

Package ‘arpautils’

December 6, 2014

Type Package

Title R utility functions for Arpa ER air quality data

Version 0.8.7

Date 2014-12-05

Author G.Bonafe'

Maintainer G.Bonafe' <gbonafe@arpa.emr.it>

Depends ROracle (>= 1.1-11), xts (>= 0.9-7), caTools (>= 1.17)

Description Funzioni utili per gestire i dati di qualita' dell'aria di Arpa Emilia-Romagna

License GPL-2

R topics documented:

arpautils-package	2
aot	2
aqstat.functions	3
dbqa.config	5
dbqa.connect	6
dbqa.data2xts	6
dbqa.delete	7
dbqa.get.datasens	8
dbqa.get.datastaz	9
dbqa.get.idcfgsens	10
dbqa.insert	11
dbqa.list.active.staz	12
dbqa.list.fields	13
dbqa.list.tables	13
dbqa.round	14
dbqa.view.param	15
dbqa.view.staz	15
generic_annual_report	16

ozone_annual_report	18
ozone_daily_report	20
round.awayfromzero	21
simple.query	22
squeeze	22
time.functions	23
xts.blend	24
xts.regolarize	25
Index	26

arpautils-package	<i>The Arpa ER Utils Package</i>
-------------------	----------------------------------

Description

R utility functions for Arpa ER

Details

Package: arpautils
Type: Package
Version: 0.8.7
Date: 2014-12-05
License: GPL-2

Funzioni utili per gestire i dati di qualita' dell'aria di Arpa Emilia-Romagna

Author(s)

G.Bonafe' <gbonafe@arpa.emr.it>

aot	<i>Calcola AOT</i>
-----	--------------------

Description

Calcola una generica Accumulated exposure Over Threshold

Usage

aot(x, hr, threshold = 80, estimate = T, hr.min = 8, hr.max = 19)

Arguments

x	vettore dei valori di concentrazione
hr	vettore, della stessa lunghezza di x, che identifica l'orario
threshold	soglia
estimate	variabile logica, se TRUE viene eseguita la stima di AOT corretta per i dati mancanti, conformemente alla normativa
hr.min	prima ora della fascia oraria da considerare
hr.max	ultima ora della fascia oraria da considerare

Author(s)

G.Bonafantini

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.

## The function is currently defined as
function (x, hr, threshold = 80, estimate = T, hr.min = 8, hr.max = 19)
{
  if (length(x) != length(hr))
    stop("x and hr must have the same length!")
  delta.positive <- pmax(x - threshold, 0)
  in.hr <- hr >= hr.min & hr <= hr.max
  valid <- !is.na(x)
  in.hr.valid <- in.hr & valid
  Aot <- sum(delta.positive, na.rm = T)
  if (estimate)
    Aot <- Aot * sum(in.hr)/sum(in.hr.valid)
  PercValid <- sum(in.hr.valid)/sum(in.hr) * 100
  out <- list(Aot = Aot, PercValid = PercValid)
  return(out)
}
```

Description

Funzioni base per il calcolo di statistiche di legge su dati QA.

Usage

```

stat.period(x, period, necess, FUN = mean)
stat.period2(x, period, nmax.missing, FUN = mean)
which.period(x, period, necess, FUN=which.max)
exc.period(x, period, necess, threshold)

stat.window(x, window, necess, FUN = mean)
mean.window(x, k, necess)

detect.event(x, threshold)

shift(x, k)

```

Arguments

x	vettore dei valori di concentrazione
period	vettore, della stessa lunghezza di x, che identifica i periodi
window	vettore numerico di due elementi; descrive l'ampiezza della finestra mobile, p.es. c(-7,0) per la media mobile su 8 ore
necess	numero di dati validi necessari in ciascun periodo
nmax.missing	numero massimo di dati mancanti accettabili in ciascun periodo
FUN	funzione da applicare
threshold	soglia
k	in shift, numero di passi di cui si vuole spostare x; in mean.window, ampiezza della finestra

Details

Le funzioni `stat.period` e `stat.period2` calcolano una statistica FUN su periodi definiti, con approcci diversi nella gestione dei mancanti. La funzione `which.period` lavora come `stat.period`, ma si puo' usare per funzioni (come `which.min` o `which.max`) che non accettano l'argomento `na.rm`.

Invece `stat.window` opera su una finestra mobile, e chiama `shift` che sposta la serie temporale in avanti o indietro nel tempo. Piu' efficiente, ma limitata alla media mobile, e' `mean.window`.

La funzione `exc.period` fa il conteggio dei superamenti di una data soglia. Invece `detect.event` restituisce una matrice contenente la data e l'orario dei superamenti e la loro durata (espressa in numero di timestep).

Author(s)

G.Bonafantini

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do  help(data=index)  for the standard data sets.

## The function is currently defined as
function (x, period, necess, FUN = mean)
{
  xmean <- tapply(X = x, INDEX = period, FUN = FUN, na.rm = T)
  xvalid <- tapply(X = !is.na(x), INDEX = period, FUN = sum,
    na.rm = T)
  xmean[xvalid < necess] <- NA
  return(xmean)
}
```

dbqa.config

Prepara le credenziali di accesso al DB

Description

Prepara le credenziali di accesso al DB di qualita' dell'aria

Usage

```
dbqa.config(db_usr, db_pwd, db_name, db_tz)
```

Arguments

db_usr	user
db_pwd	password
db_name	indirizzo
db_tz	timezone

Author(s)

Giovanni Bonafe', Arpa Emilia-Romagna

dbqa.connect	<i>Si connette al DB QA</i>
--------------	-----------------------------

Description

Si connette al DB QA

Usage

```
dbqa.connect(db_usr, db_pwd, db_name, db_tz="BST")
```

Arguments

db_usr	user
db_pwd	password
db_name	indirizzo
db_tz	timezone. Default 'BST', ora solare locale italiana.

Author(s)

Giovanni Bonafe', Arpa Emilia-Romagna

dbqa.data2xts	<i>Converte i dati estratti da DB in un oggetto di tipo xts</i>
---------------	---

Description

Converte i dati estratti da DB in un oggetto di tipo xts

Usage

```
dbqa.data2xts(data, Date = "TS_INIZIO_RIL", Value = "VALORE", TZ="BST")
```

Arguments

data	matrice o dataframe che include i dati, così come sono estratti dal DB
Date	il nome del campo che contiene data (e ora) in data
Value	il nome del campo che contiene le concentrazioni in data
TZ	timezone (vedi Warning)

Warning

Si fa riferimento all'ora "BST" (British Summer Time) poiché (assieme a "Africa/Algiers") è l'unico standard codificato corrispondente all'ora del DB, nonché all'ora a cui si riferisce la normativa: Central Europe Time senza Daylight Saving Time (ora legale).

Author(s)

Giovanni Bonafe', Arpa Emilia-Romagna

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.

## The function is currently defined as
function (data, Date = "TS_INIZIO_RIL", Value = "VALORE")
{
  library("xts")
  out <- xts(data[[Value]], as.POSIXct(data[[Date]]))
  return(out)
}
```

dbqa.delete

Cancella record di una tabella nel DB QA

Description

Cancella record di una tabella nel DB QA

Usage

```
dbqa.delete(con, tab, keys, values, verbose = F)
```

Arguments

con	connessione al DB
tab	nome della tabella
keys	chiavi primarie
values	valori delle chiavi primarie che identificano i record da rimuovere
verbose	se TRUE fornisce informazioni utili per il debug

Author(s)

Giovanni Bonafe'

dbqa.get.datasens	<i>Estrae i dati di un sensore</i>
-------------------	------------------------------------

Description

Estrae i dati di un sensore

Usage

```
dbqa.get.datasens(con, ts.range, id.cfgsens, id.param, flg = 1, flg.excl=NULL, flg.null=FALSE, verbose)
```

Arguments

con	connessione al DB
ts.range	periodo di interesse, definito come vettore c(datainizio,datafine)
id.cfgsens	codice numerico di configurazione del sensore
id.param	codice numerico dell'inquinante
flg	valore (o valori) accettati per la flag di qualita' FLG_A. Se NULL non viene posta alcuna condizione includente sulla flag
flg.excl	valore (o valori) non accettati per la flag di qualita' FLG_A. Se NULL non viene posta alcuna condizione escludente sulla flag
flg.null	se TRUE vengono estratti anche record con valori di flag FLG_4 pari a 'NULL' nel DB, cio<c3><a8> non ancora sottoposti a validazione quotidiana
verbose	parametro logico. Se TRUE, verranno visualizzate informazioni utili per il debug
table	stringa che definisce la tabella da cui estrarre i dati. Al momento implementate le opzioni "storico" e "annuale".
...	argomenti opzionali, passati a dbSendQuery

Author(s)

Giovanni Bonafe', Arpa Emilia-Romagna

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.

## The function is currently defined as
function (con, ts.range, id.cfgsens, id.param, flg = 1, ...)
{
  tab <- "AA_ARIA.T$01$DATI_STORICO"
  what <- c("TO_CHAR(TS_INIZIO_RIL,'YYYY-MM-DD HH24:MI') AS TS_INIZIO_RIL",
    "VALORE", "FLG_A")
  crit <- c(paste("ID_PARAMETRO=", id.param, sep = ""), paste("ID_CONFIG_SENSORE=",
    id.cfgsens, sep = ""), paste("to_char(TS_INIZIO_RIL,'yyyy-mm-dd') between '",
```



```

        ts.range[1], "' and '", ts.range[2], "'", sep = ""))
  if (!is.null(flq)) {
    crit <- c(crit, paste(paste("FLG_A=", flq, sep = ""),
      collapse = " OR "))
  }
  query <- simple.query(tab, what, crit)
  print(query)
  library("ROracle")
  data <- dbGetQuery(con, query, ...)
  data$TS_INIZIO_RIL <- as.POSIXct(data$TS_INIZIO_RIL)
  return(data)
}

```

dbqa.get.datastaz	<i>Estrae i dati di una stazione per un dato parametro (inquinante)</i>
-------------------	---

Description

Estrae i dati di una stazione per un dato parametro (inquinante)

Usage

```
dbqa.get.datastaz(con, ts.range, id.staz, id.param, flg = 1, tstep, ...)
```

Arguments

con	connessione al DB
ts.range	periodo di interesse, definito come vettore c(datainizio,datafine)
id.staz	codice numerico identificativo della stazione
id.param	codice numerico dell'inquinante
flg	valore (o valori) accettati per la flag di qualita'
tstep	stringa di carattere che identifica il timestep ("H" o "d", orario o giornaliero)
...	argomenti opzionali, passati a dbqa.get.datasens

Author(s)

Giovanni Bonafe', Arpa Emilia-Romagna

Examples

```

##---- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do  help(data=index)  for the standard data sets.

## The function is currently defined as
function (con, ts.range, id.staz, id.param, flg = 1, ...)
{

```

```

cfgsens <- dbqa.get.idcfgsens(con, id.param, i.date = ts.range[1],
                             f.date = ts.range[2], id.staz)
DATA <- NULL
nsens <- length(cfgsens$idcfgsens)
for (i in 1:nsens) {
  idate <- cfgsens$idate[i]
  fdate <- cfgsens$fdate[i]
  if (is.na(idate))
    idate <- as.POSIXct(ts.range[1])
  if (is.na(fdate))
    fdate <- as.POSIXct(ts.range[2])
  dat <- dbqa.get.datasens(con, c(idate, fdate), cfgsens$idcfgsens[i],
                           id.param, flg = flg, ...)
  DATA <- data.frame(rbind(DATA, dat))
}
return(DATA)
}

```

dbqa.get.idcfgsens	<i>Estrae l'elenco dei codici sensore corrispondenti ad una data stazione, per un dato periodo, per un dato parametro (inquinante)</i>
--------------------	--

Description

Estrae l'elenco dei codici sensore corrispondenti ad una data stazione, per un dato periodo, per un dato parametro (inquinante)

Usage

```
dbqa.get.idcfgsens(con, id.param, i.date = NULL, f.date = NULL, id.staz)
```

Arguments

con	connessione al DB
id.param	codice numerico dell'inquinante
i.date	data iniziale
f.date	data finale
id.staz	codice numerico identificativo della stazione

Author(s)

Giovanni Bonafe', Arpa Emilia-Romagna

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do  help(data=index)  for the standard data sets.

## The function is currently defined as
function (con, id.param, i.date = NULL, f.date = NULL, id.staz)
{
  tab <- "AA_ARIA.ANG_CONFIG_SENSORI"
  crit <- c(paste("ID_PARAMETRO=", id.param, sep = ""), paste("ID_STAZIONE=",
    id.staz, sep = ""))
  if (!is.null(i.date)) {
    crit <- c(crit, paste("(DATA_FINE is null OR to_char(DATA_FINE,'yyyymmdd')>=",
      format(as.POSIXct(i.date), format = "%Y%m%d"), "'')",
      sep = ""))
  }
  if (!is.null(f.date)) {
    crit <- c(crit, paste("(DATA_INIZIO is null OR to_char(DATA_INIZIO,'yyyymmdd')<=",
      format(as.POSIXct(f.date), format = "%Y%m%d"), "'')",
      sep = ""))
  }
  library("ROracle")
  query.idcfigsens <- simple.query(tab, what = "ID_CONFIG_SENSORE",
    crit)
  idcfigsens <- fetch(ds <- dbSendQuery(con, query.idcfigsens))[,
    1]
  query.idate <- simple.query(tab, what = "DATA_INIZIO", crit)
  idate <- fetch(ds <- dbSendQuery(con, query.idate))[, 1]
  idate <- pmax(as.POSIXct(idate), as.POSIXct(i.date))
  query.fdate <- simple.query(tab, what = "DATA_FINE", crit)
  fdate <- fetch(ds <- dbSendQuery(con, query.fdate))[, 1]
  fdate <- pmin(as.POSIXct(fdate), as.POSIXct(f.date))
  idx <- order(idate, decreasing = F)
  idate <- idate[idx]
  fdate <- fdate[idx]
  idcfigsens <- idcfigsens[idx]
  out <- list(idcfigsens = idcfigsens, idate = idate, fdate = fdate)
  return(out)
}
```

dbqa.insert

Inserisce valori in una tabella del DB.

Description

Inserisce valori in una tabella del DB.

Usage

```
dbqa.insert(con, tab, values, columns = colnames(values), to_date=NULL, update = F, verbose = F)
```

Arguments

con	connessione al DB
tab	nome della tabella
values	vettore o data.frame contenente i valori da inserire in tabella
columns	nomi dei campi da scrivere
to_date	vettore numerico che identifica le colonne che sono carattere e in formato SQL 'YYYY-MM-DD HH24:MI', da convertire in tipo DATE di Oracle, attraverso la funzione TO_DATE
update	se TRUE aggiorna record esistenti
verbose	se TRUE fornisce informazioni utili al debug

Author(s)

Giovanni Bonafe'

dbqa.list.active.staz *Restituisce la lista delle stazioni attive.*

Description

Restituisce la lista delle stazioni attive in una provincia, in una data

Usage

```
dbqa.list.active.staz(con, prov, Day = Sys.Date())
```

Arguments

con	connessione al DB
prov	sigla della provincia
Day	data richiesta, in formato POSIX

Author(s)

Giovanni Bonafe'

dbqa.list.fields	<i>Elenca i campi presenti in una tabella del DB</i>
------------------	--

Description

Elenca i campi presenti in una tabella del DB

Usage

```
dbqa.list.fields(con, tab)
```

Arguments

con	connessione al DB
tab	nome della tabella del DB

Author(s)

Giovanni Bonafe', Arpa Emilia-Romagna

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.

## The function is currently defined as
function (con, tab)
{
  query <- paste("select * from ", tab, " where rownum=0")
  library("ROracle")
  data <- fetch(ds <- dbSendQuery(con, query))
  out <- colnames(data)
  return(out)
}
```

dbqa.list.tables	<i>Elenca le tabelle disponibili</i>
------------------	--------------------------------------

Description

Elenca le tabelle disponibili nel DB

Usage

```
dbqa.list.tables(con)
```

Arguments

con connessione al DB

Author(s)

Giovanni Bonafe', Arpa Emilia-Romagna

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.

## The function is currently defined as
function (con)
{
  library("ROracle")
  tr <- dbSendQuery(con, "select table_name from all_tables")
  tables <- paste("AA_ARIA.", sort(fetch(tr)[, 1]), sep = "")
  return(tables)
}
```

dbqa.round

arrotonda concentrazioni

Description

arrotonda concentrazioni in visualizzazione, secondo le indicazioni del Gruppo di Lavoro (Istruzione Operativa)

Usage

```
dbqa.round(x, id.param)
```

Arguments

x valore da arrotondare
 id.param identificativo del parametro nel DB

Examples

```
# concentrazioni casuali tra 0 e 50
x <- runif(n=100, min=0, max=50)
dbqa.round(x,7) # come fosse ozono
dbqa.round(x,14) # come fosse cadmio
## Not run:
dbqa.round(x,17) # come fosse antani

## End(Not run)
```

dbqa.view.param	<i>Elenca i parametri (inquinanti)</i>
-----------------	--

Description

Elenca i parametri (inquinanti) disponibili nel DB

Usage

```
dbqa.view.param(con, FUN=View)
```

Arguments

con	connessione al DB
FUN	azione da applicare per la visualizzazione della matrice (usare return se si vuole avere la matrice come output)

Author(s)

Giovanni Bonafe', Arpa Emilia-Romagna

dbqa.view.staz	<i>Elenca le stazioni QA.</i>
----------------	-------------------------------

Description

Elenca le stazioni, applicando la query `select distinct ID_STAZIONE, NOME_STAZIONE, COMUNE, PROVINCIA from AA_ARIA.ANG_CONFIG_SENSORI`

Usage

```
dbqa.view.staz(con, FUN = View)
```

Arguments

con	connessione al DB
FUN	azione da applicare per la visualizzazione della matrice (usare return se si vuole avere la matrice come output, vedi Examples)

Examples

```
## Not run:
db_usr="yourUsername"
db_pwd="yourPassword"
db_name="DBaddress"
con <- dbqa.connect(db_usr, db_pwd, db_name)
dbqa.view.staz(con)

## ...oppure equivalente
Dat <- dbqa.view.staz(con, FUN=return)
View(Dat)
dbDisconnect(con)

## End(Not run)
```

generic_annual_report *Funzioni per produrre statistiche annuali (una stazione)*

Description

Funzioni per produrre le statistiche annuali tipiche (media annua, superamenti giornalieri della media e del max della media mobile su 8h) per una singola stazione: estrazione, calcoli, scrittura su DB.

Usage

```
prepare_annual_report(con, id.staz, id.param, year=NULL, tstep, ...)
calculate_annual_report(data, id.param, thr.daily.ave=NULL, thr.ave8h.max=NULL,
                        thr.hourly=NULL, thr.multihourly=NULL, NH=3,
                        critical.months=NULL)
write_annual_report(con, AR, id.param, verbose=F, ...)
```

Arguments

con	identificativo della connessione al DB (stringa)
id.staz	codice numerico identificativo della stazione
id.param	codice numerico identificativo dell'inquinante
year	anno per cui si richiede il report. Se lasciato NULL prende l'anno di 5 mesi fa, facendo riferimento a Sys.Date().
tstep	stringa di carattere che identifica il timestep ("H" o "d", orario o giornaliero)
data	dati estratti da prepare_annual_report
thr.daily.ave	soglia per la media giornaliera
thr.ave8h.max	soglia per il max giornaliero della media mobile su 8h
thr.hourly	soglia oraria

thr.multihourly	soglia per superamenti orari di pi<c3><b9> ore consecutive
NH	numero di ore consecutive di superamento da contare se thr.multihourly non <c3><a8> NULL
critical.months	vettore numerico dei mesi su cui calcolare la media di periodo (p.es. c(1:3,10:12) per la media invernale)
AR	lista di data.frame prodotta da calculate.annual_report (vedi Value)
verbose	scrive a video alcune informazioni utili per il debug
...	parametri opzionali. La funzione prepare.annual_report li passa a dbqa.get.datastaz; la funzione write.annual_report li passa a dbqa.insert.

Value

La funzione calculate.annual_report restituisce un data.frame con:

annual.mean	media annua
annual.nValid	numero di dati validi usati per il calcolo della media annua
annual.percValid	percentuale di dati validi usati per il calcolo della media annua
annual.nExpected	numero di dati attesi nell'anno
annual.efficiency	rapporto tra numero di dati disponibili e attesi nell'anno

se thr.daily.ave non <c3><a8> NULL allora data.frame ha anche:

daily.nexc	superamenti della media giornaliera
daily.nValid	numero di dati validi usati per il calcolo dei superamenti della media giornaliera
daily.percValid	percentuale di dati validi usati per il calcolo dei superamenti della media giornaliera

se thr.ave8h.max non <c3><a8> NULL allora data.frame ha anche:

ave8h.nexc	superamenti del max giornaliero della media mobile su 8h
ave8h.nValid	numero di dati validi usati per il calcolo dei superamenti del max giornaliero della media mobile su 8h
ave8h.percValid	percentuale di dati validi usati per il calcolo dei superamenti del max giornaliero della media mobile su 8h

se thr.hourly o thr.multihourly non sono NULL allora data.frame ha anche:

hourly.nValid	numero di dati orari validi
hourly.percValid	percentuale di dati orari validi

se thr.hourly non <c3><a8> NULL allora data.frame ha anche:

hourly.nexc numero di superamenti orari

se thr.multiphourly non <c3><a8> NULL allora data.frame ha anche:

multiphourly.nexc numero di superamenti orari di almeno NH ore consecutive

se critical.months non <c3><a8> NULL allora data.frame ha anche:

critmonths.mean media dei mesi selezionati

critmonths.nValid numero di dati validi nei mesi selezionati

critmonths.percValid percentuale di dati validi nei mesi selezionati

critmonths.nExpected numero di dati attesi nei mesi selezionati

critmonths.ency rapporto tra numero di dati disponibili e attesi nei mesi selezionati

Author(s)

Giovanni Bonafe', Arpa Emilia-Romagna

Examples

```
## Not run:
db_usr="yourUsername"
db_pwd="yourPassword"
db_name="DBaddress"
con1 <- dbqa.connect(db_usr, db_pwd, db_name)
dat <- prepare.annual_report(con=con1, id.staz="2000003")
calculate.annual_report(data=dat)
dbDisconnect(con1)

## End(Not run)
```

ozone_annual_report	<i>Funzioni per produrre le statistiche annuali per l'ozono (una stazione)</i>
---------------------	--

Description

Funzioni per produrre le statistiche annuali per una singola stazione di ozono: estrazione, calcoli, scrittura su DB.

Usage

```
prepare.ozone_annual_report(con, id.staz, year=NULL, ...)
calculate.ozone_annual_report(data)
write.ozone_annual_report(con, OAR, verbose=F, ...)
```

Arguments

<code>con</code>	identificativo della connessione al DB (stringa)
<code>id.staz</code>	codice numerico identificativo della stazione
<code>year</code>	anno per cui si richiede il report. Se lasciata NULL prende l'anno di 5 mesi fa, facendo riferimento a <code>Sys.Date()</code> .
<code>data</code>	dati estratti da <code>prepare.ozone_annual_report</code>
<code>OAR</code>	lista di <code>data.frame</code> prodotta da <code>calculate.ozone_annual_report</code> (vedi Value)
<code>verbose</code>	scrive a video alcune informazioni utili per il debug
<code>...</code>	parametri opzionali. La funzione <code>prepare.ozone_annual_report</code> li passa a <code>dbqa.get.datastaz</code> ; la funzione <code>write.ozone_annual_report</code> li passa a <code>dbqa.insert</code> .

Value

La funzione `calculate.ozone_annual_report` restituisce una lista di due oggetti:

`annual.report` e' un `data.frame` di un sola riga con le seguenti colonne:

<code>cumul.nexc.180</code>	numero di superamenti orari della soglia 180
<code>cumul.nexc.240</code>	numero di superamenti orari della soglia 240
<code>cumul.nexc.120</code>	numero di superamenti di max.ave.8h della soglia 120
<code>aot40.veget</code>	AOT40 per la vegetazione
<code>aot40.veget.PercValid</code>	percentuale di dati validi per il calcolo dell'AOT40 per la vegetazione
<code>aot40.forest</code>	AOT40 per le foreste
<code>aot40.forest.PercValid</code>	percentuale di dati validi per il calcolo dell'AOT40 per le foreste

`events` e' una lista contenente i due `data.frame`, ciascuno dei quali avente colonne `start.time` e `duration`:

<code>exc.180</code>	contenente gli eventi orari di superamento della soglia 180
<code>exc.240</code>	contenente gli eventi orari di superamento della soglia 240

Author(s)

Giovanni Bonafe', Arpa Emilia-Romagna

Examples

```
## Not run:
db_usr="yourUsername"
db_pwd="yourPassword"
db_name="DBaddress"
con1 <- dbqa.connect(db_usr, db_pwd, db_name)
```

```

dat <- prepare.annual_report(con=con1, id.staz="2000003")
calculate.ozone_annual_report(data=dat)
dbDisconnect(con1)

## End(Not run)

```

ozone_daily_report *Funzioni per produrre i bollettini quotidiani per l'ozono (una stazione)*

Description

Funzioni per produrre i bollettini quotidiani per una singola stazione di ozono: estrazione, calcoli, scrittura su DB.

Usage

```

prepare.ozone_daily_report(con, id.staz, Date=NULL, ...)
calculate.ozone_daily_report(data)
write.ozone_daily_report(con, ODR, empty=F, verbose=F, ...)

```

Arguments

con	identificativo della connessione al DB (stringa)
id.staz	codice numerico identificativo della stazione
Date	data per cui si richiede il report. Se lasciata NULL prende la giornata di ieri, facendo riferimento a Sys.Date().
data	dati estratti da prepare.ozone_daily_report
ODR	lista di data.frame prodotta da calculate.ozone_daily_report (vedi Value)
empty	svuota tutta la tabella del DB prima di scriverci
verbose	scrive a video alcune informazioni utili per il debug
...	parametri opzionali. La funzione prepare.ozone_daily_report li passa a dbqa.get.datastaz; la funzione write.ozone_daily_report li passa a dbqa.insert.

Value

La funzione calculate.ozone_daily_report restituisce una lista di due oggetti: daily.report e' un data.frame di un sola riga con le seguenti colonne:

max.day	massimo giornaliero
hour.max.day	ora in cui si e' verificato il massimo (inizio dell'intervallo orario)
max.ave.8h	massimi giornaliero della media mobile su 8 ore
cumul.nexc.180	numero di superamenti orari della soglia 180, dall'inizio dell'anno
cumul.nexc.240	numero di superamenti orari della soglia 240, dall'inizio dell'anno

```
cumul.nexc.120
```

numero di superamenti di max.ave.8h della soglia 120, dall'inizio dell'anno

events e' una lista contenente i due data.frame, ciascuno dei quali avente colonne start.time e duration:

```
exc.180
```

contenente gli eventi orari di superamento della soglia 180

```
exc.240
```

contenente gli eventi orari di superamento della soglia 240

Warning

Usare l'opzione empty=TRUE solo se strettamente necessario: svuota tutta la tabella delle statistiche giornaliere dell'ozono sul DB.

Author(s)

Giovanni Bonafe', Arpa Emilia-Romagna

Examples

```
## Not run:
## report di ieri per Cittadella
db_usr="yourUsername"
db_pwd="yourPassword"
db_name="DBaddress"
day <- format(Sys.Date()-1,format='%Y-%m-%d')
con <- dbqa.connect(db_usr, db_pwd, db_name)
Dat <- prepare.ozone_daily_report(con,id.staz="2000003",Date=day)
ODR <- calculate.ozone_daily_report(Dat)
dbDisconnect(con)
```

```
## End(Not run)
```

round.awayfromzero *Arrotondamento conforme all'Istruzione Operativa I70502/SA*

Usage

```
round.awayfromzero(x, digits = 0)
```

Arguments

x valore da arrotondare

digits cifre decimali

simple.query	<i>Costruisce una semplice query SQL.</i>
--------------	---

Description

Costruisce una semplice query SQL.

Usage

```
simple.query(tab, what, crit)
```

Arguments

tab	nome della tabella del DB
what	campo/i da estrarre
crit	criterio di selezione

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.

## The function is currently defined as
function (tab, what, crit)
{
  out <- paste("SELECT", paste(what, collapse = ","), "FROM",
    tab, ("WHERE")[!is.null(crit)], paste(crit, collapse = " AND "))
  return(out)
}
```

squeeze	<i>Dato un vettore numerico, lo rappresenta come stringa sintetica</i>
---------	--

Description

Dato un vettore di interi, ne da' una rappresentazione sintetica in una stringa.

Usage

```
squeeze(x)
```

Arguments

x

Examples

```
x <- c(2:5,8,13:21)
x
paste(x,collapse=",")
squeeze(x)
squeeze(sample(x=1:1000,size=995))
```

time.functions

*Funzioni che gestiscono le date, ore, ecc.***Description**

Funzioni che gestiscono i riferimenti temporali (date, ore, ecc.) di una serie temporale di tipo xts.

Usage

```
Hour(x, tz = "BST")
Month(x, tz = "BST")
Year(x, tz = "BST")
Ymd(x, tz = "BST")
Ym(x, tz = "BST")
YQ(x, tz = "BST")
Ndays(x, tz = "BST")
Nmonths(x, tz = "BST")
Ndays.in.year(year, tz = "BST")
```

Arguments

x	vettore di date POSIXct
year	anno
tz	timezone

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.

## The function is currently defined as
function (x, tz = "BST")
as.numeric(format(as.POSIXct(x, tz = tz), format = "%H"))
```

xts.blend	<i>fonde tra loro due oggetti xts</i>
-----------	---------------------------------------

Description

fonde tra loro due oggetti xts

Usage

```
xts.blend(tstep, TZ = "BST", ...)
```

Arguments

tstep	stringa di carattere che identifica il timestep ("H" o "d", orario o giornaliero)
TZ	timezone
...	

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.

## The function is currently defined as
function (step, TZ = "BST", ...)
{
  if (!(step %in% c("d", "H")))
    stop(paste("Invalid timestep value, only `d` or `H` allowed"))
  Xm <- merge.xts(..., tzone = TZ)
  idx.time <- index(Xm)
  f.time <- idx.time[1]
  l.time <- idx.time[length(idx.time)]
  timestring <- paste(format(f.time, "%Y-%m-%d %H:%M"), format(l.time,
    "%Y-%m-%d %H:%M"), step, sep = "/")
  s0 <- timeBasedSeq(timestring)
  x0 <- xts(rep(NA, length(s0)), s0, tzone = TZ)
  Xm <- merge.xts(x0, Xm, join = "left", tzone = TZ)
  dm <- apply(Xm, MARGIN = 1, FUN = max, na.rm = T)
  dm[is.infinite(dm)] <- NA
  dm[is.nan(dm)] <- NA
  xm <- xts(dm, s0)
  return(xm)
}
```

xts.regularize	<i>Regolarizza la scansione temporale di un oggetto xts</i>
----------------	---

Description

Regolarizza la scansione temporale di un oggetto xts

Usage

```
xts.regularize(tstep, x, f.time=(i<-index(x))[1], l.time=i[length(i)], TZ="BST")
```

Arguments

tstep	stringa di carattere che identifica il timestep ("H" o "d", orario o giornaliero)
x	serie temporale di tipo xts
TZ	timezone
f.time	primo istante richiesto in output
l.time	ultimo istante richiesto in output

Index

*Topic \textasciitildeair quality

- generic_annual_report, 16
- ozone_annual_report, 18
- ozone_daily_report, 20

*Topic \textasciitildeekwd1

- aot, 2
- aqstat.functions, 3
- dbqa.config, 5
- dbqa.connect, 6
- dbqa.data2xts, 6
- dbqa.delete, 7
- dbqa.get.datasens, 8
- dbqa.get.datastaz, 9
- dbqa.get.idcfgsens, 10
- dbqa.insert, 11
- dbqa.list.active.staz, 12
- dbqa.list.fields, 13
- dbqa.list.tables, 13
- dbqa.round, 14
- dbqa.view.param, 15
- dbqa.view.staz, 15
- round.awayfromzero, 21
- simple.query, 22
- squeeze, 22
- time.functions, 23
- xts.blend, 24
- xts.regularize, 25

*Topic \textasciitildeekwd2

- aot, 2
- aqstat.functions, 3
- dbqa.config, 5
- dbqa.connect, 6
- dbqa.data2xts, 6
- dbqa.delete, 7
- dbqa.get.datasens, 8
- dbqa.get.datastaz, 9
- dbqa.get.idcfgsens, 10
- dbqa.insert, 11
- dbqa.list.active.staz, 12

- dbqa.list.fields, 13
- dbqa.list.tables, 13
- dbqa.round, 14
- dbqa.view.param, 15
- dbqa.view.staz, 15
- round.awayfromzero, 21
- simple.query, 22
- squeeze, 22
- time.functions, 23
- xts.blend, 24
- xts.regularize, 25

*Topic \textasciitildeozone

- generic_annual_report, 16
- ozone_annual_report, 18
- ozone_daily_report, 20

*Topic package

- arpautils-package, 2

- aot, 2
- aqstat.functions, 3
- arpautils (arpautils-package), 2
- arpautils-package, 2

- calculate.annual_report
(generic_annual_report), 16
- calculate.ozone_annual_report
(ozone_annual_report), 18
- calculate.ozone_daily_report
(ozone_daily_report), 20

- dbqa.config, 5
- dbqa.connect, 6
- dbqa.data2xts, 6
- dbqa.delete, 7
- dbqa.get.datasens, 8
- dbqa.get.datastaz, 9
- dbqa.get.idcfgsens, 10
- dbqa.insert, 11
- dbqa.list.active.staz, 12
- dbqa.list.fields, 13

dbqa.list.tables, 13
dbqa.round, 14
dbqa.view.param, 15
dbqa.view.staz, 15
detect.event(aqstat.functions), 3

exc.period(aqstat.functions), 3

generic_annual_report, 16

Hour(time.functions), 23

mean.window(aqstat.functions), 3
Month(time.functions), 23

Ndays(time.functions), 23
Nmonths(time.functions), 23

ozone_annual_report, 18
ozone_daily_report, 20

prepare.annual_report
 (generic_annual_report), 16
prepare.ozone_annual_report
 (ozone_annual_report), 18
prepare.ozone_daily_report
 (ozone_daily_report), 20

round.awayfromzero, 21

shift(aqstat.functions), 3
simple.query, 22
squeeze, 22
stat.period(aqstat.functions), 3
stat.period2(aqstat.functions), 3
stat.window(aqstat.functions), 3

time.functions, 23

which.period(aqstat.functions), 3
write.annual_report
 (generic_annual_report), 16
write.ozone_annual_report
 (ozone_annual_report), 18
write.ozone_daily_report
 (ozone_daily_report), 20

xts.blend, 24
xts.regolarize, 25

Year(time.functions), 23
Ym(time.functions), 23
Ymd(time.functions), 23
YQ(time.functions), 23