

MosaicoUtils: um pacote R com funções utilitárias para tratamento de dados climáticos e ambientais

Geovani Sordi

Curso de Ciência da Computação, UPF
Campus 1 - BR 285 - Passo Fundo (RS) - Brasil

141974@upf.br

Abstract. *MosaicoUtils is an R package containing useful functions for manipulating climatic data, which aims to facilitate the retrieval and storage of data obtained through the EtaModelCC and EtaModelW packages. MosaicoUtils has functions that extract, manipulate, and write data in some file formats most commonly used by applications that manipulate weather data. MosaicoUtils functions provide an easy-to-use way to extract, manipulate and write the data made available, facilitating manipulation on other systems, stimulating the user's reuse of code and thus making it cleaner and more organized for reading.*

Keywords: *Climate Data, MosaicoUtils, R package, Utility functions.*

Resumo. *O MosaicoUtils é um pacote R contendo funções utilitárias para a manipulação de dados climáticos, que objetiva facilitar a obtenção e armazenamento de dados obtidos por meio dos pacotes EtaModelCC e EtaModelW. O MosaicoUtils possui funções que extraem, manipulam e gravam dados em alguns formatos de arquivos mais utilizados pelas aplicações que manipulam dados meteorológicos. As funções do MosaicoUtils disponibilizam uma maneira de fácil utilização na extração, manipulação e gravação dos dados disponibilizados, facilitando a manipulação em outros sistemas, estimulando a reutilização de código por parte do usuário e tornando-o assim mais limpo e organizado para a leitura.*

Palavras chaves: *Dados climáticos, funções utilitárias, MosaicoUtils, pacote em R.*

1. Introdução

Os projetos desenvolvidos em qualquer linguagem de programação, quando aumentam a sua complexidade, tendem a aumentar o número de funções e de arquivos que são necessários para manter o sistema organizado. Desta forma, o uso de boas práticas de programação (Hornick, 2018) facilita o desenvolvimento dos mais diversos tipos de sistemas computacionais. Um código bem encapsulado, com o uso adequado de funções, reduz a necessidade de objetos intermediários pois, para gerar um deles, basta a aplicação de uma função. Além disso, programas modulares normalmente são mais enxutos e limpos do que sem a utilização destes recursos pois, eles estimulam a reutilização de código. No caso da linguagem R, objeto deste trabalho, é interessante a criação de um pacote que disponibiliza funções utilitárias para uso no ambiente de tratamento de dados climáticos e ambientais do grupo de pesquisa Mosaico da UPF.

Com a utilização de pacotes na linguagem R, a disponibilização de funções e reutilização de código se torna mais fácil, pois é necessário apenas realizar o download e instalação do pacote a ser utilizado e sua utilização já pode ser executada, bastando apenas realizar uma chamada à função e passando os parâmetros necessários.

R é uma linguagem de programação e um ambiente para computação estatística e geração de gráficos. É um software livre sob os termos da GNU, compatível com Windows, MacOS e plataformas UNIX. A R é formada por um conjunto integrado de softwares que realizam cálculos, manutenção de dados e exibições gráficas (The R Foundation, 2019).

Uma grande quantidade de pacotes R são disponibilizados pelo Comprehensive R Archive Network (CRAN). Estes pacotes estendem as funcionalidades da linguagem R. Há algumas definições importantes que precisam ser conhecidas para contextualização de pacotes R, são elas (Leisch, 2009):

- Pacote (Package): Extensão de R, com, código, dados e documentação em um formato padrão;
- Biblioteca (Library): Um diretório contendo pacotes instalados;
- Repositório (Repository): Site onde se encontram os pacotes;
- Código fonte (Source): A versão original do pacote com texto e código;
- Binário (Binary): Versão compilada do pacote.

Pacotes em R são conjuntos de código, dados e documentação, em um formato padrão, que podem ser utilizados de acordo com a necessidade, ou ainda, desenvolvidos e compartilhados (Wickham and Bryan, 2015).

O pacote MosaicoUtils tem como objetivo a disponibilização de funções utilitárias para contribuir no tratamento de dados climáticos e ambientais do grupo de pesquisa Mosaico da UPF, auxiliando na produtividade e desenvolvimento das tarefas solicitadas, estimulando a reutilização de código.

Na seção 2 serão apresentados alguns pacotes em R e plataformas que possuem finalidades similares ao MosaicoUtils. Na seção 3 uma descrição do pacote é apresentada, dando uma base das características, funcionalidades e resultados obtidos. Na seção 4 são apresentadas as conclusões finais deste pacote bem como possibilidades de trabalhos futuros.

2. Trabalhos relacionados

Visando a construção de um pacote utilitário em R com base em dados climáticos, foram estudados e utilizados alguns pacotes em R e plataformas que disponibilizam ou utilizam dados climáticos. Foram observados o modo como os dados são disponibilizados pelas plataformas e obtidos pelos pacotes, os tipos de funções desenvolvidas e seus objetivos.

a) EtaModelCC: O pacote EtaModelCC tem como objetivo de facilitar o acesso e visualização de dados de mudanças climáticas da América do Sul, geradas pelo CPTEC/INPE e disponibilizados por meio da plataforma PROJETA. O pacote conta com funções para acesso a dados de mudanças climáticas, assim como funções de visualização que retornam gráficos e mapas ao usuário. A plataforma PROJETA disponibiliza os dados em formato json e são convertidos para o formato DataFrame do R no próprio pacote (Dezordi et. al, 2018).

b) EtaModelW: O pacote EtaModelW tem como objetivo de facilitar o acesso aos dados da previsão do tempo gerados pelo CPTEC/INPE por meio do modelo Eta com resolução espacial de 15km. O pacote conta com funções para acesso aos dados de previsão do tempo bem como para a geração de gráficos e mapas com base nos dados obtidos (Silva et al., 2019).

c) Sisalert: O Sisalert (Sistema de Previsão de Risco de Epidemias de Doenças de Plantas) é uma plataforma que coleta dados meteorológicos, processa as informações para simulação de riscos de epidemias e distribui o alerta aos usuários. Através do sistema, informações como temperatura, umidade e duração do período de molhamento foliar são coletadas por estações meteorológicas da rede do INMET (Instituto Nacional de Meteorologia) e processadas em modelos computacionais que alertam para as situações de risco (Fernandes, Pavan e Sanhueza, 2011).

d) PROJETA: A plataforma PROJETA visa automatizar o processo de extração e disponibilização dos dados das projeções climáticas regionalizadas para a América do Sul, por meio do modelo regional Eta, geradas pelo CPTEC/INPE. Este conjunto de dados está disponível para diferentes usuários via web ou API, de forma flexível quanto ao formato e ao volume de dados. Para tanto, a plataforma dispõe de uma base de dados com novas projeções que utilizam diferentes modelos climáticos globais, integrando tecnologias que possibilitam o acesso ao banco de dados de forma ágil e fácil, para posterior utilização em aplicações que necessitam destes conjuntos de dados climáticos (Hölbj et al. 2018).

e) The Climate Explorer: O Climate Explorer é uma aplicação web que oferece mapas e gráficos interativos contendo mapas de temperatura observada e projetada, precipitação e variáveis climáticas relacionadas para cada condado nos Estados Unidos, o mesmo ajuda as pessoas a explorar as condições climáticas projetadas que podem colocar pessoas, propriedades e outros ativos em risco. A ferramenta exibe informações históricas de temperatura, precipitação e variáveis relacionadas de 1950 a 2013 para os Estados Unidos. Os dados são disponibilizados no formato png e csv (Sanderson and Wehner, 2017).

f) ERA5 (Copernicus): O ERA5, serviço da plataforma Copernicus, disponibiliza dados sobre os níveis de pressão, realizando uma reanálise atmosférica de quinta geração do clima global. A reanálise combina dados de modelo com observações de todo o mundo em um conjunto de dados globalmente completo e consistente. É baseado no método usado pelos centros de previsão numérica do tempo, onde a cada 12 horas uma previsão anterior é combinada com as novas observações disponíveis de uma maneira ótima para produzir uma nova melhor estimativa do estado da atmosfera, a partir do qual uma previsão atualizada e melhorada é emitida. Atualmente ele possui cobertura do ano de 1979 até o presente, com uma resolução temporária por hora. Os dados são disponibilizados nos formatos GRIB e NetCDF (Copernicus, 2020).

g) HarvestChoice: O objetivo principal do projeto HarvestChoice era liderar o desenvolvimento de dados padronizados e especialmente relevantes para a agricultura, combinando a partir de múltiplas fontes de dados, dando sentido a esses dados e informando as decisões estratégicas relativas à mudança de técnica na agricultura. O projeto HarvestChoice desenvolveu um conjunto de ferramentas para permitir que os usuários explorem e visualizem o conjunto de dados principais e aproveitem esses dados

em apoio à tomada de decisão na agricultura da África Subsariana por meio de uma agricultura mais produtiva e rentável (IFPRI, 2020).

Além destas ferramentas, autores de alguns pacotes R agrupam uma coleção de funções pequenas e úteis em algum tipo de arquivo e geralmente não exportam as funções, uma vez que são geralmente projetadas para trabalhar em componentes internos do pacote em vez de expor sua funcionalidade por meio da API do pacote exportado (Rudis, 2018). Exemplos de pacotes R com este objetivo de oferecer funções utilitários são:

h) microbiomeutilities: é um pacote utilitário voltado para a área de microbioma, o pacote possui funções que geram relatórios HTML com informações sobre análises preliminares de controle de qualidade, Diversidade Alfa, ordenação e composição de tabelas OTU (Shetty and Lahti, 2020);

i) bigutilsr: possui funções utilitárias para dados em grande escala. Por enquanto, o pacote inclui principalmente funções para detecção de outlier e projeção de PCA (Privé, 2020);

j) R.utils: o pacote R.utils possui funções utilitárias para programar e desenvolver pacotes em R (Bengtsson, 2020);

k) utils: o pacote utils é um pacote do próprio R onde contém uma coleção de funções utilitárias.

Os pacotes EtaModelCC e EtaModelW têm grande importância para o desenvolvimento do MosaicoUtils, devido ao MosaicoUtils ser complementar aos mesmos, sendo que eles são responsáveis pelo recebimento e disponibilização dos dados já manipulados por meio de funções internas, as quais disponibilizam os dados prontos para utilização apenas realizando uma chamada na função e passando os parâmetros dos dados que precisamos obter. As plataformas estudadas tiveram um papel essencial, pois são elas que disponibilizaram toda a base de dados que deram o prosseguimento no desenvolvimento do pacote. Entender como os dados são disponibilizados e os formatos que os dados possuem, traz maior facilidade para a manipulação e extração das informações que precisamos.

3. O pacote MosaicoUtils

O pacote MosaicoUtils foi desenvolvido visando a disponibilização de funções utilitárias como um complemento ao manuseio de dados gerados ou obtidos por meio dos pacotes EtaModelCC e EtaModelW. Foram criadas funções que permitem ao usuário manipular de maneira mais simples os dados obtidos por estes pacotes. O pacote MosaicoUtils permite a desenvolvedores e usuários da linguagem R, que utilizem suas funções para acessar e obter os dados, sem a necessidade de novas implementações.

Para atingir esses objetivos, o pacote inclui quatro funções:

- a) `MU_read_ctl()` : retorna os dados do arquivo ctl informado;
- b) `MU_convert()` : realiza a gravação de um arquivo em formato passado como parâmetro;
- c) `script_dir()` : retorna o diretório de trabalho utilizado no script.
- d) `MU_shapefile()` : realiza a gravação de um arquivo em formato shapefile.

Atualmente, o MosaicoUtils está disponível na plataforma de código fonte GitHub (Fig. 1), podendo ser acessado no seguinte endereço: <https://github.com/holbig/MosaicoUtils>. Para realizar a instalação a partir do R é necessário utilizar o pacote `devtools` e a função `install_github()` no seguinte formato `devtools::install_github("holbig/MosaicoUtils")`. Apesar do pacote MosaicoUtils não estar atualmente disponível no CRAN (Comprehensive R Archive Network) ele segue as políticas de repositório do CRAN, facilitando uma possível futura inclusão.

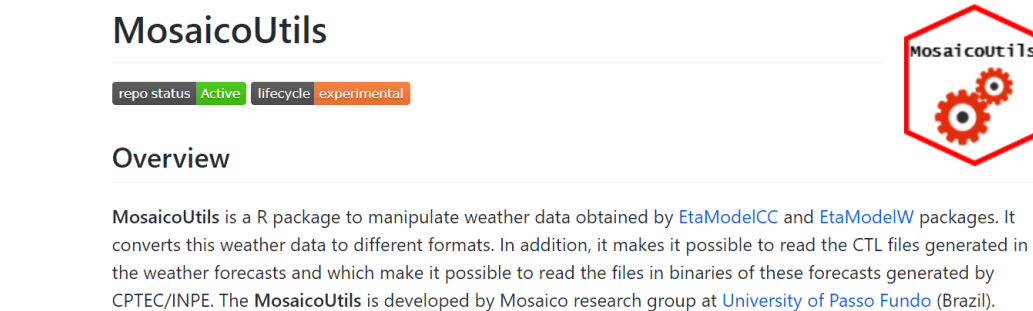


Figura 1: Página principal do pacote MosaicoUtils no Github

3.1. Obtendo dados de previsão do tempo

As funções a seguir são responsáveis por disponibilizar os dados, conforme as requisições do usuário:

a) `MU_read_ctl()`: essa função é responsável por obter, manipular e retornar os dados de um arquivo em formato `.ctl` passado como parâmetro na função. O retorno da função é uma lista, onde a partir dos dados obtidos, estes serão utilizados em conjunto com funções já existentes internamente no pacote `EtaModelW`. Esta função se faz necessária para a leitura dos arquivos de previsão de tempo gerados pelo CPTEC-INPE (modelos Eta e WRF) que estão em formato binário, onde as suas estruturas estão descritas no arquivo `.ctl` correspondente, conforme exemplo apresentado na Fig. 2, onde as informações constantes neste arquivo permitem que o arquivo binário seja lido de forma correta. A função `MU_read_ctl("ctl file")`, recebe o endereço completo da localização do arquivo `.ctl` (local ou na web), lê o arquivo e organiza as informações das variáveis de previsão, necessárias para se ler corretamente o arquivo binário, em uma variável tipo lista do R (Fig. 3). Com esta variável criada pela função consegue-se ler de forma correta o arquivo binário.

```
dset ^tp2m_Eta15km2020061012.bin
title File
undef 1e+20
xdef 375 linear -86.000000 0.150000
ydef 500 linear -58.000000 0.150000
zdef 1 levels 1013
tdef 265 linear 12Z10Jun2020 1hr
vars 1
tp2m 0 99 2 METRE TEMPERATURE [K]
```

```
endvars
```

Figura 2: Exemplo de dados entrada para a função `MU_read_ctl()`.

```
> link <- "http://ftp1.cptec.inpe.br/pesquisa/grpeta/tempo/Products/Eta_tempo/
Eta15km/bin_var/2020061012/tp2m_Eta15km2020061012.ct1"
> dados <- MU_read_ctl(link)
> dados
$file_bin
[1] "tp2m_Eta15km2020061012.bin"

$xddef
[1] 375

$ydef
[1] 500

$zdef
[1] 1

$stdef
[1] 265

$svars
[1] 1

$var_nick
[1] "tp2m"

$var_desc
[1] "2 METRE TEMPERATURE"

$var_unit
[1] "K"

$lat
[1] -58

$lon
[1] -86

$resolucao
[1] 0.15

$levels
[1] 1013
```

Figura 3: Chamada e retorno da função `MU_read_ctl()`.

b) `script_dir()`: essa função retorna o diretório de trabalho que está sendo utilizado no script e fixa o mesmo para que possa ser realizado qualquer gravação de arquivos no mesmo diretório. A função está adaptada para ser utilizada em trabalhos utilizando-se o RStudio, bem como pela utilização por linha de comando, conforme código apresentado na Fig. 4.

```
script_dir <- function(){
  this_file = gsub("--file=", "", commandArgs()[grepl("--file",
commandArgs())])
```

```

if (length(this_file) > 0){
  wd <- paste(head(strsplit(this_file, '[/|\\'])[[1]], -1), collapse
= .Platform$file.sep)
}else{
  setwd(dirname(rstudioapi::getActiveDocumentContext()$path))
}
}

```

Figura 4: Código da função script_dir().

3.2. Convertendo dados dos modelos climáticos em diferentes formatos

A função a seguir é responsável por realizar a gravação de dados diretamente no diretório onde o script está sendo executado.

a) MU_convert(): essa função é responsável por realizar a gravação de um arquivo com os dados passados por parâmetro para a função, conforme exemplo apresentado na Fig. 5. Os formatos que a função possui suporte para a gravação do arquivo são: csv, json, xml, rda, tiff, grads e NetCDF (Tabela 1). Na Fig. 6, com o apoio de uma ferramenta de leitura de arquivos no formato json, está sendo mostrado como os dados no formato json são gravados. Na Fig. 7, é apresentado diretamente no R Studio, utilizando a função de plotagem como o arquivo no formato tiff é visualizado após a gravação.

```

> head(clima$Data)
  Latitude Longitude Year    Value
1   -26.6     -57.8 2018 19.62531
2   -26.6     -57.6 2018 19.63815
3   -26.6     -57.4 2018 19.64053
4   -26.6     -57.2 2018 19.63523
5   -26.6     -57.0 2018 19.67530
6   -26.6     -56.8 2018 19.63874

```

Figura 5: Exemplo de dados de entrada da função MU_convert().

Tabela 1: Formatos suportados na função MU_convert()

Tipo do arquivo	Descrição
csv	Formato de arquivo simples usado para armazenar dados tabulares, como uma planilha ou banco de dados.
json	Formato de arquivo padrão aberto e de intercâmbio de dados, usado para armazenar e transmitir objetos de dados .
xml	Arquivo usado para criar documentos com dados organizados.

rda	Arquivo específico do R que pode armazenar quantos objetos quiser em um único arquivo, em formato binário.
tiff	Arquivo raster para imagens digitais.
grads	Arquivo que trabalha com matrizes de dados nas quais as variáveis podem possuir até 4 dimensões.
NetCDF	Arquivo para armazenar dados científicos multidimensionais.

```
[
  {
    "Latitude": -26.6,
    "Longitude": -57.8,
    "Year": 2018,
    "Value": 19.6253
  },
  {
    "Latitude": -26.6,
    "Longitude": -57.6,
    "Year": 2018,
    "Value": 19.6381
  },
  {
    "Latitude": -26.6,
    "Longitude": -57.4,
    "Year": 2018,
    "Value": 19.6405
  },
  {
    "Latitude": -26.6,
    "Longitude": -57.2,
    "Year": 2018,
    "Value": 19.6352
  },
]
```

Figura 6: Exemplo de saída da função `MU_convert()` no formato json.

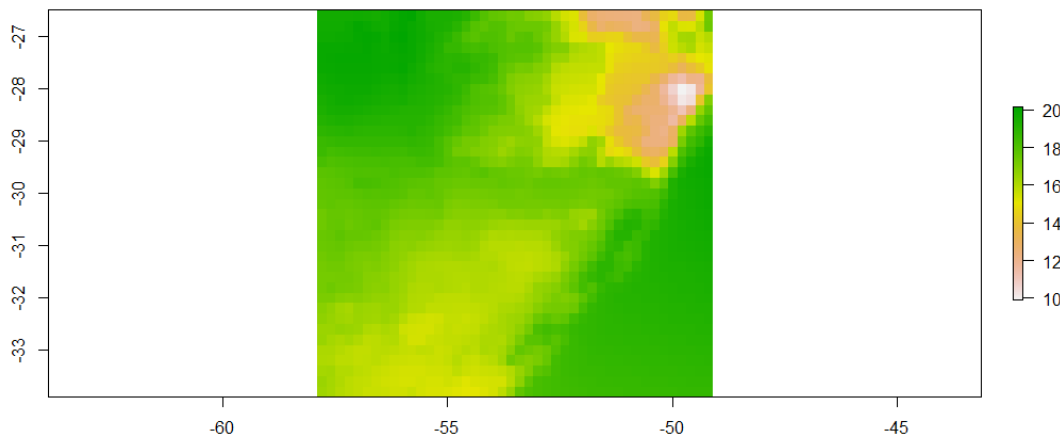


Figura 7: Exemplo de saída da função `MU_convert()` no formato tiff.

b) `MU_shapefile()`: essa função é responsável por realizar a gravação de um arquivo shapefile com os dados passados por parâmetro para a função. O parâmetro de entrada da função é um dataframe que contém uma coluna de latitude e uma coluna de longitude, podendo ou não conter outros dados, após o recebimento deste dataframe a função realiza uma manipulação interna de todos os pontos e transforma em um polígono, logo após realiza a criação e gravação do shapefile contendo somente o contorno deste polígono. Esse arquivo no formato shapefile pode ser visualizado com a utilização do pacote `leaflet`, conforme exemplo apresentado na Fig. 8.

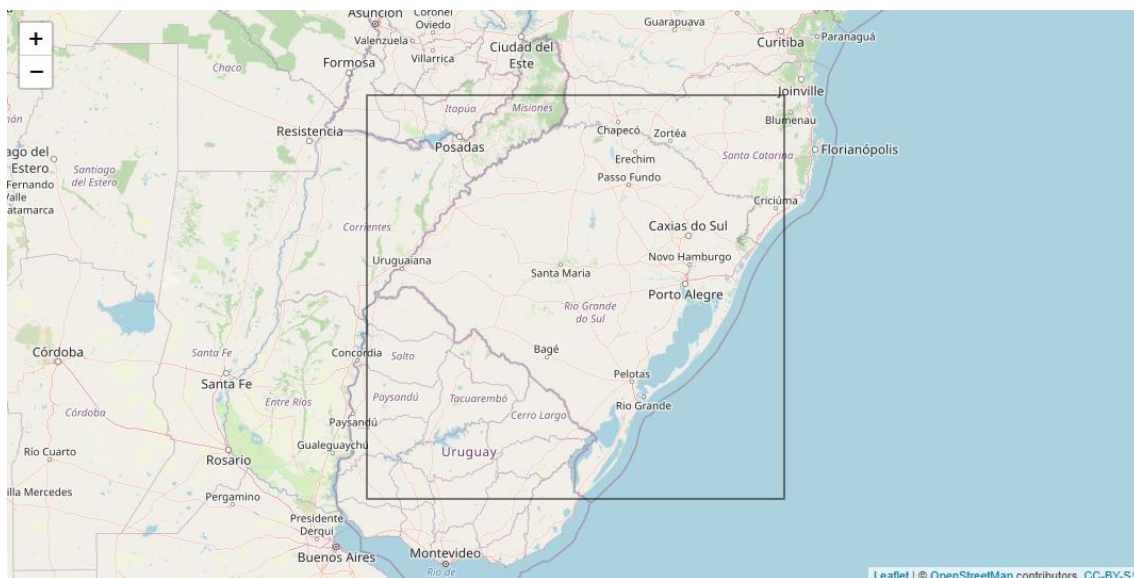


Figura 8: Exemplo de saída da função `MU_shapefile()`.

Por fim, para a criação destas funções, foram utilizados pacotes externos para auxiliar no desenvolvimento, o pacote `stringr` foi utilizado na manipulação de strings, os pacotes `grDevices`, `jsonlite`, `kulife`, `raster`, `rgdal`, `rvest` e `sp` que auxiliaram na gravação de arquivos, contando também com os pacotes `devtools` e `roxygen2` que

auxiliaram na criação e documentação do pacote como um todo, além daqueles incluídos por padrão na distribuição da linguagem R, `utils` e `base`.

4. Conclusões e trabalhos futuros

O pacote MosaicoUtils desenvolvido neste trabalho oferece uma forma simplificada de acesso a funções utilitárias com necessidades gerais para utilização no grupo de pesquisa Mosaico da UPF. Ele complementa os pacotes EtaModelCC e EtaModelW, onde juntos podem extrair dados e informações significativas de forma simples e rápida.

A partir da instalação do pacote, o usuário pode fazer uso das funções para obter informações, converter e gravar dados em diversos formatos, apenas realizando uma chamada à função e passando os parâmetros necessários, sem a necessidade de demandar tempo no desenvolvimento de funções para a extração de dados ou utilização de dados estáticos, além de estimular a reutilização de código por parte do usuário.

O desenvolvimento do pacote MosaicoUtils continuará em evolução, aperfeiçoando as funções já existentes e criando novas funções para a visualização de dados climáticos com base na demanda dos usuários da plataforma Projeta.

5. Referências

Claus Ekstrom, Ib M. Skovgaard and Torben Martinussen (2013). kulife: Datasets and functions from the (now non-existing) Faculty of Life Sciences, University of Copenhagen. R package version 0.1-14. <https://CRAN.R-project.org/package=kulife>

Copernicus (2020). Climate Data Store. Disponível em <https://cds.climate.copernicus.eu/>. Acesso em: 22 de Out. de 2020.

Dezordi, Marina Lopes; Hölbis, Carlos Amaral; Pavan, Willingthon; Fernandes, José Maurício Cunha. An R package to access climate change data for South America regionalized by the Eta Model of CPTEC/INPE. In: WORKSHOP DE COMPUTAÇÃO APLICADA À GESTÃO DO MEIO AMBIENTE E RECURSOS NATURAIS (WCAMA_CSBC), 9., 2018. 9º Workshop de Computação Aplicada a Gestão do Meio Ambiente e Recursos Naturais (WCAMA_CSBC 2018). Porto Alegre: Sociedade Brasileira de Computação, june 2018. ISSN 2595-6124.

Bob Rudis (2018) Dissecting R Package “Utility Belts”. Disponível em: <<https://rud.is/b/2018/04/08/dissecting-r-package-utility-belts/>>. Acesso em 15 de out. de 2020.

Florian Privé (2020). bigutilsr: Utility Functions for Large-scale Data. R package version 0.3.3. Disponível em <https://CRAN.R-project.org/package=bigutilsr>

Hadley Wickham (2019). stringr: Simple, Consistent Wrappers for Common String Operations. R package version 1.4.0. <https://CRAN.R-project.org/package=stringr>

Hadley Wickham (2020). rvest: Easily Harvest (Scrape) Web Pages. R package version 0.3.6. <https://CRAN.R-project.org/package=rvest>

Hadley Wickham, Jim Hester and Winston Chang (2020). devtools: Tools to Make Developing R Packages Easier. R package version 2.3.1. <https://CRAN.R-project.org/package=devtools>

Hadley Wickham, Jennifer Bryan, 2015. R Packages: Organize, Test, Document, and Share Your Code, First edition. O'Reilly Media, Inc., Sebastopol, EUA. <https://r-pkgs.org>

Hadley Wickham, Peter Danenberg, Gábor Csárdi and Manuel Eugster (2020). roxygen2: In-Line Documentation for R. R package version 7.1.1. <https://CRAN.R-project.org/package=roxygen2>

Henrik Bengtsson (2020). R.utils: Various Programming Utilities. R package version 2.10.1. <https://CRAN.R-project.org/package=R.utils>

Hölbig, C. A.; Mazzonetto, A.; Borella, F.; Pavan, W.; Fernandes, J. M. C.; Chagas, D. J.; Gomes, J. L.; Chou, S. C. PROJETA platform: accessing high resolution climate change projections over Central and South America using the Eta model. Agrometeoros, Passo Fundo, v.26, n.1, p.71-81, 2018.

International Food Policy Research Institute (2020). Project HARVESTCHOICE. Disponível em: <https://www.ifpri.org/project/harvestchoice>, Acesso em: 22 de Out. de 2020.

Jeroen Ooms (2014). The jsonlite Package: A Practical and Consistent Mapping Between JSON Data and R Objects. arXiv:1403.2805 [stat.CO] URL <https://arxiv.org/abs/1403.2805>.

Joe Cheng, Bhaskar Karambelkar and Yihui Xie (2019). leaflet: Create Interactive Web Maps with the JavaScript 'Leaflet' Library. R package version 2.0.3. <https://CRAN.R-project.org/package=leaflet>

Kevin Ushey, JJ Allaire, Hadley Wickham and Gary Ritchie (2020). rstudioapi: Safely Access the RStudio API. R package version 0.11. <https://CRAN.R-project.org/package=rstudioapi>

Sudarshan A. Shetty, & Leo Lahti. (2020). microbiomeutilities: Utilities for Microbiome Analytics (Version 1.00.10). Disponível em <http://microbiome.github.com/microbiome>

Pebesma, E.J., R.S. Bivand, 2005. Classes and methods for spatial data in R. R News 5 (2), <https://cran.r-project.org/doc/Rnews/>.

Roger Bivand, Tim Keitt and Barry Rowlingson (2020). rgdal: Bindings for the 'Geospatial' Data Abstraction Library. R package version 1.5-16. <https://CRAN.R-project.org/package=rgdal>

R Core Team (2019). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL <https://www.R-project.org/>.

Robert J. Hijmans (2020). raster: Geographic Data Analysis and Modeling. R package version 3.3-13. <https://CRAN.R-project.org/package=raster>

Sanderson, B.M. and M.F. Wehner (2017). Model weighting strategy. In: Climate Science Special Report: Fourth National Climate Assessment, Volume I [Wuebbles, D.J., D.W. Fahey, K.A. Hibbard, D.J. Dokken, B.C. Stewart, and T.K. Maycock (eds.)]. U.S. Global Change Research Program, Washington, DC, USA, pp. 436-442. Doi: [10.7930/J06T0JS3](https://doi.org/10.7930/J06T0JS3).

Silva, R. R. P. ; Dezordi, M. L. ; Pavan, W. ; Hölbig, C. A. (2019) EtaModelW: um pacote R para disponibilização de dados de previsão do tempo gerados pelo CPTEC/INPE. In:

VII Seminário Argentina-Brasil de Tecnologias da Informação e da Comunicação, 2019, Rio Grande. Anais do VII Seminário Argentina-Brasil de Tecnologias da Informação e da Comunicação. Rio Grande: FURG, 2019. v. 1. p. 1-15.