

Package ‘hydrosanity’

April 17, 2009

Type Package

Title Graphical user interface for exploring hydrological time series

Version 0.8.76

Date 2007-12-08

Author Felix Andrews <felix@nfrac.org>

Maintainer Felix Andrews <felix@nfrac.org>

Depends R (>= 2.5.0), playwith (>= 0.8.24), lattice (>= 0.16)

Imports RGtk2, gWidgets, reshape, grid, grDevices, stats, utils

Suggests sp, rgdal, tripack, gpclib, akima, oz, maps, mapdata, maptools

Description Hydrosanity provides a graphical user interface for exploring hydrological time series. It is designed to work with catchment surface hydrology data (mainly rainfall and streamflow time series at a set of locations). There are functions to import from a database or files; summarise and visualise the dataset in various ways; estimate areal rainfall; fill gaps in rainfall data; and estimate the rainfall-runoff relationship. Probably the most useful features are the interactive graphical displays of a spatial set of time series. **WARNING:** this package is under development and should not be considered stable. An introductory paper is included, but there is not much detailed documentation. Hydrosanity’s Graphical User Interface was based on Rattle by Graham Williams.

License GPL (>= 2)

URL <http://hydrosanity.googlecode.com/>

Repository CRAN

Date/Publication 2007-12-08 11:03:52

R topics documented:

aggregate.timeblob	2
arealSubPolygons	3
gaps	4
grid.timeline.plot	5
grid.timeseries.plot	6
grid.xaxis.POSIXt	8
grid.yaxis.log	9
hydrosanity	10
hydrosanity.caption	11
impute.timeblobs	12
panel.levelplot.interp	14
read.timeblob	15
smooth.timeblob	17
summaryMissing.timeblobs	18
sync.timeblobs	19
timeblob	20
truncMonth	21
window.timeblob	22
Index	24

aggregate.timeblob *Aggregate to Longer Timesteps*

Description

~~ A concise (1-5 lines) description of what the function does. ~~

Usage

```
## S3 method for class 'timeblob':
aggregate(x, by = "1 year", FUN = mean, fun.qual = c("worst", "median", "mode", "on
```

Arguments

```
x           ~~Describe blob here~~
by          ~~Describe by here~~
FUN         ~~Describe FUN here~~
fun.qual    ~~Describe fun.qual here~~
start.month ~~Describe start.month here~~
...        ~~Describe ... here~~
```

Details

~~ If necessary, more details than the description above ~~

Value

~Describe the value returned If it is a LIST, use

comp1 Description of 'comp1'

comp2 Description of 'comp2'

...

Author(s)

~~who you are~~

References

~put references to the literature/web site here ~

See Also

~~objects to See Also as [help](#), ~~~

arealSubPolygons *Compute sub-regions closest to each point*

Description

Compute the Voronoi mosaic of a set of points, and intersect it with a given polygon. This gives the sub-regions of the polygon for which each point is the closest.

Usage

```
arealSubPolygons(x, y = NULL, IDs = row.names(x), boundary, min.area.pct = 0.5)
```

Arguments

x x coordinate of points, or a list containing x and y.

y y coordinate of points.

IDs identification strings for each point.

boundary polygon coordinates (coerced to "gpc.poly" class).

min.area.pct minimum percentage area of the polygon to be allocated to any one site. Sites which would have less area than this will be excluded.

Details

Voronoi mosaic, AKA Thiessen polygons, AKA Dirichlet tessellation.

Value

An object of class `SpatialPolygons` with ID slots from the original `IDs` argument.

Author(s)

Felix Andrews (felix@nfrac.org)

See Also

`voronoi.mosaic`, `gpc.poly-class`, `SpatialPolygons`

Examples

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

gaps

Find Runs of Consecutive Missing Values, etc

Description

~~ A concise (1-5 lines) description of what the function does. ~~

Usage

```
gaps(x, max.length = Inf, internal.only = T)

expand.indices(info)

rises(x)

lastTime(x)
```

Arguments

```
x           ~~Describe x here~~
internal.only
            ~~Describe internal.only here~~
max.length  ~~Describe max.length here~~
info        ~~Describe info here~~
```

Details

~~ If necessary, more details than the description above ~~

Value

~Describe the value returned If it is a LIST, use

comp1 Description of 'comp1'

comp2 Description of 'comp2'

...

Author(s)

~~who you are~~

References

~put references to the literature/web site here ~

See Also

~~objects to See Also as [help](#), ~~~

grid.timeline.plot *Plot Timeline of Data and its Quality*

Description

~~ A concise (1-5 lines) description of what the function does. ~~

Usage

```
grid.timeline.plot(blob.list, xlim = NULL, ylim=NULL, colMap = NULL, barThickness =
```

```
grid.timeline.bar <- function(blob, colMap = NULL, name = "timeline.bar", vp = NULL
```

```
timelineColMapDefault(colMap = list(good="black", suspect=trellis.par.get("superpos
```

Arguments

blob.list ~~Describe blob.list here~~

xlim ~~Describe xlim here~~

ylim ~~Describe ylim here~~

colMap ~~Describe colMap here~~

barThickness ~~Describe thickness here~~

auto.key ~~Describe auto.key here~~

maxLabelChars ~~Describe maxLabelChars here~~

pad ~~Describe pad here~~

grill ~~Describe grill here~~
main ~~Describe main here~~
sub ~~Describe sub here~~
newpage ~~Describe newpage here~~
blob ~~Describe blob here~~
name ~~Describe name here~~
vp ~~Describe vp here~~

Details

~~ If necessary, more details than the description above ~~

Value

~Describe the value returned If it is a LIST, use

comp1 Description of 'comp1'
comp2 Description of 'comp2'

...

Author(s)

~~who you are~~

References

~put references to the literature/web site here ~

See Also

~~objects to See Also as [help](#), ~~~

grid.timeseries.plot

Plot Time Series

Description

~~ A concise (1-5 lines) description of what the function does. ~~

Usage

```

grid.timeseries.plot(blob.list, xlim = NULL, ylim = NULL, sameScales = T,
  logScale = F, qualTimeline = F, colMap = NULL,
  barThickness = unit(0.5, "lines"), auto.key = T, maxLabelChars
  pad = unit(1, "lines"), between = unit(0, "lines"), superPos = 1,
  newScale = T, main = NULL, sub = T, newpage = (superPos==1),
  nSuperpose = 1, gp=gpar(col=rep(trellis.par.get("superpose.lin
    alpha=rep(trellis.par.get("superpose.line")$alpha, len=super
    lty=rep(trellis.par.get("superpose.line")$lty, len=superPos)
    lwd=rep(trellis.par.get("superpose.line")$lwd, len=superPos)

grid.timeseries.plot.superpose(superpose.blob.list, allSameScales = F, xlim = NULL,

grid.timeseries.steps(blob, logScale = F, name = "timeseries", gp = NULL, vp = NULL

```

Arguments

```

blob.list      ~~Describe blob.list here~~
xlim           ~~Describe xlim here~~
ylim           ~~Describe ylim here~~
sameScales     ~~Describe sameScales here~~
logScale       ~~Describe logScale here~~
qualTimeline   ~~Describe qualTimeline here~~
colMap         ~~Describe colMap here~~
barThickness   ~~Describe thickness here~~
auto.key       ~~Describe auto.key here~~
maxLabelChars  ~~Describe maxLabelChars here~~
pad            ~~Describe pad here~~
between        ~~Describe between here~~
superPos       ~~Describe superPos here~~
newScale       ~~Describe newScale here~~
main           ~~Describe main here~~
sub            ~~Describe sub here~~
newpage        ~~Describe newpage here~~
nSuperpose     ~~Describe nSuperpose here~~
gp             ~~Describe gp here~~
superpose.blob.list
               ~~Describe superpose.blob.list here~~
allSameScales  ~~Describe allSameScales here~~
...            ~~Describe ... here~~

```

```
blob      ~~Describe blob here~~
name      ~~Describe blob here~~
vp        ~~Describe blob here~~
```

Details

~~ If necessary, more details than the description above ~~

Value

~Describe the value returned If it is a LIST, use

```
comp1      Description of 'comp1'
comp2      Description of 'comp2'
...

```

Author(s)

~~who you are~~

References

~put references to the literature/web site here ~

See Also

~~objects to See Also as [help](#), ~~~

grid.xaxis.POSIXt *Time Axis for Grid*

Description

~~ A concise (1-5 lines) description of what the function does. ~~

Usage

```
grid.xaxis.POSIXt(lim = convertX(unit(0:1, "npc"), "native", valueOnly = T), label
```

Arguments

```
lim        ~~Describe lim here~~
label      ~~Describe label here~~
draw       ~~Describe draw here~~
name       ~~Describe name here~~
...        ~~Describe ... here~~
```

Details

~~ If necessary, more details than the description above ~~

Value

~Describe the value returned If it is a LIST, use

comp1 Description of 'comp1'

comp2 Description of 'comp2'

...

Author(s)

~~who you are~~

References

~put references to the literature/web site here ~

See Also

~~objects to See Also as [help](#), ~~~

grid.yaxis.log *Log Scale Axis for Grid/Lattice*

Description

~~ A concise (1-5 lines) description of what the function does. ~~

Usage

```
grid.yaxis.log(logLim = as.numeric(convertY(unit(c(0,1), "npc"), "native")), label  
grid.xaxis.log(logLim = as.numeric(convertX(unit(c(0,1), "npc"), "native")), label
```

```
lattice.y.prettylog(lim, ...)  
lattice.x.prettylog(lim, ...)
```

```
lattice.y.sqrt(lim, ...)  
lattice.x.sqrt(lim, ...)
```

Arguments

logLim ~~Describe logLim here~~
 label ~~Describe label here~~
 draw ~~Describe draw here~~
 name ~~Describe name here~~
 lim ~~Describe lim here~~
 ... ~~Describe ... here~~

Details

~~ If necessary, more details than the description above ~~

Value

~Describe the value returned If it is a LIST, use

comp1 Description of 'comp1'
 comp2 Description of 'comp2'
 ...

Author(s)

~~who you are~~

References

~put references to the literature/web site here ~

See Also

~~objects to See Also as [help](#), ~~~

hydrosanity

Start the Hydrosanity GUI

Description

Start the Hydrosanity Graphical User Interface.

Usage

hydrosanity(project = NULL)

Arguments

project Path to a hydrosanity project file to open.

Details

...should list available documentation here.

Author(s)

Felix Andrews <felix@nfrac.org>

References

Andrews, F. (2007). Hydrosanity: a starting point for hydrological analysis. To appear in *Proceedings of MODSIM 2007 - International Congress on Modelling and Simulation, Christchurch, New Zealand, 10-13 December 2007*. Modelling and Simulation Society of Australia and New Zealand Inc.

Package home page: <http://hydrosanity.googlecode.com/>

Examples

```
## Not run:
hydrosanity()

# view the introductory paper
RShowDoc("Andrews_2007_Hydrosanity", package="hydrosanity")

## End(Not run)
```

hydrosanity.caption
Metadata caption for plots

Description

~~ A concise (1-5 lines) description of what the function does. ~~

Usage

```
hydrosanity.caption(timelim, by, n, series = NA, x = unit(1, "npc") - unit(1, "mm"))
```

Arguments

- timelim ~~Describe timelim here~~
- by ~~Describe by here~~
- n ~~Describe n here~~
- series ~~Describe series here~~
- x ~~Describe x here~~
- y ~~Describe y here~~
- just ~~Describe just here~~
- gp ~~Describe gp here~~

Details

~~ If necessary, more details than the description above ~~

Value

~Describe the value returned If it is a LIST, use

comp1 Description of 'comp1'

comp2 Description of 'comp2'

...

Author(s)

~~who you are~~

References

~put references to the literature/web site here ~

See Also

~~objects to See Also as [help](#), ~~~

`impute.timeblobs` *Fill Gaps in Time Series*

Description

~~ A concise (1-5 lines) description of what the function does. ~~

Usage

```
impute.timeblobs(blob.list, which.impute = names(blob.list), timelim = NULL, extend
```

```
imputeGaps.timeblobs(blob.list, which.impute = names(blob.list), type = c("disaccum
```

```
quick.disaccumulate.timeblob(blob)
```

```
unimputeGaps.timeblobs(blob.list, timelim = NULL, type = c("imputed", "disaccumulat
```

Arguments

blob.list ~~Describe blob.list here~~
which.impute ~~Describe which.impute here~~
timelim ~~Describe timelim here~~
extend ~~Describe extend here~~
withinTimeframe
 ~~Describe withinTimeframe here~~
method ~~Describe method here~~
constant ~~Describe constant here~~
trim ~~Describe trim here~~
type ~~Describe type here~~
fallBackToConstantDisaccum
 ~~Describe fallBackToConstantDisaccum here~~
maxGapLength ~~Describe maxGapLength here~~
... ~~Describe ... here~~
blob ~~Describe blob here~~

Details

~~ If necessary, more details than the description above ~~

Value

~Describe the value returned If it is a LIST, use

comp1 Description of 'comp1'
comp2 Description of 'comp2'
...

Author(s)

~~who you are~~

References

~put references to the literature/web site here ~

See Also

~~objects to See Also as [help](#), ~~~

panel.levelplot.interp

Panel functions for spatial layers

Description

Functions for plot-time spatial interpolation, which can be used as panel functions for levelplot. Also some generally useful spatial layers, assuming a latitude-longitude coordinate system.

Usage

```
panel.levelplot.mosaic <- function(x, y, z, subscripts = T,
                                  at = seq(min(z, na.rm = T), max(z, na.rm = T), length = 100),
                                  col.regions = regions$col)
```

```
panel.levelplot.interp(x, y, z, subscripts = T, xo.length = 40, yo.length = xo.length,
                      linear = T, extrap = F, contour = F, region = T, at, ...)
```

```
panel.contourplot.interp(..., contour = T, region = F)
```

```
panel.worldmap(col = "black", ...)
panel.rivers(col = "blue", lty = "longdash", ...)
panel.cities(pch = 15, col = "black", ...)
```

```
prepanel.extend.10(...)
```

Arguments

<code>x, y, z</code>	<code>x</code> and <code>y</code> coordinates and <code>z</code> value to be interpolated.
<code>subscripts</code>	used by Lattice for conditioning.
<code>at</code>	<code>z</code> values at which to indicate changes. May be omitted, see details below.
<code>col.regions</code>	a vector of colours representing levels. See levelplot .
<code>xo.length, yo.length</code>	resolution of interpolated surface. The plot region is divided into <code>xo.length * yo.length</code> cells.
<code>linear</code>	use bicubic rather than linear interpolation.
<code>extrap</code>	for <code>linear=F</code> , use spatial extrapolation outside the convex hull of the data.
<code>contour</code>	draw contour lines (passed to <code>panel.levelplot</code>).
<code>region</code>	draw shaded image (passed to <code>panel.levelplot</code>).
<code>...</code>	further arguments passed to <code>panel.levelplot</code> .
<code>col, lty, pch</code>	passed on to the usual drawing functions.

Details

For `panel.levelplot.interp`, if the `at` argument is missing it is taken as `pretty(z)` if `contour=T` and a 100-point sequence between the ranges of `z` otherwise.

Value

~Describe the value returned If it is a LIST, use

`comp1` Description of 'comp1'

`comp2` Description of 'comp2'

...

Author(s)

Felix Andrews (felix@nfrac.org)

See Also

[levelplot](#), [voronoi.mosaic](#), [interp](#), [map](#)

Examples

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

`read.timeblob` *Read Timeblob from a Text File*

Description

Read in a timeblob (time series) from a file.

Usage

```
read.timeblob(file, skip = 1, sep = ",", sitename = NULL, dataname = "Data",
              dataCol = 2, qualCol = 3, extraCols = c(),
              extraNames = paste("Extra", extraCols), readTimesFromFile = T,
              timeCol = 1, timeFormat = "%d %b %Y", startTime = NA,
              tz = "GMT", timeSeqBy = "days", timeOffset = NULL, ...)
```

Arguments

file	path to a file, or a connection.
skip	number of lines in file before data begins.
sep	column separator, see read.table .
sitename	site name attribute.
dataname	data name attribute.
dataCol	file column number (starting from 1) containing data.
qualCol	file column number (starting from 1) containing quality codes.
extraCols	a vector of other file column numbers to read in.
extraNames	names of the columns given above.
readTimesFromFile	whether to read time series times from the file (column timeCol).
timeCol	used when readTimesFromFile = TRUE. It should give the file column number (starting from 1) containing times.
timeFormat	format of times in timeCol: see strptime .
startTime	used when readTimesFromFile = FALSE. It should be (coercible to) a POSIXt object, or else be a list like <code>list(year=1, month=2, day=3)</code> (etc) where these give column numbers to read the start time components (from the first line in file).
tz	timezone, see POSIXt .
timeSeqBy	time step to go with startTime, passed to seq.POSIXt .
timeOffset	a difftime object to add to the times.
...	passed to read.table .

Value

An object of class `timeblob`. It is just a [data.frame](#) with some extra attributes.

Author(s)

Felix Andrews (felix@nfrac.org)

See Also

[read.table](#)

smooth.timeblob *Apply smoothing filter to a timeseries object*

Description

~~ A concise (1-5 lines) description of what the function does. ~~

Usage

```
smooth.timeblob(blob, by = "1 year")
```

Arguments

blob ~~Describe blob here~~
by ~~Describe by here~~

Details

~~ If necessary, more details than the description above ~~

Value

~Describe the value returned If it is a LIST, use

comp1 Description of 'comp1'
comp2 Description of 'comp2'
...

Author(s)

~~who you are~~

References

~put references to the literature/web site here ~

See Also

~~objects to See Also as [help](#), ~~~

```
summaryMissing.timeblobs
```

Summarise Missing Data in a Set of Timeblobs

Description

~~ A concise (1-5 lines) description of what the function does. ~~

Usage

```
summaryMissing.timeblobs(blob.list, timelim = NULL, timestep = NULL)
```

Arguments

```
blob.list    ~~Describe blob.list here~~  
timelim      ~~Describe timelim here~~  
timestep     ~~Describe timestep here~~
```

Details

~~ If necessary, more details than the description above ~~

Value

~Describe the value returned If it is a LIST, use

```
comp1        Description of 'comp1'  
comp2        Description of 'comp2'  
...
```

Author(s)

~~who you are~~

References

~put references to the literature/web site here ~

See Also

~~objects to See Also as [help](#), ~~~

sync.timeblobs *Synchronise Time Series (Timeblobs)*

Description

~~ A concise (1-5 lines) description of what the function does. ~~

Usage

```
sync.timeblobs(blob.list, timestep = NULL, timelim = NULL, extractColumn = "Data")
syncTo.timeblobs(blob.list, blob, extractColumn = "Data")
matchtimes.timeblob(blob, times)
common.timestep.timeblobs(blob.list, default = "DSTdays")
```

Arguments

```
blob.list      ~~Describe blob.list here~~
timestep      ~~Describe timestep here~~
timelim        ~~Describe timelim here~~
extractColumn      ~~Describe extractColumn here~~
blob            ~~Describe blob here~~
times          ~~Describe times here~~
default        ~~Describe default here~~
```

Details

~~ If necessary, more details than the description above ~~

Value

~Describe the value returned If it is a LIST, use

```
comp1          Description of 'comp1'
comp2          Description of 'comp2'
```

...

Author(s)

~~who you are~~

References

~put references to the literature/web site here ~

See Also

~~objects to See Also as [help](#), ~~~

timeblob

Time Series Object (Timeblob) with Times and Data

Description

~~ A concise (1-5 lines) description of what the function does. ~~

Usage

```
timeblob(Time, Data, Qual = NULL, extras = NULL, timestep = NULL, sitename = "Unkn
```

```
is.timeblob(x)
```

```
lapply.timeblob.data(blob.list, FUN, ...)
```

```
sapply.timeblob.data(blob.list, FUN, ...)
```

Arguments

Time	~~Describe Time here~~
Data	~~Describe Data here~~
Qual	~~Describe Qual here~~
extras	~~Describe extras here~~
timestep	~~Describe timestep here~~
sitename	~~Describe sitename here~~
dataname	~~Describe dataname here~~
role	~~Describe role here~~
x	~~Describe x here~~
blob.list	~~Describe blob.list here~~
FUN	~~Describe FUN here~~
...	~~Describe ... here~~

Details

~~ If necessary, more details than the description above ~~

Value

~Describe the value returned If it is a LIST, use

comp1 Description of 'comp1'

comp2 Description of 'comp2'

...

Author(s)

~~who you are~~

References

~put references to the literature/web site here ~

See Also

~~objects to See Also as [help](#), ~~~

truncMonth	<i>Round Down to a Month, Year or Decade</i>
------------	--

Description

~~ A concise (1-5 lines) description of what the function does. ~~

Usage

truncMonth(x)
truncYear(x)
truncDecade(x)

Arguments

x ~~Describe x here~~

Details

~~ If necessary, more details than the description above ~~

Value

~Describe the value returned If it is a LIST, use

comp1 Description of 'comp1'

comp2 Description of 'comp2'

...

Author(s)

~~who you are~~

References

~put references to the literature/web site here ~

See Also

~~objects to See Also as [help](#), ~~~

window.timeblob *Time Windows of Time Series (Timeblobs)*

Description

~~ A concise (1-5 lines) description of what the function does. ~~

Usage

```
## S3 method for class 'timeblob':
window(x, start = NULL, end = NULL, inclusive = F, return.indices = F, extend = F,

## S3 method for class 'timeblob':
start(x, ...)
## S3 method for class 'timeblob':
end(x, ...)

start.timeblobs(x, ...)
end.timeblobs(x, ...)
timelim.timeblobs(x)
```

Arguments

```
x                   ~~Describe x here~~
start                ~~Describe start here~~
end                   ~~Describe end here~~
inclusive            ~~Describe inclusive here~~
return.indices        ~~Describe return.indices here~~
extend                ~~Describe extend here~~
...                   ~~Describe ... here~~
```

Details

~~ If necessary, more details than the description above ~~

Value

~Describe the value returned If it is a LIST, use

comp1 Description of 'comp1'

comp2 Description of 'comp2'

... returns length 0 if blob is empty

Author(s)

~~who you are~~

References

~put references to the literature/web site here ~

See Also

~~objects to See Also as [help](#), ~~~

Index

*Topic **aplot**

hydrosanitiy.caption, 11
panel.levelplot.interp, 13

*Topic **dplot**

grid.xaxis.POSIXt, 8
grid.yaxis.log, 9
panel.levelplot.interp, 13

*Topic **environment**

hydrosanitiy, 10

*Topic **spatial**

arealSubPolygons, 3

*Topic **ts**

aggregate.timeblob, 2
gaps, 4
grid.timeline.plot, 5
grid.timeseries.plot, 6
grid.xaxis.POSIXt, 8
impute.timeblobs, 12
read.timeblob, 14
smooth.timeblob, 16
summaryMissing.timeblobs, 17
sync.timeblobs, 18
timeblob, 19
truncMonth, 20
window.timeblob, 21

aggregate.timeblob, 2

arealSubPolygons, 3

common.timestep.timeblobs
(sync.timeblobs), 18

data.frame, 15

difftime, 15

end.timeblob (window.timeblob), 21

end.timeblobs (window.timeblob),
21

expand.indices (gaps), 4

gaps, 4

gpc.poly-class, 3

grid.timeline.bar
(grid.timeline.plot), 5

grid.timeline.plot, 5

grid.timeseries.plot, 6

grid.timeseries.steps
(grid.timeseries.plot), 6

grid.xaxis.log (grid.yaxis.log), 9

grid.xaxis.POSIXt, 8

grid.yaxis.log, 9

help, 2, 4, 6–8, 10, 11, 13, 16, 17, 19–22

hydrosanitiy, 10

hydrosanitiy.caption, 11

impute.timeblobs, 12

imputeGaps.timeblobs
(impute.timeblobs), 12

interp, 14

is.timeblob (timeblob), 19

lapply.timeblob.data (timeblob),
19

lastTime (gaps), 4

lattice.x.prettylog
(grid.yaxis.log), 9

lattice.x.sqrt (grid.yaxis.log), 9

lattice.y.prettylog
(grid.yaxis.log), 9

lattice.y.sqrt (grid.yaxis.log), 9

levelplot, 13, 14

map, 14

matchtimes.timeblob
(sync.timeblobs), 18

panel.cities
(panel.levelplot.interp),
13

`panel.contourplot.interp`
 (`panel.levelplot.interp`),
 13

`panel.levelplot.interp`, 13

`panel.levelplot.mosaic`
 (`panel.levelplot.interp`),
 13

`panel.rivers`
 (`panel.levelplot.interp`),
 13

`panel.worldmap`
 (`panel.levelplot.interp`),
 13

POSIXt, 15

`prepanel.extend.10`
 (`panel.levelplot.interp`),
 13

`quick.disaccumulate.timeblob`
 (`impute.timeblobs`), 12

`read.table`, 15

`read.timeblob`, 14

`rises(gaps)`, 4

`sapply.timeblob.data(timeblob)`,
 19

`seq.POSIXt`, 15

`smooth.timeblob`, 16

SpatialPolygons, 3

`start.timeblob(window.timeblob)`,
 21

`start.timeblobs`
 (`window.timeblob`), 21

`strptime`, 15

`summaryMissing.timeblobs`, 17

`sync.timeblobs`, 18

`syncTo.timeblobs`
 (`sync.timeblobs`), 18

`timeblob`, 19

`timelim.timeblobs`
 (`window.timeblob`), 21

`timelineColMapDefault`
 (`grid.timeline.plot`), 5

`truncDecade(truncMonth)`, 20

`truncMonth`, 20

`truncYear(truncMonth)`, 20

`unimputeGaps.timeblobs`
 (`impute.timeblobs`), 12

`voronoi.mosaic`, 3, 14

`window.timeblob`, 21