

Package ‘meteoForecast’

July 7, 2014

Type Package

Title Retrieve meteorological forecast from Meteogalicia and OpenMeteo services

Version 0.31

Date 2014-07-07

Description Retrieve meteorological forecast from Meteogalicia and OpenMeteo services

URL <http://github.com/oscarperpinan/meteoForecast>

License GPL-3 | file LICENSE

Depends raster, sp, zoo

Imports ncdf, rjson

Suggests ncdf4, rgdal, lattice, rasterVis

Author Oscar Perpinan Lamigueiro [cre, aut], Marcelo Almeida [ctb]

Maintainer Oscar Perpinan Lamigueiro <oscar.perpinan@gmail.com>

NeedsCompilation no

Repository CRAN

Date/Publication 2014-07-07 15:42:08

R topics documented:

meteoForecast-package	2
Forecast variables	3
getPoint	3
getRaster	5

Index	9
--------------	----------

meteoForecast-package *A package to retrieve WRF-NWP forecasts*

Description

The Weather Research and Forecasting (WRF) Model is a numerical weather prediction (NWP) system. NWP refers to the simulation and prediction of the atmosphere with a computer model, and WRF is a set of software for this.

meteoForecast downloads data from the MeteoGalicia and OpenMeteo NWP-WRF services using the NetCDF Subset Service.

Details

`getRaster` and `getRasterDays` get data inside a bounding box and provide a multilayer raster data using the `RasterBrick` class defined in the package `raster`.

`getPoint`, `getPointDays`, and `getPointRuns` get data for a certain location and produce a time series using the `zoo` class.

The default service choice is MeteoGalicia. The use of the OpenMeteo service is still experimental, and some functions are not fully implemented for it.

The MeteoGalicia service produces forecasts for the geographical area comprised approximately between 21.58W and 33.63E, 6.36N and 49.57N.

Author(s)

Oscar Perpiñán, with contributions from Marcelo Almeida

References

<http://www.wrf-model.org/index.php>

<http://www.unidata.ucar.edu/software/thredds/current/tds/reference/NetcdfSubsetServiceReference.html>

<http://www.meteogalicia.es/web/modelos/threddsIndex.action>

https://openmeteoforecast.org/wiki/Main_Page

See Also

[raster zoo](#)

Forecast variables	<i>Forecast Variables</i>
--------------------	---------------------------

Description

Variables available in the MeteoGalicia and OpenMeteo services.

Usage

```
data(varsMG)
data(varsOM)
```

Format

Name, Description and Units of the variables available in the MeteoGalicia (varsMG) and OpenMeteo (varsOM) services.

Source

<http://www.meteogalicia.es/web/modelos/threddsIndex.action>
https://openmeteoforecast.org/wiki/Main_Page

Examples

```
data(varsMG)
data(varsOM)
```

getPoint	<i>NWP forecasts for a location</i>
----------	-------------------------------------

Description

The `getPoint*` functions get outputs of the NWP models run by MeteoGalicia and OpenMeteo for a single location.

Usage

```
getPoint(point, vars = "swflx", day = Sys.Date(), run = "00",
         service = "meteogalicia")

getPointDays(point, vars = "swflx", start = Sys.Date(), end,
             service = "meteogalicia")

getPointRuns(point, var = "swflx", start = Sys.Date() - 1,
             end = Sys.Date(),
             service = "meteogalicia")
```

Arguments

point	Coordinates of the location. It can be a <code>SpatialPoints</code> or a numeric of length 2.
var, vars	Character. The name of the variables to retrieve. Inspect <code>varsMG</code> and <code>varsOM</code> to know what variables are available in each service. <code>getPointRuns</code> only allows for one variable.
day	Date or character
run	Character. The meteogalicia service executes the model at 00UTC and 12UTC. Therefore run can be '00' or '12'.
start	Date or character. First day of the time period to retrieve.
end	Date or character. Last day of the time period to retrieve.
service	Character, which service to use, 'meteogalicia' or 'openmeteo'. Please note that the support for the 'openmeteo' service is still experimental and limited.

Details

These functions download data from the MeteoGalicia and OpenMeteo servers using the NetCDF Subset Service. The result is returned as a zoo time series object, with one or more csv files stored in the temporary folder (as defined by `tempdir()`).

Value

`getPoint` and `getPointDays` produce a zoo time series with a column for each variable included in `vars`.

The time series returned by `getPoint` starts at 01UTC of day if `run = '00'` or 13UTC if `run = '12'`. It spans over 4 days (96 hours) if `run = '00'` or 84 hours if `run = '12'`.

The time series returned by `getPointDays` starts at 01UTC of `start` and finishes at 00UTC of `end + 1`. Each day comprised in the time period is constructed with the forecast outputs corresponding to the 00UTC run of that day. Therefore, only the first 24 values obtained with `getPoint` are used for each day.

The time series returned by `getPointRuns` starts at 01UTC of `start` and finishes at 00UTC of `end + 1`. It has 4 columns, named "D3_00", "D2_00", "D1_00" and "D0_00". The column "D3_00" corresponds to the forecast results produced 3 days before the time stamp of each row, and so on.

Author(s)

Oscar Perpiñán Lamigueiro with contributions from Marcelo Almeida

References

<http://www.meteogalicia.es/web/modelos/threddsIndex.action>

https://openmeteoforecast.org/wiki/Main_Page

See Also[getRaster](#)**Examples**

```
## Not run:
## temperature (Kelvin) forecast from meteogalicia
tempK <- getPoint(c(0, 40), vars = 'temp')
## Cell does not coincide exactly with request
attr(tempK, 'lat')
attr(tempK, 'lon')
## Units conversion
tempC <- tempK - 273

library(lattice)
xyplot(tempC)

## Multiple variables
vars <- getPoint(c(0, 40), vars = c('swflx', 'temp'))
xyplot(vars)

## Time sequence
radDays <- getPointDays(c(0, 40), start = '2013-01-01',
                        end = '2013-01-15')

xyplot(radDays)

## Variability between runs
radRuns <- getPointRuns(c(0, 40), start = '2013-01-01',
                       end = '2013-01-15')
xyplot(radRuns, superpose = TRUE)

## variability around the average
radAv <- rowMeans(radRuns)
radVar <- sweep(radRuns, 1, radAv)
xyplot(radVar, superpose = TRUE)

## End(Not run)
```

getRaster

NWP forecasts for a region

Description

The `getRaster*` functions get outputs of the NWP models run by MeteoGalicia and OpenMeteo for an area.

Usage

```
getRaster(var = "swflx", day = Sys.Date(), run = "00",
          frames = 'complete', box, names,
          remote = TRUE, service = "meteogalicia",
          dataDir = ".", ...)
```

```
getRasterDays(var = "swflx", start = Sys.Date(), end,
              remote = TRUE, dataDir = ".", ...)
```

```
checkDays(start, end, vars, remote = FALSE, service = 'meteogalicia',
           dataDir = '.')
```

Arguments

var, vars	Character. The name of the variable (or variables in checkDays) to retrieve. Inspect varsMG and varsOM to know what variables are available in each service.
day	Date or character
run	Character. The meteogalicia service executes the model at OOUTC and 12UTC. Therefore run can be '00' or '12'.
start	Date or character. First day of the time period to retrieve.
end	Date or character. Last day of the time period to retrieve.
frames	Numeric. It defines the number of hourly forecasts (frames) to retrieve. If frames = 'complete', the full set of frames is downloaded. The meteogalicia service produces 96 hourly forecasts (frames) with run='00' and 84 frames with run='12'.
box	The bounding box. A Extent or an object that can be coerced to that class with extent : a 2x2 matrix (first row: xmin, xmax; second row: ymin, ymax), vector (length=4; order= xmin, xmax, ymin, ymax) or list (with at least two elements, with names 'x' and 'y').
names	Character. Names of the layers of the resulting RasterBrick. If missing, a predefined vector is assigned the combination of day and hour.
remote	Logical. If TRUE (default) data is downloaded from the remote service. If FALSE the RasterBrick is produced with the files available in the local folder.
service	Character, which service to use, 'meteogalicia' or 'openmeteo'. Please note that the support for the 'openmeteo' service is still experimental and limited.
dataDir	Character, path of the folder where files are stored (if remote = 'FALSE')
...	Additional arguments. Not used in getRaster.

Details

getRaster downloads data from the Meteogalicia and OpenMeteo servers using the NetCDF Subset Service. The result is returned as a RasterBrick object, with one or more NetCDF files stored

in the temporary folder (as defined by `tempdir()`). Each frame or layer of the `RasterBrick` corresponds to a certain hour of the forecast.

`getRasterDays` uses `getRaster` to download the results cast each day comprised between `start` and `end` using the 00UTC run. Then it subsets the first 24 frames of each result, and binds them together to produce a `RasterBrick`. Therefore, each frame of this `RasterBrick` is a forecast for an hour of the day when the forecast was cast.

`checkDays` explores a local folder looking for NetCDF files corresponding to a time sequence and a set of variables. It returns a `Date` vector comprising the days with files available for the requested variables. If `remote = TRUE` it only checks that `start` is after 2008-01-01 (first date of the archived forecasts of MeteoGalicía.)

Value

The `getRaster*` functions return a `RasterBrick` with a layer for each hour of the NWP forecast.

The time zone of the time index of this `RasterBrick`, stored in its z slot (accessible with `getZ`) is UTC.

Both MeteoGalicía and OpenMeteo use the Lambert Conic Conformal projection.

Author(s)

Oscar Perpiñán with contributions from Marcelo Almeida.

References

<http://www.meteogalicía.es/web/modelos/threddsIndex.action>

https://openmeteoforecast.org/wiki/Main_Page

https://openmeteoforecast.org/wiki/OPeNDAP_Server

Examples

```
## Not run:
library(rasterVis)

## Retrieve raster data
wrf <- getRaster('temp', '2014-01-25', '00', remote=TRUE)

levelplot(wrf, layers = 10:19)

hovmoller(wrf)
## Day sequence of cloud cover
wrfDays <- getRasterDays('cft',
                        start = '2014-01-01',
                        end = '2014-01-05',
                        box = c(-2, 35, 2, 40))

levelplot(wrfDays, layers = 10:19)

## animation
levelplot(wrfDays, layout = c(1, 1), par.settings = BTCTheme)
```

```
## Hövmoller graphic
hovmoller(wrfDays, par.settings = BTCTheme, contour = TRUE, cuts = 10)
## Extract data at some locations

st <- data.frame(name=c('Almeria', 'Granada', 'Huelva', 'Malaga', 'Caceres'),
                 elev=c(42, 702, 38, 29, 448))

coordinates(st) <- cbind(c(-2.46, -3.60, -6.94, -4.42, -6.37),
                        c(36.84, 37.18, 37.26, 36.63, 39.47)
                        )
proj4string(st) <- '+proj=longlat +datum=WGS84 +ellps=WGS84 +towgs84=0,0,0'

## Extract values for some locations
vals <- extract(wrf, st)
vals <- zoo(t(vals), getZ(wrf))
names(vals) <- st$name

xyplot(vals)

## End(Not run)
```

Index

*Topic **datasets**

Forecast variables, 3

*Topic **package**

meteoForecast-package, 2

*Topic **raster**

getRaster, 5

*Topic **spatial**

getPoint, 3

getRaster, 5

*Topic **time series**

getPoint, 3

checkDays (getRaster), 5

Extent, 6

extent, 6

Forecast variables, 3

getPoint, 3

getPointDays (getPoint), 3

getPointRuns (getPoint), 3

getRaster, 5, 5

getRasterDays (getRaster), 5

getZ, 7

meteoForecast (meteoForecast-package), 2

meteoForecast-package, 2

raster, 2

varsMG, 4, 6

varsMG (Forecast variables), 3

varsOM, 4, 6

varsOM (Forecast variables), 3

zoo, 2