Package 'AirExposure'

May 6, 2026

Type Package

Title Exposure Model to Air Pollutants Based on Mobility and Daily Activities

Version 1.0

Date 2024-07-26

Author Josefina Urquiza [aut, cre],
Maria Florencia Tames [aut],
Maela Lupo [aut],
Alfredo Rigalli [aut]

Maintainer Josefina Urquiza <jurquiza@conicet.gov.ar>

Description Model that assesses daily exposure to air pollution, which considers daily population mobility on a geographical scale and the spatial and temporal variability of pollutant concentrations, in addition to traditional parameters such as exposure time and pollutant concentration.

License GPL-3

URL https://github.com/flortames/Air-Exposure-Model

Depends R (>= 4.1.1)

Imports dplyr, htmltools, httr, jsonlite, leaflet, htmlwidgets, lubridate, sf

NeedsCompilation no

Repository CRAN

Date/Publication 2024-08-29 19:00:05 UTC

Contents

`AirExposure-package` *AirExposure*

Description

Model that assesses daily exposure to air pollution, taking into account several dynamic variables. The essence of this model focuses on integrating data related to daily routines, people's mobility and hourly concentrations of air pollutants. Users must provide accurate details about their daily activities, including information about their residence, activity sites, activity schedules, and the commuting. A key element is the use of the TomTom API, which determinated urban mobility patterns for a more complete view of travel in urban environments. The atmospheric pollutants considered in this model include are the criteria pollutants: carbon monoxide (CO), nitrogen dioxide (NO2), particulate matter (PM), ozone (O3) and sulfur dioxide (SO2) due to their potential health impacts. Additionally, a model with fixed variables is included, where it is assumed that a person is present 24 hours at a location (typically at home). The purpose of this function is to compare both methodologies and identify more realistic ways to estimate exposure and understand potential health effects.

Author(s)

Josefina Urquiza, Maria Florencia Tames, Maela Lupo, Alfredo Rigalli

Maintainer: Josefina Urquiza <jurquiza@conicet.gov.ar>

`alternative_trajectories` *Alternative Trajectories*

Description

Visualization of the response obtained from the request made to the TomTom API. It shows six travel alternatives between the origin and destination points

Usage

```
alternative_trajectories(origin, dest, mode, dir, key, output, hours = NULL, gridID,
shapeValue, units, pollutant)
```

Arguments

origin	String vector with the geographical coordinates of the trip's origin in decimal degree (EPSG:4326). This point is considered as the person's home. It should follow the structure 'latitude,longitude'. Please avoid inserting spaces between the data to ensure the correct format
dest	String vector with the geographical coordinates of the trip's destination in decimal degree (EPSG:4326). It should follow the structure 'latitude,longitude'. Please avoid inserting spaces between the data to ensure the correct format.
mode	String vector which details the type of transport selected to make the trip from the origin to the destination point. The types of mobility will depend on the information provided by the TomTom API for the study site. The most common alternatives are: 'car', 'motorcycle', 'pedestrian', 'bicycle'. For more information visit: https://developer.tomtom.com/routing-api/documentation/routing/calculate-route
dir	String vector with the path to the local directory where the shape files with the hourly pollutant grids are located.
key	String vector with your personal Tom-Tom API password. This API is free of charge and allows up to 2500 requests per day but accessing the information requires a user account. For more information, visit: https://developer.tomtom.com/how-to-get-tomtom-api-key.
output	String vector with the selected output format. It can be: a) plot, which show the alternative routes on an interactive map through the Leaflet library; b) df returns a dataframe with information on time, distance, concentrations of the alternative routes; c) polyline, returns a sf/data.frame (a line) object, which can be exported in shapefile format with sf library. The selection must be in lower case.
hours	String vector with the date and time when the commuting takes place. It must be entered with the following structure 'YYYY-MM-DD HH:MM:SS' so that it is recognized as an object of type POSIXct or POSIXt within the model (e.g. 2019-08-01 08:00:00).
gridID	String vector with the name of the field containing the unique identifier (ID) for the grid. The ID is a unique number assigned to each pixel within the concentration grid in the shapefile (e.g. 'GRI1_ID')
shapeValue	String vector with the name of the field which correspond to the concentrations values in the shapefile (e.g. 'value')
units	String with the units of measurement of the pollutant concentrations used to estimate the exposure. For, PM2.5 and PM10 should be in 'ug/m3'; CO and O3 should be in 'ppm'; SO2 and NO2 should be in 'ppb'. The units of measurement as well as the pollutants are based on the Air Quality Index(AQI) determined by the US EPA.
pollutant	String vector with the name of the selected pollutant. The model can estimate exposure for the following pollutants: PM10, PM2.5, SO2, NO2, CO, O3.

Details

From the request for information on route alternatives returned by trajectories_tomtom, this function has the purpose to classify and shows the routes obtained in three different formats. After obtaining

the six route alternatives, the next step involves classifying the route types into: i) 'faster' (according to travel time - faster); ii) 'shorter' (according to travel distance - short); iii) more polluted and less polluted (according to the concentrations for the specified day and time - 'morepol' and 'lesspol'); iv) higher and lower exposure (according to the concentrations and the duration of each route - 'moreexpos' and 'lessexpos'). The routes can be the same for different alternatives. For instance, the most polluted route may coincide with the fastest route. The function is used by total_exposure to obtain each of the trips taken during the day by an individual.

Value

Returns the specified object in the output parameter:

`df`	it provides information in a dataframe objset about the route. The variables include: alternative; ID, long, lat, departureTime, arrivalTime, lengthInMeters trafficLengthInMeters, travelMode, trafficDelayInSeconds, travelTimeInSeconds, liveTrafficIncidentsTravelTimeInSeconds, historicTrafficTravelTimeInSeconds, noTrafficTravelTimeInSeconds daily_pol_value_mean, exposure_value_mean and type
`plot`	It returns an object of type leaflet and htmlwidget corresponding to an interactive map where it is possible to visualize the streets through which each of the alternatives will pass
`polyline`	It returns an sf object with the same variables as df output but it can be represented as a geospatial object on a map, displaying spatial lines corresponding to each of the alternatives that traverse streets and routes.

Note

For more information about the variables obtained from the traffic data request to the TomTom API, please review the documentation of the calculateRoute service. TomTom Routing API Documentation https://developer.tomtom.com/routing-api/documentation/product-information/introduction

Author(s)

Josefina Urquiza, Maria Florencia Tames, Maela Lupo, Alfredo Rigalli

Maintainer: Josefina Urquiza <jurquiza@conicet.gov.ar>

`function_hours` *Hours*

Description

It aims to convert minutes into hours

Usage

```
function_hours(minutes)
```

Arguments

minutes	number of minutes

Details

Convert minutes into hours

Value

Returns a number corresponding to the number of hours.

Author(s)

Josefina Urquiza, Maria Florencia Tames, Maela Lupo, Alfredo Rigalli

Maintainer: Josefina Urquiza <jurquiza@conicet.gov.ar>

Examples

```
minutes<-560
function_hours(minutes)
# return de value "09:20"
```

---

hourly_grid *hourly_grid*

Description

Internal model function to search and identify the name of the concentration hourly grid associated with the date-time of the input data.

Usage

```
hourly_grid(hour_input, time_format, dir)
```

Arguments

hour_input	String vector with the date-time to determine the concentrations grid of interest. It is recommended to input with the following structure: "YYYY-MM-DD HH:MM:SS" to transform in an object POSIXct or POSIXt type.
time_format	String vector representing the structure using special strptime abbreviations, indicating how the date data was provided (e.g. Y-m-d H:M:S).
dir	String with the local directory path where the shapefiles containing information about concentrations of the pollutants of interest are located.

Details

The purpose of the function is to navigate through the user-provided directory containing the shapefiles with concentration grids to search and identify the name of the concentration hourly grid. The information is related to the date on which the daily air exposure is to be estimated. This is an internal function of the airExposure model, but is available if the user wishes to use it to search and check their files.

Value

Returns a string vector with the name of the file matching the input variables (date-time) which is the date of interest for estimating air exposure

Note

The shapefile files located in the local path should be 24 per day, that is, one grid per hour per day. Each shapefile should be named as YYYY-MM-DD_HHMM (e.g. 2019-08-01_1600). In the case of the 00:00 hour, must be set in a different way, with a 1 value in the the last value of minutes: YYYY-MM-DD_HHM1 (e.g. 2019-08-01_0001). All files must have at least two fields, such as: grid identification (e.g., ID) and pollutant concentration value (e.g., value).

Author(s)

Josefina Urquiza, Maria Florencia Tames, Maela Lupo, Alfredo Rigalli

Maintainer: Josefina Urquiza <jurquiza@conicet.gov.ar>

`map_colors` *Map colors.*

Description

Its purpose is to color the pollutant grids according to the EPA Air Quality Index (AQI). It corresponds to an internal function since it is used in the functions total_exposure, traditional_model and alternative_trajectories. It needs the pollutant grid and its name, adding two new columns with the color and the AQI breakpoints. The colors and their ranges will depend on the pollutant of interest.

Usage

```
map_colors(grid, pollutant)
```

Arguments

`grid`	Variable corresponding to the grid of interest in shapefile format.
`pollutant`	String vector with the name of the selected pollutant. The model can estimate exposure for the following pollutants: PM10, PM2.5, SO2, NO2, CO, O3.

Details

Allows for the configuration of colors for pollutant grids based on the EPA's Air Quality Index (AQI) ranges, determining potential effects of exposure to specific atmospheric pollutants. Within the package are functions like total_exposure, traditional_model and alternative_trajectories, that facilitate visualizing outputs via an interactive leaflet map. The map_color function uses pollutant concentration grids and, depending on the chosen function mention, generates a new grid with additional fields where colors and ranges are set for potential pollutant effects. Subsequently, each of the mentioned functions takes these columns and returns an interactive map with colored grids. The categories could be string values like "Good", "Moderate","Unhealthy for sensible groups", "Unhealthy", "Very Unhealthy", "Hazardous". The field colors is a string with the HEX color code like "#abdda4", "#f8fd66", "#fdde61", "#d74a4c", "#b687ba","#590e632"

Value

Returns the input grid with two additional fields named categories and color which are utilized in the total_exposure, traditional_model and alternative_trajectories functions to visualize data in interactive Leaflet maps.

Author(s)

Josefina Urquiza, Maria Florencia Tames, Maela Lupo, Alfredo Rigalli

Maintainer: Josefina Urquiza <jurquiza@conicet.gov.ar>

References

United States Environmental Protection Agency. (2018). Technical Assistance Document for the Reporting of Daily Air Quality - the Air Quality Index (AQI). Environmental Protection, 22. https://airnowtest.epa.gov/sites/default/files/2018-05/aqi-technical-assistance-document-may2016.pdf

---

`points_to_line` *Points to line*

Description

The purpose is to connect spatial points obtained with the request to the Tom-Tom API using lines on the street routes. Although it is an internal function, it can be used by the user considering the necessary arguments for this purpose

Usage

```
points_to_line(data, long, lat, id_field = NULL, sort_field = NULL)
```

Arguments

data	DataFrame that contains the coordinates of the points to be joined with lines. It must include information fields about latitude and longitude coordinates in decimal degree (EPSG:4326), line identification field (ID) and an additional column with point ordering values. The DataFrame used in this case corresponds to the output of the alternative_trajectories function, which must be modified to include the mentioned columns.
long	String vector with the name of the field of the dataframe (data) that contains the longitude data. The values of the field should be in decimal degree (EPSG:4326).
lat	String vector with the name of the field of the dataframe (data) that contains the latitude data. The values of the field should be in decimal degree (EPSG:4326).
id_field	String with the name of the field containing the line identification field (e.g. "ID"). The points with the same ID belong to the same line/street. ID values can be numeric or string values. Can receive NULL values.
sort_field	String vector with the name of the field with an identifier to order the points from smallest to largest according to their spatial location. The values must be numeric to represent the order sequence in which the points will be joined. This column is different from the "field_id" column

Value

Returns an object of type "sfc_LINESTRING" and "sfc" of the sf library, which can be visualized with the (plot) function.

Author(s)

Josefina Urquiza, Maria Florencia Tames, Maela Lupo, Alfredo Rigalli

Maintainer: Josefina Urquiza <jurquiza@conicet.gov.ar>

---

temporary_grid_search *Temporary grid search*

Description

The aim is to search for the grid with the name obtained in the function time_grid in the local path entered. If only one hour is entered, it returns the grid in shapefile format saved in the local directory. In the case of a range of hours, an average grid of the time range is returned.

Usage

```
temporary_grid_search(start_hour, end_hour = NULL, dir, time_format, gridID, shapeValue)
```

Arguments

start_hour	String vector that represents the date and time of interest to estimate the exposure, indicating the start time of the activity and/or the trip.
end_hour	String vector that represents the date and time of interest to estimate the exposure, indicating the end of the activity and/or the trip. If you want to obtain a specific time interval that denotes the duration of an activity and/or trip of the same day, you need to provide this parameter. If not, it defaults to NULL.
dir	String vector with the path to the local directory where the shapefiles that have the information about the concentrations of the pollutants of interest are located.
time_format	String vector representing the structure using special strptime abbreviations, indicating how the date data was provided (e.g.
gridID	String vector with the name of the field containing the unique identifier (ID) for the grid. The ID is a unique number assigned to each pixel within the concentration grid in the shapefile (e.g. GRI1_ID)
shapeValue	String vector with the name of the field corresponding to the concentrations in the shapefile files (e.g. value)

Details

The aim is to search for the grid with the name obtained in the function time_grid in the local path entered. If only one hour is entered, it returns the grid in shapefile format saved in the local directory. In the case of a range of hours, an average grid of the time range is returned.

Value

This function is used internally in the model of dynamic and fixed variables. The output is the reading of the grid in local path, where a file of class "sf" and "data.frame" is read.

Author(s)

Josefina Urquiza, Maria Florencia Tames, Maela Lupo, Alfredo Rigalli

Maintainer: Josefina Urquiza <jurquiza@conicet.gov.ar>

---

total_exposure *Total exposure*

Description

Dynamic exposure model to atmospheric pollutants with dynamic variables

Usage

```
total_exposure(travel_list, mode, dir, key, selection, output_exp, departure_time_home,
               activity_minutes, pollutant, shapeValue, gridID, units)
```

Arguments

travel_list	Dataframe with the coordinates (latitude and longitude) in decimal degree (EPSG:4326) of the origin(home) and destination(s)
mode	String indicating the mode of transport used for each of the trips made to the destination(s). Last, the type of transportation back home must be included. This information depends on what is provided by the Tom-Tom API, specifically for the site of interest. The most common alternatives are: "bus", "car", "motorcycle", "pedestrian", "bicycle". The different alternatives can be combined. For more information, please visit https://developer.tomtom.com/.
dir	String vector with the path to the local directory where the shape files with the hourly pollutant grids are located.
key	String vector with your personal TomTom API password. This API is free of charge and allows up to 2500 requests per day but accessing the information requires a user account. For more information, visit: https://developer.tomtom.com/how-to-get-tomtom-api-key.
selection	String vector where the type of route to be used in each of the routes is indicated. The alternatives of this parameter can be: "faster" route, "shorter" route, route with lower concentrations as "lesspol", route with higher concentrations as "morepol", route with lower exposure as "lessexpos", route with higher exposure as "moreexpos" or a combination of them. You must enter the same number of data as the number of coordinates, since the return trip home is considered.
output_exp	String vector with the selected output format. It can be: a) "plot", which show the selected routes on an interactive map through the Leaflet library with information about de daily exposure estimated; b) "df" returns a dataframe with information on time, concentration and exposure in the 3 moments, during stay in origin, commuting and destination.
departure_time_home	String vector that provides the date-time of leaving home for the first time in the day to a destination. Must be in the format YYYYY-MM-DD HH:MM:SS to transform in an object POSIXct or POSIXt type.
activity_minutes	Dataframe with the time in minutes (number) of the duration of the activity(ies) performed at each destination.
pollutant	String vector with the name of the selected pollutant to be estimate. The model estimates exposures to the criteria pollutants according to the EPA's Air Quality Index (AQI). They must be entered in the following way: "PM10", "PM2.5", "SO2", "NO2", "CO", "O3".
shapeValue	String vector with the name of the field corresponding to the concentrations in the shapefile files (e.g. "value").
gridID	String vector with the name of the field containing the unique identifier (ID) for the grid. The ID is a unique number assigned to each pixel within the concentration grid in the shapefile (e.g. "GRI1_ID")
units	String vector with the units of the concentrations of the pollutant. For example, PM2.5 and PM10 should be in ug/ m3, CO and O3 should be in ppm, SO2 and NO2 should be in ppb.

Details

Estimates daily exposure by taking into account dynamic variables, specifically the mobility and activities of individuals throughout the day and in different locations. Accounts for the 3 exposure moments of the day, at the origin, during commuting, and at the destination.

Value

It returns the result of the daily exposure estimation of the pollutant of interest considering the routes and activities. The function can return two types of outputs

`Dataframe` with the results of the estimation divided into each site visited and each route taken.

`Interactive map` with Leaflet showing the selected routes and the daily concentration grid.

Note

The function incorporates other functions like trajectories_tomtom, alternative_trajectories, points_to_line, temporary_grid_search, map_colors and hourly_grid.

Author(s)

Josefina Urquiza, Maria Florencia Tames, Maela Lupo, Alfredo Rigalli

Maintainer: Josefina Urquiza <jurquiza@conicet.gov.ar>

See Also

function_hours, map_colors(), hourly_grid(),alternative_trajectories

---

`trajectories_tomtom` *Trajectories TomTom*

Description

Requests information from the Tom-Tom API related to traffic.

Usage

```
trajectories_tomtom(origin, dest, mode, hour_trajectory, key)
```

Arguments

origin	String vector with the geographical coordinates of the trip's origin in decimal degree (EPSG:4326).The point is considered as the person's home. It should follow the structure 'latitude,longitude'. Please avoid inserting spaces between the data to ensure the correct format.
dest	String vector with the geographical coordinates of the trip's destination in decimal degree (EPSG:4326). It should follow the structure 'latitude,longitude'. It corresponds to the place where the person is doing an activity. Please avoid inserting spaces between the data to ensure the correct format.
mode	String vector which details the type of transport selected to make the trip from the origin to the destination point. The types of mobility will depend on the information provided by the TomTom API for the study site. The most common alternatives are: "car", "motorcycle", "pedestrian", "bicycle". For more information visit: https://developer.tomtom.com/routing-api/documentation/routing/calculate-route.
hour_trajectory	String vector with the date and time when the journey took place. It must be entered in text string format with the following structure: 'YYYY-MM-DD HH:MM:SS
key	String vector with your personal TomTom API password. This API is free of charge and allows up to 2500 requests per day but accessing the information requires a user account. For more information, visit: https://developer.tomtom.com/how-to-get-tomtom-api-key

Details

Seeks to obtain traffic information from the Tom-Tom API, by requesting six different travel alternatives based on the day, time, mode of transport, and the origin and destination points provided by the user.

Value

Returns a dataframe with 14 fields, where: 1) ID corresponds to an identification number of each of the 6 alternatives provided by the API; 2) long and 3) lat correspond to the latitude and longitude of all the points along the route taken; 4) departureTime departure time from the origin to destination; 5) arrivalTime arrival time at the destination; 6) "alternative" name of the alternatives, from alternative_1 to alternative_6. he next fields contain information about the travel. Please see de documentation of Tom-Tom API of: "trafficLengthInMeters", "travelMode", "trafficDelayInSeconds",] "travelTimeInSeconds", "liveTrafficIncidentsTravelTimeInSeconds", "historicTrafficTravelTimeInSeconds", "noTrafficTravelTimeInSeconds"

Note

An Internet connection is required to make requests to the TomTom API.

Author(s)

Josefina Urquiza, Maria Florencia Tames, Maela Lupo, Alfredo Rigalli

Maintainer: Josefina Urquiza <jurquiza@conicet.gov.ar>

Index