The LWP-Trends library

July 2018

Title LWP-Trends library Version 1804: ***LWPTrends_v1808.r***

Date 18 July 2018

Authors Ton Snelder and Caroline Fraser LWP Ltd, Christchurch NEW ZEALAND.

Contacts: ton@lwp.nz or caroline@lwp.nz

Description Tools for analysing water quality trends

Requires: packages: plyr and gam are installed.

Contents:

Reference:

Snelder,T. and Fraser, C. (2018) *"The LWP-Trends Library; July 2018",* LWP Ltd Report, p20

Disclaimer

Software downloaded from the landwaterpeople (LWP) website (or provided by direct communication with an LWP employee) is provided 'as is' without warranty of any kind, either express or implied, including, but not limited to, the implied warranties of fitness for a purpose, or the warranty of non-infringement. Without limiting the foregoing, LWP makes no warranty that:

1. the software will meet your requirements
2. the software will be uninterrupted, timely, secure or error-free
3. the results that may be obtained from the use of the software will be effective, accurate or reliable
4. the quality of the software will meet your expectations
5. any errors in the software obtained from the LWP web site will be corrected.

Software and its documentation made available on the LWP website:

1. could include technical or other mistakes, inaccuracies, or typographical errors. LWP may make changes to the software or documentation made available on its website.
2. may be out of date, and LWP makes no commitment to update such materials.

LWP assumes no responsibility for errors or omissions in the software or documentation available from its website.

In no event shall LWP be liable to you or any third parties for any special, punitive, incidental, indirect or consequential damages of any kind, or any damages whatsoever, including, without limitation, those resulting from loss of use, data or profits, whether or not LWP has been advised of the possibility of such damages, and on any theory of liability, arising out of or in connection with the use of this software.

The use of the software downloaded through the LWP site is done at your own discretion and risk and with agreement that you will be solely responsible for any damage to your computer system or loss of data that results from such activities. No advice or information, whether oral or written, obtained by you from LWP or from the LWP website shall create any warranty for the software.

1 Introduction

This document describes how to use the functions in the LWP-Trends library to undertake water quality trend analysis in the R statistical computing environment. The functions were developed from scratch to provide up to date methods to evaluate trends.

Water quality trends are evaluated by fitting a regression to the relationship between the water quality variable (e.g., chemical concentration) and time, using the non-parametric Sen slope estimator[1]. The Sen slope is an estimate of the rate of change in the central tendency of the water quality variable through the period. This method is robust to three common characteristics of water quality data: non-normal distributions, censored values and missing data.

Evaluations of water quality trends at individual sites are always uncertain. The level of uncertainty depends on the number of samples and the magnitude of the water quality change through the period of analysis. Traditionally, a statistical significance test was undertaken that evaluates the uncertainty of the trend by considering if it could have been observed if the true trend were exactly zero[1]. An insignificant test indicates that the observed trend could have been observed by chance (at a defined level of alpha; generally 0.05) if the true trend was zero.

Recently, the logic underlying this significance test has been questioned and a new procedure has been adopted that evaluates the 100 – α confidence interval for the estimated trend slope[2][3]. Briefly, if a symmetric confidence interval around the trend slope does not contain zero, then the trend direction (either positive of negative) is “established with confidence”. If it does contain zero, it is concluded that the trend is “uncertain”[4]. It is this new procedure that is implemented in the LWP-Trends library. However, for consistency the functions also return the results of the traditional significance tests.

Trends are most robust when there are few censored values in the time-period of analysis. It has been common to substitute the censored values with 0.5×detection limit and 1.1×reporting limit. Although common, replacement of censored values with constant multiples of the detection and reporting limits can result in misleading results when statistical tests are subsequently applied to those data[5]. Originally, the new procedure was implemented by substituting censored values with values that were imputed from the data. The functions in LWP-Trends library use different methods for dealing with censored values that are based on robust handling of censored values in trend analysis[5]. These same methods have recently been implemented in the TimeTrends software[6] (Jowett, 2017), which is commonly used by regional councils in New Zealand.

2 Handling Censored Values

Censored values are those above or below a detection limit (e.g., >2.5 or <0.001). Values above the detection limit are described as right censored and values below the detection level are described as left censored. Trends are most robust when there are few censored values in the time-period of analysis. It has been common to substitute the censored values with

[1] Hirsch, R.M., J.R. Slack, and R.A. Smith, 1982. Techniques of Trend Analysis for Monthly Water Quality Data. Water Resources Research 18:107–121.
[2] Larned, S., T. Snelder, M. Unwin, and G. McBride, 2016. Water Quality in New Zealand Rivers: Current State and Trends. New Zealand Journal of Marine and Freshwater Research 50:389–417.
[3] McBride, G.B., Submitted 2018. Has Water Quality Improved or Been Maintained? A Quantitative Assessment Procedure.
[4] It is noted that a 100(1–2α)% two-sided (symmetrical) CI is used in the procedure to define the 100 – α level of confidence (see Larned et al., 2016 for details).
[5] Helsel, D.R., 2012. Statistics for Censored Environmental Data Using Minitab and R. John Wiley & Sons, Inc., Hoboken, New Jersey.
[6] Jowett, I.G., 2017. Time Trends. Hamilton, New Zealand.

0.5×detection limit and 1.1×reporting limit. Although common, replacement of censored values with constant multiples of the detection and reporting limits can result in misleading results when statistical tests are subsequently applied to those data[1].

These functions described below treat censored values in the manner described by Nondetects and Data Analysis for Environmental Data[7] and Statistics for censored environmental data using MINITAB and R[8]. Key calculations that are affected by censored values are the calculation of Kendall's S and its variance, the estimation of the Sen slope (including the seasonal Sen slope) and the estimation of the confidence intervals for Sen slopes. To calculate S and varS, and the confidence intervals for Sen slopes, the functions below use code obtained from the cenken() function in the R package NADA, which implements the analyses discussed in the above two books. The code is contained within the function GetKendal() in these functions. Briefly, for left-censored data, increases and decreases in a water quality variable are measured whenever possible. Thus, a change from <1 to 10 is an increase. A change from a <1 to a detected 0.5 is considered a tie, as is a <1 to a <5, because neither can definitively be called an increase or decrease. Similar logic applies to right censored values. The variance of the S statistic is adjusted for ties (it is reduced) and this influences the computation of confidence intervals.

The slope between any combination of observations in which either one or both are censored cannot be definitively calculated. The slopes associated with censored values are therefore ignored (i.e., removed) and SSE and SSSE are calculated as the median of all real valued slopes between sample dates. The removal of slopes associated with censored values has the effect of decreasing the number of samples used to determine the SSE and SSSE, therefore reducing statistical power and increasing the width of the confidence interval. This means that when there are many censored values, the analysis produces a low degree of confidence in the evaluated trend direction. Where there are fewer than five total and three unique, non-censored observations (but when the other filtering criteria are otherwise met), the method will not analyse the data and these cases are reported as "not analysed".

It is noted that the functions have the option to scan the data to find the highest censoring limit (for left censored data), set any recorded values that are less than this value to the highest censoring limit and set these as censored (at the highest censoring limit). This may be appropriate if changing analytical methods over time have changed the reporting limit and thereby risk inducing a trend in the data. This is achieved by setting the argument HiCensor = TRUE in the Kendall test and Sen slope estimator functions described below.

3 Data and preliminary set up

These functions are designed to undertake trend analyses on data pertaining to a single site + variable.

The functions can be used to analyse data that pertains to many sites + variable combinations by applying the functions to appropriately sub-setted data using (for example the ddply function from the plyr package).

It is expected that the data is in an R data frame format such that each water quality observation is a row. Each row must have columns that define the value, the date and the value of any covariate (e.g., flow). If the data pertains to many sites + variable combinations, a column must specify the variable name. An example of this type of data is shown in Figure 1.

[7] Helsel, D. R., 2005. Nondectects and Data Analysis; Statistics for censored environmental data. John Wiley and Sons, USA, NJ.
[8] Helsel, D. R., 2012. Statistics for censored environmental data using MINITAB and R, John Wiley & Sons.

```
                    sID       Date  npID      Value
1 Whareama River at Gauge 24/09/2003 NO3-N      0.273
2 Whareama River at Gauge 10/10/2003 NO3-N      0.211
3 Whareama River at Gauge 10/11/2003 NO3-N <0.0100000
4 Whareama River at Gauge  8/12/2003 NO3-N      0.012
5 Whareama River at Gauge 16/01/2004 NO3-N <0.0100000
6 Whareama River at Gauge 19/02/2004 NO3-N      0.414
```

Figure 1. Example of minimum water quality data for trend analysis. In this example, sID is the site name/identifier and npID is the water quality variable name.

3.1 Date Format

The first step is to add a column called myDate that represents the date of each observation as a vector of class "Date". This is achieved for the above data (which are a data frame called WQData) with following command:

```
WQData$myDate <- as.Date(as.character(WQData$Date),"%d/%m/%Y")
```

(Note, the format "%d/%m/%Y" will need to be adjusted to match the format of the user input data)

3.2 Adding additional date labels

Additional date information can be added with the function GetMoreDateInfo(). This function uses myDate and has an optional argument "firstMonth" that can be used to choose an alternative start month for the analysis.

The second step is to select the column that defines the time-period increment (or "season"). Note: the year must be a numeric field and the season must be a factor. Any user-defined season can be analysed. For example, selecting months as seasons, can be implemented by:

```
WQData$Season<-WQData$Month
```

A string describing the season names must also be specified.

```
SeasonString<-levels(WQData$Season)
```

3.3 Processing Censored Data

The third step assumes that the water quality measures contain less than and greater than signs that signify the data are censored (below detection limit is "<" and above the "reporting limit" ">"). These must be converted their face values + information concerning censoring. This is achieved with the function RemoveAlphaDetect() as follows:

```
NewValues <- RemoveAlphaDetect(WQData$Value)
```

This output is a data frame with three columns and as many rows as the input data frame. The columns represent the face value of the water quality measures (named RawValue), a logical indicating if the observation was censored (named Censored) and the type of censoring (less than (lt), greater than (gt) or not censored (not). This data frame is then concatenated to the original data with the following command:

```
WQData <- cbind.data.frame(WQData, NewValues)
```

The modified WQData data frame is shown in Figure 2.

	sID	Date	npID	Value	myDate	Year	Season	RawValue	Censored	CenType
1	Whareama River at Gauge	24/09/2003	NO3-N	0.273	2003-09-24	2003	Sep	0.273	FALSE	not
2	Whareama River at Gauge	10/10/2003	NO3-N	0.211	2003-10-10	2003	Oct	0.211	FALSE	not
3	Whareama River at Gauge	10/11/2003	NO3-N	<0.0100000	2003-11-10	2003	Nov	0.010	TRUE	lt
4	Whareama River at Gauge	8/12/2003	NO3-N	0.012	2003-12-08	2003	Dec	0.012	FALSE	not
5	Whareama River at Gauge	16/01/2004	NO3-N	<0.0100000	2004-01-16	2004	Jan	0.010	TRUE	lt
6	Whareama River at Gauge	19/02/2004	NO3-N	0.414	2004-02-19	2004	Feb	0.414	FALSE	not

Figure 2. Water quality data after initial set up steps.

4 Analysis

4.1 Inspecting the data

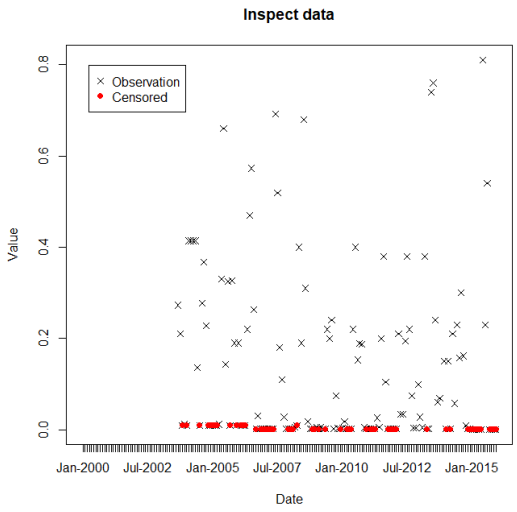
4.1.1 Inspect Data

The function InspectData() assists with inspection of the data. It is assumed that the trend analysis is for a specific time-period that is defined by the arguments Year and StartYear and EndYear. The function outputs a data frame that defines the number of observation occasions (nOccasions – i.e. a specific Year + Season combination) in the specified time-period, the number of actual observations (nData), the number of missed occasions (nMissing), the proportion of observations that are censored (propCen) and the number of observations with flow data (nFlow). The function takes the median value of observations if there are more than one observation in Year + Season combination. When there is more than one value in a Year + Season combination and one or more of these is censored, the function assigns the observations as Censored if the median value is less than or equal to the maximum censored value within that Year + Season combination.

The argument plotType is used to specify a plot of the data as either a time series or heat plot matrix (Years x Seasons). The InspectData() function is used to produce a time series plot with following commands:

```
x11(); InspectData(WQData, StartYear = 2000, EndYear = 2015, Year = "Year", plotType = "TimeSeries", main= "Inspect data")
```

(note, use Year="CustomYear" if not using calendar years)



	nOccasions	YearsInPeriod	nData	nMissing	firstYear	I	nYears	propCen	nFlow
1	192	16	147	45	2003		2015	13	0.3809524	0

Figure 3. Plot and data frame output from InspectData(). Note that the time series extends from the specified start data to the end date but that there were no observations until September 2003.

The InspectData() function is used to produce a matrix plot of the data with following commands:

```
x11(); InspectData(WQData, StartYear = 2000, EndYear = 2015, plo"Type =""Matrix",
"lotMat="RawData")
```

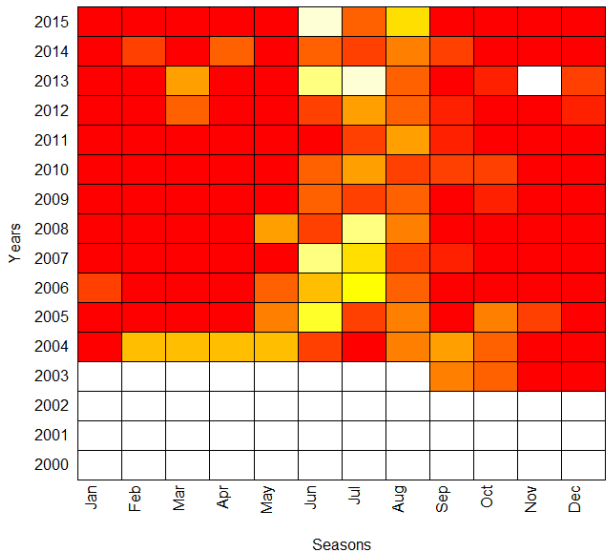


Figure 4. Heat matrix plot output from InspectData(). Note that the time series extends from the specified start data to the end date but that there were no observations until September 2003.

The InspectData() function is used to produce a matrix plot of the censoring occasions with following commands:

```
x11(); InspectData(WQData, StartYear = 2000, EndYear = 2015, plotType = "Matrix",
PlotMat = "Censored")
```

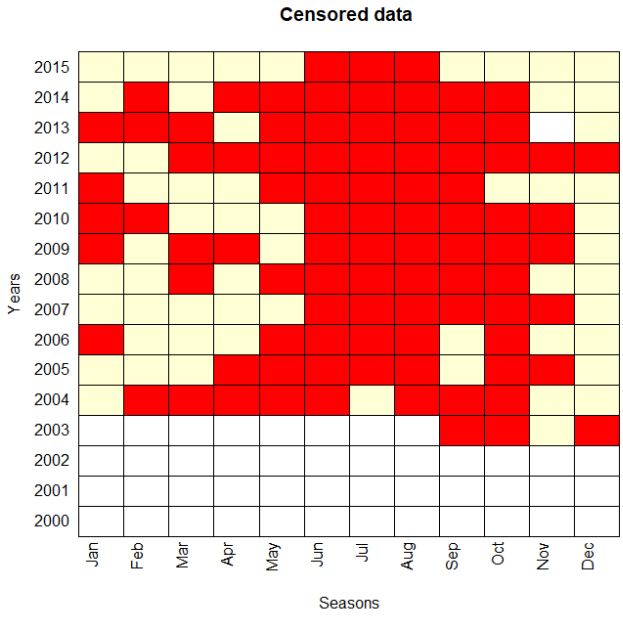


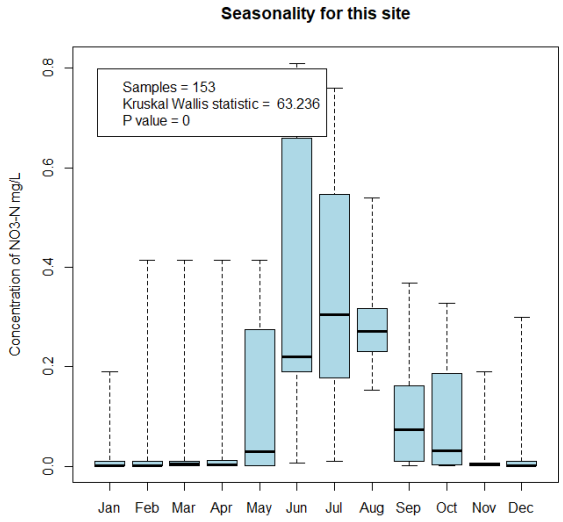
Figure 5. Heat matrix plot output showing the censored observations from InspectData().

Note, if there are no censored data the plot will be one colour; orange.

4.1.2 Seasonality Test

The function SeasonalityTest() tests if the data are seasonal. If the data are seasonal, the trend analysis should be seasonal (i.e. the Kendall test should be the seasonal Kendall test and the Sen slope should be a seasonal Sen slope). The SeasonalityTest() function performs a Kruskal Wallis test (non-parametric ANOVA) on the observations using season as the explanatory (categorical) variable. The function also produces a box plot of the data grouped by season.

Following the treatment suggested by Helsel (2012), all censored values and values less than the highest non-detect (<) are assigned the same low value and treated as ties. If the censored values are > then all non-detects and values higher than the lowest non-detect (>) are assigned the same high value and are treated as ties. Note, this treatment can have significant impacts on the seasonality tests if there are many observations below the maximum censored value (see Figure 7).




```
                         Observations    KWstat       pvalue
Kruskal-Wallis chi-squared      153 63.23643 2.309259e-09
```

Figure 6. Plot and data frame output from SeasonalityTest().

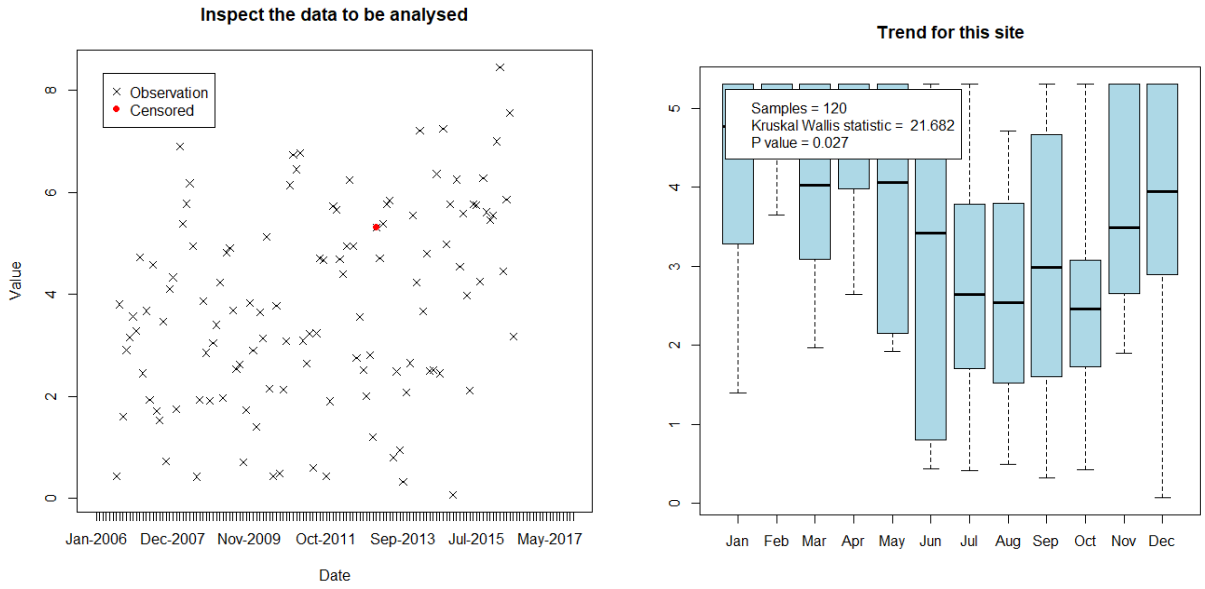


Figure 7: (a) timeseries of data with a high censored value. (b) impact on seasonality test (compared to Figure 6)

4.2 Mann Kendall

The function MannKendall() performs a Mann Kendall test of correlation between the observations and time. The function handles the censored values as described in Handling Censored Values section. The output is a data frame of the test results. Note that strictly, these data should be analysed using a seasonal Mann Kendall test because they are seasonal (Figure 6).

```
MannKendall(WQData)
```

```
     nObs    S      vars      D         tau          Z          p
1     152   279  387347.3 11476   0.02431161  0.4466779  0.6551076
```

Figure 8. Data frame output from the MannKendall function.

Note, the MannKendall() function and the other trend assessment functions below, have the argument HiCensor, If the argument HiCensor = TRUE the functions find the highest censoring limit (for left censored data) and set any recorded values that are less than this value to the highest censoring limit and set these as censored (at the highest censoring limit).

4.3 Sen slope

The function SenSlope() performs a non-parametric regression (Sen's slope estimator, also known as the Theil–Sen estimator) to the observations versus time. The function handles the censored values as described in Handling Censored Values section. The output is a data frame of the test results and, if specified, a plot showing the data and the fitted regression. Note that strictly, these data should be analysed using a seasonal Mann Kendall test because they are seasonal (Figure 6).

```
x11();SenSlope(WQData)
```

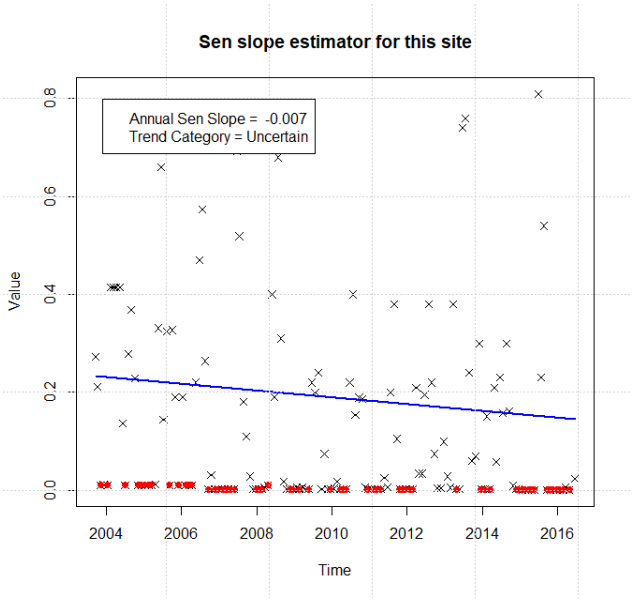


Figure 9. Plot and data frame output from SenSlope().

The probability output is the probability that the true trend is decreasing. This is evaluated by interpolating from a value of zero on the cumulative distribution of the ranked inter-observation slopes. For variables with low precision or large numbers of censored values, there can be many pairs of observations with the same value (i.e., ties) resulting in many inter-observation slopes equal to zero. This situation requires that choices are made when interpolating the probability. The LWP-Trends default is to return the mean probability of all inter-observation slopes that are equal to zero (Figure 10). Other possible choices are to use either the maximum or minimum probability for which inter-observation slopes are equal to zero (Figure 10). These probabilities are included in the outputs of the SenSlope() and SeasonalSenSlope functions as “ProbabilitiyMin”, and “ProbabilityMax”.

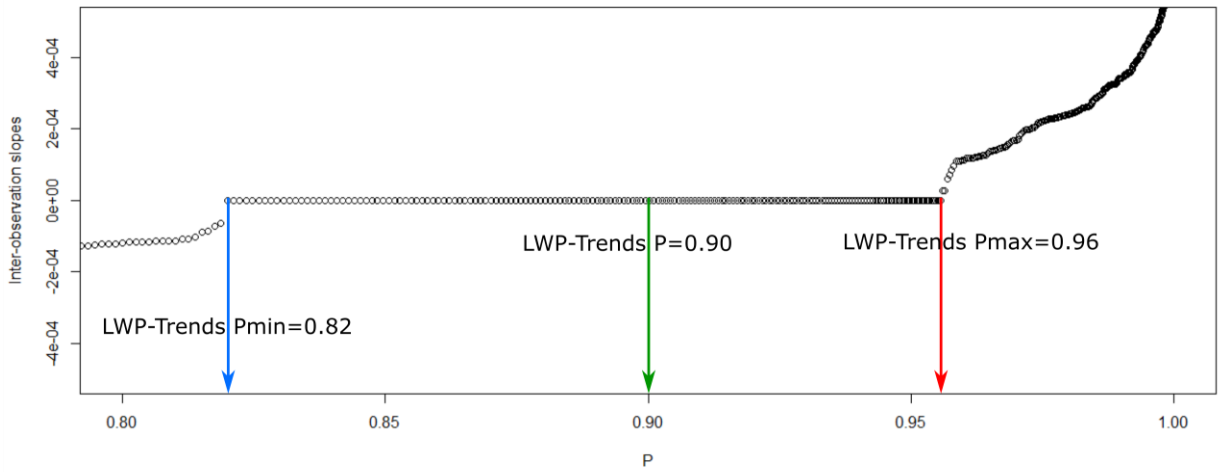


Figure 10: Example of cumulative distribution of the ranked inter-observation slopes – for site with negative Sen slope and many inter-observation ties.

The trend direction is a descriptive category that is evaluated based on the sign of the Sen slope. The categories are Decreasing, Increasing or Indeterminate (Sen slope zero).

Trend categories are evaluated based on consideration of both the direction of the Sen slope as well as the probability. An increasing trend is defined as Sen slope > 0 and Probability <0.05; a decreasing trend is defined as Sen slope < 0 and Probability >0.95. Sites with Sen slope = 0 or Probability <0.95 or >0.05 are classified as “uncertain”.

The percentage annual change in trend slope is calculated as the Sen slope divided by the median, multiplied by 100. The default is to calculate the median using the face values of the data (including censored data). Options also exist for calculating the percentage Sen slope as the Sen slope divided by a median value derived from imputed data, also to calculate a median value where all censored values are represented as half the censor value; see the function descriptions later in the document for details of how to invoke these options.

When there are insufficient data, the models will return NULL outputs – labelled “Not analysed”. This occurs when there are less than 5 valid observations or less than three unique observations.

4.4 Seasonal Mann Kendall

The function SeasonalKendall() performs a seasonal Mann Kendall test of correlation between the observations and time. The function handles the censored values as described in Handling Censored Values section. The output is a data frame of the test results. Note that these data have already been shown to be seasonal (Figure 6) so the seasonal Mann Kendall test is appropriate.

```
SeasonalKendall(WQData)
```

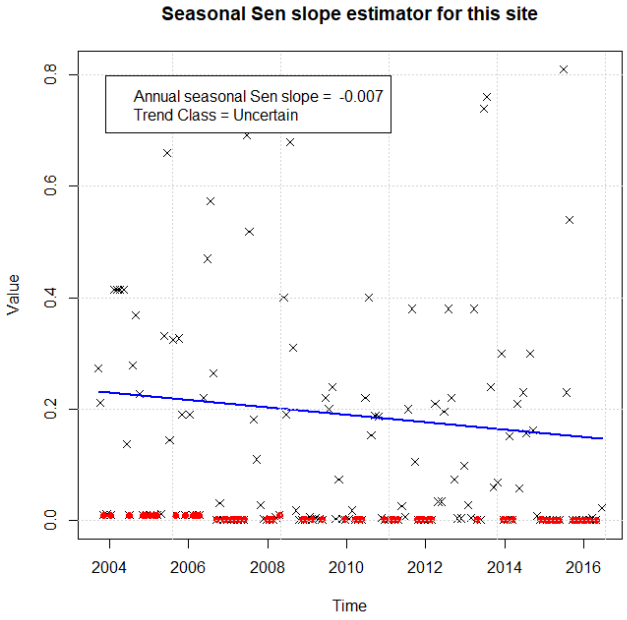
```
  nObs   S     VarS   D        tau         Z          p
1      152 -92 2706.667 888 -0.1036036 -1.749138 0.08026722
```

Figure 11. Data frame output from the SeasonalKendall function.

4.5 Seasonal Sen slope

The function SeasonalSenSlope() performs a seasonal version of the non-parametric regression (Sen's slope estimator) of the observations versus time. The function handles the censored values as described in Handling Censored Values section. The output is a data frame of the test results and, if specified, a plot showing the data and the fitted regression. Note that these data have already been shown to be seasonal (Figure 6) so the seasonal Sen slope is an appropriate estimator.

```
x11();SeasonalSenSlope(WQData)
```




```
  Median     VarS AnnualSenSlope Intercept         Lci         Uci TrendCategory TrendDirection Probability Percent.annual.change
1   0.01 2706.667       -0.00675 0.2320312 -0.02304104 0.003216728     Uncertain     Decreasing   0.8503533               -67.5
```

Figure 12. Plot and data frame output from SeasonalSenSlope().

The same additional outputs as described for the Sen slope are also provided for the seasonal Sen slope. Note, there are generally more “not analysed” sites for the seasonal Sen slope tests, as there must be sufficient data for each season.

4.6 Flow adjustment

The function AdjustValues() performs an adjustment of the data based on a covariate (for example flow if the data represents river concentrations). The function fits a variety of models to the observation versus covariate relationship. The user needs to consider which of these models is the most appropriate basis for adjustment. The function returns a data frame with the adjusted values (median + regression residuals) for each of the models. The column that represents the chosen model in this data frame is then used in any of the functions above. The adjustment is achieved with following command:

```
FlowAdjusted <- AdjustValues(myFrame=WQData2, method = c("LOESS"), ValuesToAdjust =
"RawValue", Span= c(0.2,0.4,0.7), Flow = "Flow", doPlot = T)
```

The adjusted output is shown on Figure 13 and the plots are shown in Figure 14.

```
head(FlowAdjusted)
```

```
  LOESS0.2 LOESS0.4 LOESS0.75 LOESS0.9
1 3.245812 3.167150  3.294772 3.340340
2 3.851416 4.185701  3.812076 3.729782
3 3.693242 3.677522  3.793812 3.833757
4 2.997264 3.338149  3.538863 3.489194
5 4.775816 4.611776  4.592621 4.597527
6 3.749104 3.887667  4.057188 4.000640
```

Figure 13. Data frame output by the AdjustValues function

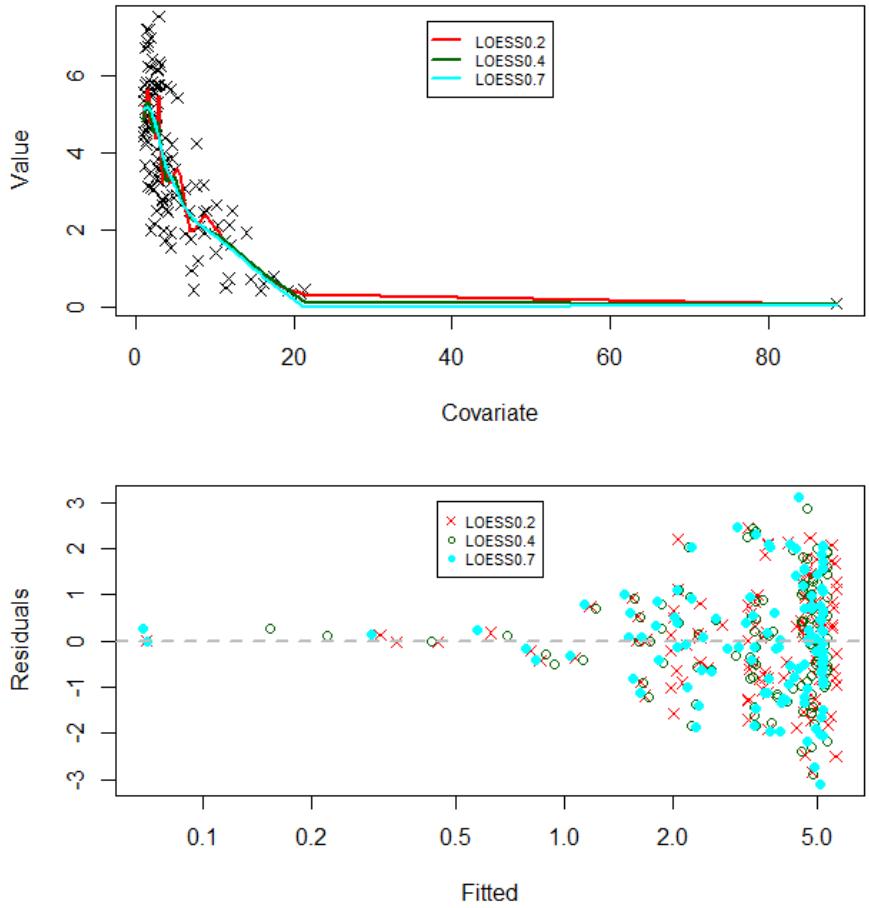
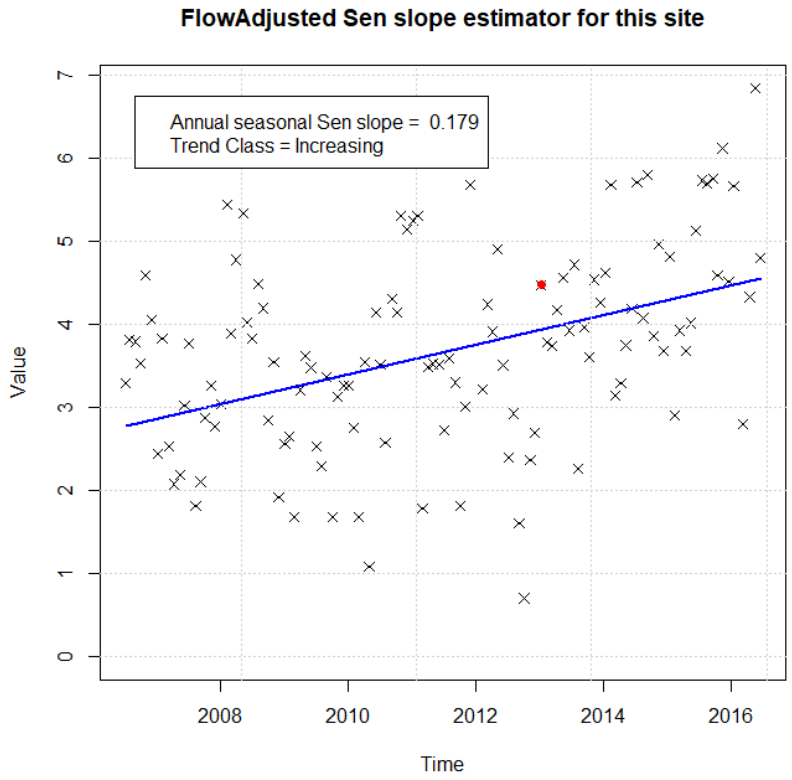


Figure 14. Output plots from the GetAdjustment function.

In the above example the loess0.7 model is the most appropriate (note that the other two models conform too closely to the data and produce concentration – flow relationships that are unrealistic). The adjusted values that are output by the GetAdjustment function are used in any of the above four trend analysis functions with following commands as follows:

```
WQData2$FlowAdjusted <- FlowAdjusted[, "LOESS0.75"]

SeasonalSenSlope(x=WQData2, ValuesToUse = "FlowAdjusted", main="FlowAdjusted Trend for
this site")
```



	Median	VarS	AnnualSenSlope	Intercept	Lci	Uci	TrendCategory	TrendDirection	Probability	Percent.annual.change
1	3.678149	1467	0.1792353	2.781485	0.1129991	0.2300573	Increasing	Increasing	1.775616e-05	0.04872976

Figure 15. Plot and data frame output from SeasonalSenSlope() using flow adjusted values.

5 Function descriptions

`GetMoreDateInfo`

Description

Takes dates and produces additional columns summarising, months, years, and quarters. If specified, the firstMonth can be used to shift the analysis year. This also automatically shifts the factor levels of the months and quarters to start from this month. The outputs from this function can be then used to specify a season for the analysis. Any other season definition would need to be manually shofted to reflect the custom year, by the user.

Usage

```
GetMoreDateInfo(x,firstMonth=1)
```

Arguments

`x`	Dataframe or vector containing myDate
`firstMonth`	Specify the first month (numeric (1:12)) for the analysis year

Value

A data frame with fields as follows (in addition to those columns in x)

`Year`	Calendar Year (numeric)
`Custom Year`	Custom Year (matches Calendar year of last month of the custom year) – only output if firstMonth!=1. numeric
`Month`	Month String (factor)
`Qtr`	Quarter string (factor)

`RemoveAlphaDetect`

Description

Takes a character timeseries of observed values that includes censored values (specified as the prefix “>” or “<”) and returns face values and information about the nature of the censoring.

Usage

```
RemoveAlphaDetect(x)
```

Arguments

`x`	Dataframe or vector containing Value
`AdjustDL`	Option to adjust face value of low censored values by a factor (optional, default = 1)
`AdjustUpper`	Option to adjust face value of high censored values by a factor (optional, default = 1)

Value

A data frame with fields as follows

`RawValue`	Numeric face value from x$Value
`Censored`	Logical: whether the observation is censored or not
`CenType`	Factor indicating the censor type (lt: less than; gt: greater than, not: not censored)

`InspectData`

Description

InspectData provides an analysis of the input data, providing summary statistics and optional graphs.

Usage

```
InspectData (x, plotType = c("TimeSeries", "Matrix"), PlotMat =
c("RawData","Censored"),Year = "Year", StartYear = 1990, EndYear =
2015, doPlot = TRUE, ...)
```

Arguments

`x`	Input data frame must contain: myDate, RawData, Censored
`plotType`	Select type of plot time series or matrix
`PlotMat`	The column to be plotted in the matrix (RawData or Censored)
`Year`	Column name for Year data to be used
`StartYear`	Sets the start year of the time series.
`EndYear`	Sets the end year of the time series.
`doPlot`	Produce a plot if TRUE
…	Passed to plot function

Value

A data frame and (optionally) a plot. Data frame fields as follows

`nOccasions`	Number of potential sampling occasions in the selected period
`YearsInPeriod`	Number of years in the period selected
`nData`	Number of actual sampling occasions in the period
`nMissing`	Number of missing observations for the selected period
`firstYear`	The year of the first observation
`lastyear`	The year of the last observation
`nYears`	The number of years with data in the sampling period
`propCen`	Proportion of observations that are censored
`nFlow`	The number of flow observations for the period

`SeasonalityTest`

Description

The function SeasonalityTest() tests if the data are seasonal. The SeasonalityTest() function performs a Kruskal Wallis test (non-parametric ANOVA) on the observations using season as the explanatory (categorical) variable. The function also produces a box plot of the data grouped by season.

Usage

```
SeasonalityTest(x, ValuesToUse="RawValue", do.plot=TRUE)
```

Arguments

`x`	Input data frame must contain, Season, "ValuesToUse",Centype, Censored
`ValuesToUse`	Select Column with data to use in the test
`do.plot`	Produce a plot if TRUE
…	Passed to plot function

Value

A data frame and (optionally) a plot. Data frame fields as follows

`Observations`	Number of observations
`KWStat`	the Kruskal-Wallis rank sum statistic.
`pvalue`	The p-value for the test (null hypothesis is that the data is NOT seasonal)

MannKendall OR SeasonalKendall

Description

The function MannKendall() performs a Mann Kendall test of correlation between the observations and time. The function SeasonalKendall() performs a seasonal Mann Kendall test of correlation between the observations and time, and should be used if the data are determined to be seasonal by SeasonalityTest().

Usage

```
MannKendall(x, ValuesToUse = "RawValue", Year = "Year",
HiCensor=FALSE)

OR

SeasonalKendall(x, ValuesToUse = "RawValue", Year = "Year",
HiCensor=FALSE)
```

Arguments

x	Input data frame must contain: myDate, Season,"ValuesToUse"
ValuesToUse	Select Column with data to use in the test
Year	Column name for Year data to be used
HiCensor	If TRUE the function finds the highest censoring limit (for left censored data) and sets any recorded values that are less than this value to the highest censoring limit and sets these as censored (at the highest censoring limit).

Value

A data frame with fields as follows

nObs	Number of observations
S	S-statistic
VarS	Variance
D	n * (n - 1)/2
tau	Kendall's tau
Z	Z-statistic
p	p-value

SenSlope OR SeasonalSenSlope

Description

The function SenSlope() performs a non-parametric regression (Sen's slope estimator, also known as the Theil–Sen estimator) of the observations versus time. The function SeasonalSenSlope() performs a seasonal version of the non-parametric regression (Sen's slope estimator) to the observations versus time, and should be used if the data are determined to be seasonal by SeasonalityTest().

Usage

```
MannKendall(x, ValuesToUse = "RawValue", ValuesToUseforMedian=NULL,
Year = "Year", HiCensor=FALSE, mymain=”My trend plot”)
```

Arguments

x	Input data frame must contain: myDate, Season,”ValuesToUse”,CenType
ValuesToUse	Select Column with data to use in the test
ValuesToUsefor Median	Default is to use “ValuesToUse”. Alternatives are to name another column (i.e. ImputedData), or to specify “HalfCen” (censored medians are set at half the censored value)
Year	Column name for Year data to be used
HiCensor	If TRUE the function finds the highest censoring limit (for left censored data) and sets any recorded values that are less than this value to the highest censoring limit and sets these as censored (at the highest censoring limit).
doPlot	Produce a plot if TRUE
mymain	Optional title for plot. Default uses npID*sID if they exist

Value

A data frame and (optionally) a plot. Data frame fields as follows

Median	Data Median (based on data from ValuestoUseforMedian)
VarS	Variance
AnnualSenSlope	Annual Sen slope (attribute units/year)
Intercept	Predicted value at time t=t[1]
Lci	Lower confidence interval for annual Sen slope
Uci	Upper confidence interval for annual Sen slope
TrendCategory	Trend Category
TrendDirection	Trend Direction
Probability	Probability of a decreasing trend (mean)
Probabilitymax	Probability of a decreasing trend (max)
Probabilitymin	Probability of a decreasing trend (min)
Percent.annual .change	Percent annual change in Sen slope (value between 0 and 1)

AdjustValues

Description

Performs an adjustment of the data based on covariate data (for example flow if the data represents river concentrations). The function fits one or more models to the observation versus covariate relationship. The user needs to consider which of these models is the most appropriate basis for adjustment.

Usage

```
AdjustValues(x, method = c("Gam", "LogLog", "LOESS"), ValuesToAdjust =
"RawValue", Covariate = "Flow", Span = c(0.5, 0.75), doPlot = T)
```

Arguments

x	Dataframe or vector containing: myDate, “ValuesToAdjust”,”Covariate”
method	Method to fit relationship between the observations and covariate. One or more of "Gam", "LogLog", "LOESS"),
ValuesToAdjust	Column name of observations to perform adjustment on
Covariate	Column name of the covariate to be used
Span	Vector of spans to be tested for LOESS models
doPlot	Produce a plot if TRUE

Value

A data frame and (optionally) a plot. Data frame fields as follows

myDate	Observation date
…	One column of adjusted observations for each method selected