Appendix A

Package 'Trends'

April 18, 2012

Version 0.1-2

Date 2012-04-18

Title Analysis of Data Collected from a Monitoring Network

Author Jason C. Fisher and Linda C. Davis

Maintainer Jason C. Fisher <jfisher@usgs.gov>

Depends R (>= 2.15.0), tcltk, MASS, NADA

SystemRequirements Tcl/Tk (>= 8.5)

Description This package is for identifying trends in data from multiple observation sites. Nonparametric regression is applied to both censored and uncensored data.

License Unlimited

URL https://github.com/jfisher-usgs/Trends

BugReports https://github.com/jfisher-usgs/Trends/issues

ByteCompile yes

R topics documented:

DrawPlot *Draw Single Plot*

Description

This function draws a single plot on a graphics device.

Usage

```
DrawPlot(d, tbl.par, cen.var = NULL,
         xlim = c(NA, NA), ylim = c(NA, NA),
         regr = NULL, regr.lower = NULL, regr.upper = NULL,
         regr.type = "Regression line",
         main = NULL, ylab = NULL, leg.box.col = "#FFFFFF",
         tick.lines = TRUE, p.value = NULL)
```

Arguments

d	data.frame; coordinate vectors of points to plot and a logical vector indicating those values that are censored. The first component contains date and time values of class POSIXct.
tbl.par	data.frame; parameter configuration table, see ShowParameters.
cen.var	character or integer; the name or index number of the logical component in d indicating whether observations are censored.
xlim	POSIXct; the x limits (x1, x2) of the plot.
ylim	numeric; the y limits (y1, y2) of the plot.
regr	function; a regression line, only drawn for single parameter plots.
regr.lower	function; lower confidence interval for regression.
regr.upper	function; upper confidence interval for regression.
regr.type	character; a label for the regression line, shown in legend.
main	character; a main title for the plot.
ylab	character; a label for the y axis.
leg.box.col	character; the background color for the legend box with semi-transparency supported for some graphics devices.
tick.lines	logical; if TRUE, horizontal and vertical lines are drawn at the major tick marks.
p.value	numeric; the estimated two-sided *p*-value of the regression line.

Details

Uncensored data is plotted as points and censored data as vertical dashed lines. Regression and optional confidence interval lines are only drawn if d contains a single parameter.

Author(s)

J.C. Fisher

See Also

plot.window, axis, axis.POSIXct, abline, polygon, lines, points, legend

Examples

```
f <- system.file("extdata/Example_Data.txt", package = "Trends")
d <- ReadTrendData(f)
d <- d[d$Site_id == 433657112563601, ]

f <- system.file("extdata/Example_Config_Parameters.txt", package = "Trends")
p <- read.table(file = f, header = TRUE, sep = "\t",
                stringsAsFactors = FALSE, comment.char = "", row.names = 1)

windows()

d1 <- d[, c("Datetime", "Temp", "AS")]
DrawPlot(d1, tbl.par = p, main = "Uncensored Data with Multiple Parameters",
         ylab = "Concentration, in milligrams per liter",
         leg.box.col = "#FFFFFFBB")

d2 <- d[, c("Datetime", "AS")]
fit <- lm(AS ~ Datetime, d2)
b <- fit$coefficients[[1]]
m <- fit$coefficients[[2]]
regr <- function(x) {m * as.numeric(x) + b}
regr.lower <- function(x) {m * as.numeric(x) + b - 0.1 * b}
regr.upper <- function(x) {m * as.numeric(x) + b + 0.1 * b}
DrawPlot(d2, tbl.par = p, regr = regr,
         regr.lower = regr.lower, regr.upper = regr.upper,
         regr.type = "Simple linear regression",
         main = "Uncensored Data with Single Parameter and Trend Lines",
         ylab = "Arsenic, in micrograms per liter")

d3 <- d[, c("Datetime", "AS", "AS_code")]
d3[, "AS_code"] <- d3[, "AS_code"] == 1
DrawPlot(d3, tbl.par = p, cen.var = 3, xlim = c("1/1/1985", NA), regr = regr,
         main = "Censored Data with Single Parameter and Trend Line",
         ylab = "Arsenic, in micrograms per liter")
```

GetPath *Get Path to File or Directory*

Description

This function is used to select a file or directory path.

Usage

```
GetPath(type, path = NULL, initial.dir = getwd())
```

Arguments

type	character; code used to identify the path type, see 'Details'.
path	character; a file or directory path. If NULL, a dialog pops up.
initial.dir	character; specifies the initial directory that should be displayed when the dialog pops up. Its default is the working directory.

Details

Options for path type include: "input_data", a data input file; "config_para", a parameter configuration input file; "config_plot", a plots configuration input file; "config_stat", a statistics configuration input file; "output_stat", a statistics output file; and "output_figs", a output directory where graphic files are saved. A check is made to confirm that the file or directory exists.

Author(s)

J.C. Fisher

Examples

```
f <- GetPath("input_data")
d <- GetPath("output_figs")
```

OpenGraphicsDevice *Open Graphics Device*

Description

This function starts the device driver for producing graphics.

Usage

```
OpenGraphicsDevice(figs.dir, file.name, gr.type,
                   w = 8.5, h = 11, p = 12, png.res = 300)
```

Arguments

figs.dir	character; the directory where graphic files are saved.
file.name	character; the name of the graphics file and excluding a file extension.
gr.type	character; the name of graphics driver, either "pdf", "postscript", "png", or "windows".
w, h	numeric; the (nominal) width and height of the canvas of the plotting window in inches. Default is 8.5 and 11, respectively.
p	numeric; the default point size of plotted text, its default is 12.
png.res	numeric; the nominal resolution in points per inch (ppi) which will be recorded in the PNG file, its default is 300 ppi to set the size of text and line widths.

Details

A "postscript" graphics device produces an EPS file and is recommended for producing figures of publication quality.

Value

A plot device is opened.

Author(s)

J.C. Fisher

See Also

windows, pdf, png, postscript

Examples

```
OpenGraphicsDevice(getwd(), "fig01", "pdf")
```

PlotTrendData *Plot Trend Data*

Description

This function queries and plots trend data.

Usage

```
PlotTrendData(d, site.names, sdate = NA, edate = NA, initial.dir = getwd(),
              file.par = NULL, file.plots = NULL, figs.dir = NULL,
              gr.type = "pdf")
```

Arguments

d	data.frame; trend data, see ReadTrendData.
site.names	character; an array of local site identifiers; if missing, all sites in the plots configuration file are used.
sdate, edate	character; start and end date corresponding to the period of record, in M/D/YYYY (%m/%d/%Y).
initial.dir	character; specifies the initial directory in the Open Files pop-up box; only used when file.par or file.plots is NULL.
file.par	character; file path for the parameters configuration table, see ShowParameters.
file.plots	character; file path for the plot configuration table, see 'Format'.
figs.dir	character; the directory where data plots are saved, only applicable when gr.type is not equal to "windows".
gr.type	character; see OpenGraphicsDevice, its default is "pdf".

Format

The imported plot configuration table requires the following fields: "Site_id", a unique numerical identifier used to access site data; "Site_name", a local site name; "Parameters", comma separated parameter identifiers corresponding to header fields in the data table, see ReadTrendData; "Min", the minimum y-axis plot limit; "Max", the maximum y-axis plot limit; and "Axis_title", the y-axis title. Each line of the table represents a single plot with unique point symbols drawn for each parameter in the "Parameters" field.

Details

A new graphics device is opened for each site and after four plots have been drawn.

Author(s)

J.C. Fisher and L.C. Davis

See Also

DrawPlot

Examples

```
f <- system.file("extdata/Example_Data.txt", package = "Trends")
d <- ReadTrendData(f)

file.par <- system.file("extdata/Example_Config_Parameters.txt",
                        package = "Trends")
file.plots <- system.file("extdata/Example_Config_Plots.txt",
                          package = "Trends")

PlotTrendData(d, site.names = c("ANP 6", "ARBOR TEST"),
              sdate = "01/01/1990", edate = "01/01/2010",
              file.par = file.par, file.plots = file.plots,
              gr.type = "windows")
```

ReadTrendData *Read Trend Data*

Description

This function reads table formatted data from a text file and creates a data frame from it.

Usage

```
ReadTrendData(file.data = NULL, initial.dir = getwd())
```

Arguments

file.data	character; the file path which the data are to be read from.
initial.dir	character; specifies the initial directory in the open files pop-up box; only used when file.data is NULL.

Format

The imported data table requires the following field names: "Site_id", "Site_name", "Dates", and "Times". The required date and time format is M/D/YYYY (%m/%d/%Y) and HM (%H%M), respectively. Parameter field names are arbitrary; however, they must be consistent with names specified in the configuration files.

Details

Parameter fields in the imported data table are read as class `character` and are checked for character codes in the first digit of each value. These fields are stripped of their single digit character codes and converted to class `numeric`. A new component is added to the data frame and populated with integer codes representing their respective character code. The variable name for this new component is based on a concatenation of the parameter name and `"_code"`. Codes are identified using the following criteria:

"0-9", "."	No character code	0
"<"	Less than, censored data	1
"E"	Estimated value	2
"V"	Sample contaminated	3
"U"	Undetectable	4

A warning is given if the character code is not recognized and its value set to `NA`. Values with character codes of `"V"` and `"U"` are set to `NA`.

Value

Returns a data frame with components:

`Site_id`	numeric; a unique site identifier.
`Site_name`	numeric; a local site name.
`Datetime`	POSIXct; the date and time corresponding to when the sample was collected.
`...`	numeric; parameter fields and integer codes.

Author(s)

J.C. Fisher and L.C. Davis

See Also

`read.table`

Examples

```
f <- system.file("extdata/Example_Data.txt", package = "Trends")
d <- ReadTrendData(f)
```

`RunTheilSen` *Run Theil-Sen Estimator*

Description

This function computes the Theil-Sen estimator, nonparametric regression line, and 95 percent confidence interval for uncensored data.

Usage

```
RunTheilSen(x, y, alpha = 0.05, pr = FALSE, xout = FALSE)
```

Arguments

x	numeric; a vector; such as the number of seconds since some origin date; for example, "1920-01-01 00:00:00.0".
y	numeric; a vector; such as parameter values.
alpha	numeric; Type I probability error, sometimes referred to as the confidence of the test, or the level of significance of the test.
pr	logical; should error messages be reported? Its default is FALSE.
xout	logical; if TRUE, leverage points are removed based on the minimum volume ellipsoid (MVE) method, an outlier detection method.

Details

All of the code used in RunTheilSen is provided by Rand R. Wilcox http://www-rcf.usc.edu/~rwilcox/. The code format has been slightly altered for readability purposes and packaged into a single wrapper function.

Value

Returns a list with a single 2 by 5 matrix component, regci. The matrix has column names of "ci.low", "ci.up", "Estimate", "S.E.", and "p-value"; and row names of "intercept" and "X". The two-sided *p*-value for the Theil-Sen estimator [2,5]; slope [2,3] and intercept [1,3] of the regression line; slope [2,1] and intercept [1,1] of the lower confidence interval; and slope [2,2] and intercept [1,2] of the upper confidence interval are included in the matrix.

Author(s)

R.R. Wilcox and J.C. Fisher

References

Wilcox, R.R., 2012, *Modern Statistics for the Social and Behavioral Sciences: A Practical Introduction*, New York, CRC Press, 840 p.

Examples

```
x <- c(504946800, 536482800, 694249200, 725871600, 757407600, 788943600,
       820479600, 852102000, 883638000, 915174000, 946710000, 978332400)
y <- c(12.000, 12.000, 15.000, 12.000, 14.000, 15.000, 16.508, 19.625,
       18.894, 20.500, 18.830, 21.540)
out <- RunTheilSen(x, y)$regci
```

RunTrendStats *Open Graphics Device*

Description

This function performs a statistical analysis on censored and uncensored data.

Usage

```
RunTrendStats(d, site.names, is.censored = FALSE, initial.dir = getwd(),
              file.par = NULL, file.stats = NULL, write.tbl.out = FALSE,
              file.out = NULL, figs.dir = NULL, gr.type = "pdf",
              cenken.tol = 1e-12, cenken.iter = 1e+6, dt.breaks = NULL,
              xout = FALSE, draw.ci = FALSE)
```

Arguments

`d`	data.frame; trend data, see `ReadTrendData`.
`site.names`	character; an array of local site names; if missing, all sites in the statistics configuration file are used.
`is.censored`	logical; if `TRUE`, the data is assumed to be censored, its default is `FALSE`, see 'Details'.
`initial.dir`	character; specifies the initial directory in the open files pop-up box.
`file.par`	character; file path for the parameters configuration table, see `ShowParameters`.
`file.stats`	character; file path for the statistics configuration table, see 'Format'.
`write.tbl.out`	logical; if `TRUE`, the output table is written to a text file, its default is `FALSE`.
`file.out`	character; file path for the statistics summary table, see 'Value'.
`figs.dir`	character; if `gr.type` not equal to `"windows"`, the directory where statistic plots are saved.
`gr.type`	character; see `OpenGraphicsDevice`, its default is `"pdf"`.
`cenken.tol`	numeric; a tolerance used as a stopping criteria, see 'Details'.
`cenken.iter`	numeric; maximum number of iterations, see 'Details'.
`dt.breaks`	character; an interval specification, one of `"sec"`, `"min"`, `"hour"`, `"day"`, `"week"`, `"month"`, `"quarter"` or `"year"`. Parameter values are averaged over this time interval.
`xout`	logical; if `TRUE`, leverage points for uncensored data are removed based on an outlier detection method.
`draw.ci`	logical; if `TRUE`, 95 percent confidence interval for uncensored data is drawn on plot.

Format

The imported statistics configuration table requires the following fields: `"Site_id"`, a unique numerical identifier used to access site data; `"Site_name"`, a local site identifier; `"Parameters"`, comma seperated parameter names corresponding to header fields in the data table, see `ReadTrendData`; `"Start_date"` and `"End_date"`, the start and end date in M/D/YYYY (`%m/%d/%Y`); and `"Remark"`, user comments (optional).

Details

The censored data is analyzed using functions provided in the **NADA** package. Summary statistics are calculated using `cenfit` and the Kendall's tau correlation coefficient and associated Akritas-Theil-Sen (ATS) nonparametric regression line are calculated using `kendallATS`, a hidden function in **NADA**. The ATS slope estimation uses an iterative bisection search with a stopping criteria defined by `cenken.tol` and `cenken.iter`.

The Theil-Sen estimator and trend line for uncensored data is calculated using robust statistic functions provided by R.R. Wilcox, see `RunTheilSen` for details.

A new graphics device is opened for each site and after four plots have been drawn.

Value

Returns a data frame with variables:

`Site_id`	factor; a unique identifier used to access site data.
`Site_name`	character; a local site name.
`Parameter`	character; parameter identifier.
`Start_date`	POSIXct; lower temporal limit.
`End_date`	POSIXct; upper temporal limit.
`n`	integer; number of non-`NA` observations.
`n_above_rl`	integer; number of observations with values above the recording limit, that is values coded with '<' or 'E', see `ReadTrendData`.
`n_cen`	integer; number of observations with censored values.
`mean`	numeric; mean.
`median`	numeric; median.
`min`	numeric; minimum value.
`max`	numeric; maximum value.
`std_dev`	numeric; standard deviation.
`len_record`	difftime; period of record.
`p_value`	numeric; estimated two-sided *p*-value, if below the acceptance level (5 percent), reject the null hypothesis that the parameter is statistically independent of time and accept the hypothesis that they are dependent.
`tau`	numeric; Kendall's tau (censored only).
`slope`	numeric; slope of nonparametric regression line, in parameter change per year.
`lower`	numeric; slope of lower confidence interval (uncensored only).
`upper`	numeric; slope of upper confidence interval (uncensored only).
`int`	numeric; intercept of nonparametric regression line.
`int_lower`	numeric; intercept of lower confidence interval (uncensored only).
`int_upper`	numeric; intercept of upper confidence interval (uncensored only).
`slope_percent`	numeric; percent change per year of nonparametric regression line.
`lower_percent`	numeric; percent change per year of lower confidence interval (uncensored only).
`upper_percent`	numeric; percent change per year of upper confidence interval (uncensored only).
`trend`	character; significant trends are indicated by a *p*-value less than or equal to 0.05, a 5 percent acceptance level. The sign of the `slope` indicates whether the significant trend is positive (`"+"`) or negative (`"-"`). *p*-values greater than 0.05 are specified as `"no trend"`.

Author(s)

J.C. Fisher and L.C. Davis

References

Helsel, D.R., 2005, *Nondectects and Data Analysis; Statistics for censored environmental data*, New Jersey, John Wiley and Sons, Inc., 250 p.

Lorenz, D.L., and others, 2011, USGS library for S-PLUS for Windows – Release 4.0: U.S. Geological Survey Open-File Report 2011-1130.

Examples

```
f <- system.file("extdata/Example_Data.txt", package = "Trends")
d <- ReadTrendData(f)
file.par <- system.file("extdata/Example_Config_Parameters.txt",
                        package = "Trends")

file.stats <- system.file("extdata/Example_Config_Censored.txt",
                          package = "Trends")
out <- RunTrendStats(d, site.names = c("ANP 9", "HIGHWAY 3"),
                     is.censored = TRUE, file.par = file.par,
                     file.stats = file.stats, gr.type = "windows")

file.stats <- system.file("extdata/Example_Config_Uncensored.txt",
                          package = "Trends")
out <- RunTrendStats(d, site.names = c("ANP 6", "ATOMIC CITY"),
                     is.censored = FALSE, file.par = file.par,
                     file.stats = file.stats, gr.type = "windows",
                     draw.ci = TRUE)
```

ShowParameters *Draw All Parameters in a Legend*

Description

This function draws all possible parameters within a legend, and should assist with the selection of unique symbols and colors for each parameter.

Usage

```
ShowParameters(file.par = NULL)
```

Arguments

`file.par`	character; file path for the parameter configuration table, see 'Format'

Format

The imported parameter configuration table requires the following fields: `"Parameter"`, a unique identifier for each parameter and corresponding to field names in the data table; `"Name"`, the local parameter name, used for legend text and axis labels; `"Units"`, the parameter units used for axis labels; `"pch"`, an integer specifying the symbol to use in plotting points; `"col"`, the color code; and `"bg"`, the background (fill) color for open plot symbols, `pch = 21:25`. Each line of the table represents a single unique parameter.

Author(s)

J.C. Fisher

See Also

`legend`

Examples

```
f <- system.file("extdata/Example_Config_Parameters.txt", package = "Trends")
ShowParameters(f)
```

Index