Package: airPb (via r-universe)

August 20, 2024

Title Calculate Estimates of Airborne Lead Exposure at Point Locations in the Cincinnati, OH Area

Version 0.9

Maintainer Erika Rasnick <erika.rasnick@cchmc.org>

Description Easily and reproducibly assess exposure to airborne lead at specific locations in and around Cincinnati, Ohio. The package calculates predictions of air lead exposure from a land use random forest model developed by Dr. Cole Brokamp based on ambient air sampling in Cincinnati, OH between 2001 and 2005. Additionally scale estimates to account for temporal variation in air lead in the region.

License GPL-3

Encoding UTF-8

LazyData true

URL https://github.com/geomarker-io/airPb

BugReports https://github.com/geomarker-io/airPb/issues

Depends R (>= 2.10)

Imports randomForest (>= 4.6-14), dplyr (>= 0.8.3), purrr (>= 0.3.3), sf (>= 0.8-0), tidyr (>= 1.0.0), raster (>= 3.0-7), sp (>= 1.3-1), stats

Roxygen list(markdown = TRUE)

RoxygenNote 6.1.1

Suggests knitr, rmarkdown, lubridate (>= 1.7.4), tibble (>= 3.0.3)

VignetteBuilder knitr

Repository https://geomarker-io.r-universe.dev

RemoteUrl https://github.com/geomarker-io/airPb

RemoteRef HEAD

RemoteSha 369adac4344515a173d0f102981f55210ee6d3e5

Contents

`add_scaled_airPb` *Calculate temporally scaled air lead exposure estimates at specific locations.*

Description

`calculate_scaled_airPb()` is a wrapper function that estimates airborne lead exposures at provided locations by calling `calculate_airPb()`, then temporally scales those estimates using scaling factors computed by calling `calculate_scaling_factors()`. This function is particularly useful for calculating exposures at the same locations on different dates.

Usage

```
add_scaled_airPb(locations)
```

Arguments

`locations`	Data.frame with columns 'id', 'lat','lon', 'start_date', and 'end_date' at minimum.

Value

A numeric vector of air lead estimates (ug/m3).

References

Cole Brokamp, Roman Jandarov, MB Rao, Grace LeMasters, Patrick Ryan. Exposure assessment models for elemental components of particulate matter in an urban environment: A comparison of regression and random forest approaches. Atmospheric Environment. 151. 1-11. 2017. http://dx.doi.org/10.1016/j.atmosenv.2016.11.066

Examples

```
my_data <- data.frame(id = rep(1,3),
    lat = c(39.19674, 39.19674, 39.19674),
    lon = c(-84.58260, -84.58260, -84.58260),
    start_date = c(as.Date("2010-01-08"), as.Date("2012-06-08"), as.Date("2015-04-09")),
    end_date = c(as.Date("2010-02-08"), as.Date("2012-07-08"), as.Date("2015-05-09")))

airPb_scaled <- add_scaled_airPb(my_data)
```

calculate_airPb — *Calculate air lead exposure estimates at specific locations.*

Description

`calculate_airPb()` uses a land use random forest model developed by Dr. Cole Brokamp based on ambient air sampling in Cincinnati, OH between 2001 and 2005 to estimate exposure to airborne lead at point locations in the area specified by latitude and longitude. The model predictors include greenspace (NDVI) within 1000 meters, population density within 500 meters, length of bus routes within 900 meters, percent pasure within 800 meters, percent developed open land within 1100 meters, percent developed medium land within 400 meters, percent developed low land within 900 meters, and percent developed high land within 1500 meters.

Usage

```
calculate_airPb(locations, return.LU.vars = FALSE)
```

Arguments

`locations`	Data.frame with columns 'id', 'lat', and 'lon' at minimum.
`return.LU.vars`	When `return.LU.vars = TRUE`, the land use predictors used to generate the air lead values are also returned.

Value

If `return.LU.vars = FALSE`, a numeric vector of air lead estimates (ug/m3) ? is returned. If `return.LU.vars = TRUE`, the `locations` data.frame with additional columns for air lead values and the land use predictors used to generate the air lead values is returned.

References

Cole Brokamp, Roman Jandarov, MB Rao, Grace LeMasters, Patrick Ryan. Exposure assessment models for elemental components of particulate matter in an urban environment: A comparison of regression and random forest approaches. Atmospheric Environment. 151. 1-11. 2017. http://dx.doi.org/10.1016/j.atmosenv.2016.11.066

Examples

```
my_data <- data.frame(id = 1:3,
    lat = c(39.19674, 39.12731, 39.28765),
    lon = c(-84.58260, -84.52700, -84.51017))

lead_est <- calculate_airPb(my_data, return.LU.vars = FALSE)
lead_est <- calculate_airPb(my_data, return.LU.vars = TRUE)
```

`calculate_scaling_factors`
Calculate temporal scaling factors based on EPA measurements of airborne lead.

Description

`calculate_scaling_factors()` constructs temporal scaling factors based on measurements of airborne lead recorded by the EPA in the Cincinnati area. These scaling factors are the average lead measured over the time period specified by `start_date` and `end_date`, divided by the average lead recorded over the ambient air sampling period (2001 to 2005). Scaling factors can be multiplied by air lead estimates from `calculate_airPb()` to adjust for temporal variability in airborne lead in the Cincinnati area over time.

Usage

```
calculate_scaling_factors(dates)
```

Arguments

`dates`	A data.frame with 2 columns called 'start_date' and 'end_date' at minimum. Both columns must be of class `Date`. See `as.Date` for help converting a character vector to a Date vector.

Details

EPA data in this package is available from November 9, 2001 through November 28, 2018. Scaling factors that attempt to average over air lead measured on dates outside this range will not be calculated. In addition, it is important to be mindful of the frequency of air lead measurements recorded by the EPA. Note that air lead was measured every 6 days through the end of 2010, and every 3 days starting in 2011. If there are less than 4 measurements of air lead between the start_date and end_date, the scaling factor will not be calculated and NA will be returned.

Value

A numeric vector of temporal scaling factors.

Examples

```
my_dates <- data.frame(start_date = c("2010-01-08", "2012-06-08", "2010-01-09",
                                      "2015-04-09", "2010-01-10"),
                       end_date = c("2010-02-08", "2012-07-08", "2010-02-09",
                                    "2015-05-09", "2010-02-10"))

## Not run:
class(my_dates$start_date)  # character vector
scaling1m <- calculate_scaling_factors(my_dates)
```

```
## End(Not run)

my_dates$start_date <- as.Date(my_dates$start_date)
my_dates$end_date <- as.Date(my_dates$end_date)
scaling1m <- calculate_scaling_factors(my_dates)
```

Index