# Package ‘aermod’

September 22, 2021

**Title** Work with AERMOD POST files

**Version** 0.0.0.9000

**Description** Convert Fortran Unformatted I/O format POST files to/from R matrix. Convert 1-hr SO2 NAAQS Design value for linear combination of matrices.

**License** MIT + file LICENSE

**Encoding** UTF-8

**LazyData** true

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.1.2

**Suggests** testthat (>= 3.0.0)

**Config/testthat/edition** 3

**Imports** magrittr,
dplyr,
purrr,
stringr,
tidyselect,
tidyr,
tibble,
lubridate

## R topics documented:

## blend *Blend two impact matrices together.*

### Description

The rows from `src` replace corresponding rows in `target`

### Usage

```
blend(src, indices, target)
```

### Arguments

`src` Matrix corresponding to source group for which intermittent operation is simulated.

`indices` Hours of operation (integer vector). Typically generated with `hrs_rand`

`target` Matrix representing source group that does *not* operate when intermittent source does. By default this is the zero matrix (i.e., intermittent source may operate at same time as all other sources).

### Value

Blended matrix. Has an attribute listing all of the hours for which a substitution was made. Existing attributes are carried over from `target`.

## get_dv *Calculate SO2 Design value*

### Description

Calculate the 2010 SO2 NAAQS design value for an impact matrix. The impact matrix should have rows corresponding to the number of hours in a five-year meteorological database (e.g., 43848 hours for 2016–2020). The number of receptors (columns) can be arbitrary.

### Usage

```
get_dv(mat, diag = FALSE)
```

### Arguments

`mat` An impact matrix, typically a linear combination of impact matrices, which may be the result of a Monte Carlo simulation.

`diag` Do not report full calculation. Instead, output a tibble giving the top four daily highs at each receptor x year.

## Details

Per EPA's 2014 guidance (pp. A-24–A-25), the calculation is carried out as follows:

l. At each receptor, for each hour of the modeled period, calculate a total concentration across all sources including backgro1md concentrations if applicable. This can be done in AERMOD using SRCGROUP ALL or by adding individual source groups outside of AERMOD, using hourly POSTFILEs. If the user is totaling the concentrations outside of AERMOD, the source groups used in the calculations need to be mutually exclusive, i.e. no one source should be in multiple source groups.

1. From the total concentrations calculated in step 1, obtain the 1-hr maximum concentration at each receptor for each modeled day.
2. From the output of step 2, for each year modeled, calculate the 99th percentile (4th highest) daily maximum 1-hour concentration at each receptor. If modeling 5 years of meteorological data, this results in five 99th percentile concentrations at each receptor.
3. Average the 99th percentile (or 4th highest) concentrations across the modeled years to obtain a design value at each receptor.
4. Modeled source contributions to a NAAQS violation can be determined by analyzing the hourly concentrations from the individual source groups POSTFILES corresponding to the same hour as the 4th daily maximum 1-hour concentration from each year. See 75 FR at 35540.

## Value

A double corresponding to the calculated design value.

---

# hrs_rand *Generate random hours of operation*

---

## Description

Generate random hours of operation over the meteorological database. Hours within each meteorological year are randomly sampled (without replacement). Random sampling can consider blocks of consecutive hours, or individual hours (block length of 1).

## Usage

```
hrs_rand(
  yrspan = 2016L:2020L,
  nblock = 6L,
  blocklen = 1L,
  which_filter = 1L,
  units = c("hours", "days"),
  emis_scale = 1
)
```

### Arguments

`yrspan` Calendar years for which operating hours are to be generated. A numeric vector

`nblock` Number of blocks of consecutive hours.

`blocklen` Number of hours in each block. Default is 1.

`which_filter` Integer specifying special filters on the random number generation. Default is 1 (no filter). 2 is case where nblock = 1 and blocks in successive years must be spaced at least 365 days apart.

`units` for `blocklen`, either `"hours"` (default) or `"days"`. Return value is always vector of hours.

`emis_scale` , for binned randomized emission rates, a vector of emission scalars of length `blocklen`. Default is `1` (no binning).

### Value

An integer vector of operating hours, which may be used to index rows in an impact matrix.

## make_simplan *Generate a list of random operating hours, along with the PRNG seeds needed to recreate them.*

### Description

Generate a list of random operating hours, along with the PRNG seeds needed to recreate them.

### Usage

```
make_simplan(N_sim = 5, ...)
```

### Arguments

`N_sim` The number of sets of operating hours to generate.

`...` Input parameters for call to `hrs_rand`.

### Value

a nested list of N_sim elements, each of which is a list with elements `seed` and `hrs`

## `scan_postfile` *Scan POST file*

### Description

Reads Fortran unformatted I/O POSTFILE and conver to matrix. Hours are not written individually, nor are source groups as they are saved as attributes of the matrix.

### Usage

```
scan_postfile(f, ...)
```

### Arguments

| | |
|---|---|
| `f` | path to an unformatted Fortran I/O binary POST file generated by AERMOD. |
| `...` | arguments passed thru to `readBin`. Primary use is to specify endianness in case POST file was generated on a machine with different endianness from your own. |

### Value

matrix

## `serialize_post` *Serialize impact matrix*

### Description

Serialize impact matrix to AERMOD unformatted binary POST file.

### Usage

```
serialize_post(mat, fname, srcgrp)
```

### Arguments

| | |
|---|---|
| `mat` | matrix |
| `fname` | character |
| `srcgrp` | character |

### Value

logical

# Index