

EtaModelW: um pacote R para disponibilização de dados de previsão do tempo gerados pelo CPTEC/INPE

Rubens Rangel Pereira da Silva

Curso de Ciência da Computação – Universidade de Passo Fundo (UPF) – Campus 1
BR 285 - Passo Fundo (RS) - Brasil
124245@upf.br

Abstract: *This paper presents the EtaModelW package, developed in the R programming language, which aims to facilitate access to the weather forecast data generated by CPTEC/INPE through the Eta Model 15km simulation model. The EtaModelW package has functions for accessing weather data as well as generating charts and maps based on this data. With this package, developers and users of the R language can count on an easy-to-use tool for accessing and manipulating the weather forecast data generated by CPTEC/INPE.*

Resumo: Este artigo apresenta o pacote EtaModelW, desenvolvido na linguagem de programação R, o qual tem por objetivo facilitar o acesso aos dados da previsão do tempo gerados pelo CPTEC/INPE por meio do modelo de simulação Eta Model 15km. O pacote EtaModelW possui funções para acesso aos dados de previsão do tempo bem como a geração de gráficos e mapas com base nestes dados. Com este pacote, desenvolvedores e usuários da linguagem R podem contar com uma ferramenta de fácil utilização para o acesso e manipulação dos dados de previsão do tempo gerados pelo CPTEC/INPE.

1 Introdução

O modelo numérico regional de previsão de tempo Eta Model, foi desenvolvido pela Universidade de Belgrado na Sérvia no início da década de setenta, entrando em produção no *National Centers for Environmental Prediction* (NCEP) [Mesinger 2004]. O Centro de Previsão do Tempo e Estudos Climáticos (CPTEC), do Instituto Nacional de Pesquisas Espaciais (INPE), adotou esse modelo em 1996 com o objetivo de fornecer a Previsão Numérica do Tempo sobre a área da América do Sul [Chou et. al 2012]. O modelo Eta utilizado no CPTEC possui versões com grade de resolução de 40, 20, 15 e 5km. Para este trabalho foi utilizado o modelo Eta com resolução de 15km. Os dados gerados pelo modelo Eta para a previsão do tempo são disponibilizados pelo CPTEC/INPE para serem utilizados por outros softwares ou aplicações em formato binário, por protocolo de transferência de arquivos FTP. A partir disso, o pacote EtaModelW foi criado com a intenção de facilitar o acesso aos dados gerados e disponibilizados pelo CPTEC/INPE para desenvolvedores de software que utilizam a linguagem de programação R, tendo como objetivo que suas funções possam facilitar o acesso aos dados de maneira intuitiva e simples.

Desta forma, este artigo foi organizado da seguinte forma: no capítulo 2 são descritos os pacotes R que, assim como o pacote `EtaModelW`, tem por objetivo o acesso a dados da previsão do tempo e de clima, gerados por diversos modelos de simulação computacionais e fornecidos por diversas fontes de dados. Ainda, neste capítulo, é apresentada uma tabela comparativa para as características observadas em cada pacote. No capítulo 3 é descrito como um pacote R é criado, sua natureza, estrutura, além de uma tabela comparativa entre pacotes. No capítulo 4 tem-se a descrição completa do pacote `EtaModelW`, suas funções externas e internas, a forma de uso destas, exemplos e seus resultados após cada execução. No capítulo 5 são apresentadas as conclusões e trabalhos futuros.

2 Trabalhos relacionados

Alguns dos pacotes da linguagem R, com objetivos similares ao `EtaModelW`, foram avaliados para dar embasamento a pesquisa. Foram analisadas características como as diversas funções implementadas por cada pacote e seus respectivos objetivos. Na Tabela 1 são apresentadas, de forma comparativa, as características que mais se destacam nestes pacotes, já incluindo o pacote `EtaModelW` que foi desenvolvido neste trabalho.

- a) **rNOMADS - an interface to the NOAA Operational Model Archive and Distribution System:** O pacote `rNOMADS` automatiza o *download* dos dados de previsão do tempo do *National Oceanic and Atmospheric Administration's Operational Model Archive and Distribution System* (NOMADS) [Bowman and Lees 2015]. Atualmente, oferece suporte a uma significativa variedade de modelos, desde modelos meteorológicos globais a uma altitude de mais de 40 km, modelos climáticos regionais de alta resolução e a modelos de ondas e gelo marinho. O pacote é o primeiro a fornecer acesso a dados de modelos operacionais praticamente em tempo real, ou seja, próximo de quando estes dados são produzidos, o conhecimento do estado atual e futuro da atmosfera e oceanos é fundamental para muitas áreas do conhecimento como as geociências, por exemplo, dispersão de cinzas vulcânicas, infra-sons, propagação e modelagem de inundação por tempestades. Os dados da previsão do tempo utilizados pelo pacote são gerados quatro vezes ao dia a cada seis horas pelo modelo de simulação *Global Forecast System* (GFS). A leitura de dados pode ser feita de duas formas: ler dados em ASCII direto do servidor usando o sistema de DODS-GRADS, ou através de *download* de arquivos em formatos GRIB1 ou GRIB2.
- b) **rdwd - an R package to select, download and read climate data from the German Weather Service:** Desenvolvido para obter dados do *Climate Data Center* (CDC) do *German Weather Service* (DWD). Baseado no conceito de índices que ajudam a selecionar, baixar e processar os arquivos de um servidor FTP que disponibiliza

gratuitamente milhares de séries climáticas geradas nas estações meteorológicas alemãs [Boessenkool 2019]. A proposta deste pacote é facilitar o acesso de cientistas a estas séries de dados, mesmo quando estes não estiverem familiarizados com a linguagem R.

- c) **rnoaa - NOAA Weather Data from R:** O pacote é um cliente da API do *National Oceanic and Atmospheric Administration* (NOAA), que possui várias fontes de dados, agrupadas primariamente e denominadas de `datasets`. Para cada recurso da API; como dados, tipos de dados, categorias de tipos dados, localização, categorias de localização e estações, existem funções específicas para sua utilização. Dentre os `datasets`, o qual se destaca por possuir uma maior quantidade de funções de consulta é o `dataset` do *National Climatic Data Center* (NCDC); suas funções são reconhecidas por serem precedidas por `ncdc_*` [Chamberlain et. al 2019]. Este `dataset` é o responsável pela previsão numérica do tempo neste pacote, utilizando dados originados pelo modelo de simulação *Global Ensemble Forecast System* (GEFS). O pacote possui aproximadamente uma centena de funções, utilizando protocolos HTTP e FTP para a transferência de dados, que permitem além do acesso à previsão do tempo, o acesso à informações climáticas de mais de um século de observações. O pacote possui ainda três `datasets` implícitos com metadados sobre localizações e tempestades, a descontinuação do fornecimento de dados de uma determinada fonte do NOAA pode acontecer por parte do Governo Americano e afetar significativamente as funções do pacote.
- d) **weathercan - Download and format weather data from Environment and Climate Change Canada:** Pacote R para acesso aos dados históricos de meteorologia do *Environment and Climate Change Canada* (ECCC), um serviço público e gratuito do Governo do Canadá, que fornece dados de aproximadamente oito mil estações climáticas registrados desde 1840 até a presente data, em frequências mensais, diárias e horárias. Tem como principal característica permitir a montagem de conjuntos de dados a partir da consulta a informações de múltiplas estações meteorológicas, isso combinado com intervalos de tempo e de frequência dos dados, além da capacidade de interpolação destes [LaZerte and Albers 2018]. O pacote possui ainda `datasets` com metadados da fonte de dados do pacote, de estações de captação e de variáveis da previsão do tempo.
- e) **darksky - R interface to the Dark Sky API:** O pacote `darksky` fornece acesso à *Dark Sky API*, disponibilizando dados das condições meteorológicas globais atuais e da previsão do tempo [Rudis 2019]. Esta API por sua vez é responsável por acessar dados de várias outras fontes como o NOAA. Os períodos da previsão são: sete dias com frequência diária, dois dias com frequência horária e por uma hora com frequência de minuto em minuto.
- f) **weatherData - Get Weather Data from the Web:** O pacote `weatherData` possui funções que ajudam na obtenção de dados meteorológicos históricos tendo como fonte a

Weather Underground API [Narasimhan 2019]. A partir de uma localização e de um intervalo de datas, o pacote facilita a busca por dados meteorológicos como temperatura, pressão, entre outros. O pacote possui ainda `datasets` com metadados sobre estações climatológicas e dados climáticos de algumas cidades em anos específicos. A empresa proprietária da *Weather Underground API* foi adquirida pela IBM em 2012, sendo os serviços gratuitos encerrados ao final de 2018, enquanto os serviços pagos foram migrados para outras API's da IBM.

- g) **RNCEP - Obtain, Organize, and Visualize NCEP Weather Data:** O Pacote RNCEP contém funções para recuperar, organizar e visualizar os dados meteorológicos provenientes do *National Centers for Environmental Prediction* (NCEP) e do *National Center Atmospheric Research* (NCAR), ambos parte do NOAA. Os dados podem ser filtrados por intervalos de tempo e localização. Com o RNCEP é possível obter a visualização destes dados por meio de mapa, assim como simular trajetórias de voos de acordo com dados definidos pelo usuário [Kemp et. al 2017].
- h) **rwunderground - R Interface to Weather Underground API:** O pacote `rwunderground` possui funções para obter dados meteorológicos históricos e da previsão do tempo a partir da *Weather Underground API*. Algumas das variáveis da previsão do tempo que podem ser obtidas: temperatura mínima e máxima, precipitação, chuva, neve, velocidade máxima e média do vento, umidade mínima, média e máxima para os próximos três ou dez dias, todas estas variáveis, e ainda ponto de orvalho, direção do vento, índice radiação UV, sensação de vento, índice de calor e sensação térmica com previsão horária para as próximas vinte e quatro horas ou dez dias. Além disso, é possível obter informações sobre o nascer e pôr do sol, condições de maré, imagens de satélite, alertas meteorológicos e de furacões, fase da Lua e temperaturas históricas para os últimos trinta anos [Shum 2019]. O pacote também possui um `dataset` com metadados de aeroportos distribuídos por todos os continentes. Este pacote passa pela mesma situação com relação à sua fonte de dados, assim como ocorre com o pacote `weatherData`, sua API foi adquirida pela IBM, os serviços gratuitos foram encerrados e os pagos foram migrados para outras API's.
- i) **riem - Accesses Weather Data from the Iowa Environment Mesonet:** Pacote que acessa informações climáticas da *Iowa Environmental Mesonet* da *Iowa State University*, fornece através de uma API dados históricos da rede de estações *Automated Surface Observing System*, conhecida como rede ASOS, e da rede de estações *Automated Weather Observing System*, conhecida como rede AWOS [Salmon and Anderson 2016]. O pacote fornece apenas três funções básicas, uma consulta às redes de estações, uma consulta às estações de uma determinada rede de estações e uma consulta diretamente aos dados de

uma estação em um determinado período de tempo, nesta consulta é possível obter dados até a hora da execução da consulta.

- j) **SkyWiseDataTransferR - SkyWiseNetCDFR:** Pacotes que funcionam em conjunto e extraem dados das API's da plataforma *WeatherOps*, esta por sua vez, fornece serviços de apoio à decisão em negociações de commodities nos ramos da agricultura, energia e transporte. O primeiro pacote é responsável pela interação e transferência de dados. O segundo pacote é responsável pela conversão dos dados que são disponibilizados pela API no formato *NetCDF*, um formato otimizado para dados científicos que tem como características principais ser portátil e escalável, para os tipos de dados da linguagem R, o pacote ainda é responsável pela consulta à API [Thompson 2017].

A seguir é apresentada uma tabela comparativa entre as características principais dos pacotes descritos anteriormente.

Tabela 1: Comparativo das características de alguns pacotes em R que acessam e disponibilizam dados de previsão do tempo e de clima.

A - rNOMADS; B - rdwd; C - moaa; D - weathercan; E - darksky; F - weatherData; G - RNCEP; H - rwunderground; I - riem;
 J - SkyWiseDataTransfer/SkyWiseNetCDFR;K - EtaModelW.

Características	Pacotes R												
	A	B	C	D	E	F	G	H	I	J	K		
Previsão	Tempo	●		●		●			●		●	●	
	Clima		●	●	●					●			
	Período	16 dias	100 a.	16d/110a	180 a.	7 dias	-	70 a.	10d/30a	46 a.	-	8 dias	
	Coordenadas Geográficas	●	●	●	●	●		●	●		●	●	
	Fonte	NOMADS	DWD	NOAA	ECCC	DarkSky API	WU	NCEP-NCAR	WU	IEM	WO API	INPE	
	Open source	●	●	●	●	●	●	●	●	●	●	●	
	Versão	2.4.1	1.0.2	0.8.5	0.2.8	1.3.0	0.6.0	1.0.8	0.1.8	0.1.1	0.1.0	1.0.0	
	Última Atualização	03/2019	05/2019	01/2019	10/2018	09/2017	06/2017	01/2017	05/2018	09/2016	-	06/2019	
	Repositório: 1 CRAN 2 GitHub	1 2	1 2	1 2	1 2	1 2	1 2	1 2	1 2	1 2	2	2	
	Sistema Operac.	Linux	●	●	●	●	●	●	●	●	●	●	●
		Windows	●	●	●	●	●	●	●	●	●	●	●
OS X		●	●	●	●	●	●	●	●	●	●	●	
Texto e/ou tabelas		●	●	●	●	●	●	●	●	●	●	●	
Formato da saída	Gráficos	●		●								●	
	Mapas							●				●	

3 Criação de pacotes em R

R é uma linguagem de programação com um conjunto integrado de recursos de software para manipulação, cálculo, computação estatística e visualização gráfica de dados [Venables, Smith and R Core Team 2019]. A linguagem é altamente extensível por meio de módulos adicionais denominados de pacotes. “Em R, a unidade fundamental do código compartilhável é o pacote. Um pacote reúne código, dados, documentação e testes e é fácil de compartilhar com outras pessoas” [Wickham 2015]. Para usá-los, eles devem ser disponibilizados em um lugar onde a linguagem ou interface de desenvolvimento da linguagem como o RStudio, os possa encontrar.

O sistema de gerenciamento de pacotes R é conhecido como *The Comprehensive R Archive Network* (CRAN), que além de manter os próprios pacotes contém metadados tais como dependências e informações de licença relevantes para o processo de instalação. O CRAN fornece principalmente arquivos com códigos fontes, arquivos binários e uma riqueza de documentação, como manuais, tutoriais e referências [Hornik 2012].

Um pacote é, por natureza, um diretório de arquivos que estendem as funções básicas da linguagem de programação R. A estrutura mais simples possível de um pacote R consiste basicamente em um diretório raiz que contém os arquivos DESCRIPTION, que por sua vez contém informações básicas sobre este. O arquivo NAMESPACE, que tem a definição das funções que estarão disponíveis para usuários e as funções de outros pacotes que o pacote R necessitará usar e o subdiretório R/, onde encontramos os códigos-fonte deste. Ainda é possível ter no diretório raiz os subdiretórios data, demo, exec, inst, man, po, src, tests, tools e vignettes, e os arquivos INDEX, configure, cleanup, LICENSE e NEWS. Além destes ainda é possível ter junto ao diretório raiz os arquivos INSTALL com instruções de instalação do pacote, README/README.md e ChangeLog que poderão ser úteis para usuários e desenvolvedores [R Development Core Team, 2019]. Um pacote pode assumir diferentes estados em sua vida útil, Source Package, Bundled Package, Binary Package, Installed Package e In-Memory Package. A Figura 1 apresenta uma estrutura bastante comum de um pacote.



Figura 1: Um pacote R é uma convenção para organizar arquivos em diretórios, esta figura mostra como estão dispostas as sete partes mais comuns um pacote R.

Fonte: <https://github.com/rstudio/cheatsheets/raw/master/package-development.pdf>

4 O pacote EtaModelW

O pacote `EtaModelW` foi desenvolvido para ser disponibilizado de forma pública no repositório remoto `GitHub`, seguindo os padrões de desenvolvimento da linguagem `R` exigidos pelo `CRAN`, conforme documentação oficial do repositório. Ele tem como objetivo principal a disponibilização de dados referentes à previsão do tempo geradas e fornecidas diariamente pelo `CPTEC/INPE`. Dessa forma foram criadas funções que retornam ao usuário informações fundamentais da previsão do tempo, a partir da passagem de parâmetros. Estas funções quando disponibilizadas por meio de um pacote facilitam o desenvolvimento de outros softwares, sistemas e até outros pacotes, abstraindo implementações complexas que demandam uma quantidade de tempo significativa do desenvolvedor. Para a construção deste foram utilizados os seus pares, `jsonlite`, `ggplot2`, `RCurl`, `XML`, `leaflet`, `magrittr`, `methods`, `stats` e `stringr`, além daqueles incluídos por padrão na distribuição da linguagem `R`, `utils` e `base`. O pacote `EtaModelW` utiliza dados de uma única fonte, `CPTEC/INPE`, e de um único modelo de simulação, `Eta Model 15 km`.

Para atingir os objetivos acima descritos, o pacote possui onze funções:

- a) `info_EtaModel()` - função visível ao usuário que retorna metadados das variáveis da previsão do tempo disponíveis para a data corrente, além de dados que se referem ao próprio modelo de simulação;
- b) `getWeatherData()` - função visível ao usuário que retorna um subconjunto de dados de uma variável da previsão do tempo, para uma determinada coordenada, hora e dia;
- c) `coord_checker()` - função não visível ao usuário que verifica se as coordenadas informadas pelo usuário estão dentro de um intervalo de coordenadas válidas do modelo;
- d) `coord_adjuster()` - função não visível ao usuário que ajusta a coordenada informada como parâmetro pelo usuário, para a coordenada mais próxima e válida pelo modelo;
- e) `binfile_downloader()` - função não visível ao usuário que a partir de uma determinada variável da previsão do tempo e a data corrente, realiza exclusivamente o *download* de arquivo binário disponibilizado via protocolo `FTP` pelo `CPTEC/INPE`;
- f) `binfile_converter()` - função não visível ao usuário que a partir de uma determinada variável da previsão do tempo e a data corrente, realiza a conversão de arquivo binário em um `Data Frame`;
- g) `download_forecast()` - função visível ao usuário que controla a execução das funções `binfile_downloader()` e `binfile_converter()`. A função cria um arquivo no formato `.RDA` com os dados da variável da previsão do tempo solicitada;
- h) `plot_variable()` - função que gera um gráfico de uma variável da previsão do tempo, com base em uma coordenada e em um intervalo de tempo;
- i) `plot_thp()` - função que gera gráfico único com as variáveis, temperatura a 2 metros da superfície, umidade do ar e a precipitação pluviométrica, com base em uma coordenada e em um intervalo de tempo;

- j) `plot_meteogram()` - função que gera um meteograma com as variáveis, temperatura a 2 metros da superfície, umidade do ar, precipitação pluviométrica, radiação solar e ventos a 10 metros, com base em uma coordenada e em um intervalo de tempo;
- k) `plot_map()` - função que gera um mapa com um conjunto de territórios de estados brasileiros ou todo o território brasileiro, em um determinado dia e hora para uma determinada variável da previsão do tempo.

4.1 Informações sobre o modelo de simulação da previsão do tempo.

A função do pacote `EtaModelW` que fornece os metadados do modelo, bem como as variáveis da previsão do tempo disponíveis para a data corrente é descrita a seguir:

- a) `info_EtaModel()` - esta função retorna informações referentes ao modelo de simulação computacional do pacote e as variáveis da previsão do tempo disponíveis, que auxiliarão na utilização correta da função `getWeatherData()`. Dentre estas informações tem-se: resolução do modelo, o intervalo de latitudes e longitudes válidas para o modelo, as variáveis disponíveis, e o período de previsão do tempo disponível. O código fonte da função está disponível no arquivo `weather.R` do subdiretório `R/` do pacote. A execução e o resultado dessa função podem ser vistos na Figura 2.

```

~/Área de Trabalho/tcc2/EtaModelW/ →
> info_EtaModel()
CPTEC/INPE
Eta Model 15km Brazil
AREA COVERED IN THE MODEL:
--> LONGITUDE: -75.05 to -33.95
--> LATITUDE : -35.05 to 5.90
LAST FORECAST AVAILABLE: 2019-06-26

FORECAST PERIOD (8 days - 182 hours)
--> Initial forecast: 2019062600 (YearMonthDayHour)

Available variables:
variable  description                                     unit.measure
albe      SURFACE MIDDAY ALBEDO [%]                     %
clsf      LATENT HEAT FLUX FROM SURFACE [W/m2]          W/m2
cssf      SENSIBLE HEAT FLUX FROM SURFACE [W/m2]        W/m2
dp2m      2 METRE DEWPOINT TEMPERATURE [K]             K
ghfl      TIME AVE GROUND HT FLX [W/m2]                W/m2
hinv      HIGH CLOUD COVER [0-1]                       0-1
lwnv      LOW CLOUD COVER [0-1]                        0-1
mdnv      MEDIUM CLOUD COVER [0-1]                    0-1
mntp      MINIMUM TEMPERATURE [K]                      K
mxtp      MAXIMUM TEMPERATURE [K]                      K
neve      SNOWFALL [Kg/m2/day]                         Kg/m2/day
oces      UPWARD SHORT WAVE AT GROUND [W/m2]           W/m2
ocis      DOWNWARD SHORT WAVE AT GROUND [W/m2]         W/m2

```

Figura 2: Chamada e retorno da função `info_EtaModel()`, destacado em vermelho, os metadados do modelo, e logo abaixo as variáveis da previsão do tempo disponíveis.

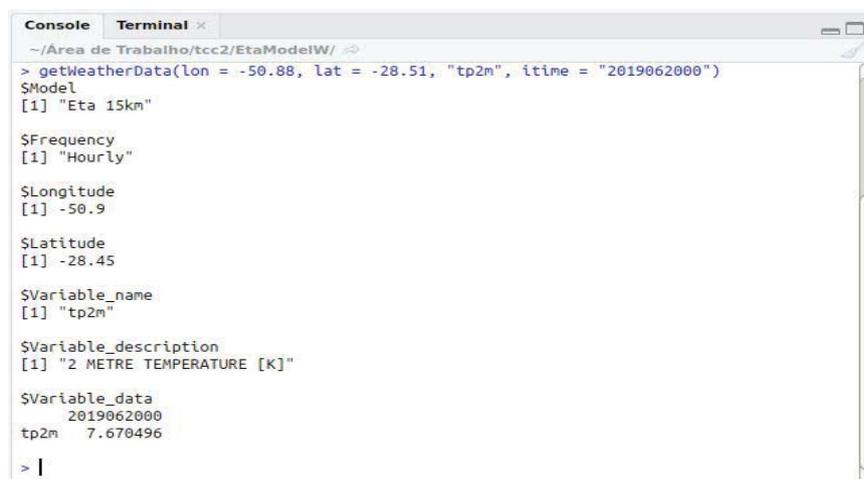
4.2 Acesso aos dados gerados pelo modelo

O acesso aos dados da previsão do tempo através das funções do pacote, são descritas em maiores detalhes a seguir:

- a) `getWeatherData()` - essa função é visível ao usuário e recebe como parâmetros, `lon`, `lat`, `variable`, `itime` e `ftime`. Os parâmetros `lat` e `lon`, representam a latitude e longitude e são obrigatórias. Pelo fato de a previsão ser gerada para um ponto de grade, estas coordenadas serão ajustadas caso necessário, utilizando funções auxiliares do pacote, para uma coordenada válida pelo modelo de simulação. O parâmetro `variable` faz referência à variável da previsão do tempo que se deseja obter os dados, deve ser uma cadeia de quatro caracteres conforme os valores retornados pela função `info_EtaModel()` no campo `variable`.

Os parâmetros `itime` e `ftime` representam a data/hora inicial e final, seus formatos devem ser uma cadeia de caracteres, onde os quatro primeiros se referem ao ano, o quinto e sexto se referem ao mês, o sétimo e oitavo se referem ao dia e por fim os dois últimos caracteres se referem à hora. Veja-se como exemplo a cadeia de caracteres "2019032704": tem-se neste exemplo a seguinte data e hora: 27/03/2019 às 4hrs. A função permite que um destes dois parâmetros seja omitido ou ambos, quando informado apenas `itime` tem como retorno os dados apenas daquela data e hora específica, ao informar ambos os parâmetros obtém-se então o retorno dos dados do intervalo de dias e horas informados. Por fim, ao omitir ambos os parâmetros tem-se como retorno da função todos os dados da previsão numérica do tempo para uma variável específica no período de oito dias à contar da data de *download* dos dados do CPTEC/INPE. Na Figura 3 é apresentado o retorno de dados pela função a partir do seguinte código e exemplo:

```
getWeatherData(lon = -50.88, lat = -28.51, "tp2m", itime = "2019062000")
```



```
Console Terminal x
~/Área de Trabalho/tcc2/EtaModelW/ >
> getWeatherData(lon = -50.88, lat = -28.51, "tp2m", itime = "2019062000")
$model
[1] "Eta 15km"

$Frequency
[1] "Hourly"

$Longitude
[1] -50.9

$Latitude
[1] -28.45

$Variable_name
[1] "tp2m"

$Variable_description
[1] "2 METRE TEMPERATURE [K]"

$Variable_data
  2019062000
tp2m  7.670496

> |
```

Figura 3: Chamada e retorno da função `getWeatherData()`, um objeto do tipo List é apresentado, onde o último atributo deste contém os dados solicitados à função.

Neste exemplo foi informado uma coordenada que precisou ser ajustada para um ponto válido mais próximo. Uma data inicial foi descrita, mas a data final não, sendo assim, apenas os dados da data inicial foram retornados pela função.

4.3 Funções auxiliares

Foram implementadas cinco funções que dão suporte às demais funções, tendo como objetivos gerais o *download* de arquivo binário via protocolo FTP do CPTEC/INPE, a conversão do arquivo binário em um `Data Frame` e por fim a gravação de um arquivo no formato `.RDA` no subdiretório `data/` do diretório de trabalho. Essas funções estão implementadas no arquivo `utils.R` do subdiretório `R/`.

As duas funções a seguir tem o papel de validar e ajustar respectivamente a coordenada informada como parâmetro na função `getWeatherData()`. A validação é necessária uma vez que o modelo Eta tem uma área de abrangência limitada, o ajuste é necessário já que nem todas as coordenadas da área de abrangência do modelo são de fato utilizadas por este.

- a) `coord_checker()` - essa função auxiliar é responsável por verificar se as coordenadas informadas pelo usuário como parâmetro estão inseridas no intervalo de coordenadas de abrangência do modelo Eta. A função não possui retorno, apenas pode encerrar a aplicação caso encontre coordenadas fora do intervalo de coordenadas abrangidas pelo modelo.
- b) `coord_adjuster()` - uma vez que a função `coord_checker()` validou a coordenada, essa função auxiliar é responsável pelo ajuste necessário sobre a coordenada informada como parâmetro, quando a função recebe esta, ela busca um ponto de coordenada válido pelo modelo Eta mais próximo daquela informada como parâmetro e a retorna.

As três funções a seguir tem o objetivo de fazer o *download* do arquivo binário disponibilizado pelo CPTEC/INPE e transformá-lo em um arquivo no formato `.RDA` para ficar disponível ao usuário, trabalham em conjunto e são gerenciadas a partir da última função.

- c) `binfile_downloader()` - função auxiliar que tem o objetivo exclusivo de definir os requisitos para o *download* do arquivo binário e realizar de fato o *download*. Recebe da função `download_forecast()` os parâmetros `variable` e `curr_date`, que se referem à variável da previsão do tempo e a data corrente respectivamente, a função salva o arquivo binário no subdiretório `data/` do diretório de trabalho.
- d) `binfile_converter()` - função auxiliar que tem o objetivo exclusivo de converter o arquivo binário em um `Data Frame`. Recebe da função `download_forecast()` os parâmetros `variable` e `curr_date`, que se referem à variável da previsão do tempo e a data corrente respectivamente, esta retorna o `dataframe` para a função que a invocou com os dados da previsão do tempo para a variável em questão, para o período de oito dias com frequência horária.

- e) `download_forecast()` - função visível ao usuário, tem o objetivo de gerenciar o *download* do arquivo binário, a conversão deste em um `dataframe` e sua gravação em um arquivo no formato `.RDA`, recebe o parâmetro `variable` que se refere à variável da previsão do tempo que se deseja, a função não tem retorno de dados, salva o arquivo `.RDA` no subdiretório `data/` do diretório de trabalho.

4.4 Funções de visualização de dados

Foram implementadas quatro funções de visualização de dados no pacote: dois gráficos, um meteograma e um mapa. Algumas funções têm a visualização dos dados para apenas uma variável da previsão do tempo, enquanto outras fazem uma relação entre mais de uma variável, o que faz com que o usuário tenha o cuidado de ter disponível no subdiretório `data/` do diretório de trabalho os arquivos `.RDA` necessários para o bom funcionamento das funções.

- a) `plot_variable()` - função visível ao usuário que gera um gráfico para apenas uma variável da previsão do tempo. Os parâmetros da função, `lon`, `lat`, `variable`, `itime` e `ftime`, são respectivamente, a longitude e latitude, a variável da previsão do tempo desejada e a data/hora inicial e final, sendo estes dois últimos parâmetros opcionais uma vez que estão definidos estaticamente na função como nulos. Na figura 4 é apresentado o gráfico gerado pela função a partir do seguinte código e exemplo:

```
plot_variable(lon = -50.88, lat = -28.51, variable = "tp2m", itime = "2019061400", ftime = "2019061423")
```

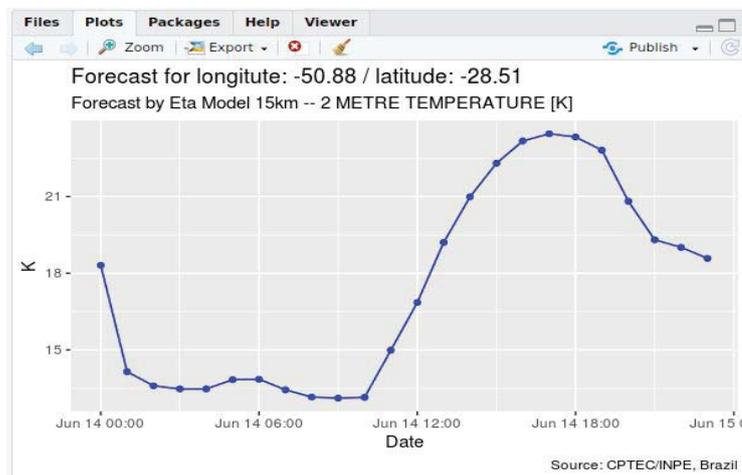


Figura 4: Gráfico resultado da função `plot_variable()`.

O gráfico mostra a variação de valores da variável da previsão do tempo no eixo vertical, e ao longo do tempo no eixo horizontal.

- b) `plot_thp()` - função visível ao usuário que gera um gráfico único com as variáveis da previsão do tempo: temperatura do ar a 2 metros da superfície, umidade relativa do ar e precipitação pluviométrica. Os parâmetros da função, `lon`, `lat`, `itime` e `ftime` são respectivamente, a longitude e latitude, a data/hora inicial e final, sendo estes dois últimos parâmetros opcionais uma vez que estão definidos estaticamente na função como nulos. Na figura 5 é apresentado o gráfico único gerado pela função a partir do seguinte código e exemplo:

```
plot_thp(lon = -50.82, lat = -28.55, itime = "2019042000", ftime = "2019042023")
```

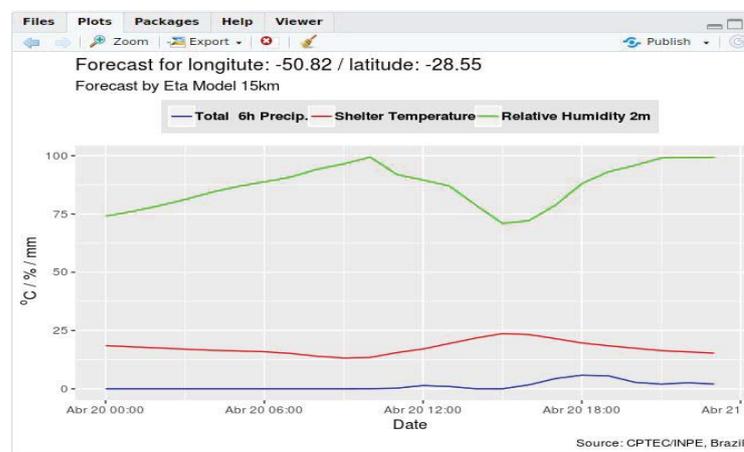


Figura 5: Gráfico resultado da função `plot_thp()`.

O gráfico mostra a variação de valores das três variáveis da previsão do tempo no eixo vertical, e ao longo do tempo no eixo horizontal.

- c) `plot_meteogram()` - função visível ao usuário que gera um meteograma com cinco variáveis da previsão do tempo: temperatura do ar a 2 metros da superfície, umidade do ar, precipitação pluviométrica, radiação solar e ventos a 10 metros da superfície. Os parâmetros da função: `lon`, `lat`, `itime` e `ftime` são respectivamente, a longitude e latitude, a data/hora inicial e final, sendo estes dois últimos parâmetros opcionais uma vez que estão definidos estaticamente na função como nulos. Na figura 6 é apresentado o meteograma gerado pela função a partir do seguinte código e exemplo:

```
plot_meteogram(lon = -50.82, lat = -28.55, itime = "2019042000", ftime = "2019042023")
```

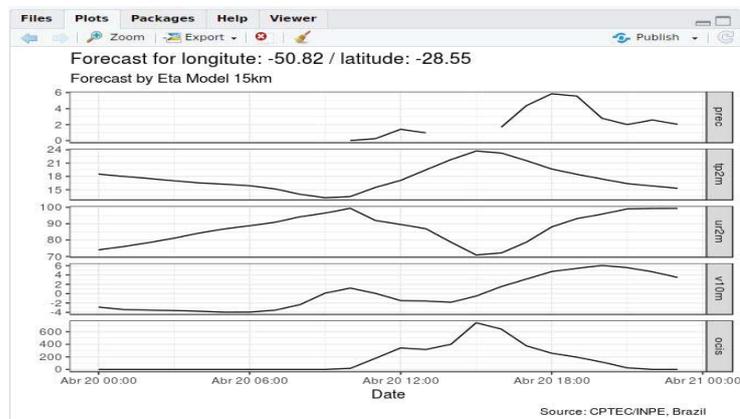


Figura 6: Meteograma resultado da função `plot_meteorogram()`.

O meteograma mostra a variação de valores das cinco variáveis da previsão do tempo no eixo vertical, e ao longo do tempo no eixo horizontal.

- d) `plot_map()` - função visível ao usuário que gera um mapa do território brasileiro, ou de um conjunto de territórios de estados brasileiros, ou do território de apenas um estado brasileiro, para apenas uma variável da previsão do tempo, em uma determinada data e hora. Os parâmetros da função: `variable`, `itime` e `state_br` são respectivamente, a variável da previsão do tempo, a data/hora específica e o(s) território(s) desejado(s). Na figura 7 é apresentado o mapa gerado pela função a partir do seguinte código e exemplo:

```
plot_map(variable = "tp2m", itime = "2019061400", state_br = "RS")
```

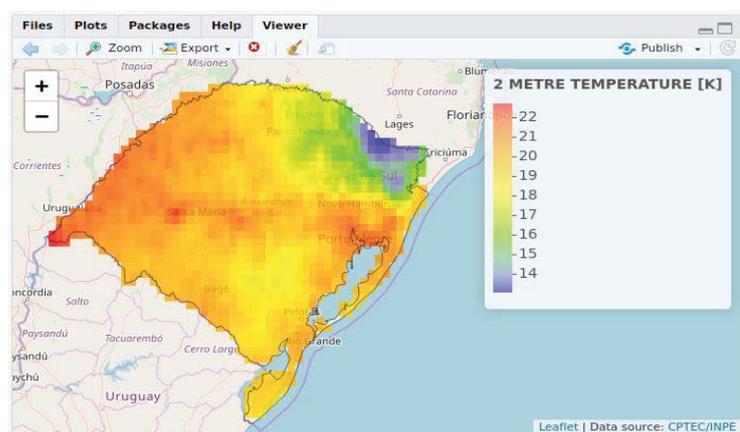


Figura 7: Mapa resultado da função `plot_map()`.

O Mapa mostra a variação de valores da variável da previsão do tempo através da variação de cores no mapa.

5 Conclusões e trabalhos futuros

O pacote `EtaModelW`, em sua primeira versão, demonstrou características suficientes para ser tratado como um pacote da linguagem R. Suas funções têm o potencial de prestar informações sobre o modelo de simulação usado na previsão do tempo, de transferir arquivos de dados em formato binário combinada com a conversão de arquivo binário em `dataframe`, o carregamento deste para a memória de trabalho e de gerar gráficos e mapas possibilitando assim, praticidade para o usuário utilizar e visualizar os dados da previsão do tempo.

Ao se comparar o pacote `EtaModelW` com os demais pacotes relacionados para este trabalho, pode-se observar como vantagem o fato deste ser disponibilizado gratuitamente, quando alguns pacotes acessam fontes em que há um serviço pago para o acesso a dados da previsão do tempo, com os pacotes `rwunderground`, `weatherData` e `SkyWise`. Quando comparado a pacotes como `rNOMADS` e `rnoaa`, o fato do pacote `EtaModelW` estar em sua versão inicial provê ao usuário a facilidade de uso do pacote como uma característica natural, tendo em vista a grande quantidade de fontes de dados, modelos de simulação, funções, formas de transferência de dados, tipos de dados, entre outras características que cada um destes pacotes fornecem ao usuário. Apesar de não ter tido acesso às funcionalidades de alguns dos pacotes relacionados para o trabalho, por estes possuírem fontes de dados como serviços pagos, pode-se observar que do ponto de vista da visualização de dados, o pacote `EtaModelW` possui ferramentas adicionais em relação à maioria pacotes, como é o caso dos pacotes `riem` e `rdwd` que não fornecem funções de visualização de dados. Outra vantagem do pacote `EtaModelW` é que assim como os pacotes mais completos ou em constante evolução, ele também possibilita ao usuário trabalhar com uma grande quantidade de variáveis da previsão do tempo.

Deste modo é possível concluir que mesmo em sua versão inicial o pacote `EtaModelW` disponibiliza funções importantes do ponto de vista dos itens básicos que um pacote necessita para trabalhar com dados da previsão do tempo, sendo assim, o pacote cumpre seu objetivo inicial de facilitar o acesso de desenvolvedores e usuários da linguagem R aos dados da previsão do tempo fornecidos pelo CPTEC/INPE.

Para trabalhos futuros o pacote `EtaModelW` deve ser ampliado com novas funções gráficas procurando explorar esta característica que torna a linguagem R especial. Possibilitá-lo a trabalhar com outras versões do modelo Eta que possuem resoluções diferentes da apresentada neste trabalho se visualiza como um meio de estendê-lo. Uma outra forma de acesso aos dados que possa diminuir sua sobrecarga de trabalho ao transferir os dados do CPTEC/INPE para o diretório de trabalho do pacote seria importante, por exemplo, por meio de uma Interface de Programação de Aplicações (API). Por fim, outros recursos da linguagem de programação R devem ser testados visando incrementar e expandir este pacote.

Referências Bibliográficas

- Boessenkool, B. rdwd: Select and Download Climate Data from 'DWD' (German Weather Service). 2019. Disponível em <<https://cran.r-project.org/package=rdwd>> Acesso: maio/2019.
- Bowman, D. C.; Lees, J. M. Near real time weather and ocean model data access with rNOMADS. Computers and Geosciences vol. 78, pg. 88-95, 2015.
- Chamberlain, S; et al. rnoaa: 'NOAA' Weather Data from R. 2019. Disponível em <<https://CRAN.R-project.org/package=rnoaa>> Acesso: maio/2019.
- Chou, S. C.; Marengo, J. A.; Lyra, A.; et al. Downscaling of South America Present Climate Driven by 4-Member HadCM3 Runs. Climate Dynamics, vol. 38, p. 635-653, 2012.
- Hornik, K. The Comprehensive R Archive Network. WIREs Computational Statistics, vol. 4, no. 4, p. 394-398, 2012.
- Kemp, M. U.; Loon, E. E. van; Shamoun-Baranes, J.; Bouten, W. RNCEP: Obtain, Organize, and Visualize NCEP Weather Data. 2017. Disponível em <<https://cran.r-project.org/web/packages/RNCEP/RNCEP.pdf>> Acesso: junho/2019.
- LaZerte, S.; Albers, S. weathercan: Download and format weather data from Environment and Climate Change Canada. The Journal of Open Source Software, no. 571, 2018.
- Mesinger, F. The Eta Model: Design, History, Performance, What Lessons have we Learned? Maryland, 2004.
- Narasimhan, R. weatherData: Get Weather Data from the Web. 2014. Disponível em <<https://ram-n.github.io/weatherData>> Acesso: maio/2019.
- R Development Core Team. Writing R Extensions. 2019. Disponível em: <<https://cran.r-project.org/doc/manuals/r-release/R-exts.pdf>> Acesso: maio/2019.
- Rudis, B. darksky: Tools to Work with the Dark Sky API. 2019. Disponível em <<https://CRAN.R-project.org/package=darksky>> Acesso: maio/2019.
- Salmon, M.; Anderson, B. riem: Accesses Weather Data from the Iowa Environment Mesonet. 2016. Disponível em <<https://cran.r-project.org/package=riem>> Acesso: maio/2019.
- Shum, A. rwunderground: R Interface to Weather Underground API. 2018. Disponível em <<https://CRAN.R-project.org/package=rwunderground>> Acesso: maio/2019.
- Thompson, Daphne. High-Resolution Weather Data with R. 2017. Disponível em <<https://blog.weatherops.com/high-resolution-weather-data-with-r>> Acesso: junho/2019.
- Venables, W. N.; Smith, D. M.; R Core Team. An Introduction to R. Notes on R: A Programming Environment for Data Analysis and Graphics. 2019. Disponível em <<https://cran.r-project.org/doc/manuals/r-release/R-intro.pdf>> Acesso: junho/2019.
- Wickham, H. R Packages: Organize, test, document and share your code. O'Reilly Media, 2015. Disponível em <<http://r-pkgs.had.co.nz/>>. Acesso: maio/2019.