

Arguments

map                    A character string giving the map database from which to pull

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.

hurr\_choropleth                    Create a map customized for this package

Description

Creates a county choropleth map customized for displaying hurricane exposure. This function is used as a helper function within other mapping functions in the package.

Usage

```
hurr_choropleth(  
  map_data,  
  metric = "distance",  
  wind_var = "vmax_sust",  
  wind_source = "modeled"  
)
```

Arguments

map\_data                    A dataframe with columns with FIPS numbers (in numeric class) for all counties in the eastern US (region) and the exposure value (value)

metric                    Character string giving the metric to plot. Current options are "distance", "wind", and "rainfall". These options are used to customize the color palette and scale of the choropleth map produced by this function.

wind\_var                    A character string giving the wind variable to use. Choices are "vmax\_sust" (maximum sustained winds; default), "vmax\_gust" (maximum gust winds), "sust\_dur" (minutes of sustained winds of 20 m / s or higher) and "gust\_dur" (minutes of gust winds of 20 m / s or higher). If the Extended Best Tracks wind radii are used as the source of wind data, the "gust\_dur" option cannot be selected.

wind\_source                    A character string specifying the source to use for the winds. Options are "modeled", for estimates based on running a wind model from Best Tracks data inputs, and "ext\_tracks", for estimates based on the wind radii in the Extended Best Tracks data. See the help files for the datasets storm\_winds and ext\_tracks\_wind in the hurricaneexposedata package for more details on each of these sources for wind estimates. For the gust wind estimates, these are based on applying a gust factor of 1.49 to the sustained wind estimates in both wind data sources.

Capitalized?



Details

The function only maps counties in states likely to be exposed to Atlantic basin tropical storms.

Value

A ggplot object with a map of hurricane exposure in eastern US counties

Not sure if we want e or E but both have been used throughout

interp_track	Interpolate a storm track
--------------	---------------------------

Description

This function takes a wider-spaced storm track (e.g., every 6 hours) and interpolates to every 15 minutes. To do this, it uses natural cubic spline interpolation using the 'spline' function from the 'stats' package. The track is only interpolated if there are three or more observations on the central location of the storm (this is almost always the case for storms tracked in the HURDAT2 dataset).

Usage

```
interp_track(track)
```

Arguments

track                      A dataframe with hurricane track data for a single storm

Value

A dataframe with hurricane track data for a single storm, interpolated to 15-minute intervals.

map_counties	Map counties
--------------	--------------

Description

Map counties

Usage

```
map_counties(  
  storm,  
  metric = "distance",  
  wind_var = "vmax_sust",  
  days_included = c(-2, -1, 0, 1),  
  add_track = TRUE,  
  wind_source = "modeled"  
)
```

**Arguments**

storm	Character string giving the name of the storm to plot (e.g., "Floyd-1999")
metric	Character string giving the metric to plot. Current options are "distance", "wind", and "rainfall". These options are used to customize the color palette and scale of the choropleth map produced by this function.
wind_var	A character string giving the wind variable to use. Choices are "vmax_sust" (maximum sustained winds; default), "vmax_gust" (maximum gust winds), "sust_dur" (minutes of sustained winds of 20 m / s or higher) and "gust_dur" (minutes of gust winds of 20 m / s or higher). If the Extended Best Tracks wind radii are used as the source of wind data, the "gust_dur" option cannot be selected.
days_included	A numeric vector listing the days to include when calculating total precipitation. Negative numbers are days before the closest date of the storm to a county. For example, c(-1, 0, 1) would calculate rain for a county as the sum of the rainfall for the day before, the day of, and the day after the date when the storm center was closest to the county center. Values can range from -5 to 3 (i.e., at most, you can calculate the total rainfall from five days to three days after the day when the storm is closest to the county).
add_track	TRUE / FALSE of whether to add the storm's track to the map. The default is TRUE.
wind_source	A character string specifying the source to use for the winds. Options are "modeled", for estimates based on running a wind model from Best Tracks data inputs, and "ext_tracks", for estimates based on the wind radii in the Extended Best Tracks data. See the help files for the datasets storm_winds and ext_tracks_wind in the hurricaneexposuredata package for more details on each of these sources for wind estimates. For the gust wind estimates, these are based on applying a gust factor of 1.49 to the sustained wind estimates in both wind data sources.

← period needed

**Value**

This function creates a choropleth map of counties in the eastern part of the United States, showing distance from a storm track, maximum wind speed (or duration of winds at or above 20 m / s), or total rainfall over a given window of one or more days near the date of the storm's closest approach.

**Examples**

```
# Ensure that data package is available before running the example.
# If it is not, see the `hurricaneexposure` package vignette for details
# on installing the required data package.
if (requireNamespace("hurricaneexposuredata", quietly = TRUE)) {

  map_counties("Floyd-1999", metric = "rainfall", days_included = c(-2, -1, 0, 1))

  map_counties("Katrina-2005", metric = "wind")
  map_counties("Katrina-2005", metric = "wind", wind_var = "vmax_gust")
  map_counties("Katrina-2005", metric = "wind", wind_var = "sust_dur")
  map_counties("Katrina-2005", metric = "wind", wind_source = "ext_tracks")
}
```



```
#' map_counties("Michael-2018", metric = "wind")
map_counties("Michael-2018", metric = "wind", wind_var = "vmax_gust")
map_counties("Michael-2018", metric = "wind", wind_source = "ext_tracks")
}
```

---

map\_distance\_exposure *Map counties with distance exposure*

---

### Description

Map counties as "exposed" or "unexposed" based on the criterion that the storm came within a given distance (specified by `dist_limit`) of the county's population mean center.

### Usage

```
map_distance_exposure(storm, dist_limit, add_track = TRUE)
```

### Arguments

<code>storm</code>	Character string giving the name of the storm to plot (e.g., "Floyd-1999").
<code>dist_limit</code>	Maximum distance, in kilometers, of how close the storm track must come to the county's population mean center to classify the county as "exposed" to the storm.
<code>add_track</code>	TRUE / FALSE of whether to add the storm's track to the map. The default is TRUE.

← period needed

### Value

Plots a map showing whether eastern US counties were exposed or unexposed to a specific storm based on a distance criterion.

### Examples

```
# Ensure that data package is available before running the example.
# If it is not, see the `hurricaneexposure` package vignette for details
# on installing the required data package.
if (requireNamespace("hurricaneexposedata", quietly = TRUE)) {

  floyd_map <- map_distance_exposure(storm = "Floyd-1999", dist_limit = 75)
  floyd_map

  allison_map <- map_distance_exposure(storm = "Allison-2001",
                                       dist_limit = 75)
  map_tracks("Allison-2001", plot_points = FALSE, plot_object = allison_map)
}
```

map_event_exposure	Map county-level exposure based on reported events
--------------------	--

Description

Map counties as "exposed" or "unexposed" based on the criterion that the county had an event listing of a specified type in the NOAA Storm Events database. For more information on the underlying data, see the helpfile for the storm\_events dataset.

Usage

```
map_event_exposure(storm_id, event_type, add_track = TRUE)
```

Arguments

storm_id	Character vector with the storm for which to map events (e.g., "Katrina-2005").
event_type	Character string with the type of event to use to identify county exposures. Options include "flood", "tornado", "wind", and "tropical_storm".
add_track	TRUE / FALSE of whether to add the storm's track to the map. The default is TRUE.

← periods needed

Value

A map showing whether eastern US counties were exposed or unexposed to a specific storm based on event listings.

Note

Note that flood events are not available for any year before 1996.

Examples

```
# Ensure that data package is available before running the example.
# If it is not, see the 'hurricaneexposure' package vignette for details
# on installing the required data package.
if (requireNamespace("hurricaneexposuredata", quietly = TRUE)) {

  map_event_exposure(storm_id = "Floyd-1999", event_type = "flood")
  map_event_exposure(storm_id = "Floyd-1999", event_type = "tornado")
  map_event_exposure(storm_id = "Floyd-1999", event_type = "wind")
  map_event_exposure(storm_id = "Floyd-1999", event_type = "tropical_storm")

  map_event_exposure(storm_id = "Florence-2018", event_type = "flood")
  map_event_exposure(storm_id = "Florence-2018", event_type = "tropical_storm")

  map_event_exposure(storm_id = "Michael-2018", event_type = "wind")
  map_event_exposure(storm_id = "Michael-2018", event_type = "tropical_storm")
}
```

---

map_rain_exposure	Map counties with rain exposure
-------------------	---------------------------------

---

**Description**

Map counties as "exposed" or "unexposed" based on the criteria that the storm came within a given distance (specified by `dist_limit`) of the county's population mean center and a certain amount of rain `rain_limit` fell during a specified window of days (`days_included`).

**Usage**

```
map_rain_exposure(  
  storm,  
  rain_limit,  
  dist_limit,  
  days_included = c(-2, -1, 0, 1),  
  add_track = TRUE  
)
```

**Arguments**

storm	Character string giving the name of the storm to plot (e.g., "Floyd-1999")
rain_limit	Minimum of rainfall, in millimeters, summed across the days selected to be included ( <code>days_included</code> ), that must fall in a county for the county to be classified as "exposed" to the storm.
dist_limit	Maximum distance, in kilometers, of how close the storm track must come to the county's population mean center to classify the county as "exposed" to the storm.
days_included	A numeric vector listing the days to include when calculating total precipitation. Negative numbers are days before the closest date of the storm to a county. For example, <code>c(-1, 0, 1)</code> would calculate rain for a county as the sum of the rainfall for the day before, the day of, and the day after the date when the storm center was closest to the county center. Values can range from -5 to 3 (i.e., at most, you can calculate the total rainfall from five days to three days after the day when the storm is closest to the county).
add_track	TRUE / FALSE of whether to add the storm's track to the map. The default is TRUE.

← periods needed

**Examples**

```
# Ensure that data package is available before running the example.  
# If it is not, see the `hurricaneexposure` package vignette for details  
# on installing the required data package.  
if (requireNamespace("hurricaneexposedata", quietly = TRUE)) {  
  
  map_rain_exposure(storm = "Floyd-1999", rain_limit = 50, dist_limit = 100)
```

```
# Example of customizing track appearance
allison_map <- map_rain_exposure(storm = "Allison-2001", rain_limit = 125,
                                dist_limit = 100, days_included = -5:3,
                                add_track = FALSE)
map_tracks("Allison-2001", plot_object = allison_map, plot_points = TRUE)
}
```

---

map_tracks	<i>Plot Atlantic basin hurricane tracks</i>
------------	---

---

**Description**

Plot the tracks of any selected storms in the hurricane tracking dataset for the Atlantic basin. This function allows you to plot a new map or add the tracks to an existing ggplot object.

**Usage**

```
map_tracks(
  storms,
  plot_object = NULL,
  padding = 2,
  plot_points = FALSE,
  alpha = 1,
  color = "firebrick"
)
```

**Arguments**

storms	Character vector with the names of all storms to plot. This parameter must use the unique storm identifiers from the 'storm_id' column of the 'hurr_tracks' dataframe.
plot_object	NULL or the name of a ggplot object to use as the underlying plot object. If NULL, the function will generate a new map of the eastern US states using 'default_map'.
padding	Numerical value giving the number of degrees to add to the outer limits of the plot object (or default map if 'plot_object' is left as NULL) when cropping hurricane tracks.
plot_points	TRUE / FALSE indicator of whether to include points, as well as lines, when plotting the hurricane tracks. These points show the times for which observations were recorded in the hurricane track data.
alpha	Numerical value designating the amount of transparency to use for plotting tracks.
color	Character string giving the color to use to plot the tracks.



Value

Returns a ggplot object with plotting data for the storm tracks of the selected storms. This object can be printed directly or added to with other ggplot2 commands.

Examples

```
# Ensure that data package is available before running the example.
# If it is not, see the `hurricaneexposure` package vignette for details
# on installing the required data package.
if (requireNamespace("hurricaneexposuredata", quietly = TRUE)) {

  map_tracks(storms = "Sandy-2012")
  map_tracks(storms = "Floyd-1999", plot_points = TRUE)
  map_tracks(storms = c("Sandy-2012", "Floyd-1999"))
  a <- map_tracks(storms = "Sandy-2012", color = "blue", alpha = 0.3)
  b <- map_tracks(storms = "Floyd-1999", plot_object = a)
  b
}
```

---

map_wind_exposure	Map counties with wind exposure
-------------------	---------------------------------

---

Description

Map counties as "exposed" or "unexposed" based on the criterion that the storm had a certain wind speed (specified by wind\_limit, in meters per second).

Usage

```
map_wind_exposure(
  storm,
  wind_var = "vmax_sust",
  wind_limit,
  add_track = TRUE,
  wind_source = "modeled"
)
```

Arguments

storm	A character string giving the storm ID (e.g., "Floyd-1999")
wind_var	A character string giving the wind variable to use. Choices are "vmax_sust" (maximum sustained winds; default), "vmax_gust" (maximum gust winds), "sust_dur" (minutes of sustained winds of 20 m / s or higher) and "gust_dur" (minutes of gust winds of 20 m / s or higher). If the Extended Best Tracks wind radii are used as the source of wind data, the "gust_dur" option cannot be selected.

• ← period needed



wind_limit	A numeric vector of length one giving the minimum wind speed (in meters per second) or duration of winds of 20 m / s or more (in minutes) to use in the filter. The units of this variable will depend on the user's choice for the wind_var parameter. If the Extended Best Tracks wind radii are used as the source of the wind data, the sustained winds will only be available for cutpoints of 34 knots, 50 knots, and 64 knots, so these values should be used (e.g., to get all counties with winds of 34 knots or higher, you could use wind_limit = 17.4, with the limit given as a value just below 34 knots in the units meters per second).
add_track	TRUE / FALSE of whether to add the storm's track to the map. The default is TRUE.
wind_source	A character string specifying the source to use for the winds. Options are "modeled", for estimates based on running a wind model from Best Tracks data inputs, and "ext_tracks", for estimates based on the wind radii in the Extended Best Tracks data. See the help files for the datasets storm_winds and ext_tracks_wind in the hurricaneexposedata package for more details on each of these sources for wind estimates. For the gust wind estimates, these are based on applying a gust factor of 1.49 to the sustained wind estimates in both wind data sources.

Value

Plots a map showing whether eastern US counties were exposed or unexposed to a specific storm based on a wind criterion.

Examples

```
# Ensure that data package is available before running the example.
# If it is not, see the `hurricaneexposure` package vignette for details
# on installing the required data package.
if (requireNamespace("hurricaneexposedata", quietly = TRUE)) {

  map_wind_exposure(storm = "Beryl-1988", wind_limit = 15)
}
```

---

multi_county_distance	<i>Hurricane exposure by distance for communities</i>
-----------------------	---

---

Description

This function takes a dataframe with multi-county communities and returns a community-level dataframe of "exposed" storms, based on the shortest distance between the storm's track and the population-based centers of each county in the community.

Usage

```
multi_county_distance(communities, start_year, end_year, dist_limit)
```

Arguments

communities	A dataframe with the FIPS codes for all counties within each community. It must include columns with a column identifier (commun) and with the FIPS codes of counties included in each community (fips). See the example code.
start_year	Four-digit integer with first year to consider.
end_year	Four-digit integer with last year to consider.
dist_limit	Maximum distance, in kilometers, of how close the storm track must come to the county's population mean center to classify the county as "exposed" to the storm.

Value

Returns the same type dataframe as county\_distance, but with storms listed by community instead of county.

Examples

```
# Ensure that data package is available before running the example.
# If it is not, see the `hurricaneexposure` package vignette for details
# on installing the required data package.
if (requireNamespace("hurricaneexposuredata", quietly = TRUE)) {

  communities <- data.frame(community_name = c(rep("ny", 6), "no", "new"),
                             fips = c("36005", "36047", "36061",
                                       "36085", "36081", "36119",
                                       "22071", "51700"))

  distance_df <- multi_county_distance(communities = communities,
                                       start_year = 1995, end_year = 2005,
                                       dist_limit = 75)

}
```

---

multi_county_events	<i>Hurricane exposure by events for communities</i>
---------------------	---

---

Description

This function takes a dataframe with multi-county communities and returns a community-level dataframe of "exposed" storms, based on the type of the event.

Usage

```
multi_county_events(communities, start_year, end_year, event_type)
```

Arguments

communities	A dataframe with the FIPS codes for all counties within each community. It must include columns with a column identifier (commun) and with the FIPS codes of counties included in each community (fips). See the example code.
start_year	Four-digit integer with first year to consider.
end_year	Four-digit integer with last year to consider.
event_type	Character string with the type of event to use to identify county exposures. Options include "flood", "tornado", "wind", and "tropical_storm".

Value

Returns the same type dataframe as county\_events, but with storms listed by community instead of county.

Examples

```
# Ensure that data package is available before running the example.
# If it is not, see the 'hurricaneexposure' package vignette for details
# on installing the required data package.
if (requireNamespace("hurricaneexposedata", quietly = TRUE)) {

  communities <- data.frame(community_name = c(rep("ny", 6), "no", "new"),
                             fips = c("36005", "36047", "36061",
                                       "36085", "36081", "36119",
                                       "22071", "51700"))

  distance_df <- multi_county_events(communities = communities,
                                     start_year = 1995, end_year = 2005,
                                     event_type = "flood")

}
```

---

multi_county_rain	<i>Hurricane exposure by rain for communities</i>
-------------------	---

---

Description

This function takes a dataframe with multi-county communities (see example for the proper format) and returns a community-level dataframe of storms to which the community was exposed, based on the average distance between the storm's track and the population-based centers of each county in the community and the given threshold of rainfall, summed over the days included in the rainfall measurement.

Usage

```
multi_county_rain(
  communities,
  start_year,
  end_year,
```



```

    rain_limit,
    dist_limit,
    days_included = c(-2, -1, 0, 1)
)

```

### Arguments

communities	A dataframe with the FIPS codes for all counties within each community. It must include columns with a column identifier (commun) and with the FIPS codes of counties included in each community (fips). See the example code.
start_year	Four-digit integer with first year to consider.
end_year	Four-digit integer with last year to consider.
rain_limit	Minimum of rainfall, in millimeters, summed across the days selected to be included (days_included), that must fall in a county for the county to be classified as "exposed" to the storm.
dist_limit	Maximum distance, in kilometers, of how close the storm track must come to the county's population mean center to classify the county as "exposed" to the storm.
days_included	A numeric vector listing the days to include when calculating total precipitation. Negative numbers are days before the closest date of the storm to a county. For example, c(-1, 0, 1) would calculate rain for a county as the sum of the rainfall for the day before, the day of, and the day after the date when the storm center was closest to the county center. Values can range from -5 to 3 (i.e., at most, you can calculate the total rainfall from five days to three days after the day when the storm is closest to the county).

### Value

Returns a dataframe with a row for each county-storm pair and with columns for:

- commun: Each community's unique id. ← period needed
- storm\_id: Unique storm identifier with the storm name and year, separated by a hyphen(e.g., "Alberto-1988", "Katrina-2005"). ← period needed
- closest\_date: Date (based on local time) of the closest approach of the storm to the county's population mean center.
- local\_time: Local time of the closest approach of the storm to the county's population mean center, based on storm tracks linearly interpolated to 15-minute increments.
- closest\_time\_utc: Time, in UTC, of the closest approach of the storm to the county's population mean center, based on storm tracks linearly interpolated to 15-minute increments.
- mean\_dist: Average of the minimum distance (in kilometers) between the storm's track and the population mean centers of all the counties in the community.
- mean\_rain: Average of cumulative rainfall, in millimeters, in the counties in the community for the days selected using the days\_included option.
- min\_dist: The smallest minimum distance (in kilometers) between the storm's track and the population mean centers of any of the counties in the community.
- max\_rain: The maximum cumulative rainfall, in millimeters, in any of the counties in the community for the days selected using the days\_included option.

References

Al-Hamdan MZ, Crosson WL, Economou SA, Estes MG, Estes SM, Hemmings SN, Kent ST, Puckette M, Quattrochi DA, Rickman DL, Wade GM, McClure LA, 2014. Environmental public health applications using remotely sensed data. *Geocarto International* 29(1):85-98.

North America Land Data Assimilation System (NLDAS) Daily Precipitation years 1979-2011 on CDC WONDER Online Database, released 2012. <http://wonder.cdc.gov/wonder/help/Precipitation.html>

Rui H, Mocko D, 2014. README Document for North America Land Data Assimilation System Phase 2 (NLDAS-2) Products. Goddard Earth Sciences Data and Information Services Center.

Examples

```
# Ensure that data package is available before running the example.
# If it is not, see the 'hurricaneexposure' package vignette for details
# on installing the required data package.
if (requireNamespace("hurricaneexposuredata", quietly = TRUE)) {

  communities <- data.frame(community_name = c(rep("ny", 6), "no", "new"),
                             fips = c("36005", "36047", "36061",
                                       "36085", "36081", "36119",
                                       "22071", "51700"))

  rain_storm_df <- multi_county_rain(communities = communities,
                                    start_year = 1995, end_year = 2005,
                                    rain_limit = 100, dist_limit = 100)

}
```

---

multi_county_wind	<i>Hurricane exposure by wind for communities</i>
-------------------	---

---

Description

This function takes a dataframe with multi-county communities and returns a community-level dataframe of "exposed" storms, based on the highest of the maximum sustained wind speed for each county in the community.

Usage

```
multi_county_wind(communities, start_year, end_year, wind_limit)
```

Arguments

- |             |  |
|-------------|--|
| communities | A dataframe with the FIPS codes for all counties within each community. It must include columns with a column identifier (commun) and with the FIPS codes of counties included in each community (fips). See the example code. |
| start_year  | Four-digit integer with first year to consider.  |
| end_year    | Four-digit integer with last year to consider.   |

**wind\_limit**      A numeric vector of length one giving the minimum wind speed (in meters per second) or duration of winds of 20 m / s or more (in minutes) to use in the filter. The units of this variable will depend on the user’s choice for the `wind_var` parameter. If the Extended Best Tracks wind radii are used as the source of the wind data, the sustained winds will only be available for cutpoints of 34 knots, 50 knots, and 64 knots, so these values should be used (e.g., to get all counties with winds of 34 knots or higher, you could use `wind_limit = 17.4`, with the limit given as a value just below 34 knots in the units meters per second).

**Value**

Returns the same type dataframe as `county_rain`, but with storms listed by community instead of county.

**Note**

This function currently will only input a threshold for the sustained wind metric. If you would like to use gust winds or duration of winds, you will need to use the `county_wind` function to pull storms and aggregate to the multi-county community level yourself.

**Examples**

```
# Ensure that data package is available before running the example.
# If it is not, see the `hurricaneexposure` package vignette for details
# on installing the required data package.
if (requireNamespace("hurricaneexposuredata", quietly = TRUE)) {

  communities <- data.frame(community_name = c(rep("ny", 6), "no", "new"),
                             fips = c("36005", "36047", "36061",
                                       "36085", "36081", "36119",
                                       "22071", "51700"))

  wind_df <- multi_county_wind(communities = communities,
                              start_year = 1988, end_year = 2005,
                              wind_limit = 20)

}
```

---

rain_exposure	Write storm rain exposure files
---------------	---------------------------------

---

**Description**

This function takes an input of locations (either a vector of county FIPS or a dataframe of multi-county FIPS, with all FIPS listed for each county; see examples) and creates time series dataframes with the dates and exposures for all storms meeting the given rainfall and storm distance criteria. These exposure time series can then be merged with other time series (e.g., community-specific daily counts of health outcomes).



**Usage**

```
rain_exposure(
  locations,
  start_year,
  end_year,
  rain_limit,
  dist_limit,
  days_included = c(-2, -1, 0, 1),
  out_dir,
  out_type = "csv"
)
```

**Arguments**

locations	Either a vector of FIPS county codes, for county-level output, or a dataframe with columns for community identifier (commun) and associated FIPS codes (fips), for multi-county community output. See the examples for the proper format for this argument.
start_year	Four-digit integer with first year to consider.
end_year	Four-digit integer with last year to consider.
rain_limit	Minimum <sup>amount</sup> of rainfall, in millimeters, summed across the days selected to be included (days_included), that must fall in a county for the county to be classified as "exposed" to the storm.
dist_limit	Maximum distance, in kilometers, of how close the storm track must come to the county's population mean center to classify the county as "exposed" to the storm.
days_included	A numeric vector listing the days to include when calculating total precipitation. Negative numbers are days before the closest date of the storm to a county. For example, c(-1, 0, 1) would calculate rain for a county as the sum of the rainfall for the day before, the day of, and the day after the date when the storm center was closest to the county center. Values can range from -5 to 3 (i.e., at most, you can calculate the total rainfall from five days to three days after the day when the storm is closest to the county). <sup>✓ 1 or 2 words?</sup>
out_dir	Character string giving the <u>pathname</u> of the directory in which to write output. This directory should already exist on your computer.
out_type	Character string giving the type of output files you'd like. Options are "csv" (default) and "rds".

**Value**

This function writes out rain exposure files for each county or community indicated to the specified output directory (out\_dir). For more details on the columns in the output files, see the documentation for county\_rain and multi\_county\_rain.