Package: RedeAgroRadar (via r-universe)

October 15, 2026

Type Package

Title Weather Radar Monitoring and 'Telegram' Alerts for Agriculture Research

Version 0.1.1

Description Provides tools to download, process, and analyze real-time meteorological radar images from Simepar (Paraná, Brazil) <https://www.simepar.br/simepar/radar_msc>. Designed to support the 'Rede Agropesquisa' hydrological monitoring, it includes functions to detect rainfall intensity based on Red, Green, and Blue (RGB) color values within predefined circular study areas. Features automated integration with the 'Telegram Bot API' <https://core.telegram.org/bots/api> to send spatialized image alerts and an interactive 'shiny' dashboard for easy configuration and continuous weather tracking.

License GPL-3

Encoding UTF-8

Language pt-BR

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Imports httr, magick, shiny, bslib

URL https://github.com/santoshbdias/RedeAgroRadar

BugReports https://github.com/santoshbdias/RedeAgroRadar/issues

Config/pak/sysreqs cmake make libmagick++-dev gsfonts libuv1-dev libssl-dev zlib1g-dev

Repository https://santoshbdias.r-universe.dev

Date/Publication 2026-03-13 02:26:01 UTC

RemoteUrl https://github.com/santoshbdias/redeagroradar

RemoteRef HEAD

RemoteSha 0f702b494a999314147e4a29bd4766454a9424e0

Contents

analisar_radar_PR 2
baixar_radar_PR 3
executar_alerta_telegram 4
gerar_imagem_radar 5
Run_Monitor_RedeAgro 7
status_diario 8

Index **10**

`analisar_radar_PR` *Analisar a Cor Média em uma Região Circular do Radar*

Description

Esta função baixa a imagem atual do radar meteorológico do Paraná e extrai os valores médios dos canais RGB (Red, Green, Blue) em uma área circular ao redor de uma coordenada específica (cidade ou ponto de interesse).

Usage

```
analisar_radar_PR(
  mega = "PresidenteCasteloBranco",
  raio = 50,
  coords_custom = NULL
)
```

Arguments

`mega`	Caractere. Nome da cidade ou ponto de interesse. Padrão é `'PresidenteCasteloBranco'`.
`raio`	Numérico. Raio em pixels da área circular a ser analisada ao redor do ponto central. Padrão é 50.
`coords_custom`	Lista nomeada opcional. Permite definir coordenadas customizadas com valores de `x` e `y` (ex: `list("Cascavel" = list(x = 200, y = 300))`).

Details

A função depende de `baixar_radar_PR()` para obter a imagem de satélite do 'Simepar'. A varredura dos pixels é feita utilizando coordenadas polares para cobrir a área definida pelo raio estipulado.

Value

Retorna uma lista nomeada contendo os valores numéricos médios de `R`, `G` e `B`, além do número total de pixels validados (`n_pixels`). Caso a imagem não possa ser baixada, retorna `FALSE`.

Examples

```
# Analisar a chuva na regiao de Cianorte com raio padrao (50px)
resultado <- analisar_radar_PR(mega = "Cianorte")

# Verifica se o resultado e uma lista (garante que baixou a imagem sem erro de internet)
if (is.list(resultado)) {
  print(resultado$R) # Valor do canal Vermelho (chuva forte)
}

# Analisar com um raio maior (salvando em variavel para evitar impressoes soltas)
res_ponta_grossa <- analisar_radar_PR(mega = "PontaGrossa", raio = 80)

# Inserir uma cidade que nao esta na lista padrao
novas_coords <- list("Cascavel" = list(x = 250, y = 310))
res_cascavel <- analisar_radar_PR(mega = "Cascavel", raio = 50,
                                  coords_custom = novas_coords)
```

---

baixar_radar_PR *Baixar Imagem do Radar Meteorológico do Paraná ('Simepar')*

Description

Faz o download da imagem mais recente do radar meteorológico do Paraná, disponibilizada publicamente peloo 'Simepar' (https://www.simepar.br/simepar/radar_msc), e a carrega diretamente na memória do 'R' como um objeto de imagem.

Usage

```
baixar_radar_PR(
  url = "https://lb01.simepar.br/riak/pgw-radar/product1.jpeg",
  timeout = 10
)
```

Arguments

url	Caractere. String contendo a URL direta da imagem. O padrão aponta para o produto atual de radar do 'Simepar'.
timeout	Numérico. Tempo máximo de espera em segundos pela resposta do servidor antes de cancelar a operação. Padrão é 10.

Details

A função realiza uma requisição HTTP GET utilizando cabeçalhos (headers) customizados para evitar bloqueios do servidor. O carregamento é feito diretamente na memória (em formato raw), sem a necessidade de gravar arquivos temporários no disco, o que otimiza a performance. Inclui também uma trava de tempo (timeout) para evitar travamentos infinitos caso o servidor do radar esteja fora do ar, uma prática exigida pelas políticas do 'CRAN'.

Value

Retorna um objeto de imagem da classe `magick-image`. Em caso de falha na conexão, esgotamento do tempo limite (timeout) ou erro de formatação, retorna `NULL` e exibe uma mensagem de alerta.

Examples

```
# Baixar a imagem atual do radar com timeout padrao (10s)
radar_img <- baixar_radar_PR()

# Exibir a imagem no painel de plots, se o download foi bem-sucedido
if (!is.null(radar_img)) {
  plot(radar_img)
}

# Tentar baixar com um timeout maior caso a internet esteja lenta
radar_img_lento <- baixar_radar_PR(timeout = 20)
```

`executar_alerta_telegram` *Avalia Radar e Dispara Alerta Fotográfico via Telegram*

Description

Avalia os valores médios de cor (RGB) de uma imagem de radar para uma determinada região. Se os valores ultrapassarem os limites pré-estabelecidos para chuva leve (amarelo) ou forte (vermelho), a função gera uma imagem do radar e a envia como alerta em um grupo do Telegram.

Usage

```
executar_alerta_telegram(
  mega = "Cianorte",
  chat_id,
  bot_token,
  raio = 50,
  vermelho = c(70, 25),
  amarelo = c(65, 33),
  coords_custom = NULL
)
```

Arguments

`mega`	Caractere. Nome da cidade ou ponto de interesse (ex: "Cianorte"). Padrão é "Cianorte".
`chat_id`	Caractere ou Numérico. ID do grupo/chat de destino no Telegram. Obrigatório.
`bot_token`	Caractere. Token do bot gerado pelo @BotFather. Obrigatório.

raio	Numérico. Raio de análise ao redor das coordenadas centrais em pixels. Padrão é 50.
vermelho	Vetor numérico de tamanho 2. Valores limite para considerar chuva forte. O primeiro valor é o mínimo para o canal Red (R), o segundo é o máximo para o canal Blue (B). Padrão é `c(70, 25)`.
amarelo	Vetor numérico de tamanho 2. Valores limite para considerar chuva leve. Padrão é `c(65, 33)`.
coords_custom	Lista nomeada opcional com valores de `x` e `y`. Usado se a cidade não estiver no dicionário padrão.

Details

A função depende das rotinas auxiliares `analisar_radar_PR()` e `gerar_imagem_radar()` para extrair as cores e montar o mapa. A comunicação com a API do Telegram é feita via requisição HTTP POST.

Value

Retorna invisivelmente `TRUE` se um alerta de chuva foi disparado com sucesso, ou `FALSE` se não houver condição de alerta.

Examples

```
if (interactive()) {
# Exemplo necessita de token e chatID 'Telegram'
executar_alerta_telegram(
  mega = "Cianorte",
  chat_id = "-1001951367890",
  bot_token = "SEU_TOKEN_AQUI",
  raio = 50,
  vermelho = c(80, 20),
  amarelo = c(60, 40)
)
}
```

`gerar_imagem_radar` *Gerar Imagem do Radar Meteorológico com Marcações Circulares*

Description

Esta função baixa a imagem do radar meteorológico do Paraná (ou utiliza uma já fornecida) e adiciona marcações circulares (raio de análise) ao redor das coordenadas de uma cidade ou ponto de interesse específico.

Usage

```
gerar_imagem_radar(
  cidade,
  raio,
  img_radar = NULL,
  coords_custom = NULL,
  caminho_salvar = NULL
)
```

Arguments

cidade	Caractere. Nome da cidade ou ponto de interesse (ex: "Cianorte"). Deve corresponder a uma chave na lista de coordenadas interna ou em coords_custom.
raio	Numérico. Raio da área de análise em pixels que será desenhado na imagem.
img_radar	Objeto magick-image opcional. Imagem de radar previamente baixada. Se NULL (padrão), a função tentará baixar uma nova imagem usando baixar_radar_PR().
coords_custom	Lista nomeada opcional. Permite adicionar novas coordenadas além das predefinidas. Formato esperado: list("NomeDaCidade" = list(x = 100, y = 200)).
caminho_salvar	Caractere opcional. Caminho completo (incluindo o nome do arquivo e extensão .png) para salvar a imagem gerada. Se NULL, salva em um diretório temporário.

Value

Retorna um objeto de imagem do 'magick' (magick-image) com as marcações desenhadas, ou NULL em caso de falha na obtenção da imagem. Como efeito colateral, salva a imagem gerada no disco.

Examples

```
# Gerar imagem para Cianorte com raio de 50 pixels (salva automaticamente em tempfile)
img_cianorte <- gerar_imagem_radar(cidade = "Cianorte", raio = 50)

# Adicionar uma nova cidade e salvar em um caminho temporario seguro para o 'CRAN'
novas_coords <- list("Londrina" = list(x = 520, y = 175))
caminho_teste <- tempfile(fileext = ".png")

img_londrina <- gerar_imagem_radar(
  cidade = "Londrina",
  raio = 50,
  coords_custom = novas_coords,
  caminho_salvar = caminho_teste
)
```

`Run_Monitor_RedeAgro` *Iniciar o Painel de Monitoramento da 'RedeAgro' ('RedeAgroRadar')*

Description

Lanca um aplicativo 'shiny' interativo para configurar e monitorar dados de radares meteorologicos da Rede Agropesquisa (Paraná). O painel permite buscar grupos na API do 'Telegram' (https://core.telegram.org/bots/api), testar o radar em tempo real e ativar um "robô" de monitoramento que roda a cada 10 minutos para enviar alertas baseados na intensidade da chuva.

Usage

```
Run_Monitor_RedeAgro(
  Token = "",
  chatID = "",
  limites_personalizados = list(vr = 70, vb = 25, ar = 65, ab = 33)
)
```

Arguments

`Token`	String de texto. O token de API gerado pelo '@BotFather' no 'Telegram'. Pode ser deixado em branco e preenchido diretamente na interface web.
`chatID`	String de texto. O ID do grupo ou chat do 'Telegram' que recebera os alertas. Pode ser deixado em branco e pesquisado via painel.
`limites_personalizados`	Lista nomeada. Define os limites globais de gatilho para os canais Red e Blue lidos do radar. Espera-se os elementos: `vr` (Alerta-Vermelho Red), `vb` (Alerta-Vermelho Blue), `ar` (Alerta-Amarelo Red) e `ab` (Alerta-Amarelo Blue). O padrao e `list(vr = 70, vb = 25, ar = 65, ab = 33)`. Chuva forte (vermelho) - (Red > 70 & Blue < 25) Chuva leve (amarelo) - (Red > 65 & Blue < 33). E utilizado a leitura de RGB do mapa do simepar.

Details

O aplicativo possui duas abas principais:

- **Configuracao do Telegram**: Para conectar um bot, buscar o ID do chat alvo e estabelecer a ponte de comunicacao.
- **Painel de Controle**: Permite definir a cidade monitorada, raio de analise em pixels, acionar o robo de envio de alertas ou fazer testes de radar de forma manual.

Value

Inicia uma interface web no navegador padrao rodando o app 'shiny'. A funcao nao retorna um objeto do R para o console.

Note

Para que a execucao automatica funcione perfeitamente, e imprescindivel que as funcoes internas auxiliares (`analisar_radar_PR`, `executar_alerta_telegram`, `gerar_imagem_radar` e `status_diario`) estejam carregadas no mesmo ambiente.

Examples

```
# Todos os exemplos que abrem servidores locais (como o 'shiny') devem
# estar dentro de if(interactive()) para nao travarem os testes de servidores.

if (interactive()) {
  # Exemplo 1: Iniciar o aplicativo sem credenciais iniciais
  Run_Monitor_RedeAgro()

  # Exemplo 2: Iniciar com Token e ChatID pre-configurados para teste
  Run_Monitor_RedeAgro(
    Token = "123456:ABC-DEF15858ghIkl-zyx57W2v1u12345ew11",
    chatID = "-10686784567890",
    limites_personalizados = list(vr = 80, vb = 20, ar = 60, ab = 30)
  )
}
```

---

status_diario *Envia Mensagem de Status com Imagem do Radar Meteorológico via Telegram*

Description

Esta função captura a imagem atual do radar meteorológico do Paraná e a envia para um chat do Telegram, servindo como um "sinal de vida" diário para indicar que o sistema de monitoramento está ativo e operando corretamente.

Usage

```
status_diario(
  bot_token,
  chat_id,
  hora_alerta = NULL,
  mensagem = paste0("Mensagem diaria de status. ",
    "Sistema de alerta meteorologico ativo e a funcionar perfeitamente.")
)
```

Arguments

`bot_token`	Caractere. Token de autenticação do bot do Telegram (fornecido pelo @BotFather).
`chat_id`	Caractere ou Numérico. ID do chat ou grupo do Telegram de destino.

hora_alerta	Caractere opcional. Horário programado para envio no formato "HH:MM". Se NULL (padrão), a mensagem é enviada imediatamente sem verificação de horas.
mensagem	Caractere. Texto da legenda que acompanhará a imagem. O padrão é uma mensagem confirmando que o sistema está ativo.

Details

A função depende de baixar_radar_PR() para obter a imagem atual. O arquivo é salvo em um diretório temporário e apagado automaticamente após o envio ou em caso de erro, respeitando as políticas do CRAN para manipulação de arquivos. A comunicação com o Telegram tem um *timeout* de segurança de 15 segundos.

Value

Retorna um valor lógico de forma invisível: TRUE para sucesso no envio, ou FALSE para falha (seja por erro de download, rede ou na API do Telegram).

Examples

```
if (interactive()) {
# Exemplo necessita de token e chatID 'Telegram'
# Enviar apenas se o horario local for exatamente 13:00
status_diario(
  bot_token = "SEU_TOKEN_AQUI",
  chat_id = "-1001234567890",
  hora_alerta = "13:00",
  mensagem = "Teste manual de status do radar."
)
}
```

Index

analisar_radar_PR, 2

baixar_radar_PR, 3

executar_alerta_telegram, 4

gerar_imagem_radar, 5

Run_Monitor_RedeAgro, 7

status_diario, 8