

Package ‘rSARP’

May 25, 2016

Title Functions to Create and Evaluate Search and Rescue Plans

Version 1.0.0

Author John Hutcheson [aut, cre]

Maintainer John Hutcheson <jacknx8a@gmail.com>

Description Tools to create, evaluate, critique, revise, track progress, and communicate a detailed wilderness or urban search plan to management. This package uses and creates csv files in the R working directory to document inputs and results. It also creates a series of PDF and PNG files to accomplish communication of the plan. The program creates and revises search plans using Bayesian models. The package includes functions `bestsearch()`, `searchstatus()` and `searchme()` to model the number of searchers and hours required to search an area, calculate the probability of detection, probability of success, and project the best plan given limited resources.

Depends R (>= 3.2.4)

Imports ggplot2, lattice, qcc, grid, methods, grDevices, graphics, stats

License GPL (>= 3)

LazyData TRUE

BuildVignettes TRUE

Suggests R.rsp

VignetteBuilder R.rsp

RoxygenNote 5.0.1

NeedsCompilation no

Repository CRAN

Date/Publication 2016-05-25 06:09:17

R topics documented:

<code>bestsearch</code>	2
<code>rSARP</code>	3

SearchInput.csv	4
searchme	5
SearchOut.csv	6
searchstatus	8
Terrains.csv	10
tracking	10
Tracking.csv	11
Weathers.csv	12

Index	13
--------------	-----------

bestsearch	<i>bestsearch - Optimization of Search for Fixed Number of Searchers</i>
------------	--

Description

bestsearch() This function takes a fixed number of searchers and reports the area and POS that could be covered by them.

Usage

```
bestsearch(AvailablePeeps = 25, sdirectory = "Sensitivity", FTt = c(1, 2,
3, 4, 5), STitle = "Search Name", graphs = 2)
```

Arguments

AvailablePeeps - expected headcount of searchers for this operational period

sdirectory - the sub directory within the search directory where the results will be stored for this 'sensitivity' analysis. Defaults to the "Sensistivity" subdirectory. The function will create a new subdirectory if one doesn't already exist.

FTt - vector of Fast to Thorough parameters to search for this solution. FTt is a parametric version of the relative spacing between team members. Defaults to range of integers 1 thru 5. 1 represents a very Loose Grid search. 5 represents a tight grid approaching.

STitle - string representing the name of the search for use as a title on the graphic output

graphs - integer that controls the level of output from the searchme function - enter 1 for everything, 2 to get everything but the sector graphs, and 3 to skip all but the console output

Details

bestsearch() is an optimization function which is designed to examine the ways that a limited number of searchers can be used to search a given search field. The function repeatedly runs searchme() and captures the results graphically, displaying the area covered, the maximum POS available, the average AMDR, average team spacing used on the field, and an inflated AMDR value. The inflated AMDR value is the value the measured AMDR would need to climb to in order to return the highest

observed POS for the search plan. This becomes a measure of the 'risk' of missing the subject if a faster search method is used. The AMDR, which is a strictly visual measure of optimal team spacing, can be inflated if the subject can be assumed to be responsive or mobile. This function helps the management team identify the best plan to use for a limited number of searchers. It becomes the discussion vehicle for how fast to search in a search plan. The function creates a graph displaying the expected search area covered and POS returned for searches across a span of searcher spacings and sector headcounts.

Value

This function returns a pdf file containing two graphs in the R working directory. The first graph shows the area covered and POS achieved as a function of the FtT used to plan the search. The second graph shows the average spacing for the area covered, the average AMDR used for the plan, and the average inflated AMDR. The average inflated AMDR is the amount the AMDR would have to increase to increase the POS achieved to the highest observed value in the analysis.

Author(s)

John F. Hutcheson

Examples

```
## Not run: bestsearch(AvailablePeeps=84, STitle="HighTurnout Alternatives", graphs= 3)
```

rSARP

Create, modify, critique, and report against search and rescue plans using rSARP.

Description

rSARP allows you to create and manage search and rescue search grid plans using CSV files and the graphic output from R. The package includes 3 main functions. The `searchme()` function provides modeling by sector and is the core of the package. The `searchstatus()` function provides graphic reporting against the core plan and is typically used several times within a planning cycle as plans are modified. The `bestsearch()` function is used at the beginning of a planning cycle to determine the best search methods to use for a limited number of searchers. The `tracking()` function is used to track task assignments (525s) as they are assigned and completed.

Details

The first function you're likely to need from rSARP is `searchme`. Otherwise refer to the package documentation to see how to proceed.

SearchInput.csv

SearchInput.csv - typical Data input file for rSARP

Description

A dataset containing the sectors or search areas and key attributes of each that will be searched. This file is stored in the working directory of R and must be named SearchInput.csv. The variables described in the file are as follows:

Format

A data frame with 32 rows and 15 variables

Details

- **Sector.** Name of the sector or grid to be searched. Usually Alpha characters excluding I and O.
- **Area.** Area of the sector to be searched in Hectares or Acres. Defaults to Hectares and requires parametric change in call to searchme() if Acres is used.
- **AreaCoverage.** Percentage of the the sector Area that was or will be covered. Typically used if the search teams are unable to cover the entire sector due to unforeseen hazards in the sector. If the value given is negative, then it signifies that portion of the area is being searched for the first time, bypassing bayesian treatments, even if the sector has a POScum value (indicating the a portion of the sector was previously searched).
- **Terrain.** Keyword describing the terrain of the sector which can be used to provide the expected speed of travel and estimated AMDR. Currently limited to use of a set of keywords. Use HeavyWoods, OpenWoods, OpenArea, SteepTerrain, ModerateTerrain, FlatArea, Urban. See guide for further specifics.
- **TOD.** Time of Day. Keyword describing time of search and currently limited to Day or Night.
- **WX.** Weather keyword describing conditions during the search. Currently limited to Normal or Snow6to8in.
- **THours.** Target duration Hours. Hours team will be deployed for the search in the field not counting breaks or transportation time. If -1 entered for this value program will optimize using internal algorithms.
- **TSearchers.** Target Searchers. Count of searchers, excluding non-searching supervisors, in the field assigned to that sector. If -1 entered for this value program will optimize using internal algorithms.
- **TSpacing.** Target Spacing. The distance in meters between searchers planned for this sector search. If -1 entered for this value program will optimize using internal algorithms.
- **AMDR.** Average Maximum Detection Range in meters. Initially estimated, then updated as measured AMDRs are reported in. If -1 entered for this value program will estimate based on Terrain values
- **Rank.** Total rank of the sector from sector ladder ranking exercise. Use raw score totals - software will normalize and handle appropriately.

- POScum. Cumulative Probability of Success for the sector in percentage points. Represents the POS that has already been 'harvested' for that area. If entered as negative number, will revise reports to indicate that no further searching is planned for this sector.
- TerrainWords. Listing of the key words which can be used in Terrain values.
- TODWords. Listing of the key words which can be used in TOD values.
- WXWords. Listing of the key words which can be used in WX values.

Source

John Hutcheson

searchme	searchme - <i>Creates or modifies a Search Plan.</i>
----------	--

Description

The searchme() function takes the input csv file SearchInput.csv and returns the SearchOut.csv file which defines a search plan.

Usage

```
searchme(FtT = 3, filout = "SearchOut.csv", AreaUnits = "Hectares",
graphs = 1, directory = "Searches", STitle = "")
```

Arguments

FtT	- Fast to Thorough parameter - sets the relative hastiness of the search 1 = very fast, 5 = very thorough. This is a parametric form of the team spacing.
filout	- name of the output csv file to store in the working directory. Defaults to "SearchOut.csv". This output files contains the input and output variables for each sector that comprises the search plan.
AreaUnits	- a variable that toggles between Acres and Hectares area units. Defaults to "Hectares" for sectors. All area units are consistent with this selection within the program.
graphs	- variable that toggles the creation and storage of graphs for each sector which describe the selected strategy for searching each sector. This parameter has the biggest impact on run time. Defaults to graphs=1 (graphs will be produced and program will be at it's slowest). Set to graphs = 2 to turn off sector graphs. Set to 3 to turn off all output in file form but still report to the console.
directory	- name of the directory used to store output data from running this function. Defaults to the R working directory. The function will create a new subdirectory if the default value is changed. Used to segregate output from different runs of the model.
STitle	- string representing the name of the search for use as a title on the graphic output

Details

The `searchme()` function takes inputs including Sector, Area, AreaCoverage, Terrain, WX, TOD, THours, TSearchers, TSpacing, AMDR, Rank, and POScum in the form of a CSV file (SearchInput.csv) and returns values to the console and files to the working directory that form the basis of a search plan. At the console, a dataframe with new columns containing estimates for the man hours required, ideal team size, time to clear the sector, POS, POD, cumulative POS, and estimates to range these values between NASAR worst case estimates and Rule of Thumb estimates is returned. Then the function returns a summary of descriptive stats summarizing the plan followed by another display of the dataframe sorted alphabetically by name. The function stores the dataframe in the default SearchOut.csv file, along with a number of graphics displaying the model details by sector. After all sector stats and charts are generated, the function calls the `searchstatus()` function to create summary graphs for the search plan. See the documentation for `searchstatus()` for further information on those features. The `searchme()` function is designed to predict optimal values for all key input variables where it finds the value -1. In other cases, the program accepts the given value and assumes the planner knows best, but graphs these values to allow planners to easily compare them to an Urban and a NASAR model.

Value

df - data frame that includes the input table with all unknown variables defined

Author(s)

John F. Hutcheson

Examples

```
## Not run: searchme(FtT=1.5, graphs=2, AreaUnits = "Acres", STitle="Example Loose Grid Search")
## Not run: searchme(FtT=4, graphs=3, STitle = "Example Tight Grid Search")
```

SearchOut.csv

SearchOut.csv - typical Data input file for rSARP

Description

A dataset containing the sectors or search segments and key attributes of each that will be searched, including the outputs. The variables are as follows:

Format

A data frame with 32 rows and 38 variables

Details

- Sector. Name of the sector or grid to be searched. Usually Alpha characters excluding I and O.
- Area. Area of the sector to be searched in Hectares or Acres. Defaults to Hectares and requires parametric change in call to searchme() if Acres is used.
- AreaCoverage. Percentage of the the sector Area that was or will be covered. Typically used if the search teams are unable to execute due to unforeseen hazards in the sector. If the value being given is negative, then the area is being searched for the first time and bayesian treatments are skipped.
- Terrain. Keyword describing the terrain of the sector. Limited to use of a limited set of keywords. See guide for further specifics.
- TOD. Time of Day. Keyword describing time of search and currently limited to Day or Night.
- WX. Weather keyword describing conditions during the search. Currently limited to Normal or Snow6to8in.
- THours. Target duration Hours. Hours team will be deployed for the search in the field not counting breaks or transportation time. If -1 entered for this value program will optimize using internal algorithms.
- TSearchers. Target Searchers. Count of searchers, excluding non-searching supervisors, in the field assigned to that sector. If -1 entered for this value program will optimize using internal algorithms.
- TSpacing. Target Spacing. The distance in meters between searchers planned for this sector search. If -1 entered for this value program will optimize using internal algorithms.
- AMDR. Average Maximum Detection Range in meters. Initially estimated, then updated as measured AMDRs are reported in. If -1 entered for this value program will estimate based on Terrain values
- Rank. Total rank of the sector from sector ladder ranking exercise. Use raw scores - software will normalize and handle appropriately.
- POScum. Cumulative Probability of Success for the sector in percentage points. Represents the POS that has already been 'harvested' for that area. If entered as negative number, will revise reports to indicate that no further searching is planned for this sector.
- fnCovered. Fraction of the planned area covered by the search.
- fnMissed. Fraction of the planned area missed by the search.
- AreaCov. Area Covered. Area of the search seactor actually covered in Hectares or Acres.
- POC. Probability of Containment. The normalized sector ranking.
- terspd. Terrain Speed. The estimated ground speed for the team in miles per hour based on terrain or given by planner constraints.
- terheads. Terrain Manhours. The estimated man hours to search the sector given spacing and speed constraints.
- FirstPass. Parameter indicating if the area to be cleared has already been searched or will be searched for the first time.
- ROThrs. Rule of Thumb hours. Estimated search man hours based on Model used in Kent County, primarily for urban search efforts.

- NHRSH. NASAR Man hours. Estimated man hours using NASAR rule of thumb.
- SAVHRS. Duration search hours saved if one additional person added to team to 'lengthen the line'.
- ESW. Effective spacing width. Currently 1.5 times the AMDR but technically the length of Coverage in meters. Used to determine COV.
- COV. Coverage as a fraction. The TSpacing divided by ESW. Used to determine POD.
- POD. Probability of Detection as a fraction. Likelihood of detecting the object in the sector if the object is present.
- delspc. The change in spacing between team members if an additional person were added to the line but the overall length kept the same.
- dPOD. The change in the Probability of Detection that would result if the team spacing were changed by delspc.
- POS. Probability of Success. The probability that the subject would be found if present in that sector as a percent.
- dPOS. The change in POS in percent if the spacing between team members is changed by delspc.
- OPOScum. Overall cumulative Probability of Success. The running tally on POS by sector including bayesian corrections.
- RTotlHeads. Running total heads. The number of searchers required to as a running total by sector to execute the plan.
- RTotlPOS. Running total POS. The percentage of POS accumulated if the plan is executed by sector as a running total.
- TotlArea. The running total area covered by this search plan in Hectares or Acres.
- AvgSpacing. The area averaged spacing between team members for all sectors this row and above.
- rPOC. The running Probability of Containment for all sectors this row and above.
- AvgAMDR. The running area averaged AMDR for all sectors this row and above.
- POSm. THE probability of success that was missed due to the area missed.

Source

John Hutcheson

searchstatus

searchstatus() - *provides summary graphs of search.*

Description

searchstatus() - this function called by searchme to produce summary graphs of search for management.

Usage

```
searchstatus(filout = "SearchOut.csv", directory = "Searches",
             STitle = "")
```

Arguments

filout	- string variable. Name of the output csv file generated by the searchme function. This file usually is named SearchOut.csv but can be changed from the default by the user. If the user renames the output file from searchme, the same name must be used here as this file is located in the working directory and is used as the source of input values.
directory	- string variable. Name of the directory created and used by the searchme function to store output charts and files under the R working directory. This directory is expected to exist at the time the function is run.
STitle	- string representing the name of the search for use as a title on the graphic output

Details

searchstatus() summarizes a search in a series of graphs which display search plans. The graphs display all sectors in order of sector ladder importance or in order of POS (probability of success) expected along with other metrics critical to search management. The leading chart is a barchart which shows the available POS, the POS already "harvested", and the POS expected in this planning cycle. Pareto charts then summarize the remaining metrics which include POS expected, POS remaining, Searchers required, duration hours planned, sectors with the highest potential to go faster with more manpower, sectors with the best potential to return POS with manpower added, man hours required, and manhours per POS. An additional page summarizes the sectors that have been set aside as "complete" and where no further search effort is planned. This function stores the charts in the R working directory or a subdirectory if one is called out through the **directory** parameter

Value

searchstatus returns a pdf file that contains 2 or 3 pages of graphs summarizing a search.

Author(s)

John F. Hutcheson

Examples

```
## Not run: searchstatus(STitle="Alternate Search Plan B")
```

 Terrains.csv

Terrains.csv - typical Data input file for rSARP

Description

A dataset containing the terrains and key attributes of each that are relevant to search. The variables are as follows:

Format

A data frame with 7 rows and 4 variables

Details

- Terrain. Name of the Terrain described. Usually string descriptions that are easy to remember.
- Hours. Relative man hours of search required for the terrain.
- AMDR. Average maximum detection range that is typical for the Terrain.
- Speed. Expected travel speed through the terrain in miles/hr.

Source

John Hutcheson

 tracking

tracking - Tracking 525s completion within a Search

Description

`tracking()` This function lives to track 525 progress towards completion for a search. Running this function after the `tracking.csv` file has been updated produces a graph that displays the progress of search by 525.

Usage

```
tracking(title = "", directory = "Searches")
```

Arguments

- | | |
|------------------------|---|
| <code>title</code> | - string that represents the Name of the Search. Used to label the graph with the search name. |
| <code>directory</code> | - a string representing the sub directory within the search directory where the results will be stored for this analysis. Defaults to the R working directory. The function will create a new directory if the default is not accepted. |

Details

tracking() is a tracking function built to track 525 progress graphically for reports to the SM / PIO. The height of each bar defines the number of searchers working on this task. The location of the triangle relative to the top of the bar indicates the

Value

This function returns a graphic showing a bar chart analysis of 525s within a search by area covered and POS achieved

Author(s)

John F. Hutcheson

Examples

```
tracking(title="Midland County Search B")
```

Tracking.csv

Tracking.csv - typical Data input file for tracking()

Description

A dataset containing the names or assignment numbers of the 525 tasks assigned within a search, the type of task, the number of people assigned, and the completion of the task. This file is stored in the working directory of R and must be named Tracking.csv. The variables described in the file are as follows:

Format

A data frame with 33 rows and 5 variables

Details

- Task. Name or assignment number of the task. Usually Alpha characters excluding I and O if search sector names are used.
- PerCentComplete. Relative task completion amount expressed as a percentage. An entry of 25 indicates the task is one quarter complete.
- TSearchers. Number of persons assigned to this task.
- Type. String representing the category of task this assignment belongs to. Generally use Initial, Loose, and Tight types, but the user is free to create sub categories as needed.
- Comment. Time of Day. Keyword describing time of search and currently limited to Day or Night.

Source

John Hutcheson

Weathers.csv

Weathers.csv - typical Data input file for **rSARP**

Description

A dataset containing the Weather conditions that affect search and the speed impacts. The variables are as follows:

Format

A data frame with 6 rows and 2 variables

Details

- Weather. Name of the Weather described. Usually string descriptions that are easy to remember and descriptive of conditions.
- Speed. Expected travel speed through the Weather in miles/hr.

Source

John Hutcheson

Index

bestsearch, [2](#)

rSARP, [3](#)

rSARP-package (rSARP), [3](#)

SearchInput.csv, [4](#)

searchme, [3](#), [5](#)

SearchOut.csv, [6](#)

searchstatus, [8](#)

Terrains.csv, [10](#)

tracking, [10](#)

Tracking.csv, [11](#)

Weathers.csv, [12](#)