Package 'StormR'

November 24, 2023

Title Analyzing the Behaviour of Wind Generated by Tropical Storms and Cyclones

Maintainer Thomas Arsouze <thomas.arsouze@cirad.fr>

Version 0.1.1

URL https://umr-amap.github.io/StormR/

BugReports https://github.com/umr-amap/StormR/issues/new/choose

Description Set of functions to quantify and map the behaviour of winds generated by tropical storms and cyclones in space and time. It includes functions to compute and analyze fields such as the maximum sustained wind field, power dissipation index and duration of exposure to winds above a given threshold. It also includes functions to map the trajectories as well as characteristics of the storms.

License GPL (>= 3)

Depends R (>= 2.10)

Imports graphics, maps, methods, ncdf4, rworldmap, sf, stringr, terra, utils, zoo

Suggests knitr, rmarkdown, rworldxtra, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

Encoding UTF-8

LazyData true

RoxygenNote 7.2.3

NeedsCompilation no

Author Baptiste Delaporte [aut],
Thomas Ibanez [aut] (<https://orcid.org/0000-0002-3192-1721>),
Gunnar Keppel [aut] (<https://orcid.org/0000-0001-7092-6149>),
Swen Jullien [aut] (<https://orcid.org/0000-0002-5389-0532>),
Christophe Menkes [aut] (<https://orcid.org/0000-0002-1457-9696>),
Thomas Arsouze [aut, cre] (<https://orcid.org/0000-0002-8871-6120>)

Repository CRAN

Date/Publication 2023-11-24 13:40:06 UTC

R topics documented:

`defStormsDataset` *Creating a* `stormsDataset` *object*

Description

The `defStormsDataset()` function creates a `stormsDataset` object from a NetCDF file. This is an essential first step before other `stormR` functions can be used.

Usage

```
defStormsDataset(
  filename = system.file("extdata", "test_dataset.nc", package = "StormR"),
 fields = c(names = "name", seasons = "season", isoTime = "iso_time", lon = "usa_lon",
    lat = "usa_lat", msw = "usa_wind", basin = "basin", sshs = "usa_sshs", rmw =
    "usa_rmw", pressure = "usa_pres", poci = "usa_poci"),
  basin = NULL,
  seasons = c(1980, as.numeric(format(Sys.time(), "%Y"))),
 unitConversion = c(msw = "knt2ms", rmw = "nm2km", pressure = "mb2pa", poci = "mb2pa"),
  verbose = 1
)
```

Arguments

`filename` character. Name of the NetCDF (.nc) file. Default is the `test_dataset.nc` file located in the `inst/extdata` repository of the directory (accessible by `system.file("extdata", "test_dataset.nc", package = "StormR")`). This test dataset is extracted from the IBTrACS.SP.v04r00.nc file and provides all the tropical cyclones that occurred around Vanuatu from 2015 to 2016 and around New Caledonia from 2020 to 2021.

`fields` named character vector. This argument allows to specify the corresponding variable names in the input NetCDF file for each field in the output `stormsDataset`. By default, the corresponding variable names are set up to import data from a NetCDF file from the IBTrACS database (Knapp et al., 2010). Corresponding variable names for following fields have to (mandatory fields) or can be (recommended or optional fields) provided:

- `"names"`, names of the storms (mandatory),
- `"seasons"`, years of observations (mandatory),
- `"isoTime"`, date and time of observations (mandatory),
- `"lon"`, longitude of the observations (mandatory),
- `"lat"`, latitude of the observations (mandatory),
- `"msw"`, maximum sustained wind speed (mandatory),
- `"basin"`, name of the area where the storm originated (recommended),
- `"rmw"`, radius of maximum winds: distance between the centre of the storm and its band of strongest winds (recommended),
- `"pressure"`, central pressure (recommended),
- `"poci"`, pressure of the last closed isobar (recommended), and
- `"sshs"`, Saffir-Simpson hurricane wind scale rating based on msw (optional).

`basin` character. If the basin field is provided, then storm track data will only be extracted for the named basin. By default `basin=NULL`, meaning that all storms irrespective of the basin they originated in are extracted. Seven basins can be used to filter the data set:

- `"NA"`, for North Atlantic basin,
- `"SA"`, for South Atlantic basin,
- `"EP"`, for Eastern North Pacific basin,
- `"WP"`, for Western North Pacific basin,
- `"SP"`, for South Pacific basin,
- `"SI"`, for South India basin, or
- `"NI"`, for North India basin.

`seasons` numeric vector. Seasons of occurrence of the storms (e.g., c(2020,2022)). In the Southern Hemisphere, the cyclone season extends across two consecutive years. Therefore, to capture the 2021 to 2022 cyclone season both years should be specified, with cyclones assigned for the year that originated in. By default all storms occurring since 1980 are extracted.

`unitConversion` named character vector. `StormR` functions use the metric system (international system of units), therefore `msw` has to be provided in $m.s^{-1}$, `rmw` in km, `pressure` and `poci` in Pa. By default `unitConversion=c(msw = "knt2ms", rmw = "nm2km", pressure = "mb2pa", poci = "mb2pa")` to meet the requirements when importing a NetCDF file from the IBTrACS database. This argument is mandatory even if no conversion is needed. If no conversion is needed then use `"None"` in the corresponding fields. The following unit conversions are implemented:

For `msw`,

- `"knt2ms"`, to convert knot to meter per second (default setting),
- `"kmh2ms"`, to convert kilometre per hour to meter per second,
- `"mph2ms"`, to convert miles per hour to meter per second, or
- `"None"`, if no conversion is needed.

For `rmw`,

- `"nm_to_ms"`to convert nautical miles to kilometre (default setting), or
- `"None"`if no conversion is needed.

For `pressure` and `poci`,

- `"mb2pa"`, to convert millibar to Pascal (default setting),
- `"b2pa"`, to convert bar to Pascal,
- `"atm2pa"`, to convert atmosphere to Pascal,
- `"psi2pa"`, to convert psi to Pascal, or
- `"None"`, if no conversion is needed.

`verbose` numeric. Whether the function should display (`= 1`) or not (`= 0`) information about the processes.

Value

The `defStormsDataset()` function returns a `stormsDataset` object.

References

Knapp, K. R., Kruk, M. C., Levinson, D. H., Diamond, H. J., & Neumann, C. J. (2010). The International Best Track Archive for Climate Stewardship (IBTrACS). Bulletin of the American Meteorological Society, 91(3), Article 3. https://doi.org/10.1175/2009bams2755.1

Examples

```
# Creating a `stormsDataset` object with storms between 2010 and 2015
# in the South Pacific using the NetCDF provided with the package
SP_2015_2020 <- defStormsDataset(seasons = c(2010, 2015))
str(SP_2015_2020)
```

`defStormsList` *Creating a* `stormsList` *object*

Description

The `defStormsList()` function extracts storm track data from a `stormsDataset` and creates a `stormsList` object based on specified arguments relating to location of interest, seasons, and names of the storms.

Usage

```
defStormsList(
  sds,
  loi,
  seasons = c(sds@seasons["min"], sds@seasons["max"]),
  names = NULL,
  maxDist = 300,
  removeTD = TRUE,
  verbose = 2
)
```

Arguments

`sds`	`stormsDataset` object.
`loi`	Location of interest. Can be defined using, • character, a country name (e.g., "Vanuatu") • character, a basin name among "NA", "SA", "EP", "WP", "SP", "SI" and "NI" • numeric vector, a point coordinate (lon, lat in decimal degrees, e.g., c(169.5, -19.2)) • sp (SpatialPolygon) or a sf (simple features) object (e.g., created from a shapefile)
`seasons`	numeric vector. Seasons of occurrence of the storms (e.g., c(2020,2022)). In the Southern Hemisphere, the cyclone season extends across two consecutive years. Therefore, to capture the 2021 to 2022 cyclone season both years should be specified, with cyclones assigned for the year that originated in. By default all storms from `sds` are extracted.
`names`	character vector. Names of specific storms (in capital letters).
`maxDist`	numeric. Maximum distance between the location of interest and the storm for which track data are extracted. Default `maxDist` is set to 300 km.
`removeTD`	logical. Whether (TRUE) or not (FALSE) removing tropical depressions ($msw < 18m.s^{-1}$, not considered to be stormed in Saffir-Simpson hurricane wind scale) are removed. Default value is set to `TRUE`.
`verbose`	numeric. Type of information the function displays. Can be:

- `2`, information about both the processes and the outputs are displayed (default value),
- `1`, only information about the processes are displayed, or
- `0`, nothing is displayed.

Details

The available countries for the `loi` are those provided in the `rwolrdxtra` package. This package provide high resolution vector country boundaries derived from Natural Earth data. More informations on the Natural Earth data here: http://www.naturalearthdata.com/downloads/10m-cultural-vectors/.

Value

The `defStormsList()` function returns a `stormsList` object containing track data for all storms meeting the specified criteria (e.g., name, season, location).

References

Knapp, K. R., Kruk, M. C., Levinson, D. H., Diamond, H. J., & Neumann, C. J. (2010). The International Best Track Archive for Climate Stewardship (IBTrACS). Bulletin of the American Meteorological Society, 91(3), Article 3. doi:10.1175/2009bams2755.1

Examples

```
#Creating a stormsDataset
sds <- defStormsDataset()

#Getting data using country names
vanuatu.st <- defStormsList(sds = sds, loi = "Vanuatu")

#Getting data using a specific point location
pt <- c(169, -19)
pam.pt <- defStormsList(sds = sds, loi = pt, names = "PAM")

#Getting data using country and storm names
niran.nc <- defStormsList(sds = sds, loi = "New Caledonia", names = c("NIRAN"))

#Getting data using a user defined spatial polygon
poly <- cbind(c(135, 290, 290, 135, 135),c(-60, -60, 0, 0, -60))
sp <- sf::st_polygon(list(poly))
sp <- sf::st_sfc(sp, crs = 4326)
sp <- sf::st_as_sf(sp)
sts_sp <- defStormsList(sds = sds, loi = sp)
```

`eezNC` *EEZ of New Caledonia*

Description

Provide the geographic limits of the eez of New Caledonia

Usage

```
eezNC
```

Format

`eezNC`**:**
a sf object

`getBuffer` *Getting the buffered location of interest*

Description

The `getBuffer()` function returns the buffered location of interest from a `stormsList` object.

Usage

```
getBuffer(sts)

## S4 method for signature 'stormsList'
getBuffer(sts)
```

Arguments

`sts`	`stormsList`

Value

A `sf` object.

Examples

```
#Creating a stormsDataset
sds <- defStormsDataset()

#Getting storm track data for all storms near New Caledonia
sts <- defStormsList(sds=sds, loi = "New Caledonia")

#Getting the buffered location of interest from the sts object
buff <- getBuffer(sts)
```

getBufferSize *Getting the buffer size*

Description

The `getBufferSize()` returns the buffer size used to generate the buffered location of interest for a `stormsList` object.

Usage

```
getBufferSize(sts)

## S4 method for signature 'stormsList'
getBufferSize(sts)
```

Arguments

`sts`	`stormsList`

Value

numeric.

Examples

```
#Creating a stormsDataset

sds <- defStormsDataset()

#Getting storm track data for all storms near New Caledonia
sts <- defStormsList(sds=sds, loi = "New Caledonia")

#Getting the buffer size from the sts object
buffsize <- getBufferSize(sts)
```

`getInObs` *Getting the number of the observations*

Description

The `getInObs()` function returns the number of the observations in a given `storm` or `stormsList` object.

Usage

```
getInObs(s, ...)

## S4 method for signature 'stormsList'
getInObs(s, name, season = NULL)

## S4 method for signature 'storm'
getInObs(s)
```

Arguments

`s`	`storm` or `stormsList` object.
`...`	extra argument for `stormsList`
`name`	character. Name of the storm in capital letters.
`season`	numeric. Cyclonic season of the `storm`. Required only if several `storm` in `s` object have the same name. Default value is set to `NULL`

Value

Numeric vector.

Examples

```
#Creating a stormsDataset

sds <- defStormsDataset()

#Getting storm track data for all storms near New Caledonia
sts <- defStormsList(sds=sds, loi = "New Caledonia")

#Getting the number of the observation for the tropical cyclone Niran in the sts object
getInObs(getStorm(sts, name = "NIRAN"))
getInObs(sts, name = "NIRAN")
```

getLOI *Getting the location of interest*

Description

The `getLOI()` functions returns the location of interest for the given `stormsList`.

Usage

```
getLOI(sts)

## S4 method for signature 'stormsList'
getLOI(sts)
```

Arguments

`sts`	`stormsList` object

Value

sf object.

Examples

```
#Creating a stormsDataset

sds <- defStormsDataset()

#Getting storm track data for all storms near New Caledonia
sts <- defStormsList(sds=sds, loi = "New Caledonia")

#Getting the location of interest for the sts object
loi <- getLOI(sts)
```

getNames *Getting the names of the storms*

Description

The `getNames()` function returns the names of the storms in a `storm` or a `stormsList` object.

Usage

```
getNames(s)

## S4 method for signature 'storm'
getNames(s)

## S4 method for signature 'stormsList'
getNames(s)
```

Arguments

`s`	`storm` or `stormsList` object.

Value

character vector.

Examples

```
#Creating a stormsDataset
sds <- defStormsDataset()

#Getting storm track data for all storms near New Caledonia
sts <- defStormsList(sds=sds, loi = "New Caledonia")

#Getting the names of the storms from the sts object
getNames(sts)
```

`getNbObs` *Getting the number of observations*

Description

The getNbObs() function returns the number of observations for a storm in a `storm` or `stormsList` object.

Usage

```
getNbObs(s, ...)

## S4 method for signature 'storm'
getNbObs(s)

## S4 method for signature 'stormsList'
getNbObs(s, name, season = NULL)
```

Arguments

`s`	`storm` or `stormsList` object.
`...`	extra arguments for `stormsList`
`name`	character. Name of the storm in capital letters.
`season`	numeric. Cyclonic season of the `storm`. Required only if several `storm` in the `s` have the same name. Default value is set to `NULL`.

Value

numeric.

Examples

```
#Creating a stormsDataset
sds <- defStormsDataset()

#Getting storm track data for all storms near New Caledonia
sts <- defStormsList(sds=sds, loi = "New Caledonia")

##Getting the number of observations for the tropical cyclone Niran in the sts object
getNbObs(getStorm(sts, name = "NIRAN"))
getNbObs(sts, name = "NIRAN")
```

`getNbStorms` *Getting the number of* `storm`

Description

The `getNbStorms()` returns the number of `storm` objects in the given `stormsList` object.

Usage

```
getNbStorms(sts)

## S4 method for signature 'stormsList'
getNbStorms(sts)
```

Arguments

`sts`	`stormsList`

Value

numeric, the number of `storm` objects.

Examples

```
#Creating a stormsDataset

sds <- defStormsDataset()

#Getting storm track data for all storms near New Caledonia
sts <- defStormsList(sds=sds, loi = "New Caledonia")

#Getting the number of storms in the sts object
getNbStorms(sts)
```

---

getObs *Getting observations*

Description

The `getObs()` function returns observed track data for a storm in a `storm` or `stormsList` object.

Usage

```
getObs(s, ...)

## S4 method for signature 'stormsList'
getObs(s, name, season = NULL)

## S4 method for signature 'storm'
getObs(s)
```

Arguments

`s`	`storm` or `stormsList` object
`...`	extra argument for `stormsList`
`name`	character. Name of the storm in capital letters.
`season`	numeric. Cyclonic season of the `storm`. Required only if several `storm` in the `s` object have the same name. Default value is set to `NULL`.

Value

A data.frame.

Examples

```
#Creating a stormsDataset
sds <- defStormsDataset()

#Getting storm track data for all storms near New Caledonia
sts <- defStormsList(sds=sds, loi = "New Caledonia")

#Getting the observed track data for the tropical
#cyclone Niran in the sts object
getObs(getStorm(sts, name = "NIRAN"))
getObs(sts, name = "NIRAN")
```

`getSeasons` *Getting cyclonic seasons of the storms*

Description

The `getSeasons()` function returns the cyclonic season of each storm in a `storm` or `stormsList` object.

Usage

```
getSeasons(s)

## S4 method for signature 'storm'
getSeasons(s)

## S4 method for signature 'stormsList'
getSeasons(s)
```

Arguments

`s`	`storm` or `stormsList` object.

Value

numeric vector.

Examples

```
#Creating a stormsDataset
sds <- defStormsDataset()

#Getting storm track data for all storms near New Caledonia
sts <- defStormsList(sds=sds, loi = "New Caledonia")
```

```
#Getting the cyclonic seasons of the storms from the sts object
getSeasons(sts)
```

getSSHS — *Getting maximum Saffir-Simpson hurricane wind scale category*

Description

The `getSSHS()` function return the maximum Saffir-Simpson hurricane wind scale category reached by each storm in the `storm` or `stormsList` object.

Usage

```
getSSHS(s)

## S4 method for signature 'storm'
getSSHS(s)

## S4 method for signature 'stormsList'
getSSHS(s)
```

Arguments

`s`	`storm` or `stormsList` object.

Value

numeric vector.

Examples

```
#Creating a stormsDataset
sds <- defStormsDataset()

#Getting storm track data for all storms near New Caledonia
sts <- defStormsList(sds=sds, loi = "New Caledonia")

#Getting maximum Saffir-Simpson hurricane wind scale category
#reached by each storm in the sts object
getSSHS(sts)
```

getStorm *Extracting a* `storm`

Description

The `getStorm()` function extracts a specific `storm` object from a `stormsList` object.

Usage

```
getStorm(sts, name, season = NULL)

## S4 method for signature 'stormsList'
getStorm(sts, name, season = NULL)
```

Arguments

`sts`	`stormsList`
`name`	character. Name of the storm to extract.
`season`	numeric vector. Seasons of occurrence of the storms (e.g., c(2020,2022)). In the Southern Hemisphere, the cyclone season extends across two consecutive years. Therefore, to capture the 2021 to 2022 cyclone season both years should be specified, with cyclones assigned for the year #' that originated in. By default all storms occurring since 1980 are extracted.

Value

A `storm` object.

Examples

```
#Creating a stormsDataset

sds <- defStormsDataset()

#Getting storm track data for all storms near New Caledonia
sts <- defStormsList(sds=sds, loi = "New Caledonia")

#Getting `storm` for the tropical cyclone Niran
st <- getStorm(sts, name = "NIRAN")
```

`plotBehaviour` *Plotting spatial wind behaviour*

Description

The `plotBehaviour()` function allows plotting spatial statistics generated using the `spatialBehaviour()` function and stored in `SpatRaster` objects.

Usage

```
plotBehaviour(
  sts,
  rasterProduct,
  colorPalette = NULL,
  main = NULL,
  xlim = NULL,
  ylim = NULL,
  labels = FALSE,
  by = 8,
  pos = 3,
  legends = "topleft"
)
```

Arguments

`sts`	`StormsList` object.
`rasterProduct`	layer name in a `SpatRaster` object. The names of the layers follow the following terminology: • for "MSW" or "PDI", the name of the storm in capital letters and the name of the statistic separated by underscores (e.g., "PAM_MSW", "PAM_PDI"), • for duration of exposure, the name of the storm in capital letters, "Exposure", and the threshold value separated by underscores (e.g., "PAM_Exposure_18", "PAM_Exposure_33", ...). • for wind profiles, the name of the storm in capital letters, "Speed" or "Direction", and the indices of the observation separated by underscores (e.g., "PAM_Speed_41", "PAM_Direction_41",...).
`colorPalette`	character vector. The color palette used to plot the raster layer. If `colorPalette=NULL` (default setting), the default color palette is used.
`main`	character. Title of the plot. If `main=NULL` (default setting), a default title is generated based on the name of the layer.
`xlim`	numeric vector. The x limits of the plot.
`ylim`	numeric vector. The y limits of the plot.
`labels`	logical. Whether (TRUE) or not (FALSE, default setting) to add labels with the name of the storm and the indices and ISO times of the observation.

`by`	numeric. If `labels=TRUE`, defines the frequency at which labels are plotted. Default value is set to `8` which corresponds to a 24h (or 48h) time interval between the labelled observations when observations are made every 3 (or 6) hours.
`pos`	numeric. If `labels=TRUE`, defines the position of the labels, `1` (above the observation), `2` (on the left), `3` (below, default setting), and `4` (on the right).
`legends`	character. Indicates where to plot the legend, `"topright"`, `"topleft"` (default setting), `"bottomleft"`, `"bottomright"`, or `"none"` (legend not plotted).

Value

A plot of the storm track data with the raster layer.

Examples

```
# Creating a stormsDataset
sds <- defStormsDataset()

# Getting storm track data for tropical cyclone Pam (2015)
pam <- defStormsList(sds = sds, loi = "Vanuatu", names = "PAM")

# Plotting maximum sustained wind speed for Pam (2015) near Vanuatu
pam.msw <- spatialBehaviour(pam, verbose = 0)
plotBehaviour(pam, pam.msw)

# Plotting 2D wind speed profile for Pam (2015) near Vanuatu
pam.prof <- spatialBehaviour(pam, product = "Profiles", verbose = 0)
plotBehaviour(pam, pam.prof$PAM_Speed_37, labels = TRUE, pos = 4)
```

plotStorms *Plotting storm track data*

Description

This `plotStorms()` function allows plotting storm track data stored in a `StormsList` object.

Usage

```
plotStorms(
  sts,
  names = NULL,
  category = NULL,
  xlim = NULL,
  ylim = NULL,
  labels = FALSE,
  by = 8,
  pos = 3,
  legends = "topleft",
  loi = TRUE
)
```

Arguments

`sts`	`StormsList` object
`names`	character vector. Name(s) of the storm(s) in capital letters. If `names = NULL` (default setting), all storms are plotted.
`category`	numeric vector. Category of storms to be plotted in the Saffir-Simpson hurricane wind scale. Can be a single category or a range of categories among: • -1, for tropical depression, • 0, for tropical storms, • 1, for category 1 tropical cyclone, • 2, for category 2 tropical cyclone, • 3, for category 3 tropical cyclone, • 4, for category 4 tropical cyclone, or • 5, for category 5 tropical cyclone. If `category=NULL` (default setting), all storms are plotted.
`xlim`	numeric vector. The x limits of the plot.
`ylim`	numeric vector. The y limits of the plot.
`labels`	logical. Whether (TRUE) or not (FALSE, default setting) to add labels with the name of the storm and the indices and ISO times of the observation.
`by`	numeric. If `labels=TRUE`, defines the frequency at which labels are plotted. Default value is set to 8 which corresponds to a 24h (or 48h) time interval between the labelled observations when observations are made every 3 (or 6) hours.
`pos`	numeric. If `labels=TRUE`, defines the position of the labels, `1` (above the observation), `2` (on the left), `3` (below, default setting), and `4` (on the right).
`legends`	character. Indicates where to plot the legend, `"topright"`, `"topleft"` (default setting), `"bottomleft"`, `"bottomright"`, or `"none"` (legend not plotted).
`loi`	logical. Whether (TRUE, default setting) or not (FALSE) to plot the extent of the buffered location of interest on the map.

Value

A plot of the storm track data.

Examples

```
#' #Creating a stormsDataset

dev.off()
sds <- defStormsDataset()

# Getting storm track data for tropical cyclone Pam (2015)
```

```
pam <- defStormsList(sds = sds, loi = "Vanuatu", names = "PAM")

# Plotting Pam over Vanuatu with labels every 24h
plotStorms(pam, labels = TRUE)

# Plotting Pam over Vanuatu with labels every 6h on the right side of the observations
plotStorms(pam, labels = TRUE, by = 2, pos = 4)
```

`spatialBehaviour` *Computing wind behaviour and summary statistics over given areas*

Description

The spatialBehaviour() function allows computing wind speed and direction for each cell of a regular grid (i.e., a raster) for a given tropical cyclone or set of tropical cyclones. It also allows to compute three associated summary statistics.

Usage

```
spatialBehaviour(
  sts,
  product = "MSW",
  windThreshold = c(18, 33, 42, 49, 58, 70),
  method = "Willoughby",
  asymmetry = "Chen",
  empiricalRMW = FALSE,
  spaceRes = "2.5min",
  tempRes = 1,
  verbose = 2
)
```

Arguments

`sts`	`StormsList` object
`product`	character vector. Desired output statistics: • `"Profiles"`, for 2D wind speed and direction fields, • `"MSW"`, for maximum sustained wind speed (default setting), • `"PDI"`, for power dissipation index, or • `"Exposure"`, for duration of exposure.
`windThreshold`	numeric vector. Minimal wind threshold(s) (in $m.s^{-1}$) used to compute the duration of exposure when `product="Exposure"`. By default the thresholds used in the Saffir-Simpson hurricane wind scale are used (i.e., 18, 33, 42, 49, 58, 70 $m.s^{-1}$).
`method`	character. Model used to compute wind speed and direction. Three different models are implemented:

	• "Willoughby", for the symmetrical model developed by Willoughby et al. (2006) (default setting), • "Holland", for the symmetrical model developed by Holland (1980), or • "Boose", for the asymmetrical model developed by Boose et al. (2004).
`asymmetry`	character. If `method="Holland"` or `method="Willoughby"`, this argument specifies the method used to add asymmetry. Can be: • `"Chen"`, for the model developed by Chen (1994) (default setting), • `"Miyazaki"`, for the model developed by Miyazaki et al. (1962), or • `"None"`, for no asymmetry.
`empiricalRMW`	logical. Whether (TRUE) or not (FALSE) to compute the radius of maximum wind (`rmw`) empirically using the model developed by Willoughby et al. (2006). If `empiricalRMW==FALSE` (default setting) then the `rmw` provided in the `StormsList` is used.
`spaceRes`	character. Spatial resolution. Can be `"30 sec"` (~1 km at the equator), `"2.5 min"` (~4.5 km at the equator), `"5 min"` (~9 km at the equator) or `"10 min"` (~18.6 km at the equator). Default setting is `"2.5 min"`.
`tempRes`	numeric. Temporal resolution. Can be `1` (for 60 min, default setting), `0.75` (for 45min), `0.5` (for 30 min), and `0.25` (15 for min).
`verbose`	numeric. Whether or not the function should display informations about the process and/or outputs. Can be: • `2`, information about the processes and outputs are displayed (default setting), • `1`, information about the processes are displayed, pr • `0`, no information displayed.

Details

Storm track data sets, such as those extracted from IBRTrACKS (Knapp et al., 2010), usually provide observation at a 3- or 6-hours temporal resolution. In the spatialBehaviour() function, linear interpolations are used to reach the temporal resolution specified in the `tempRes` argument (default value = 1 hour). When `product = "MSW"`, `product = "PDI"`, or `product = "Exposure"` the `focal()` function from the `terra` R package is used to smooth the results using moving windows.

The Holland (1980) model, widely used in the literature, is based on the 'gradient wind balance in mature tropical cyclones. The wind speed distribution is computed from the circular air pressure field, which can be derived from the central and environmental pressure and the radius of maximum winds.

$$v_r = \sqrt{\frac{b}{\rho} \times \left(\frac{R_m}{r}\right)^b \times (p_{oci} - p_c) \times e^{-\left(\frac{R_m}{r}\right)^b} + \left(\frac{r \times f}{2}\right)^2} - \left(\frac{r \times f}{2}\right)$$

with,

$$b = \frac{\rho \times e \times v_m^2}{p_{oci} - p_c}$$

$$f = 2 \times 7.29 \times 10^{-5} \sin(\phi)$$

where, v_r is the tangential wind speed (in $m.s^{-1}$), b is the shape parameter, ρ is the air density set to $1.15kg.m^{-3}$, e being the base of natural logarithms (~2.718282), v_m the maximum sustained wind speed (in $m.s^{-1}$), p_{oci} is the pressure at outermost closed isobar of the storm (in Pa), p_c is the pressure at the centre of the storm (in Pa), r is the distance to the eye of the storm (in km), R_m is the radius of maximum sustained wind speed (in km), f is the Coriolis force (in $N.kg^{-1}$, and ϕ being the latitude).

The Willoughby et al. (2006) model is an empirical model fitted to aircraft observations. The model considers two regions: inside the eye and at external radii, for which the wind formulations use different exponents to better match observations. In this model, the wind speed increases as a power function of the radius inside the eye and decays exponentially outside the eye after a smooth polynomial transition across the eyewall.

$$\begin{cases} v_r = v_m \times \left(\frac{r}{R_m}\right)^n & if \quad r < R_m \\ v_r = v_m \times \left((1-A) \times e^{-\frac{|r-R_m|}{X1}} + A \times e^{-\frac{|r-R_m|}{X2}}\right) & if \quad r \geq R_m \end{cases}$$

with,

$n = 2.1340 + 0.0077 \times v_m - 0.4522 \times \ln(R_m) - 0.0038 \times |\phi|$

$X1 = 287.6 - 1.942 \times v_m + 7.799 \times \ln(R_m) + 1.819 \times |\phi|$

$A = 0.5913 + 0.0029 \times v_m - 0.1361 \times \ln(R_m) - 0.0042 \times |\phi|$ and $A \geq 0$

where, v_r is the tangential wind speed (in $m.s^{-1}$), v_m is the maximum sustained wind speed (in $m.s^{-1}$), r is the distance to the eye of the storm (in km), R_m is the radius of maximum sustained wind speed (in km), ϕ is the latitude of the centre of the storm, $X2 = 25$.

Asymmetry can be added to Holland (1980) and Willoughby et al. (2006) wind fields as follows,

$\vec{V} = \vec{V_c} + C \times \vec{V_t}$

where, $\vec{V}$ is the combined asymmetric wind field, $\vec{V_c}$ is symmetric wind field, $\vec{V_t}$ is the translation speed of the storm, and C is function of r, the distance to the eye of the storm (in km).

Two formulations of C proposed by Miyazaki et al. (1962) and Chen (1994) are implemented.

Miyazaki et al. (1962) $C = e^{(-\frac{r}{500} \times \pi)}$

Chen (1994) $C = \frac{3 \times R_m^{\frac{3}{2}} \times r^{\frac{3}{2}}}{R_m^3 + r^3 + R_m^{\frac{3}{2}} \times r^{\frac{3}{2}}}$

where, R_m is the radius of maximum sustained wind speed (in km)

The Boose et al. (2004) model, or “HURRECON” model, is a modification of the Holland (1980) model. In addition to adding asymmetry, this model treats of water and land differently, using different surface friction coefficient for each.

$v_r = F\left(v_m - S \times (1 - \sin(T)) \times \frac{v_h}{2}\right) \times \sqrt{\left(\frac{R_m}{r}\right)^b \times e^{1-\left(\frac{R_m}{r}\right)^b}}$

with,

$b = \frac{\rho \times e \times v_m^2}{p_{oci} - p_c}$

where, v_r is the tangential wind speed (in $m.s^{-1}$), F is a scaling parameter for friction (1.0 in water, 0.8 in land), v_m is the maximum sustained wind speed (in $m.s^{-1}$), S is a scaling parameter for asymmetry (usually set to 1), T is the oriented angle (clockwise/counter clockwise in Northern/Southern Hemisphere) between the forward trajectory of the storm and a radial line from the eye of the storm to point r v_h is the storm velocity (in $m.s^{-1}$), R_m is the radius of maximum sustained wind speed (in km), r is the distance to the eye of the storm (in km), b is the shape parameter, $\rho = 1.15$ is the air density (in $kg.m^{-3}$), p_{oci} is the pressure at outermost closed isobar of the storm (in Pa), and p_c is the pressure at the centre of the storm ($pressure$ in Pa).

Value

The spatialBehaviour() function returns SpatRaster objects (in WGS84). The number of layers in the output depends on the number of storms in the inputs, on the desired `product`, as well as the `tempRes` argument:

- if `product = "MSW"`, the function returns one layer for each `Storm`. The names of the layer follow the following terminology, the name of the storm in capital letters and "MSW" separated by underscores (e.g., "PAM_MSW"),
- if `product = "PDI"`, the function returns one layer for each `Storm`. The names of the layer follow the following terminology, the name of the storm in capital letters and "PDI" separated by underscores (e.g., "PAM_PDI"),
- if `product ="Exposure"`, the function returns one layer for each wind speed values in the `windThreshold` argument and for each `Storm`. The names of the layer follow the following terminology, the name of the storm in capital letters, "Exposure", and the threshold value separated by underscores (e.g., "PAM_Exposure_18", "PAM_Exposure_33", ...),
- if `product = "Profiles"` the function returns one layer for wind speed and one layer for wind direction for each observation or interpolated observation and each `Storm`. The names of the layer follow the following terminology, the name of the storm in capital letters, "Speed" or "Direction", and the indices of the observation separated by underscores (e.g., "PAM_Speed_41", "PAM_Direction_41",...).

References

Boose, E. R., Serrano, M. I., & Foster, D. R. (2004). Landscape and regional impacts of hurricanes in Puerto Rico. Ecological Monographs, 74(2), Article 2. https://doi.org/10.1890/02-4057

Chen, K.-M. (1994). A computation method for typhoon wind field. Tropic Oceanology, 13(2), 41–48.

Holland, G. J. (1980). An Analytic Model of the Wind and Pressure Profiles in Hurricanes. Monthly Weather Review, 108(8), 1212–1218. https://doi.org/10.1175/1520-0493(1980)108<1212:AAMOTW>2.0.CO;2

Knapp, K. R., Kruk, M. C., Levinson, D. H., Diamond, H. J., & Neumann, C. J. (2010). The International Best Track Archive for Climate Stewardship (IBTrACS). Bulletin of the American Meteorological Society, 91(3), Article 3. https://doi.org/10.1175/2009bams2755.1

Miyazaki, M., Ueno, T., & Unoki, S. (1962). The theoretical investigations of typhoon surges along the Japanese coast (II). Oceanographical Magazine, 13(2), 103–117.

Willoughby, H. E., Darling, R. W. R., & Rahn, M. E. (2006). Parametric Representation of the Primary Hurricane Vortex. Part II: A New Family of Sectionally Continuous Profiles. Monthly Weather Review, 134(4), 1102–1120. https://doi.org/10.1175/MWR3106.1

Examples

```
# Creating a stormsDataset
sds <- defStormsDataset()

# Geting storm track data for tropical cyclone Pam (2015) near Vanuatu
pam <- defStormsList(sds = sds, loi = "Vanuatu", names = "PAM")
```

```
# Computing maximum sustained wind speed generated by Pam (2015) near Vanuatu
# using default settings
msw.pam <- spatialBehaviour(pam)

# Computing PDI generated by Pam (2015) near Vanuatu using the Holland model without asymmetry
pdi.pam <- spatialBehaviour(pam, method = "Holland", product = "PDI", asymmetry = "None")

# Computing duration of exposure to Saffir-Simpson hurricane wind scale threshold values
# during Pam (2015) near Vanuatu using default settings
exp.pam <- spatialBehaviour(pam, product = "Exposure")

# Computing wind speed and direction profiles  generated by Pam (2015) near Vanuatu
# using Boose model
prof.pam <- spatialBehaviour(pam, product = "Profiles", method = "Boose")
```

storm-class — storm *object*

Description

Gather all the needed informations to model a single storm

Value

A storm object.

- name, character.
- season, numeric.
- sshs, numeric.
- obs, numeric.
- obs.all, data.frame.

Slots

name character. Name of the storm

season numeric. Cyclonic season in which the storm has occured

sshs numeric. Maximum category reached in the Saffir Simpson Hurricane Scale

obs numeric vector. Indices of observations within the location of interest extented with its corresponding buffer (See stormsList class)

obs.all data.frame. Contains all of the observations available. An observation is made up of several fields which are:

- iso.time, Date and hours of observations (UTC)
- lon, Longitude coordinate (Eastern degree)
- lat, Latitude coordinate (Northern degree)

- `msw`, Maximum Sustained Wind (m/s)
- `sshs`, Category in the Saffir Simpson Hurricane Scale

The following field is not mandatory but highly recommended

- `rmw`, Radius of Maximum Wind (km)

Also, the following fields are only mandatory to perform Holland and Boose models (See `Details`)

- `pres`, Pressure at the center (pa)
- `poci`, Pressure of the Outermost Closed Isobar (pa)

`stormsDataset-class` *stormsDataset*

Description

Choose the database to use within the package's functions

Details

The fields input must provide at least 6 mandatory fields (and at most 11) in order to benefit from all the functionalities of this package:

- A field `names`: which dimension contains the names of storms in the netcdf database
- A field `seasons`: which dimension contains the cyclonic seasons of storms in the netcdf database
- A field `isoTime`: which dimension contains the ISO times of each (3 or 6 hourly) observations for all storms in the database
- A field `lon`: which dimension contains the longitude coordinates of each observations for all storms in the netcdf database
- A field `lat`: which dimension contains the latitude coordinates of each observations for all storms in the netcdf database
- A field `msw`: which dimension contains the maximum sustained wind speed of each observations for all storms in the netcdf database

The following fields are optional but highly recommended:

- A field `basin`: which dimension contains the basin location of storms in the netcdf database. Used to filter the storms in the netcdf database
- A field `rmw`: which dimension contains the radius of maximum wind speed of each observations for all storms in the netcdf database (See spatialBehaviour, temporalBehaviour)
- A field `sshs`: which dimension contains the Saffir Simpson Hurricane Scale index of each observations for all storms in the netcdf database

Finally these following fields are optional but mandatory to perform Holland model (See `spatialBehaviour`, `temporalBehaviour`)

- A field `pressure`: which dimension contains the pressure in the eye for of each observations for all storms in the netcdf database
- A field `poci`: which dimension contains the Pressure at the Outermost Closed Isobar for of each observations for all storms in the nectdf database

Default value is set according to the most relevant dimensions of IBTrACS databases: `fields = c(basin = "basin", names = "name", seasons = "season", isoTime = "iso_time", lon = "usa_lon", lat = "usa_lat", msw = "usa_wind", rmw = "usa_rmw", pressure = "usa_pres", poci = "usa_poci", sshs = "usa_sshs")`

Slots

`filename` character. Name of the database to load. Must be a netcdf file

`fields` named character vector. Dictionary that provides all the name of dimensions to extract from the netcdf database (See `Details`)

`basin` character. Basin name to filter the database within its boundaries. It must be either

- `"NA"`: North Atlantic
- `"SA"`: South Atlantic
- `"EP"`: Eastern North Pacific
- `"WP"`: Western North Pacific
- `"SP"`: South Pacific
- `"SI"`: South India
- `"NI"`: North India
- `"None"`: No particular basin

`seasons` numeric vector. Range of calendar years to filter storms. For cyclones that formed in one year and dissipated in the following year, the latter should be used

`database` list of 6 to 10 slots depending on the fields input. Each slot is either a 1D array of dimension (number of storms) for `names` and `seasons` fields, or a 2D array of dimension (Maximum number of observations:number of storms), for the remaining fields which are `isoTime`, `lon`, `lat`, `msw`, `rmw`, `pressure`, `poci`, `sshs`

`stormsList-class` `stormsList` *object*

Description

Gather all the needed informations to model a set of storms

Value

A `stormsList` object.

- `data`, list.
- `buffer`, numeric.
- `spatialLoi`, sf.
- `spatialLoiBuffer`, sf.

Slots

`data`	A list of `storm` (See `storm` class)
`buffer`	numeric. Buffer used to extent `spatialLoi` (km)
`spatialLoi`	sf object. Represents the location of interest. Projection is EPSG:4326
`spatialLoiBuffer`	sf object. Buffer extension of `spatialLoi`

`temporalBehaviour` *Computing wind behaviour time series and summary statistics at given point locations*

Description

The `temporalBehaviour()` function allows computing wind speed and direction for a given location or set of locations along the lifespan of a tropical cyclone. It also allows to compute three associated summary statistics.

Usage

```
temporalBehaviour(
  sts,
  points,
  product = "TS",
  windThreshold = c(18, 33, 42, 49, 58, 70),
  method = "Willoughby",
  asymmetry = "Chen",
  empiricalRMW = FALSE,
  tempRes = 1,
  verbose = 1
)
```

Arguments

`sts`	`StormsList` object.
`points`	data.frame. Consisting of two columns names as "x" (for the longitude) and "y" (for the latitude), providing the coordinates in decimal degrees of the point locations. Row names can also be provided to named the locations.
`product`	character. Desired output statistics: • `"TS"`, for time series of wind speeds and directions (default setting), • `"PDI"`, for power dissipation index, or • `"Exposure"`, for the duration of exposure to defined wind thresholds.
`windThreshold`	numeric vector. Minimal wind threshold(s) (in $m.s^{-1}$) used to compute the duration of exposure when `product="Exposure"`. By default the thresholds used in the Saffir-Simpson hurricane wind scale are used (i.e., 18, 33, 42, 49, 58, 70 $m.s^{-1}$).

`method`	character. Model used to compute wind speed and direction. Three different models are implemented: • `"Willoughby"`, for the symmetrical model developed by Willoughby et al. (2006) (default setting), • `"Holland"`, for the symmetrical model developed by Holland (1980), or • `"Boose"`, for the asymmetrical model developed by Boose et al. (2004).
`asymmetry`	character. If `method="Holland"` or `method="Willoughby"`, this argument specifies the method used to add asymmetry. Can be: • `"Chen"`, for the model developed by Chen (1994) (default setting), • `"Miyazaki"`, for the model developed by Miyazaki et al. (1962), or • `"None"`, for no asymmetry.
`empiricalRMW`	logical. Whether (TRUE) or not (FALSE) to compute the radius of maximum wind (`rmw`) empirically using the model developed by Willoughby et al. (2006). If `empiricalRMW==FALSE` (default setting) then the `rmw` provided in the `StormsList` is used.
`tempRes`	numeric. Temporal resolution. Can be `1` (for 60 min, default setting), `0.75` (for 45min), `0.5` (for 30 min), and `0.25` (15 for min).
`verbose`	numeric. Information displayed. Can be: • `2`, information about the processes and outputs are displayed (default setting), • `1`, information about the processes are displayed, or • `0`, no information displayed.

Details

Storm track data sets, such as those extracted from IBRTrACKS (Knapp et al., 2010), usually provide observation at a 3- or 6-hours temporal resolution. In the temporalBehaviour() function, linear interpolations are used to reach the temporal resolution specified in the `tempRes` argument (default value = 1 hour).

The Holland (1980) model, widely used in the literature, is based on the 'gradient wind balance in mature tropical cyclones. The wind speed distribution is computed from the circular air pressure field, which can be derived from the central and environmental pressure and the radius of maximum winds.

$$v_r = \sqrt{\frac{b}{\rho} \times \left(\frac{R_m}{r}\right)^b \times (p_{oci} - p_c) \times e^{-\left(\frac{R_m}{r}\right)^b} + \left(\frac{r \times f}{2}\right)^2} - \left(\frac{r \times f}{2}\right)$$

with,

$$b = \frac{\rho \times e \times v_m^2}{p_{oci} - p_c}$$

$$f = 2 \times 7.29 \times 10^{-5} \sin(\phi)$$

where, v_r is the tangential wind speed (in $m.s^{-1}$), b is the shape parameter, ρ is the air density set to $1.15kg.m^{-3}$, e being the base of natural logarithms (~2.718282), v_m the maximum sustained wind speed (in $m.s^{-1}$), p_{oci} is the pressure at outermost closed isobar of the storm (in Pa), p_c is the pressure at the centre of the storm (in Pa), r is the distance to the eye of the storm (in km), R_m is the radius of maximum sustained wind speed (in km), f is the Coriolis force (in $N.kg^{-1}$, and ϕ being the latitude).

The Willoughby et al. (2006) model is an empirical model fitted to aircraft observations. The model considers two regions: inside the eye and at external radii, for which the wind formulations use different exponents to better match observations. In this model, the wind speed increases as a power function of the radius inside the eye and decays exponentially outside the eye after a smooth polynomial transition across the eyewall.

$$\begin{cases} v_r = v_m \times \left(\frac{r}{R_m}\right)^n \quad if \quad r < R_m \\ v_r = v_m \times \left((1-A) \times e^{-\frac{|r-R_m|}{X1}} + A \times e^{-\frac{|r-R_m|}{X2}}\right) \quad if \quad r \geq R_m \end{cases}$$

with,

$n = 2.1340 + 0.0077 \times v_m - 0.4522 \times \ln(R_m) - 0.0038 \times |\phi|$

$X1 = 287.6 - 1.942 \times v_m + 7.799 \times \ln(R_m) + 1.819 \times |\phi|$

$A = 0.5913 + 0.0029 \times v_m - 0.1361 \times \ln(R_m) - 0.0042 \times |\phi|$ and $A \geq 0$

where, v_r is the tangential wind speed (in $m.s^{-1}$), v_m is the maximum sustained wind speed (in $m.s^{-1}$), r is the distance to the eye of the storm (in km), R_m is the radius of maximum sustained wind speed (in km), ϕ is the latitude of the centre of the storm, $X2 = 25$.

Asymmetry can be added to Holland (1980) and Willoughby et al. (2006) wind fields as follows,

$\vec{V} = \vec{V_c} + C \times \vec{V_t}$

where, $\vec{V}$ is the combined asymmetric wind field, $\vec{V_c}$ is symmetric wind field, $\vec{V_t}$ is the translation speed of the storm, and C is function of r, the distance to the eye of the storm (in km).

Two formulations of C proposed by Miyazaki et al. (1962) and Chen (1994) are implemented.

Miyazaki et al. (1962) $C = e^{(-\frac{r}{500} \times \pi)}$

Chen (1994) $C = \frac{3 \times R_m^{\frac{3}{2}} \times r^{\frac{3}{2}}}{R_m^3 + r^3 + R_m^{\frac{3}{2}} \times r^{\frac{3}{2}}}$

where, R_m is the radius of maximum sustained wind speed (in km)

The Boose et al. (2004) model, or "HURRECON" model, is a modification of the Holland (1980) model. In addition to adding asymmetry, this model treats of water and land differently, using different surface friction coefficient for each.

$v_r = F\left(v_m - S \times (1 - \sin(T)) \times \frac{v_h}{2}\right) \times \sqrt{\left(\frac{R_m}{r}\right)^b \times e^{1-\left(\frac{R_m}{r}\right)^b}}$

with,

$b = \frac{\rho \times e \times v_m^2}{p_{oci} - p_c}$

where, v_r is the tangential wind speed (in $m.s^{-1}$), F is a scaling parameter for friction (1.0 in water, 0.8 in land), v_m is the maximum sustained wind speed (in $m.s^{-1}$), S is a scaling parameter for asymmetry (usually set to 1), T is the oriented angle (clockwise/counter clockwise in Northern/Southern Hemisphere) between the forward trajectory of the storm and a radial line from the eye of the storm to point r v_h is the storm velocity (in $m.s^{-1}$), R_m is the radius of maximum sustained wind speed (in km), r is the distance to the eye of the storm (in km), b is the shape parameter, $\rho = 1.15$ is the air density (in $kg.m^{-3}$), p_{oci} is the pressure at outermost closed isobar of the storm (in Pa), and p_c is the pressure at the centre of the storm ($pressure$ in Pa).

Value

For each storm and each point location, the `temporalBehaviour()` function returns a data.frame. The data frames are organised into named lists. Depending on the `product` argument different data.frame are returned:

- if `product == "TS"`, the function returns a data.frame with one row for each observation (or interpolated observation) and four columns for wind speed (in $m.s^{-1}$), wind direction (in degree), the observation number, and the ISO time of observations,
- if `product == "PDI"`, the function returns a data.frame with one row for each point location and one column for the PDI,
- if `product == "Exposure"`, the function returns a data.frame with one row for the duration of exposure to winds above each wind speed threshold defined by the `windThreshold` argument and one column for each point location.

References

Boose, E. R., Serrano, M. I., & Foster, D. R. (2004). Landscape and regional impacts of hurricanes in Puerto Rico. Ecological Monographs, 74(2), Article 2. https://doi.org/10.1890/02-4057

Chen, K.-M. (1994). A computation method for typhoon wind field. Tropic Oceanology, 13(2), 41–48.

Holland, G. J. (1980). An Analytic Model of the Wind and Pressure Profiles in Hurricanes. Monthly Weather Review, 108(8), 1212–1218. https://doi.org/10.1175/1520-0493(1980)108<1212:AAMOTW>2.0.CO;2

Knapp, K. R., Kruk, M. C., Levinson, D. H., Diamond, H. J., & Neumann, C. J. (2010). The International Best Track Archive for Climate Stewardship (IBTrACS). Bulletin of the American Meteorological Society, 91(3), Article 3. https://doi.org/10.1175/2009bams2755.1

Miyazaki, M., Ueno, T., & Unoki, S. (1962). The theoretical investigations of typhoon surges along the Japanese coast (II). Oceanographical Magazine, 13(2), 103–117.

Willoughby, H. E., Darling, R. W. R., & Rahn, M. E. (2006). Parametric Representation of the Primary Hurricane Vortex. Part II: A New Family of Sectionally Continuous Profiles. Monthly Weather Review, 134(4), 1102–1120. https://doi.org/10.1175/MWR3106.1

Examples

```
# Creating a stormsDataset

sds <- defStormsDataset()

# Geting storm track data for tropical cyclone Pam (2015) near Vanuatu
pam <- defStormsList(sds = sds, loi = "Vanuatu", names = "PAM")

pts <- data.frame(x = c(168.5, 168), y = c(-17.9, -16.3))
row.names(pts) <- c("point_1", "point_2")

# Computing time series of wind speed and direction for Pam
# over points 1 and 2 defined above
ts.pam <- temporalBehaviour(pam, points = pts)

# Computing PDI for Pam over points 1 and 2 defined above
```

```
pdi.pam <- temporalBehaviour(pam, points = pts, product = "PDI")

# Computing the duration of exposure to wind speeds above the thresholds
# used by the Saffir-Simpson hurricane wind scale for Pam
# over points 1 and 2 defined above
exp.pam <- temporalBehaviour(pam, points = pts, product = "Exposure")
```

`writeRast` *Exporting rasters to GeoTIFF or NetCDF files*

Description

The `writeRast()` function exports rasters stored in SpatRaster objects to GeoTIFF or NetCDF files.

Usage

```
writeRast(rast, filename = NULL, path = "./", ...)
```

Arguments

`rast`	`SpatRaster` object.
`filename`	character. Output file name. Can be either a `".tiff"` file (GeoTIFF) or `".nc"` (NetCDF). By default `filename=NULL` and the name of the file is generated based on raster information (storms and products) with the '.tiff' extension.
`path`	character. Path to the directory where the file is exported to. By default `path='./'`.
`...`	Additional arguments to be passed to `terra::writeRaster` function when saving to `".tiff"` format.

Value

`NULL`

Examples

```
# Creating a stormsDataset

sds <- defStormsDataset()

# Getting storm track data for tropical cyclone Pam (2015) near Vanuatu
pam <- defStormsList(sds = sds, loi = "Vanuatu", names = "PAM")

# Computing maximum sustained wind speed
pam.msw <- spatialBehaviour(pam)

# Exporting maximum sustained wind speed raster layer to a GeoTIFF file
writeRast(pam.msw, path = paste0(tempdir(), "/"))
```

```
# Computing power dissipation index for several storms near New Caledonia
sts.nc <- defStormsList(sds = sds, loi = "New Caledonia")
pdi.nc <- spatialBehaviour(sts.nc, product = "PDI")

# Exporting the power dissipation index raster layers to a NetCDF file
writeRast(pdi.nc, path = paste0(tempdir(), "/"))
```

Index