

Package ‘adw’

September 14, 2025

Title Angular Distance Weighting Interpolation

Version 0.4.1

Maintainer Panfeng Zhang <zhangpf08@163.com>

Description

The irregularly-spaced data are interpolated onto regular latitude-longitude grids by weighting each station according to its distance and angle from the center of a search radius. In addition to this, we also provide a simple way (Jones and Hulme, 1996) to grid the irregularly-spaced data points onto regular latitude-longitude grids by averaging all stations in grid-boxes.

URL <https://github.com/PanfengZhang/adw>

BugReports <https://github.com/PanfengZhang/adw/issues>

Depends R (>= 4.3.0)

Imports methods, sf, terra, cnmap

License GPL-3

Encoding UTF-8

RoxygenNote 7.3.3

Suggests knitr, rmarkdown, ggplot2, tibble

VignetteBuilder knitr

NeedsCompilation no

Author Panfeng Zhang [aut, cre] (ORCID:
<<https://orcid.org/0000-0001-6084-9231>>),
Guoyu Ren [ctb],
Yun Qin [ctb],
Chenchen Ren [ctb],
Xiang Zheng [ctb]

Repository CRAN

Date/Publication 2025-09-13 22:00:09 UTC

Contents

adw	2
adw_sf	3
adw_sv	4
adw_vector	6
awa	7
points2grid	8
points2grid_sf	9
points2grid_sv	10
points2grid_vector	11
Index	12

adw	<i>Angular Distance Weighting Interpolation.</i>
-----	--

Description

The irregularly-spaced data are interpolated onto regular latitude-longitude grids by weighting each station according to its distance and angle from the center of a search radius.

Usage

```
adw(ds, extent, gridsize = 5, cdd = 1000, m = 4, nmin = 3, nmax = 10)
```

Arguments

ds	a input dataframe which contains the column names of lon, lat, value.
extent	a extent numeric vector (latitude and longitude) of length 4 in the order c(xmin, xmax, ymin, ymax), or a polygon object with class 'sf' (package 'sf'), or a polygon object with class 'SpatVector' (package 'terra'). Assume that the coordinate reference system is WGS1984 (EPSG: 4326).
gridsize	the grid size, i.e. the grid resolution. units: degree.
cdd	correlation decay distance, i.e. the maximum search radius. unit: kilometer. default value: 1000km.
m	is used to adjust the weighting function further, higher values of m increase the rate at which the weight decays with distance. default value 4.
nmin	the minimum number of observation points required to interpolate a grid within the search radius (i.e. cdd); if the number of stations within the search radius (cdd) is less than nmin, a missing value will be generated to fill this grid. default value 3.
nmax	The number of nearest points within the search radius to use for interpolation. default value 10.

Value

a regular latitude-longitude dataframe grid (interpolated values).

References

Caesar, J., L. Alexander, and R. Vose, 2006: Large-scale changes in observed daily maximum and minimum temperatures: Creation and analysis of a new gridded data set. *Journal of Geophysical Research*, 111, <https://doi.org/10.1029/2005JD006280>.

Examples

```
set.seed(2)
dd <- data.frame(lon = runif(100, min = 110, max = 117),
                 lat = runif(100, min = 31, max = 37),
                 value = runif(100, min = -10, max = 10))

head(dd)

# example 1
grd <- adw(dd, extent = c(110, 117, 31, 37), gridsize = 0.5, cdd = 500)
head(grd)

# example 2
hmap <- cnmap::getMap(code = "410000") |> sf::st_make_valid() # return a 'sf' object.
grd <- adw(dd, extent = hmap, gridsize = 0.5, cdd = 500)
head(grd)

# example 3
hmap <- cnmap::getMap(code = "410000", returnClass = "sv") # return a 'SpatVector' object.
grd <- adw(dd, extent = hmap, gridsize = 0.5, cdd = 500)
head(grd)
```

adw_sf

Angular Distance Weighting Interpolation for the extent of 'simple feature'.

Description

The irregularly-spaced data are interpolated onto regular latitude-longitude grids by weighting each station according to its distance and angle from the center of a search radius.

Usage

```
adw_sf(ds, extent, gridsize = 5, cdd = 1000, m = 4, nmin = 3, nmax = 10)
```

Arguments

<code>ds</code>	a input dataframe which contains the column names of lon, lat, value.
<code>extent</code>	a polygon object with class 'sf' (package 'sf'). Assume that the coordinate reference system is WGS1984 (EPSG: 4326).
<code>gridsize</code>	the grid size, i.e. the grid resolution. units: degree.
<code>cdd</code>	correlation decay distance, i.e. the maximum search radius. unit: kilometer. default value: 1000km.
<code>m</code>	is used to adjust the weighting function further, higher values of m increase the rate at which the weight decays with distance. default value 4.
<code>nmin</code>	the minimum number of observation points required to interpolate a grid within the search radius (i.e. cdd); if the number of stations within the search radius (cdd) is less than nmin, a missing value will be generated to fill this grid. default value 3.
<code>nmax</code>	The number of nearest points within the search radius to use for interpolation. default value 10.

Value

a regular latitude-longitude dataframe grid (interpolated values).

References

Caesar, J., L. Alexander, and R. Vose, 2006: Large-scale changes in observed daily maximum and minimum temperatures: Creation and analysis of a new gridded data set. *Journal of Geophysical Research*, 111, <https://doi.org/10.1029/2005JD006280>.

Examples

```
set.seed(2)
dd <- data.frame(lon = runif(100, min = 110, max = 117),
                 lat = runif(100, min = 31, max = 37),
                 value = runif(100, min = -10, max = 10))
head(dd)
hmap <- cnmap::getMap(code = "410000") |> sf::st_make_valid() # return a 'sf' object.
grd <- adw_sf(dd, extent = hmap, gridsize = 0.5, cdd = 500)
head(grd)
```

adw_sv

Angular Distance Weighting Interpolation for the extent of 'SpatVector'.

Description

The irregularly-spaced data are interpolated onto regular latitude-longitude grids by weighting each station according to its distance and angle from the center of a search radius.

Usage

```
adw_sv(ds, extent, gridsize = 5, cdd = 1000, m = 4, nmin = 3, nmax = 10)
```

Arguments

<code>ds</code>	a input dataframe which contains the column names of lon, lat, value.
<code>extent</code>	a polygon object with class 'SpatVector' (package 'terra'). Assume that the coordinate reference system is WGS1984 (EPSG: 4326).
<code>gridsize</code>	the grid size, i.e. the grid resolution. units: degree.
<code>cdd</code>	correlation decay distance, i.e. the maximum search radius. unit: kilometer. default value: 1000km.
<code>m</code>	is used to adjust the weighting function further, higher values of m increase the rate at which the weight decays with distance. default value 4.
<code>nmin</code>	the minimum number of observation points required to interpolate a grid within the search radius (i.e. cdd); if the number of stations within the search radius (cdd) is less than nmin, a missing value will be generated to fill this grid. default value 3.
<code>nmax</code>	The number of nearest points within the search radius to use for interpolation. default value 10.

Value

a regular latitude-longitude dataframe grid (interpolated values).

References

Caesar, J., L. Alexander, and R. Vose, 2006: Large-scale changes in observed daily maximum and minimum temperatures: Creation and analysis of a new gridded data set. *Journal of Geophysical Research*, 111, <https://doi.org/10.1029/2005JD006280>.

Examples

```
set.seed(2)
dd <- data.frame(lon = runif(100, min = 110, max = 117),
                 lat = runif(100, min = 31, max = 37),
                 value = runif(100, min = -10, max = 10))

head(dd)
# example
hmap <- cnmap::getMap(code = "410000", returnClass = "sv") # return a 'SpatVector' object.
grd <- adw_sv(dd, extent = hmap, gridsize = 0.5, cdd = 500)
head(grd)
```

adw_vector

*Angular Distance Weighting Interpolation for the extent of vector.***Description**

The irregularly-spaced data are interpolated onto regular latitude-longitude grids by weighting each station according to its distance and angle from the center of a search radius.

Usage

```
adw_vector(ds, extent, gridsize = 5, cdd = 1000, m = 4, nmin = 3, nmax = 10)
```

Arguments

ds	a input dataframe which contains the column names of lon, lat, value.
extent	a extent numeric vector (latitude and longitude) of length 4 in the order c(xmin, xmax, ymin, ymax).
gridsize	the grid size, i.e. the grid resolution. units: degree.
cdd	correlation decay distance, i.e. the maximum search radius. unit: kilometer. default value: 1000km.
m	is used to adjust the weighting function further, higher values of m increase the rate at which the weight decays with distance. default value 4.
nmin	the minimum number of observation points required to interpolate a grid within the search radius (i.e. cdd); if the number of stations within the search radius (cdd) is less than nmin, a missing value will be generated to fill this grid. default value 3.
nmax	The number of nearest points within the search radius to use for interpolation. default value 10.

Value

a regular latitude-longitude dataframe grid (interpolated values).

References

Caesar, J., L. Alexander, and R. Vose, 2006: Large-scale changes in observed daily maximum and minimum temperatures: Creation and analysis of a new gridded data set. *Journal of Geophysical Research*, 111, <https://doi.org/10.1029/2005JD006280>.

Examples

```
set.seed(2)
dd <- data.frame(lon = runif(100, min = 110, max = 117),
                 lat = runif(100, min = 31, max = 37),
                 value = runif(100, min = -10, max = 10))
head(dd)
```

```
# example
grd <- adw_vector(dd, extent = c(110, 117, 31, 37), gridsize = 0.5, cdd = 500)
head(grd)
```

awa	<i>Area weighted average.</i>
-----	-------------------------------

Description

The large area, or hemispheric, or global averages can be calculated dependent on the area represented by the grid-point or grid-box. The weight of latitude-longitude grid-points-boxes should be the cosine of the latitude of the *i*th grid-point-box.

Usage

```
awa(dat, lat)
```

Arguments

dat	a numeric vector of grid data. The missing values are not allowed.
lat	a latitude numeric vector of grid data. The cosine of latitude is used as the weight coefficient.

Value

a scalar value, i.e the value of area weighted average.

References

Jones, P. D., and M. Hulme, 1996: Calculating regional climatic time series for temperature and precipitation: Methods and illustrations. *Int. J. Climatol.*, 16, 361–377, [https://doi.org/10.1002/\(SICI\)1097-0088\(199604\)16:4<361::AID-JOC53>3.0.CO;2-F](https://doi.org/10.1002/(SICI)1097-0088(199604)16:4<361::AID-JOC53>3.0.CO;2-F).

Examples

```
# set.seed(2)
# dd <- data.frame(lon = runif(100, min = 110, max = 117),
#                 lat = runif(100, min = 31, max = 37),
#                 value = runif(100, min = -10, max = 10))
# grd <- points2grid(dd, extent = c(110, 117, 31, 37), gridsize = 0.5)
# grd <- na.omit(grd)
# awa(grd$value, grd$lat) # area weighted average
```

points2grid

Points were to converted grids using a local gridding method.

Description

the irregularly-spaced data of points are converted onto regular latitude-longitude grids by averaging all stations in grid-boxes.

Usage

```
points2grid(dd, extent, gridsize = 0.5)
```

Arguments

dd	a input dataframe which contains the column names of lon, lat, value.
extent	a extent numeric vector (latitude and longitude) of length 4 in the order c(xmin, xmax, ymin, ymax), or a polygon object with class 'sf' (package 'sf'), or a polygon object with class 'SpatVector' (package 'terra'). Assume that the coordinate reference system is WGS1984 (EPSG: 4326).
gridsize	the grid size, i.e. the grid resolution. units: degree.

Value

a regular latitude-longitude dataframe grid (grid values).

References

Jones, P. D., and M. Hulme, 1996: Calculating regional climatic time series for temperature and precipitation: Methods and illustrations. *Int. J. Climatol.*, 16, 361–377, [https://doi.org/10.1002/\(SICI\)1097-0088\(199604\)16:4<361::AID-JOC53>3.0.CO;2-F](https://doi.org/10.1002/(SICI)1097-0088(199604)16:4<361::AID-JOC53>3.0.CO;2-F).

Examples

```
# set.seed(2)
# dd <- data.frame(lon = runif(100, min = 110, max = 117),
#                  lat = runif(100, min = 31, max = 37),
#                  value = runif(100, min = -10, max = 10))
# head(dd)
#
# # example 1
# grd <- points2grid(dd, extent = c(110, 117, 31, 37), gridsize = 0.5)
# head(grd)
#
# # example 2
# hmap <- cnmap::getMap(code = "410000", return = "sf") |> sf::st_make_valid()
# grd <- points2grid(dd, extent = hmap, gridsize = 0.5)
# head(grd)
#
```

```
# # example 3
# hmap <- cnmap::getMap(code = "410000", return = "sv")
# grd <- points2grid(dd, extent = hmap, gridsize = 0.5)
# head(grd)
```

points2grid_sf

Points were to converted grids using a local gridding method.

Description

the irregularly-spaced data of points are converted onto regular latitude-longitude grids by averaging all stations in grid-boxes.

Usage

```
points2grid_sf(dd, extent, gridsize = 5)
```

Arguments

dd	a input dataframe which contains the column names of lon, lat, value.
extent	a polygon object of simple feature (come from package 'sf'). Assume that the coordinate reference system is WGS1984 (EPSG: 4326).
gridsize	the grid size, i.e. the grid resolution. units: degree.

Value

a regular latitude-longitude dataframe grid (grid values).

References

Jones, P. D., and M. Hulme, 1996: Calculating regional climatic time series for temperature and precipitation: Methods and illustrations. Int. J. Climatol., 16, 361–377, [https://doi.org/10.1002/\(SICI\)1097-0088\(199604\)16:4<361::AID-JOC53>3.0.CO;2-F](https://doi.org/10.1002/(SICI)1097-0088(199604)16:4<361::AID-JOC53>3.0.CO;2-F).

Examples

```
# set.seed(2)
# dd <- data.frame(lon = runif(100, min = 110, max = 117),
#                 lat = runif(100, min = 31, max = 37),
#                 value = runif(100, min = -10, max = 10))
# head(dd)
# # example
# hmap <- cnmap::getMap(code = 410000) |> sf::st_make_valid()
# grd <- points2grid_sf(dd, extent = hmap, gridsize = 0.5)
# head(grd)
```

points2grid_sv	<i>Points were to converted grids using a local gridding method.</i>
----------------	--

Description

the irregularly-spaced data of points are converted onto regular latitude-longitude grids by averaging all stations in grid-boxes.

Usage

```
points2grid_sv(dd, extent, gridsize = 5)
```

Arguments

dd	a input dataframe which contains the column names of lon, lat, value.
extent	a polygon object of SpatVector (from package 'terra'). Assume that the coordinate reference system is WGS1984 (EPSG: 4326).
gridsize	the grid size, i.e. the grid resolution. units: degree.

Value

a regular latitude-longitude dataframe grid (grid values).

References

Jones, P. D., and M. Hulme, 1996: Calculating regional climatic time series for temperature and precipitation: Methods and illustrations. Int. J. Climatol., 16, 361–377, [https://doi.org/10.1002/\(SICI\)1097-0088\(199604\)16:4<361::AID-JOC53>3.0.CO;2-F](https://doi.org/10.1002/(SICI)1097-0088(199604)16:4<361::AID-JOC53>3.0.CO;2-F).

Examples

```
# set.seed(2)
# dd <- data.frame(lon = runif(100, min = 110, max = 117),
#                  lat = runif(100, min = 31, max = 37),
#                  value = runif(100, min = -10, max = 10))
# head(dd)
# # example
# hmap <- cnmap::getMap(code = 410000, returnClass = "sv")
# grd <- points2grid_sv(dd, extent = hmap, gridsize = 0.5)
# head(grd)
```

points2grid_vector *Points were to converted grids using a local gridding method.*

Description

The irregularly-spaced data of points are converted onto regular latitude-longitude grids by averaging all stations in grid-boxes.

Usage

```
points2grid_vector(dd, extent, gridsize = 5)
```

Arguments

dd	a input dataframe which contains the column names of lon, lat, value.
extent	a extent numeric vector (latitude and longitude) of length 4 in the order c(xmin, xmax, ymin, ymax).
gridsize	the grid size, i.e. the grid resolution. units: degree.

Value

a regular latitude-longitude dataframe grid (grid values).

References

Jones, P. D., and M. Hulme, 1996: Calculating regional climatic time series for temperature and precipitation: Methods and illustrations. Int. J. Climatol., 16, 361–377, [https://doi.org/10.1002/\(SICI\)1097-0088\(199604\)16:4<361::AID-JOC53>3.0.CO;2-F](https://doi.org/10.1002/(SICI)1097-0088(199604)16:4<361::AID-JOC53>3.0.CO;2-F).

Examples

```
# set.seed(2)
# dd <- data.frame(lon = runif(100, min = 110, max = 117),
#                  lat = runif(100, min = 31, max = 37),
#                  value = runif(100, min = -10, max = 10))
# head(dd)
# # example
# grd <- points2grid(dd, extent = c(110, 117, 31, 37), gridsize = 0.5)
# head(grd)
```

Index

adw, [2](#)

adw_sf, [3](#)

adw_sv, [4](#)

adw_vector, [6](#)

awa, [7](#)

points2grid, [8](#)

points2grid_sf, [9](#)

points2grid_sv, [10](#)

points2grid_vector, [11](#)