Package ‘rWind’

October 14, 2022

Encoding UTF-8
Type Package
Title Download, Edit and Include Wind and Sea Currents Data in Ecological and Evolutionary Analysis
Version 1.1.7
BugReports https://github.com/jabiologo/rWind/issues
Maintainer Javier Fernández-López <jflopez.bio@gmail.com>
URL http://allthiswasfield.blogspot.com.es/
License GPL (>= 3)
LazyData true
Imports raster (>= 2.5-8), gdistance, Matrix, lubridate
Suggests testthat, rmarkdown, knitr
VignetteBuilder knitr
Depends R (>= 3.4)
RoxygenNote 7.1.1
NeedsCompilation no
Author Javier Fernández-López [aut, cre], Klaus Schliep [aut], Yurena Arjona [aut]
Repository CRAN
Date/Publication 2021-10-19 16:00:02 UTC

R topics documented:

<table>
<thead>
<tr>
<th>rWind-package</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>arrowDir</td>
<td>3</td>
</tr>
</tbody>
</table>
Description

rWind contain tools for downloading, editing and transforming wind and sea currents data from Global Forecast System (GFS) and Ocean Surface Current Analyses Real-time (OSCAR). It also allows to use wind and sea currents data to compute the minimum cost path from wind speed and direction to perform connectivity analysis.

Details

The complete list of functions can be displayed with `library(help = rWind)`. For more information, please check: http://allthiswasfield.blogspot.com.es/

Author(s)

Javier Fernández-López
Klaus Schliep
Yurena Arjona

Maintainer: Javier Fernández-López jlopez.bio@gmail.com
**Description**

`arrowDir` adapts wind direction value to be used by Arrowhead function from "shape" package to plot wind direction for each coordinate.

**Usage**

```r
arrowDir(W)
```

**Arguments**

- `W`:
  
  An object of class `rWind` or a data.frame which should contain a column named "dir".

**Details**

Angle argument of Arrowhead function from "shape" package needs to be fed in an anti-clockwise way, relative to x-axis, in degrees 0,360. `arrowDir` function adapts wind direction provided by `wind.fit` (clockwise, relative to y-axis) to requirements of Arrowhead.

**Value**

A vector with angles for each arrow to be plotted by Arrowhead.

**Note**

`arrowDir` function works always together with Arrowhead function from "shape" package.

**Author(s)**

Javier Fernández-López

**References**


**See Also**

`wind.dl`
Examples

data(wind.data)

# Create a vector with wind direction (angles) adapted
alpha <- arrowDir(wind.data)

## Not run:
# Now, you can plot wind direction with Arrowhead function from shapes package
# Load "shape package
require(shape)
plot(wind.data$lon, wind.data$lat, type = "n")
Arrowhead(wind.data$lon, wind.data$lat,
    angle = alpha,
    arr.length = 0.1, arr.type = "curved"
)

## End(Not run)

cost.FMGS  

Compute flow-based cost or conductance

Description

flow.dispersion computes movement conductance through a flow either, sea or wind currents. It
implements the formula described in Felícisimo et al. 2008:

Usage

cost.FMGS(wind.direction, wind.speed, target, type = "active")

flow.dispersion(x, fun = cost.FMGS, output = "transitionLayer", ...)

Arguments

- wind.direction: A vector or scalar containing wind directions.
- wind.speed: A vector or scalar containing wind speeds.
- target: direction of the target cell
- type: Could be either "passive" or "active". In "passive" mode, movement against flow
direction is not allowed (deviations from the wind direction higher than 90). In "active"
mode, the movement can go against flow direction, by increasing the
cost.
- x: RasterStack object with layers obtained from wind2raster function ("rWind"
package) with direction and speed flow values.
- fun: A function to compute the cost to move between cells. The default is cost.FMGS
from Felícisimo et al. (2008), see details.
This argument allows to select different kinds of output. "raw" mode creates a matrix (class "dgCMatrix") with transition costs between all cells in the raster. "transitionLayer" creates a TransitionLayer object with conductance values to be used with "gdistance" package.

... Further arguments passed to or from other methods.

Details

Cost=(1/Speed)*(HorizontalFactor)

being HorizontalFactor a "function that incrementally penalized angular deviations from the wind direction" (Felicísimo et al. 2008).

Value

In "transitionLayer" output, the function returns conductance values (1/cost) to move between all cells in a raster having into account flow speed and direction obtained from wind.fit function ("rWind" package). As wind or sea currents implies directionality, flow.dispersion produces an anisotropic conductance matrix (asymmetric). Conductance values are used later to built a TransitionLayer object from "gdistance" package.

In "raw" output, flow.dispersion creates a sparse Matrix with cost values.

Note

Note that for large data sets, it could take a while. For large study areas is strongly advised perform the analysis in a remote computer or a cluster.

Author(s)

Javier Fernández-López; Klaus Schliep; Yurena Arjona

References


See Also

wind.dl, wind2raster

Examples

require(gdistance)
data(wind.data)
wind <- wind2raster(wind.data)
Conductance <- flow.dispersion(wind, type = "passive")

transitionMatrix(Conductance)
image(transitionMatrix(Conductance))

seaOscar.dl  

**OSCAR Sea currents data download**

**Description**

seaOscar.dl downloads sea currents data from the Ocean Surface Current Analyses Real-time (OSCAR) (https://coastwatch.pfeg.noaa.gov/erddap/info/jplOscar_LonPM180/index.html). Geospatial resolution is 0.33 degrees and sea currents are calculated for 15 m depth. CAUTION: OSCAR database has no data between 0 and 20 longitude degrees. You can use SCUD database instead (coming soon...)

**Usage**

```r
seaOscar.dl(
  yyyy, 
  mm, 
  dd, 
  lon1, 
  lon2, 
  lat1, 
  lat2, 
  type = "read-data", 
  trace = 1
)
```

**Arguments**

- **yyyy**  Selected year.
- **mm**  Selected month.
- **dd**  Selected day.
- **lon1**  Western longitude
- **lon2**  Eastern longitude
- **lat1**  Northern latitude
- **lat2**  Southern latitude
- **type**  Output type. "read-data" is selected by default, creating an R object. If you choose "csv", seaOscar.dl create a a CSV file in your working directory named "oscar_yyyy_mm_dd.csv".
- **trace**  if trace = 1 (by default) track downloaded files
tidy

Details

The output type is determined by type="csv" or type="read-data". If type="csv" is selected, the function creates a "sea_yyyy_mm_dd.csv" file that is downloaded at the work directory. If type="read-data" is selected, an R object (data.frame) is created.

Value

"rWind" and "data.frame" class object or .csv file with U and V vector components and sea current direction and speed for each coordinate in the study area defined by lon1/lon2 and lat1/lat2.

Author(s)

Javier Fernández-López (jflopez.bio@gmail.com)

References

http://www.digital-geography.com/cloud-gis-getting-weather-data/#.WDOWmbV1DCL
https://coastwatch.pfeg.noaa.gov/erddap/info/jplOscar_LonPM180/index.html

See Also

wind.dl_2, wind2raster

Examples

# Download sea currents for Galapagos Islands
## Not run:
seaOscar.dl(2015, 1, 1, -93, -88, 2, -3)
## End(Not run)

 tidy(x, ...)  Transforming a rWind_series object into a data.frame

Description

The output of tidy is always a data.frame. It is therefore suited for further manipulation by packages like dplyr, reshape2, ggplot2 and ggvis.

Usage

tidy(x, ...)

## S3 method for class 'rWind_series'
tidy(x, ...)
Arguments

x  An object to be converted into a tidy data frame
... extra arguments

Examples

data(wind.series)
df <- tidy(wind.series)
head(df)
## Not run:
# use the tidyverse
library(dplyr)
mean_speed <- tidy(wind.series) %>%
  group_by(lat, lon) %>%
  summarise(speed = mean(speed))
wind_average2 <- wind.mean(wind.series)
all.equal(wind_average2$speed, mean_speed$speed)
## End(Not run)

uv2ds

Transform U and V components in direction and speed and vice versa

Description

Transform U and V components in direction and speed and vice versa

Usage

uv2ds(u, v)
ds2uv(d, s)

Arguments

u  U component.
v  U component.
d  direction (degrees).
s  speed (m/s).

Value

"uv2ds" returns a matrix with direction and speed values
"ds2uv" returns a matrix with U and V values
wind.data

Note

Multiple U and V values can be processed. "dir" denotes where the wind/sea current is going (toward), not from where is coming.

Author(s)

Javier Fernández-López (jflopez.bio@gmail.com)

See Also

wind.mean, wind2raster

Examples

(ds <- uv2ds(c(1, 1, 3, 1), c(1, 1.7, 3, 1)))
ds2uv(ds[, 1], ds[, 2])

---

wind.data Wind data example

Description

This is an example of wind data obtained with wind.dl function for the Iberian Peninsula coordinates on 12/February/2015 at 00:00 (UTC)

Format

A list with one data.frame with 651 observations on the following 7 variables:

- list("time (UTC)") a numeric with selected time of wind data
- list("latitude (degrees_north)") a numeric with latitude values
- list("longitude (degrees_east)") a numeric with longitude values
- list("ugrd10m (m s-1)") a numeric with U component of wind data
- list("vgrd10m (m s-1)") a numeric with V component of wind data
- list("dir") a numeric with direction of wind data
- list("speed") a numeric with speed of wind data

Details

This data set is the result of:

wind.data <- wind.dl(2015, 2, 12, 0, -10, 5, 35, 45)

Source

References


Examples

data(wind.data)
str(wind.data)
head(wind.data[[1]])

wind.dl Wind-data download

Description

wind.dl downloads wind data from the Global Forecast System (GFS) of the USA's National Weather Service (NWS) (https://www.ncei.noaa.gov/products/weather-climate-models/global-forecast). Wind data are taken from NOAA/NCEP Global Forecast System (GFS) Atmospheric Model collection. Geospatial resolution is 0.5 degrees (approximately 50 km), and wind is calculated for Earth surface, at 10 m. More metadata information: https://pae-paha.pacioos.hawaii.edu/erddap/griddap/ncep_global.graph

Usage

wind.dl(
  yyyy,
  mm,
  dd,
  tt,
  lon1,
  lon2,
  lat1,
  lat2,
  type = "read-data",
  trace = 1
)

read.rWind(file)

Arguments

yyyy Selected year.
mm Selected month.
dd Selected day.
tt Selected time. There are currently several options at the GFS database: 00:00 - 03:00 - 06:00 - 09:00 - 12:00 - 15:00 - 18:00 - 21:00 (UTC).
lon1 Western longitude
wind.dl

lon2 Eastern longitude
lat1 Southern latitude
lat2 Northern latitude
type Output type. "read-data" is selected by default, creating an R object. If you choose "csv", wind.dl create a a CSV file in your working directory named "wind_yyyy_mm_dd_tt.csv".
trace if trace = 1 (by default) track downloaded files
file file name of the saved ".csv" files.

Details
The output type is determined by type="csv" or type="read-data". If type="csv" is selected, the function creates a "wind_yyyy_mm_dd_tt.csv" file that is downloaded at the work directory. If type="read-data" is selected, an R object (data.frame) is created.

Value
"rWind" and "data.frame" class object or .csv file with U and V vector components and wind direction and speed for each coordinate in the study area defined by lon1/lon2 and lat1/lat2.

Note
Longitude coordinate are provided by GFS dataset in 0/360 notation and transformed internally into -180/180. Wind "dir" denotes where the wind is going (toward), not from where is coming.

Author(s)
Javier Fernández-López (jflopez.bio@gmail.com)

References
http://www.digital-geography.com/cloud-gis-getting-weather-data/#.WDOWmbV1DCL
https://pae-paha.pacioos.hawaii.edu/erddap/griddap/ncep_global.graph

See Also
wind.dl_2, wind2raster

Examples

# Download wind for Iberian Peninsula region at 2015, February 12, 00:00
## Not run:
wind.dl(2015, 2, 12, 0, -10, 5, 35, 45)
## End(Not run)
wind.dl_2  

**Wind-data download**

### Description

wind.dl_2 downloads time-series wind data from the Global Forecast System (GFS) of the USA’s National Weather Service (NWS) (https://www.ncei.noaa.gov/products/weather-climate-models/global-forecast). Wind data are taken from NOAA/NCEP Global Forecast System (GFS) Atmospheric Model collection. Geospatial resolution is 0.5 degrees (approximately 50 km), and wind is calculated for Earth surface, at 10 m. More metadata information: http://oos.soest.hawaii.edu/erddap/info/NCEP_Global_Best/index.html

### Usage

```r
wind.dl_2(time, lon1, lon2, lat1, lat2, type = "read-data", trace = 1)
```

#### S3 method for class 'rWind_series'

```r
x[[i, exact = TRUE]]
```

### Arguments

- **time**: a scalar or vector of POSIXt or Date objects or an character which can transformed into those, see example below. There are currently these options at the GFS database for the hours: 00:00 - 03:00 - 06:00 - 09:00 - 12:00 - 15:00 - 18:00 - 21:00 (UTC) (TO).
- **lon1**: Western longitude
- **lon2**: Eastern longitude
- **lat1**: Southern latitude
- **lat2**: Northern latitude
- **type**: Output type. "read-data" is selected by default, creating an R object. If you choose "csv", wind.dl create a a CSV file in your work directory named "wind_yyyy_mm_dd_tt.csv".
- **trace**: if trace = 1 (by default) track downloaded files
- **x**: object from which to extract element(s).
- **i**: indices specifying elements to extract.
- **exact**: Controls possible partial matching (not used yet).

### Details

To get the same format as wind.dl, you should run tidy function from wind.dl_2 output. The output type is determined by type="csv" or type="read-data". If type="csv" is selected, the function creates a "wind_yyyy_mm_dd_tt.csv" file that is downloaded at the work directory. If type="read-data" is selected, an rWind_series object is created.

### Value

an object of class rWind_series or .csv file/s with U and V vector components and wind direction and speed for each coordinate in the study area defined by lon1/lon2 and lat1/lat2.
Note
wind.dl_2 requires two dates that represent the boundaries of the time lapse to download wind series data. U and V vector components allow you to create wind averages or tendencies for each coordinate at the study area. Longitude coordinates are provided by GFS dataset in 0/360 notation and transformed internally into -180/180. "dir" denotes where the wind/sea current is going (toward), not from where is coming.

Author(s)
Javier Fernández-López (jflopez.bio@gmail.com)

References
http://www.digital-geography.com/cloud-gis-getting-weather-data/#.WDOWmbV1DCL
http://oos.soest.hawaii.edu/erddap/griddap/NCEP_Global_Best.graph

See Also
wind.mean, wind2raster, wind.dl, as_datetime, as.POSIXct

Examples

# Download wind for Iberian Peninsula region at 2015, February 12, 00:00
## Not run:
wind.dl_2("2018/3/15 9:00:00", -10, 5, 35, 45)

library(lubridate)
dt <- seq(ymd_hms(paste(2018, 1, 1, 00, 00, 00, sep = "-")),
          ymd_hms(paste(2018, 1, 2, 21, 00, 00, sep = "-")),
          by = "3 hours")
ww <- wind.dl_2(dt, -10, 5, 35, 45)
tidy(ww)
## End(Not run)

Description
wind.mean computes the mean (average) wind speed and wind direction of a time series dataset of winds of the same region. Summaries of time series are not trivial to compute. We compute the arithmetic mean for the wind speed. The direction as the circular mean, see https://en.wikipedia.org/wiki/Circular_mean for more details. The U and V components are afterwards transformed from these values.
Usage

wind.mean(x)

Arguments

x An object of class rWind_series

Value

An object of class rWind, which is a data.frame

Note

For large time series, it could take a while.

Author(s)

Javier Fernández-López (jflopez.bio@gmail.com)

References


See Also

wind.dl

Examples

data(wind.series)
wind_average <- wind.mean(wind.series)

---

<table>
<thead>
<tr>
<th>wind.series</th>
<th>Wind series example</th>
</tr>
</thead>
</table>

Description

This is an example of a wind series data obtained with wind.dl function for New Zealand area on 3/January/2015 at all the available times: 00:00 - 03:00 - 06:00 - 09:00 - 12:00 - 15:00 - 18:00 - 21:00 (UTC)
**wind2raster**

**Format**

The format is an rWind list of 8 data.frame. Each data.frame contain 961 observations on the following 7 variables:

- **list("time (UTC)")**: a factor with selected time of wind data
- **list("latitude (degrees_north)")**: a factor with latitude values
- **list("longitude (degrees_east)")**: a factor with longitude values
- **list("ugrd10m (m s-1)")**: a factor with U component of wind data
- **list("vgrd10m (m s-1)")**: a factor with V component of wind data
- **list("dir")**: a numeric with direction of wind data
- **list("speed")**: a numeric with speed of wind data

**Details**

This data set is the result of:

```r
library(lubridate) dt <- seq(ymd_h(paste(2015,1,3,00, sep="-")), ymd_h(paste(2015,1,3,21, sep="-")),by="3 hours") wind.series <- wind.dl_2(dt, 164, 179, -48, -33)
```

**Source**


**References**


**Examples**

```r
data(wind.series)
str(tidy(wind.series))
```

---

**wind2raster**  
**Wind-data to raster file**

**Description**

wind2raster crates a raster stack (gridded) with 2 layers: wind speed and wind direction for an object of rWind. Latitude and longitude values are used to locate raster file and to create raster using rasterFromXYZ function from raster package. If the input file is a list of wind data created by wind.dl, a list of raster stacks will be returned

**Usage**

```r
wind2raster(x)
```
Arguments

x an "rWind list" obtained by wind.fit

Details

WGS84 datum (non-projected) CRS is selected by default to build the raster file.

Value

A raster stack or a list of raster stacks representing wind direction and speed.

Author(s)

Javier Fernández-López (jflopez.bio@gmail.com)

See Also

wind.dl

Examples

data(wind.data)

# Create raster stack from the downloaded data with wind direction and speed
# layers

wind2raster(wind.data)
Index

* ~anisotropy
  cost.FMGS, 4
* ~average
  wind.mean, 13
* ~conductance
  cost.FMGS, 4
* ~currents
  seaOscar.dl, 6
* ~gfs
  wind.dl, 10
  wind.dl_2, 12
  wind2raster, 15
* ~mean
  wind.mean, 13
* ~sea
  seaOscar.dl, 6
* ~wind
  arrowDir, 3
  uv2ds, 8
  wind.dl, 10
  wind.dl_2, 12
  wind2raster, 15
* datasets
  wind.data, 9
  wind.series, 14
* package
  rWind-package, 2
  [[.rWind_series (wind.dl_2), 12

  arrowDir, 3
  as.POSIXct, 13
  as_datetime, 13
  cost.FMGS, 4
  ds2uv (uv2ds), 8
  flow.dispersion (cost.FMGS), 4
  read.rWind (wind.dl), 10
  rWind (rWind-package), 2

rWind-package, 2
seaOscar.dl, 6
tidy, 7
uv2ds, 8
wind.data, 9
wind.dl, 3, 5, 10, 13, 14, 16
wind.dl_2, 7, 11, 12
wind.mean, 9, 13, 13
wind.series, 14
wind2raster, 5, 7, 9, 11, 13, 15