Package 'foqat'

September 30, 2023

Type Package

Title Field Observation Quick Analysis Toolkit

Version 2.0.8.2

Author Tianshu Chen

Maintainer Tianshu Chen <tianshu129@163.com>

Description Tools for quickly processing and analyzing field observation data and air quality data. This tools contain functions that facilitate analysis in atmospheric chemistry (especially in ozone pollution). Some functions of time series are also applicable to other fields. For detail please view homepage<https://github.com/tianshu129/foqat>.
Scientific Reference:
1. The Hydroxyl Radical (OH) Reactivity: Roger Atkinson and Janet Arey (2003) <doi:10.1021/cr0206420>.
2. Ozone Formation Potential (OFP): <http://ww2.arb.ca.gov/sites/default/files/barcu/regact/2009/mir2009/mir10.pdf>, Zhang et al.(2021) <doi:10.5194/acp-21-11053-2021>.
3. Aerosol Formation Potential (AFP): Wenjing Wu et al. (2016) <doi:10.1016/j.jes.2016.03.025>.
4. TUV model: <https://www2.acom.ucar.edu/modeling/tropospheric-ultraviolet-and-visible-tuv-radiation-model>.

URL https://github.com/tianshu129/foqat, https://tianshu129.github.io/foqat/

BugReports https://github.com/tianshu129/foqat/issues

Depends R (>= 3.5.0)

Imports lubridate, magrittr, dplyr, plyr, stats, stringr, utils, lmodel2, reshape2, ggplot2, ggplotify, gridExtra, scales, rvest, xml2, ggnewscale, patchwork

License GPL-3 | file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.2.3

NeedsCompilation no

Suggests knitr, rmarkdown

VignetteBuilder knitr

Repository CRAN

Date/Publication 2023-09-30 06:10:02 UTC

R topics documented:

afp . . . 3
anylm . . . 4
aqi . . . 6
avri . . . 6
dm8n . . . 8
dm8n_batch . . . 9
dm8n_np . . . 10
fm . . . 11
geom_avri . . . 12
geom_avri_batch . . . 13
geom_psd . . . 14
geom_ts . . . 15
geom_tsw . . . 18
geom_ts_batch . . . 19
koh . . . 20
loh . . . 21
met . . . 22
nsvp . . . 23
ofp . . . 24
prop . . . 25
setup_tuv . . . 26
statdf . . . 26
svri . . . 27
transp . . . 28
trs . . . 29
tuv . . . 30
tuv_batch . . . 31
tuv_core . . . 33
voc . . . 36
vocct . . . 36

Index **38**

afp *Calculate aerosol formation potential*

Description

Calculate Aerosol Formation Potential (AFP) of VOC time series. Unit of AFP is ug/m3. Note: for Chinese VOC name, please also use English punctuation.

Usage

```
afp(
  df,
  inunit = "ppbv",
  t = 25,
  p = 101.325,
  stcd = FALSE,
  sortd = TRUE,
  chn = FALSE
)
```

Arguments

df	dataframe contains time series.
inunit	input's unit for VOC concentration. A character vector from these options: "ugm" or "ppbv". "ugm" means ug/m3. "ppbv" means part per billion volumn. The default vaule is "ppbv".
t	Temperature, in Degrees Celsius, used to convert data in micrograms per cubic meter to standard conditions (25 Degrees Celsius, 101.325 kPa). By default, t equals to 25 Degrees Celsius.
p	Pressure, in kPa, used to convert data in micrograms per cubic meter to standard conditions (25 Degrees Celsius, 101.325 kPa). By default, p equals to 101.325 kPa.
stcd	logical. Does it output results in standard conditions? The default vaule is FALSE.
sortd	logical value. It determines whether the VOC species are sorted or not. By default, sortd has value "TRUE". If TRUE, VOC species in time series will be arranged according to VOC group, relative molecular weight, and SOAY value.
chn	logical. Dose colnames present as Chinese? The default vaule is FALSE.

Details

The CAS number is matched for each VOC speices (from column name), and the average SOA yield (SOAY) is matched through the CAS number and used for time series calculation.
The average SOAY comes from "W. Wu, B. Zhao, S. Wang, J. Hao, Ozone and secondary organic aerosol formation potential from anthropogenic volatile organic compounds emissions in China. J Environ Sci. 53, 224–237 (2017)". Note: If input VOC species contain M,P-xylene, it will be automatically divided into m-xylene and P-xylene evenly.

Value

a list contains 5 tables: SOAY_Result: matched SOAY value result; AFP_Result: AFP time series of VOC by species; AFP_Result_mean: the average value and proportion of AFP of VOC by species (sorted from large to small); AFP_Result_group: AFP time series of VOC classified by groups; AFP_Result_group_mean: the average value and proportion of AFP of VOC according to major groups (sorted from large to small).

`anylm` *Analysis of linear regression for time series in batch*

Description

Analyse linear regression for time series in batch

Usage

```
anylm(
  df,
  xd = 2,
  yd = 3,
  zd = NULL,
  td = NULL,
  mi = 1,
  range.y = "interval",
  range.x = "interval",
  nperm = 99,
  showpage = TRUE,
  scint = FALSE,
  dign = 1,
  zfill = "lightgray",
  ppsize = 2,
  showinfo = TRUE,
  ptsize = 12,
  pncol = NULL
)
```

Arguments

`df`	dataframe of time series.
`xd`	species or columns for x axis, vector of number or colnames. Default vaule is '2'.
`yd`	species or columns for y axis, vector of number or colnames. Default vaule is '3'.
`zd`	species or columns to fill points, vector of number or colnames. Default vaule is 'NULL'. If zd is setted, labels for scaled color represent Percentile value (0, 0.25, 0.5,0.75, 1).

td	1 column to group data, number or colname. Default vaule is 'NULL'.
mi	index (1~4) of methods: 1. ordinary least squares (OLS); 2. major axis (MA); 3. standard major axis (SMA); 4. and ranged major axis (RMA). Refered from R package 'lmodel2'. Default vaule is '1'.
range.y	Parametres for ranged major axis regression (RMA). If range.y = NULL and range.x = NULL, RMA will not be computed. If only one of them is NULL, the program will stop. If range.y = "relative": variable y has a true zero (relative-scale variable). If range.y = "interval": variable y possibly includes negative values (interval-scale variable). If range.x = "relative": variable x has a true zero (relative-scale variable). If range.x = "interval": variable x possibly includes negative values (interval-scale variable). Refered from R package 'lmodel2'.
range.x	Parametres, please see 'range.y'.
nperm	Number of permutations for the tests. If nperm = 0, tests will not be computed. Refered from R package 'lmodel2'.
showpage	logical value for showing all plots. If TRUE, print all plot in 1 page. Default vaule is 'TRUE'.
scint	logical value for displaying scientific notion in legend and plot title. Default vaule is 'FALSE'.
dign	numeric value for digists in legend and plot title. Default vaule is '1'.
zfill	color for points, only valid when zd is NULL. Default vaule is "lightgray".
ppsize	size for points. Default vaule is "lightgray".
showinfo	logical value for displaying regression information in plot title. Default vaule is 'TRUE'.
ptsize	font size for plot title. Default vaule is '12'.
pncol	number of columns for plots in page. Refered from R package 'gridExtra'. Default vaule is 'NULL'.

Details

X axis, Y axis, scaled color for points are flexible for multiple columns. Data could also be grouped according to 1 column.

Value

a list contains: data_list, lm_df, lm_list, plot_list, all_plot.
data_list: a list contains data for linear regression.
lm_df: a dataframe for key results of linear regression. row index of lm_df corresponds to 'id' of plot in 'all_plot'.
lm_list: a list contains detail results of linear regression.
plot_list: a list contains plots for linear regression.
all_plot: a page for all plots in 'plot_list'.
To see page, please use this function: 'gridExtra::grid.arrange(grobs=...)'.
To see page, please use this 2-lines function:
'g=gridExtra::arrangeGrob(grobs=...)',

'ggplot2::ggsave(filename = "example.jpg", plot =g)'.
'id' of plot corresponds to row index of 'lm_df'.

Examples

```
anylm(aqi, xd=c(2,3), yd=6, zd=4, td=NULL, dign=3)
```

`aqi` *Demo data of air quality*

Description

5 days air quality data (1 min resolution) includes: NO, NO2, CO, SO2, O3. The variables are as follows:

Usage

```
aqi
```

Format

A data frame with 7140 rows and 6 variables:

Time Time for data

NO Nitric oxide (NO)

NO2 Nitrogen Dioxide (NO2)

CO Carbon monoxide (CO)

SO2 Sulfur dioxide (SO2)

O3 Ozone (O3)

`avri` *Calculate average of variation*

Description

Calculates average of variation of time series. (contain but not limited to: average daily variation, average monthly variation, average annual variation)

Usage

```
avri(
  df,
  bkip = NULL,
  mode = "recipes",
  value = "day",
  st = NULL,
  et = NULL,
  na.rm = TRUE,
  digits = 2,
  wind = FALSE,
  coliws = 2,
  coliwd = 3,
  sn = FALSE
)
```

Arguments

df	dataframe of time series.
bkip	the basic time reslution for average variation, such as '1 hour'. If mode "custom" is selected, do not need to enter bkip.
mode	for calculating cycles: "recipes", "ncycle", "custom". "recipes" means using internal setting for calculation. "ncycle" means setting number of items for per cycle. "custom" means using 1 column in dataframe as a list of grouping elements for calculation.
value	for detail setting of mode. Possible values for "recipes" are "day", "week", "month", year". "day" equals to 24 (hours) values in 1 day. "week" equals to 7 (days) values in 1 week. "month" equals to 31 (days) values in 1 month. "year" equals to 12 (months) values in 1 year. values for "ncycle" is a number representing number of items in per cycle. values for "custom" is a number representing column index in dataframe.
st	start time of resampling. The default value is the fisrt value of datetime column.
et	end time of resampling. The default value is the last value of datetime column.
na.rm	logical value. Remove NA value or not?
digits	numeric value, digits for result dataframe.
wind	logical value. if TRUE, please set coliwd, coliws.
coliws	numeric value, column index of wind speed in dataframe.
coliwd	numeric value, column index of wind direction (degree) in dataframe.
sn	logical value. if TRUE, the results will be presented by scientific notation (string).

Details

If you have wind data (wind speed, and wind direction in dgree), please set 'wind' as 'TRUE', and set values for 'coliwd' and 'coliws'.

Value

a data frame which contains both the average variations and the standard deviations. Note that when the pattern USES "ncycle" or "custom", the start time determines the start time of the first element in the average variation. For example, if the first timestamp of data is "2010-05-01 12:00:00", the resolution is 1 hour, the mode is "ncycle", and the value is 24, then the result represents diurnal variation starting from 12 o'clock.

Examples

```
avri(met, bkip = "1 hour", mode = "recipes", value = "day",
st = "2017-05-01 00:00:00", wind = TRUE, coliws = 4, coliwd = 5)
```

`dm8n` *Calculate daily maximum-8-hour ozone*

Description

Calculates daily maximum-8-hour ozone from ozone observation data.

Usage

```
dm8n(
  df,
  colid = 1,
  colio = 2,
  starthour = 0,
  endhour = 16,
  nh = 6,
  nc = 14,
  na.rm = TRUE,
  outputmode = 1,
  unitlb = NA
)
```

Arguments

`df`	dataframe of time series for ozone and other related parameters.
`colid`	column index for date-time. By default, it equals to 1.
`colio`	column index for ozone. By default, it equals to 2.
`starthour`	numeric, start hour for calculating 8-hour ozone. By default, it equals to 0.
`endhour`	numeric, end hour for calculating 8-hour ozone. By default, it equals to 16 which means averaging ozone between 16~23.
`nh`	numeric. The number of effective hourly concentrations per 8-hour period.
`nc`	numeric. The number of effective 8-hour average concentrations per day.

na.rm	logical. Should missing values (including NaN) be omitted from the calculations?
outputmode	numeric, the format of the output, possible value: 1 or 2. See 'value' for the results of filling in 1 or 2.
unitlb	labels for y axis of dma8 plot. By default, it equals to NA. If set this parameter, the order of species should same as that in the dataframe.

Details

This function can calculate daily maximum-8-hour ozone and other parameters corresponding to it.

Value

a dataframe depends on the value of 'outputMode'. Value 1 will output 1 list which incudes 1 table (maximum-8-hour ozone) and 1 plot (dma8 plot). Value 2 will output 1 list which contains 4 tables (8-hour ozone, statistics of the number of effective hourly concentrations in each 8-hour average concentration, statistics of the number of effective 8-hour average concentrations in each day, maximum-8-hour ozone) and 1 plot (dma8 plot).

Examples

```
## Not run:
dm8n(aqi,colio=6,unitlb=c("NO (ppbv)", "NO2 (ppbv)", "CO (ppbv)", "SO2 (ppbv)", "O3 (ppbv)"))

## End(Not run)
```

---

dm8n_batch *Calculate daily maximum-8-hour ozone in batch*

Description

Calculates daily maximum-8-hour ozone in batch

Usage

```
dm8n_batch(
  df,
  starthour = 0,
  endhour = 16,
  nh = 6,
  nc = 14,
  na.rm = TRUE,
  outputmode = 1
)
```

Arguments

df	dataframe of time series for ozone and other related parameters.
starthour	numeric, start hour for calculating 8-hour ozone. By default, it equals to 0.
endhour	numeric, end hour for calculating 8-hour ozone. By default, it equals to 16 which means averaging ozone between 16~23.
nh	numeric. The number of effective hourly concentrations per 8-hour period.
nc	numeric. The number of effective 8-hour average concentrations per day.
na.rm	logical. Should missing values (including NaN) be omitted from the calculations?
outputmode	numeric, the format of the output, possible value: 1 or 2. See 'value' for the results of filling in 1 or 2.

Details

This function can calculate daily maximum-8-hour ozone in batch.

Value

a dataframe depends on the value of 'outputMode'. Value 1 will output 1 list which incudes 1 table (maximum-8-hour ozone). Value 2 will output 1 list which contains 4 tables (8-hour ozone, statistics of the number of effective hourly concentrations in each 8-hour average concentration, statistics of the number of effective 8-hour average concentrations in each day, maximum-8-hour ozone).

---

dm8n_np *Calculate daily maximum-8-hour ozone without printing plot*

Description

Calculates daily maximum-8-hour ozone from ozone observation data without printing plot

Usage

```
dm8n_np(
  df,
  colid = 1,
  colio = 2,
  starthour = 0,
  endhour = 16,
  nh = 6,
  nc = 14,
  na.rm = TRUE,
  outputmode = 1
)
```

Arguments

df	dataframe of time series for ozone and other related parameters.
colid	column index for date-time. By default, it equals to 1.
colio	column index for ozone. By default, it equals to 2.
starthour	numeric, start hour for calculating 8-hour ozone. By default, it equals to 0.
endhour	numeric, end hour for calculating 8-hour ozone. By default, it equals to 16 which means averaging ozone between 16~23.
nh	numeric. The number of effective hourly concentrations per 8-hour period.
nc	numeric. The number of effective 8-hour average concentrations per day.
na.rm	logical. Should missing values (including NaN) be omitted from the calculations?
outputmode	numeric, the format of the output, possible value: 1 or 2. See 'value' for the results of filling in 1 or 2.

Details

This function can calculate daily maximum-8-hour ozone and other parameters corresponding to it.

Value

a dataframe depends on the value of 'outputMode'. Value 1 will output 1 list which incudes 1 table (maximum-8-hour ozone). Value 2 will output 1 list which contains 4 tables (8-hour ozone, statistics of the number of effective hourly concentrations in each 8-hour average concentration, statistics of the number of effective 8-hour average concentrations in each day, maximum-8-hour ozone).

Examples

```
dm8n_np(aqi,colio=6)
```

---

fm *format the theme of plot*

Description

Format the theme of plot.

Usage

```
fm(p, fsz = 13, lsz = 0.5, tkl = 0.2)
```

Arguments

p	a ggplot-format plot.
fsz	font size in plot.
lsz	line size of panel border and axis in plot.
tkl	tick length in plot.

Value

a plot with a new theme.

---

geom_avri *Plot the average variation*

Description

Easy way to plot the average variation.

Usage

```
geom_avri(
  df,
  cave = 2,
  csd = 3,
  ssd = 1,
  alpha = 0.5,
  xlab = NULL,
  ylab = NULL,
  lcc = NULL,
  lsize = 1,
  rff = NULL
)
```

Arguments

df	dataframe contains average variation value and their standard deviation.
cave	column index of average variation. The default vaule is 2.
csd	column index of standard deviation. The default vaule is 3.
ssd	scale value for standard deviation. The default vaule is 1.
alpha	the alpha value of ribbon. The default vaule is 0.5.
xlab	text expression of x axis label. The default vaule is NULL.
ylab	text expression of y axis label. The default vaule is NULL.
lcc	color of line. The default vaule is NULL.
lsize	size of line. The default vaule is NULL.The default vaule is 1.
rff	fill color of ribbon. The default vaule is NULL.

Examples

```
## Not run:
x=avri(aqi, bkip = "1 hour", mode = "recipes", value
 = "day", st = "2017-05-01 00:00:00")
geom_avri(x,cave=6, csd=11, alpha=0.5, lcc="#0050b3",
 rff="#40a9ff", xlab="Time",ylab=bquote(O[3]~" "~(ppbv)))

## End(Not run)
```

`geom_avri_batch` *Plot the average variation in batch*

Description

Easy way to plot the average variation in batch.

Usage

```
geom_avri_batch(
  df,
  ssd = 1,
  alpha = 0.5,
  xlab = NULL,
  ylab = NULL,
  lcc = NULL,
  lsize = 1,
  rff = NULL,
  ncol = 2,
  bquote = FALSE
)
```

Arguments

`df`	dataframe contains average variation value and their standard deviation.
`ssd`	scale value for standard deviation. The default vaule is 1.
`alpha`	the alpha value of ribbon. The default vaule is 0.5.
`xlab`	text expression of x axis label. The default vaule is NULL. Need to set with ylab at the same time.
`ylab`	text expression of y axis label. The default vaule is NULL. Need to set with xlab at the same time.
`lcc`	colors of lines. The default vaule is NULL. Need to set with rff at the same time.
`lsize`	size of lines. The default vaule is NULL.The default vaule is 1.
`rff`	fill colors of ribbons. The default vaule is NULL. Need to set with lcc at the same time.
`ncol`	number of figure columns in final plot layout. The default vaule is 2.
`bquote`	logical value. Set to TRUE if you want to use bquote in labs (xlab and y lab). The default vaule is FALSE.

Examples

```
## Not run:
#example 1
x=avri(aqi, bkip = "1 hour", mode = "recipes", value
 = "day", st = "2017-05-01 00:00:00")
geom_avri_batch(x)
#example 2
x=avri(aqi, bkip = "1 hour", mode = "recipes", value
 = "day", st = "2017-05-01 00:00:00")
lcc=c("#f5222d","#fa8c16","#52c41a","#1890ff","#722ed1")
rff=c("#ff7875","#ffc069","#95de64","#69c0ff","#b37feb")
xlab1=list(bquote(Time~""),bquote(Time~""),bquote(Time~""),
bquote(Time~""),bquote(Time~""))
ylab1=list(bquote(NO~" "~(ppbv)), bquote(NO[2]~" "~(ppbv)),
bquote(CO~" "~(ppmv)), bquote(SO[2]~" "~(ppbv)), bquote(O[3]~" "~(ppbv)))
geom_avri_batch(x, alpha=0.6, xlab=xlab1, ylab=ylab1,
 lcc=lcc, rff=rff, bquote=TRUE)
#example 3
x=avri(aqi, bkip = "1 hour", mode = "recipes", value
 = "day", st = "2017-05-01 00:00:00")
xlab2=rep("Time",5)
ylab2=c("NO","NO2","CO","SO2","O3")
geom_avri_batch(x, alpha=0.6, xlab=xlab2, ylab=ylab2,
 lcc=lcc, rff=rff, bquote=FALSE)

## End(Not run)
```

---

geom_psd *Plot the time series of particle size distribution.*

Description

Plot the time series of particle size distribution.

Usage

```
geom_psd(
  df,
  labxyl = NULL,
  logy = TRUE,
  ybk = NULL,
  nlmt = NULL,
  csbk = pretty_breaks(4),
  trans = "identity",
  colsz = 1,
  fsz = 13,
  lsz = 0.4,
  tkl = 0.2
)
```

Arguments

df	dataframe of particle size data: the first column of input is datetime; the other columns are number concentration (N, unit: #/cm3) or log number concentration (dN/dlogdp, unit: #/cm3) for each particle size channel. Column names of the other columns are the middle particle size for each particle size channel.
labxyl	vector, Set the title of x axis, y axis, legend. The default vaule is NULL. Bquote grammer is accepted.
logy	logical. Plot the data with log y axis. The default vaule is TRUE.
ybk	numeric vector, breaks of y axis.
nlmt	numeric value, range of particle number for colorscales of plot.
csbk	numeric vector, breaks of color bar.
trans	character string, "identity" or "log10". transformation of color bar breaks.
colsz	numeric value, size of columns in plot.
fsz	font size in plot.
lsz	line size of panel border and axis in plot.
tkl	tick length in plot.

Value

a plot for the time series of particle size distribution.

Examples

```
## Not run:
dn_table = read.delim(system.file("extdata", "smps.txt", package = "foqat"),
check.names = FALSE)
dn1_table=dn_table[,c(1,5:148)]
dn1_table[,1]=as.POSIXct(dn1_table[,1], format="%m/%d/%Y %H:%M:%S", tz="GMT")
geom_psd(dn1_table,fsz=10)

## End(Not run)
```

---

geom_ts *Plot time series*

Description

Easy way to plot time series.

Usage

```
geom_ts(
  df,
  yl = NULL,
  yr = NULL,
  yllab = NULL,
  yrlab = NULL,
  xlab = NULL,
  llist = NULL,
  plist = NULL,
  alist = NULL,
  blist = NULL,
  llab = NULL,
  plab = NULL,
  alab = NULL,
  blab = NULL,
  ltype = NULL,
  pshape = NULL,
  lsize = 1,
  psize = 1,
  lcc = NULL,
  pcc = NULL,
  aff = NULL,
  bff = NULL,
  ana = TRUE,
  apos = "stack",
  bna = TRUE,
  bpos = "identity",
  yl_limit = NULL,
  yr_limit = NULL,
  yl_breaks = waiver(),
  yr_breaks = waiver(),
  yl_minor_breaks = waiver()
)
```

Arguments

df	dataframe contains time series.
yl	vector, col index of species to be putted in the left y axis.
yr	vector, col index of species to be putted in the right y axis. The default vaule is NULL.
yllab	text expression of left y axis label. The default vaule is NULL.
yrlab	text expression of right y axis label. The default vaule is NULL.
xlab	text expression of x axis label. The default vaule is NULL.
llist	vector, col index of species to be ploted by line.The default vaule is NULL.
plist	vector, col index of species to be ploted by points.The default vaule is NULL.

alist	plist vector, col index of species to be ploted by areas. The default vaule is NULL.
blist	plist vector, col index of species to be ploted by bars. The default vaule is NULL.
llab	list of text expressions of legend labels of lines. The default vaule is NULL.
plab	list of text expressions of legend labels of points. The default vaule is NULL.
alab	list of text expressions of legend labels of areas. The default vaule is NULL.
blab	list of text expressions of legend labels of bars. The default vaule is NULL.
ltype	vector, type of lines. The default vaule is NULL.
pshape	vector, shape of points. The default vaule is NULL.
lsize	vector, size of lines. The default vaule is NULL.The default vaule is 1.
psize	vector, size of points. The default vaule is NULL.The default vaule is 1.
lcc	vector, colors of lines. The default vaule is NULL. The default vaule is NULL.
pcc	vector, colors of points. The default vaule is NULL. The default vaule is NULL.
aff	fill color of areas. The default vaule is NULL.
bff	fill color of bars. The default vaule is NULL.
ana	logical value, the way to handle NA values for areas. If you select FALSE, NA value will be replaced by 0.
apos	Position adjustment for areas, either as a string, or the result of a call to a position adjustment function.
bna	logical value, the way to handle NA values for bars. If you select FALSE, NA value will be replaced by 0.
bpos	Position adjustment for bars, either as a string, or the result of a call to a position adjustment function.
yl_limit	two numeric values, specifying the lower limit and the upper limit of the scale in left y axis.
yr_limit	two numeric values, specifying the lower limit and the upper limit of the scale in right y axis.
yl_breaks	a numeric vector of positions for breaks in left y axis.
yr_breaks	a numeric vector of positions for breaks in right y axis.
yl_minor_breaks	a numeric vector of positions for minor breaks in left y axis.

Examples

```
## Not run:
aqi2=aqi
aqi2$NO[aqi2$NO>7]=NA
aqi2$NO2=aqi2$NO2*0.3
geom_ts(
     df=aqi2,
     yl=c(3,2),
     yr=6,
     alist=c(3,2),
```

```
     llist=6,
    alab=list(bquote(NO[2]~" "), bquote(NO~" ")),
llab=list(bquote(O[3]~" ")),
     yllab=bquote(NO[x]~" "~(ppbv)),
     yrlab=bquote(O[3]~" "~(ppbv)),
     lcc="#ff4d4f",
     aff=c("#096dd9","#69c0ff"),
xlab="Datetime")

## End(Not run)
```

---

geom_tsw *Plot time series*

Description

Easy way to plot time series.

Usage

```
geom_tsw(
  df,
  coliws = 2,
  coliwd = 3,
  lsize = 0.8,
  psize = NA,
  msize = 8,
  mlabel = "West wind",
  mx = 0.05,
  my = -0.1,
  mwd = 270
)
```

Arguments

df	dataframe contains time series.
coliws	column index of wind speed. The default vaule is 2.
coliwd	column index of wind direction. The default vaule is 3.
lsize	size of line (wind speed). The default vaule is 0.8.
psize	size of point (wind speed). The default vaule is NA.
msize	size of mark (wind direction). The default vaule is 8.
mlabel	label of mark (wind direction). The default vaule is "West wind".
mx	adjust value for the x position of mark (wind direction). The default vaule is 0.05.
my	adjust value for the y position of mark (wind direction). The default vaule is -0.1.
mwd	direction of mark (wind direction). The default vaule is 270.

Examples

```
## Not run:
metds=trs(met, bkip="15 mins")
geom_tsw(metds, coliws=4, coliwd=5)

## End(Not run)
```

`geom_ts_batch` *Plot time series in batch*

Description

Easy way to plot time series in batch.

Usage

```
geom_ts_batch(
  df,
  xlab = NULL,
  ylab = NULL,
  cclist = NULL,
  bquote = FALSE,
  breaks = waiver(),
  date_breaks = waiver(),
  labels = waiver(),
  date_labels = waiver(),
  minor_breaks = waiver(),
  date_minor_breaks = waiver(),
  expand = c(0, 0),
  panelgap = 1
)
```

Arguments

`df`	dataframe of time series.
`xlab`	text expression of x axis label. The default vaule is NULL.
`ylab`	text expression of y axis label. The default vaule is NULL.
`cclist`	vector, colors of lines. The default vaule is NULL.
`bquote`	logical value. Set to TRUE if you want to use bquote in labs (xlab and y lab). The default vaule is FALSE.
`breaks`	One of: - ‘NULL‘ for no breaks - ‘waiver()‘ for the breaks specified by ‘date_breaks‘ - A ‘Date‘/‘POSIXct‘ vector giving positions of breaks - A function that takes the limits as input and returns breaks as output
`date_breaks`	A string giving the distance between breaks like "2 weeks", or "10 years". If both ‘breaks‘ and ‘date_breaks‘ are specified, ‘date_breaks‘ wins.

labels	One of: - 'NULL' no labels - 'waiver()' for the default labels computed by the transformation object - A character vector giving labels (must be same length as 'breaks') - A function that takes the breaks as input and returns labels as output. Also accepts rlang lambda function notation.
date_labels	A string giving the formatting specification for the labels. Codes are defined in [strftime()]. If both 'labels' and 'date_labels' are specified, 'date_labels' wins.
minor_breaks	One of: - 'NULL' for no breaks - 'waiver()' for the breaks specified by 'date_minor_breaks' - A 'Date'/'POSIXct' vector giving positions of minor breaks - A function that takes the limits as input and returns minor breaks as output
date_minor_breaks	A string giving the distance between minor breaks like "2 weeks", or "10 years". If both 'minor_breaks' and 'date_minor_breaks' are specified, 'date_minor_breaks' wins.
expand	For position scales, a vector of range expansion constants used to add some padding around the data to ensure that they are placed some distance away from the axes. Use the convenience function expansion() to generate the values for the expand argument. The defaults are to expand the scale by 5 side for discrete variables.
panelgap	gap of panels. The default vaule is 0.5.

Examples

```
## Not run:
#example 1
geom_ts_batch(aqi)
#example 2
xlab1="Time"
ylab1=c("NO","NO2","CO","SO2","O3")
geom_ts_batch(aqi, xlab=xlab1, ylab=ylab1)
#example 3
xlab2=bquote(Time~"")
ylab2=list(bquote(NO~" "~(ppbv)), bquote(NO[2]~"
 "~(ppbv)), bquote(CO~" "~(ppmv)), bquote(SO[2]~"
 "~(ppbv)), bquote(O[3]~" "~(ppbv)))
cclist=c("#eb2f96", "#1890ff", "#52c41a", "#faad14", "#f5222d")
geom_ts_batch(aqi, xlab=xlab2, ylab=ylab2, cclist=cclist, bquote=TRUE)

## End(Not run)
```

---

koh *get kOH*

Description

Searches kOH value from 'chemspider.com'.

Usage

```
koh(spec)
```

Arguments

spec	chemical specise to be searched. chemical specise's name or CAS Number is acceptable.

Details

Theoretical values of the species' OH reaction constant kOH at 25 degrees were obtained from 'Chemspider.com'. Value source: US Environmental Protection Agency's EPISuite.
Unit is cm3/molecule-sec.
Condition is 25 deg C.

Value

the theoretical value of the species' OH reaction constant kOH at 25 degrees.

---

loh *Calculate OH reactivity*

Description

Calculate OH reactivity of VOC time series in 25 degree celsius. Note: for Chinese VOC name, please also use English punctuation.

Usage

```
loh(
  df,
  unit = "ppbv",
  t = 25,
  p = 101.325,
  stcd = FALSE,
  sortd = TRUE,
  atk = TRUE,
  chn = FALSE,
  bvoc = TRUE
)
```

Arguments

df	dataframe contains time series.
unit	unit for VOC concentration. A character vector from these options: "ugm" or "ppbv". "ugm" means ug/m3. "ppbv" means part per billion volumn.

t	Temperature, in Degrees Celsius, used to convert data in micrograms per cubic meter to standard conditions (25 Degrees Celsius, 101.325 kPa). By default, t equals to 25 Degrees Celsius.
p	Pressure, in kPa, used to convert data in micrograms per cubic meter to standard conditions (25 Degrees Celsius, 101.325 kPa). By default, p equals to 101.325 kPa.
stcd	logical. Does it output the concentration in standard condition? The default vaule is FALSE.
sortd	logical value. It determines whether the VOC species are sorted or not. By default, sortd has value "TRUE". If TRUE, VOC species in time series will be arranged according to VOC group, relative molecular weight, and OH Rate Constant.
atk	logical. use kOH value from atk or not? If not, kOH comes from 'AopWin v1.92' will be used. The default vaule is TRUE.
chn	logical. Dose colnames present as Chinese? The default vaule is FALSE.
bvoc	logical. Whether you want to list BVOC as a separate VOC group? The default vaule is TRUE.

Details

The CAS number is matched for each VOC speices (from column name), and the OH Rate Constant is matched through the CAS number and used for time series calculation.
The OH Rate Constant comes from 'AopWin v1.92' in 25 degree celsius.

Value

a list contains 5 tables: KOH_Result: the matched KOH value results; LOH_Result: the LOH time series of VOC by species; LOH_Result_stat: the statistics of LOH of VOC by species; LOH_Result_group: the LOH time series of VOC classified by groups; LOH_Result_group_mean: the statistics of LOH of VOC according to major groups.

Examples

```
voc_loh=loh(voc)
summary(voc_loh)
```

---

met *Demo data of meteorology*

Description

5 days meteorology data (5 mins resolution) includes: Temperature, Humidity, Wind speed, Wind direction. The variables are as follows:

Usage

```
met
```

Format

A data frame with 1287 rows and 5 variables:

Time Time for data

TEM Temperature

HUM Humidity

WS Wind speed

WD Wind direction

`nsvp` *Calculate Surface Area, Volume, Mass of particle by particle number concentration*

Description

Calculate Surface Area, Volume, Mass of particle by particle number concentration.

Usage

```
nsvp(df, dlogdp = FALSE, dsty = 1)
```

Arguments

`df`	dataframe of particle size data: the first column of input is datetime; the other columns are number concentration (N, unit: #/cm3) or log number concentration (dN/dlogdp, unit: #/cm3) for each particle size channel. Column names of the other columns are the middle particle size for each particle size channel.
`dlogdp`	logical value, TRUE if the third column is log number concentration (dN/dlogdp).
`dsty`	numeric value, density of particle namtter.

Value

a list with 2 dataframe. The first dataframe is a time series for Surface Area (unit: μm2/cm3), Volume (unit: μm3/cm3), Mass (unit: μg/m3) of each channels; the second dataframe is a time series for total Surface Area, Volume, Mass of all channels.

ofp *Calculate ozone formation potential*

Description

Calculate Ozone Formation Potential (OFP) of VOC time series. Note: for Chinese VOC name, please also use English punctuation.

Usage

```
ofp(
  df,
  inunit = "ppbv",
  outunit = "ppbv",
  t = 25,
  p = 101.325,
  stcd = FALSE,
  sortd = TRUE,
  chn = FALSE,
  mtype = "usa",
  bvoc = TRUE
)
```

Arguments

df	dataframe contains time series.
inunit	input's unit for VOC concentration. A character vector from these options: "ugm" or "ppbv". "ugm" means ug/m3. "ppbv" means part per billion volumn. The default vaule is "ppbv".
outunit	output's unit for VOC concentration. A character from these options: "ugm" or "ppbv". "ugm" means ug/m3. "ppbv" means part per billion volumn. The default vaule is "ppbv".
t	Temperature, in Degrees Celsius, used to convert data in micrograms per cubic meter to standard conditions (25 Degrees Celsius, 101.325 kPa). By default, t equals to 25 Degrees Celsius.
p	Pressure, in kPa, used to convert data in micrograms per cubic meter to standard conditions (25 Degrees Celsius, 101.325 kPa). By default, p equals to 101.325 kPa.
stcd	logical. Does it output results in standard conditions? The default vaule is FALSE.
sortd	logical value. It determines whether the VOC species are sorted or not. By default, sortd has value "TRUE". If TRUE, VOC species in time series will be arranged according to VOC group, relative molecular weight, and MIR value.
chn	logical. Dose colnames present as Chinese? The default vaule is FALSE.
mtype	text. "usa" for MIR value from USA, "chn" for MIR value from CHINA.

bvoc	logical. Whether you want to list BVOC as a separate VOC group? The default vaule is TRUE.

Details

The CAS number is matched for each VOC speices (from column name), and the Maximum Incremental Reactivity (MIR) value is matched through the CAS number and used for time series calculation.
The MIR value comes from <https://ww2.arb.ca.gov/sites/default/files/classic/regact/2009/mir2009/mir10.pdf>, Zhang et al.(2021) <doi:10.5194/acp-21-11053-2021>.

Value

a list contains 5 tables: MIR_Result: the matched MIR value results; OFP_Result: the OFP time series of VOC by species; OFP_Result_stat: the statistics of OFP of VOC by species; OFP_Result_group: OFP the time series of VOC classified by groups; OFP_Result_group_stat: the statistics of OFP of VOC according to major groups.

Examples

```
voc_ofp=ofp(voc)
summary(voc_ofp)
```

---

prop *Convert time series into proportion time series*

Description

Convert time series into proportion time series.

Usage

```
prop(df, cmcase = FALSE)
```

Arguments

df	dataframe of time series.
cmcase	logical value. Set to TRUE if you only want to retain cases which are complete, i.e., have no missing values. The default vaule is FALSE.

Value

a dataframe with proportion time series.

Examples

```
prop(voc)
```

setup_tuv *Demo data of setup for tuv*

Description

5 days setup data for tuv includes: nt, lat, lon, o3col. The variables are as follows:

Usage

```
setup_tuv
```

Format

A data frame with 5 rows and 5 variables:

date date for each day

nt data point for each day

lat lat for each day

lon lon for each day

o3col o3 column concentration for each day

statdf *Summary of dataframe*

Description

Summary of dataframe.

Usage

```
statdf(df, n = 2, cmcase = FALSE, prop = FALSE)
```

Arguments

df	dataframe of time series.
n	digits for reuslt in dataframe.
cmcase	logical value. Set to TRUE if you only want to summary cases which are complete, i.e., have no missing values.
prop	logical value. Convert time series into proportion time series before summary.

Details

Summary of dataframe: mean, standard deviation (sd), minimum (min), percentiles (0.25, 0.50, 0.75), maximum (max).

Value

a dataframe, columns stands for parameters, rows stands for variables.

Examples

```
statdf(voc)
```

`svri` *Compute the variation of summary statistics*

Description

Compute the variation of summary statistics for time series.

Usage

```
svri(
  df,
  bkip = NULL,
  mode = "recipes",
  value = "day",
  st = NULL,
  et = NULL,
  fun = "mean",
  probs = 0.5,
  na.rm = TRUE,
  digits = 2,
  wind = FALSE,
  coliws = 2,
  coliwd = 3,
  sn = FALSE
)
```

Arguments

`df`	dataframe of time series.
`bkip`	the basic time reslution for variation, such as '1 hour'. If mode "custom" is selected, do not need to enter bkip.
`mode`	for calculating cycles: "recipes", "ncycle", "custom". "recipes" means using internal setting for calculation. "ncycle" means setting number of items for per cycle. "custom" means using 1 column in dataframe as a list of grouping elements for calculation.
`value`	for detail setting of mode. Possible values for "recipes" are "day", "week", "month", year". "day" equals to 24 (hours) values in 1 day. "week" equals to 7 (days) values in 1 week. "month" equals to 31 (days) values in 1 month.

	"year" equals to 12 (months) values in 1 year. values for "ncycle" is a number representing number of items in per cycle. values for "custom" is a number representing column index in dataframe.
st	start time of resampling. The default value is the fisrt value of datetime column.
et	end time of resampling. The default value is the last value of datetime column.
fun	a function to compute the summary statistics which can be applied to all data subsets: 'sum', 'mean', 'median', 'min', 'max', 'sd' and 'quantile'.
probs	numeric vector of probabilities with values in \([0,1]\).
na.rm	logical value. Remove NA value or not?
digits	numeric value, digits for result dataframe.
wind	logical value. if TRUE, please set coliwd, coliws.
coliws	numeric value, column index of wind speed in dataframe.
coliwd	numeric value, column index of wind direction (degree) in dataframe.
sn	logical value. if TRUE, the results will be presented by scientific notation (string).

Details

If you have wind data (wind speed, and wind direction in dgree), please set 'wind' as 'TRUE', and set values for 'coliwd' and 'coliws'.

Value

the variation of summary statistics Note that when the pattern USES "ncycle" or "custom", the start time determines the start time of the first element in the average variation. For example, if the first timestamp of data is "2010-05-01 12:00:00", the resolution is 1 hour, the mode is "ncycle", and the value is 24, then the result represents diurnal variation starting from 12 o'clock.

Examples

```
svri(met, bkip = "1 hour", mode = "recipes", value = "day", fun = 'quantile', probs=0.5,
st = "2017-05-01 00:00:00")
```

---

transp *Convert the format of particle size data*

Description

Converting the format of particle size data. There are 2 types of particle size data: table and list. For table format: the first column of input is datetime; the other column is the number concentration of each particle size channel, column name is the middle particle size of the particle size channel. For list format: the first column of input is datetime. The second column of input is for middle ranges of channels. The third column of input is for particle number concentration of each channel at each timepoint.

Usage

```
transp(df)
```

Arguments

df	dataframe of particle size data: a table or a list.

Value

a dataframe. If the input is a table, the output is a list, and if the input is a list, the output is a table.

---

trs *Resample time series by summary statistics*

Description

Resamples time series by summary statistics, and returns complete time series with new time resolution. (wind data is acceptable)

Usage

```
trs(
  df,
  bkip,
  st = NULL,
  et = NULL,
  fun = "mean",
  probs = 0.5,
  na.rm = TRUE,
  wind = FALSE,
  coliws = 2,
  coliwd = 3,
  cpms = TRUE
)
```

Arguments

df	dataframe of time series.
bkip	new resolution breaking input of time series, such as '1 hour'.
st	start time of resampling. The default value is the fisrt value of datetime column.
et	end time of resampling. The default value is the last value of datetime column.
fun	a function to compute the summary statistics which can be applied to all data subsets: 'sum', 'mean', 'median', 'min', 'max', 'sd' and 'quantile'.
probs	numeric vector of probabilities with values in \([0,1]\).
na.rm	logical value. Remove NA value or not?

wind	logical value. if TRUE, please set coliwd, coliws.
coliws	numeric value, column index of wind speed in dataframe.
coliwd	numeric value, column index of wind direction (degree) in dataframe.
cpms	logical value. Compensate the insufficient amount of the millisecond bit for datetime column.

Details

If you have wind data (wind speed, and wind direction in dgree), please set 'wind' as 'TRUE', and set values for 'coliwd' and 'coliws'.

Value

a dataframe which contains a time series for summary statistics with a new time resolution.

Examples

```
trs(met, bkip = "1 hour", st = "2017-05-01 00:00:00", wind = TRUE, coliws = 4, coliwd = 5)
```

---

tuv *Calculate TUV in batch*

Description

This function runs TUV in batch by reading the time series for the parameters to be entered, and summarizes the results to the new dataframe.

Usage

```
tuv(pathtuv, df, colid = 1)
```

Arguments

pathtuv	path for parent folder of TUV executable for Windows, such as "c:/tuv5.3.1.exe".
df	dataframe of the time series for the parameters to be entered, such as 'date', 'o3col'. It must includes date column.
colid	column index of date. The default value is 1.

Details

There are online and offline versions of the TUV model, but both need to run on a daily basis (that means manually reset parameters for each day's simulation).
This function runs TUV in batch by reading the time series for the parameters to be entered, and summarizes the results to the new dataframe.
Currently only mode 2 (mode that outputs the photolysis rates) is supported.
Logical variables are not supported currently!!!
Please download TUV executable for Windows before you use this function.

Columns of photolysis rate coefficients (s-1):
1 = O3 -> O2 + O(1D)
2 = H2O2 -> 2 OH
3 = NO2 -> NO + O(3P)
4 = NO3 -> NO + O2
5 = NO3 -> NO2 + O(3P)
6 = CH2O -> H + HCO
7 = CH2O -> H2 + CO

Value

a dataframe.The first column is datetime. The second column is the solar altitude Angle. The rates of photolysis for each reaction(Unit: s-1) start from third column: 1 = O3 -> O2 + O1D

`tuv_batch` *Calculate TUV in Batch Online*

Description

This function runs TUV in batch online by reading the time series for the parameters to be entered, and summarizes the results to the new dataframe.

Usage

```
tuv_batch(df, inputMode = 0, outputMode = 2, nStreams = -2)
```

Arguments

`df`	Dataframe of time series of parameters. The first column of df should be datetime. The other columns (names) could be set as following: wStart -> Shortest wavelength. The default value is 280. wStop -> Longest wavelength. The default value is 420. wIntervals -> Number of equal-sized subdivisions of the range End-Start. The default value is 140. latitude -> Latitudes: positive North of equator, negative South of equator. The default value is 0. longitude -> Longitudes: positive East of the Greenwich meridian, negative

West of the Greenwich meridian. The default value is 0.
zenith -> Solar zenith angle (deg). The default value is 0.
ozone -> Ozone column, in Dobson Units (du), vertical, from ground (even if above sea level) to space. The US Standard Atmosphere O3 is used to specify the shape of the vertical profile but the total column is re-scaled to the value selected here by the user. The default value is 300.
albedo -> Surface albedo: Assumes a Lambertian reflection (isotropic radiance) Values for snow can reach 0.90-0.99, but otherwise values at UV wavelengths are in the range 0.02-0.20 depending on the precise surface. The default value is 0.1. gAltitude -> Ground elevation: The elevation of the ground, in km above mean sea level. The default value is 0.
mAltitude -> Measurement altitude: The altitude in the atmosphere for which results are requested. This should not be confused with the ground elevation. For example, if you have measurements made from an airplane, flying at 6 km above the ground, and the surface is at 1.5 km, then you will want to request results for a measurement altitude of 7.5 km asl. The default value is 0.
taucld -> Cloud Optical Depth: vertical optical depth of the cloud. The default value is 0.00.
zbase -> Cloud base: base of cloud, in km (asl). The default value is 4.00.
ztop -> Cloud top: top of cloud, in km (asl). The default value is 5.00.
tauaer -> Optical Depth: total extinction (absorption + scattering) at 550 nm, vertical, from ground to space. The default value is 0.235.
ssaaer -> Single Scattering Albedo (S-S alb), assumed independent of wavelength. The default value is 0.990.
alpha -> Alpha (Angstrom exponent), gives wavelength dependence of optical depth, by multiplying the 550 nm value by (550 nm/wavelength, nm)**alpha. The default value is 1.000.
dirsun -> Direct beam, direct solar beam. The default value is 1.0.
difdn -> Diffuse down, down-ward propagating scattered radiation (diffuse sky light). The default value is 1.0.
difup -> Diffuse up, up-ward propagating scattered radiation (diffuse light from below). The default value is NA.

`inputMode` The default value is 0. InputMode 0: User-specified geographic location and time/date. The code computes the appropriate solar zenith angle and Earth-Sun distance. InputMode 1: User specifies the solar zenith angle, and the annual average Earth-Sun distance is used. To avoid inconsistencies (e.g. overhead sun at the poles), options 1 and 2 cannot be invoked at the same time.

`outputMode` The default value is 2. OutputMode 2: Molecular photolysis frequencies (109 photoreactions). OutputMode 3: Weighted irradiance (27 weighting functions). OutputMode 4: Spectral actinic flux. OutputMode 5: Spectral irradiance.

`nStreams` The default value is -2. NStreams -2: Pseudo-spherical 2 streams (faster, less accurate). NStreams 4: Pseudo-spherical discrete ordinate 4 streams (slower, more accurate).

Value

a dataframe. The contents of dataframe are diterminated by OutputMode.
OutputMode 2: Molecular photolysis frequencies (109 photoreactions).
OutputMode 3: Weighted irradiance (27 weighting functions).
OutputMode 4: Spectral actinic flux.
OutputMode 5: Spectral irradiance.

`tuv_core` *Calculate TUV Online*

Description

This function runs TUV online by reading the input parameters, and summarizes the results to the new dataframe.

Usage

```
tuv_core(
  wStart = 280,
  wStop = 420,
  wIntervals = 140,
  inputMode = 0,
  latitude = 0,
  longitude = 0,
  date = 20150630,
  timeStamp = "12:00:00",
  zenith = 0,
  ozone = 300,
  albedo = 0.1,
  gAltitude = 0,
  mAltitude = 0,
  taucld = 0,
  zbase = 4,
  ztop = 5,
  tauaer = 0.235,
  ssaaer = 0.99,
  alpha = 1,
  time = 12,
  outputMode = 2,
  nStreams = -2,
  dirsun = 1,
  difdn = 1,
  difup = NA
)
```

Arguments

wStart	Shortest wavelength. The default value is 280.
wStop	Longest wavelength. The default value is 420.
wIntervals	Number of equal-sized subdivisions of the range End-Start. The default value is 140.
inputMode	The default value is 0. InputMode 0: User-specified geographic location and time/date. The code computes the appropriate solar zenith angle and Earth-Sun distance. InputMode 1: User specifies the solar zenith angle, and the annual average Earth-Sun distance is used. To avoid inconsistencies (e.g. overhead sun at the poles), options 1 and 2 cannot be invoked at the same time.
latitude	Latitudes: positive North of equator, negative South of equator. The default value is 0.
longitude	Longitudes: positive East of the Greenwich meridian, negative West of the Greenwich meridian. The default value is 0.
date	Date (format: (YYYYMMDD, GMT). The default value is 20150630.
timeStamp	-> Timestamp (format: hh:mm:ss, GMT). The default value is "12:00:00".
zenith	Solar zenith angle (deg). The default value is 0.
ozone	Ozone column, in Dobson Units (du), vertical, from ground (even if above sea level) to space. The US Standard Atmosphere O3 is used to specify the shape of the vertical profile but the total column is re-scaled to the value selected here by the user. The default value is 300.
albedo	Surface albedo: Assumes a Lambertian reflection (isotropic radiance) Values for snow can reach 0.90-0.99, but otherwise values at UV wavelengths are in the range 0.02-0.20 depending on the precise surface. The default value is 0.1.
gAltitude	Ground elevation: The elevation of the ground, in km above mean sea level. The default value is 0.
mAltitude	Measurement altitude: The altitude in the atmosphere for which results are requested. This should not be confused with the ground elevation. For example, if you have measurements made from an airplane, flying at 6 km above the ground, and the surface is at 1.5 km, then you will want to request results for a measurement altitude of 7.5 km asl. The default value is 0.
taucld	Cloud Optical Depth: vertical optical depth of the cloud. The default value is 0.00.

zbase	Cloud base: base of cloud, in km (asl). The default value is 4.00.
ztop	Cloud top: top of cloud, in km (asl). The default value is 5.00.
tauaer	Optical Depth: total extinction (absorption + scattering) at 550 nm, vertical, from ground to space. The default value is 0.235.
ssaaer	Single Scattering Albedo (S-S alb), assumed independent of wavelength. The default value is 0.990.
alpha	Alpha (Angstrom exponent), gives wavelength dependence of optical depth, by multiplying the 550 nm value by (550 nm/wavelength, nm)**alpha. The default value is 1.000.
time	Hour. The default value is 12.
outputMode	The default value is 2. OutputMode 2: Molecular photolysis frequencies (109 photoreactions). OutputMode 3: Weighted irradiance (27 weighting functions). OutputMode 4: Spectral actinic flux. OutputMode 5: Spectral irradiance.
nStreams	The default value is -2. NStreams -2: Pseudo-spherical 2 streams (faster, less accurate). NStreams 4: Pseudo-spherical discrete ordinate 4 streams (slower, more accurate).
dirsun	Direct beam, direct solar beam. The default value is 1.0.
difdn	Diffuse down, down-ward propagating scattered radiation (diffuse sky light). The default value is 1.0.
difup	Diffuse up, up-ward propagating scattered radiation (diffuse light from below). The default value is NA.

Value

a dataframe. The contents of dataframe are diterminated by OutputMode.
OutputMode 2: Molecular photolysis frequencies (109 photoreactions).
OutputMode 3: Weighted irradiance (27 weighting functions).
OutputMode 4: Spectral actinic flux.
OutputMode 5: Spectral irradiance.

voc *Demo data of volatile organic compounds (VOCs)*

Description

5 days VOCs data (1 hour resolution) includes: Propylene, Acetylene, n-Butane, trans-2-Butene, Cyclohexane. The variables are as follows:

Usage

```
voc
```

Format

A data frame with 120 rows and 6 variables:

Time Time for data

Propylene Propylene

Acetylene Acetylene

n.Butane n-Butane

trans.2.Butene trans-2-Butene

Cyclohexane Cyclohexane

vocct *Convertion and analysis of VOC concentrations*

Description

convert unit of VOCs between micrograms per cubic meter (ugm) and parts per billion by volume (ppbv); conduct statistics of VOC concentrations. Note: for Chinese VOC name, please also use English punctuation.

Usage

```
vocct(
  df,
  unit = "ppbv",
  t = 25,
  p = 101.325,
  stcd = FALSE,
  sortd = TRUE,
  chn = FALSE,
  bvoc = TRUE
)
```

Arguments

df	dataframe contains time series.
unit	unit for VOC concentration. A character vector from these options: "ugm" or "ppbv". "ugm" means ug/m3. "ppbv" means part per billion volumn.
t	Temperature, in Degrees Celsius, used to convert data in micrograms per cubic meter to standard conditions (25 Degrees Celsius, 101.325 kPa). By default, t equals to 25 Degrees Celsius.
p	Pressure, in kPa, used to convert data in micrograms per cubic meter to standard conditions (25 Degrees Celsius, 101.325 kPa). By default, p equals to 101.325 kPa.
stcd	logical. Does it output results in standard conditions? The default vaule is FALSE.
sortd	logical value. It determines whether the VOC species are sorted or not. By default, sortd has value "TRUE". If TRUE, VOC species in time series will be arranged according to VOC group, relative molecular weight, and MIR value.
chn	logical. Dose colnames present as Chinese? The default vaule is FALSE.
bvoc	logical. Whether you want to list BVOC as a separate VOC group? The default vaule is TRUE.

Details

The CAS number was matched for each VOC speices (from column name), and the Molecular Weight (MW) value and Maximum Incremental Reactivity (MIR) value are matched through the CAS number and used for time series calculation.
The MIR value comes from "Carter, W. P. (2009). Updated maximum incremental reactivity scale and hydrocarbon bin reactivities for regulatory applications. California Air Resources Board Contract, 2009, 339" (revised January 28, 2010).

Value

a list contains 9 tables: MW_Result: the matched Molecular Weight (MW) value result; Con_ugm: the time series of VOC mass concentration by species; Con_ugm_stat: the statistics of VOC mass concentration by species; Con_ugm_group: the time series of VOC mass concentration classified by groups; Con_ugm_group_stat: the statistics of VOC mass concentrationaccording to major groups; Con_ppbv: time series of VOC volume concentration by species; Con_ppbv_stat: the statistics of VOC volume concentration by species; Con_ppbv_group: the time series of VOC volume concentration according to major groups; Con_ppbv_group_stat: the time series of VOC volume concentration classified by groups.

Examples

```
voc_con=vocct(voc)
summary(voc_con)
```

Index

∗ **datasets**
    aqi, 6
    met, 22
    setup_tuv, 26
    voc, 36

afp, 3
anylm, 4
aqi, 6
avri, 6

dm8n, 8
dm8n_batch, 9
dm8n_np, 10

fm, 11

geom_avri, 12
geom_avri_batch, 13
geom_psd, 14
geom_ts, 15
geom_ts_batch, 19
geom_tsw, 18

koh, 20

loh, 21

met, 22

nsvp, 23

ofp, 24

prop, 25

setup_tuv, 26
statdf, 26
svri, 27

transp, 28
trs, 29

tuv, 30
tuv_batch, 31
tuv_core, 33

voc, 36
vocct, 36