

# Package ‘pargasite’

July 23, 2025

**Title** Pollution-Associated Risk Geospatial Analysis Site

**Version** 2.1.1

**Description** Offers tools to estimate and visualize levels of major pollutants (CO, NO<sub>2</sub>, SO<sub>2</sub>, Ozone, PM<sub>2.5</sub> and PM<sub>10</sub>) across the conterminous United States for user-defined time ranges. Provides functions to retrieve pollutant data from the U.S. Environmental Protection Agency’s ‘Air Quality System’ (AQS) API service <[https://aqs.epa.gov/aqsweb/documents/data\\_api.html](https://aqs.epa.gov/aqsweb/documents/data_api.html)> for interactive visualization through a ‘shiny’ application, allowing users to explore pollutant levels for a given location over time relative to the National Ambient Air Quality Standards (NAAQS).

**License** BSD\_3\_clause + file LICENSE

**Encoding** UTF-8

**RoxygenNote** 7.3.2

**Imports** cli, gstat, leaflet, leafsync, methods, raqs, rlang, sf (>= 1.0-15), shiny, shinycssloaders, stars (>= 0.6-5), stats, utils

**Depends** R (>= 4.1.0)

**LazyData** true

**Suggests** knitr, rmarkdown

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** Jaehyun Joo [aut, cre],  
Rebecca Greenblatt [aut],  
Avantika Diwadkar [aut],  
Nisha Narayanan [aut],  
Blanca Himes [aut]

**Maintainer** Jaehyun Joo <jaehyunjoo@outlook.com>

**Repository** CRAN

**Date/Publication** 2024-11-16 15:50:02 UTC

## Contents

pargasite-package . . . . .	2
create_pargasite_data . . . . .	2
create_raster . . . . .	5
list_pollutant_standards . . . . .	6
ozone20km . . . . .	7
run_pargasite . . . . .	8

<b>Index</b>	<b>9</b>
--------------	----------

---

pargasite-package	<i>pargasite: Pollution-Associated Risk Geospatial Analysis Site</i>
-------------------	--

---

### Description

Offers tools to estimate and visualize levels of major pollutants (CO, NO<sub>2</sub>, SO<sub>2</sub>, Ozone, PM<sub>2.5</sub> and PM<sub>10</sub>) across the conterminous United States for user-defined time ranges. Provides functions to retrieve pollutant data from the U.S. Environmental Protection Agency's 'Air Quality System' (AQS) API service [https://aq5.epa.gov/aq5web/documents/data\\_api.html](https://aq5.epa.gov/aq5web/documents/data_api.html) for interactive visualization through a 'shiny' application, allowing users to explore pollutant levels for a given location over time relative to the National Ambient Air Quality Standards (NAAQS).

### Author(s)

**Maintainer:** Jaehyun Joo <jaehyunjoo@outlook.com>

Authors:

- Rebecca Greenblatt
- Avantika Diwadkar
- Nisha Narayanan
- Blanca Himes

---

create_pargasite_data	<i>Create a data cube for air pollutant levels covering the conterminous United States</i>
-----------------------	--

---

### Description

A function to create a raster-based pollutant concentration input for pargasite's shiny application. It downloads pollutant data via the Environmental Protection Agency's (EPA) Air Quality System (AQS) API service, filters the data by exceptional event (e.g., wildfire) status, and performs the inverse distance weighted (IDW) interpolation to estimate pollutant concentrations covering the conterminous United States (CONUS) at user-defined time ranges.

**Usage**

```

create_pargasite_data(
  pollutant = c("CO", "SO2", "NO2", "Ozone", "PM2.5", "PM10"),
  data_field = c("NAAQS_statistic", "arithmetic_mean"),
  event_filter = c("Events Included", "Events Excluded", "Concurred Events Excluded"),
  year,
  by_month = FALSE,
  cell_size = 10000,
  nmax = Inf,
  aqs_email = get_aqs_email(),
  aqs_key = get_aqs_key(),
  download_chunk_size = c("2-week", "month")
)

```

**Arguments**

pollutant	A string specifying an air pollutant to create a raster data cube. Must be one of CO2, SO2, NO2, Ozone, PM2.5 and PM10.
data_field	A vector of strings specifying whether which data fields are used to summarize the data. Must be either 'NAAQS_statistic', 'arithmetic_mean', or both. 'NAAQS_statistic' try to chooses an appropriate field based on National Ambient Air Quality Standards (NAAQS) in the AQS yearly data (e.g, for CO 1-hour average, 'second_max_value' would be chosen). 'arithmetic_mean' represents the measure of central tendency in the yearly data. Ignored when by_month = TRUE.
event_filter	A vector of strings indicating whether data measured during exceptional events are included in the summary. 'Events Included' means that events occurred and the data from theme is included in the summary. 'Events Excluded' means that events occurred but data from them is excluded from the summary. 'Concurred Events Excluded' means that events occurred but only EPA concurred exclusions are removed from the summary. If multiple values are specified, pollutant levels for each filter are stored in event dimension in the resulting output.
year	A vector of 4-digit numeric values specifying years to retrieve pollutant levels.
by_month	A logical value indicating whether data summarized at monthly level instead of yearly level.
cell_size	A numeric value specifying a cell size of grid cells in meters.
nmax	An integer value specifying the number of nearest observations that should be used for spatial interpolation.
aqs_email	A string specifying the registered email for AQS API service.
aqs_key	A string specifying the registered key for AQS API service.
download_chunk_size	A string specifying a chunk size for AQS API daily data download to prevent an unexpected server timeout error. Ignored when by_month = FALSE.

## Details

By default, it returns yearly-summarized concentrations using AQS's annual data but can also provide monthly-summarized concentrations by aggregating AQS's daily data. Note that the function chooses an appropriate data field for each pollutant to check the air quality status based on the National Ambient Air Quality Standard (NAAQS) for yearly-summarized outputs as follows

- CO 1-hour: second\_max\_value field
- CO 8-hour: second\_max\_nonoverlap field
- SO2 1-hour: ninety\_ninth\_percentile field
- NO2 1-hour: nineth\_eighth\_percentile field
- NO2 Annual: arithmetic\_mean field
- Ozone 8-hour: fourth\_max\_value field
- PM10 24-hour: primary\_exceedance\_count field
- PM25 24-hour: ninety\_eighth\_percentile field
- PM25 Annual: arithmetic\_mean field

For monthly-summarized outputs, it uses the arithmetic\_mean field of daily data. Please check AQS API metaData/fieldsByService (see [raqs::metadata\\_fieldsbyservice](#)) and [AQS data dictionary](#) for the details of field descriptions.

For spatial interpolation, the AQS data is projected to EPSG:6350 (NAD83 CONUS Albers), and thus, cell\_size value is represented in meters (5,000 creates 5km x 5km grid). The smaller cell\_size, the more processing time is required.

## Value

A stars object containing the interpolated pollutant levels over CONUS.

## Examples

```
## Not run:

## Set your AQS API key first using [raqs::set_aqs_user] to run the example.

## SO2 and CO concentrations through 2020 to 2022
so2 <- create_pargasite_data("SO2", "Events Included", year = 2020:2022)
co <- create_pargasite_data("CO", "Events Included", year = 2020:2022)

## Combine them; can combine other pollutant grids in the same way
pargasite_input <- c(so2, co)

## End(Not run)
```

---

create_raster	<i>Raster creation function</i>
---------------	---------------------------------

---

## Description

This function is used internally by [create\\_pargasite\\_data](#), allowing users to specify a longitude/latitude bounding box. It is exported for package development, not for direct user calls.

## Usage

```
create_raster(
  parameter_code,
  pollutant_standard = NULL,
  data_field = c("NAAQS_statistic", "arithmetic_mean"),
  event_filter = c("Events Included", "Events Excluded", "Concurred Events Excluded"),
  year,
  by_month = FALSE,
  minlat = 24,
  maxlat = 50,
  minlon = -124,
  maxlon = -66,
  crs = 6350,
  cell_size = 10000,
  aqs_email = get_aqs_email(),
  aqs_key = get_aqs_key(),
  nmax = Inf,
  download_chunk_size = c("2-week", "month")
)
```

## Arguments

parameter_code	A numeric value specifying a AQS parameter code.
pollutant_standard	A vector of strings specifying unique pollutant standards in AQS (e.g., PM25 Annual 2012).
data_field	A vector of strings specifying whether which data fields are used to summarize the data. Must be either 'NAAQS_statistic', 'arithmetic_mean', or both. 'NAAQS_statistic' try to chooses an appropriate field based on National Ambient Air Quality Standards (NAAQS) in the AQS yearly data (e.g, for CO 1-hour average, 'second_max_value' would be chosen). 'arithmetic_mean' represents the measure of central tendency in the yearly data. Ignored when by_month = TRUE.
event_filter	A vector of strings indicating whether data measured during exceptional events are included in the summary. 'Events Included' means that events occurred and the data from theme is included in the summary. 'Events Excluded' means that events occurred but data from them is excluded from the summary. 'Concurred

	Events Excluded' means that events occurred but only EPA concurred exclusions are removed from the summary. If multiple values are specified, pollutant levels for each filter are stored in event dimension in the resulting output.
year	A vector of 4-digit numeric values specifying years to retrieve pollutant levels.
by_month	A logical value indicating whether data summarized at monthly level instead of yearly level.
minlat, maxlat, minlon, maxlon	A numeric value specifying a bounding box defined by two latitudes and two longitudes.
crs	A target coordinate reference system.
cell_size	A numeric value specifying a cell size of grid cells in meters.
aqs_email	A string specifying the registered email for AQS API service.
aqs_key	A string specifying the registered key for AQS API service.
nmax	An integer value specifying the number of nearest observations that should be used for spatial interpolation.
download_chunk_size	A string specifying a chunk size for AQS API daily data download to prevent an unexpected server timeout error. Ignored when by_month = FALSE.

**Value**

A stars object containing the interpolated pollutant levels covering the user-specified bounding box.

**Examples**

```
## Not run:

## Set your AQS API key first using [raqs::set_aqs_user] to run the example.

## SO2 through 2021 to 2022
so2 <- create_raster(42401, event_filter = "Events Included",
                    year = 2021:2022)

## End(Not run)
```

---

list\_pollutant\_standards

*Look up National Ambient Air Quality Standards (NAAQS)*

---

**Description**

A function to show pollutant standard information on criteria air pollutants.

**Usage**

```
list_pollutant_standards(parameter_code = NULL, detail = FALSE)
```

**Arguments**

`parameter_code` An AQS parameter code to retrieve specific standard information. If NULL (default), display all pollutant standards.

`detail` A logical value indicating whether detailed information is retrieved.

**Value**

A data.frame containing the pollutant standard information

**References**

Data source: [https://aqs.epa.gov/aqsweb/documents/codetables/pollutant\\_standards.html](https://aqs.epa.gov/aqsweb/documents/codetables/pollutant_standards.html)

**Examples**

```
list_pollutant_standards()
```

---

ozone20km

*Ozone concentration data*


---

**Description**

A stars object containing ozone concentrations covering the conterminous United States for years 2021-2022 on a 20km x 20km grid.

**Usage**

```
ozone20km
```

**Format**

A stars object with 4 dimensions and 1 attribute

**x, y** dimension; raster coordinates

**event** dimension; exceptional event filter

**year** dimension; years of data available

**ozone\_8\_hour\_2015** attribute; ozone levels based on Ozone 8-hour 2015 standard

---

`run_pargasite`*Run pargasite application*

---

**Description**

Launch a Shiny application to visualize pollutant levels of the conterminous US. The system's default web browser will be launched automatically after the app is started.

**Usage**

```
run_pargasite(x)
```

**Arguments**

`x` A stars object returned by [create\\_pargasite\\_data](#).

**Value**

This function normally does not return; interrupt R to stop the application (usually by pressing Ctrl + C or ESC).

**Examples**

```
if (interactive()) {  
  run_pargasite(ozone20km)  
}
```



# Index

## \* datasets

- ozone20km, [7](#)
- create\_pargasite\_data, [2](#), [5](#), [8](#)
- create\_raster, [5](#)
- list\_pollutant\_standards, [6](#)
- ozone20km, [7](#)
- pargasite (pargasite-package), [2](#)
- pargasite-package, [2](#)
- raqs::metadata\_fieldsbyservice, [4](#)
- run\_pargasite, [8](#)