Package 'stormwindmodel'

October 15, 2018

Type Package

Title Model Tropical Cyclone Wind Speeds

Version 0.1.1

Date 2018-10-14

Encoding UTF-8

Description Allows users to input tracking data for a hurricane or other tropical storm, along with a data frame of grid points at which to model wind speeds. Functions in this package will then calculate wind speeds at each point based on wind model equations. This modeling framework is currently set up to model winds for North American locations with Atlantic basin storms. This work was supported in part by grants from the National Institute of Environmental Health Sciences (R00ES022631), the National Science Foundation (1331399), and the Department of Energy (DE-FG02-08ER64644).

URL https://github.com/geanders/stormwindmodel

BugReports https://github.com/geanders/stormwindmodel/issues

License GPL (>= 2)

Imports dplyr (>= 0.7.6), ggplot2 (>= 3.0.0), lubridate (>= 1.7.4), maps (>= 3.3.0), plyr (>= 1.8.4), stringr (>= 1.3.1), tidyr (>= 0.8.1), weathermetrics (>= 1.2.2)

LazyData TRUE

RoxygenNote 6.1.0

Depends R (>= 2.10)

Suggests gridExtra (>= 2.3.0), knitr (>= 1.20.0), mapproj (>= 1.2.6), rgeos (>= 0.3.28), rmarkdown (>= 1.10.0), sf (>= 0.6.3), sp (>= 1.3.1), tigris (>= 0.7.0), viridis (>= 0.5.1)

VignetteBuilder knitr

NeedsCompilation no

Author Brooke Anderson [aut, cre], Andrea Schumacher [aut],

Seth Guikema [aut],
Steven Quiring [aut],
Joshua Ferreri [aut],
Andrea Staid [ctb],
Michael Guo [ctb],
Lei Ming [ctb],
Laiyin Zhu [ctb]

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

Repository CRAN

Date/Publication 2018-10-15 05:20:03 UTC

R topics documented:

`add_forward_speed` *Adds forward speed component to modeled surface wind*

Description

Adds the storm's forward speed component (i.e., motion asymmetery) back into the estimated surface wind speed at a grid point location after rotational winds have been modeled for the location.

Usage

```
add_forward_speed(wind_sfc_sym, tcspd_u, tcspd_v, swd, cdist, Rmax)
```

Arguments

`wind_sfc_sym`	A numeric vector with maximum 10-meter 1-minute sustained wind with motion asymmetry removed (m / s).
`tcspd_u`	A numeric vector with the tropical cyclone speed, u-component (m / s).
`tcspd_v`	A numeric vector with the tropical cyclone speed, v-component (m / s).
`swd`	A numeric vector with surface wind direction (in degrees).
`cdist`	A numeric vector giving radius (in kilometers) from the storm center to the location being modeled.
`Rmax`	A numeric vector giving the radius to maximum winds (in kilometers) for the tropical storm.

Details

This function uses equation 12 from Phadke et al. (2003).

Value

A numeric vector giving asymmeric surface windspeed (m / s) at the location being modeled.

References

Phadke AC, Martino CD, Cheung KF, and Houston SH. 2003. Modeling of tropical cyclone winds and waves for emergency management. Ocean Engineering 30(4):553-578.

`add_inflow` *Add inflow angle*

Description

This function adds an inflow angle to the angle of the wind direction. It calculates an inflow angle as a function of the distance from the storm center to a location (Phadke et al. 2003), and then adds 20 degrees to this inflow angle to account for the location being over land rather than over water.

Usage

```
add_inflow(gwd, cdist, Rmax)
```

Arguments

`gwd`	A numeric vector giving direction of gradient wind at a location, in degrees. Due east is 0 degrees, due north 90 degrees, etc.
`cdist`	A numeric vector giving radius (in kilometers) from the storm center to the location being modeled.
`Rmax`	A numeric vector giving the radius to maximum winds (in kilometers) for the tropical storm.

Details

This function uses equations 11a-c from Phadke et al. (2003).

Value

Numeric vector with the gradient wind direction (in degrees), adjusted with an inflow angle appropriate for being over land and for the location's distance from the storm's center.

Note

This function is only appropriate for modeling wind speeds for locations that are over land.

References

Phadke AC, Martino CD, Cheung KF, and Houston SH. 2003. Modeling of tropical cyclone winds and waves for emergency management. Ocean Engineering 30(4):553-578.

Examples

```
add_inflow(gwd = 160, cdist = 100, Rmax = 20)
```

add_storm_track *Plot Atlantic basin hurricane tracks*

Description

Plot the tracks of a selected tropical storm to a map of modeled wind speed.

Usage

```
add_storm_track(storm_tracks, plot_object, plot_points = FALSE,
  alpha = 1, color = "firebrick")
```

Arguments

storm_tracks	A data frame with best tracks data for the storm track you would like to add. See the example floyd_tracks data for an example of the required format. This dataset must include columns for date (date-time of the track observation), latitude, and longitude.
plot_object	NULL or the name of a ggplot object to use as the underlying plot object (e.g., the output from a call to map_wind)
plot_points	TRUE / FALSE indicator of whether to include points, as well as lines, when plotting the hurricane tracks.
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

A ggplot object that includes a line with the track of a given tropical storm. This object can be printed directly or added on to with other ggplot commands.

Examples

```
## Not run:
library(ggplot2)
data("county_points")
data("floyd_tracks")
grid_winds_floyd <- get_grid_winds(hurr_track = floyd_tracks,
                                   grid_df = county_points)
floyd_map <- map_wind(grid_winds_floyd, value = "vmax_sust",
                      wind_metric = "knots") +
             ggtitle("Maximum sustained wind speeds")
add_storm_track(floyd_tracks, plot_object = floyd_map)

## End(Not run)
```

`add_wind_radii` *Add Willoughby inputs and parameters*

Description

Calculates a number of inputs and parameters needed for the Willoughby model for each observation of a storm track. This function intakes an inputed storm track and calculates these parameters for each observation in the storm tracks.These inputs and parameters are later used to model wind speed at each grid point for every observation.

Usage

```
add_wind_radii(full_track = create_full_track())
```

Arguments

`full_track` A dataframe with interpolated hurricane track data, as created by `create_full_track`. This dataframe must have the following columns:

- `date:` Date-time, in POSIXct format and UTC time zone
- `tclat:` Latitude (decimal degree)
- `tclon:` Longitude (decimal degrees), expressed as positive values (this model assumes all longitudes will be in the Western hemisphere)
- `vmax:` Maximum 10-meter sustained wind speed (m / s)

Value

The input dataset, but with columns added for the Willoughby parameters for every observations. Added columns include:

- `tcdir:` The bearing of the storm, in degrees, with 0 degrees indicating the storm is moving due east, 90 degrees indicating the storm is moving due north, etc.
- `tcspd_u, tcspd_v:` The u- and v-components of the forward speed of the storm, in meters per second
- `vmax_gl:` The maximum gradient-level 1-minute sustained wind, in m / s
- `Rmax:` The radius to maximum winds, in kilometers
- `X1, n, A, R1, R2:` Parameters needed for the Willoughby wind model

References

Willoughby HE, Darling RWR, and Rahn ME. 2006. Parametric representation of the primary hurricane vortex. Part II: A new family of sectionally continuous profiles. Monthly Weather Review 134(4):1102-1120.

Examples

```
## Not run:
data("floyd_tracks")
full_track <- create_full_track(hurr_track = floyd_tracks)
with_wind_radii <- add_wind_radii(full_track = full_track)
head(with_wind_radii)

## End(Not run)
```

`calc_and_summarize_grid_wind` *Calculate and summarize grid winds*

Description

This function combines the `calc_grid_wind` and `summarize_grid_wind` functions so they can be run jointly in the overall `get_grid_winds` function. This function calculates wind characteristics at just one location. Within the package, the function used within another function (`get_grid_winds`) to model wind speeds at all grid locations.

Usage

```
calc_and_summarize_grid_wind(grid_point = stormwindmodel::county_points[1,
], with_wind_radii = add_wind_radii(), tint = 0.25,
gust_duration_cut = 20, sust_duration_cut = 20)
```

Arguments

`grid_point`	A one-row dataframe with the grid id, latitude, and longitude for a single location for which you want to model winds.
`with_wind_radii`	A dataframe of storm tracks, including inputs and parameters for the Willoughby wind model, as created by `add_wind_radii`.
`tint`	Interval (in hours) to which to interpolate the tracks. The default is 0.25 (i.e., 15 minutes)
`gust_duration_cut`	The wind speed, in meters per second, to use as a cutoff point for determining the duration of gust winds. The function will calculate the minutes during the storm when surface-level gust winds were above this speed at the location.
`sust_duration_cut`	The wind speed, in meters per second, to use as a cutoff point for determining the duration of gust winds. The function will calculate the minutes during the storm when surface-level gust winds were above this speed at the location.

Value

Returns a one-row matrix with wind characteristics for a single location. The wind characteristics given are:

- vmax_gust: Maximum value of surface-level (10 meters) gust winds, in meters per second, over the length of the storm at the given location
- vmax_sust: Maximum value of surface-level (10 meters) sustained winds, in meters per second, over the length of the storm at the given location
- gust_dur: Length of time, in minutes, that surface-level gust winds were above a specified speed (default is 20 m / s)
- sust_dur: Length of time, in minutes, that surface-level sustained winds were above a specified speed (default is 20 m / s)

Examples

```
## Not run:
library(dplyr)
data(county_points)
data("floyd_tracks")
full_track <- create_full_track(hurr_track = floyd_tracks, tint = 0.25)
with_wind_radii <- add_wind_radii(full_track = full_track)
grid_point <- county_points %>% filter(gridid == "37055")
grid_wind_summary <- calc_and_summarize_grid_wind(grid_point = grid_point,
   with_wind_radii = with_wind_radii, gust_duration_cut = 15,
   sust_duration_cut = 15)

## End(Not run)
```

---

calc_bearing *Calculate bearing from one location to another*

Description

Calculates the bearing of a second location, as seen from the first location, based on latitude and longitude coordinates for both locations.

Usage

```
calc_bearing(tclat_1, tclon_1, tclat_2, tclon_2)
```

Arguments

tclat_1	A numeric vector giving latitude of the first location (degrees)
tclon_1	A numeric vector giving longitude of the first location (degrees). This value should be expressed as a positive value for Western hemisphere longitudes.
tclat_2	A numeric vector giving latitude of the second location (degrees)
tclon_2	A numeric vector giving longitude of the second location (degrees). This value should be expressed as a positive value for Western hemisphere longitudes.

Details

This function uses the following equations to calculate the bearing from one latitude-longitude pair to another:

$$S = cos(\phi_2) * sin(L_1 - L_2)$$

$$C = cos(\phi_1) * sin(\phi_2) - sin(\phi_1) * cos(\phi_2) * cos(L_1 - L_2)$$

$$\theta = atan2(S, C) * \frac{180}{\pi} + 90$$

where:

- ϕ_1: Latitude of first location, in radians
- L_1: Longitude of first location, in radians
- ϕ_2: Latitude of second location, in radians
- L_2: Longitude of second location, in radians
- S, C: Intermediary results
- θ: Direction of the storm movement, in degrees

In cases where this equation results in values below 0 degrees or above 360 degrees, the function applies modular arithmetic to bring the value back within the 0–360 degree range.

Value

A numeric vector giving the direction of the second location from the first location, in degrees. A direction of 0 degrees indicates the second location is due east of the first, 90 degrees indicates the second location is due north of the first, etc.

`calc_forward_speed` *Calculate storm's forward speed*

Description

This storm takes two storm locations and their observations times and calculates the average speed of the storm between the two observations.

Usage

```
calc_forward_speed(tclat_1, tclon_1, time_1, tclat_2, tclon_2, time_2)
```

Arguments

tclat_1	A numeric vector giving latitude of the first location (degrees)
tclon_1	A numeric vector giving longitude of the first location (degrees). This value should be expressed as a positive value for Western hemisphere longitudes.
time_1	A date-time vector giving the time of the first observation.
tclat_2	A numeric vector giving latitude of the second location (degrees)
tclon_2	A numeric vector giving longitude of the second location (degrees). This value should be expressed as a positive value for Western hemisphere longitudes.
time_2	A date-time vector giving the time of the second observation.

Value

A numeric vector with the average forward speed of the storm between the two observations, in meters per second.

---

calc_gradient_speed *Convert symmetric surface wind to gradient wind.*

Description

Converts maximum 10-m 1-minute symmetric sustained wind speed to gradient wind speed. The conversion factor depends on whether the storm is over land or water.

Usage

```
calc_gradient_speed(vmax_sfc_sym, over_land)
```

Arguments

vmax_sfc_sym	A numeric vector of 1-minute sustained wind speed at 10 meters, with motion asymmetry removed (m / s).
over_land	TRUE / FALSE of whether the storm is over land (TRUE) or water (FALSE).

Details

This function uses the following conversion:

$$V_{max,G} = \frac{V_{max,sym}}{f_r}$$

where:

- $V_{max,G}$: Max gradient-level 1-min sustained wind (m / s)
- $V_{max,sym}$: Max 10-m 1-min sustained wind with motion asymmetry removed (m / s)
- f_r: Reduction factor (see below)

The function uses a reduction factor based on Figure 3 in Knaff et al., 2011. If over water and within 100 kilometers of the storm's center, the ratio of gradient wind speed to surface wind speed is assumed to be 0.90. If over land, this reduction factor is reduced by 20% ($0.9 * 0.8 = 0.72$).

For this calculation, we assume that the radius of maximum wind for all storms is 100 kilometers or smaller (in the code, we do not calculate R_{max} until after we estimate gradient wind speed from surface wind speed, so we don't have that storm-specific estimate to use here).

Value

A numeric vector with maximum gradient-level 1-min wind speed (m / s).

References

Knaff JA, DeMaria M, Molenar DA, Sampson CR, and Seybold MG. 2011. An automated, objective, multiple-satellite-platform tropical cyclone surface wind speed analysis. Journal of Applied Meteorology and Climatology 50(10):2149-2166

`calc_grid_wind` *Calculate wind speed at grid points*

Description

Uses the Willoughby wind model (Willoughby et al. 2006) to calculate wind speed at a grid point location. This function calculates a wind time series at just one location.

Usage

```
calc_grid_wind(grid_point = stormwindmodel::county_points[1, ],
  with_wind_radii = add_wind_radii())
```

Arguments

`grid_point`	A one-row dataframe with the grid id, latitude, and longitude for a single location for which you want to model winds.
`with_wind_radii`	A dataframe of storm tracks, including inputs and parameters for the Willoughby wind model, as created by `add_wind_radii`.

Value

A dataframe with date (`date`) and modeled wind speed (`windspeed`, in m / s) at the grid point location for all storm observations.

References

Knaff JA, DeMaria M, Molenar DA, Sampson CR, and Seybold MG. 2011. An automated, objective, multiple-satellite-platform tropical cyclone surface wind speed analysis. Journal of Applied Meteorology and Climatology 50(10):2149-2166

Phadke AC, Martino CD, Cheung KF, and Houston SH. 2003. Modeling of tropical cyclone winds and waves for emergency management. Ocean Engineering 30(4):553-578.

Willoughby HE, Darling RWR, and Rahn ME. 2006. Parametric representation of the primary hurricane vortex. Part II: A new family of sectionally continuous profiles. Monthly Weather Review 134(4):1102-1120.

Examples

```
## Not run:
data("floyd_tracks")
data("county_points")
full_track <- create_full_track(hurr_track = floyd_tracks)
with_wind_radii <- add_wind_radii(full_track = full_track)
wind_grid <- calc_grid_wind(grid_point = county_points[1, ],
                            with_wind_radii = with_wind_radii)
head(wind_grid)

## End(Not run)
```

`calc_R1` *Calculate radius to start of transition region*

Description

Once you've solved for ξ, this function uses this value and the estimated R_{max} to determine $R1$, the radius from the storm center to the start of the transition region.

Usage

```
calc_R1(Rmax, xi)
```

Arguments

`Rmax`	Numeric vector of the radius at which the maximum wind occurs, in kilometers
`xi`	A numeric value with the ξ value determined by `solve_for_xi`

Details

This function is calculating the equation:

$$R_1 = R_{max} - \xi(R_2 - R_1)$$

where:

- R_1: A parameter for the Willoughby wind model (radius to start of transition region)
- R_{max}: Radius (in kilometers) to highest winds
- $R_2 - R_1$: Width of the transition region. This is assumed to be 25 kilometers if R_{max} is greater than 20 kilometers and 15 kilometers otherwise.

Value

A numeric vector with the estimated value of R_1, a parameter required for the Willoughby wind model.

References

Willoughby HE, Darling RWR, and Rahn ME. 2006. Parametric representation of the primary hurricane vortex. Part II: A new family of sectionally continuous profiles. Monthly Weather Review 134(4):1102-1120.

---

check_over_land *Determine if storm is over land or water*

Description

Determines if the storm is over land or water at its observed location. This function finds the closest grid point in the `landmask` dataframe, then checks if that grid point is over land or water.

Usage

```
check_over_land(tclat, tclon)
```

Arguments

`tclat`	Numeric vector of the absolute value of latitude, in degrees.
`tclon`	Numeric vector of the absolute value of latitude, in degrees.

Value

A logical vector of whether the storm is over land (TRUE) or water (FALSE)

county_points *Eastern U.S. county latitude and longitudes*

Description

A dataframe containing locations of population mean centers for counties in the eastern United States. Each county is identified by its 5-digit Federal Information Processing Standard (FIPS) code. This dataframe can be used to model storm winds at each county center. This dataset was put together using a dataframe from the U.S. Census Bureau, which was pulled from the website listed in "Source".

Usage

```
county_points
```

Format

A dataframe with 2,396 rows and 3 variables:

fips A character vector giving the county's five-digit Federal Information Processing Standard (FIPS) code

glat A numeric vector giving the latitude of the population mean center of each county

glon A numeric vector giving the longitude of the population mean center of each county

Source

http://www2.census.gov/geo/docs/reference/cenpop2010/county/CenPop2010_Mean_CO.txt

create_full_track *Impute hurricane tracks to finer time scale*

Description

Inputs data on a hurricane's track and imputes to a finer time resolution. For example, if the hurricane tracks are recorded at 6-hour intervals, this could be used to impute locations and windspeeds at 15-minute intervals. This function also does some reformatting necessary for later functions in the `stormwindmodel` package.

Usage

```
create_full_track(hurr_track = stormwindmodel::floyd_tracks,
  tint = 0.25)
```

Arguments

hurr_track	Dataframe with hurricane track data for a single storm. The dataframe must include columns for date-time (year, month, day, hour, minute; e.g., "198808051800" for August 5, 1988, 18:00 UTC), latitude, longitude, and wind speed (in knots). The column names for each of these must be `date`, `latitude`, `longitude`, and `wind`. See the example `floyd_tracks` dataset for an example of the required format.
tint	Interval (in hours) to which to interpolate the tracks. The default is 0.25 (i.e., 15 minutes)

Details

The function uses natural cubic splines for interpolation, both for location and for wind speed. Degrees of freedom are based on the number of available observations for the storm.

Value

A version of the storm's track data with latitude, longitude, and wind speed interpolated between observed values. Also, wind speed is converted in this function to m / s and the absolute value of the latitude is taken (necessary for further wind speed calculations). Finally, the names of some columns are changed (`tclat` for latitude, `tclon` for longitude, and `vmax` for wind speed.)

Note

This function imputes between each original data point, and it starts by determing the difference in time between each pair of data points. Because of this, the function can handle data that includes a point that is not at one of the four daily synoptic times (00:00, 06:00, 12:00, and 18:00). Typically, the only time hurricane observations are given outside of synoptic times for best tracks data is at landfall.

After imputing the tracks, longitude is expressed as a positive number. This is so the output will work correctly in later functions to fit the wind model. However, be aware that you should use the negative value of longitude for mapping tracks from the output from this function.

Examples

```
data("floyd_tracks")
full_track <- create_full_track(hurr_track = floyd_tracks)

# Interpolate to every half hour (instead of default 15 minutes)
full_track <- create_full_track(hurr_track = floyd_tracks, tint = 0.5)
```

degrees_to_radians *Convert from degrees to radians*

Description

Convert an angle from degrees to radians.

Usage

```
degrees_to_radians(degrees)
```

Arguments

degrees	A numeric vector with measurements in degrees.

Value

A numeric vector with measurement in radians.

floyd_tracks *Hurricane Floyd tracks data*

Description

A dataframe containing hurricane best tracks for Hurricane Floyd in 1999. This dataframe is included for use as an example hurricane tracks dataframe in the package documentation. This data originally came from the Extended Best Track dataset: http://rammb.cira.colostate.edu/research/tropical_cyclones/tc_extended_best_track_dataset/

Usage

```
floyd_tracks
```

Format

A dataframe with 48 rows and 4 variables:

date A character string giving the date and time of the observation

latitude A numeric vector giving the storm's latitude at that observation time

longitude A numeric vector giving the storm's longitude at that observation time

wind A numeric vector giving the estimated maximum sustained wind of that storm at that observation time, in knots

Source

http://rammb.cira.colostate.edu/research/tropical_cyclones/tc_extended_best_track_dataset/

`get_grid_winds` *Determine hurricane winds at locations*

Description

This function inputs a storm track and a dataset of locations and calculates highest wind speeds (sustained and maximum) and duration of winds above a certain speed at each location. The dataset of locations can either be a regularly-spaced grid or can be the central points of locations, like counties or cities. For counties in the eastern half of the United States, the `county_points` dataset that comes with the package can be used as the `grid_point` input.

Usage

```
get_grid_winds(hurr_track = stormwindmodel::floyd_tracks,
  grid_df = stormwindmodel::county_points, tint = 0.25,
  gust_duration_cut = 20, sust_duration_cut = 20)
```

Arguments

`hurr_track`	Dataframe with hurricane track data for a single storm. The dataframe must include columns for date-time (year, month, day, hour, minute; e.g., "198808051800" for August 5, 1988, 18:00 UTC), latitude, longitude, and wind speed (in knots). The column names for each of these must be `date`, `latitude`, `longitude`, and `wind`. See the example `floyd_tracks` dataset for an example of the required format.
`grid_df`	A dataframe of locations at which to calculate wind characteristics. This dataframe must include columns for latitude and longitude for each point, and these columns must be named `glat` and `glon`. The latitudes and longitudes should be in decimal degrees, with longitudes in the Western hemisphere (so, almost all those for Atlantic basin storms) expressed as negative values.
`tint`	Interval (in hours) to which to interpolate the tracks. The default is 0.25 (i.e., 15 minutes)
`gust_duration_cut`	The wind speed, in meters per second, to use as a cutoff point for determining the duration of gust winds. The function will calculate the minutes during the storm when surface-level gust winds were above this speed at the location.
`sust_duration_cut`	The wind speed, in meters per second, to use as a cutoff point for determining the duration of gust winds. The function will calculate the minutes during the storm when surface-level gust winds were above this speed at the location.

Value

The dataframe of locations input, with the following columns of wind characteristics added for each location:

- vmax_gust: Maximum value of surface-level (10 meters) gust winds, in meters per second, over the length of the storm at the given location
- vmax_sust: Maximum value of surface-level (10 meters) sustained winds, in meters per second, over the length of the storm at the given location
- gust_duration: Length of time, in minutes, that surface-level gust winds were above a specified value (default is 20 m / s)
- sust_duration: Length of time, in minutes, that surface-level sustained winds were above a specified value (default is 20 m / s)

Note

This function can take a few minutes to run, depending on the number of locations that are being modeled.

Examples

```
## Not run:
data("floyd_tracks")
data("county_points")
grid_winds <- get_grid_winds(hurr_track = floyd_tracks,
                             grid_df = county_points)

## End(Not run)
```

---

gradient_to_surface *Calculate surface wind speed from gradient*

Description

Calculates the surface wind speed based on an estimated gradient wind speed at a point and the radius from the storm center to the grid point.

Usage

```
gradient_to_surface(wind_gl_aa, cdist)
```

Arguments

wind_gl_aa	A numerical value with estimated gradient-level wind speed (m / s) at a grid point.
cdist	A numerical value with radius from the storm center to the grid point, in kilometers.

Details

The reduction factor is based on Figure 3 of Knaff et al., 2003. Over water, it is estimated to be 0.9 up to a radius of 100 km, 0.75 for a radius of 700 km or more, and decrease linearly between a radius of 100 km and 700 km. Points over land should use a reduction factor that is 20% lower. Because all of the counties are over land, the function makes this adjustment for all grid points.

Value

A numeric vector with the estimated symmetric surface wind speed at the grid point, in meters / second.

Note

This function is only appropriate to use for points that are over land.

References

Knaff JA, DeMaria M, Molenar DA, Sampson CR, and Seybold MG. 2011. An automated, objective, multiple-satellite-platform tropical cyclone surface wind speed analysis. Journal of Applied Meteorology and Climatology 50(10):2149-2166

`interp_track` *Interpolate a storm track*

Description

This function takes a wider-spaced storm track (e.g., every 6 hours) and interpolates to a finer interval (e.g., every 15 minutes). To do this, it fits GLMs of latitude and longitude regressed on natural cubic splines of date-time, and then predicts these splines to new intervals. These splines use degrees of freedom equal to the number of original observations divided by two.

Usage

```
interp_track(track, tint = 0.25)
```

Arguments

`track`	A dataframe with hurricane track data for a single storm. See the `floyd_tracks` dataset that comes with the package for an example of the required format for this dataframe.
`tint`	A numeric vector giving the time interval to impute to, in units of hours (e.g., 0.25, the default, interpolates to 15 minute-intervals).

Value

A dataframe with hurricane track data for a single storm, interpolated to the interval specified by `tint`.

katrina_tracks *Hurricane Katrina tracks data*

Description

A dataframe containing hurricane best tracks for Hurricane Katrina in 2005. This dataframe is included for use as an example hurricane tracks dataframe in the package documentation. This data originally came from the Extended Best Track dataset: http://rammb.cira.colostate.edu/research/tropical_cyclones/tc_extended_best_track_dataset/

Usage

```
katrina_tracks
```

Format

A dataframe with 48 rows and 4 variables:

date A character string giving the date and time of the observation

latitude A numeric vector giving the storm's latitude at that observation time

longitude A numeric vector giving the storm's longitude at that observation time

wind A numeric vector giving the estimated maximum sustained wind of that storm at that observation time, in knots

Source

http://rammb.cira.colostate.edu/research/tropical_cyclones/tc_extended_best_track_dataset/

landmask *Land-sea mask*

Description

A dataframe with gridded locations in the eastern United States and whether each point is land or water. This land-sea mask is used to identify whether hurricane center observations are more likely over land or water, so an appropriate conversion factor can be used to estimate gradient winds from sustained surface winds.

Usage

```
landmask
```

Format

A dataframe with 30,351 rows and 3 variables:

longitude A numeric vector with the longitude of the grid point

latitude A numeric vector with the latitude of the grid point

land A factor specifying whether that grid point is land or water

---

`latlon_to_km` *Calculate distance between two locations*

Description

This function takes latitudes and longitudes for two locations and calculates the distance (in meters) between the locations using the Haversine method.

Usage

```
latlon_to_km(tclat_1, tclon_1, tclat_2, tclon_2, Rearth = 6378.14)
```

Arguments

`tclat_1`	A numeric vector giving latitude of the first location (degrees)
`tclon_1`	A numeric vector giving longitude of the first location (degrees). This value should be expressed as a positive value for Western hemisphere longitudes.
`tclat_2`	A numeric vector giving latitude of the second location (degrees)
`tclon_2`	A numeric vector giving longitude of the second location (degrees). This value should be expressed as a positive value for Western hemisphere longitudes.
`Rearth`	Radius of the earth (km). Default is 6378.14 km.

Details

This function uses the Haversine method with great circle distance to calculate this distance. It is applying the following equations to determine distance (in kilometers) between two latitude-longitude pairs:

$$hav(\gamma) = hav(\phi_1 - \phi_2) + cos(\phi_1) * cos(\phi_2) * hav(L_1 - L_2)$$

where:

- ϕ_1: Latitude of first location, in radians
- ϕ_2: Latitude of second location, in radians
- L_1: Longitude of first location, in radians
- L_2: Longitude of second location, in radians
- $hav(\gamma)$: The haversine function, $hav(\gamma) = sin^2\left(\frac{\gamma}{2}\right)$
- R_earth: Radius of the earth, here assumed to be 6378.14 kilometers
- D: Distance between the two locations, in kilometers

Value

A vector with the distance between the two locations, in kilometers.

`map_wind` *Map wind exposure at the county level*

Description

Inputs a dataframe with modeled winds for each eastern U.S. county and maps these modeled winds.

Usage

```
map_wind(grid_winds, value = "vmax_sust", break_point = NULL,
  wind_metric = "mps")
```

Arguments

`grid_winds`	A dataframe that is the output of running `get_grid_winds` using eastern U.S. county centers as the grid point locations for modeling the winds.
`value`	A character string giving the value to plot. Possible options are `"vmax_gust"` (maximum gust wind speeds) and `"vmax_sust"` (maximum sustained wind speeds).
`break_point`	An numeric value giving the value of the `value` parameter (e.g.,maximum gust wind speeds or maximum sustained wind speeds) at which to break for a binary map showing exposure versus no exposure. The default for this parameter is `NULL`, which returns a map with continuous wind speed values. If the `break_point` argument is set to a numeric value, the function will return a map where counties are given binary classifications of "exposed" or "not exposed" based on whether modeleded wind speed for the county is above or below this break point.
`wind_metric`	A character vector with the wind metric to use for the map. Possible values are `"knots"` and `"mps"` (m / s, the default).

Value

This function returns a map of the `ggplot` class, plotting exposure to hurricane winds by county for the eastern half of the United States.

Examples

```
## Not run:
data("katrina_tracks")
data("county_points")
grid_winds_katrina <- get_grid_winds(hurr_track = katrina_tracks,
                                     grid_df = county_points)
map_wind(grid_winds_katrina)
map_wind(grid_winds_katrina, wind_metric = "knots")
map_wind(grid_winds_katrina, value = "vmax_gust")
```

```
map_wind(grid_winds_katrina, break_point = 20)

## End(Not run)
```

`radians_to_degrees` *Convert from radians to degrees*

Description

Convert from radians to degrees

Usage

```
radians_to_degrees(radians)
```

Arguments

`radians`	A numeric vector with measurements in radians.

Value

A numeric vector with the measurement in degrees.

`remove_forward_speed` *Remove forward speed from maximum wind speed*

Description

Removes the forward speed of the storm from the maximum storm wind speed, V_{max}, to estimate $V_{max,sym}$, the storm's maximum 10-m 1-min sustained wind with motion asymmetry removed.

Usage

```
remove_forward_speed(vmax, tcspd)
```

Arguments

`vmax`	A numeric vector giving maximum 10-m 1-minute sustained wind (m / s)
`tcspd`	A numeric vector giving the tropical cyclone's forward speed (m / s).

Details

This function is based on equation 12 (and accompanying text) in Phadke et al. 2003. Based on this paper, the correction factor for forward speed is at its highest at the radius of maximum winds, where it equals 0.5 times the forward wind speed.

Value

A numerical vector with $V_{max,sym}$, the storm's maximum 10-m 1-min sustained wind with motion asymmetry removed, in m / s.

References

Phadke AC, Martino CD, Cheung KF, and Houston SH. 2003. Modeling of tropical cyclone winds and waves for emergency management. Ocean Engineering 30(4):553-578.

`solve_for_xi` *Numerically solve Willoughby Eqn. 3 for xi*

Description

This function uses the Newton-Raphson method to solve equation 3 (the dual-exponential profile version) for ξ in Willoughby et al. (2006). This value of ξ can then be used to determine R_1 for that storm observation.

Usage

```
solve_for_xi(xi0 = 0.5, eq3_right, eps = 0.001, itmax = 100)
```

Arguments

`xi0`	A numeric value giving the starting guess for ξ
`eq3_right`	A numerical value with the right-hand side of Willoughby et al. (2006), Eqn. 3, the dual-exponential version. This value is calculated at each storm observation point using the `will3_right` function.
`eps`	The convergence threshold for determining if the algorithm has converged.
`itmax`	The maximum number of iterations to try before deciding that the algorithm did not converge.

Note

If this algorithm does not converge, the function returns a missing value for ξ.

References

Jones O, Maillardet R, and Robinson A. 2009. Introduction to Scientific Programming and Simulation Using R. Boca Raton, FL: Chapman & Hall/CRC Press.

Press WH, Teukolsky SA, Vetterling WT, and Flannery BP. 2002. Numerical Recipes in C++: The Art of Scientific Computing. 2nd ed. Cambridge, UK: Cambridge University Press.

Willoughby HE, Darling RWR, and Rahn ME. 2006. Parametric representation of the primary hurricane vortex. Part II: A new family of sectionally continuous profiles. Monthly Weather Review 134(4):1102-1120.

summarize_grid_wind *Generate wind summaries for grid point*

Description

Summarizes the wind time series for a single grid point, as created by calc_grid_wind.

Usage

```
summarize_grid_wind(grid_wind, tint = 0.25, gust_duration_cut = 20,
  sust_duration_cut = 20)
```

Arguments

grid_wind	A dataframe with a time series of modeled wind speeds at a location, as created by calc_grid_wind.
tint	Interval (in hours) to which to interpolate the tracks. The default is 0.25 (i.e., 15 minutes)
gust_duration_cut	The wind speed, in meters per second, to use as a cutoff point for determining the duration of gust winds. The function will calculate the minutes during the storm when surface-level gust winds were above this speed at the location.
sust_duration_cut	The wind speed, in meters per second, to use as a cutoff point for determining the duration of gust winds. The function will calculate the minutes during the storm when surface-level gust winds were above this speed at the location.

Value

Returns a one-row matrix with wind characteristics for a single location. The wind characteristics given are:

- vmax_gust: Maximum value of surface-level (10 meters) gust winds, in meters per second, over the length of the storm at the given location
- vmax_sust: Maximum value of surface-level (10 meters) sustained winds, in meters per second, over the length of the storm at the given location
- gust_duration: Length of time, in minutes, that surface-level gust winds were above a specified value (default is 20 meters per second)
- sust_duration: Length of time, in minutes, that surface-level sustained winds were above a specified value (default is 20 meters per second

`will1` *Model wind speed at a grid point for a storm track observation*

Description

Models the gradient wind speed at a certain radius from a storm's center. To do this, it uses different equations and subfunctions depending on how large the radius is (see details). This function requires, as inputs, Willoughby wind model parameters calculated using the `add_wind_radii` function.

Usage

```
will1(cdist, Rmax, R1, R2, vmax_gl, n, A, X1, X2 = 25)
```

Arguments

`cdist`	Distance (in km) from center of tropical cyclone to grid point.
`Rmax`	Numeric vector of the radius at which the maximum wind occurs, in kilometers
`R1`	A numeric vector of one of the parameters of the Willoughby model.
`R2`	A numeric vector of one of the parameters of the Willoughby model.
`vmax_gl`	Numeric vector of the tangential wind component of the maximum gradient wind speed, in meters per second
`n`	A numeric vector of one of the parameters of the Willoughby model.
`A`	A numeric vector of one of the parameters of the Willoughby model.
`X1`	A numeric vector of one of the parameters of the Willoughby model.
`X2`	A numeric vector of one of the parameters of the Willoughby model.

Details

If $r \leq R_1$, this function is calculating the equation:

$$V(r) = V_i = V_{max} \left(\frac{r}{R_{max}} \right)^n$$

where:

- $V(r)$: Maximum sustained gradient wind speed at a radius of r from the storm's center
- r: Radius from the storm center, in kilometers
- $V_{max,G}$: Maximum sustained gradient wind speed of the storm, in meters per second
- R_1: A parameter for the Willoughby wind model (radius to start of transition region)
- R_{max}: Radius (in kilometers) to highest winds
- n: A parameter for the Willoughby wind model

If $R_2 < r$, this function is calculating the equation:

$$V(r) = V_o = V_{max}\left[(1-A)e^{\frac{R_{max}-r}{X_1}} + Ae^{\frac{R_{max}-r}{X_2}}\right]$$

where:

- $V(r)$: Maximum sustained gradient wind speed at a radius of r kilometers from the storm's center
- r: Radius from the storm center, in kilometers
- $V_{max,G}$: Maximum sustained gradient wind speed of the storm, in meters per second
- R_{max}: Radius (in kilometers) to highest winds
- A, X_1, X_2: Parameters for the Willoughby wind model

If $R_1 < r \leq R_2$, this function uses the equations:

$$\xi = \frac{r - R_1}{R_2 - R_1}$$

and, if $0 \leq \xi <\leq 1$ (otherwise, $w = 0$):

$$w = 126\xi^5 - 420\xi^6 + 540\xi^7 - 315\xi^8 + 70\xi^9$$

and then:

$$V(r) = V_i(1-w) + V_o w, (R_1 \leq r \leq R_2)$$

where, for this series of equations:

- $V(r)$: Maximum sustained gradient wind speed at a radius of r kilometers from the storm's center
- r: Radius from the storm center, in kilometers
- w: Weighting variable
- R_1, R_2: Parameters for the Willoughby wind model

Value

Returns a numeric vector with gradient wind speed at a radius of r from the storm's center, in meters per second.

References

Willoughby HE, Darling RWR, and Rahn ME. 2006. Parametric representation of the primary hurricane vortex. Part II: A new family of sectionally continuous profiles. Monthly Weather Review 134(4):1102-1120.

`will10a` *Calculate X1 for Willoughby model*

Description

Calculates X_1, a parameter for the Willoughby wind model using equation 10a (Willoughby et al. 2006).

Usage

```
will10a(vmax_gl, tclat)
```

Arguments

`vmax_gl`	Numeric vector of the tangential wind component of the maximum gradient wind speed, in meters per second
`tclat`	Numeric vector of the absolute value of latitude, in degrees.

Details

This function uses the following equation (equation 10a, Willoughby et al. 2006) to calculate the X_1 parameter:

$$X_1 = 317.1 - 2.026V_{max,G} + 1.915\phi$$

where:

- X_1: Parameter for the Willoughby wind model
- $V_{max,G}$: Maximum gradient-level 1-min sustained wind (m / s)
- ϕ: Latitude, in decimal degrees

Value

A numeric vector giving one of the parameters (X_1) required for the Willoughby wind model.

References

Willoughby HE, Darling RWR, and Rahn ME. 2006. Parametric representation of the primary hurricane vortex. Part II: A new family of sectionally continuous profiles. Monthly Weather Review 134(4):1102-1120.

will10b *Calculate n for the Willoughby model*

Description

Calculates n, a parameter for the Willoughby model, using equation 10b (Willoughby et al. 2006).

Usage

```
will10b(vmax_gl, tclat)
```

Arguments

vmax_gl	Numeric vector of the tangential wind component of the maximum gradient wind speed, in meters per second
tclat	Numeric vector of the absolute value of latitude, in degrees.

Details

This function is calculating the equation:

$$n = 0.4067 + 0.0144 V_{max,G} - 0.0038\phi$$

where:

- n: Parameter for the Willoughby wind model
- $V_{max,G}$: Maximum gradient-level 1-min sustained wind (m / s)
- ϕ: Latitude, in decimal degrees

Value

A numeric vector for n, a parameter needed for the Willoughby wind model.

References

Willoughby HE, Darling RWR, and Rahn ME. 2006. Parametric representation of the primary hurricane vortex. Part II: A new family of sectionally continuous profiles. Monthly Weather Review 134(4):1102-1120.

`will10c` *Calculate A for Willoughby model*

Description

Calculates A, a paramter for the Willoughby wind model, using equation 10c (Willoughby et al. 2006).

Usage

```
will10c(vmax_gl, tclat)
```

Arguments

`vmax_gl`	Numeric vector of the tangential wind component of the maximum gradient wind speed, in meters per second
`tclat`	Numeric vector of the absolute value of latitude, in degrees.

Details

This function calculates A using equation 10c (Willoughby et al. 2006):

$$A = 0.0696 + 0.0049 V_{max,G} - 0.0064\phi$$

where:

- A: Parameter for the Willoughby wind model (any value of A calculated as negative is re-set to 0)
- $V_{max,G}$: Tangential component of the maximum gradient-level sustained wind speed (in m / s)
- ϕ: Latitude, in decimal degrees

Any negative values of A are reset to 0.

Value

A numeric vector that is a parameter required for the Willoughby model.

References

Willoughby HE, Darling RWR, and Rahn ME. 2006. Parametric representation of the primary hurricane vortex. Part II: A new family of sectionally continuous profiles. Monthly Weather Review 134(4):1102-1120.

will1a *Willoughby et al. (2006), Equation 1(a)*

Description

Willoughby et al. (2006), Equation 1(a)

Usage

```
will1a(vmax_gl, r, Rmax, n)
```

Arguments

vmax_gl	Numeric vector of the tangential wind component of the maximum gradient wind speed, in meters per second
r	Numeric vector of radius from the storm center to the point you are measuring, in kilometers
Rmax	Numeric vector of the radius at which the maximum wind occurs, in kilometers
n	A numeric vector of one of the parameters of the Willoughby model.

will2 *Willoughby et al. (2006), Equation 2*

Description

Calculates equation 2 from Willoughby et al. (2006).

Usage

```
will2(r, R1)
```

Arguments

r	Numeric vector of radius from the storm center to the point you are measuring, in kilometers
R1	Numeric vector of radius at start of transition zone, in kilometers

Value

A numeric vector of the weighting parameter for Willoughby's wind profile equations (w).

will3_deriv_func *Calculate the function value and derivative for Willoughby Eqn. 3*

Description

Calculates values of both the function and the derivative of the function for which you are trying to solve the root in Eqn. 3 (the version using the dual exponential profile) of Willoughby et al. (2006).

Usage

```
will3_deriv_func(xi, eq3_right)
```

Arguments

xi	A numerical value for ξ from Willoughby et al. (2006), Eqn. 2
eq3_right	A numerical value with the right-hand side of Willoughby et al. (2006), Eqn. 3, the dual-exponential version. This value is calculated at each storm observation point using the will3_right function.

Value

A numeric vector of length two. The first value is the calculated value of $f'(x)$ for $x = \xi$, while the second value is the calculated value of $f(x)$ for $x = \xi$. These two values are used in iterating through the Newton-Raphson method to determine ξ.

References

Willoughby HE, Darling RWR, and Rahn ME. 2006. Parametric representation of the primary hurricane vortex. Part II: A new family of sectionally continuous profiles. Monthly Weather Review 134(4):1102-1120.

will3_right *Calculate right-hand side of Willoughby Eqn. 3*

Description

Calculates the right hand side of the version of Eqn. 3 in Willoughby et al. (2006) with the dual exponential profile.

Usage

```
will3_right(n, A, X1, Rmax)
```

Arguments

n	A numeric vector of one of the parameters of the Willoughby model.
A	A numeric vector of one of the parameters of the Willoughby model.
X1	A numeric vector of one of the parameters of the Willoughby model.
Rmax	A numeric vector giving the radius to maximum winds (in kilometers) for the tropical storm.

Value

A numeric vector with the value for the right-hand side of Eqn. 3 in Willoughby et al. 2006, using the dual exponential version of that equation.

References

Willoughby HE, Darling RWR, and Rahn ME. 2006. Parametric representation of the primary hurricane vortex. Part II: A new family of sectionally continuous profiles. Monthly Weather Review 134(4):1102-1120.

---

will7a *Calculate radius of maximum winds*

Description

Calculates the radius at which the maximum wind occurs (R_{max}), based on the maximum gradient-level 1-min sustained wind ($V_{max,G}$) and latitude (ϕ). This function implements Willoughby et al. (2006), Equation 7a.

Usage

```
will7a(vmax_gl, tclat)
```

Arguments

vmax_gl	Numeric vector of the tangential wind component of the maximum gradient wind speed, in meters per second
tclat	Numeric vector of the absolute value of latitude, in degrees.

Details

This function fits the following equation:

$$R_{max} = 46.4e^{-0.0155V_{max,G}+0.0169\phi}$$

where:

- R_{max}: Radius from the storm center to the point at which the maximum wind occurs (km)
- $V_{max,G}$: Tangential wind component of the gradient-level maximum wind speed (m / s)
- ϕ: Latitude (degrees)

Value

A numeric vector with R_{max}, the radius of maximum winds, in kilometers.

References

Willoughby HE, Darling RWR, and Rahn ME. 2006. Parametric representation of the primary hurricane vortex. Part II: A new family of sectionally continuous profiles. Monthly Weather Review 134(4):1102-1120.

Index