

Package ‘rsatscan’

March 17, 2015

Title Tools, Classes, and Methods for Interfacing with SaTScan
Stand-Alone Software

Description SaTScan(TM) (<http://www.satscan.org>) is software for finding regions in Time, Space, or Time-Space that have excess risk, based on scan statistics, and uses Monte Carlo hypothesis testing to generate P-values for these regions. The rsatscan package provides functions for writing R data frames in SaTScan-readable formats, for setting SaTScan parameters, for running SaTScan in the OS, and for reading the files that SaTScan creates.

Version 0.3.9200

Date 2015-03-17

URL <http://www.satscan.org>

Maintainer Ken Kleinman <ken.kleinman@gmail.com>

Depends R (>= 3.0.2)

Imports utils, foreign

Suggests knitr, rgdal (>= 0.9.1), sp (>= 1.0-9)

VignetteBuilder knitr

License GPL-3

LazyData true

NeedsCompilation no

Author Ken Kleinman [aut, cre]

Repository CRAN

Date/Publication 2015-03-17 18:02:49

R topics documented:

charlistopts	2
NHumbersidecas	3
NHumbersidectl	3
NHumbersidegeo	4
NMcas	4

NMgeo	5
NMpop	5
NYCfevercas	6
NYCfevergeo	6
print.satscan	7
read.col	7
read.gis	8
read.llr	8
read.rr	9
read.satscanmain	9
read.sci	10
rsatscan	11
satscan	11
ss.options	13
ss.options.extra	14
subin	14
summary.satscan	15
write.adj	15
write.cas	16
write.ctl	17
write.geo	17
write.grd	18
write.ha	18
write.max	19
write.met	19
write.nbr	20
write.pop	20
write.ss.prm	21
Index	22

charlistopts	<i>Change list version of paramaters into char vector</i>
--------------	---

Description

Turns a list of options into a charvar of options

Usage

charlistopts(x)

Arguments

x A list.

Details

The resulting charvar has values such as "name=value" where "name" was the named item of the list.

Value

A character vector

Not expected to be used directly.

NHumbersidecas

North Humberside leukemian and lymphoma example– cases

Description

A data set from North Humberside. The variables are as follows:

Format

A data frame with 191 observations and 2 variables

Details

- locationid: Postal code ID
- numcases: The number of cases observed

Source

Distributed with SaTScan software: <http://www.satscan.org>

NHumbersidectl

North Humberside leukemian and lymphoma example– controls

Description

A data set from North Humberside. The variables are as follows:

Format

A data frame with 191 observations and 2 variables

Details

- locationid: Postal code ID
- numcontrols: The number of controls observed

Source

Distributed with SaTScan software: <http://www.satscan.org>

 NHumbersidegeo

North Humberside leukemian and lymphoma example– geography

Description

A data set from North Humberside. The variables are as follows:

Format

A data frame with 191 observations and 3 variables

Details

- locationid: Postal code ID
- x-coordinate: x-coordinate
- y-coordinate: y-coordinate

Source

Distributed with SaTScan software: <http://www.satscan.org>

 NMcas

New Mexico Brain Cancer example– cases

Description

A data set from New Mexico. The variables are as follows:

Format

A data frame with 1175 observations and 5 variables

Details

- county: County name
- cases: Number of cases
- year: Year of case
- agegroup: Age group of case
- sex: Sex of case

Source

Distributed with SaTScan software: <http://www.satscan.org>

NMgeo	<i>New Mexico Brain Cancer example– geography</i>
-------	---

Description

A data set from New Mexico. The variables are as follows:

Format

A data frame with 194 observations and 3 variables

Details

- county: The US Postal Service ZIP code
- cases: The number of cases observed
- long: The date on which the cases were observed in that ZIP code

Source

Distributed with SaTScan software: <http://www.satscan.org>

NMpop	<i>New Mexico Brain Cancer example– population</i>
-------	--

Description

A data set from New Mexico. The variables are as follows:

Format

A data frame with 3456 observations and 5 variables

Details

- county: County name
- year: year
- year: population
- agegroup: Age group
- sex: Sex

Source

Distributed with SaTScan software: <http://www.satscan.org>

NYCfevercas

New York City Fever example– cases

Description

A data set from New York City. The variables are as follows:

Format

A data frame with 194 observations and 3 variables

Details

- zip: The US Postal Service ZIP code
- cases: The number of cases observed
- long: The date on which the cases were observed in that ZIP code

Source

Distributed with SaTScan software: <http://www.satscan.org>

NYCfevergeo

New York City Fever example– geography

Description

A data set from New York City. The variables are as follows:

Format

A data frame with 192 observations and 3 variables

Details

- zip: The US Postal Service ZIP code
- lat: Decimal latitude north
- long: Decimal longitude

Source

Distributed with SaTScan software: <http://www.satscan.org>

print.satscan	<i>Methods for satscan-class objects</i>
---------------	--

Description

These functions define the default methods for satscan-class objects, which are the result objects from a call to satscan()

Usage

```
## S3 method for class 'satscan'
print(x, ...)
```

Arguments

x	is a satscan object
...	vestigial, for compatability with the default summary method

Value

x, invisibly. Side effect is to display ss\$main, the SaTScan text report

read.col	<i>Read SaTScan output files</i>
----------	----------------------------------

Description

Reads a SaTScan output .dbf file.

Usage

```
read.col(location, file)
```

Arguments

location	A directory location, including the trailing "/"
file	A file name, without the extension.

Details

This is expected to be a purely internal function. It's called by satscan() with the location and file name provided to that function. Since it's nothing more than foreign::read.dbf(), it's probably not necessary to even have it as a function.

Value

A data frame.

read.gis	<i>Read SaTScan output files</i>
----------	----------------------------------

Description

Reads a SaTScan output .dbf file.

Usage

```
read.gis(location, file)
```

Arguments

location	A directory location, including the trailing "/"
file	A file name, without the extension.

Details

This is expected to be a purely internal function. It's called by `satscan()` with the location and file name provided to that function. Since it's nothing more than `foreign::read.dbf()`, it's probably not necessary to even have it as a function.

Value

A data frame.

read.llr	<i>Read SaTScan output files</i>
----------	----------------------------------

Description

Reads a SaTScan output .dbf file.

Usage

```
read.llr(location, file)
```

Arguments

location	A directory location, including the trailing "/"
file	A file name, without the extension.

Details

This is expected to be a purely internal function. It's called by `satscan()` with the location and file name provided to that function. Since it's nothing more than `foreign::read.dbf()`, it's probably not necessary to even have it as a function.

Value

A data frame.

read.rr	<i>Read SaTScan output files</i>
---------	----------------------------------

Description

Reads a SaTScan output .dbf file.

Usage

```
read.rr(location, file)
```

Arguments

location	A directory location, including the trailing "/"
file	A file name, without the extension.

Details

This is expected to be a purely internal function. It's called by `satscan()` with the location and file name provided to that function. Since it's nothing more than `foreign::read.dbf()`, it's probably not necessary to even have it as a function.

Value

A data frame.

read.satscanmain	<i>Read SaTScan output files</i>
------------------	----------------------------------

Description

Reads a SaTScan output .dbf file.

Usage

```
read.satscanmain(location, file)
```

Arguments

location	A directory location, including the trailing "/"
file	A file name, without the extension.

Details

This is expected to be a purely internal function. It's called by `satscan()` with the location and file `nmae` provided to that function. Since it's nothing more than `readLines()`, it's probably not necessary to even have it as a function.

Value

A data frame.

<code>read.sci</code>	<i>Read SaTScan output files</i>
-----------------------	----------------------------------

Description

Reads a SaTScan output `.dbf` file.

Usage

```
read.sci(location, file)
```

Arguments

- | | |
|-----------------------|---|
| <code>location</code> | A directory location, including the trailing <code>"/"</code> |
| <code>file</code> | A file name, without the extension. |

Details

This is expected to be a purely internal function. It's called by `satscan()` with the location and file `nmae` provided to that function. Since it's nothing more than `foreign::read.dbf()`, it's probably not necessary to even have it as a function.

Value

A data frame.

rsatscan	<i>R functions, a class, and methods for working with SaTScan stand-alone software.</i>
----------	---

Description

rsatscan provides a suite of functions that allows you to easily write SaTScan parameter files in the OS, run SaTScan in the OS, and read the output files that SaTScan generates.

Details

The parameter files are constructed in R using the `ss.options` function and written to the OS using the `write.ss.prm` function. SaTScan is run using the `satscan` function. The `satscan` function returns a `satscan-class` object that has a slot for every possible file that SaTScan makes, plus one for the parameter file you used to generate the output.

The package also includes `write.???` functions which will write case, control, geography, population, etc., files in the format expected by SaTScan, if you happen to have them (or make them) in R and want to write them into the OS for SaTScan to use.

There are summary and print methods for `satscan-class` objects. There are also plot methods in the `sp` package, which can be used if the `rgdal` package and `sp` packages are installed and SaTScan generated a shapefile.

Currently the package works with SaTScan ≥ 9.2 and has been tested on Windows 7 and Ubuntu 14.04.1. Please contact the author if you find success or trouble on other OSes.

satscan	<i>Run SaTScan in the OS</i>
---------	------------------------------

Description

Calls out into the OS to run SaTScan, with the parameter file specified

Usage

```
satscan(prmlocation, prmfilename, sslocation = "c:/progra~2/satscan",
        ssbatchfilename = "SaTScanBatch", cleanup = TRUE, verbose = FALSE)
```

Arguments

prmlocation	A string containing the directory location where the parameter file is located.
prmfilename	A string containing the name of the parameter file, without the extension, i.e., no ".prm".
sslocation	A string containing the directory location where satscanbatch.exe (Windows) is located. The default value is a common location in Windows 7.

<code>ssbatchfilename</code>	Name of the file containing the SaTScan executable. This is likely to be either SaTScanBatch or SaTScanBatch64. Omit the file extension.
<code>cleanup</code>	If true, deletes any SaTScan output files from the OS.
<code>verbose</code>	If true, will display the results in the R console as if running SaTScan in batch. This may be especially useful if you expect SaTScan to take a long time to run.

Details

The parameter file may have been made by the `ss.options` function or not. If not, or if the `matchout = FALSE` parameter was set in `ss.options`, then the return object will include the main text output from SaTScan only you manually set the `ResultsFile` SaTScan parameter to have the same name as the parameter file.

Value

A `satscan`-class object, which is a list of 8 items, not all of which are always made, depending on SaTScan options and whether the program call was successful or not:

main A character vector containing the main text output from SaTScan. This is probably identical to the material displayed when `verbose=True`

col A data frame with the basic cluster information dataset SaTScan makes.

rr A data frame with the risk ratio dataset SaTScan makes.

gis A data frame with the geographic information dataset SaTScan makes.

llr A data frame with the log likelihood ratios dataset SaTScan makes.

sci A data frame with the other cluster information dataset SaTScan makes.

shapeclust A list object, of class `SpatialPolygonsDataFrame`, defined by the `sp` package. It contains the ESRI shapefile(s) SaTScan makes. This is made only if the `rgdal` package is available.

prm A character vector containing the contents of the parameter file you told SaTScan to use.

If an item is not made by SaTScan, it will be NA.

See Also

`ss.options`, `write.ss.prm`

Examples

```
## Not run:
## Please see vignette("rsatscan"); example() code doesn't make sense since
## all examples rely on calls to SaTScan in the OS.

## End(Not run)
```

ss.options

Set or reset parameters to be used by SaTScan

Description

Set or reset parameters to be used by SaTScan

Usage

```
ss.options(invals = NULL, reset = FALSE)
```

Arguments

invals	A list with entries of the form name=value, where value should be in quotes unless it is a number. Alternatively, may be a character vector whose entries are of the form "name=value". The "name" in either case should be a valid SaTScan parameter name; unrecognized names will generate a warning and will do nothing.
reset	If TRUE, will restore the default parameter values described in the "Details" section.

Details

ss.options() is intended to function like par() or options(). There is a default set of parameter settings that resembles the one used by SaTScan, except that it produces all possible output files and makes them as .dbf files instead of text.

Value

If invals == NULL, returns the current parameter set, as altered by previous calls to ss.options() since the last call with reset=TRUE. Otherwise returns modified parameter set invisibly. The side effect, if invals != NULL, is to set the current values of the parameters per the value of invals and reset.

Examples

```
## Not run:
head(ss.options(),3)
ss.options(list(CaseFile="NYCfever.cas"))
head(ss.options(),3)

# reset; shows whole parameter file without invisible()
invisible(ss.options(reset=TRUE))
head(ss.options(),3)

## End(Not run)
```

ss.options.extra	<i>Add lines to the current SaTScan parameter list</i>
------------------	--

Description

Allows you to add arbitrary lines to the current set of SaTScan parameters

Usage

```
ss.options.extra(invals = NULL)
```

Arguments

invals	A character vector, which will be added to the end of the current paramter list.
--------	--

Details

For certain SaTScan models or inputs (multiple data sets, Polygon), SaTScan allows a variable number of parameters; these parameters are not used/allowed for other models or inputs. This function allows the user to add arbitray lines to the current list of parameters. In addition to the options mentioned, it could also be used to add comments to the parameter file.

Value

Nothing.

subin	<i>Substitute new values into the input object</i>
-------	--

Description

Replaces existing values found in one object with new values

Usage

```
subin(x, ssparams)
```

Arguments

x	A character vector of the form "name=value"
ssparams	A character vector with arbitrary lines, currently imagined to be .ss.params

Details

For each line of `x`, the function: 1) finds the "name" and the "value" 2) checks to see whether the "name" exists in `ssparams`; if not, prints a warning but if so, replaces the existing line of `ssparams` with that line of `x`.

Not expected to be used directly.

Value

The modified `ssparams`.

summary.satscan	<i>Methods for satscan-class objects</i>
-----------------	--

Description

These functions define the default methods for `satscan`-class objects, which are the result objects from a call to `satscan()`

Usage

```
## S3 method for class 'satscan'
summary(object, ...)
```

Arguments

object	is a <code>satscan</code> object
...	vestigial, for compatability with the default summary method

Value

object, invisibly. Side effect is to display minimal facts contained in `ss`

write.adj	<i>Write a SaTScan adj file</i>
-----------	---------------------------------

Description

Write a SaTScan adj file

Usage

```
write.adj(x, location, filename, userownames = FALSE)
```

Arguments

x	Your data frame.
location	Directory location where the file should be written
filename	Name for the output file in the OS; .adj will be added.
userownames	If TRUE, will write the row names into the file.

Details

Writes the input data frame to the OS, using the .adj extension. Contents of the data frame should be only what you want SaTScan to see. This is a simple function that calls write.table, since SaTScan just needs ASCII files.

write.cas	<i>Write a SaTScan cas (case) file</i>
-----------	--

Description

Write a SaTScan cas (case) file

Usage

```
write.cas(x, location, filename, userownames = FALSE)
```

Arguments

x	Your data frame.
location	Directory location where the file should be written
filename	Name for the output file in the OS; .cas will be added.
userownames	If TRUE, will write the row names into the file.

Details

Writes the input data frame to the OS, using the .cas extension. Contents of the data frame should be only what you want SaTScan to see. This is a simple function that calls write.table, since SaTScan just needs ASCII files.

write.ctl	<i>Write a SaTScan ctl (control) file</i>
-----------	---

Description

Write a SaTScan ctl (control) file

Usage

```
write.ctl(x, location, filename, userownames = FALSE)
```

Arguments

x	Your data frame.
location	Directory location where the file should be written
filename	Name for the output file in the OS; .ctl will be added.
userownames	If TRUE, will write the row names into the file.

Details

Writes the input data frame to the OS, using the .geo extension. Contents of the data frame should be only what you want SaTScan to see. This is a simple function that calls write.table, since SaTScan just needs ASCII files.

write.geo	<i>Write a SaTScan geo file</i>
-----------	---------------------------------

Description

Write a SaTScan geo file

Usage

```
write.geo(x, location, filename, userownames = FALSE)
```

Arguments

x	Your data frame.
location	Directory location where the file should be written
filename	Name for the output file in the OS; .geo extension will be added.
userownames	If TRUE, will write the row names into the file.

Details

Writes the input data frame to a file in the OS, using the .geo extension. Contents of the data frame should be only what you want SaTScan to see. This is a simple function that calls write.table, since SaTScan just needs ASCII files.

write.grd	<i>Write a SaTScan grd (grid) file</i>
-----------	--

Description

Write a SaTScan grd (grid) file

Usage

```
write.grd(x, location, filename, userownames = FALSE)
```

Arguments

x	Your data frame.
location	Directory location where the file should be written
filename	Name for the output file in the OS; .grd will be added.
userownames	If TRUE, will write the row names into the file.

Details

Writes the input data frame to the OS, using the .grd extension. Contents of the data frame should be only what you want SaTScan to see. This is a simple function that calls write.table, since SaTScan just needs ASCII files.

write.ha	<i>Write a SaTScan ha (alternative hypothesis) file</i>
----------	---

Description

Write a SaTScan ha (alternatove hypothesis) file

Usage

```
write.ha(x, location, filename, userownames = FALSE)
```

Arguments

x	Your data frame.
location	Directory location where the file should be written
filename	Name for the output file in the OS; .ha will be added.
userownames	If TRUE, will write the row names into the file.

Details

Writes the input data frame to the OS, using the .ha extension. Contents of the data frame should be only what you want SaTScan to see. This is a simple function that calls write.table, since SaTScan just needs ASCII files.

write.max	<i>Write a SaTScan max file</i>
-----------	---------------------------------

Description

Write a SaTScan max file

Usage

```
write.max(x, location, filename, userownames = FALSE)
```

Arguments

x	Your data frame.
location	Directory location where the file should be written
filename	Name for the output file in the OS; .max will be added.
userownames	If TRUE, will write the row names into the file.

Details

Writes the input data frame to the OS, using the .max extension. Contents of the data frame should be only what you want SaTScan to see. This is a simple function that calls write.table, since SaTScan just needs ASCII files.

write.met	<i>Write a SaTScan met file</i>
-----------	---------------------------------

Description

Write a SaTScan met file

Usage

```
write.met(x, location, filename, userownames = FALSE)
```

Arguments

x	Your data frame.
location	Directory location where the file should be written
filename	Name for the output file in the OS; .met will be added.
userownames	If TRUE, will write the row names into the file.

Details

Writes the input data frame to the OS, using the .met extension. Contents of the data frame should be only what you want SaTScan to see. This is a simple function that calls write.table, since SaTScan just needs ASCII files.

write.nbr	<i>Write a SaTScan nbr (neighbor) file</i>
-----------	--

Description

Write a SaTScan nbr (neighbor) file

Usage

```
write.nbr(x, location, filename, userownames = FALSE)
```

Arguments

x	Your data frame.
location	Directory location where the file should be written
filename	Name for the output file in the OS; .nbr will be added.
userownames	If TRUE, will write the row names into the file.

Details

Writes the input data frame to the OS, using the .nbr extension. Contents of the data frame should be only what you want SaTScan to see. This is a simple function that calls write.table, since SaTScan just needs ASCII files.

write.pop	<i>Write a SaTScan pop (population) file</i>
-----------	--

Description

Write a SaTScan pop (population) file

Usage

```
write.pop(x, location, filename, userownames = FALSE)
```

Arguments

x	Your data frame.
location	Directory location where the file should be written
filename	Name for the output file in the OS; .pop will be added.
userownames	If TRUE, will write the row names into the file.

Details

Writes the input data frame to the OS, using the .pop extension. Contents of the data frame should be only what you want SaTScan to see. This is a simple function that calls write.table, since SaTScan just needs ASCII files.

write.ss.prm	<i>Write the SaTScan parameter file</i>
--------------	---

Description

Writes the current set of SaTScan parameters to a specified location in the OS.

Usage

```
write.ss.prm(location, filename, matchout = TRUE)
```

Arguments

location	A directory location, excluding the trailing "/".
filename	The name of the file to be written to the OS; The extension ".prm" will be appended.
matchout	If false, the ResultsFile parameter will not be touched; note that this will likely result in undesirable performance from calls to <code>satcan()</code> using the parameter file. If true, the ResultsFile is reset to share the filename given here.

Details

The current SaTScan options can be reset or modified `ss.options()` and/or `ss.options.extra()`. Once they are set as desired, they can be written to the OS using this function.

Value

Nothing. (Invisibly.) Side effect is to write a file in the OS.

See Also

[ss.options](#), [ss.options.extra](#)

Examples

```
## Not run:  
## Would write the current ss.options() to c:/temp/NYCfever.prm  
write.ss.prm("c:/tmp", "NYCfever")  
  
## End(Not run)
```

Index

charlistopts, [2](#)

NHumbersidecas, [3](#)

NHumbersidectl, [3](#)

NHumbersidegeo, [4](#)

NMcas, [4](#)

NMgeo, [5](#)

NMpop, [5](#)

NYCfevercas, [6](#)

NYCfevergeo, [6](#)

print.satscan, [7](#)

read.col, [7](#)

read.gis, [8](#)

read.llr, [8](#)

read.rr, [9](#)

read.satscanmain, [9](#)

read.sci, [10](#)

rsatscan, [11](#)

rsatscan-package (rsatscan), [11](#)

satscan, [11](#)

ss.options, [12](#), [13](#), [21](#)

ss.options.extra, [14](#), [21](#)

subin, [14](#)

summary.satscan, [15](#)

write.adj, [15](#)

write.cas, [16](#)

write.ctl, [17](#)

write.geo, [17](#)

write.grd, [18](#)

write.ha, [18](#)

write.max, [19](#)

write.met, [19](#)

write.nbr, [20](#)

write.pop, [20](#)

write.ss.prm, [12](#), [21](#)