

# Package ‘EnvExpInd’

October 23, 2020

**Type** Package

**Title** Environmental Exposure on the Individual Level

**Imports** gstat,RCurl,dplyr,stringi,sp,maptools,zoo

**Version** 0.1.0

**Depends** R(>= 3.5.0)

**Description** Tools for the assessment of the environmental exposure. The package provides three methods (nearest monitoring site, inverse distance weighted as described in Li Wu (2017) <doi:10.1016/j.envint.2016.11.013>,and ordinary kriging) to calculate the environmental exposure (e.g. air pollution) on the individual level.

**URL** <https://github.com/Spatial-R/EnvExpInd>

**License** GPL-3

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.1.0

**Suggests** knitr, rmarkdown

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** Bing Zhang [aut, cre]

**Maintainer** Bing Zhang <zhangb269@mail2.sysu.edu.cn>

**Repository** CRAN

**Date/Publication** 2020-10-23 15:50:02 UTC

## R topics documented:

exposure_estimate_idw . . . . .	2
exposure_estimate_krige . . . . .	4
expoure_estimate_simple . . . . .	6
get_latlon_china . . . . .	7
get_refrence_id_simple . . . . .	8
individual_data . . . . .	10

pollutant_data . . . . .	10
site_data . . . . .	11
timeseries_imput . . . . .	11

<b>Index</b>	<b>13</b>
--------------	-----------

---

exposure\_estimate\_idw *Estimate the pollutant exposure using the inverse distance weighting method*

---

### Description

Used the pollutant concentration in the individual location as the reference point to estimate the environmental exposure. The pollutant concentration at the reference point was calculated based on the inverse distance weighting method.

### Usage

```
exposure_estimate_idw(
  individual_data,
  individual_id,
  exposure_date,
  individual_lat,
  individual_lon,
  pollutant_data,
  pollutant_date = "date",
  pollutant_site_lat,
  pollutant_site_lon,
  pollutant_name = c("pm10", "so2"),
  estimate_interval = c(0:30)
)
```

### Arguments

individual_data	data.frame, contains the reference id, individual_id and exposure_date
individual_id	character, varibale name in individual_data, represents the unique id for each individual
exposure_date	character, varibale name in individual_data, which represents the start date to estimate the environment exposure
individual_lat	character, varibale name in individual_data, represents the latitude information of each individual
individual_lon	character, varibale name in individual_data, represents the longitude information of each individual
pollutant_data	data.frame, contains the pollutant and site informatin. One column represents the site information and other columns represent the concentration of pollutants

**pollutant\_date** character, variable name represents the date information for the air pollutant dataset  
**pollutant\_site\_lat** character, variable name in pollutant\_data, includes the latitude information of each monitoring site  
**pollutant\_site\_lon** character, variable name in pollutant\_data, includes the longitude information of each monitoring site  
**pollutant\_name** vector, pollutant name in the pollutant\_data, which represent the name of the target pollutants to be estimated  
**estimate\_interval** continue numeric vector, the estimation period, for example: 0:30, for each individual we estimate the environment exposure ranging from the exposure\_date to exposure\_date + 30 days

### Value

A list. For each element in the list, there is a dataframe with the first column representing the individual id, the remaining columns represent the exposure estimation in different time points.

### Author(s)

Bing Zhang, <https://github.com/Spatial-R/EnvExpInd>

### Examples

```

library(EnvExpInd)
individual_data$date <- as.Date(individual_data$date)
pollutant_data$date <- as.Date(pollutant_data$date)
pollutant_data_full <- timeseries_imput(data= pollutant_data, date_var = "date",
site_var = "site.name", imput_col = 3:8)
pollutant_data_tem <- merge(pollutant_data_full, site_data, by.x = "site.name", by.y = "site")
exposure_estimate_idw(
  individual_data = individual_data,
  individual_id = "id",
  exposure_date = "date",
  individual_lat = "lat",
  individual_lon = "lon",
  pollutant_data = pollutant_data_tem,
  pollutant_date = "date",
  pollutant_site_lat = "lat",
  pollutant_site_lon = "lon",
  pollutant_name = c("PM10", "PM2.5"),
  estimate_interval = c(0:10))
  
```

---

 exposure\_estimate\_krige

*Assess the environmental exposure using the kriging method*


---

### Description

Based on the kriging method, the pollutant exposure in each individual location was estimated and then assess the total pollutant exposure through the estimate\_interval

### Usage

```
exposure_estimate_krige(
  individual_data,
  individual_id,
  exposure_date,
  individual_lat,
  individual_lon,
  pollutant_data,
  pollutant_date = "date",
  pollutant_site_lat,
  pollutant_site_lon,
  pollutant_name = c("pm10", "so2"),
  estimate_interval = c(0:30),
  kriging_model,
  nmax = 7,
  kriging_method = "med"
)
```

### Arguments

individual_data	data.frame, contains the reference id, individual_id and exposure_date
individual_id	character, variable name in individual_data, represents the unique id for each individual
exposure_date	character, variable name in individual_data, which represents the start date to estimate the environment exposure
individual_lat	character, variable name in individual_data, represents the latitude information of each individual
individual_lon	character, variable name in individual_data, represents the longitude information of each individual
pollutant_data	data.frame, contains the pollutant and site information. One column represents the site information and other columns represent the concentration of pollutants
pollutant_date	character, variable name represents the date information for the air pollutant dataset

pollutant\_site\_lat  
                   character, varibale name in pollutant\_data, includes the latitude information of  
                   each monitoring site  
 pollutant\_site\_lon  
                   character, varibale name in pollutant\_data, includes the longitude information  
                   of each monitoring site  
 pollutant\_name  vector, pollutant name in the pollutant\_data need to be estimated  
 estimate\_interval  
                   continue numeric vector, the estimation period, for example: 0:30, for each in-  
                   dividual we estimate the environment exposure ranging from the exposure\_date  
                   to exposure\_date + 30 days  
 krige\_model     ?krige  
 nmax            ?krige  
 krige\_method    ?krige

### Value

A list. For each element in the list, there is a dataframe with the first column representing the individual id, the remaining columns represent the exposure estimation in different time points.

### Author(s)

Bing Zhang, <https://github.com/Spatial-R/EnvExpInd>

### Examples

```

## Not run:
library(EnvExpInd)
library(maptools)
library(gstat)
individual_data$date <- as.Date(individual_data$date)
pollutant_data$date <- as.Date(pollutant_data$date)
pollutant_data_full <- timeseries_imput(data= pollutant_data,date_var = "date",
site_var = "site.name",imput_col = 3:8)
pollutant_data_tem <- merge(pollutant_data_full,site_data,by.x = "site.name",by.y = "site")
test.pollutant <- pollutant_data_tem[pollutant_data_tem$date == "2014-09-20",]
coordinates(test.pollutant) = ~lat + lon
##### please define the variogram in a right way #####
m <- fit.variogram(variogram(PM10~1, test.pollutant), vgm(1, "Sph", 200, 1))
exposure_estimate_krige(
  individual_data = individual_data,
  individual_id = "id",
  exposure_date = "date",
  individual_lat = "lat",
  individual_lon = "lon",
  pollutant_data = pollutant_data_tem,
  pollutant_date = "date",
  pollutant_site_lat = "lat",
  pollutant_site_lon = "lon",
  pollutant_name = c("PM10", "PM2.5"),

```

```

krige_model = m,
nmax = 7,
krige_method = "med",
estimate_interval = c(0:10))

## End(Not run)

```

---

exposure\_estimate\_simple

*Assess the environmental exposure using the simplest method: nearest monitoring site method*

---

### Description

Using the nearest surveillance site as the reference site to estimate the pollutant exposure.

### Usage

```

exposure_estimate_simple(
  individual_data,
  individual_id,
  reference_id,
  exposure_date,
  pollutant_data,
  pollutant_site = "site",
  pollutant_date = "date",
  pollutant_name = c("pm10", "so2"),
  estimate_interval
)

```

### Arguments

individual_data	data.frame, includes the reference id, individual_id and exposure_date
individual_id	character, variable name in the individual_data, which represents the unique id for each individual
reference_id	character, variable name in the individual_data, which represents the nearest surveillance site for each individual
exposure_date	character, variable name in the individual_data, which represents the start date to estimate the environment exposure
pollutant_data	data.frame, contains the pollutant and site information. One column represents the site information and other columns represent the concentration of pollutants
pollutant_site	character, variable name in the pollutant_data, which represents the monitoring site information
pollutant_date	character, variable name in the pollutant_data, which represents the surveillance date for pollutant concentration

`pollutant_name` vector, variable names in the `pollutant_data`, which represent the name of the target pollutants to be estimated

`estimate_interval`  
 continue numeric vector, the estimation period, for example: 0:30, for each individual we estimate the environment exposure ranging from the `exposure_date` to `exposure_date + 30 days`

### Value

A list. For each element in the list, there is a dataframe with the first column representing the individual id, the remaining columns represent the exposure estimation in different time points.

### Author(s)

Bing Zhang, <https://github.com/Spatial-R/EnvExpInd>

### Examples

```
library(EnvExpInd)
individual_data$date <- as.Date(individual_data$date)
pollutant_data$date <- as.Date(pollutant_data$date)
pollutant_data_full <- timeseries_imput(data= pollutant_data,
  date_var = "date",site_var = "site.name",imput_col = 3:8)
pollutant_data_tem <- merge(pollutant_data_full,site_data,by.x = "site.name",by.y = "site")
individual_data$reference_id <- get_reference_id_simple(
  individual_data = individual_data,
  individual_lat = "lat",
  individual_lon = "lon",
  individual_id = "id",
  site_data = site_data,
  site_lon = "lon",
  site_lat = "lat",
  site_id = "site")
exposure_estimate_simple(
  individual_data = individual_data,
  individual_id = "id",
  reference_id = "reference_id",
  exposure_date = "date",
  pollutant_data = pollutant_data_tem,
  pollutant_site = "site.name",
  pollutant_date = "date",
  pollutant_name = c("PM10","PM2.5"),
  estimate_interval = c(0:10))
```

**Description**

Based on the Baidumap api, get\_latlon\_china function covers the detailed address into the longitude and latitude

**Usage**

```
get_latlon_china(data, add_var = "address", api_key = "")
```

**Arguments**

data	data frame, contains the address information
add_var	character, variable name in the data, which represents the address information
api_key	character, baidumap api key, seeing: <a href="http://lbsyun.baidu.com/index.php?title=webapi/guide/webservice-geocoding">http://lbsyun.baidu.com/index.php?title=webapi/guide/webservice-geocoding</a>

**Value**

two columns (lon and lat) was added into the origin data.frame

**Author(s)**

Bing Zhang, <https://github.com/Spatial-R/EnvExpInd>

**Examples**

```
## Not run:
  get_latlon_china(wuhan.sem, add_var = "add", api_key = "sksksksksksk")

## End(Not run)
```

---

```
get_refrence_id_simple
```

*Match the nearing monitoring site for each individual*

---

**Description**

Match the nearing monitoring site for each individual

**Usage**

```
get_refrence_id_simple(
  individual_data,
  individual_lat,
  individual_lon,
  individual_id,
  site_data,
  site_lat,
```



```
    site_lon,  
    site_id  
  )
```

**Arguments**

<code>individual_data</code>	data.frame, including three variables ( <code>individual_lat</code> , <code>individual_lon</code> and <code>individual_id</code> )
<code>individual_lat</code>	character, varibale name in <code>individual_data</code> , includes the latitude information of each individual
<code>individual_lon</code>	character, varibale name in <code>individual_data</code> , includes the longtitude information of each individual
<code>individual_id</code>	character, varibale name in <code>individual_data</code> , includes the unique id for each individual
<code>site_data</code>	data.frame, including three variables ( <code>site_lat</code> , <code>site_lon</code> and <code>site_id</code> )
<code>site_lat</code>	character varibale includes the latitude value of the site
<code>site_lon</code>	character varibale includes the longtitude value of the site
<code>site_id</code>	character varibale includes the id for each site

**Value**

A vector, including the `refrence_id` for each individual

**Author(s)**

Bing Zhang, <https://github.com/Spatial-R/EnvExpInd>

**Examples**

```
get_refrence_id_simple(  
  individual_data = individual_data,  
  individual_lat = "lat",  
  individual_lon = "lon",  
  individual_id = "id",  
  site_data = site_data,  
  site_lon = "lon",  
  site_lat = "lat",  
  site_id = "site")
```

---

individual_data	<i>The detailed information for each individual.</i>
-----------------	--

---

**Description**

A dataset containing the detailed information for each individual

**Usage**

individual\_data

**Format**

A data frame with 21 rows and 3 variables:

**id** id number for each individual

**date** the monitoring time point

**lat** the latitude for each individual

**lon** the longitude for each individual ...

---

pollutant_data	<i>The concentration of air pollutant at each time point.</i>
----------------	---

---

**Description**

A dataset containing the concentration of air pollutant at each time point

**Usage**

pollutant\_data

**Format**

A data frame with 11090 rows and 8 variables:

**date** the monitoring time point

**site.name** the names of the monitoring site

**SO2** the concentration of SO2

**NO2** the concentration of NO2

**PM10** the concentration of PM10

**CO** the concentration of CO

**O3** the concentration of O3

**PM2.5** the concentration of PM2.5 ...

---

site_data	<i>Monitoring sites.</i>
-----------	--------------------------

---

**Description**

A dataset containing the information of the monitoring sites

**Usage**

```
site_data
```

**Format**

A data frame with 10 rows and 2 variables:

**site** the name of monitoring sites

**lat** the latitude for each monitoring site

**lon** the longitude for each monitoring site ...

---

timeseries_imput	<i>Impute the missing value for the timeseries using the linear interpolation</i>
------------------	---

---

**Description**

Complete the time series using the linear interpolation

**Usage**

```
timeseries_imput(data, date_var, site_var, imput_col)
```

**Arguments**

**data** data.frame, contains the refrence id, individual\_id and exposure\_date

**date\_var** character, varibale name in data, represents the monitoring date.

**site\_var** character, varibale name in data, represents the name of monitoring site.

**imput\_col** numeric,the column position of the target variables need to be imputed

**Value**

a data.frame

**Author(s)**

Bing Zhang, <https://github.com/Spatial-R/EnvExpInd>

**Examples**

```
library(EnvExpInd)
pollutant_data_com <- timeseries_imput(data= pollutant_data, date_var = "date",
                                       site_var = "site.name", imput_col = 3:8)
```

# Index

## \* datasets

individual\_data, [10](#)

pollutant\_data, [10](#)

site\_data, [11](#)

exposure\_estimate\_idw, [2](#)

exposure\_estimate\_krige, [4](#)

exposure\_estimate\_simple, [6](#)

get\_latlon\_china, [7](#)

get\_reference\_id\_simple, [8](#)

individual\_data, [10](#)

pollutant\_data, [10](#)

site\_data, [11](#)

timeseries\_imput, [11](#)