

meteoForecast: predicciones meteorológicas de modelos NWP en R

Oscar Perpiñán Lamigueiro y Marcelo Pinho Almeida

24 de Octubre de 2014

Introducción

Primeros pasos

NWP para una
región:
`getRaster*`

NWP para un
punto: `getPoint*`

Introducción

Primeros pasos

NWP para una región: `getRaster*`

NWP para un punto: `getPoint*`

¿Qué es?

meteoForecast es un paquete que permite obtener predicciones de modelos numéricos meteorológicos producidos por diferentes servicios en formato raster o como series temporales.

Marco de trabajo

El desarrollo de este paquete se enmarca dentro del proyecto europeo **PVCROPS**.

Introducción

Primeros pasos

NWP para una
región:
`getRaster*`

NWP para un
punto: `getPoint*`

Instalación

meteoForecast:
predicciones
meteorológicas de
modelos NWP en
R

Oscar Perpiñán
Lamigueiro y
Marcelo Pinho
Almeida

- La versión de desarrollo está en GitHub:

```
install.packages("devtools")  
devtools::install_github("oscarperpinan/  
meteoForecast")
```

- La versión estable está publicada en [CRAN](#):

```
install.packages('meteoForecast')
```

- Empezamos

```
library(meteoForecast)
```

Introducción

Primeros pasos

NWP para una
región:
getRaster*

NWP para un
punto: getPoint*

Introducción

Primeros pasos

NWP para una región: `getRaster*`

NWP para un punto: `getPoint*`

Variables

- ▶ Cada servicio proporciona un conjunto diferente de variables con sus propios nombres.
- ▶ Su nombre y descripción están disponibles en `varsMG`, `varsGFS`, etc.

```
data(varsMG)
```

- ▶ `grepVar` facilita la tarea de buscar la variable que interesa:

```
grepVar('cloud', service = 'gfs')
```

```
[1] "Temperature_low_cloud_top"           "Pressure_middle_cloud_top"
[3] "Temperature_middle_cloud_top"        "Total_cloud_cover_middle_cloud"
[5] "Cloud_Work_Function"                 "Pressure_low_cloud_bottom"
[7] "Pressure_convective_cloud_top"       "Pressure_convective_cloud_bottom"
[9] "Total_cloud_cover_high_cloud"        "Total_cloud_cover"
[11] "Pressure_low_cloud_top"              "Pressure_high_cloud_top"
[13] "Pressure_middle_cloud_bottom"         "Cloud_mixing_ratio"
[15] "Pressure_high_cloud_bottom"          "Total_cloud_cover_convective_cloud"
[17] "Cloud_water"                         "Total_cloud_cover_entire_atmosphere"
[19] "Total_cloud_cover_low_cloud"         "Temperature_high_cloud_top"
```

meteoForecast:
predicciones
meteorológicas de
modelos NWP en
R

Oscar Perpiñán
Lamigueiro y
Marcelo Pinho
Almeida

Introducción

Primeros pasos

NWP para una
región:
`getRaster*`

NWP para un
punto: `getPoint*`

Servicios

- ▶ Cada función admite un argumento `service` para elegir el servicio.
- ▶ Al cargar el paquete el servicio por defecto es `MeteoGalicia`.

```
mfService()
```

```
[1] "meteogalicia"
```

- ▶ Se puede cambiar (para una sesión) usando `mfService` con el nombre del servicio.

```
mfService('gfs')
```

```
Option service changed to gfs
```

```
mfService('meteogalicia')
```

```
Option service changed to meteogalicia
```

meteoForecast:
predicciones
meteorológicas de
modelos NWP en
R

Oscar Perpiñán
Lamigueiro y
Marcelo Pinho
Almeida

Introducción

Primeros pasos

NWP para una
región:
`getRaster*`

NWP para un
punto: `getPoint*`

Información sobre cada servicio

- `mfProj4` devuelve la proyección (Proj4) de un servicio:

```
mfProj4('nam')
```

```
[1] "+proj=lcc +lat_1=25 +lat_0=25 +lon_0=-95 +k_0=1 +x_0=0 +y_0=0 +a=6367470.21484375 +b=6367470.21484375 +un
```

- `mfExtent` devuelve la extensión de un servicio (usando la clase `Extent` del paquete `raster`):

```
mfExtent('meteogalicia', resolution = 36)
```

```
class      : Extent
xmin       : -49.18259
xmax       : 18.789
ymin       : 24.03791
ymax       : 56.06608
```

meteoForecast:
predicciones
meteorológicas de
modelos NWP en
R

Oscar Perpiñán
Lamigueiro y
Marcelo Pinho
Almeida

Introducción

Primeros pasos

NWP para una
región:
`getRaster*`

NWP para un
punto: `getPoint*`

Introducción

Primeros pasos

NWP para una región: `getRaster*`

NWP para un punto: `getPoint*`

getRaster

meteoForecast:
predicciones
meteorológicas de
modelos NWP en
R

Oscar Perpiñán
Lamigueiro y
Marcelo Pinho
Almeida

Introducción

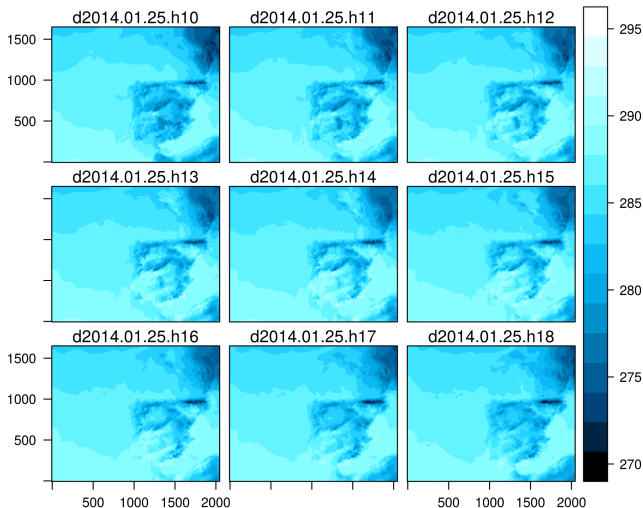
Primeros pasos

NWP para una
región:
`getRaster*`

NWP para un
punto: `getPoint*`

- ▶ `getRaster` descarga ficheros NetCDF con resultados del modelo NWP para un región emitidos un día determinado y los acondiciona en un objeto `RasterBrick`.
- ▶ La extensión, la resolución temporal, y el horizonte de predicción dependen de cada servicio.

```
## temperature at 2m  
wrf <- getRaster(var = 'temp',  
                 day = '2014-01-25',  
                 run = '00')
```



Limitando la región y el periodo temporal

```
cloudNAM <- getRaster('Total_cloud_cover',  
                      day = '2014-10-01',  
                      box = c(-100, -80, 30, 50),  
                      frames = 10,  
                      service = 'nam')
```

```
class      : RasterLayer  
dimensions : 196, 159, 31164  (nrow, ncol, ncell)  
resolution : 0.1537046, 0.1084714  (x, y)  
extent     : -101.1972, -76.75821, 29.35018, 50.61057  (xmin, xmax, ymin, ymax)  
coord. ref.: +proj=longlat +datum=WGS84 +ellps=WGS84 +towgs=0,0,0
```

```
getZ(cloudNAM)
```

```
[1] "2014-10-01 01:00:00 UTC" "2014-10-01 02:00:00 UTC"  
[3] "2014-10-01 03:00:00 UTC" "2014-10-01 04:00:00 UTC"  
[5] "2014-10-01 05:00:00 UTC" "2014-10-01 06:00:00 UTC"  
[7] "2014-10-01 07:00:00 UTC" "2014-10-01 08:00:00 UTC"  
[9] "2014-10-01 09:00:00 UTC" "2014-10-01 10:00:00 UTC"
```

getRasterDay y getRasterDays

- getRasterDay y getRasterDays se basan en getRaster para obtener resultados exclusivamente para un día determinado y una secuencia de días, respectivamente.

```
## cloud cover at low and mid levels
wrfDays <- getRasterDays(var = 'cft',
                          start = '2014-01-01',
                          end = '2014-01-05',
                          box = c(-2, 35, 2, 40))
```

```
class      : RasterStack
dimensions : 65, 41, 2665, 120  (nrow, ncol, ncell, nlayers)
resolution : 12, 12  (x, y)
extent     : 1554, 2046, -6, 774  (xmin, xmax, ymin, ymax)
coord. ref.: +proj=lcc +lat_1=43 +lat_2=43 +lat_0=34.82300186157227 +lon_0=-14.10000038146973 +x_0=536402.34
names      : d2014.01.01.h01, d2014.01.01.h02, d2014.01.01.h03, d2014.01.01.h04, d2014.01.01.h05, d2014.01.01
min values :      0.00000000,      0.00000000,      0.00000000,      0.00000000,      0.00000000,      0.0000
max values :      0.6915230,      0.9363602,      1.0209019,      1.0181180,      0.9741192,      1.009
time       : 2014-01-01 01:00:00 - 2014-01-06 00:00:00 (range)
```

meteoForecast:
predicciones
meteorológicas de
modelos NWP en
R

Oscar Perpiñán
Lamigueiro y
Marcelo Pinho
Almeida

Introducción

Primeros pasos

NWP para una
región:
getRaster*

NWP para un
punto: getPoint*

meteoForecast: predicciones meteorológicas de modelos NWP en R

Oscar Perpiñán
Lamigueiro y
Marcelo Pinho
Almeida

Introducción

Primeros pasos

NWP para una
región:
`getRaster*`

NWP para un
punto: `getPoint*`

Introducción

Primeros pasos

NWP para una región: `getRaster*`

NWP para un punto: `getPoint*`

getPoint

meteoForecast:
predicciones
meteorológicas de
modelos NWP en
R

Oscar Perpiñán
Lamigueiro y
Marcelo Pinho
Almeida

- getPoint descarga resultados emitidos un día determinado por un modelo NWP para un punto y los acondiciona como serie temporal usando la clase zoo.

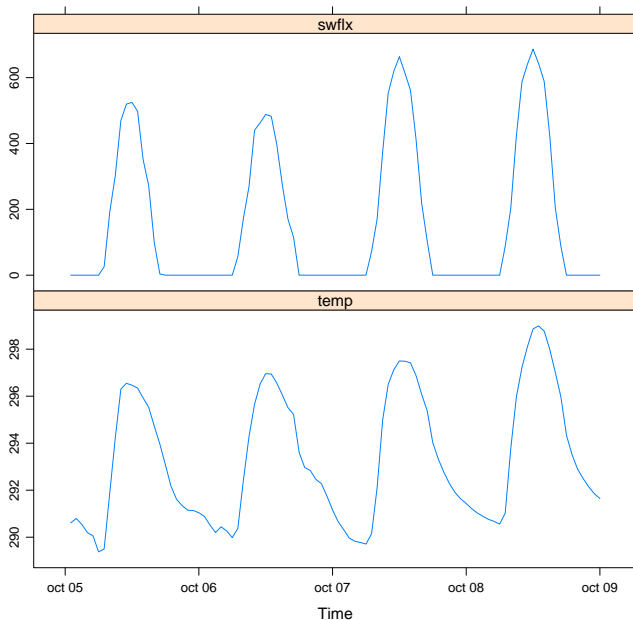
```
## Radiación solar y temperatura
vars <- getPoint(point = c(0, 40),
                 day = Sys.Date() - 1,
                 vars = c('swflx', 'temp'))
attr(vars, 'lat')
attr(vars, 'lon')
```

Introducción

Primeros pasos

NWP para una
región:
getRaster*

NWP para un
punto: getPoint*



getPointDays

meteoForecast:
predicciones
meteorológicas de
modelos NWP en
R

Oscar Perpiñán
Lamigueiro y
Marcelo Pinho
Almeida

- getPointDays usa getPoint para construir una secuencia de días.

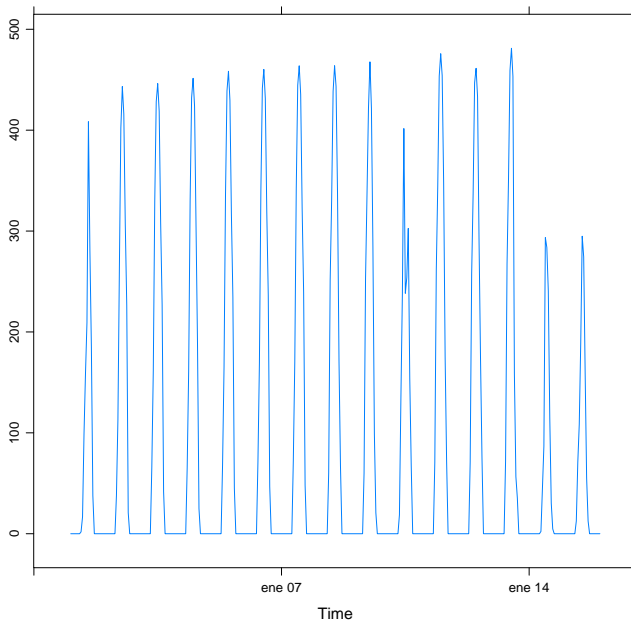
```
radDays <- getPointDays(point = c(0, 40),  
                          var = 'swflx',  
                          start = '2013-01-01',  
                          end = '2013-01-15')
```

Introducción

Primeros pasos

NWP para una
región:
getRaster*

NWP para un
punto: getPoint*



getPointRuns

meteoForecast:
predicciones
meteorológicas de
modelos NWP en
R

Oscar Perpiñán
Lamigueiro y
Marcelo Pinho
Almeida

- getPointRuns usa getPoint para producir una serie temporal de predicciones, donde cada columna indica cuando fue emitida esa predicción.

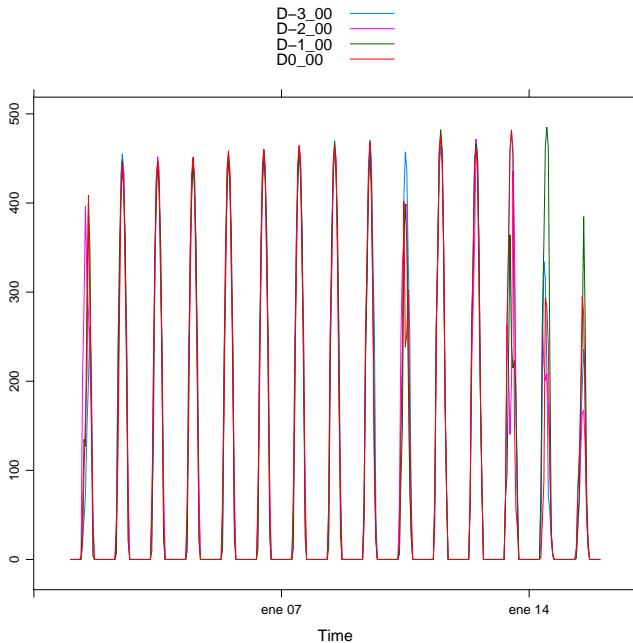
```
## Variability between runs  
radRuns <- getPointRuns(c(0, 40),  
                        var = 'swflx',  
                        start = '2013-01-01',  
                        end = '2013-01-15')
```

Introducción

Primeros pasos

NWP para una
región:
getRaster*

NWP para un
punto: getPoint*



```
## variability around the average  
radAv <- rowMeans(radRuns)  
radVar <- sweep(radRuns, 1, radAv)
```

