

# Package ‘snht’

October 17, 2017

**Type** Package

**Title** Standard Normal Homogeneity Test

**Version** 1.0.5

**Date** 2017-10-17

**Imports** mgcv, ggplot2, gridExtra, zoo, plyr, reshape2, methods

**Author** Josh Browning [aut, cre],  
Carina Schneider [aut]

**Maintainer** Josh Browning <rockclimber112358@gmail.com>

**Description** Implementation of robust and non-robust Standard Normal Homogeneity Test (SNHT) for changepoint detection. This test statistic is equal sided, as proposed in "Homogenization of Radiosonde Temperature Time Series Using Innovation Statistics" by Haimberger, L., 2007. However, the statistic contains an estimate of  $\sigma^2$  in the denominator instead of  $\sigma$ , which seems to be a more appropriate value (based on the paper "Homogenization of Swedish temperature data. Part I: Homogeneity test for linear trends." by Alexandersson, H., and A. Moberg, 1997).

**License** GPL-3

**VignetteBuilder** knitr

**Suggests** mvtnorm, knitr

**RoxygenNote** 5.0.1

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2017-10-17 20:10:24 UTC

## R topics documented:

snht-package	2
createCandidateMatrix	2
getPairs	3
getUniquePairs	3
pairwiseSNHT	4

plotSNHT . . . . .	5
removeSeasonalPeriod . . . . .	6
robustSNHT . . . . .	6
robustSNHTunequal . . . . .	8
snht . . . . .	9
unconfoundCandidateMatrix . . . . .	11

## Index 13

---

snht-package	<i>Robust and Non-Robust Standard Normal Homogeneity Test</i>
--------------	---

---

### Description

Computes test statistics for the SNHT and robust SNHT test. Additionally, users may supply a custom function for estimating the mean and standard deviation, and this function will be used for computing the test statistic.

### Details

The only function you're likely to need from **snht** is [snht](#). Otherwise refer to the vignettes.

---

createCandidateMatrix	<i>Create candidate matrix</i>
-----------------------	--------------------------------

---

### Description

This function creates `candidate`, a matrix where the  $(i,j)$ th entry corresponds to the number of changepoints in difference series for location  $j$  that occurred at time  $i$ . For example, suppose location  $i_1$  was paired with  $i_2$ ,  $i_3$ , and  $i_4$ . If the statistic for  $i_1-i_2$  and  $i_1-i_4$  exceeded the threshold at time  $j$ , then `candidatei,j = 2`.

### Usage

```
createCandidateMatrix(data, statistics, pairs, crit)
```

### Arguments

<code>data</code>	The data.frame containing the observations, restructured as in <code>pairwiseSNHT</code> . So, the first column should be time, and the other columns should be named with the locations and contain the observed values at each location.
<code>statistics</code>	The time $\times$ (number of pairs) matrix of SNHT statistics computed for each difference series.
<code>pairs</code>	The list object whose $i$ th element specifies the neighboring locations to the $i$ th location.
<code>crit</code>	The critical value such that if the <code>snht</code> statistic is larger than <code>crit</code> , a changepoint is assumed to have occurred. Defaults to 100, as recommended in Haimberger (see references).

**Value**

A matrix of dimension time x (number of locations). The (i,j) element of this matrix indicates the number of changepoints found in difference series containing the jth location at time i.

---

getPairs	<i>Gets Pairs from Distances</i>
----------	----------------------------------

---

**Description**

For each location, we wish to determine the k closest locations. This function takes the distance matrix and computes the returns a list of the k closest locations to each individual location.

**Usage**

```
getPairs(dist, k)
```

**Arguments**

dist	The distance matrix describing the distance between locations.
k	The number of closest neighbors to be located for each location.

**Value**

A named list. Each element of the list corresponds to a particular location, and the value at element i is the k closest locations to location i.

---

getUniquePairs	<i>Get Unique Pairs</i>
----------------	-------------------------

---

**Description**

For the pairwise SNHT, many difference time-series must be computed. However, if location i is used for location j, then it's very likely that location j will be used for location i. Thus, the pairs object will likely have many duplicate pairs. To save computation time, this function finds which pairs are unique.

**Usage**

```
getUniquePairs(pairs)
```

**Arguments**

pairs	The pairs list object, as returned by ?getPairs.
-------	--

**Value**

data.frame with columns loc1 and loc2. This data.frame will have no duplicates, and describes all the pairs that need to be computed.

---

pairwiseSNHT                      *Pairwise Standard Normal Homogeneity Test*

---

### Description

This function performs a pairwise standard normal homogeneity test on the data supplied, as described in Menne & Williams (2009).

### Usage

```
pairwiseSNHT(data, dist, k, period, crit=100, returnStat=FALSE, ...)
```

### Arguments

data	The data to be analyzed for changepoints. It must be a data.frame and contain either two or three columns. The mandatory columns are data and location, named as such. The option column is time, and this argument will be passed to snht.
dist	A distance matrix which provides the distance between location i and location j. Rows and columns must be named with the locations in data. Note that non-symmetric distances may be used. In that case, neighbors for station i will be determined by the smallest values in the row of dist corresponding to i.
k	How many of the nearest neighbors should be used to construct pairwise difference time series? Note that more than k neighbors may be used if there are ties in the distances between locations.
period	The SNHT works by calculating the mean of the data on the previous period observations and the following period observations. Thus, this argument controls the window size for the test statistics.
crit	The critical value such that if the snht statistic is larger than crit, a changepoint is assumed to have occurred. Defaults to 100, as recommended in Haimberger (see references).
returnStat	See return value. If TRUE, the snht statistics for each time point and for each difference pair are returned.
...	Additional arguments to pass to the snht function (such as robust, time, or estimator).

### Details

The pairwise snht works with a set of time series. For each time series, it's closest k neighbors are determined, and a time series of the difference between each of those time series is created. The snht is then applied to each of these difference time series. Changepoints in one time series can be detected by searching for large values of the test statistic across all difference time series for a particular location.

The usefulness of the pairwise snht is that it removes any patterns in the data that could affect the basic snht. For example, seasonal and linear trends that exist globally will be removed from the difference series, and thus changepoints are more easily detected.

**Value**

If returnStat is TRUE, the snht statistics for each time point and for each difference pair are returned. Otherwise, a named list is returned. The first element, data, contains the homogenized data in the same format as the supplied data. The second element, breaks, contains a data.frame where the first column is the location where a break occurred, the second column is the time of the break, and the third column is the amount that data after the break was shifted by.

**Author(s)**

Josh Browning (jbrownin@mines.edu) keyword ~snht ~homogeneity ~pairwise

**References**

L. Haimberger. Homogenization of radiosonde temperature time series using innovation statistics. *Journal of Climate*, 20(7): 1377-1403, 2007.

Menne, M. J., & Williams Jr, C. N. (2009). Homogenization of temperature series via pairwise comparisons. *Journal of Climate*, 22(7), 1700-1717.

**Examples**

```
dist = matrix(1, nrow=2, ncol=2)
colnames(dist) = c(1, 2)
rownames(dist) = c(1, 2)
dist = dist - diag(2)
pairwiseSNHT(data=data.frame(data=c(rep(1, 10), rep(2, 10), rep(3, 10), rep(4, 10)),
                               location=rep(1:2, each=10)), dist=dist, k=1, period=2)
```

---

plotSNHT

*Plot SNHT*

---

**Description**

Function to plot the result of the SNHT fit

**Usage**

```
plotSNHT(data, stat, time = NULL, alpha = NULL)
```

**Arguments**

data	The vector of time series observations that were input to the snht function.
stat	A data.frame as returned by the snht function.
time	If the observations in data are not equally spaced, then this vector will specify the times of the observations. This object should be numeric or should be able to be coerced to numeric.
alpha	The confidence level for the SNHT test. Note, though, that multiple tests are being performed and that is NOT accounted for in this function.

**Value**

No object is returned, but a plot is instead generated.

---

removeSeasonalPeriod    *Remove Seasonal Period*

---

**Description**

This function estimates the seasonal period of a time series via a GAM model. The data series with the seasonal period removed is then returned.

**Usage**

```
removeSeasonalPeriod(x, period, time = 1:length(x))
```

**Arguments**

x	The time series to be analyzed.
period	The period of the seasonality of the data.
time	If not provided, then the observations are assumed to occur at integer times 1, 2, ..., length(x). Otherwise, the time vector may specify when these observations occur.

**Value**

Returns a vector of data with the seasonality component removed.

**Author(s)**

Josh Browning (jbrownin@mines.edu)

---

robustSNHT                      *Robust SNHT*

---

**Description**

This function performs a standard normal homogeneity test using a robust estimator of the mean and standard deviation. It also allows for a user- defined definition of these statistics.

**Usage**

```
robustSNHT(data, period, scaled=TRUE, rmSeasonalPeriod=Inf
, estimator=function(x, minObs=5){
  x = x[!is.na(x)]
  if(length(x)<minObs) #Too many NA values, don't return a result
    return(c(NA,NA))
  if(max(table(x))>length(x)/2) #Too many duplicate values, MAD will be 0
    return(c(NA,NA))
  fit = MASS::huber(x)
  return(c(fit[[1]], fit[[2]]))
})
```

**Arguments**

data	The data to be analyzed for changepoints.
period	The SNHT works by calculating the mean of the data on the previous period observations and the following period observations. Thus, this argument controls the window size for the test statistics.
scaled	See ?snht.
rmSeasonalPeriod	See ?snht.
estimator	A custom function may be supplied to this function which computes estimates for the mean and standard deviation. The function should only take one argument (a numeric vector of data) and should return a vector of length two: the estimated center and spread. The huber function from MASS is implemented for the robust SNHT by default (along with some data quality checks).

**Details**

The SNHT works by calculating the mean of the data on the previous period and on the following period. The test statistic at each observation is then computed as described in Haimberger (2007). Essentially, though, it just compares the means of these two periods and normalizes by the standard deviation.

Note: if there are not enough observations both before and after the current observation, no test is performed.

Large values of the test statistic suggests the presence of a changepoint. Haimberger (see references) suggests values larger than 100 should be considered changepoints. However, this does not apply if scaled = TRUE.

**Value**

Returns a data.frame, with columns score, leftMean, and rightMean, and time. Statistic is the SNHT test statistic described above, and leftMean (rightMean) are the means to the left (right) of the current observation.

Note that new (missing) observations were introduced to the dataset to ensure the same number of observations occur per day.

**Author(s)**

Josh Browning (jbrownin@mines.edu)

**References**

L. Haimberger. Homogenization of radiosonde temperature time series using innovation statistics. *Journal of Climate*, 20(7): 1377-1403, 2007.

**See Also**

[huber](#)

Other snht functions: [robustSNHTunequal](#), [snht](#)

---

robustSNHTunequal	<i>Robust SNHT with Unequal Times</i>
-------------------	---------------------------------------

---

**Description**

This function performs a standard normal homogeneity test, but allows for unequally spaced observations in time.

**Usage**

```
robustSNHTunequal(data, period, time, estimator = NULL, scaled=TRUE
,rmSeasonalPeriod = Inf)
```

**Arguments**

data	The data to be analyzed for changepoints.
period	The SNHT works by calculating the mean of the data on the previous period observations and the following period observations. Thus, this argument controls the window size for the test statistics.
time	Numeric vector specifying times for the observations. If not supplied, it is assumed that each observation occurs on one time period. If supplied, then the algorithm will create a new dataset with the same number of observations for each time unit by adding missing values.
estimator	See ?robustSNHT
scaled	See ?snht.
rmSeasonalPeriod	See ?snht.



## Details

The SNHT works by calculating the mean of the data on the previous period and on the following period. The test statistic at each observation is then computed as described in Haimberger (2007). Essentially, though, it just compares the means of these two periods and normalizes by the standard deviation.

Note: if there are not enough observations both before and after the current observation, no test is performed.

Large values of the test statistic suggests the presence of a changepoint. Haimberger (see references) suggests values larger than 100 should be considered changepoints. However, this does not apply if `scaled = TRUE`.

## Value

Returns a data.frame, with columns `score`, `leftMean`, and `rightMean`, and `time`. `Statistic` is the SNHT test statistic described above, and `leftMean` (`rightMean`) are the means to the left (right) of the current observation.

Note that new (missing) observations were introduced to the dataset to ensure the same number of observations occur per day.

## Author(s)

Josh Browning ([jbrownin@mines.edu](mailto:jbrownin@mines.edu))

## References

L. Haimberger. Homogenization of radiosonde temperature time series using innovation statistics. *Journal of Climate*, 20(7): 1377-1403, 2007.

## See Also

[huber](#)

Other snht functions: [robustSNHT](#), [snht](#)

---

snht

*Standard Normal Homogeneity Test*

---

## Description

This function performs a standard normal homogeneity test on the data supplied. This test searches the data for potential changepoints.

## Usage

```
snht(data, period, robust = F, time = NULL, scaled = TRUE,  
      rmSeasonalPeriod = Inf, ...)
```

## Arguments

<code>data</code>	The data to be analyzed for changepoints.
<code>period</code>	The SNHT works by calculating the mean of the data on the previous period observations and the following period observations. Thus, this argument controls the window size for the test statistics.
<code>robust</code>	Flag indicating whether or not robust estimators should be used. If T, then Huber's robust estimator for the mean and variance will be used (see <code>?MASS::huber</code> ).
<code>time</code>	Numeric vector specifying times for the observations. If not supplied, it is assumed that each observation occurs on one time period. If supplied, then the algorithm will create a new dataset with the same number of observations for each time unit by adding missing values.
<code>scaled</code>	In the Haimberger paper, a typo is reported in the test statistic. The denominator ought to be $s^2$ instead of $s$ , as the distribution of the test statistic then becomes chi-squared assuming errors are normal. If <code>scaled = TRUE</code> , the corrected version is used. In this case, the test statistics can be compared to a chi-squared. However, if <code>scaled = FALSE</code> , there is no clear distribution to compare with.
<code>rmSeasonalPeriod</code>	This algorithm will overestimate the standard error (and hence incorrectly estimate the test statistic) if there is strong seasonality in the data. By setting <code>rmSeasonalPeriod</code> to some value, a GAM model will be built to capture that seasonality. Once it is estimated, it is removed and the SNHT statistic is computed on the resulting dataset. Setting this argument to <code>Inf</code> prevents any modeling.
<code>...</code>	Other parameters, see <code>?robustSNHT</code> , <code>?robustSNHTunequal</code> .

## Details

The SNHT works by calculating the mean of the data on the previous period and on the following period. The test statistic at each observation is then computed as described in Haimberger (2007). Essentially, though, it just compares the means of these two periods and normalizes by the standard deviation.

Note: if there are not enough observations both before and after the current observation, no test is performed.

Large values of the test statistic suggests the presence of a changepoint. Haimberger (see references) suggests values larger than 100 should be considered changepoints. However, this does not apply if `scaled = TRUE`.

Observations which are less than `period` away from the start or end of the dataset do not have valid SNHT statistics. Thus, the statistic for these observations is returned as NA.

## Value

Returns a `data.frame`, with columns `score`, `leftMean`, and `rightMean`. `Statistic` is the SNHT test statistic described above, and `leftMean` (`rightMean`) are the means to the left (right) of the current observation.

Additionally, if `time` is supplied, then `time` is returned on the output `data.frame`. Note that new (missing) observations were introduced to the dataset to ensure the same number of observations occur per day.

**Author(s)**

Josh Browning (jbrownin@mines.edu), Carina Schneider (carina.schneider@uzh.ch)

**References**

L. Haimberger. Homogenization of radiosonde temperature time series using innovation statistics. *Journal of Climate*, 20(7): 1377-1403, 2007.

**See Also**

[huber](#)

Other snht functions: [robustSNHTunequal](#), [robustSNHT](#)

**Examples**

```
data = rnorm(1000)
brk = sample(1000, size=1)
data[1:brk] = data[1:brk]-2
out = snht( data, period=50, robust=FALSE )
summary(out)

data = rnorm(1000)
time = 1:1000 + rnorm(1000)
brk = sample(1000, size=1)
data[1:brk] = data[1:brk]-2
out = snht( data, period=50, time=time, robust=FALSE )
summary(out)
```

---

unconfoundCandidateMatrix

*Unconfound candidate matrix*

---

**Description**

This function "unconfounds" the candidate matrix. At each time point and for each location, we have the number of difference series which resulted in a changepoint. The location with the largest count is assumed to be the location where the changepoint occurs. Assignment of changepoints should then proceed iteratively, where each new changepoint is assigned based on the current highest count.

**Usage**

```
unconfoundCandidateMatrix(candidate, pairs, statistics, data, period, avgDiff)
```

**Arguments**

<code>candidate</code>	The candidate matrix, as computed by <code>?createCandidateMatrix</code> .
<code>pairs</code>	The list object whose <i>i</i> th element specifies the neighboring locations to the <i>i</i> th location.
<code>statistics</code>	The time $\times$ (number of pairs) matrix of SNHT statistics computed for each difference series.
<code>data</code>	The <code>data.frame</code> containing the observations, restructured as in <code>pairwiseSNHT</code> . So, the first column should be time, and the other columns should be named with the locations and contain the observed values at each location.
<code>period</code>	The SNHT works by calculating the mean of the data on the previous period observations and the following period observations. Thus, this argument controls the window size for the test statistics.
<code>avgDiff</code>	A matrix containing the average differences between time series pairs. Generally this is created within <code>pairwiseSNHT()</code> .

**Value**

A list of two elements. The first element contains the data after the breaks have been removed. The second element is a `data.frame` with information regarding the detected changepoints (or `NULL` if none are found).

# Index

\*Topic **\textasciitildehomogeneity**

robustSNHT, 6  
robustSNHTunequal, 8  
snht, 9

\*Topic **\textasciitilderobust**

robustSNHT, 6  
robustSNHTunequal, 8

\*Topic **\textasciitildesnht**

robustSNHT, 6  
robustSNHTunequal, 8  
snht, 9

createCandidateMatrix, 2

getPairs, 3

getUniquePairs, 3

huber, 8, 9, 11

pairwiseSNHT, 4

plotSNHT, 5

removeSeasonalPeriod, 6

robustSNHT, 6, 9, 11

robustSNHTunequal, 8, 8, 11

snht, 2, 8, 9, 9

snht-package, 2

unconfoundCandidateMatrix, 11