

# Package ‘bomrang’

September 12, 2018

**Type** Package

**Title** Australian Government Bureau of Meteorology (BOM) Data from R

**Version** 0.4.0

**Description** Provides functions to interface with Australian Government Bureau of Meteorology (BOM) data, fetching data and returning a tidy data frame of précis forecasts, historical and current weather data from stations, agriculture bulletin data, BOM 0900 or 1500 weather bulletins and downloading and importing radar and satellite imagery files. Data (c) Australian Government Bureau of Meteorology Creative Commons (CC) Attribution 3.0 licence or Public Access Licence (PAL) as appropriate. See <<http://www.bom.gov.au/other/copyright.shtml>> for further details.

**URL** <https://github.com/ropensci/bomrang>,  
<https://ropensci.github.io/bomrang/>

**BugReports** <https://github.com/ropensci/bomrang/issues>

**License** MIT + file LICENSE

**Depends** R (>= 3.2.0)

**Imports** curl (>= 2.8.1), data.table (>= 1.10.4), dplyr (>= 0.7.0), foreign, httr (>= 1.2.1), hoardr, janitor (>= 1.0.0), jsonlite (>= 1.5), lubridate, magrittr (>= 1.5), raster, rgdal, readr (>= 1.1.1), rvest, tidyr (>= 0.6.3), utils, xml2 (>= 1.1.1)

**Encoding** UTF-8

**LazyData** true

**Suggests** ASGS.foyer, covr, testthat, knitr, rmarkdown, ggplot2, ggthemes, grid, gridExtra, maps, mapproj, rappdirs, roxygen2 (>= 6.1.0)

**RoxyenNote** 6.1.0

**NeedsCompilation** no

**ByteCompile** TRUE

**VignetteBuilder** knitr

**X-schema.org-applicationCategory** Tools

**X-schema.org-keywords** bom, meteorological-data, weather-forecast, australia, weather, weather-data, meteorology, australia-bureau-of-meteorology

**X-schema.org-isPartOf** <https://ropensci.org>

**Author** Adam Sparks [aut, cre] (<<https://orcid.org/0000-0002-0061-8359>>), Jonathan Carroll [aut] (<<https://orcid.org/0000-0002-1404-5264>>), Dean Marchiori [aut], Mark Padgham [aut, rev] (<<https://orcid.org/0000-0003-2172-5265>>), Hugh Parsonage [aut] (<<https://orcid.org/0000-0003-4055-0835>>), Keith Pembleton [aut] (<<https://orcid.org/0000-0002-1896-4516>>), James Balamuta [ctb] (<<https://orcid.org/0000-0003-2826-8458>>), Brooke Anderson [rev] (<<https://orcid.org/0000-0002-5012-9035>>)

**Maintainer** Adam Sparks <[adamhsparks@gmail.com](mailto:adamhsparks@gmail.com)>

**Repository** CRAN

**Date/Publication** 2018-09-12 08:40:03 UTC

## R topics documented:

bomrang . . . . .	2
get_ag_bulletin . . . . .	3
get_available_imagery . . . . .	4
get_available_radar . . . . .	5
get_coastal_forecast . . . . .	6
get_current_weather . . . . .	8
get_historical . . . . .	9
get_precis_forecast . . . . .	11
get_radar_imagery . . . . .	12
get_satellite_imagery . . . . .	13
get_weather_bulletin . . . . .	15
manage_cache . . . . .	16
sweep_for_stations . . . . .	17
update_forecast_towns . . . . .	18
update_station_locations . . . . .	19

**Index** **20**

---

bomrang	<i>Australian Government Bureau of Meteorology (BOM) Data from R</i>
---------	----------------------------------------------------------------------

---

### Description

Australian Government Bureau of Meteorology (BOM) Data from R

### Author(s)

Adam H Sparks and Jonathan Carroll and Mark Padgham and Hugh Parsonage and Keith Pembleton

**See Also****Useful links:**

- Development repository: <https://github.com/ropensci/bomrang>
- Static documentation: <https://ropensci.github.io/bomrang/>
- Report bugs at <https://github.com/ropensci/bomrang/issues>

---

`get_ag_bulletin`*Get BOM Agriculture Bulletin Information for Select Stations*

---

**Description**

Fetch the BOM agricultural bulletin information and return it in a tidy data frame

**Usage**

```
get_ag_bulletin(state = "AUS")
```

**Arguments**

<code>state</code>	Australian state or territory as full name or postal code. Fuzzy string matching via <code>agrep</code> is done. Defaults to "AUS" returning all state bulletins, see details for more.
--------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

**Details**

Allowed state and territory postal codes, only one state per request or all using AUS.

**NSW** New South Wales

**NT** Northern Territory

**QLD** Queensland

**SA** South Australia

**TAS** Tasmania

**VIC** Victoria

**WA** Western Australia

**AUS** Australia, returns bulletin for all states and NT

**Value**

Tidy data frame of a Australia BOM agricultural bulletin information. For full details of fields and units returned see Appendix 3 in the **bomrang** vignette, use `vignette("bomrang", package = "bomrang")` to view.

**Author(s)**

Adam H Sparks, <adamhsparks@gmail.com>

## References

Agricultural observations are retrieved from the Australian Bureau of Meteorology (BOM) Weather Data Services Agriculture Bulletins,

<http://www.bom.gov.au/catalogue/observations/about-agricultural.shtml>

and

Australian Bureau of Meteorology (BOM) Weather Data Services Observation of Rainfall,

<http://www.bom.gov.au/climate/how/observations/rain-measure.shtml>

Station location and other metadata are sourced from the Australian Bureau of Meteorology (BOM) webpage, Bureau of Meteorology Site Numbers:

<http://www.bom.gov.au/climate/cdo/about/site-num.shtml>

## Examples

```
## Not run:
ag_bulletin <- get_ag_bulletin(state = "QLD")

## End(Not run)
```

---

get\_available\_imagery *Get a Listing of Available BOM Satellite GeoTIFF Imagery*

---

## Description

Fetch a listing of BOM GeoTIFF satellite imagery from <ftp://ftp.bom.gov.au/anon/gen/gms/> to determine which files are currently available for download. Files are available at ten minute update frequency with a 24 hour delete time. Useful to know the most recent files available and then specify in the [get\\_satellite\\_imagery](#) function.

## Usage

```
get_available_imagery(product_id = "all")
```

## Arguments

product_id	Character. BOM product ID of interest for which a list of available images will be returned. Defaults to all images currently available.
------------	------------------------------------------------------------------------------------------------------------------------------------------

## Details

Valid BOM satellite Product IDs for GeoTIFF files include:

**IDE00420** AHI cloud cover only 2km FD GEOS GIS

**IDE00421** AHI IR (Ch13) greyscale 2km FD GEOS GIS

**IDE00422** AHI VIS (Ch3) greyscale 2km FD GEOS GIS

**IDE00423** AHI IR (Ch13) Zehr 2km FD GEOS GIS

**IDE00425** AHI VIS (true colour) / IR (Ch13 greyscale) composite 1km FD GEOS GIS  
**IDE00426** AHI VIS (true colour) / IR (Ch13 greyscale) composite 2km FD GEOS GIS  
**IDE00427** AHI WV (Ch8) 2km FD GEOS GIS  
**IDE00430** AHI cloud cover only 2km AUS equirect. GIS  
**IDE00431** AHI IR (Ch13) greyscale 2km AUS equirect. GIS  
**IDE00432** AHI VIS (Ch3) greyscale 2km AUS equirect. GIS  
**IDE00433** AHI IR (Ch13) Zehr 2km AUS equirect. GIS  
**IDE00435** AHI VIS (true colour) / IR (Ch13 greyscale) composite 1km AUS equirect. GIS  
**IDE00436** AHI VIS (true colour) / IR (Ch13 greyscale) composite 2km AUS equirect. GIS  
**IDE00437** AHI WV (Ch8) 2km AUS equirect. GIS  
**IDE00439** AHI VIS (Ch3) greyscale 0.5km AUS equirect. GIS

**Value**

A vector of all available files for the requested Product ID(s).

**Author(s)**

Adam H Sparks, <adamhsparks@gmail.com>

**References**

Australian Bureau of Meteorology (BOM) High-definition satellite images <http://www.bom.gov.au/australia/satellite/index.shtml>

**Examples**

```
## Not run:  
Check availability of AHI VIS (true colour) / IR (Ch13 greyscale) composite  
1km FD GEOS GIS images  
imagery <- get_available_imagery(product_id = "IDE00425")  
  
## End(Not run)
```

---

get\_available\_radar     *Get a Listing of Available BOM Radar Imagery*

---

**Description**

Fetch a listing of available BOM radar imagery from <ftp://ftp.bom.gov.au/anon/gen/radar/> to determine which files are currently available for download. The files available are the most recent radar imagery for each location, which are updated approximately every 6 to 10 minutes by the BOM.

**Usage**

```
get_available_radar(radar_id = "all")
```

**Arguments**

radar\_id            Character. BOM radar ID of interest for which a list of available images will be returned. Defaults to all images currently available.

**Details**

Valid BOM radar ID for each location required.

**Value**

A data frame of all selected radar locations with location information and *product\_ids*.

**Author(s)**

Dean Marchiori, <deanmarchiori@gmail.com>

**References**

Australian Bureau of Meteorology (BOM) radar images <http://www.bom.gov.au/australia/radar/>

**Examples**

```
## Not run:  
Check availability radar imagey for Wollongong (radar_id = 3)  
imagery <- get_available_radar(radar_id = "3")  
  
## End(Not run)
```

---

get\_coastal\_forecast    *Get BOM Coastal Waters Forecast*

---

**Description**

Fetch the BOM daily Coastal Waters Forecast and return a tidy data frame of the forecast regions for a specified state or region.

**Usage**

```
get_coastal_forecast(state = "AUS")
```

## Arguments

**state** Australian state or territory as full name or postal code. Fuzzy string matching via [agrep](#) is done. Defaults to "AUS" returning all state forecasts, see details for further information.

## Details

Allowed state and territory postal codes, only one state per request or all using AUS.

**ACT** Australian Capital Territory (will return NSW)

**NSW** New South Wales

**NT** Northern Territory

**QLD** Queensland

**SA** South Australia

**TAS** Tasmania

**VIC** Victoria

**WA** Western Australia

**AUS** Australia, returns forecast for all states, NT and ACT

## Value

Tidy [data.frame](#) of a Australia BOM Coastal Waters Forecast.

## Author(s)

Dean Marchiori, <[deanmarchiori@gmail.com](mailto:deanmarchiori@gmail.com)>

## References

Forecast data come from Australian Bureau of Meteorology (BOM) Weather Data Services

<http://www.bom.gov.au/catalogue/data-feeds.shtml>

Location data and other metadata come from the BOM anonymous FTP server with spatial data

<ftp://ftp.bom.gov.au/anon/home/adfd/spatial/>, specifically the DBF file portion of a shape-file,

<ftp://ftp.bom.gov.au/anon/home/adfd/spatial/IDM00003.dbf>

## Examples

```
## Not run:  
coastal_forecast <- get_coastal_forecast(state = "NSW")  
  
## End(Not run)
```

---

get\_current\_weather    *Get Current Weather Observations of a BOM Station*

---

## Description

Get Current Weather Observations of a BOM Station

## Usage

```
get_current_weather(station_name, strict = FALSE, latlon = NULL,
  raw = FALSE, emit_latlon_msg = TRUE, as.data.table = FALSE)
```

## Arguments

station_name	The name of the weather station. Fuzzy string matching via <a href="#">agrep</a> is done.
strict	(logical) If TRUE, station_name must match the station name exactly, except that station_name need not be upper case. Note this may be different to full_name in the response. See <b>Details</b> .
latlon	A length-2 numeric vector giving the decimal degree latitude and longitude (in that order), e.g. latlon = c(-34, 151) for Sydney. When given instead of station_name, the nearest station (in this package) is used, with a message indicating the nearest such station. (See also <a href="#">sweep_for_stations</a> .) Ignored if used in combination with station_name, with a warning.
raw	Logical. Do not convert the columns data.table to the appropriate classes. (FALSE by default.)
emit_latlon_msg	Logical. If TRUE (the default), and latlon is selected, a message is emitted before the table is returned indicating which station was actually used (i.e. which station was found to be nearest to the given coordinate).
as.data.table	Return result as a <a href="#">data.table</a> .

## Details

Station names are not consistently named within the Bureau, so the response may contain a different full\_name to the one matched, even if *strict = TRUE*. For example, `get_current_weather("CASTLEMAINE PRISON")[[ "full_name" ]][1]` is Castlemaine, not Castlemaine Prison.

Note that the column local\_date\_time\_full is set to a POSIXct object in the local time of the **user**. For more details see the vignette "Current Weather Fields": `vignette("Current Weather Fields", package = "bomrang")` for a complete list of fields and units.

## Value

Tidy data frame of requested BOM station's current and prior 72hr data. For full details of fields and units returned, see Appendix 1 in the **bomrang** vignette, use `vignette("bomrang", package = "bomrang")` to view.



**Author(s)**

Hugh Parsonage, <hugh.parsonage@gmail.com>

**References**

Weather data observations are retrieved from: Australian Bureau of Meteorology (BOM) Weather Data Services, Observations - individual stations:

<http://www.bom.gov.au/catalogue/data-feeds.shtml>

Station location and other metadata are sourced from the Australian Bureau of Meteorology (BOM) webpage, Bureau of Meteorology Site Numbers:

<http://www.bom.gov.au/climate/cdo/about/site-num.shtml>

**Examples**

```
## Not run:
# warning
Melbourne_weather <- get_current_weather("Melbourne")

# no warning
Melbourne_weather <- get_current_weather("Melbourne (Olympic Park)")

# Get weather by latitude and longitude:
get_current_weather(latlon = c(-34, 151))

## End(Not run)
```

---

get\_historical

*Obtain Historical BOM Data*

---

**Description**

Retrieves daily observations for a given station.

**Usage**

```
get_historical(stationid = NULL, latlon = NULL, type = c("rain",
  "min", "max", "solar"), meta = FALSE)
```

**Arguments**

stationid	BOM station ID. See Details.
latlon	Length-2 numeric vector of Latitude/Longitude. See Details.
type	Measurement type, either daily "rain", "min" (temp), "max" (temp), or "solar" (exposure). Partial matching is performed. If not specified returns the first matching type in the order listed.
meta	Logical switch to include metadata information on the station and data from BOM. If set to TRUE a list is returned with a

**Value**

By default a complete `data.frame` of historical observations for the chosen station, with some subset of the following columns

<b>Product_code:</b>	BOM internal code.
<b>Station_number:</b>	BOM station ID.
<b>Year:</b>	Year of observation (YYYY).
<b>Month:</b>	Month of observation (1-12).
<b>Day:</b>	Day of observation (1-31).
<b>Min_temperature:</b>	Minimum daily recorded temperature (degrees C).
<b>Max_temperature:</b>	Maximum daily recorded temperature (degrees C).
<b>Accum_days_min:</b>	Accumulated number of days of minimum temperature.
<b>Accum_days_max:</b>	Accumulated number of days of maximum temperature.
<b>Rainfall:</b>	Daily recorded rainfall in mm.
<b>Period:</b>	Period over which rainfall was measured.
<b>Solar_exposure:</b>	Daily global solar exposure in MJ/m <sup>2</sup> .
<b>Quality:</b>	Y, N, or missing. Data which have not yet completed the routine quality control process are marked accordingly.

If *meta* is set TRUE, then a list is returned with an additional `data.frame` with the following columns giving information on the station and data.

<b>site:</b>	BOM station ID.
<b>name:</b>	BOM station name.
<b>lat:</b>	Latitude in decimal degrees.
<b>lon:</b>	Longitude in decimal degrees.
<b>start:</b>	Date observations start.
<b>end:</b>	Date observations end.
<b>years:</b>	Available number of years data.
<b>percent:</b>	Percent complete.
<b>AWS:</b>	Automated weather station?
<b>type:</b>	Measurement types available for the station.

Temperature data prior to 1910 should be used with extreme caution as many stations, prior to that date, were exposed in non-standard shelters, some of which give readings which are several degrees warmer or cooler than those measured according to post-1910 standards.

Daily maximum temperatures usually occur in the afternoon and daily minimum temperatures overnight or near dawn. Occasionally, however, the lowest temperature in the 24 hours to prior to 9 AM can occur around 9 AM the previous day if the night was particularly warm.

Either *stationid* or *latlon* must be provided, but if both are, then *stationid* will be used as it is more reliable.

In some cases data is available back to the 1800s, so tens of thousands of daily records will be returned. Other stations will be newer and will return fewer observations.

**Author(s)**

Jonathan Carroll, <rpkg@jcarroll.com.au>

**Examples**

```
## Not run:
get_historical(stationid = "023000", type = "max") ## ~48,000+ daily records
get_historical(latlon = c(-35.2809, 149.1300),
               type = "min") ## 3,500+ daily records

## End(Not run)
```

---

get\_precis\_forecast    *Get BOM Daily Précis Forecast for Select Towns*

---

**Description**

Fetch the BOM daily précis forecast and return a tidy data frame of the seven day town forecast for a specified state or territory.

**Usage**

```
get_precis_forecast(state = "AUS")
```

**Arguments**

**state**            Australian state or territory as full name or postal code. Fuzzy string matching via [agrep](#) is done. Defaults to "AUS" returning all state bulletins, see details for further information.

**Details**

Allowed state and territory postal codes, only one state per request or all using AUS.

**ACT** Australian Capital Territory (will return NSW)

**NSW** New South Wales

**NT** Northern Territory

**QLD** Queensland

**SA** South Australia

**TAS** Tasmania

**VIC** Victoria

**WA** Western Australia

**AUS** Australia, returns forecast for all states, NT and ACT

**Value**

Tidy `data.frame` of a Australia BOM précis seven day forecasts for select towns. For full details of fields and units returned see Appendix 2 in the **bomrang** vignette, use `vignette("bomrang", package = "bomrang")` to view.

**Author(s)**

Adam H Sparks, <adamhsparks@gmail.com> and Keith Pembleton, <keith.pembleton@usq.edu.au>

**References**

Forecast data come from Australian Bureau of Meteorology (BOM) Weather Data Services <http://www.bom.gov.au/catalogue/data-feeds.shtml>

Location data and other metadata for towns come from the BOM anonymous FTP server with spatial data

<ftp://ftp.bom.gov.au/anon/home/adfd/spatial/>, specifically the DBF file portion of a shapefile,

<ftp://ftp.bom.gov.au/anon/home/adfd/spatial/IDM00013.dbf>

**Examples**

```
## Not run:
BOM_forecast <- get_precis_forecast(state = "QLD")

## End(Not run)
```

---

get\_radar\_imagery      *Get BOM Radar Imagery*

---

**Description**

Fetch BOM radar imagery from <ftp://ftp.bom.gov.au/anon/gen/radar/> and return a raster `raster` object. Files available are the most recent radar snapshot which are updated approximately every 6 to 10 minutes. Suggested to check file availability first by using `get_available_radar`.

**Usage**

```
get_radar_imagery(product_id, path = NULL, download_only = FALSE)
```

**Arguments**

product_id	Character. BOM product ID to download and import as a <code>raster</code> object. Value is required.
path	Character. A character string with the name where the downloaded file is saved. If not provided, the default value NULL is used which saves the file in a temp directory.
download_only	Logical. Whether the radar image is loaded into the environment as a <code>raster</code> layer, or just downloaded.

**Details**

Valid BOM satellite Product IDs for radar imagery can be obtained from [get\\_available\\_radar](#).

**Value**

A raster layer based on the most recent '.gif' radar image snapshot published by the BOM. If `download_only = TRUE` there will be a NULL return value with the download path printed in the console as a message.

**Author(s)**

Dean Marchiori, <deanmarchiori@gmail.com>

**References**

Australian Bureau of Meteorology (BOM) radar images <http://www.bom.gov.au/australia/radar/>

**See Also**

[get\\_available\\_radar](#)

**Examples**

```
## Not run:
# Fetch most recent radar image for Wollongong 256km radar
imagery <- get_radar_imagery(product_id = "IDR032")
raster::plot(imagery)

# Save imagery to a local path
imagery <- get_radar_imagery(product_id = "IDR032", path = 'image.gif')

## End(Not run)
```

---

get\_satellite\_imagery *Get BOM Satellite GeoTIFF Imagery*

---

**Description**

Fetch BOM satellite GeoTIFF imagery from <ftp://ftp.bom.gov.au/anon/gen/gms/> and return a raster [stack](#) object of GeoTIFF files. Files are available at ten minute update frequency with a 24 hour delete time. Suggested to check file availability first by using [get\\_available\\_imagery](#).

**Usage**

```
get_satellite_imagery(product_id, scans = 1, cache = FALSE)
```

**Arguments**

product_id	Character. BOM product ID to download in GeoTIFF format and import as a <a href="#">stack</a> object. A vector of values from <a href="#">get_available_imagery</a> may be used here. Value is required.
scans	Numeric. Number of scans to download, starting with most recent and progressing backwards, <i>e.g.</i> , 1 - the most recent single scan available, 6 - the most recent hour available, 12 - the most recent 2 hours available, etc. Negating will return the oldest files first. Defaults to 1. Value is optional.
cache	Logical. Store image files locally for later use? If FALSE, the downloaded files are removed when R session is closed. To take advantage of cached files in future sessions, use <code>cache = TRUE</code> . Defaults to FALSE. Value is optional.

**Details**

Valid BOM satellite Product IDs include:

- IDE00420** AHI cloud cover only 2km FD GEOS GIS
- IDE00421** AHI IR (Ch13) greyscale 2km FD GEOS GIS
- IDE00422** AHI VIS (Ch3) greyscale 2km FD GEOS GIS
- IDE00423** AHI IR (Ch13) Zehr 2km FD GEOS GIS
- IDE00425** AHI VIS (true colour) / IR (Ch13 greyscale) composite 1km FD GEOS GIS
- IDE00426** AHI VIS (true colour) / IR (Ch13 greyscale) composite 2km FD GEOS GIS
- IDE00427** AHI WV (Ch8) 2km FD GEOS GIS
- IDE00430** AHI cloud cover only 2km AUS equirect. GIS
- IDE00431** AHI IR (Ch13) greyscale 2km AUS equirect. GIS
- IDE00432** AHI VIS (Ch3) greyscale 2km AUS equirect. GIS
- IDE00433** AHI IR (Ch13) Zehr 2km AUS equirect. GIS
- IDE00435** AHI VIS (true colour) / IR (Ch13 greyscale) composite 1km AUS equirect. GIS
- IDE00436** AHI VIS (true colour) / IR (Ch13 greyscale) composite 2km AUS equirect. GIS
- IDE00437** AHI WV (Ch8) 2km AUS equirect. GIS
- IDE00439** AHI VIS (Ch3) greyscale 0.5km AUS equirect. GIS

We cache using **hoardr**, find your cache folder by executing `manage_cache$cache_path_get`.

**Value**

A raster stack of GeoTIFF images with layers named by BOM Product ID, timestamp and band.

**Author(s)**

Adam H Sparks, <adamhsparks@gmail.com>

**References**

Australian Bureau of Meteorology (BOM) high-definition satellite images  
<http://www.bom.gov.au/australia/satellite/index.shtml>

**See Also**

[get\\_available\\_imagery](#) [manage\\_cache](#)

**Examples**

```
## Not run:
# Fetch AHI VIS (true colour) / IR (Ch13 greyscale) composite 1km FD
# GEOS GIS raster stack for most recent single scan available

imagery <- get_satellite_imagery(product_id = "IDE00425", scans = 1)

# Get a list of available image files and use that to specify files for
# download, downloading the two most recent files available

avail <- get_available_imagery(product_id = "IDE00425")
imagery <- get_satellite_imagery(product_id = avail, scans = 2)

## End(Not run)
```

---

get\_weather\_bulletin *Get BOM 0900 or 1500 Weather Bulletin*

---

**Description**

Fetch the daily BOM 0900 or 1500 weather bulletins and return a tidy data frame for a specified state or territory.

**Usage**

```
get_weather_bulletin(state = "qld", morning = TRUE)
```

**Arguments**

state	Australian state or territory as full name or postal code. Fuzzy string matching via <a href="#">agrep</a> is done.
morning	If TRUE, return the 9am bulletin for the nominated state; otherwise return the 3pm bulletin.

**Details**

Allowed state and territory postal codes:

**ACT** Australian Capital Territory (will return NSW)

**NSW** New South Wales

**NT** Northern Territory

**QLD** Queensland

**SA** South Australia

**TAS** Tasmania

**VIC** Victoria

**WA** Western Australia

It is not possible to return weather bulletins for the entire country in a single call. Rainfall figures for the 9am bulletin are generally for the preceding 24 hours, while those for the 3pm bulletin are for the preceding 6 hours since 9am. Note that values are manually entered into the bulletins and sometimes contain typographical errors which may lead to warnings about "NAs introduced by coercion".

### Value

Tidy data frame of Australian 9am or 3pm weather observations for a state. For full details of fields and units returned see Appendix 4 in the **bomrang** vignette, use `vignette("bomrang", package = "bomrang")` to view.

### Author(s)

Mark Padgham, <mark.padgham@email.com>

### References

Daily observation data come from Australian Bureau of Meteorology (BOM) website. The 3pm bulletin for Queensland is, for example, [http://www.bom.gov.au/qld/observations/3pm\\_bulletin.shtml](http://www.bom.gov.au/qld/observations/3pm_bulletin.shtml)

### Examples

```
## Not run:
qld_weather <- get_weather_bulletin (state = "QLD", morning = FALSE)

## End(Not run)
```

---

manage\_cache

*Manage Locally Cached BOM Files*

---

### Description

Manage cached **bomrang** satellite imagery files with **hoardr**.

### Details

The default cache directory is `file.path(rappdirs::user_cache_dir(), "R/bomrang")`, but you can set your own path using `manage_cache$cache_path_set()`

`manage_cache$cache_delete` only accepts one file name, while `manage_cache$cache_delete_all` does not accept any names, but deletes all files. For deleting many specific files, use `manage_cache$cache_delete` in an `lapply` type call.



**Useful user functions**

- `manage_cache$cache_path_get()` - get cache path
- `manage_cache$cache_path_set()` - set cache path
- `manage_cache$list()` - returns a character vector of full path file names
- `manage_cache$files()` - returns file objects with metadata
- `manage_cache$details()` - returns files with details
- `manage_cache$delete()` - delete specific files
- `manage_cache$delete_all()` - delete all files, returns nothing

**Examples**

```
## Not run:  
  
# list files in cache  
manage_cache$list()  
  
# delete certain database files  
manage_cache$delete("file path")  
manage_cache$list()  
  
# delete all files in cache  
manage_cache$delete_all()  
manage_cache$list()  
  
# set a different cache path from the default  
manage_cache$cache_path_set("~/tmp")  
  
## End(Not run)
```

---

sweep\_for\_stations      *Find Nearest BOM Weather Stations*

---

**Description**

Find Nearest BOM Weather Stations

**Usage**

```
sweep_for_stations(latlon = c(-35.3, 149.2))
```

**Arguments**

latlon                    A length-2 numeric vector. By default, Canberra (approximately).

**Value**

A data frame of all weather stations (in this package) sorted by distance from latlon, ascending.

**Author(s)**

Hugh Parsonage, <hugh.parsonage@gmail.com>

---

update\_forecast\_towns *Update bomrang Internal Database with Latest BOM Forecast Towns*

---

**Description**

Download the latest select forecast towns from the BOM server and update bomrang's internal database of précis forecast town names and AAC codes used by [get\\_precis\\_forecast](#). There is no need to use this unless you know that a forecast town exists in a more current version of the BOM précis forecast town name database that is not available in the database distributed with [bomrang].

**Usage**

```
update_forecast_towns()
```

**Value**

Updated database of BOM précis forecast towns

**Author(s)**

Adam H Sparks, <adamhsparks@gmail.com>

**References**

Data are sourced from: Australian Bureau of Meteorology (BOM) webpage, "Weather Data Services", <http://www.bom.gov.au/catalogue/data-feeds.shtml>

**Examples**

```
## Not run:  
update_forecast_towns()  
  
## End(Not run)
```

---

`update_station_locations`*Update bomrang Internal Databases with Latest BOM Station Metadata*

---

### Description

Download the latest station locations and metadata and update bomrang's internal databases that support the use of `get_current_weather`, `get_ag_bulletin` and `get_historical`. There is no need to use this unless you know that a station exists in BOM's database that is not available in the databases distributed with **bomrang**. In fact, for reproducibility purposes, users are discouraged from using this function.

### Usage

```
update_station_locations()
```

### Details

If **ASGS.foyer** is installed locally, this function will automatically check and correct any invalid state values for stations located in Australia. If **ASGS.foyer** is not installed, the function will update the internal database without validating the state values for stations by reported lon/lat location.

### Value

Updated internal databases of BOM station locations and JSON URLs

### Author(s)

Adam H Sparks, <adamhsparks@gmail.com>

### References

Station location and other metadata are sourced from the Australian Bureau of Meteorology (BOM) webpage, Bureau of Meteorology Site Numbers:  
<http://www.bom.gov.au/climate/cdo/about/site-num.shtml>

### Examples

```
## Not run:  
update_station_locations()  
  
## End(Not run)
```

# Index

`agrep`, [3](#), [7](#), [8](#), [11](#), [15](#)

`bomrang`, [2](#)

`bomrang-package` (`bomrang`), [2](#)

`data.frame`, [7](#), [10](#), [12](#)

`data.table`, [8](#)

`get_ag_bulletin`, [3](#), [19](#)

`get_available_imagery`, [4](#), [13–15](#)

`get_available_radar`, [5](#), [12](#), [13](#)

`get_coastal_forecast`, [6](#)

`get_current_weather`, [8](#), [19](#)

`get_historical`, [9](#), [19](#)

`get_precis_forecast`, [11](#), [18](#)

`get_radar_imagery`, [12](#)

`get_satellite_imagery`, [4](#), [13](#)

`get_weather_bulletin`, [15](#)

`lapply`, [16](#)

`manage_cache`, [15](#), [16](#)

`raster`, [12](#)

`stack`, [13](#), [14](#)

`sweep_for_stations`, [8](#), [17](#)

`update_forecast_towns`, [18](#)

`update_station_locations`, [19](#)