

# Package ‘arpautils’

January 19, 2015

**Type** Package

**Title** R utility functions for Arpa ER air quality data

**Version** 0.8.10

**Date** 2015-01-15

**Author** G.Bonafè

**Maintainer** G.Bonafè <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

**Encoding** latin1

## R topics documented:

aot . . . . .	2
aqstat.functions . . . . .	3
dbqa.config . . . . .	4
dbqa.connect . . . . .	4
dbqa.data2xts . . . . .	5
dbqa.delete . . . . .	5
dbqa.get.datasens . . . . .	6
dbqa.get.datastaz . . . . .	7
dbqa.get.idcfsens . . . . .	7
dbqa.insert . . . . .	8
dbqa.isrrqa . . . . .	8
dbqa.list.active.staz . . . . .	9
dbqa.list.fields . . . . .	9
dbqa.list.tables . . . . .	10
dbqa.round . . . . .	10
dbqa.view.param . . . . .	11

dbqa.view.staz . . . . .	11
generic_annual_report . . . . .	12
ozone_annual_report . . . . .	14
ozone_daily_report . . . . .	16
round.awayfromzero . . . . .	17
simple.query . . . . .	18
squeeze . . . . .	18
time.functions . . . . .	19
xts.blend . . . . .	19
xts.regolarize . . . . .	20
<b>Index</b>	<b>21</b>

---

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.Bonafè

**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.Bonafè

---

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 Bonafè, Arpa Emilia-Romagna

---

dbqa.connect*Si connette al DB QA*

---

**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 Bonafè, 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 Bonafè, Arpa Emilia-Romagna

---

dbqa.delete	<i>Cancella record di una tabella nel DB QA</i>
-------------	---

---

**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 Bonafè

---

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, cioÃ" 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 Bonafè, Arpa Emilia-Romagna

---

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 Bonafè, Arpa Emilia-Romagna

---

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 Bonafè, Arpa Emilia-Romagna

---

dbqa.insert	<i>Inserisce valori in una tabella del DB.</i>
-------------	--

---

**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 Bonafè

---

dbqa.isrrqa	<i>controlla se una stazione e' in RRQA</i>
-------------	---

---

**Description**

controlla se una stazione e' parte della rete RRQA

**Usage**

```
dbqa.isrrqa(con, Id)
```

**Arguments**

con	connessione al DB
Id	codice identificativo (o un vettore o una lista di codici)



---

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 Bonafè

---

dbqa.list.fields *Elenca i campi presenti in una tabella del DB*

---

**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 Bonafè, Arpa Emilia-Romagna

---

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

---

dbqa.round	<i>arrotonda concentrazioni</i>
------------	---------------------------------

---

**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 Bonafè, 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

<code>thr.multihourly</code>	soglia per superamenti orari di più ore consecutive
<code>NH</code>	numero di ore consecutive di superamento da contare se <code>thr.multihourly</code> non è NULL
<code>critical.months</code>	vettore numerico dei mesi su cui calcolare la media di periodo (p.es. <code>c(1:3,10:12)</code> per la media invernale)
<code>AR</code>	lista di <code>data.frame</code> prodotta da <code>calculate.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.annual_report</code> li passa a <code>dbqa.get.datastaz</code> ; la funzione <code>write.annual_report</code> li passa a <code>dbqa.insert</code> .

### Value

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

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

se `thr.daily.ave` non è NULL allora `data.frame` ha anche:

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

se `thr.ave8h.max` non è NULL allora `data.frame` ha anche:

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

<code>hourly.nValid</code>	numero di dati orari validi
<code>hourly.percValid</code>	percentuale di dati orari validi

se `thr.hourly` non è NULL allora `data.frame` ha anche:

hourly.nexc      numero di superamenti orari

se thr.multiphourly non è NULL allora data.frame ha anche:

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

se critical.months non è 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.encyency  
                 rapporto tra numero di dati disponibili e attesi nei mesi selezionati

### Author(s)

Giovanni Bonafè, 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` è 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` è 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 Bonafè, 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 ? 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 ? 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 Bonafè, 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`*Costruisce una semplice query SQL.*

---

**Description**

Costruisce una semplice query SQL.

**Usage**

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

**Arguments**

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

---

`squeeze`*Dato un vettore numerico, lo rappresenta come stringa sintetica*

---

**Description**

Dato un vettore di interi, ne dà 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	<i>Funzioni che gestiscono le date, ore, ecc.</i>
----------------	---

---

**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

---

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
...	

---

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, 12
- ozone\_annual\_report, 14
- ozone\_daily\_report, 16

## \*Topic \textasciitildeekwd1

- aot, 2
- aqstat.functions, 3
- dbqa.config, 4
- dbqa.connect, 4
- dbqa.data2xts, 5
- dbqa.delete, 5
- dbqa.get.datasens, 6
- dbqa.get.datastaz, 7
- dbqa.get.idcfsens, 7
- dbqa.insert, 8
- dbqa.list.active.staz, 9
- dbqa.list.fields, 9
- dbqa.list.tables, 10
- dbqa.round, 10
- dbqa.view.param, 11
- dbqa.view.staz, 11
- round.awayfromzero, 17
- simple.query, 18
- squeeze, 18
- time.functions, 19
- xts.blend, 19
- xts.regularize, 20

## \*Topic \textasciitildeekwd2

- aot, 2
- aqstat.functions, 3
- dbqa.config, 4
- dbqa.connect, 4
- dbqa.data2xts, 5
- dbqa.delete, 5
- dbqa.get.datasens, 6
- dbqa.get.datastaz, 7
- dbqa.get.idcfsens, 7
- dbqa.insert, 8
- dbqa.list.active.staz, 9

- dbqa.list.fields, 9
- dbqa.list.tables, 10
- dbqa.round, 10
- dbqa.view.param, 11
- dbqa.view.staz, 11
- round.awayfromzero, 17
- simple.query, 18
- squeeze, 18
- time.functions, 19
- xts.blend, 19
- xts.regularize, 20

## \*Topic \textasciitildeozone

- generic\_annual\_report, 12
- ozone\_annual\_report, 14
- ozone\_daily\_report, 16

- aot, 2

- aqstat.functions, 3

- calculate.annual\_report  
(generic\_annual\_report), 12
- calculate.ozone\_annual\_report  
(ozone\_annual\_report), 14
- calculate.ozone\_daily\_report  
(ozone\_daily\_report), 16

- dbqa.config, 4
- dbqa.connect, 4
- dbqa.data2xts, 5
- dbqa.delete, 5
- dbqa.get.datasens, 6
- dbqa.get.datastaz, 7
- dbqa.get.idcfsens, 7
- dbqa.insert, 8
- dbqa.isrrqa, 8
- dbqa.list.active.staz, 9
- dbqa.list.fields, 9
- dbqa.list.tables, 10
- dbqa.round, 10
- dbqa.view.param, 11

dbqa.view.staz, [11](#)  
detect.event(aqstat.functions), [3](#)  
exc.period(aqstat.functions), [3](#)  
generic\_annual\_report, [12](#)  
Hour(time.functions), [19](#)  
mean.window(aqstat.functions), [3](#)  
Month(time.functions), [19](#)  
Ndays(time.functions), [19](#)  
Nmonths(time.functions), [19](#)  
ozone\_annual\_report, [14](#)  
ozone\_daily\_report, [16](#)  
prepare.annual\_report  
    (generic\_annual\_report), [12](#)  
prepare.ozone\_annual\_report  
    (ozone\_annual\_report), [14](#)  
prepare.ozone\_daily\_report  
    (ozone\_daily\_report), [16](#)  
round.awayfromzero, [17](#)  
shift(aqstat.functions), [3](#)  
simple.query, [18](#)  
squeeze, [18](#)  
stat.period(aqstat.functions), [3](#)  
stat.period2(aqstat.functions), [3](#)  
stat.window(aqstat.functions), [3](#)  
time.functions, [19](#)  
which.period(aqstat.functions), [3](#)  
write.annual\_report  
    (generic\_annual\_report), [12](#)  
write.ozone\_annual\_report  
    (ozone\_annual\_report), [14](#)  
write.ozone\_daily\_report  
    (ozone\_daily\_report), [16](#)  
xts.blend, [19](#)  
xts.regolarize, [20](#)  
Year(time.functions), [19](#)  
Ym(time.functions), [19](#)  
Ymd(time.functions), [19](#)  
YQ(time.functions), [19](#)