), choroplethrMaps (>= 1.0.1), data.table (>= 1.13.0), dplyr (>= 1.0.2), forcats (>= 0.5.0), ggplot2 (>= 3.3.2), hurricaneexposure (>= 0.1.1), lubridate (>= 1.7.9), maps (>= 3.3.0), magrittr (>= 1.5), plyr (>= 1.8.6), RColorBrewer (>= 1.1.2), RCurl (>= 1.98-1.2), rlang (>= 0.4.7), stringr (>= 1.4.0), tibble (>= 3.0.3), tidyr (>= 1.1.1), viridis (>= 0.5.1), XML (>= 3.99-0.3)

RoxygenNote 7.1.1

Depends R (>= 3.5)

Suggests hurricaneexposedata (>= 0.1.0), knitr (>= 1.29.0), pander (>= 0.6.3), rmarkdown (>= 2.3), testthat (>= 2.3.2)

VignetteBuilder knitr

Additional_repositories <https://geanders.github.io/drat>

Encoding UTF-8

NeedsCompilation no

Author Brooke Anderson [aut, cre],
 Ziyu Chen [aut],
 Therese Kondash [aut]

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

Repository CRAN

Date/Publication 2021-01-21 10:20:06 UTC

R topics documented:

adjust_storm_data	2
clean_storm_data	3
create_storm_data	4
download_storm_data	4
find_events	5
find_file_name	6
get_county_map	7
map_events	8
match_forecast_county	9
parse_damage	11
process_input_args	12
Index	13

adjust_storm_data	<i>Adjust storm data</i>
-------------------	--------------------------

Description

Adjusts storm data based on user selections on date range, distance limit to a storm, etc.

Usage

```
adjust_storm_data(
  storm_data,
  date_range = NULL,
  event_types = NULL,
  dist_limit = NULL,
  storm = NULL
)
```

Arguments

storm_data	A dataset of storm data. This dataset must include certain columns given in the NOAA Storm Events datasets for which this package was created.
date_range	A character vector of length two with the start and end dates to pull data for (e.g., c("1999-10-16", "1999-10-18")).

event_types	Character vector with the types of storm events that should be kept. The default value (NULL) keeps all types of events. See the "Details" vignette for this package for more details on possible event types.
dist_limit	A numeric scalar with the distance (in kilometers) that a county must be from the storm's path to be included. The default (NULL) does not eliminate any events based on distance from a storm's path. This option should only be used when also specifying a storm with the storm parameter.
storm	A character string with the name of the storm to pull storm events data for. This string must follow the format "[storm-name]-[4-digit storm year]" (e.g., "Floyd-1999"). Currently, this functionality only works for storms included in the extended hurricane best tracks, which covers 1988 to 2015.

`clean_storm_data`*Clean storm dataset*

Description

Cleans the storm dataset to prepare for further processing. This includes changing all variable names to lowercase, removing some unneeded columns, and removing the narratives if requested by the user.

Usage

```
clean_storm_data(storm_data, include_narratives)
```

Arguments

storm_data	A dataset of storm data. This dataset must include certain columns given in the NOAA Storm Events datasets for which this package was created.
include_narratives	A logical value for whether the final data data frame should include columns for episode and event narratives (TRUE) or not (FALSE, the default)

Value

A cleaned version of the dataset input to the function.

create_storm_data *Get storm data based on date range or storm name*

Description

This function pulls storm events data based on a specified date range and / or storm name. (Note: This function pulls full years' worth of data. Later functions filter down to the exact date range desired.)

Usage

```
create_storm_data(date_range = NULL, storm = NULL, file_type = "details")
```

Arguments

date_range	A character vector of length two with the start and end dates to pull data for (e.g., c("1999-10-16", "1999-10-18")).
storm	A character string with the name of the storm to pull storm events data for. This string must follow the format "[storm-name]-[4-digit storm year]" (e.g., "Floyd-1999"). Currently, this functionality only works for storms included in the extended hurricane best tracks, which covers 1988 to 2015.
file_type	A character string specifying the type of file you would like to pull. Choices include: "details" (the default), "fatalities", or "locations".

Examples

```
## Not run:
floyd_data <- create_storm_data(date_range = c("1999-10-16", "1999-10-18"))
floyd_data2 <- create_storm_data(storm = "Floyd-1999")

## End(Not run)
```

download_storm_data *Download storm data file for a given year*

Description

This function takes a year for which you want to download storm data, checks to see if it's already been downloaded and cached, and, if not, downloads and caches it from the NOAA's online storm events files.

Usage

```
download_storm_data(year, file_type = "details")
```

Arguments

year	A four-digit numeric or character string giving the year for which the user would like to download data.
file_type	A character string specifying the type of file you would like to pull. Choices include: "details" (the default), "fatalities", or "locations".

Note

This function caches downloaded storm data into an object called `lst` that persist throughout the R session but is deleted at the end of the R session (as long as the R history is not saved at the end of the session). This saves time if the user uses the storm data from the same year for several commands.

find_events	<i>Find all event listings for date range</i>
-------------	---

Description

This function will find all of the events in the US for a specified date range.

Usage

```
find_events(
  date_range = NULL,
  event_types = NULL,
  dist_limit = NULL,
  storm = NULL,
  include_narratives = FALSE,
  include_ids = FALSE,
  clean_damage = FALSE
)
```

Arguments

date_range	A character vector of length two with the start and end dates to pull data for (e.g., <code>c("1999-10-16", "1999-10-18")</code>).
event_types	Character vector with the types of storm events that should be kept. The default value (NULL) keeps all types of events. See the "Details" vignette for this package for more details on possible event types.
dist_limit	A numeric scalar with the distance (in kilometers) that a county must be from the storm's path to be included. The default (NULL) does not eliminate any events based on distance from a storm's path. This option should only be used when also specifying a storm with the <code>storm</code> parameter.
storm	A character string with the name of the storm to pull storm events data for. This string must follow the format "[storm-name]-[4-digit storm year]" (e.g., "Floyd-1999"). Currently, this functionality only works for storms included in the extended hurricane best tracks, which covers 1988 to 2015.

include_narratives	A logical value for whether the final data data frame should include columns for episode and event narratives (TRUE) or not (FALSE, the default)
include_ids	A logical value for whether the final data frame should include columns for event and episode IDs (TRUE) or not (FALSE, the default). If included, these IDs could be used in some cases to link events to data in the "fatalities" or "locations" files available through the NOAA Storm Events database.
clean_damage	TRUE / FALSE of whether additional cleaning should be done to try to exclude incorrect damage listings. If TRUE, any property or crop damages for which the listing for that single event exceeds all other damages in the state combined for the event dataset, the damages for that event listing will be set to missing. Default is FALSE (i.e., this additional check is not performed). In some cases, it seems that a single listing by forecast zone gives the state total for damages, and this option may help in identifying and excluding such listings (for example, one listing in North Carolina for Hurricane Floyd seems to be the state total for damages, rather than a county-specific damage estimate).

Examples

```
## Not run:
# Events by date range
find_events(date_range = c("1999-09-10", "1999-09-30"))

# Events within a certain distance and time range of a tropical storm
find_events(storm = "Floyd-1999", dist_limit = 200)

# Limit output to events that are floods or flash floods
find_events(storm = "Floyd-1999", dist_limit = 200, event_types = c("Flood", "Flash Flood"))

## End(Not run)
```

find_file_name	<i>Find a database file name</i>
----------------	----------------------------------

Description

This function will find the name of the detailed file from Storm Events Database for a specific year and specific type of file. This file name can then be used (in other functions) to download the data for a given year.

Usage

```
find_file_name(year = NULL, file_type = "details")
```

Arguments

year	A four-digit numeric or character string giving the year for which the user would like to download data.
file_type	A character string specifying the type of file you would like to pull. Choices include: "details" (the default), "fatalities", or "locations".

Details

This function creates a list of all file names available on <https://www1.ncdc.noaa.gov/pub/data/swdi/stormevents/csvfiles/> and then uses regular expressions to search that list for the name of the file for the year and type of file requested. While the files are named consistently, part of the name includes the date the file was last updated, which changes frequently. The method used here is robust to changes in this "last updated" date within the file names.

Examples

```
## Not run:  
find_file_name(year = 1999)  
find_file_name(year = 2003, file_type = "fatalities")  
  
## End(Not run)
```

get_county_map	<i>Get map data for counties</i>
----------------	----------------------------------

Description

Get map data for counties

Usage

```
get_county_map(states = "east")
```

Arguments

states	A character string specifying either a state name or names or one of "all" (map all states in the continental US) or "east" (plot states in the Eastern half of the US. The default is "east").
--------	---

Value

A dataframe with map data pulled using the map_data function in ggplot2, filtered to states in the eastern half of the United States if the user specifies east_only.

 map_events

Map storm events for a date range

Description

This function maps all storm events listed with a starting date within a specified date range.

Usage

```
map_events(
  event_data,
  states = "east",
  plot_type = "any events",
  storm = NULL,
  add_tracks = FALSE
)
```

Arguments

event_data	A dataframe of event data, as returned by the <code>find_events</code> function.
states	A character string specifying either a state name or names or one of "all" (map all states in the continental US) or "east" (plot states in the Eastern half of the US. The default is "east").
plot_type	Specifies the type of plot wanted. It can be "any events", "number of events", "direct deaths", "indirect deaths", "direct injuries", "indirect injuries", "property damage", or "crop damage".
storm	A character string with the name of the storm to pull storm events data for. This string must follow the format "[storm-name]-[4-digit storm year]" (e.g., "Floyd-1999"). Currently, this functionality only works for storms included in the extended hurricane best tracks, which covers 1988 to 2015.
add_tracks	A logical value specifying whether to add the tracks of a hurricane to the map (default = FALSE).

Note

Indirect deaths and injuries seem to be reported very rarely, so it is likely that trying to map either of these outcomes will result in a note that no indirect deaths / injuries were reported for the selected events.

Examples

```
## Not run:
# Map for events pulled by a date range
event_data <- find_events(date_range = c("1999-09-10", "1999-09-30"))
map_events(event_data)
map_events(event_data, plot_type = "number of events")
```



```

# Map for a specific type of event
event_data <- find_events(date_range = c("1999-09-10", "1999-09-30"),
                          event_types = c("Flood", "Flash Flood"))
map_events(event_data, states = "north carolina", plot_type = "number of events")
map_events(event_data, states = "all")

# Map for events identified based on a hurricane storm track
event_data <- find_events(storm = "Floyd-1999", dist_limit = 300)
map_events(event_data, plot_type = "number of events",
           storm = "Floyd-1999", add_tracks = TRUE)
map_events(event_data, plot_type = "crop damage",
           storm = "Floyd-1999", add_tracks = TRUE,
           states = c("north carolina", "virginia", "maryland"))
map_events(event_data, plot_type = "property damage",
           storm = "Floyd-1999", add_tracks = TRUE)
map_events(event_data, plot_type = "direct deaths")

event_data <- find_events(date_range = c("1999-01-01", "1999-12-31"))
map_events(event_data, plot_type = "direct deaths")
map_events(event_data, plot_type = "indirect deaths")
map_events(event_data, plot_type = "direct injuries")
map_events(event_data, plot_type = "indirect injuries")
map_events(event_data, plot_type = "crop damage")

## End(Not run)

```

match_forecast_county *Match events by forecast zone to county*

Description

For events reported by forecast zone, use regular expressions to match as many as possible to counties.

Usage

```
match_forecast_county(storm_data_z)
```

Arguments

storm_data_z A dataframe of storm events reported by forecast zone (i.e., `cz_type == "Z"`) rather than county. This dataframe should include the columns:

- `state`: State name, in lowercase
- `cz_name`: Location name, in lowercase
- `cz_fips`: Forecast zone FIPS

Details

This function tries to match the `cz_name` of each event to a state and county name from the `county.fips` dataframe that comes with the `maps` package. The following steps are taken to try to match each `cz_name` to a state and county name from `county.fips`:

1. Tries to match `cz_name` to the county name in `county.fips` after removing any periods or apostrophes in `cz_name`.
2. Next, for county names with "county" in them, try to match the word before "county" to county name in `county.fips`. Then check the two words before "county", then the one and two words before "counties".
3. Next, pull out the last word in `cz_name` and try to match it to the county name in `county.fips`. The check the last two words in `cz_name`, then check the last three words in `cz_name`.
4. Next, pull any words right before a slash and check that against the county name.
5. Finally, try removing anything in parentheses in `cz_name` before matching.

Value

The dataframe of events input to the function, with county FIPS added for events matched to a county in the `fips` column. Events that could not be matched are kept in the dataframe, but the `fips` code is set to NA.

Note

This function does not provide any matches for events outside of the continental U.S.

You may want to hand-check that event listings with names like "Lake", "Mountain", and "Park" have not been unintentionally linked to a county like "Lake County". While such examples seem rare in the example data used to develop this function (NOAA Storm Events for 2015), it can sometimes happen. To do so, you can use the `str_detect` function from the `stringr` package.

Examples

```
counties_to_parse <- dplyr::data_frame(
  event_id = c(1:19),
  cz_name = c("Suffolk",
             "Eastern Greenbrier",
             "Ventura County Mountains",
             "Central And Southeast Montgomery",
             "Western Cape May",
             "San Diego County Coastal Areas",
             "Blount/Smoky Mountains",
             "St. Mary's",
             "Central & Eastern Lake County",
             "Mountains Southwest Shasta County To Northern Lake County",
             "Kings (Brooklyn)",
             "Lower Bucks",
             "Central St. Louis",
             "Curry County Coast",
             "Lincoln County Except The Sheep Range",
             "Shasta Lake/North Shasta County",
```

```

        "Coastal Palm Beach County",
        "Larimer & Boulder Counties Between 6000 & 9000 Feet",
        "Yellowstone National Park"),
state = c("Virginia",
        "West Virginia",
        "California",
        "Maryland",
        "New Jersey",
        "California",
        "Tennessee",
        "Maryland",
        "Oregon",
        "California",
        "New York",
        "Pennsylvania",
        "Minnesota",
        "Oregon",
        "Nevada",
        "California",
        "Florida",
        "Colorado",
        "Wyoming"))
match_forecast_county(counties_to_parse)

```

parse_damage

Parse damage values

Description

Take damage values that include letters for order of magnitude (e.g., "2K" for \$2,000) and return a numeric value of damage.

Usage

```
parse_damage(damage_vector)
```

Arguments

damage_vector A character vector with damage values (e.g., the `damage_crops` or `damage_property` columns in the NOAA Storm Events data). This vector should give numbers except for specific abbreviations specifying order of magnitude (see Details).

Details

This function parses the following abbreviations for order of magnitude:

- "K": 1,000 (thousand)
- "M": 1,000,000 (million)
- "B": 1,000,000,000 (billion)
- "T": 1,000,000,000,000 (trillion)

Value

The input vector, parsed to a numeric, with abbreviations for orders of magnitude appropriately interpreted (e.g., "2K" in the input vector becomes the numeric 2000 in the output vector).

Examples

```
damage_crops <- c("150", "2K", "3.5B", NA)
parse_damage(damage_crops)
```

process_input_args *Process inputs to main functions*

Description

Processes some of the user's inputs for arguments for main package functions, looks for any errors in input, and determines elements like the year or years of storm data needed based on user inputs.

Usage

```
process_input_args(date_range = NULL, storm = NULL)
```

Arguments

date_range	A character vector of length two with the start and end dates to pull data for (e.g., c("1999-10-16", "1999-10-18")).
storm	A character string with the name of the storm to pull storm events data for. This string must follow the format "[storm-name]-[4-digit storm year]" (e.g., "Floyd-1999"). Currently, this functionality only works for storms included in the extended hurricane best tracks, which covers 1988 to 2015.

Value

A list with date ranges and storm identification based on user inputs to arguments in a main package function.

Index

[adjust_storm_data](#), 2

[clean_storm_data](#), 3

[create_storm_data](#), 4

[download_storm_data](#), 4

[find_events](#), 5

[find_file_name](#), 6

[get_county_map](#), 7

[map_events](#), 8

[match_forecast_county](#), 9

[parse_damage](#), 11

[process_input_args](#), 12