

Package ‘EmissV’

April 2, 2019

Title Vehicular Emissions by Top-Down Methods

Date 2019-04-02

Version 0.665.1.0

Maintainer Daniel Schuch <schuch@usp.br>

Description Creates emissions for use in air quality models. Vehicular emissions are estimated by a top-down approach, total emissions are calculated using the statistical description of the fleet of vehicles, the emission is spatially distributed according to satellite images or openstreetmap data <<https://www.openstreetmap.org>> and then distributed temporarily (Vara-Vela et al., 2016, <doi:10.5194/acp-16-777-2016>).

Depends R (>= 3.4)

Imports ncdf4, units(>= 0.5-1), raster, sp, sf, methods, data.table, lwgeom

Suggests testthat, covr, osmar, rgdal

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

URL <https://atmoschem.github.io/EmissV>

BugReports <https://github.com/atmoschem/EmissV/issues>

NeedsCompilation no

Author Daniel Schuch [aut, cre] (<<https://orcid.org/0000-0001-5977-4519>>), Sergio Ibarra-Espinosa [aut] (<<https://orcid.org/0000-0002-3162-1905>>)

Repository CRAN

Date/Publication 2019-04-02 17:22:53 UTC

R topics documented:

areaSource	2
emission	3
emissionFactor	4
gridInfo	5
lineSource	6
perfil	7
plumeRise	10
pointSource	11
rasterSource	12
read	13
speciation	15
species	16
streetDist	18
totalEmission	20
totalVOC	21
vehicles	22
Index	24

areaSource	<i>Distribution of emissions by area</i>
------------	--

Description

Calculate the spatial distribution by a raster kasked by shape/model grid information.

Usage

```
areaSource(s, r, grid = NA, name = "", as_frac = F, verbose = T)
```

Arguments

s	input shape object
r	input raster object
grid	grid with the output format
name	area name
as_frac	return a fraction instead of a raster
verbose	display additional data

Format

a raster

Details

About the DMSP and example data https://en.wikipedia.org/wiki/Defense_Meteorological_Satellite_Program

Source

Data available <http://www.ospo.noaa.gov/Operations/DMSP/index.html>

Examples

```
shape <- raster::shapefile(paste(system.file("extdata", package = "EmissV"),
                                   "/BR.shp", sep=""), verbose = FALSE)
shape <- shape[22,1] # subset for Sao Paulo - BR
raster <- raster::raster(paste(system.file("extdata", package = "EmissV"),
                                   "/dmsp.tiff", sep=""))
grid <- gridInfo(paste(system.file("extdata", package = "EmissV"), "/wrfinput_d02", sep=""))
SP <- areaSource(shape,raster,grid,name = "SPMA")

sp::splot(SP,scales = list(draw=TRUE),ylab="Lat",xlab="Lon",
          main=list(label="Spatial Distribution by Lights for Sao Paulo - Brazil"),
          col.regions = c("#031638", "#001E48", "#002756", "#003062",
                        "#003A6E", "#004579", "#005084", "#005C8E",
                        "#006897", "#0074A1", "#0081AA", "#008FB3",
                        "#009EBD", "#00AFC8", "#00C2D6", "#00E3F0"))
```

 emission

Emissions in the format for atmospheric models

Description

Combine area sources and total emissions to model output

Usage

```
emission(total, pol, area, grid, inventory = NULL, mm = 1,
          aerosol = F, plot = F, verbose = T)
```

Arguments

total	list of total emission
pol	pollutant name
area	list of area sources or matrix with a spatial distribution
grid	grid information
inventory	a inventory raster from read
mm	pollutant molar mass

aerosol	TRUE for aerosols and FALSE (default) for gazes
plot	TRUE for plot the final emissions
verbose	display additional information

Format

matrix of emission

Note

if Inventory is provided, the firsts tree arguments are not be used by the functon.

Is a good practice use the `set_units(fe,your_unity)`, where `fe` is your emission factory and `your_unity` is usually `g/km` on your emission factory

the list of area must be in the same order as defined in vehicles and total emission.

just WRF-Chem is suported by now

See Also

[totalEmission](#) and [areaSource](#)

Examples

```
fleet <- vehicles(example = TRUE)

EmissionFactors <- emissionFactor(example = TRUE)

TOTAL <- totalEmission(fleet,EmissionFactors,pol = c("CO"),verbose = TRUE)

grid <- gridInfo(paste0(system.file("extdata", package = "EmissV"),"/wrfinput_d01"))
shape <- raster::shapefile(paste0(system.file("extdata", package = "EmissV"),"/BR.shp"))
raster <- raster::raster(paste0(system.file("extdata", package = "EmissV"),"/dmsp.tiff"))

SP <- areaSource(shape[22,1],raster,grid,name = "SP")
RJ <- areaSource(shape[17,1],raster,grid,name = "RJ")

e_CO <- emission(TOTAL,"CO",list(SP = SP, RJ = RJ),grid,mm=28)
```

emissionFactor *Tool to set-up emission factors*

Description

Return a data frame with vehicle information. Types argument defines the diary use:

Usage

```
emissionFactor(ef, pollutant = names(ef), vnames = NA, unit = "g/km",
  example = F, verbose = T)
```

Arguments

ef	list with emission factors
pollutant	pollutant names
vnames	name of each vehicle category (optional)
unit	tring with unit from unit package, for default is "g/km"
example	TRUE to display a simple example
verbose	display additional information

Format

data frame

See Also

[areaSource](#) and [totalEmission](#)

Examples

```
EF <- emissionFactor(example = TRUE)

# or the code for the same result
EF <- emissionFactor(ef = list(CO = c(1.75,10.04,0.39,0.45,0.77,1.48,1.61,0.75),
  PM = c(0.0013,0.0,0.0010,0.0612,0.1052,0.1693,0.0,0.0)),
  vnames = c("Light Duty Vehicles Gasohol", "Light Duty Vehicles Ethanol",
    "Light Duty Vehicles Flex", "Diesel Trucks", "Diesel Urban Busses",
    "Diesel Intercity Busses", "Gasohol Motorcycles",
    "Flex Motorcycles"))
```

gridInfo

Read grid information from a NetCDF file

Description

Return a list containing information of a regular grid / domain

Usage

```
gridInfo(file = file.choose(), z = F, geo = F, verbose = T)
```

Arguments

file	file name/path to a wrfinput, wrfchemi or geog_em file
z	TRUE for read wrfinput vertical coordinades
geo	True for use geog_em files
verbose	display additional information

Note

just WRF-Chem is suported by now

Examples

```
grid_d1 <- gridInfo(paste(system.file("extdata", package = "EmissV"), "/wrfinput_d01", sep=""))

grid_d2 <- gridInfo(paste(system.file("extdata", package = "EmissV"), "/wrfinput_d02", sep=""))
grid_d3 <- gridInfo(paste(system.file("extdata", package = "EmissV"), "/wrfinput_d03", sep=""))
names(grid_d1)
# for plot the shapes
library(sp)
shape <- raster::shapefile(paste0(system.file("extdata", package = "EmissV"), "/BR.shp"))
plot(shape, xlim = c(-55, -40), ylim = c(-30, -15), main="3 nested domains")
axis(1); axis(2); box(); grid()
lines(grid_d1$Box, col = "red")
text(grid_d1$xlim[2], grid_d1$ylim[1], "d1", pos=4, offset = 0.5)
lines(grid_d2$Box, col = "red")
text(grid_d2$xlim[2], grid_d2$ylim[1], "d2", pos=4, offset = 0.5)
lines(grid_d3$Box, col = "red")
text(grid_d3$xlim[1], grid_d3$ylim[2], "d3", pos=2, offset = 0.0)
```

lineSource

Distribution of emissions by lines

Description

Create a emission distribution from 'sp' or 'sf' spatial lines data.frame or spatial lines.

There 3 modes available to create the emission grid: - using gridInfo function output (default) - using the patch to "wrfinput" (output from real.exe) file or "geo" for (output from geog.exe) - "sf" (and "sp") uses a grid in SpatialPolygons format

The variable is the column of the data.frame with contains the variable to be used as emissions, by default the idistribution taken into account the length distribution of lines into each grid cell and the output is normalized.

Usage

```
lineSource(s, grid, as_raster = F, verbose = T, type = "info",
          gcol = 100, grow = 100, variable = "length")
```

Arguments

s	SpatialLinesDataFrame of SpatialLines object
grid	grid object with the grid information or filename
as_raster	output format, TRUE for raster, FALSE for matrix
verbose	display additional information
type	"info" (default), "wrfinput", "geo", "sp" or "sf" for grid type
gcol	grid points for a "sp" or "sf" type
grow	grid points for a "sp" or "sf" type
variable	variable to use, default is line length

Source

OpenstreetMap data available <https://www.openstreetmap.org/> and <https://download.geofabrik.de/>

See Also

[gridInfo](#) and [rasterSource](#)

Examples

```
roads <- osmar::get_osm(osmar::complete_file(),
                      source = osmar::osmsource_file(paste(system.file("extdata",
                                                                    package="EmissV"),"/streets.osm.xz",sep="")))
road_lines <- osmar::as_sp(roads,what = "lines")
roads <- sf::st_as_sf(road_lines)

d3 <- gridInfo(paste0(system.file("extdata", package = "EmissV"),"/wrfinput_d03"))

roadLength <- lineSource(roads,d3,as_raster=TRUE)
sp::splot(roadLength, scales = list(draw=TRUE), ylab="Lat", xlab="Lon",main="Length of roads",
          sp.layout=list("sp.lines", road_lines))
```

Description

Set of hourly profiles that represents the mean activity for each hour (local time) of the week.

LDV Light Duty vehicles

HDV Heavy Duty vehicles

PC_JUNE_2012 passenger cars counted in June 2012

PC_JUNE_2013 passenger cars counted in June 2013

PC_JUNE_2014 passenger cars counted in June 2014

LCV_JUNE_2012 light comercial vehicles counted in June 2012

LCV_JUNE_2013 light comercial vehicles counted in June 2013

LCV_JUNE_2014 light comercial vehicles counted in June 2014

MC_JUNE_2012 motorcycles counted in June 2012

MC_JUNE_2013 motorcycles counted in June 2013

MC_JUNE_2014 motorcycles counted in June 2014

HGV_JUNE_2012 Heavy good vehicles counted in June 2012

HGV_JUNE_2013 Heavy good vehicles counted in June 2013

HGV_JUNE_2014 Heavy good vehicles counted in June 2014

PC_JANUARY_2012 passenger cars counted in january 2012

PC_JANUARY_2013 passenger cars counted in january 2013

PC_JANUARY_2014 passenger cars counted in january 2014

LCV_JANUARY_2012 light comercial vehicles counted in january 2012

LCV_JANUARY_2013 light comercial vehicles counted in january 2013

LCV_JANUARY_2014 light comercial vehicles counted in january 2014

MC_JANUARY_2012 Motorcycles counted in january 2012

MC_JANUARY_2014 Motorcycles counted in january 2014

HGV_JANUARY_2012 Heavy good vehicles counted in january 2012

HGV_JANUARY_2013 Heavy good vehicles counted in january 2013

HGV_JANUARY_2014 Heavy good vehicles counted in january 2014

POW Power generation emission profile

IND Industrial emission profile

RES Residencial emission profile

TRA Transport emission profile

AGR Agriculture emission profile

SHP Emission profile for ships

SLV Solvent use emission constant profile

WBD Waste burning emissio constant profile

PC_nov_2018 passenger cars at Janio Quadros on November 2018

HGV_nov_2018 heavy good vehicles at Janio Quadros on November 2018

TOTAL_nov_2018 total vehicle at Janio Quadros on November 2018

PC_out_2018 passenger cars at Anhanguera-Castello Branco on October 2018

MC_out_2018 Motorcycles cars at Anhanguera-Castello Branco on October 2018

Usage

```
data(perfil)
```

Format

A list of data frames with activity by hour and weekday.

Details

- Profiles 1 to 2 are from traffic count at São Paulo city from Perez Martínez et al (2014).
- Profiles 3 to 25 comes from traffic counted of toll stations located in São Paulo city, for summer and winters of 2012, 2013 and 2014.
- Profiles 26 to 33 are for different sectors from Oliver et al (2003).
- Profiles 34 to 36 are for volumetric mechanized traffic count at Janio Quadros tunnel on November 2018.
- Profiles 37 to 38 are for volumetric mechanized traffic count at Anhanguera-Castello Branco on October 2018.

Note

The profile is normalized by days (but is balanced for a complete week) it means $\text{diary_emission} \times \text{profile} = \text{hourly_emission}$.

References

Pérez-Martínez, P. J., Miranda, R. M., Nogueira, T., Guardani, M. L., Fornaro, A., Ynoue, R., & Andrade, M. F. (2014). Emission factors of air pollutants from vehicles measured inside road tunnels in São Paulo: case study comparison. *International Journal of Environmental Science and Technology*, 11(8), 2155-2168.

Olivier, J., J. Peters, C. Granier, G. Pétron, J.F. Müller, and S. Wallens, Present and future surface emissions of atmospheric compounds, POET Report #2, EU project EVK2-1999-00011, 2003.

Examples

```
# load the data
data(perfil)

# function to simple view
plot.perfil <- function(per = perfil$LDV, text="", color = "#0000FFBB"){
  plot(per[,1],ty = "l", ylim = range(per),axe = FALSE,
        xlab = "hour",ylab = "Intensity",main = text,col=color)
  for(i in 2:7){
    lines(per[,i],col = color)
  }
  for(i in 1:7){
    points(per[,i],col = "black", pch = 20)
  }
  axis(1,at=0.5+c(0,6,12,18,24),labels = c("00:00","06:00","12:00","18:00","00:00"))
  axis(2)
```

```
    box()
  }

# view all profiles in perfil data
for(i in 1:length(names(perfil))){
  cat(paste("profile", i, names(perfil)[i], "\n"))
  plot.perfil(perfil[[i]], names(perfil)[i])
}
```

plumeRise

Calculate plume rise height.

Description

Calculate the maximum height of rise based on Brigs (1975), the height is calculated using different formulations depending on stability and wind conditions.

Usage

```
plumeRise(df, imax = 10, ermax = 1/100, Hmax = T, verbose = T)
```

Arguments

df	data.frame with micrometeorological and emission data
imax	maximum number of iterations
ermax	maximum error
Hmax	use weil limit for plume rise, see details
verbose	display additional information

Format

data.frame with the input, rise (m) and effective higt (m)

Details

The input data.frame must contains the folloing colluns:

- z: height of the emission (m)
- r: source raius (m)
- Ve: emission velocity (m/s)
- Te: emission temperature (K)
- ws: wind speed (m/s)
- Temp: ambient temperature (K)
- h: height of the Atmospheric Boundary Layer-ABL (m)

- L: Monin-Obuhkov Lench (m)
- dtdz: lapse ration of potential temperature, used only for stable ABL (K/m)
- Ustar: atriction velocity, used only for neutral ABL (m/s)
- Wstar: scale of convectie velocity, used only for convective ABL (m/s)

Additionally some combination of ws, Wstar and Ustar can produce inaccurate results, Weil (1979) propose a geometric limit of $0.62 * (h - H_s)$ for the rise value.

References

The plume rise formulas are from Brigs (1975): "Brigs, G. A. Plume rise predictions, Lectures on Air Pollution and Environmental Impact Analyses. Amer. Meteor. Soc. p. 59-111, 1975." and Arya 1999: "Arya, S.P., 1999, Air Pollution Meteorology and Dispersion, Oxford University Press, New York, 310 p."

The limits are from Weil (1979): "WEIL, J.C. Assessmet of plume rise and dispersion models using LIDAR data, PPSP-MP-24. Prepared by Environmental Center, Martin Marietta Corporation, for Maryland Department of Natural Resources. 1979."

The example is data from a chimney of the Candiota thermoelectric powerplant from Arabage et al (2006) "Arabage, M. C.; Degrazia, G. A.; Moraes O. L. Simulação euleriana da dispersão local da pluma de poluente atmosférico de Candiota-RS. Revista Brasileira de Meteorologia, v.21, n.2, p. 153-160, 2006."

Examples

```
candiota <- matrix(c(150,1,20,420,3.11,273.15 + 3.16,200,-34.86,3.11,0.33,
150,1,20,420,3.81,273.15 + 4.69,300,-34.83,3.81,0.40,
150,1,20,420,3.23,273.15 + 5.53,400,-24.43,3.23,0.48,
150,1,20,420,3.47,273.15 + 6.41,500,-15.15,3.48,0.52,
150,1,20,420,3.37,273.15 + 6.35,600, -8.85,3.37,2.30,
150,1,20,420,3.69,273.15 + 5.93,800,-10.08,3.69,2.80,
150,1,20,420,3.59,273.15 + 6.08,800, -7.23,3.49,1.57,
150,1,20,420,4.14,273.15 + 6.53,900,-28.12,4.14,0.97),
ncol = 10, byrow = TRUE)
candiota <- data.frame(candiota)
names(candiota) <- c("z","r","Ve","Te","ws","Temp","h","L","Ustar","Wstar")
row.names(candiota) <- c("08:00","09:00",paste(10:15,":00",sep=""))
candiota <- plumeRise(candiota,Hmax = TRUE)
print(candiota)
```

pointSource

Emissions from point sources

Description

Transform a set of points into a grinded output

Usage

```
pointSource(emissions, grid, verbose = T)
```

Arguments

emissions	list of points
grid	grid object with the grid information
verbose	display additional information

Value

a raster

See Also

[gridInfo](#) and [rasterSource](#)

Examples

```
d1 <- gridInfo(paste(system.file("extdata", package = "EmissV"), "/wrfinput_d01", sep=""))

p = data.frame(lat      = c(-22,-22,-23.5),
               lon      = c(-46,-48,-47 ),
               z        = c(0 , 0, 0 ),
               emission = c(666,444,111 ) )

p_emissions <- pointSource(emissions = p, grid = d1)

sp::spplot(p_emissions,scales = list(draw=TRUE), ylab="Lat", xlab="Lon",
           main = "3 point sources for domain d1")
```

rasterSource

Distribution of emissions by a georeferenced image

Description

Calculate the spatial distribution by a raster

Usage

```
rasterSource(r, grid, nlevels = "all", conservative = T, verbose = T)
```

Arguments

r	input raster object
grid	grid object with the grid information
nlevels	number of vertical levels off the emission array
conservative	TRUE (default) to conserve total mass, FALSE to conserve flux
verbose	display additional information

Details

About the DMSP and example data https://en.wikipedia.org/wiki/Defense_Meteorological_Satellite_Program

Value

Returns a matrix

Source

Exemple data is a low resolution cutting from image of persistent lights of the Defense Meteorological Satellite Program (DMSP) https://pt.wikipedia.org/wiki/Defense_Meteorological_Satellite_Program

Data available <http://www.ospo.noaa.gov/Operations/DMSP/index.html>

See Also

[gridInfo](#) and [lineSource](#)

Examples

```
grid <- gridInfo(paste(system.file("extdata", package = "EmissV"),"/wrfinput_d01",sep=""))
x <- raster::raster(paste(system.file("extdata", package = "EmissV"),"/dmsp.tiff",sep=""))
test <- rasterSource(x,grid)
```

```
image(test,axe = F, main = "Spatial distribution by Persistent Nocturnal Lights from DMSP")
```

read

Read NetCDF data from global inventories

Description

Read data from global inventories, can read several files and merge into one emission and/or split into several species (speciation process)

Usage

```
read(file = file.choose(), coef = rep(1, length(file)), spec = NULL,
     version = "EDGAR", month = 1, year = 1, categories,
     as_raster = T, verbose = T)
```

Arguments

file	file name or names (variables are summed)
coef	coef to merge different sources (file) into one emission
spec	numeric speciation vector to split emission into different species
version	inventory name 'EDGAR', 'MACCITY' or 'GAINS'
month	the desired month of the inventory (MACCITY)
year	scenario index (GAINS)
categories	considered categories (MACCITY, GAINS variable names), empty for all
as_raster	return a raster (default) or matrix (with units)
verbose	display additional information

Value

Matrix or raster

Note

for 'GAINS' version, please use flux (kg m⁻² s⁻¹) NetCDF file from <https://eccad3.sedoo.fr>

Source

Read about EDGAR at <http://edgar.jrc.ec.europa.eu> and MACCITY at http://accent.aero.jussieu.fr/MACC_metadata.php

References

Janssens-Maenhout, G., Dentener, F., Van Aardenne, J., Monni, S., Pagliari, V., Orlandini, L., ... & Wankmüller, R. (2012). EDGAR-HTAP: a harmonized gridded air pollution emission dataset based on national inventories. European Commission Joint Research Centre Institute for Environment and Sustainability. JRC 68434 UR 25229 EUR 25229, ISBN 978-92-79-23123-0.

Lamarque, J.-F., Bond, T. C., Eyring, V., Granier, C., Heil, A., Klimont, Z., Lee, D., Liousse, C., Mieville, A., Owen, B., Schultz, M. G., Shindell, D., Smith, S. J., Stehfest, E., Van Aardenne, J., Cooper, O. R., Kainuma, M., Mahowald, N., McConnell, J. R., Naik, V., Riahi, K., and van Vuuren, D. P.: Historical (1850-2000) gridded anthropogenic and biomass burning emissions of reactive gases and aerosols: methodology and application, *Atmos. Chem. Phys.*, 10, 7017-7039, doi:10.5194/acp-10-7017-2010, 2010.

Z Klimont, S. J. Smith and J Cofala The last decade of global anthropogenic sulfur dioxide: 2000–2011 emissions *Environmental Research Letters* 8, 014003, 2013

See Also

[rasterSource](#) and [gridInfo](#)
[species](#)

Examples

```
d1 <- gridInfo(paste(system.file("extdata", package = "EmissV"),"/wrfinput_d01",sep=""))
d2 <- gridInfo(paste(system.file("extdata", package = "EmissV"),"/wrfinput_d02",sep=""))
print("download and untar EDGAR data from:")
print("http://edgar.jrc.ec.europa.eu/gallery.php?release=v431_v2&substance=NOx&sector=TR0")
nox <- read("v431_v2_REFERENCE_NOx_2010_10_TRO.0.1x0.1.nc")
sp::splot(nox, scales = list(draw=TRUE), xlab="Lat", ylab="Lon",main="NOx emissions from EDGAR")
nox_d1 <- rasterSource(nox,d1)
nox_d2 <- rasterSource(nox,d2)
image(nox_d1, main = "NOx emissions from transport from EDGAR 3.4.1 for d1")
image(nox_d2, main = "NOx emissions from transport from EDGAR 3.4.1 for d2")
```

speciation

Speciation of emissions in different compounds

Description

Distribute the total mass of estimated emissions into model species.

Usage

```
speciation(total, spec = NULL, verbose = T)
```

Arguments

total	emissions from totalEmissions
spec	numeric speciation vector of species
verbose	display additional information

See Also

[species](#)

Examples

```
veic <- vehicles(example = TRUE)
EmissionFactors <- emissionFactor(example = TRUE)
TOTAL <- totalEmission(veic,EmissionFactors,pol = "PM")
pm_iag <- c(E_PM25I = 0.0509200,
           E_PM25J = 0.1527600,
           E_ECI  = 0.1196620,
```

```

E_ECJ = 0.0076380,
E_ORGI = 0.0534660,
E_ORGJ = 0.2279340,
E_S04I = 0.0063784,
E_S04J = 0.0405216,
E_NO3J = 0.0024656,
E_NO3I = 0.0082544,
E_PM10 = 0.3300000)
PM <- speciation(TOTAL,pm_iag)

```

species

Species mapping tables

Description

Set of tables for speciation:

voc_radm2_mic Volatile organic compounds for RADM2

voc_cbmz_mic Volatile organic compounds for CBMZ

voc_moz_mic Volatile organic compounds for MOZART

voc_saprc99_mic volatile organic compounds for SAPRC99

vehicularvoc_radm2_iag Vehicular volatile organic compounds for RADM2 (MOL)

vehicularvoc_cbmz_iag Vehicular volatile organic compounds for CBMZ (MOL)

vehicularvoc_moz_iag Vehicular volatile organic compounds for MOZART (MOL)

vehicularvoc_saprc99_iag Vehicular volatile organic compounds for SAPRC99 (MOL)

pm_madesorgan_iag Particulate matter for made/sorgan

pm25_madesorgan_iag Fine particulate matter for made/sorgan

nox_iag Nox split Perez Martínez et al (2014)

nox_bcom Nox split usin Ntziachristos and Zamaras (2016)

voc_radm2_edgar432 Volatile organic compounds species from EDGAR 4.3.2 for RADM2 (MOL)

voc_moz_edgar432 Volatile organic compounds species from EDGAR 4.3.2 for MOZART (MOL)

- Volatile organic compounds species map from 1 to 4 are from Li et al (2014) taken into account several sources of pollutants.

- Volatile organic compounds from vehicular activity species map 5 to 8 is a by fuel and emission process from USP-IAG tunel experiments (Rafee et al., 2017) emitted by the process of exhaust (through the exhaust pipe), liquid (carter and evaporative) and vapor (fuel transfer operations).

- Particulate matter speciesmap for made/sorgan emissions 9 and 10.

- Nox split using Perez Martínez et al (2014) data (11).

- Nox split using mean of Ntziachristos and Zamaras (2016) data (12).

- Volatile organic compounds species map 13 and 14 are the corespondence from EDGAR 4.3.2 VOC specialization to RADM2 and MOZART.

Usage

data(species)

Format

List of numeric vectors with the 'names()' of the species and the values of each species.

Details

iag-voc: After estimating all the emissions of NMHC, it was used the speciation presented in (RAFEE et al., 2017). This speciation is based on tunnel measurements in São Paulo, depends on the type of fuel (E25, E100 and B5) and provides the mass of each chemical compound as mol/g. This speciation splits the NMHC from evaporative, liquid and exhaust emissions of E25, E100 and B5, into minimum compounds required for the Carbon Bond Mechanism (CBMZ) (ZAVERI; PETERS, 1999). Atmospheric simulations using the same pollutants in Brazil have resulted in good agreement with observations (ANDRADE et al., 2015).

iag-pm: data tunnel experiments at São Paulo in Perez Martínez et al (2014)

iag-nox: common NOx split for São Paulo Metropolitan area.

bcom-nox: mean of Ntziachristos and Zamaras (2016) data.

mic: from Li et al (2014).

edgar: Edgar 4.3.2 emissions Crippa et al. (2018).

Note

The units are mass ratio (mass/mass) or MOL (MOL), this last case do not change the default 'mm' into 'emission()' function.

References

- Li, M., Zhang, Q., Streets, D. G., He, K. B., Cheng, Y. F., Emmons, L. K., ... & Su, H. (2014). Mapping Asian anthropogenic emissions of non-methane volatile organic compounds to multiple chemical mechanisms. *Atmos. Chem. Phys.*, 14(11), 5617-5638.
- Huang, G., Brook, R., Crippa, M., Janssens-Maenhout, G., Schieberle, C., Dore, C., ... & Friedrich, R. (2017). Speciation of anthropogenic emissions of non-methane volatile organic compounds: a global gridded data set for 1970–2012. *Atmospheric Chemistry and Physics*, 17(12), 7683.
- Abou Rafee, S. A., Martins, L. D., Kawashima, A. B., Almeida, D. S., Morais, M. V. B., Souza, R. V. A., Oliveira, M. B. L., Souza, R. A. F., Medeiros, A. S. S., Urbina, V., Freitas, E. D., Martin, S. T., and Martins, J. A.: Contributions of mobile, stationary and biogenic sources to air pollution in the Amazon rainforest: a numerical study with the WRF-Chem model, *Atmos. Chem. Phys.*, 17, 7977-7995, <https://doi.org/10.5194/acp-17-7977-2017>, 2017.
- Martins, L. D., Andrade, M. F. D., Freitas, E., Pretto, A., Gatti, L. V., Junior, O. M. A., et al. (2006). Emission factors for gas-powered vehicles traveling through road tunnels in Sao Paulo, Brazil. *Environ. Sci. Technol.* 40, 6722–6729. doi: 10.1021/es052441u
- Pérez-Martínez, P. J., Miranda, R. M., Nogueira, T., Guardani, M. L., Fornaro, A., Ynoue, R., & Andrade, M. F. (2014). Emission factors of air pollutants from vehicles measured inside road

tunnels in São Paulo: case study comparison. *International Journal of Environmental Science and Technology*, 11(8), 2155-2168.

ANDRADE, M. d. F. et al. Air quality forecasting system for southeastern brazil. *Frontiers in Environmental Science, Frontiers*, v. 3, p. 1–12, 2015.

Ntziachristos L. and Samaras Z. 2016. Passneger cars, light commercial trucks, heavy-duty vehicles including buses and motor cycles. In: EEA, EMEP. EEA air pollutant emission inventory guidebook-2009. European Environment Agency, Copenhagen, 2016.

Crippa, M., Guizzardi, D., Muntean, M., Schaaf, E., Dentener, F., Aardenne, J. A. V., ... & Janssens-Maenhout, G. (2018). Gridded emissions of air pollutants for the period 1970–2012 within EDGAR v4.3.2. *Earth System Science Data*, 10(4), 1987-2013.

See Also

[speciation](#) and [read](#)

Examples

```
# load the mapping tables
data(species)
# names of each mapping tables
for(i in 1:length(names(species)))
  cat(paste0("specie map ",i," ",names(species)[i],"\n"))
# names of species contained in the (first) mapping table
names(species[[1]])
# The first mapping table / species and values
species[1]
```

streetDist

Distribution by OpenStreetMap street

Description

Distribute emissions by streets of OpenStreetMap

Usage

```
streetDist(emission = 1, dist = c(1, 0, 0, 0, 0), grid = NULL,
  osm = NULL, epsg = 31983, warnings = FALSE)
```

Arguments

emission	Numeric; emissions.
dist	Numeric; vector with length 5. The order represents motorway, trunk, primary, secondary and tertiary
grid	'sf' POLYGON; grid of polygons class sf.
osm	streets of OpenStreetMaps class sf
epsg	Numeric; spatial code for projecting spatial data
warnings	Logical; to show warnings.


```
s <- streetDist (emission = 1, dist = c (1, 0, 0, 0, 0), grid = ras,
              osm = dat, epsg = utm)

## End(Not run)
```

totalEmission	<i>Calculate total emissions</i>
---------------	----------------------------------

Description

Calculate the total emission with:

$$\text{Emission}(\text{pollutant}) = \text{sum}(\text{Vehicles}(n) * \text{Km_day_use}(n) * \text{Emission_Factor}(n, \text{pollutant}))$$

where n is the type of the vehicle

Usage

```
totalEmission(v, ef, pol, verbose = T)
```

Arguments

v	dataframe with the vehicle data
ef	emission factor
pol	pollutant name in ef
verbose	display additional information

Format

Return a list with the daily total emission by interest area (cities, states, countries, etc).

Note

the units (set_units("value",unit) where the recommended unit is g/d) must be used to make the ef dataframe

See Also

[rasterSource](#), [lineSource](#) and [emission](#)

Examples

```
veic <- vehicles(example = TRUE)

EmissionFactors <- emissionFactor(example = TRUE)

TOTAL <- totalEmission(veic,EmissionFactors,pol = c("CO","PM"))
```

totalVOC	<i>Calculate Total VOCs emissions (depreciated)</i>
----------	---

Description

Calculates Volatile Organic Compounds (COVs) emitted by the process of exhaust (through the exhaust pipe), liquid (carter and evaporative) and vapor (fuel transfer operations).

This function calculates the total emission using emission factors and then speciate into one of the available species. A better approach is to use 'speciate()' function and 'species' dataset.

Available COVs are: eth, hc3, hc5, hc8, ol2, olt, oli, iso, tol, xyl, ket, ch3oh and ald

Usage

```
totalVOC(v, ef, pol, verbose = T)
```

Arguments

v	data frame with the vehicle data
ef	emission factors
pol	pollutant name in ef
verbose	display additional information

Format

Return a list with the daily total emission by territory.

Note

The same ef can be used to totalEmission and voc

See Also

[speciation](#) and [species](#)

Examples

```
veic <- vehicles(example = TRUE)

COV = c("eth", "hc3", "hc5", "hc8", "ol2", "olt", "oli", "iso", "tol", "xyl", "ket", "ch3oh", "ald")
EF_COV <- as.data.frame(matrix(NA, ncol = 9, nrow = 8, byrow = TRUE),
                           row.names = row.names(veic))
names(EF_COV) <- c("vap_g", "vap_e", "vap_d",
                  "liq_g", "liq_e", "liq_d",
                  "exa_g", "exa_e", "exa_d")

EF_COV["vap_g"] <- c(0.230, 0.00, 0.120, 0.00, 0.00, 0.00, 0.00, 0.00)
EF_COV["vap_e"] <- c(0.000, 0.25, 0.120, 0.00, 0.00, 0.00, 0.00, 0.00)
```

```

EF_COV["vap_d"] <- c(0.000,0.00,0.000,0.00,0.00,0.00,0.00,0.00)
EF_COV["liq_g"] <- c(2.000,0.00,0.875,0.00,0.00,0.00,1.20,0.00)
EF_COV["liq_e"] <- c(0.000,1.50,0.875,0.00,0.00,0.00,0.00,1.20)
EF_COV["liq_d"] <- c(0.000,0.00,0.000,0.00,0.00,0.00,0.00,0.00)
EF_COV["exa_g"] <- c(0.425,0.00,0.217,0.00,0.00,0.00,1.08,0.00)
EF_COV["exa_e"] <- c(0.000,1.30,0.217,0.00,0.00,0.00,0.00,1.08)
EF_COV["exa_d"] <- c(0.000,0.00,0.000,2.05,0.00,0.00,0.00,0.00)

print(EF_COV)

COV_total <- totalVOC(veic,EF_COV,pol = COV[10])

```

vehicles

Tool to set-up vehicle data table

Description

Return a data frame with 4 columns (vehicle category, type, fuel and average kilometers driven) and an additional column with the number of vehicles for each interest area (cities, states, countries, etc).

Average daily kilometres driven are defined by vehicle type:

- LDV (Light duty Vehicles) 41 km / day
- TRUCKS (Trucks) 110 km / day
- BUS (Busses) 165 km / day
- MOTO (motorcycles and other vehicles) 140 km / day

The number of vehicles are defined by the distribution of vehicles by vehicle classes and the total number of vehicles by area.

Usage

```

vehicles(total_v, area_name = names(total_v), distribution, type,
        category = NA, fuel = NA, vnames = NA, example = F,
        verbose = T)

```

Arguments

total_v	total of vehicles by area (area length)
area_name	area names (area length)
distribution	distribution of vehicles by vehicle class
type	type of vehicle by vehicle class (distribution length)
category	category name (distribution length / NA)
fuel	fuel type by vehicle class (distribution length / NA)
vnames	name of each vehicle class (distribution length / NA)
example	a simple example
verbose	display additional information

Note

total_v and area_name must have the same length.

distribution, type, category (if used), fuel (if used) and vnames (if used) must have the same length.

See Also

[areaSource](#) and [totalEmission](#)

Examples

```
fleet <- vehicles(example = TRUE)

# or the code bellow for the same result
# DETRAN 2016 data for total number of vehicles for 5 Brazilian states (Sao Paulo,
# Rio de Janeiro, Minas Gerais, Parana and Santa Catarina)
# vehicle distribution of Sao Paulo

fleet <- vehicles(total_v = c(27332101, 6377484, 10277988, 7140439, 4772160),
  area_name = c("SP", "RJ", "MG", "PR", "SC"),
  distribution = c( 0.4253, 0.0320, 0.3602, 0.0260,
    0.0290, 0.0008, 0.1181, 0.0086),
  category = c("LDV_E25", "LDV_E100", "LDV_F", "TRUCKS_B5",
    "CBUS_B5", "MBUS_B5", "MOTO_E25", "MOTO_F"),
  type = c("LDV", "LDV", "LDV", "TRUCKS",
    "BUS", "BUS", "MOTO", "MOTO"),
  fuel = c("E25", "E100", "FLEX", "B5",
    "B5", "B5", "E25", "FLEX"),
  vnames = c("Light duty Vehicles Gasohol", "Light Duty Vehicles Ethanol",
    "Light Duty Vehicles Flex", "Diesel trucks", "Diesel urban busses",
    "Diesel intercity busses", "Gasohol motorcycles",
    "Flex motorcycles"))
```

Index

*Topic **datasets**

perfil, [7](#)

species, [16](#)

areaSource, [2](#), [4](#), [5](#), [23](#)

emission, [3](#), [20](#)

emissionFactor, [4](#)

gridInfo, [5](#), [7](#), [12](#), [13](#), [15](#)

lineSource, [6](#), [13](#), [20](#)

perfil, [7](#)

plumeRise, [10](#)

pointSource, [11](#)

rasterSource, [7](#), [12](#), [12](#), [15](#), [20](#)

read, [13](#), [18](#)

speciation, [15](#), [18](#), [21](#)

species, [15](#), [16](#), [21](#)

streetDist, [18](#)

totalEmission, [4](#), [5](#), [20](#), [23](#)

totalVOC, [21](#)

vehicles, [22](#)