Package 'foqat'

August 18, 2021

Type Package

Title Field Observation Quick Analysis Toolkit

Version 1.7.1

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): <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>.
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

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

Depends R (>= 3.5.0)

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

License GPL-3 | file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

NeedsCompilation no

Repository CRAN

Date/Publication 2021-08-18 07:50:09 UTC

R topics documented:

`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
)
```

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.

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 list with 2 dataframe (average and SD). The first column of dataframe is the serial number within the period. The average variation (or SD) start from the second column.
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

```
dm8n(aqi,colio=6,unitlb=c("NO (ppbv)", "NO2 (ppbv)", "CO (ppbv)", "SO2 (ppbv)", "O3 (ppbv)"))
```

fm *format the theme of plot*

Description

Format the theme of plot.

Usage

```
fm(p, fsz = 13, ff = "TT Arial", lsz = 0.5, tkl = 0.2)
```

Arguments

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

Value

a plot with a new theme.

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
)
```

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.

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: matched KOH value result; LOH_Result: LOH time series of VOC by species; LOH_Result_mean: the average value and proportion of LOH of VOC by species (sorted from large to small); LOH_Result_group: LOH time series of VOC classified by groups; LOH_Result_group_mean: the average value and proportion of LOH of VOC according to major groups (sorted from large to small).

Examples

```
loh(voc)
```

---

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"
)
```

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.

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: matched MIR value result; OFP_Result: OFP time series of VOC by species; OFP_Result_mean: the average value and proportion of OFP of VOC by species (sorted from large to small); OFP_Result_group: OFP time series of VOC classified by groups; OFP_Result_group_mean: the average value and proportion of OFP of VOC according to major groups (sorted from large to small).

Examples

```
ofp(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(x, n = 2)
```

Arguments

x	dataframe of time series.
n	digits for reuslt in dataframe.

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(aqi)
```

`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*

Description

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

Usage

```
trs(
  df,
  bkip,
  st = NULL,
  et = NULL,
  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.
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 with a new time resolution.

Examples

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

---

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

Description

Plot the time series of particle size distribution.

Usage

```
tsplotp(
  df,
  labxyl = NULL,
  logy = TRUE,
  ybk = NULL,
  nlmt = NULL,
  csbk = pretty_breaks(4),
  trans = "identity",
  colsz = 1,
  fsz = 13,
  ff = "TT Arial",
```

```
  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.
ff	font family 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.

---

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

---

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
)
```

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.

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: matched Molecular Weight (MW) value result; Con_ugm: time series of VOC mass concentration by species; Con_ugm_mean: the average mass concentration and proportion of VOC by species (sorted from large to small); Con_ugm_group: time series of VOC mass concentration classified by groups; Con_ugm_group_mean: the average value and proportion of VOC mass concentration (sorted from large to small) according to major groups; Con_ppbv: time series of VOC volume concentration by species; Con_ppbv_mean: the average volume concentration and proportion of VOC by species (sorted from large to small); Con_ppbv_group: time series of VOC volume concentration according to major groups; Con_ppbv_group_mean: VOC volume concentration average and proportion (sorted from large to small) according to major groups;

Examples

```
vocct(voc)
```

Index