# Package 'rWind'

November 29, 2018

**Encoding** UTF-8

**Type** Package

**Title** Download, Edit and Include Wind Data in Ecological and Evolutionary Analysis

**Version** 1.0.4

**BugReports** https://github.com/jabiologo/rWind

**Maintainer** Javier Fernández-López <jflopez.bio@gmail.com>

**Description** Tools for download and manage surface wind data from the Global Forecasting System <https://www.ncdc.noaa.gov/data-access/model-data/model-datasets/global-forcast-system-gfs> and to compute wind connectivity between locations.

**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)

**NeedsCompilation** no

**Repository** CRAN

**RoxygenNote** 6.1.1

**Author** Javier Fernández-López [aut, cre], Klaus Schliep [aut]

**Date/Publication** 2018-11-29 13:30:03 UTC

## R topics documented:

---

`rWind-package` *Download, edit and include wind data in ecological and evolutionary analysis*

---

## Description

rWind contain tools for downloading, editing and transforming wind data from Global Forecast System (GFS). It also allows to use wind 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

Maintainer: Javier Fernández-López <jflopez.bio@gmail.com>

---

`arrowDir` *Arrow direction fitting for Arrowhead function from "shape" package*

---

## Description

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

## Usage

```
arrowDir(W)
```

## Arguments

| | |
|---|---|
| W | An object of class rWind or a data.frame which should content 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

Karline Soetaert (2017). shape: Functions for Plotting Graphical Shapes, Colors. R package version 1.4.3. https://CRAN.R-project.org/package=shape

## 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 skalar containing wind directions. |
| `wind.speed` | A vector or skalar 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 Felicísimo et al. (2008), see details. |
| `output` | This argument allows to select diferent 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 incrementaly penalized angular deviations from the wind direction" (Felicísimo et al. 2008).

## Value

In "transitionLayer" output, the function returns conductance values (1/cost)to move betwen 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 (asimetric). 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 adviced perform the analysis in a remote computer or a cluster.

## Author(s)

Javier Fernández-López; Klaus Schliep

## References

Felicísimo, Á. M., Muñoz, J., & González-Solis, J. (2008). Ocean surface winds drive dynamics of transoceanic aerial movements. PLoS One, 3(8), e2928.

Jacob van Etten (2017). R Package gdistance: Distances and Routes on Geographical Grids. Journal of Statistical Software, 76(13), 1-21. doi:10.18637/jss.v076.i13

## 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))
```

---

# tidy *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

## Note

Multiple U and V values can be procesed.

## Author(s)

Javier Fernández-López (jflopez@rjb.csic.es)

## 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 coordenates 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

http://allthiswasfield.blogspot.com.es/2016/12/rwind-r-package-released.html

## References

http://oos.soest.hawaii.edu/erddap/info/NCEP_Global_Best/index.html

## 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.ncdc.noaa.gov/data-access/model-data/model-datasets/global-forcast-system-gfs). Wind data are taken from NOAA/NCEP Global Forecast System (GFS) Atmospheric Model colection. 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.htr

## 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 |

| | |
|---|---|
| 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 coordenate in the study area defined by lon1/lon2 and lat1/lat2.

## Note

Longitude coordenates are provided by GFS dataset in 0/360 notation and transformed internaly into -180/180.

## Author(s)

Javier Fernández-López (jflopez@rjb.csic.es)

## 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.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.ncdc.noaa.gov/data-access/model-data/model-datasets/global-forcast-system-gfs). Wind data are taken from NOAA/NCEP Global Forecast System (GFS) Atmospheric Model colection. 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_Bes

## Usage

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

## S3 method for class 'rWind_series'
x[[i, exact = TRUE]]
```

## Arguments

| | |
|---|---|
| `time` | a scalar or vector of POSIXt or Date objects or an character which can transfored 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 coordenate 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 tendences for each coordenate at the study area. Longitude coordenates are provided by GFS dataset in 0/360 notation and transformed internaly into -180/180.

## Author(s)

Javier Fernández-López (jflopez@rjb.csic.es)

## 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)
```

---

# wind.mean *Wind-data mean*

---

## 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/Mean_of_circular_quantities for more details. The U and V componenats 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@rjb.csic.es)

## References

https://en.wikipedia.org/wiki/Cross_product

## See Also

wind.dl

## Examples

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

---

# wind.series *Wind series example*

---

## 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)

### 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:

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

### Source

http://allthiswasfield.blogspot.com.es/2016/12/rwind-r-package-released.html

### References

http://oos.soest.hawaii.edu/erddap/info/NCEP_Global_Best/index.html

### Examples

```
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 logitude 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

```
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@rjb.csic.es)

## See Also

wind.dl

## Examples

```
data(wind.data)

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

wind2raster(wind.data)
```

# Index