

Package: climatehealth (via r-universe)

May 8, 2026

Title Statistical Tools for Modelling Climate-Health Impacts

Version 1.0.1

Date 2026-04-08

Description Tools for producing climate-health indicators and supporting official statistics from health and climate data. Implements analytical workflows for temperature-related mortality, wildfire smoke exposure, air pollution, suicides related to extreme heat, malaria, and diarrhoeal disease outcomes, with utilities for descriptive statistics, model validation, attributable fraction and attributable number estimation, relative risk estimation, minimum mortality temperature estimation, and plotting for reporting. These six indicators are endorsed by the United Nations Statistical Commission for inclusion in the Global Set of Environment and Climate Change Statistics. Implemented methods include distributed lag non-linear models (DLNM), quasi-Poisson time-series regression, case-crossover analysis, Bayesian spatio-temporal models using the Integrated Nested Laplace Approximation ('INLA'), and multivariate meta-analysis for sub-national estimates. The package is based on methods developed in the Standards for Official Statistics on Climate-Health Interactions (SOSCHI) project <<https://climate-health.officialstatistics.org>>. For methodologies, see Watkins et al. (2025) <[doi:10.5281/zenodo.14865904](https://doi.org/10.5281/zenodo.14865904)>, Brown et al. (2024) <[doi:10.5281/zenodo.14052183](https://doi.org/10.5281/zenodo.14052183)>, Pearce et al. (2024) <[doi:10.5281/zenodo.14050224](https://doi.org/10.5281/zenodo.14050224)>, Byukusenge et al. (2025) <[doi:10.5281/zenodo.15585042](https://doi.org/10.5281/zenodo.15585042)>, Dzakpa et al. (2025) <[doi:10.5281/zenodo.14881886](https://doi.org/10.5281/zenodo.14881886)>, and Dzakpa et al. (2025) <[doi:10.5281/zenodo.14871506](https://doi.org/10.5281/zenodo.14871506)>.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.3

Imports car, data.table, dlnm, dplyr, Epi, forcats, exactextractr,

ggplot2, ggtext, gnm, graphics, grDevices, gplots, lifecycle, lme4, lubridate, metafor, mgcv, mixmeta, ncd4, patchwork, pkgbuild, purrr, raster, RColorBrewer, readr, readxl, reshape2, rlang, scales, sf, spdep, splines, stats, stringr, tibble, tidy, tools, tseries, tsModel (>= 0.6-2), utils, xfun, zoo

VignetteBuilder knitr

Suggests covr, knitr, rmarkdown, devtools, DT, htmltools, INLA, mockery, mvmeta, openxlsx, patrick, pkgload, stringdist, terra, testthat (>= 3.2.1.1), withr

URL <https://climate-health.officialstatistics.org>

Additional_repositories <https://inla.r-inla-download.org/R/stable/>

Depends R (>= 4.4.0)

Config/rmdcheck/ignore-inconsequential-notes true

NeedsCompilation no

Author Charlie Browning [aut], Kenechi Omeke [aut, cre], Etse Yawo Dzakpa [aut], Gladin Jose [aut], Matt Pearce [aut], Ellie Watkins [aut], Claire Hunt [aut], Beatrice Byukusenge [aut], Cassien Habyarimana [aut], Venuste Nyagahakwa [aut], Felix Scarbrough [aut], Treesa Shaji [aut], Bonnie Lewis [aut], Maquines Odhiambo Sewe [aut], Vijendra Ingole [aut], Sean Lovell [ctb], Antony Brown [ctb], Euan Soutter [ctb], Gillian Flower [ctb], David Furley [ctb], Joe Panes [ctb], Charlotte Romaniuk [ctb], Milly Powell [ctb], Wellcome [fnd], Office for National Statistics [cph] (SOSCHI Project)

Maintainer Kenechi Omeke <climate.health@ons.gov.uk>

Repository <https://cran.r-universe.dev>

Date/Publication 2026-04-08 22:03:07 UTC

RemoteUrl <https://github.com/cran/climatehealth>

RemoteRef HEAD

RemoteSha 88d19c2542408ac34327978c7437b2fe253446f5

Contents

climatehealth-package	3
abort_climate	4
abort_column_not_found	5
abort_model_error	6
abort_validation	7
air_pollution_do_analysis	8
diarrhea_do_analysis	12
install_INLA	15
install_terra	16
is_climate_error	16

malaria_do_analysis	17
run_descriptive_stats	20
run_descriptive_stats_api	22
suicides_heat_do_analysis	24
temp_mortality_do_analysis	28
wildfire_do_analysis	32

Index	37
--------------	-----------

climatehealth-package *climatehealth: Statistical Tools for Modelling Climate-Health Impacts*

Description

Overview:

This package provides a suite of analysis functions for measuring the relationship between various climate factors (indicators) and health outcomes.

Included Indicators:

- Mortality attributable to high and low outdoor temperatures
- Mortality attributable to wildfire-related PM2.5
- Suicides attributable to extreme heat
- Mortality attributable to short-term exposure to outdoor PM2.5 exposure
- Diarrhea cases attributable to extreme temperatures and rainfall
- Malaria cases attributable to extreme temperatures and rainfall

License:

MIT

The full range of topics include:

- Temperature-related health effects
- Health effects of wildfires
- Mental Health
- Health effects of air pollution
- Water-borne diseases
- Vector-borne diseases

Author(s)

Maintainer: Kenechi Omeke <climate.health@ons.gov.uk>

Authors:

- Charlie Browning
- Etse Yawo Dzakpa
- Gladin Jose

- Matt Pearce
- Ellie Watkins
- Claire Hunt
- Beatrice Byukusenge
- Cassien Habyarimana
- Venuste Nyagahakwa
- Felix Scarbrough
- Treesa Shaji
- Bonnie Lewis
- Maquines Odhiambo Sewe
- Vijendra Ingole

Other contributors:

- Sean Lovell [contributor]
- Antony Brown [contributor]
- Euan Soutter [contributor]
- Gillian Flower [contributor]
- David Furley [contributor]
- Joe Panes [contributor]
- Charlotte Romaniuk [contributor]
- Milly Powell [contributor]
- Wellcome [funder]
- Office for National Statistics (SOSCHI Project) [copyright holder]

See Also

Useful links:

- <https://climate-health.officialstatistics.org>

abort_climate

Raise a typed error with structured metadata

Description

Creates a classed condition that can be caught and inspected by the API layer. This is the base helper - prefer using specific helpers like `abort_column_not_found()` or `abort_validation()` when applicable.

Usage

```
abort_climate(message, type = "generic_error", ..., call = rlang::caller_env())
```

Arguments

message	Human-readable error message
type	Error type for classification. One of: <ul style="list-style-type: none"> • "validation_error": Data/parameter validation issues (HTTP 400) • "column_not_found": Missing column in dataset (HTTP 400) • "model_error": Statistical model failures (HTTP 422) • "generic_error": Unclassified errors (HTTP 500)
...	Additional metadata to include in the error (e.g., column = "tmean")
call	The call to include in the error (defaults to caller's call)

Value

Never returns; always raises an error.

Examples

```
# Basic usage
err <- tryCatch(
  abort_climate("Something went wrong", "generic_error"),
  error = identity
)
inherits(err, "climate_error")

# With metadata
err <- tryCatch(
  abort_climate(
    "Invalid lag value",
    "validation_error",
    param = "nlag",
    value = -1,
    expected = "non-negative integer"
  ),
  error = identity
)
err$type
```

abort_column_not_found

Raise a column-not-found error with available columns

Description

Use this when a required column is missing from a dataset. Includes fuzzy matching to suggest the closest available column name.

Usage

```
abort_column_not_found(  
  column,  
  available,  
  dataset_name = "dataset",  
  call = rlang::caller_env()  
)
```

Arguments

column	The column name that was not found
available	Character vector of available column names
dataset_name	Optional name of the dataset for clearer messages
call	The call to include in the error

Value

Never returns; always raises an error.

Examples

```
data <- data.frame(temp = 1)  
if (!("tmean" %in% colnames(data))) {  
  err <- tryCatch(  
    abort_column_not_found("tmean", colnames(data)),  
    error = identity  
  )  
  err$suggestion  
}
```

abort_model_error *Raise a model error (statistical/computational failures)*

Description

Use this when statistical models fail to converge, produce singular matrices, or encounter other computational issues that aren't due to obvious user error.

Usage

```
abort_model_error(  
  message,  
  model_type = "unknown",  
  ...,  
  call = rlang::caller_env()  
)
```

Arguments

message	Human-readable error message
model_type	Type of model that failed (e.g., "dlnm", "glm", "meta-analysis")
...	Additional diagnostic metadata
call	The call to include in the error

Value

Never returns; always raises an error.

Examples

```
tryCatch({
  stop("convergence failed")
}, error = function(e) {
  err <- tryCatch(
    abort_model_error(
      "Model failed to converge",
      model_type = "dlnm",
      original_error = conditionMessage(e)
    ),
    error = identity
  )
  inherits(err, "model_error")
})
```

abort_validation	<i>Raise a validation error (data/parameter issues)</i>
------------------	---

Description

Use this for general validation failures where the user's input or data doesn't meet requirements. For missing columns specifically, use `abort_column_not_found()`.

Usage

```
abort_validation(message, ..., call = rlang::caller_env())
```

Arguments

message	Human-readable error message
...	Additional metadata (e.g., param = "nlag", value = -1)
call	The call to include in the error

Value

Never returns; always raises an error.

Examples

```
# Parameter validation
nlag <- -1
if (nlag < 0) {
  err <- tryCatch(
    abort_validation(
      "nlag must be >= 0",
      param = "nlag",
      value = nlag,
      expected = "non-negative integer"
    ),
    error = identity
  )
  inherits(err, "validation_error")
}
```

air_pollution_do_analysis

Comprehensive Air Pollution Analysis Pipeline

Description

Master function that runs the complete air pollution analysis including data loading, preprocessing (including lags), modeling, plotting, attribution calculations vs reference standards, power analysis and descriptive statistics

Usage

```
air_pollution_do_analysis(
  data_path,
  date_col = "date",
  region_col = "region",
  pm25_col = "pm25",
  deaths_col = "deaths",
  population_col = "population",
  humidity_col = "humidity",
  precipitation_col = "precipitation",
  tmax_col = "tmax",
  wind_speed_col = "wind_speed",
  categorical_others = NULL,
  continuous_others = NULL,
  Categorical_Others = NULL,
```

```

Continuous_Others = NULL,
max_lag = 14L,
df_seasonal = 6,
family = "quasipoisson",
reference_standards = list(list(value = 15, name = "WHO")),
output_dir = "air_pollution_results",
save_outputs = TRUE,
run_descriptive = TRUE,
run_power = TRUE,
moving_average_window = 3L,
include_national = TRUE,
years_filter = NULL,
regions_filter = NULL,
attr_thr = 95,
plot_corr_matrix = TRUE,
correlation_method = "pearson",
plot_dist = TRUE,
plot_na_counts = TRUE,
plot_scatter = TRUE,
plot_box = TRUE,
plot_seasonal = TRUE,
plot_regional = TRUE,
plot_total = TRUE,
detect_outliers = TRUE,
calculate_rate = FALSE
)

```

Arguments

data_path	Character. Path to CSV data file
date_col	Character. Name of date column
region_col	Character. Name of region column
pm25_col	Character. Name of PM2.5 column
deaths_col	Character. Name of deaths column
population_col	Character. Name of the population column.
humidity_col	Character. Name of humidity column
precipitation_col	Character. Name of precipitation column
tmax_col	Character. Name of temperature column
wind_speed_col	Character. Name of wind speed column
categorical_others	Optional character vector. Names of additional categorical variables.
continuous_others	Optional character vector. Names of additional continuous variables (e.g., "tmean")
Categorical_Others	Deprecated alias for categorical_others.

Continuous_Others	Deprecated alias for continuous_others.
max_lag	Integer. Maximum lag days. Defaults to 14.
df_seasonal	Integer. Degrees of freedom for seasonal spline. Default 6.
family	Character. Character. Probability distribution for the outcome variable. Options include "quasipoisson" (default: "quasipoisson")
reference_standards	List of reference standards, each with "PM2.5 value" and "name of of standard (e.g. WHO)"
output_dir	Directory to save outputs
save_outputs	Logical. Whether to save outputs
run_descriptive	Logical. Whether to run descriptive statistics
run_power	Logical. Whether to run power analysis
moving_average_window	Integer. Window for moving average in descriptive stats
include_national	Logical. Whether to include national results in plots. Default TRUE.
years_filter	Optional numeric vector of years to include (e.g., c(2020, 2021, 2022)). It is recommended to filter for at least 3 consecutive years for a minimum considerable time series
regions_filter	Optional character vector of regions to include
attr_thr	Numeric (0-100). Percentile threshold used in power analysis to evaluate attribution detectability. Default 95.
plot_corr_matrix	Logical. Plot correlation matrix. Default TRUE.
correlation_method	Character. Correlation method for corr matrix (e.g., "pearson", "spearman"). Default "pearson".
plot_dist	Logical. Plot distributions (hist/density) for key variables. Default TRUE.
plot_na_counts	Logical. Plot missingness/NA counts. Default TRUE.
plot_scatter	Logical. Plot scatter plots for key pairs. Default TRUE.
plot_box	Logical. Plot boxplots by region/season where applicable. Default TRUE.
plot_seasonal	Logical. Plot seasonal summaries. Default TRUE.
plot_regional	Logical. Plot regional summaries. Default TRUE.
plot_total	Logical. Plot overall totals where relevant. Default TRUE.
detect_outliers	Logical. Flag potential outliers in descriptive workflow. Default TRUE.
calculate_rate	Logical. Whether to calculate rate variables during descriptive stats (e.g., deaths per population). Default FALSE

Value

List containing:

data Processed data with lag variables

meta_analysis Meta-analysis results with AF/AN calculations

lag_analysis Lag-specific analysis results

distributed_lag_analysis Distributed lag model results (if requested)

plots List of generated plots (forest, lags, distributed lags)

power_list A list containing power information by area

exposure_response_plots Exposure-response plots for each reference standard (if requested)

reference_specific_af_an AF/AN calculations specific to each reference standard (if requested)

descriptive_stats Summary statistics of key variables

Examples

```
example_data <- data.frame(
  date = seq.Date(as.Date("2020-01-01"), by = "day", length.out = 180),
  province = "Example Province",
  pm25 = stats::runif(180, 8, 35),
  deaths = stats::rpois(180, lambda = 5),
  population = 500000,
  humidity = stats::runif(180, 40, 90),
  precipitation = stats::runif(180, 0, 20),
  tmax = stats::runif(180, 18, 35),
  wind_speed = stats::runif(180, 1, 8)
)
example_path <- tempfile(fileext = ".csv")
utils::write.csv(example_data, example_path, row.names = FALSE)
```

```
results <- air_pollution_do_analysis(
  data_path = example_path,
  date_col = "date",
  region_col = "province",
  pm25_col = "pm25",
  deaths_col = "deaths",
  population_col = "population",
  humidity_col = "humidity",
  precipitation_col = "precipitation",
  tmax_col = "tmax",
  wind_speed_col = "wind_speed",
  continuous_others = NULL,
  max_lag = 7L,
  df_seasonal = 4,
  family = "quasipoisson",
  reference_standards = list(list(value = 15, name = "WHO")),
  years_filter = NULL,
  regions_filter = NULL,
  include_national = FALSE,
  output_dir = tempdir(),
```

```
save_outputs = FALSE,  
run_descriptive = FALSE,  
run_power = FALSE,  
moving_average_window = 3L,  
attr_thr = 95,  
plot_corr_matrix = FALSE,  
correlation_method = "pearson",  
plot_dist = FALSE,  
plot_na_counts = FALSE,  
plot_scatter = FALSE,  
plot_box = FALSE,  
plot_seasonal = FALSE,  
plot_regional = FALSE,  
plot_total = FALSE,  
detect_outliers = FALSE,  
calculate_rate = FALSE  
)
```

diarrhea_do_analysis *Code for calculating Diarrhea disease cases attributable to extreme precipitation and extreme temperature Run Full diarrhea-Climate Analysis Pipeline*

Description

The `diarrhea_do_analysis` function runs the complete analysis workflow by combining multiple functions to analyze the association between diarrhea cases and climate variables. It processes health, climate, and spatial data, fits models, generates plots, and calculates attributable risk.

Usage

```
diarrhea_do_analysis(  
  health_data_path,  
  climate_data_path,  
  map_path,  
  region_col,  
  district_col,  
  date_col = NULL,  
  year_col,  
  month_col,  
  case_col,  
  tot_pop_col,  
  tmin_col,  
  tmean_col,  
  tmax_col,  
  rainfall_col,
```

```

    r_humidity_col,
    runoff_col,
    geometry_col,
    spi_col = NULL,
    ndvi_col = NULL,
    max_lag = 2,
    nk = 2,
    basis_matrices_choices,
    inla_param,
    param_term,
    level,
    param_threshold = 1,
    filter_year = NULL,
    family = "nbinomial",
    group_by_year = FALSE,
    config = TRUE,
    save_csv = FALSE,
    save_model = TRUE,
    save_fig = FALSE,
    cumulative = FALSE,
    output_dir = NULL
)

```

Arguments

health_data_path	Character. Path to the processed health data file.
climate_data_path	Character. Path to the processed climate data file.
map_path	Character. Path to the spatial data file (e.g., shapefile).
region_col	Character. Column name for the region variable.
district_col	Character. Column name for the district variable.
date_col	Character (optional). Column name for the date variable. Defaults to NULL.
year_col	Character. Column name for the year variable.
month_col	Character. Column name for the month variable.
case_col	Character. Column name for diarrhea case counts.
tot_pop_col	Character. Column name for total population.
tmin_col	Character. Column name for minimum temperature.
tmean_col	Character. Column name for mean temperature.
tmax_col	Character. Column name for maximum temperature.
rainfall_col	Character. Column name for cumulative monthly rainfall.
r_humidity_col	Character. Column name for relative humidity.
runoff_col	Character. Column name for monthly runoff data.
geometry_col	Character. Column name of the geometry column in the shapefile (usually "geometry").

spi_col	Character (optional). Column name for the Standardized Precipitation Index (SPI). Defaults to NULL.
ndvi_col	Character (optional). Column name for the Normalized Difference Vegetation Index (NDVI). Defaults to NULL.
max_lag	Numeric. Maximum temporal lag to include in the distributed lag model (e.g., 2-4). Defaults to 2.
nk	Numeric. Number of internal knots for the natural spline of each predictor, controlling its flexibility: $nk = 0$ produces a linear effect with one basis column, $nk = 1$ generates a simple spline with two columns, $nk = 2$ yields a more flexible spline with three columns, and higher values of nk further increase flexibility but may also raise collinearity among spline terms. Defaults to 2.
basis_matrices_choices	Character vector. Specifies which climate variables to include in the basis matrix (e.g., <code>c("tmax", "rainfall", "r_humidity")</code>).
inla_param	Character vector. Specifies exposure variables included in the INLA model (e.g., <code>c("tmin", "rainfall", "r_humidity")</code>).
param_term	Character or vector. Exposure variable(s) of primary interest for relative risk and attribution (e.g., <code>"tmax", "rainfall"</code>).
level	Character. Spatial disaggregation level; must be one of <code>"country", "region",</code> or <code>"district"</code> .
param_threshold	Numeric. Threshold above which exposure is considered "attributable." Defaults to 1.
filter_year	Integer or vector (optional). Year(s) to filter the data by. Defaults to NULL.
family	Character. Probability distribution for the outcome variable. Options include <code>"poisson"</code> (default) and <code>"nbinomial"</code> for a negative binomial model.
group_by_year	Logical. Whether to group attributable metrics by year. Defaults to FALSE.
config	Logical. Whether to enable additional INLA model configurations. Defaults to TRUE.
save_csv	Logical. If TRUE, saves intermediate datasets to CSV. Defaults to TRUE.
save_model	Logical. If TRUE, saves fitted INLA model results. Defaults to TRUE.
save_fig	Logical. If TRUE, saves generated plots. Defaults to TRUE.
cumulative	Boolean. If TRUE, plot and save cumulative risk of all year for the specific exposure at region and district level. Defaults to FALSE.
output_dir	Character. Directory where output files (plots, datasets, maps) are saved. Defaults to NULL.

Value

A list containing:

- Model output from INLA
- Monthly random effects plot
- Yearly random effects plot

- Contour plot
- Relative risk map
- Relative risk plot
- Attributable fraction and number summary

`install_INLA`*Install the INLA Package from Its Official Repository*

Description

This function installs the INLA package from its official repository at <https://inla.r-inla-download.org/R/stable/>. On Windows, it checks whether Rtools is available and installs the official binary package directly.

Usage

```
install_INLA(os = .Platform$OS.type)
```

Arguments

`os` The current operating system. Defaults to `.Platform$OS.type`.

Details

On Windows systems, the function verifies that Rtools is installed using `pkgbuild::has_build_tools()`. If Rtools is missing, it displays a warning and aborts the installation. The function then installs the matching Windows binary package from the official INLA repository.

On non-Windows systems, the package is installed normally from the repository.

Value

Invisibly returns `NULL`. The function is called for its side effect.

Examples

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

install_terra	<i>Install the terra Package from the CRAN Archive</i>
---------------	--

Description

This function installs the terra package at version 1.8-60 from the CRAN archive.

Usage

```
install_terra(os = .Platform$OS.type)
```

Arguments

os The current operating system. Defaults to .Platform\$OS.type.

Details

On Windows systems, the function verifies that Rtools is installed using `pkgbuild::has_build_tools()`. If Rtools is missing, it displays a warning and aborts the installation. The function then forces installation from source.

Value

Invisibly returns NULL. The function is called for its side effect.

Examples

```
## Not run:
install_terra()

## End(Not run)
```

is_climate_error	<i>Check if an error is a climate_error</i>
------------------	---

Description

Utility function to check if a caught condition is a typed climate error.

Usage

```
is_climate_error(error)
```

Arguments

error A condition object

Value

TRUE if the error inherits from "climate_error", FALSE otherwise.

Examples

```
tryCatch({
  stop("example error")
}, error = function(e) {
  if (is_climate_error(e)) {
    # Handle structured error
  } else {
    # Handle untyped error
  }
})
```

malaria_do_analysis *Code for calculating Malaria disease cases attributable to extreme rainfall and extreme temperature Run Full Malaria-Climate Analysis Pipeline*

Description

The `Malaria_do_analysis()` function executes the complete workflow for analyzing the association between malaria cases and climate variables. It integrates health, climate, and spatial data; fits spatio-temporal models using INLA; and generates a suite of diagnostic and inferential outputs, including plots and attributable risk estimates.

Usage

```
malaria_do_analysis(
  health_data_path,
  climate_data_path,
  map_path,
  region_col,
  district_col,
  date_col = NULL,
  year_col,
  month_col,
  case_col,
  tot_pop_col,
  tmin_col,
  tmean_col,
  tmax_col,
  rainfall_col,
  r_humidity_col,
  runoff_col,
```

```

geometry_col,
spi_col = NULL,
ndvi_col = NULL,
max_lag = 2,
nk = 2,
basis_matrices_choices,
inla_param,
param_term,
level,
param_threshold = 1,
filter_year = NULL,
family = "nbinomial",
group_by_year = FALSE,
cumulative = FALSE,
config = FALSE,
save_csv = FALSE,
save_model = FALSE,
save_fig = FALSE,
output_dir = NULL
)

```

Arguments

health_data_path	Character. Path to the processed health data file.
climate_data_path	Character. Path to the processed climate data file.
map_path	Character. Path to the spatial data file (e.g., shapefile).
region_col	Character. Column name for the region variable.
district_col	Character. Column name for the district variable.
date_col	Character (optional). Column name for the date variable. Defaults to NULL.
year_col	Character. Column name for the year variable.
month_col	Character. Column name for the month variable.
case_col	Character. Column name for malaria case counts.
tot_pop_col	Character. Column name for total population.
tmin_col	Character. Column name for minimum temperature.
tmean_col	Character. Column name for mean temperature.
tmax_col	Character. Column name for maximum temperature.
rainfall_col	Character. Column name for cumulative monthly rainfall.
r_humidity_col	Character. Column name for relative humidity.
runoff_col	Character. Column name for monthly runoff data.
geometry_col	Character. Column name of the geometry column in the shapefile (usually "geometry").

spi_col	Character (optional). Column name for the Standardized Precipitation Index (SPI). Defaults to NULL.
ndvi_col	Character (optional). Column name for the Normalized Difference Vegetation Index (NDVI). Defaults to NULL.
max_lag	Numeric. Maximum temporal lag to include in the distributed lag model (e.g., 2-4). Defaults to 4.
nk	Numeric. Number of internal knots for the natural spline of each predictor, controlling its flexibility: $nk = 0$ produces a linear effect with one basis column, $nk = 1$ generates a simple spline with two columns, $nk = 2$ yields a more flexible spline with three columns, and higher values of nk further increase flexibility but may also raise collinearity among spline terms. Defaults to 2.
basis_matrices_choices	Character vector. Specifies which climate variables to include in the basis matrix (e.g., <code>c("tmax", "rainfall", "r_humidity")</code>).
inla_param	Character vector. Specifies exposure variables included in the INLA model (e.g., <code>c("tmin", "rainfall", "r_humidity")</code>).
param_term	Character or vector. Exposure variable(s) of primary interest for relative risk and attribution (e.g., <code>"tmax", "rainfall"</code>).
level	Character. Spatial disaggregation level; must be one of <code>"country", "region",</code> or <code>"district"</code> .
param_threshold	Numeric. Threshold above which exposure is considered "attributable." Defaults to 1.
filter_year	Integer or vector (optional). Year(s) to filter the data by. Defaults to NULL.
family	Character. Probability distribution for the outcome variable. Options include <code>"poisson"</code> (default) and <code>"nbinomial"</code> for a negative binomial model.
group_by_year	Logical. Whether to group attributable metrics by year. Defaults to FALSE.
cumulative	Boolean. If TRUE, plot and save cumulative risk of all year for the specific exposure at region and district level. Defaults to FALSE.
config	Logical. Whether to enable additional INLA model configurations. Defaults to TRUE.
save_csv	Logical. If TRUE, saves intermediate datasets to CSV. Defaults to TRUE.
save_model	Logical. If TRUE, saves fitted INLA model results. Defaults to TRUE.
save_fig	Logical. If TRUE, saves generated plots. Defaults to TRUE.
output_dir	Character. Directory where output files (plots, datasets, maps) are saved. Defaults to NULL.

Value

A named list containing:

- `inla_result` - Fitted INLA model object and summaries.
- `plot_malaria`, `plot_tmax`, `plot_rainfall` - Exploratory time-series plots.
- `reff_plot_monthly` - Monthly random effects plot.

- reff_plot_yearly - Yearly spatial random effects plot.
- contour_plot - Exposure-response contour plot.
- rr_map_plot - Spatial relative risk map.
- rr_plot, rr_df - Relative risk plot and associated data.
- attr_frac_num - Attributable risk summary table.
- plot_AR_num, plot_AR_frac, plot_AR_per_100k - Plots of attributable number, fraction, and rate.

run_descriptive_stats *Run generic descriptive statistics and EDA outputs for indicator datasets.*

Description

Run generic descriptive statistics and EDA outputs for indicator datasets.

Usage

```
run_descriptive_stats(  
  data,  
  output_path,  
  aggregation_column = NULL,  
  population_col = NULL,  
  plot_corr_matrix = FALSE,  
  correlation_method = "pearson",  
  plot_dist = FALSE,  
  plot_ma = FALSE,  
  ma_days = 100,  
  ma_sides = 1,  
  timeseries_col = NULL,  
  dependent_col,  
  independent_cols,  
  units = NULL,  
  plot_na_counts = FALSE,  
  plot_scatter = FALSE,  
  plot_box = FALSE,  
  plot_seasonal = FALSE,  
  plot_regional = FALSE,  
  plot_total = FALSE,  
  detect_outliers = FALSE,  
  calculate_rate = FALSE,  
  run_id = NULL,  
  create_base_dir = FALSE  
)
```

Arguments

data	Dataframe or named list of dataframes. If a dataframe is provided and aggregation_column is passed, data are split by that column.
output_path	Character. Base output directory.
aggregation_column	Character. Column used to aggregate/split data by region.
population_col	Character. The column containing population data.
plot_corr_matrix	Logical. Whether to plot correlation matrix.
correlation_method	Character. Correlation method. One of 'pearson', 'spearman', 'kendall'.
plot_dist	Logical. Whether to plot distribution histograms.
plot_ma	Logical. Whether to plot moving averages over a timeseries.
ma_days	Integer. Number of days to use for moving average.
ma_sides	Integer. Sides to use for moving average (1 or 2).
timeseries_col	Character. Timeseries column used for moving averages and time-based plots.
dependent_col	Character. Dependent variable column.
independent_cols	Character vector. Independent variable columns.
units	Named character vector. Units for variables.
plot_na_counts	Logical. Whether to plot NA counts.
plot_scatter	Logical. Whether to plot scatter plots.
plot_box	Logical. Whether to plot box plots.
plot_seasonal	Logical. Whether to plot seasonal trends.
plot_regional	Logical. Whether to plot regional trends.
plot_total	Logical. Whether to plot total health outcomes by year.
detect_outliers	Logical. Whether to output an outlier table.
calculate_rate	Logical. Whether to plot annual rates per 100k.
run_id	Character. Optional run id. If NULL, a timestamped id is generated.
create_base_dir	Logical. Whether to create output_path if missing.

Value

A list with base_output_path, run_id, run_output_path, and region_output_paths.

Examples

```
df <- data.frame(  
  date = as.Date("2024-01-01") + 0:29,  
  region = rep(c("A", "B"), each = 15),  
  outcome = sample(1:20, 30, replace = TRUE),  
  temp = rnorm(30, 25, 3)  
)  
  
run_descriptive_stats(  
  data = df,  
  output_path = tempdir(),  
  aggregation_column = "region",  
  dependent_col = "outcome",  
  independent_cols = c("temp"),  
  timeseries_col = "date",  
  run_id = NULL  
)
```

run_descriptive_stats_api

Create descriptive statistics via API-friendly inputs.

Description

Create descriptive statistics via API-friendly inputs.

Usage

```
run_descriptive_stats_api(  
  data,  
  output_path,  
  aggregation_column = NULL,  
  population_col = NULL,  
  dependent_col,  
  independent_cols,  
  units = NULL,  
  plot_corr_matrix = FALSE,  
  plot_dist = FALSE,  
  plot_ma = FALSE,  
  plot_na_counts = FALSE,  
  plot_scatter = FALSE,  
  plot_box = FALSE,  
  plot_seasonal = FALSE,  
  plot_regional = FALSE,  
  plot_total = FALSE,  
  correlation_method = "pearson",
```

```

    ma_days = 100,
    ma_sides = 1,
    timeseries_col = NULL,
    detect_outliers = FALSE,
    calculate_rate = FALSE,
    run_id = NULL,
    create_base_dir = TRUE
)

```

Arguments

data	The dataset for descriptive stats (list-like object or CSV path).
output_path	Character. Base output directory.
aggregation_column	Character. Column used to aggregate/split data by region.
population_col	Character. The column containing the population.
dependent_col	Character. The dependent column.
independent_cols	Character vector. The independent columns.
units	Named character vector. Units for each variable.
plot_corr_matrix	Logical. Whether to plot a correlation matrix.
plot_dist	Logical. Whether to plot histograms.
plot_ma	Logical. Whether to plot moving averages over a timeseries.
plot_na_counts	Logical. Whether to plot counts of NAs in each column.
plot_scatter	Logical. Whether to plot dependent vs independent columns.
plot_box	Logical. Whether to generate box plots for selected columns.
plot_seasonal	Logical. Whether to plot seasonal trends.
plot_regional	Logical. Whether to plot regional trends.
plot_total	Logical. Whether to plot total dependent values per year.
correlation_method	Character. Correlation method. One of 'pearson', 'spearman', 'kendall'.
ma_days	Integer. Number of days used in moving average calculations.
ma_sides	Integer. Number of sides used in moving average calculations (1 or 2).
timeseries_col	Character. Timeseries column.
detect_outliers	Logical. Whether to output an outlier table.
calculate_rate	Logical. Whether to plot annual rates per 100k.
run_id	Character. Optional run id.
create_base_dir	Logical. Whether to create output_path if missing. Defaults to TRUE.

Value

A list with `base_output_path`, `run_id`, `run_output_path`, and `region_output_paths`.

Examples

```
run_descriptive_stats_api(
  data = list(
    date = as.character(as.Date("2024-01-01") + 0:29),
    region = rep(c("A", "B"), each = 15),
    outcome = sample(1:20, 30, replace = TRUE),
    temp = rnorm(30, 25, 3)
  ),
  output_path = tempdir(),
  aggregation_column = "region",
  dependent_col = "outcome",
  independent_cols = c("temp"),
  timeseries_col = "date",
  plot_corr_matrix = TRUE
)
```

`suicides_heat_do_analysis`

Full analysis pipeline for the suicides and extreme heat indicator

Description

Runs the full pipeline to analyse the impact of extreme heat on suicides using a time-stratified case-crossover approach with distributed lag non-linear model. This function generates relative risk of the suicide-temperature association as well as attributable numbers, rates and fractions of suicides to a specified temperature threshold. Model validation statistics are also provided.

Usage

```
suicides_heat_do_analysis(
  data_path,
  date_col,
  region_col = NULL,
  temperature_col,
  health_outcome_col,
  population_col,
  country = "National",
  meta_analysis = FALSE,
  var_fun = "bs",
  var_degree = 2,
  var_per = c(25, 50, 75),
  lag_fun = "strata",
```

```

lag_breaks = 1,
lag_days = 2,
independent_cols = NULL,
control_cols = NULL,
cenper = 50,
attr_thr = 97.5,
save_fig = FALSE,
save_csv = FALSE,
output_folder_path = NULL,
seed = NULL
)

```

Arguments

<code>data_path</code>	Path to a csv file containing a daily time series of data for a particular health outcome and climate variables, which may be disaggregated by region.
<code>date_col</code>	Character. Name of the column in the dataframe that contains the date.
<code>region_col</code>	Character. Name of the column in the dataframe that contains the region names. Defaults to NULL.
<code>temperature_col</code>	Character. Name of the column in the dataframe that contains the temperature column.
<code>health_outcome_col</code>	Character. Name of the column in the dataframe that contains the health outcome count column (e.g. number of deaths, hospital admissions).
<code>population_col</code>	Character. Name of the column in the dataframe that contains the population estimate column.
<code>country</code>	Character. Name of country for national level estimates.
<code>meta_analysis</code>	Boolean. Whether to perform a meta-analysis.
<code>var_fun</code>	Character. Exposure function for argvar (see <code>dlm::crossbasis</code>). Defaults to 'bs'.
<code>var_degree</code>	Integer. Degree of the piecewise polynomial for argvar (see <code>dlm::crossbasis</code>). Defaults to 2 (quadratic).
<code>var_per</code>	Vector. Internal knot positions for argvar (see <code>dlm::crossbasis</code>). Defaults to <code>c(25,50,75)</code> .
<code>lag_fun</code>	Character. Exposure function for arglag (see <code>dlm::crossbasis</code>). Defaults to 'strata'.
<code>lag_breaks</code>	Integer. Internal cut-off point defining the strata for arglag (see <code>dlm::crossbasis</code>). Defaults to 1.
<code>lag_days</code>	Integer. Maximum lag. Defaults to 2. (see <code>dlm::crossbasis</code>).
<code>independent_cols</code>	Additional independent variables to test in model validation
<code>control_cols</code>	A list of confounders to include in the final model adjustment. Defaults to NULL if none.
<code>cenper</code>	Integer. Value for the percentile in calculating the centering value 0-100. Defaults to 50.

attr_thr	Integer. Percentile at which to define the temperature threshold for calculating attributable risk.
save_fig	Boolean. Whether to save the plot as an output. Defaults to FALSE.
save_csv	Boolean. Whether to save the results as a CSV. Defaults to FALSE.
output_folder_path	Path to folder where plots and/or CSV should be saved. Defaults to NULL.
seed	Optional integer random seed used when sampling residuals for model validation plots. Defaults to NULL.

Details

This analysis pipeline requires a daily time series of temperature and suicide deaths with population values as a minimum. This is then processed using a conditional Poisson case-crossover analysis with distributed lag non-linear model and optional meta-analysis. Meta-analysis is recommended if the input data is disaggregated by area.

The model parameters have default values, which are recommended to keep as based on existing studies. However, if desired these can be adjusted for sensitivity analysis.

Model validation testing is provided as a standard output from the pipeline so a user can assess the quality of the model. If a user has additional independent variables these can be specified as `independent_cols` and assessed within different model combinations in the outputs of this testing. These can be added in the final model via `control_cols`.

For attributable deaths the default is to use extreme heat as a threshold, defined as the 97.5th percentile of temperature over the corresponding time period for each geography. This can be adjusted if desired, following review of the relative risk association between temperature and suicides, using `attr_thr`.

Further details on the input data requirements, methodology, quality information and guidance on interpreting outputs can be found in the accompanying published [doi:10.5281/zenodo.14050224](https://doi.org/10.5281/zenodo.14050224).

Value

- `qaic_results` A dataframe of QAIC and dispersion metrics for each model combination and geography.
- `qaic_summary` A dataframe with the mean QAIC and dispersion metrics for each model combination.
- `vif_results` A dataframe. Variance inflation factors for each independent variables by region.
- `vif_summary` A dataframe with the mean variance inflation factors for each independent variable.
- `meta_test_res` A dataframe of results from statistical tests on the meta model.
- `power_list` A list containing power information by area.
- `rr_results` Dataframe containing cumulative relative risk and confidence intervals from analysis.
- `res_attr_tot` Dataframe. Total attributable fractions, numbers and rates for each area over the whole time series.

- `attr_yr_list` List. Dataframes containing yearly estimates of attributable fractions, numbers and rates by area.
- `attr_mth_list` List. Dataframes containing total attributable fractions, numbers and rates by calendar month and area.

References

1. Pearce M, Watkins E, Glickman M, Lewis B, Ingle V. Standards for Official Statistics on Climate-Health Interactions (SOSCHI): Suicides attributed to extreme heat: methodology. Zenodo; 2024. Available from: [doi:10.5281/zenodo.14050224](https://doi.org/10.5281/zenodo.14050224)
2. Gasparrini A, Guo Y, Hashizume M, Lavigne E, Zanobetti A, Schwartz J, et al. Mortality risk attributable to high and low ambient temperature: a multicountry observational study. *Lancet*. 2015 Jul;386(9991):369-75. Available from: <https://linkinghub.elsevier.com/retrieve/pii/S0140673614621140>
3. Kim Y, Kim H, Gasparrini A, Armstrong B, Honda Y, Chung Y, et al. Suicide and Ambient Temperature: A Multi-Country Multi-City Study. *Environ Health Perspect*. 2019 Nov;127(11):1-10. Available from: <https://pubmed.ncbi.nlm.nih.gov/31769300/>
4. Gasparrini A, Armstrong B. Reducing and meta-analysing estimates from distributed lag non-linear models. *BMC Med Res Methodol*. 2013 Jan 9;13:1. Available from: [doi:10.1186/14712288131](https://doi.org/10.1186/14712288131)
5. Gasparrini A, Armstrong B, Kenward MG. Multivariate meta-analysis for non-linear and other multi-parameter associations. *Stat Med*. 2012 Dec 20;31(29):3821-39. Available from: [doi:10.1002/sim.5471](https://doi.org/10.1002/sim.5471)
6. Sera F, Armstrong B, Blangiardo M, Gasparrini A. An extended mixed-effects framework for meta-analysis. *Stat Med*. 2019 Dec 20;38(29):5429-44. Available from: [doi:10.1002/sim.8362](https://doi.org/10.1002/sim.8362)
7. Gasparrini A, Leone M. Attributable risk from distributed lag models. *BMC Med Res Methodol*. 2014 Dec 23;14(1):55. Available from: <https://link.springer.com/article/10.1186/1471-2288-14-55>

Examples

```
example_data <- data.frame(
  date = seq.Date(as.Date("2020-01-01"), by = "day", length.out = 365),
  region = "Example Region",
  tmean = stats::runif(365, 5, 30),
  suicides = stats::rpois(365, lambda = 2),
  pop = 250000
)
example_path <- tempfile(fileext = ".csv")
utils::write.csv(example_data, example_path, row.names = FALSE)

suicides_heat_do_analysis(
  data_path = example_path,
  date_col = "date",
  region_col = "region",
  temperature_col = "tmean",
  health_outcome_col = "suicides",
```

```
population_col = "pop",
country = "Example Region",
meta_analysis = FALSE,
var_fun = "bs",
var_degree = 2,
var_per = c(25, 50, 75),
lag_fun = "strata",
lag_breaks = 1,
lag_days = 2,
independent_cols = NULL,
control_cols = NULL,
cenper = 50,
attr_thr = 97.5,
save_fig = FALSE,
save_csv = FALSE,
output_folder_path = tempdir()
)
```

temp_mortality_do_analysis

Full analysis for the 'mortality attributable to high and low temperatures' indicator

Description

Runs the full methodology to analyse the impact of high and low temperatures on mortality using a quasi-Poisson time series approach with a distributed lag non-linear model. This function generates the relative risk of the temperature-mortality association as well as attributable numbers, rates and fractions of mortalities to specified temperature thresholds for high and low temperatures. Model validation statistics are also provided.

Usage

```
temp_mortality_do_analysis(
  data_path,
  date_col,
  region_col,
  temperature_col,
  dependent_col,
  population_col,
  country = "National",
  independent_cols = NULL,
  control_cols = NULL,
  var_fun = "bs",
  var_degree = 2,
  var_per = c(10, 75, 90),
  lagn = 21,
```

```

    lagnk = 3,
    dfseas = 8,
    meta_analysis = FALSE,
    attr_thr_high = 97.5,
    attr_thr_low = 2.5,
    save_fig = FALSE,
    save_csv = FALSE,
    output_folder_path = NULL,
    seed = NULL
)

```

Arguments

data_path	Path to a csv file containing a daily time series of data for a particular health outcome and climate variables, which may be disaggregated by geography.
date_col	Character. Name of the column in the dataframe containing the date.
region_col	Character. Name of the column in the dataframe that contains the geography name(s).
temperature_col	Character. Name of the column in the dataframe that contains the temperature column.
dependent_col	Character. Name of the column in the dataframe containing the dependent health outcome variable e.g. deaths.
population_col	Character. Name of the column in the dataframe that contains the population estimate per geography.
country	Character. Name of country for national-level estimates. Defaults to 'National'.
independent_cols	List. Additional independent variables to test in model validation as confounders. Defaults to NULL.
control_cols	List. Confounders to include in the final model adjustment. Defaults to NULL.
var_fun	Character. Exposure function for argvar (see <code>dlnm::crossbasis</code>). Defaults to 'bs'.
var_degree	Integer. Degree of the piecewise polynomial for argvar (see <code>dlnm::crossbasis</code>). Defaults to 2 (quadratic).
var_per	Vector. Internal knot positions for argvar (see <code>dlnm::crossbasis</code>). Defaults to <code>c(10, 75, 90)</code> .
lagn	Integer. Number of days in the lag period. Defaults to 21. (see <code>dlnm::crossbasis</code>).
lagnk	Integer. Number of knots in lag function. Defaults to 3. (see <code>dlnm::logknots</code>).
dfseas	Integer. Degrees of freedom for seasonality. Defaults to 8.
meta_analysis	Boolean. Whether to perform a meta-analysis. Defaults to FALSE.
attr_thr_high	Integer. Percentile at which to define the high temperature threshold for calculating attributable risk. Defaults to 97.5.
attr_thr_low	Integer. Percentile at which to define the low temperature threshold for calculating attributable risk. Defaults to 2.5.
save_fig	Boolean. Whether to save the plot as an output. Defaults to FALSE.

save_csv	Boolean. Whether to save the results as a CSV. Defaults to FALSE.
output_folder_path	Path to folder where plots and/or CSV should be saved. Defaults to NULL.
seed	Optional integer random seed used when sampling residuals for model validation plots. Defaults to NULL.

Details

This analysis requires a daily time series of temperature and death counts with population values as a minimum. This is then processed using a quasi-Poisson time series regression analysis with a distributed lag non-linear model and optional meta-analysis. Meta-analysis is recommended if the input data is disaggregated by area.

The model parameters have default values, which are recommended to keep as based on existing studies. However, if desired these can be adjusted for if appropriate for the user's context.

Model validation testing is provided as a standard output from the pipeline so a user can assess the quality of the model. If a user has additional independent variables these can be specified as `independent_cols` and assessed within different model combinations in the outputs of this testing. These can be added in the final model via `control_cols`. Note, a user should include variables if contextually relevant, and not simply based on model optimisation.

For attributable deaths the default is to use a high temperature threshold, defined as the 97.5th percentile of the temperature distribution over the full time period for each geography. The low temperature thresholds is similarly defined at the 2.5th percentile. These can be adjusted if desired, following review of the relative risk association between temperature and mortality using `attr_thr_high` or `attr_thr_low`.

Further details on the input data requirements, methodology, quality information and guidance on interpreting outputs can be found in the accompanying published [doi:10.5281/zenodo.14865904](https://doi.org/10.5281/zenodo.14865904).

Value

- `qaic_results` Dataframe. QAIC and dispersion metrics for each model combination and geography.
- `qaic_summary` Dataframe. Mean QAIC and dispersion metrics for each model combination.
- `vif_results` Dataframe. Variance inflation factors for each independent variables by geography.
- `vif_summary` Dataframe. Mean variance inflation factors for each independent variable.
- `adf_results` Dataframe. ADF test results for each geography.
- `power_list` List. Power information by area.
- `rr_results` Dataframe containing cumulative relative risk and confidence intervals from analysis.
- `res_attr_tot` Dataframe. Total attributable fractions, numbers and rates for each area over the whole time series.
- `attr_yr_list` List. Dataframes containing yearly estimates of attributable fractions, numbers and rates by area.
- `attr_mth_list` List. Dataframes containing total attributable fractions, numbers and rates by calendar month and area.

References

1. Watkins E, Hunt C, Lewis B, Ingole V, Glickman M. Standards for Official Statistics on Climate-Health Interactions (SOSCHI): Mortality attributed to high and low temperatures: methodology. Zenodo; 2026. Available from: [doi:10.5281/zenodo.14865904](https://doi.org/10.5281/zenodo.14865904)
2. Gasparri A, Guo Y, Hashizume M, Lavigne E, Zanobetti A, Schwartz J, et al. Mortality risk attributable to high and low ambient temperature: a multicountry observational study. *Lancet*. 2015 Jul;386(9991):369-75. Available from: <https://linkinghub.elsevier.com/retrieve/pii/S0140673614621140>
3. Gasparri A, Armstrong B. Reducing and meta-analysing estimates from distributed lag non-linear models. *BMC Medical Research Methodology*. 2013 Jan 9;13:1. Available from: [doi:10.1186/14712288131](https://doi.org/10.1186/14712288131)
4. Gasparri A, Armstrong B, Kenward MG. Multivariate meta-analysis for non-linear and other multi-parameter associations. *Statistics in Medicine*. 2012 Dec 20;31(29):3821-39. Available from: [doi:10.1002/sim.5471](https://doi.org/10.1002/sim.5471)

Examples

```
example_data <- data.frame(
  date = seq.Date(as.Date("2020-01-01"), by = "day", length.out = 365),
  region = "Example Region",
  tmean = stats::runif(365, -2, 32),
  deaths = stats::rpois(365, lambda = 8),
  pop = 500000
)
example_path <- tempfile(fileext = ".csv")
utils::write.csv(example_data, example_path, row.names = FALSE)

temp_mortality_do_analysis(
  data_path = example_path,
  date_col = "date",
  temperature_col = "tmean",
  dependent_col = "deaths",
  population_col = "pop",
  region_col = "region",
  country = "Example Region",
  meta_analysis = FALSE,
  independent_cols = NULL,
  control_cols = NULL,
  var_fun = "bs",
  var_degree = 2,
  var_per = c(10, 75, 90),
  lagn = 7,
  lagnk = 2,
  dfseas = 4,
  attr_thr_high = 97.5,
  attr_thr_low = 2.5,
  save_fig = FALSE,
  save_csv = FALSE,
  output_folder_path = tempdir()
)
```

wildfire_do_analysis *This is full analysis pipeline to analyse the impact of wildfire-related PM2.5 on a health outcome.*

Description

Runs full analysis pipeline for analysis of the impact of wildfire-related PM2.5 on a health outcome using time stratified case-crossover approach with conditional quasi-Poisson regression model. This function generates relative risk of the mortality associated to wildfire-related PM2.5 as well as attributable numbers, rates and fractions of health outcome. Model validation statistics are also provided.

Usage

```
wildfire_do_analysis(  
  health_path,  
  join_wildfire_data = FALSE,  
  ncdf_path = NULL,  
  shp_path = NULL,  
  date_col,  
  region_col,  
  shape_region_col = NULL,  
  mean_temperature_col,  
  health_outcome_col,  
  population_col = NULL,  
  rh_col = NULL,  
  wind_speed_col = NULL,  
  pm_2_5_col = NULL,  
  wildfire_lag = 3,  
  temperature_lag = 1,  
  spline_temperature_lag = 0,  
  spline_temperature_degrees_freedom = 6,  
  predictors_vif = NULL,  
  calc_relative_risk_by_region = FALSE,  
  scale_factor_wildfire_pm = 10,  
  save_fig = FALSE,  
  save_csv = FALSE,  
  output_folder_path = NULL,  
  create_run_subdir = FALSE,  
  print_vif = FALSE,  
  print_model_summaries = FALSE  
)
```

Arguments

health_path	Path to a CSV file containing a daily time series of data for a particular health outcome, which may be disaggregated by region. If this does not include a column with wildfire-related PM2.5, use join_wildfire_data = TRUE to join these data.
join_wildfire_data	Boolean. If TRUE, a daily time series of wildfire-related PM2.5 concentration is joined to the health data. If FALSE, the data set is loaded without any additional joins. Defaults to FALSE.
ncdf_path	Path to a NetCDF file containing a daily time series of gridded wildfire-related PM2.5 concentration data.
shp_path	Path to a shapefile .shp of the geographical boundaries for which to extract mean values of wildfire-related PM2.5
date_col	Character. Name of the column in the dataframe that contains the date.
region_col	Character. Name of the column in the dataframe that contains the region names.
shape_region_col	Character. Name of the column in the shapefile dataframe that contains the region names.
mean_temperature_col	Character. Name of the column in the dataframe that contains the mean temperature column.
health_outcome_col	Character. Name of the column in the dataframe that contains the health outcome count column (e.g. number of deaths, hospital admissions)
population_col	Character. Name of the column in the dataframe that contains the population data. Defaults to NULL. This is only required when requesting region-level AF/AN outputs and no pop column is already present in the input data.
rh_col	Character. Name of the column containing relative humidity values. Defaults to NULL.
wind_speed_col	Character. Name of the column containing wind speed. Defaults to NULL.
pm_2_5_col	Character. The name of the column containing PM2.5 values in micrograms. This is only required if health data isn't joined. Defaults to NULL.
wildfire_lag	Integer. The number of days for which to calculate the lags for wildfire PM2.5. Default is 3.
temperature_lag	Integer. The number of days for which to calculate the lags for temperature. Default is 1.
spline_temperature_lag	Integer. The number of days of lag in the temperature variable from which to generate splines. Default is 0 (unlagged temperature variable).
spline_temperature_degrees_freedom	Integer. Degrees of freedom for the spline(s).
predictors_vif	Character vector with each of the predictors to include in the model. Must contain at least 2 variables. Defaults to NULL.

<code>calc_relative_risk_by_region</code>	Bool. Whether to calculate Relative Risk by region. Default: FALSE
<code>scale_factor_wildfire_pm</code>	Numeric. The value to divide the wildfire PM2.5 concentration variables by for alternative interpretation of outputs. Corresponds to the unit increase in wildfire PM2.5 to give the model estimates and relative risks (e.g. <code>scale_factor = 10</code> corresponds to estimates and relative risks representing impacts of a 10 unit increase in wildfire PM2.5). Setting this parameter to 0 or 1 leaves the variable unscaled.
<code>save_fig</code>	Boolean. Whether to save the plot as an output.
<code>save_csv</code>	Boolean. Whether to save the results as a CSV
<code>output_folder_path</code>	Path. Path to folder where plots and/or CSV should be saved.
<code>create_run_subdir</code>	Boolean. If TRUE, create a timestamped subdirectory under <code>output_folder_path</code> for this run's outputs. Defaults to FALSE.
<code>print_vif</code>	Bool, whether or not to print VIF (variance inflation factor) for each predictor. Defaults to FALSE.
<code>print_model_summaries</code>	Bool. Whether to print the model summaries to console. Defaults to FALSE.

Details

This analysis pipeline requires a daily time series with mean wildfire PM2.5, mean temperature and health outcome (all-cause mortality, respiratory, cardiovascular, hospital admissions etc) with population values as a minimum. This is then processed using a time stratified case crossover approach with conditional Poisson case-crossover analysis and optional meta-analysis. Meta-analysis is recommended if the input data is disaggregated by area.

The model parameters have default values, which are recommended to keep as based on existing studies. However, if desired these can be adjusted for sensitivity analysis.

Model validation testing is provided as a standard output from the pipeline so a user can assess the quality of the model. Additionally, users can incorporate extra independent variables-such as relative humidity or wind speed-directly into the model for enhanced analysis.

Further details on the input data requirements, methodology, quality information and guidance on interpreting outputs can be found in the accompanying published [doi:10.5281/zenodo.14052184](https://doi.org/10.5281/zenodo.14052184).

Value

- `rr_results` A dataframe with relative risk estimates and confidence intervals for each region.
- `rr_pm` A dataframe of relative risk estimates for wildfire-specific PM2.5 exposure across regions as PM values changes.
- `af_an_results` A dataframe containing attributable fractions, attributable numbers and deaths per 100k population for each region
- `annual_af_an_results` A dataframe containing annual attributable numbers and fractions for each region

- `calculate_qaic` A dataframe of QAIC and dispersion metrics for each model combination and geography.
- `check_wildfire_vif` A dataframe containing Variance inflation factors for each independent variables by region.

References

1. Brown A, Soutter E, Ingole V., Standards for Official Statistics on Climate-Health Interactions (SOSCHI): Wildfires: introduction. Zenodo; 2024. Available from: <https://zenodo.org/records/14052184>
2. Hänninen R, Sofiev M, Uppstu A, Kouznetsov R. Daily surface concentration of fire related PM2.5 for 2003-2023, modelled by SILAM CTM when using the MODIS satellite data for the fire radiative power. Finnish Meteorological Institute; 2024. Available from: [doi:10.57707/fmib2share.d1cac971b3224d438d5304e945e9f16c](https://doi.org/10.57707/fmib2share.d1cac971b3224d438d5304e945e9f16c)
3. GADM. Database for Global Administrative Areas. Available from: https://gadm.org/download_country.html
4. Tobias A, Kim Y, Madaniyazi L. Time-stratified case-crossover studies for aggregated data in environmental epidemiology: a tutorial. *Int J Epidemiol.* 2024;53(2). Available from: [doi:10.1093/ije/dyae020](https://doi.org/10.1093/ije/dyae020)
5. Wu Y, Li S, Guo Y. Space-Time-Stratified Case-Crossover Design in Environmental Epidemiology Study. *Heal Data Sci.* 2021; Available from: [doi:10.34133/2021/9870798](https://doi.org/10.34133/2021/9870798)

Examples

```
example_data <- data.frame(
  date = seq.Date(as.Date("2020-01-01"), by = "day", length.out = 180),
  region = "Example Region",
  death = stats::rpois(180, lambda = 4),
  population = 400000,
  tmean = stats::runif(180, 10, 35),
  mean_PM = stats::runif(180, 0, 25)
)
example_path <- tempfile(fileext = ".csv")
utils::write.csv(example_data, example_path, row.names = FALSE)

wildfire_do_analysis(
  health_path = example_path,
  join_wildfire_data = FALSE,
  ncdf_path = NULL,
  shp_path = NULL,
  date_col = "date",
  region_col = "region",
  shape_region_col = NULL,
  mean_temperature_col = "tmean",
  health_outcome_col = "death",
  population_col = "population",
  rh_col = NULL,
  wind_speed_col = NULL,
  pm_2_5_col = " mean_PM ",
  wildfire_lag = 3,
```

```
temperature_lag = 1,  
spline_temperature_lag = 0,  
spline_temperature_degrees_freedom = 4,  
predictors_vif = NULL,  
calc_relative_risk_by_region = FALSE,  
scale_factor_wildfire_pm = 10,  
save_fig = FALSE,  
save_csv = FALSE,  
output_folder_path = tempdir(),  
create_run_subdir = FALSE,  
print_vif = FALSE,  
print_model_summaries = FALSE)
```

Index

[abort_climate](#), [4](#)
[abort_column_not_found](#), [5](#)
[abort_model_error](#), [6](#)
[abort_validation](#), [7](#)
[air_pollution_do_analysis](#), [8](#)

[climatehealth \(climatehealth-package\)](#), [3](#)
[climatehealth-package](#), [3](#)

[diarrhea_do_analysis](#), [12](#)

[install_INLA](#), [15](#)
[install_terra](#), [16](#)
[is_climate_error](#), [16](#)

[malaria_do_analysis](#), [17](#)

[run_descriptive_stats](#), [20](#)
[run_descriptive_stats_api](#), [22](#)

[suicides_heat_do_analysis](#), [24](#)

[temp_mortality_do_analysis](#), [28](#)

[wildfire_do_analysis](#), [32](#)