

Package ‘fetchR’

June 26, 2018

Type Package

Title Calculate Wind Fetch

Version 2.1-1

Date 2018-06-15

Imports rgdal, sp, methods, rgeos, plotKML, utils

Suggests knitr, ggplot2, maps, rmarkdown

URL <https://cran.r-project.org/package=fetchR>

BugReports <https://github.com/blasee/fetchR/issues>

Description Wind fetch is the unobstructed length of water over which wind can blow from a certain direction. The wind fetch is typically calculated for many directions around the compass rose for a given location, which can then be incorporated into a larger model (such as the InVEST coastal vulnerability model; http://data.naturalcapitalproject.org/invest-releases/documentation/2_2_0/coastal_vulnerability.html), or simply averaged for a reasonable measure of the overall wind exposure for a specific marine location. The process of calculating wind fetch can be extremely time-consuming and tedious, particularly if a large number of fetch vectors are required at many locations. The ‘fetchR’ package calculates wind fetch and summarises the information efficiently. There are also plot methods to help visualise the wind exposure at the various locations, and methods to output the fetch vectors to a KML file for further investigation.

License GPL-2

RoxygenNote 6.0.1

VignetteBuilder knitr

NeedsCompilation no

Author Blake Seers [aut, cre]

Maintainer Blake Seers <blake.seers@gmail.com>

Repository CRAN

Date/Publication 2018-06-26 05:22:59 UTC

R topics documented:

Fetch	2
fetch	3
fetchR	5
kml,Fetch-method	6
plot,Fetch,missing-method	7
proj4string,Fetch-method	7
spTransform,Fetch,CRS-method	8
summary,Fetch-method	8
Index	10

Fetch	<i>Fetch Class</i>
-------	--------------------

Description

Class to hold Fetch objects.

Details

A Fetch object is essentially a list of [SpatialLines](#) objects.

Slots

names character vector containing the names for each location.

max_dist numeric vector of length 1 containing the maximum distance a fetch vector is allowed.

Extends

Class "list" directly, and class "vector", by class "list".

Note

Fetch objects should only be created using the [fetch](#) function.

fetch	<i>Calculate Wind Fetch</i>
-------	-----------------------------

Description

Wind fetch is the unobstructed length of water over which wind can blow, and it is commonly used as a measure of exposure to wind and waves at coastal sites. The `fetch` function automatically calculates the wind fetch for marine locations within the boundaries of the specified coastline layer. This allows wind fetch to be calculated anywhere around the globe.

Usage

```
fetch(polygon_layer, site_layer, max_dist = 300, n_directions = 9,
      site_names, quiet = FALSE)
```

Arguments

<code>polygon_layer</code>	<code>SpatialPolygons*</code> object where the polygon geometries represent any obstructions to fetch calculations including the coastline, islands and/or exposed reefs.
<code>site_layer</code>	<code>SpatialPoints*</code> object where the point geometries represent the site locations.
<code>max_dist</code>	numeric. Maximum distance in kilometres (default 300). This will need to be scaled manually if the units for the CRS are not 'm'.
<code>n_directions</code>	numeric. The number of fetch vectors to calculate per quadrant (default 9).
<code>site_names</code>	character vector of the site names. If missing, the site names are taken from a column of the data associated with <code>site_layer</code> matching the regular expression <code>^[Nn]ames{0,1}</code> . If there is no such column, then default names are created ('Site 1', 'Site 2', ...).
<code>quiet</code>	logical. Suppress diagnostic messages? (Default FALSE).

Details

The function takes a `SpatialPolygons-class` object (`polygon_layer`) that represents the coastline, surrounding islands, and any other obstructions, and calculates the wind fetch for every specified direction. This is calculated for all the user-defined sites, that are represented as the point geometries in a `SpatialPoints-class` object.

The directions for which the wind fetch are calculated for each site are determined by the number of directions per quadrant (`n_directions`). The default value of 9 calculates 9 fetch vectors per quadrant (90 degrees), or equivalently, one fetch vector every 10 degrees. The first fetch vector is always calculated for the northerly direction (0/360 degrees).

Value

Returns a `Fetch` object.

Note

At least one of the inputs to the `polygon_layer` or `site_layer` arguments must be projected. If one of the inputs are not projected, then it will be transformed to have the same projection as the other. If both are projected, but do not have identical coordinate reference systems (CRS) then `site_layer` will be transformed to the same CRS as `polygon_layer`.

See Also

[spTransform](#) for methods on transforming map projections and datum.

[is.projected](#) for checking whether a spatial object is projected.

[fetchR](#) for an overview of this package with an extensive, reproducible example.

[summary](#), [Fetch-method](#) for summarising the fetch lengths.

Examples

```
# Create the polygon layer -----
#
# This is the layer that represents any obstacles that obstruct wind flow.

# Import map data for the Philippines.
philippines.df = ggplot2::map_data("world", region = "Philippines")

# Create a list for each separate polygon
philippines.list = split(philippines.df[, c("long", "lat")],
                        philippines.df$group)

library(sp)

philippines.Poly = lapply(philippines.list, Polygon)
philippines.Polys = list(Polygons(philippines.Poly, ID = "Philippines"))

# Include CRS information to make it a SpatialPolygons object
philippines.sp = SpatialPolygons(philippines.Polys,
                                proj4string = CRS("+init=epsg:4326"))

# Create the points layer -----
#
# The points layer represents the locations for which the wind fetch needs to
# be calculated.

# We need to calculate wind fetch for the following 3 sites:
sites.df = data.frame(lon = c(124.4824, 125.8473, 124.8416),
                      lat = c(9.167999, 9.751394, 11.478243),
                      site = c("Camiguin Island", "Bucas Grande Island",
                              "Talalora"))

# Create the SpatialPoints object
sites.sp = SpatialPoints(sites.df[, 1:2], CRS("+init=epsg:4326"))
```

```

# Map projection -----
#
# At least one of the polygon or points layers need to be projected to
# calculate wind fetch.

# All these locations lie within the Philippines zone 5 / PRS92, that has
# WGS84 Bounds: 123.8000, 5.3000, 126.7000, 12.7500
# (http://spatialreference.org/ref/epsg/3125/)
# This suggests that this is a suitable map projection.
philippines.proj = spTransform(philippines.sp, "+init=epsg:3125")

# Calculate wind fetch -----
#
# Calculate wind fetch at all the 3 locations for every 10 degrees on the
# compass rose, with a maximum distance for any fetch vector of 300 km.
my_fetch = fetch(philippines.proj, sites.sp, site_names = sites.df$site)
my_fetch

# Return only the summary data frame
summary(my_fetch)

# Transform the fetch vectors back to the original CRS
my_fetch_latlon = spTransform(my_fetch, proj4string(philippines.sp))

# Return the raw data in the original, lat/lon coordinates
my_fetch_latlon.df = as(my_fetch_latlon, "data.frame")
my_fetch_latlon.df

# Plot the wind fetch vectors -----

# Plot the fetch vectors in the projected space...
plot(my_fetch, philippines.proj, axes = TRUE)

# ... or in the original coordinate reference system
plot(my_fetch, philippines.sp, axes = TRUE)

# Output to KML -----
## Not run:

# Save a KML file in the current working directory.
kml(my_fetch, colour = "white")

## End(Not run)

```

fetchR

*Calculate Wind Exposure with the **fetchR** Package*

Description

The **fetchR** package allows for an objective calculation of wind fetch and provides methods to visualise the wind exposure and export the fetch vectors to a KML file.

Details

Fetch is an important measurement in coastal applications. It provides a measurement for the unobstructed length of water that wind from a certain direction can blow over. The higher the wind fetch from a certain direction, the more energy is imparted onto the surface of the water resulting in a larger sea state. Therefore, the larger the fetch, the larger the exposure to wind and the more likely the site experiences larger sea states.

The fetch length from all directions (and from each quadrant) can be averaged to provide an indication of the location's exposure to wind. The **fetchR** package calculates the lengths of wind fetch vectors from all directions, at any given location(s) on Earth, and can provide summaries, visualisations and KML files along with the raw data.

See Also

[fetch](#) for an extensive reproducible example.

`vignette("introduction-to-fetchR")` for a short introduction to **fetchR**.

<http://windfetch.cer.auckland.ac.nz/> for the online web application.

kml,Fetch-method	<i>Write a Fetch object to a KML file</i>
------------------	---

Description

Write a [Fetch](#) object to a KML file. Various aesthetics parameters can be set via colour, alpha, size, shape arguments.

Usage

```
## S4 method for signature 'Fetch'
kml(obj,
     folder.name = normalizeFilename(deparse(substitute(obj, env =
     parent.frame()))),
     file.name = paste(normalizeFilename(deparse(substitute(obj, env =
     parent.frame()))), ".kml", sep = ""), overwrite = FALSE, ...)
```

Arguments

<code>obj</code>	a Fetch object
<code>folder.name</code>	character; folder name in the KML file
<code>file.name</code>	character; output KML file name
<code>overwrite</code>	logical; overwrite the existing file if one exists? Default FALSE
<code>...</code>	additional aesthetics arguments passed to kml_layer.SpatialLines

plot,Fetch,missing-method
Plot a Fetch Object

Description

Plot method for [Fetch](#) objects.

Usage

```
## S4 method for signature 'Fetch,missing'  
plot(x, y, ...)
```

```
## S4 method for signature 'Fetch,SpatialPolygons'  
plot(x, y, ...)
```

```
## S4 method for signature 'SpatialPolygons,Fetch'  
plot(x, y, ...)
```

Arguments

x either a [Fetch](#) object as returned by [fetch](#), or a [SpatialPolygons](#) object.
y can be missing, otherwise same as x.
... further arguments passed to [plot,SpatialLines,missing-method](#).

Details

These plot methods allow for the fetch vectors to be plotted (missing y), or the fetch vectors to be plotted along with a [SpatialPolygons](#) object. If both x and y are supplied, then it does not matter which argument is supplied first.

proj4string,Fetch-method
Retrieve projection attributes for Fetch objects

Description

Retrieve projection attributes for [Fetch](#) objects.

Usage

```
## S4 method for signature 'Fetch'  
proj4string(obj)
```

Arguments

obj [Fetch](#) object

 spTransform,Fetch,CRS-method

spTransform for map projection and datum transformation

Description

spTransform for map projection and datum transformation

Usage

```
## S4 method for signature 'Fetch,CRS'
spTransform(x, CRSobj)
```

```
## S4 method for signature 'Fetch,character'
spTransform(x, CRSobj)
```

Arguments

x	Fetch object to be transformed
CRSobj	object of class CRS, or of class character in which case it is converted to CRS
...	further arguments (ignored)

Value

Fetch object with coordinates transformed to the new coordinate system.

 summary,Fetch-method *Summarise a Fetch Object*

Description

The summary function calculates the average fetch for the separate northerly, easterly, southerly and westerly quadrants. For example, the mean fetch for the northerly component averages over the fetch vectors with directions between 315 (inclusive) and 45 (exclusive) degrees, i.e. the fetch vectors within the interval [315, 45).

Usage

```
## S4 method for signature 'Fetch'
summary(object)
```

```
## S4 method for signature 'Fetch'
show(object)
```


Arguments

object a Fetch object that has been returned by the [fetch](#) function.

Value

The summary function returns a [data.frame](#).

Index

*Topic **package**

fetchR, 5

CRS, 8

data.frame, 9

Fetch, 2, 3, 6–8

fetch, 2, 3, 6, 7, 9

fetchR, 4, 5

fetchR-package (fetchR), 5

is.projected, 4

kml, Fetch-method, 6

kml_layer.SpatialLines, 6

plot, Fetch, missing-method, 7

plot, Fetch, SpatialPolygons-method
(plot, Fetch, missing-method), 7

plot, SpatialPolygons, Fetch-method
(plot, Fetch, missing-method), 7

proj4string, Fetch-method, 7

show, Fetch-method

(summary, Fetch-method), 8

SpatialLines, 2

SpatialPoints, 3

SpatialPolygons, 3, 7

spTransform, 4

spTransform, Fetch, character-method
(spTransform, Fetch, CRS-method),
8

spTransform, Fetch, CRS-method, 8

summary, Fetch-method, 8