Package: smwrStats (via r-universe)

November 6, 2026

Version 0.7.6

Date 2017-07-27

Title R functions to support statistical methods in water resources

Depends smwrBase (>= 1.0), smwrGraphs (>= 1.0)

Suggests smwrData (>= 1.0), dataRetrieval, testthat

Imports car, leaps, stats, randtests

Description Useful tools for the analysis of hydrologic data, not inlcuding censored water-quality data.

License CC0

LazyLoad yes

BugReports https://github.com/USGS-R/smwrStats/issues

Collate 'smwrStats-package.R' 'allreg.R' 'ancovareg.R' 'binreg.R' 'confirm.R' 'confirm.seasonalPeak.R' 'cor_all.R' 'curvi.R' 'grubbs.R' 'grubbs.test.R' 'hosmerlemeshow.R' 'jackknifeMove.1.R' 'kensen.test.R' 'lecessie.R' 'makepredictcall.R' 'move.1.R' 'move.2.R' 'multicomp.test.R' 'multreg.R' 'optimBoxCox.R' 'percentile.R' 'plot.Stats.R' 'plot.htest.R' 'ppcc.test.R' 'predict.move.1.R' 'predict.move.2.R' 'predict.senSlope.R' 'predictMVUE.R' 'predictDuan.R' 'predictFerguson.R' 'press.R' 'print.MCT.R' 'print.ancovaReg.R' 'print.binaryReg.R' 'print.cor.all.R' 'print.lecessie.R' 'print.move.1.R' 'print.move.2.R' 'print.multReg.R' 'print.optimBoxCox.R' 'print.roc.R' 'print.seasonalPeak.R' 'print.senSlope.R' 'printCor.R' 'qtiles.CI.R' 'quantile.numeric.R' 'regken.R' 'rmse.R' 'roc.R' 'seaken.R' 'seasonalPeak.R' 'seasonalWave.R' 'seasonalWave.fit.R' 'seasonalWave.wt.R' 'selBestWave.R' 'senSlope.R' 'serial.test.R' 'seriesPlot.seaken.R' 'skew.R' 'sumStats.R' 'summary.cor.all.R' 'timeWeightedMean.R' 'trends.R' 'vif.R'

Config/pak/sysreqs cmake make libicu-dev

Repository https://ldecicco-usgs.r-universe.dev

Date/Publication 2022-10-05 20:04:40 UTC

RemoteUrl https://github.com/USGS-R/smwrStats

RemoteRef HEAD

RemoteSha dbd276f02579ed0c3bd9717c8784f2cf8b4bc5b2

Contents

smwrStats-package . . . 3
allReg . . . 5
ancovaReg . . . 7
binaryReg . . . 8
confirm . . . 9
confirm.seasonalPeak . . . 10
cor.all . . . 11
curvi . . . 13
grubbs.test . . . 14
hosmerLemeshow.test . . . 15
jackknifeMove.1 . . . 16
kensen.test . . . 17
leCessie.test . . . 19
makepredictcall.trends . . . 20
move.1 . . . 21
move.2 . . . 23
multicomp.test . . . 25
multReg . . . 26
optimBoxCox . . . 27
percentile . . . 28
plot.ancovaReg . . . 29
plot.htest . . . 32
ppcc.test . . . 33
predict.move.1 . . . 34
predict.move.2 . . . 35
predict.senSlope . . . 36
predictDuan . . . 37
press . . . 38
print.ancovaReg . . . 39
print.binaryreg . . . 39
print.cor.all . . . 40
print.lecessie . . . 41
print.MCT . . . 41
print.move.1 . . . 42
print.move.2 . . . 43
print.multReg . . . 43
print.optimBoxCox . . . 44
print.roc . . . 45
print.seasonalPeak . . . 45
print.senSlope . . . 46

printCor . . . 47
qgrubbs . . . 48
qtiles.CI . . . 49
quantile.numeric . . . 50
regken . . . 51
rmse . . . 52
roc . . . 54
seaken . . . 54
seasonalPeak . . . 56
seasonalWave . . . 58
seasonalWave.fit . . . 59
seasonalWave.wt . . . 60
selBestWave . . . 61
senSlope . . . 62
serial.test . . . 64
seriesPlot.seaken . . . 65
skew . . . 66
summary.cor.all . . . 67
sumStats . . . 68
timeWeightedMean . . . 69
trends . . . 70
vif . . . 71

Index **72**

`smwrStats-package` *General tools for hydrologic data and trend analysis.*

Description

Useful tools for the analysis of hydrologic data, not inlcuding censored water-quality data.

Details

Package:	smwrStats
Type:	Package
Version:	0.7.5
Date:	2016-01-21
License:	CC0

Regression applications:
allReg
binaryReg
hosmerLemeshow.test
leCessie.test
multReg

press
rmse
selBestWave
roc
vif
predictDuan
predictFerguson
predictMVUE
senSlope
seasonalPeak
seasonalWave

Record extension applications:
move.1
jackknifeMove.1
move.2
optimBoxCox

Trend applications:
curvi
kensen.test
regken
seaken
trends
serial.test
regken
trends

Summary statistics:
cor.all
printCor
percentile
quantile.numeric
qtiles.CI
skew
sumStats
timeWeightedMean
multicomp.test

Hypothesis tests:
grubbs.test
ppcc.test
serial.test

Author(s)

Dave Lorenz <lorenz@usgs.gov>

Maintainer: Dave Lorenz <lorenz@usgs.gov>

References

Helsel, D.R. and Hirsch, R.M., 2002, Statistical methods in water resources: U.S. Geological Survey Techniques of Water-Resources Investigations, book 4, chap. A3, 522 p.

Lorenz, D.L., in preparation. smwrStats—an R package for the analysis of hydrologic data, version 0.7.5.

`allReg` *All Subsets Regression*

Description

Creates a table of the best subsets of explanatory variables for a response variable.

Usage

```
allReg(x, y, wt = rep(1, nrow(x)), nmax = ncol(x), nbst = 3,
  na.rm.x = TRUE, lin.dep = 10)
```

Arguments

`x`	matrix of candidate explanatory variables.
`y`	the response variable.
`wt`	the weight variable if needed.
`nmax`	the maximum number of explanatory variables to include in the largest model.
`nbst`	the number of best models to determine for each subset size.
`na.rm.x`	logical, if `TRUE`, then rows with missing values in `x` and the corresponding elements in `y` will be removed prior to analysis. If `FALSE`, then missing values are not removed and the analysis will fail if there are missing values in `x`.
`lin.dep`	a value to protect against linear dependencies; the number of the number of observations must be greater than the number of columns in `x` plus `lin.dep`.

Value

A data frame containing these columns:

`model.formula`	the subset model formula
`nvars`	the size (number of variables in the subset model
`stderr`	the standard error of the subsbet model
`R2`	the coefficient of determination for the subset model
`adjr2`	the adjusted r-squared of the subbset model
`Cp`	Mallow's Cp for the subset model
`press`	the press statistic for the subset model

Note

This function is a wrapper for the `regsubsets` function in the leaps package.

References

Helsel, D.R. and Hirsch, R.M., 2002, Statistical methods in water resources: U.S. Geological Survey Techniques of Water-Resources Investigations, book 4, chap. A3, 522 p.

Mallow, C.L., 1973, Come comments of Cp: Technometrics, v. 15, p. 661–675.

Miller, A.J., 1990, Subset selection in regression in Monographs on Statistics and Applied Probability 40: London, Chapman and Hall.

See Also

`regsubsets`

Examples

```
# See the regression vignette for examples
.pager <- options("pager")
options(pager="console")
vignette(package="smwrStats")
options(.pager)
```

ancovaReg *Diagnostics for Analysis of Covariance*

Description

Computes diagnostic statistics for an analysis of covariance (ANCOVA) with a single factor variable.

Usage

```
ancovaReg(object, find.best = TRUE, trace = FALSE)
```

Arguments

object	the linear regression model object.
find.best	select the "best" subset of terms in the model?
trace	print the results of the selection process if find.best is TRUE?

Details

The input model object (object) can be the complete ancova model including all interaction terms or it can be any form of an ANCOVA model. Most often, if it is not the complete ancova model, then find.best should be FALSE.
The find.best option uses the step function to select the "best" subset of terms in the model. In general, this can be used to retain or drop significant interaction terms. It will not look at individual factor levels in the model.

Value

A list of class "ancovaReg" containing these components:

aovtab	the analysis of variance table from the original model
parmests	a summary of the final object.
vif	a named vector of variance inflation factors.
diagstats	a data.frame containing the observed values, predicted values, residuals, standardized residuals, studentized residuals, leverage, Cook's D, and dfits for each observation.
crit.val	a named vector of the critical values for leverage, Cook's D, and dfits.
flagobs	a logical vector indicating which observations exceeded at least one of the critical values.
object	the lm object.
x	the model matrix of explanatory variables.
factor.var	the name of the factor variable
x.fr	the model frame of explanatory variables.

If no factor variable is found in the final model, either because one was not specified or it was dropped from the model, then an object of class "multReg" is returned instead. See `multReg` for details.

Note

Objects of class "ancovaReg" have `print` and `plot` methods.

References

Draper, N.R. and Smith, H., 1998, Applied Regression Analysis, (3rd ed.): New York, Wiley, 724 p.

See Also

`lm`, `plot.ancovaReg`, `multReg`,

`binaryReg` *Diagnostics for Logistic Regression*

Description

Computes diagnostic statistics for logistic regression.

Usage

```
binaryReg(object, lc.max = 1000)
```

Arguments

`object`	the logistic regression model object.
`lc.max`	the lmaximum number of observations for the le Cessie test.

Details

Because the le Cessie test is very slow to compute for many observations, the test is not performed if there are more than `lc.max` observations.

Value

A list of class "binaryreg" containing these components:

`regsum`	the output from `summary(object)`
`Warning`	any warnings relevant to the model
`Factors`	information about factor explanatory variables
`Profile`	summary information about the coding of the response variable
`Hosmer`	output from the Hosmer-Lemeshow test on `object`

leCessie	output from the le Cessie-van Houwelingen test on object
PctCorrect	the classification table
Concordance	the concordance table
roc	output from the receiver operating characteristics test on object
diagstats	a data frame containing the response variable, the predicted response probability, the response residual, the deviance residuals, the Pearson residuals, the leverage, the value of Cook's D, and the dfits value
crit.val	the critical values for leverage, Cook's D, and dfits
flagobs	a logical value indicating which observaitons exceeded any one of the critical values
object	the object

Note

Logistic regression can be very useful alternative method for heavily censored water-quality data. The critical values for the test criteria are computed as: leverage, *3p/n*; Cook's D, median quantile for the *F* distribution with *p+1* and *n-p* degrees of freedonm; and dfits, the .01 quantile of the *grubbs* distribution for *n* observations, where *p* is the number of parameters estiamted in the regression and *n* is the number of observations.
Objects of class "binaryreg" have print and plot methods.

References

Harrell, F.E., Jr., 2001, Regression modeling strategies with applications to linear models, logistic regression and survival analysis: New York, N.Y., Springer, 568 p.

Helsel, D.R., and Hirsch, R.M., 2002, Statistical methods in water resources: U.S. Geological Survey Techniques of Water-Resources Investigations, book 4, chap. A3, 522 p.

McFadden, D., 1974, Conditional logit analysis of qualitative choice behavior: p. 105-142 in Zarembka, P. (ed.), Frontiers in Econometrics. London, Academic Press, 252 p.

See Also

roc, leCessie.test, hosmerLemeshow.test

---

confirm *Confirm an Analysis*

Description

Reviews/accepts the results of an analysis: method for "default" data.

Usage

```
confirm(x, ...)

## Default S3 method:
confirm(x, ...)
```

Arguments

x	the object to be confirmed
...	additional arguments required for specific methods

Value

The object returned depends on the specific method.

Note

The default method simply returns the object and issues a warning.

---

confirm.seasonalPeak *Confirm Seasonal Peak*

Description

Reviews/accepts the results of an analysis: method for "seasonalPeak" object.

Usage

```
## S3 method for class 'seasonalPeak'
confirm(x, all = FALSE, plot.only = FALSE, ...)
```

Arguments

x	a seasonalPeak object.
all	a logical value indicating whether to accept the peak without interactive user input or to force the user to process the peak. The default value is FALSE, forcing the user to process the peak. Can also be set to either 1 or 2, indicating the number of peaks.
plot.only	a logical value indicating that only a plot is desired. If TRUE, then x is returned invisibly and unchanged.
...	not used, required for compatibility with other methods.

Value

An object of class seasonalPeak. The confirmed object is a single value that represents the estimate of the timing of the peak and four or five aadditional ttributes.

NumPeaks: the number of seasonal peaks; either 1 or 2.
Models: candidate loading models. The number indicates the number of months of constituent loading.
hlife: candidate half-life values. The muner indicates the half-life in terms of months.
Confirmed: logical indicating that the object has not been confirmed.

If `NumPeaks` is 2, then an additional attirbute `Second.peak` that is a data frame of candidate parameters for the second peak is included. See `seasonalWave` for details.

References

Vecchia, A.V., Martin, J.D., and Gilliom, R.J., 2008, Modeling variability and trends in pesticide concentrations in streams: Journal of the American Water Resources Association, v. 44, no. 5, p. 1308-1324.

See Also

`seasonalWave`, `seasonalPeak`

Examples

```
## Not run:
library(smwrData)
data(QW05078470)
# Simply click on the identified peak, and enter 1 for a single peak.
confirm(with(QW05078470, seasonalPeak(dectime(DATES), P00665)))

## End(Not run)
```

---

cor.all *Compute Cross Correlations*

Description

Computes correlations among numeric data.

Usage

```
cor.all(data, method = "pearson", na.method = "pairwise",
  distribution = "normal")
```

Arguments

data	any rectangular object such as a data.frame or matrix.
method	a character string indicating which correlation coefficient is to be used. One of "pearson," "kendall," or "spearman," can be abbreviated.
na.method	a character string indicating which method to use for missing values. One of "fail," "omit," "pairwise," can be abbreviated.
distribution	a character string indicating the assumed distribution of the data. One of "normal," "lognormal," or "log1p", which can be abbreviated.

Details

The null hypothesis is that the data are not correlated with one another. The alternate hypothesis is that they are correlated with one another. This is a two-sided test. For other options, see cor.all.

If method is "pearson," then the correlation is based on Pearson's product moment correlation coefficient. If method is "kendall," then Kendall's tau is used to estimate a rank-based measure of association. If method is "spearman", then Spearman's rho is used to estimate the correlation of the ranks of the data. The last two methods may be used if the data do not necessarily come from a bivariate normal distribution.

If na.method is "fail," then cor.all stops if there are any missing numeric values. If it is "omit," then all rows with any missing values is removed before the correlations are computed. That option will always produce a correlation matrix that is positive definite. If na.method is "pairwise," then missing values are removed from each pairwise correlation.

If distribution is "normal," then the assumption for method = "pearson" is that the data are bivariate normal. If distribution is "lognormal," then the assumption for method = "pearson" is that the data are bivariate log-normal and all data are natural log-transformed. If distribution is "log1p," then the assumption for method = "pearson" is that the data are bivariate log-normal after adding 1 and all data are transformed using the log1p function. The data are transformed for any method, but only produce a different result for method = "pearson."

Value

An object of class "cor.all," which has these components:

estimates	a matrix of the correlations between each pair of numeric variables in data
p.values	a matrix of the attained p-values between each pair of numeric variables in data
counts	a matrix of observations in each pair of numeric variables in data
alternative	a character string indicating the alternative hypothesis, always "two.sided"
na.method	a character string indicating the method to handle missing values
method	a character string describing the method to compute the correlations
data.name	the name of the data set, derived from data
data	a data frame of the numeric variables
call.method	a character string indicating the method to compute the correlations
distribution	a character string indicating the distribution assumption of the data

Note

The `print`, `plot`, and `summary` methods are available for an object of class "cor.all."

References

Conover, W.J., 1980, Practical nonparametric statistics (2d ed.): New York, Wiley, 512 p.

Helsel, D.R. and Hirsch, R.M., 2002, Statistical methods in water resources: U.S. Geological Survey Techniques of Water-Resources Investigations, book 4, chap. A3, 522 p.

See Also

`cor.test`, `plot.cor.all`, `summary.cor.all`

Examples

```
## Not run:
library(smwrData)
data(TNLoads)
cor.all(TNLoads[, 1:5])
cor.all(TNLoads, method="spearman")

## End(Not run)
```

---

curvi *Curvi-Linear Trends*

Description

Generates a matrix for curvi-linear modeling of trends.

Usage

```
curvi(x, ..., style = c("mw", "se"))
```

Arguments

`x`	a vector of dates/times, assumed to be in dectime format. Missing values are permitted and result in corresponding missing values in the output.
`style`	either "mw" for midpoint time and halfwidth, or "se" for start and end times.
`...`	paired vectors specifying the midpoint of the trend and the halfwidth, or the start and end times.

Details

Curvi-linear trends are described in Appendix 1 (equations 1–9 through 1–11) in Vecchia (2005). The original form of the trend was described in terms of the midpoint of the trend and the halfwidth. Specifying the start and end times of the trend are added as an easy-to-use option.

Value

A matrix with one column for each of the trends sprecified in ....

Note

Curvi-linear trends provide a pleasing visual trend with a gradual transition between trends in contrast to linear trends with sharp changes at the endpoints.
Each trend is 0 prior to the start, and then increases to a maximum of 1 and maintain that maximum value after the end. The overall change is described by the regresison coefficient, rather than as a rate described by trends, for example.

References

Vecchia, A.V., 2005, Water-quality trend analysis and sampling desgin for streams in the Red River fo the North basin, Minnesota, Norht Dakota, and South Dakota, 1970–2001: U.S. Geolgical Survey Scientific Investigations Report 2005–5224, 54 p. Available on line at http://pubs.usgs.gov/sir/2005/5224/.

See Also

trends,

Examples

```
# Model with a single curvi-linear trend from 2001 through 2003
# First using midpoint and half width (default) and then start and end.
curvi(2000 + seq(0,20)/5, c(2002, 1))
curvi(2000 + seq(0,20)/5, c(2001, 2003), style="se")
```

---

grubbs.test *Test for Outlier*

Description

Tests for a single outlier in a sample from a normal distribution.

Usage

```
grubbs.test(x, alternate = "two.sided")
```

Arguments

x	a vector of numeric values. Missing values are allowed, but are ignored in the calculation.
alternate	the alternate hypothesis; must be "two.sided" for either a high or a low outlier, "high" to test only for a high outlier, or "low" to test only for a lopw outlier.

Value

An object of class "htest" having the following components:

`statistic`	the value of the test statistic.
`p.value`	the attained p-value for the test.
`data.name`	a character string describing the name of the data used in the test.
`alternative`	a description of the altrnative and null hypotheses.
`method`	a description of the method.

References

Grubbs, F., 1969, Procedures for Detecting Outlying Observations in Samples, Technometrics, v. 11, no. 1, pp. 1-21.

See Also

`pgrubbs`, `qgrubbs`

Examples

```
# A random sample with a high value
set.seed(100)
grubbstest <- rnorm(32)
grubbs.test(grubbstest)
qqnorm(grubbstest)
```

`hosmerLemeshow.test` *The Hosmer-Lemeshow Test*

Description

Performs the Hosmer-Lemeshow test for goodness-of-fit for a logistic regression model.

Usage

```
hosmerLemeshow.test(object, groups = 10)
```

Arguments

`object`	an object of class "glm" on which to perform the test.
`groups`	the number of groups to use for the test.

Value

An object of class "htest" having these components:

`method`	a description of the method.
`statistic`	the test statistic.
`p.value`	the attained p-level of the test statistic.
`data.name`	the name of `object`.
`alternative`	the alternate hypothesis—"some lack of fit."
`estimate`	a data frame of the size, expected value, and actual counts in each group. If the model has a single explanatory variable, then the mean value is included as column 4.

Note

The null hypothesis is "no lack of fit." Rejection of the null hypothesis indicates "some lack of fit."

References

Hosmer, D.W. and Lemeshow, S., 1980, Goodness-of-fit tests for the multiple logistic regression model: Communications in Statistics — Theory and Methods, v. 9, p. 1043–1069

See Also

`binaryReg`

`jackknifeMove.1` *Jackknife move.1*

Description

Computes the bias and variance of move.1 predictions using the jackknife estimator.

Usage

```
jackknifeMove.1(object, newdata, type = c("response", "link"))
```

Arguments

`object`	an object of class "move.1" on which to base the predicted values.
`newdata`	an optional data.frame in which to look for variables with which to predict. If omitted, then the calibration data are used; the response values are sorted from smallest to largest and `type` is set to "link" if it is not set to "response" in the call.
`type`	the type of prediction ("response" or "link"). See **Details**.

Details

If type is "response," then the predicted values are back-transformed. Otherwise, the predicted values are computed directly from the model equation.

Value

A vector of predictions matching newdata or the model data.

References

Lorenz, D.L., 2015, smwrStats-an R package for analyzing hydrologic data, version 0.7.0: U.S. Geological Survey Open-File Report 2015-XXXX, XX p.

See Also

move.1, predict.move.1

Examples

```
library(smwrData)
data(IonBalance)
# Build model for non missing Alkalinity
IB.move <- move.1(Anion_sum ~ Cation_sum, data=IonBalance, subset=abs(Pct_Diff) < 10)
print(IB.move)
# Predict Anion_sum for missing Alkalinity
predict(IB.move, IonBalance[1, ])
```

---

kensen.test *Test for a Trend*

Description

Tests for a temporal trend using Kendall's tau and computes the Sen slope estimate of the trend.

Usage

```
kensen.test(y, t, n.min = 10)
```

Arguments

y	the data collected over time. Missing values (NAs) are allowed and removed before computations.
t	the time corresponding to each observations in y. Missing values are allowed only where y is missing. These should be expressed as Julian or decimal time and must be strictly increasing.
n.min	the minimum number of observations for adjusting the p-value for serial correlation. Used when t are uniformly spaced to adjust the p-value for serial correlation. Any value larger than the number of observations in y or Inf will suppress the adjustment.

Value

An object of class "htest" containing the following components:

method	a description of the method.
statistic	the value of Kendall's tau.
p.value	the p-value.
estimate	a named vector containing the Sen estimate of the slope in units per year, the median value of the data, the median value of time, the number of observations, and if the serial correction is applied, the effective number of observations (n*).
data.name	a string containing the actual name of the input series.
coef	a vector of an estimate of the intercept and slope.
alternative	a character string describing alternative to the test ("two.sided").
null.value	the value for the hypothesized slope (0).

Note

A straight line of the form

```
 ytrend = sen.slope * (
t - median.time ) + median.data
```

may be used as a trend line for graphically portraying or detrending the data. It goes through the point

```
 (t,y) = ( median.time , median.data )
```

with slope sen.slope.

Many tied values can cause misleading results.

The p-values for uniformly spaced data (t values unit value like years) are adjusted for lag-1 autoregressive serial correlation according to the method described by Yue and Wang (2004) that adjusts for trend. In keeping with the logic of seaken, the p-value adjustment is never performed for fewer than 10 observations. The user can suppress the adjustment by setting the value of n.min to Inf.

References

Conover, W.J., 1980, Practical nonparametric statistics (2d ed.): New York, Wiley, 512 p.

Helsel, D.R., and Hirsch, R.M., 2002, Statistical methods in water resources: U.S. Geological Survey Techniques of Water-Resources Investigations, book 4, chap. A3, 522 p.

Hirsch, R.M., Alexander, R.B. , and Smith, R.A., 1991, Selection of methods for the detection and estimation of trends in water quality: Water Resources Research, v. 27 p. 803–813.

Kendall, M.G., 1938, A new measure of rank correlation: Biometrika v. 30, p. 81–89.

Kendall, M.G., 1976, Rank correlation methods (4th ed.): London, Griffin, 202 p.

Sen, P.K., 1968, Estimates of regression coefficient based on Kendall's tau: Journal of the American Statisical Association, v. 63, p. 1379–1389.

Yue, S. and Wang. C., 2004, The Mann-Kendall test modified by effective sample size to detect trend in serially correlated hydrological series: Water Resources Management v. 18, p. 201-218.

See Also

dectime, seaken

Examples

```
## Not run:
library(smwrData)
data(SaddlePeaks)
with(SaddlePeaks, kensen.test(Flow, Year))

## End(Not run)
```

---

leCessie.test *The le Cessie-van Houwelingen Test*

Description

Performs the le Cressie-van Houwelingen test for goodness-of-fit for a logistic regression model.

Usage

```
leCessie.test(object, bandwidth, newterms)
```

Arguments

object	an object of class "glm" on which to perform the test.
bandwidth	the bandwidth for smoothing the residuals.
newterms	any new variables to add to the model. Expressed as the right-hand side of a formula.

Details

If bandwidth is missing, then the mean distance between observations is used.

Value

An object of class "lecessie" having these components:

`method`	a description of the method.
`statistic`	the test statistic.
`parameters`	the degrees of freedom of the chi-squared test.
`p.value`	the attained p-level of the test statistic.
`data.name`	the name of `object`.
`alternative`	the alternate hypothesis–"some lack of fit."
`estimate`	the estimated values for the test.
`object`	the original `object`.
`target.object`	the `object` with any added `newterms`.
`bandwidth`	the bandwidth used for smoothing the residuals.
`max.distance`	the maximum distance between observations.
`smoothed.residuals`	the smoothed residuals.
`distance.matrix`	a matrix of the distances between observations.
`hat`	the hat matrix.

Note

The null hypothesis is "no lack of fit." Rejection of the null hypothesis indicates "some lack of fit."

References

le Cessie, S. and van Houwelingen, H.C., 1995, Testing the fit of a regression model via score tests in random effects models: Biometrics, v. 51, p 600-614.

See Also

`binaryReg`

`makepredictcall.trends` *Utility Function for Safe Prediction*

Description

Creates the right matrices for `model.frame.default` when predicting from models with `trends` terms. Used only internally.

Usage

```
## S3 method for class 'trends'
makepredictcall(var, call)
```

Arguments

var	a variable.
call	the term in the formula, as a call.

Value

A replacement for call for the prediction variable.

---

move.1 *Maintenance of Variance Extension, Type 1*

Description

Calculates the Maintenance Of Variance Extension, Type 1 (MOVE.1) for record extension by fitting a Line of Organic Correlation (LOC) regression model.

Usage

```
move.1(formula, data, subset, na.action, distribution = "normal")
```

Arguments

formula	a formula object with the response variable on the left of the ~ operator and a single explantory variable on the right.
data	the data frame containing the variables named in formula.
subset	an optional vector specifying a subset of observations to be used in the fitting process.
na.action	a function that indicates what should happen when the data contain missing values (NAs). The default is set by the na.action setting of options and is na.fail if that is unset. Ohter possible options include na.exclude and na.omit.
distribution	either "normal," "lognormal," or "commonlog" indicating nature of the bivariate distribution, See **Details**.

Details

If distribution is "normal," then the data in x and y are assumed to have a bivariate normal distribution. Otherwise, they are assumed to have a bivariate log-normal distribution and a logarithmic transform is applied to both x and y before analysis. The natural logarithm is used if distribution is "lognormal" and the commmon logarithm is used if distribution is "commonlog."

Value

An object of class "move.1" having these components:

coefficients	the intercept and slope of the line describing the fit.
na.action	a character string indicating the special handling of NAs.
R	Pearson's correlation coefficient.
call	the matched call to move.1.
fitted.values	the fitted LOC values for the response.
residuals	a 2-column matrix containing the signed distance from the predicted to to the corresponding y and x values.
x	the (possibly transformed) values for x.
y	the (possibly transformed) values for y.
x.stats	the mean and standard deviation of the (possibly transformed) values for x.
y.stats	the mean and standard deviation of the (possibly transformed) values for y.
var.names	the names of y and x.
model	the model frame.

Note

Objects of class "move.1" have print, predict, and plot methods.

References

Hirsch, R.M., 1982, A comparison of four streamflow record extension techniques: Water Resources Research, v. 18, p. 1081–1088.

See Also

predict.move.1, plot.move.1

Examples

```
library(smwrData)
data(IonBalance)
# Build model for non missing Alkalinity
IB.move <- move.1(Anion_sum ~ Cation_sum, data=IonBalance, subset=abs(Pct_Diff) < 10)
print(IB.move)
```

move.2 *Maintenance of Variance Extension, Type 2*

Description

Calculates the Maintenance Of Variance Extension, Type 2 (MOVE.2) for record extension by fitting a Line of Organic Correlation (LOC) regression model.

Usage

```
move.2(formula, data, subset, distribution = "normal", lag = 0)
```

Arguments

formula	a formula object with the response variable on the left of the ~ operator and a single explantory variable on the right.
data	the data frame containing the variables named in formula.
subset	an optional vector specifying a subset of observations to be used in the fitting process.
distribution	either "normal," "lognormal," "commonlog," "log1p," or an object of class "optimBoxCox" indicating the nature of the bivariate distribution, see **Details**.
lag	the number of days to account for time of travel between the explanatory and response sites. If the explanatory site is upstream, then lag can be positive, otherwise, lag can be negative to account for the travel time between the sites.

Details

MOVE.2 has a necessarily predefined method for missing values—the response variable is assumed to contain missing values and they are the values to be estimated by the model equation. For the function move.2, missing values in the explanatory variable are excluded from the computations.

If distribution is "normal," then the data in the explanatory variable and the response variable are assumed to have a bivariate normal distribution. Otherwise, they are assumed to have a bivariate log-normal distribution and a logarithmic transform is applied to both the explanatory variable and the response variable before analysis. The natural logarithm is used if distribution is "lognormal" and the commmon logarithm is used if distribution is "commonlog." If either the response or the explanatory has zero values, then the "log1p" option can be used. That option adds 1 to each value and then computes the natural logarithm. Alternatively, the output from optimBoxCox that contains both the response and explanatory variables can be supplied to transform those variables by other than a logarithmic transform.

Value

An object of class "move.2" having these components:

coefficients	the intercept and slope of the line describing the fit.
R	Pearson's correlation coefficient.

p.value	the p-value from the correlation test, given a two-sided alternate hypothesis.
call	the matched call to move.1.
fitted.values	the fitted LOC values for the response.
residuals	a 2-column matrix containing the signed distance from the predicted to to the corresponding the response variable and the explanatory variable values.
x	the (possibly transformed and lagged) values for the explanatory variable.
y	the (possibly transformed) values for the response variable.
lag	the value of the lag argument.
xstats	the mean and standard deviation of the (possibly transformed) values for the explanatory variable.
ystats	the mean and standard deviation of the (possibly transformed) values for the response variable.
var.names	the names of the response variable and the explanatory variable.
model	the model frame.
data	the data frame supplied in data.
distribution	the value supplied in distribution.

Note

Objects of class "move.2" have print, predict, and plot methods.

References

Hirsch, R.M., 1982, A comparison of four streamflow record extension techniques: Water Resources Research, v. 18, p. 1081–1088.
Moog, D.B., Whiting, P.J., and Thomas, R.B., 1999, Streamflow record extension using power transformations and applicaitons to sediment transport: Water Resources Research, v. 35, p 243–254.

See Also

predict.move.2, plot.move.2, optimBoxCox

Examples

```
## Not run:
# See the vignette:
vignette("RecordExtension", package="smwrStats")

## End(Not run)
```

multicomp.test *Multiple Comparisons*

Description

Performs multiple comparison tests among groups of data. The tests may be either parametric (Yandell, 1997), nonparametric (Higgins, 2004), or Dunn's nonparametric (Glantz, 2005).

Usage

```
multicomp.test(x, g, method = "parametric", critical.value = "",
  alpha = 0.05)
```

Arguments

x	the numeric vector of observations. Missing values (NAs) are allowed and removed before the test is performed.
g	any group vector for the observations. Missing values (NAs) are allowed and removed before the test is performed.
method	a character string describing the test. Only the first character is necessary. See **Details**.
critical.value	a character string describing the method to use for determining the critical value. Only the first character is necessary. See **Details**.
alpha	the significance level of the test. See **Note**.

Details

The choices for method are "parametric," "nonparametric," and "dunn." If the method is "parametric," then the comparisons are based on the means and variances of the raw data and the valid choices for critical.value are "tukey" (default), "bonferroni," or "lsd." Otherwise, the comparisons are based on the ranks of the data. Valid choices for critical.value are "tukey" (default), "bonferroni," or "lsd" when method is "nonparametric" and "sidak" (default) or "bonferroni" when method is "dunn." The basic diffference between the default nonparametric method and Dunn's nonparametric method is in the handling of ties.

Value

An object of class MCT containing the following components:

title	a description of the test.
cv.method	the method used to compute the critical value.
alpha	the value of alpha.
crit.value	the critical value for the pairwise comparisons.
response	the name of the response variable.
groups	the name of the group variable.

means	the means for each group.
sizes	the number of observations in each group.
table	the table of the results of the pairwise comparisons.
assoc	a data frame containing the possible association for each group.

Note

All computations of the variance for unequal group sizes are based on the harmonic mean as described in Yandell (1997). That adjustment is only approximate when `critical.value` is "tukey" and `method` is "parametric" but useful when the design is slightly unbalanced.
The default nonparametric method `method = "nonparametric"` is only assymptotically unbiased when some data are tied. For smaller data sets with small numbers of ties, it may be preferable to use Dunn's nonparametric method `method = "dunn."`

References

Glantz, S.A., 2005, Primer of biostatistics: McGraw Hill, New York, 520 p.

Higgins, J.J., 2004, Introduction to modern nonparametric statistics: Pacific Grove, Calif., Brooks/Cole, 384 p.

Yandell, B.S., 1997, Practical data analysis for designed experiments: London, United Kingdom, Chapman & Hall, 437 p.

`multReg` *Diagnostics for Linear Regression*

Description

Computes diagnostics for linear regression.

Usage

```
multReg(object)
```

Arguments

object	the linear regression model object

Value

A object of class "multReg" having the following components:

aovtab	the analysis of variance table, using the type II sum of squares. See **Note**.
parmests	a summary of `object`.
vif	a named vector of variance inflation factors.

`diagstats`	a data.frame containing the observed values, predicted values, residuals, standardized residuals, studentized residuals, leverage, Cook's D, and dfits for each observation.
`crit.val`	a named vector of the critical values for leverage, Cook's D, and dfits. See **Note**
`flagobs`	a logical vector indicating which observations exceeded at least one of the critical values.
`object`	the `lm` object.
`x`	the model matrix of explanatory variables.

Note

The type II sum of squares are calculated according to the principle of marginality, testing each term after all others, except ignoring the term's higher-order relatives. This type sum of squares is useful for assessing the overall marginal effect of each term in the model.

The critical values for the test criteria are computed as: leverage, *3p/n*; Cook's D, median quantile for the *F* distribution with *p+1* and *n-p* degrees of freedom; and dfits, the .01 quantile of the *grubbs* distribution for *n* observations time the square root of (p/n), where *p* is the number of parameters estimated in the regression and *n* is the number of observations.

Objects of class "multReg" have `print` and `plot` methods.

References

Draper, N.R. and Smith, H., 1998, Applied Regression Analysis, (3rd ed.): New York, Wiley, 724 p.

See Also

`lm`, `plot.multReg`,

---

optimBoxCox *Multivariate Unconditional Box-Cox Transformations*

Description

Computes Box-Cox transformations that maximimize the log likelihood of the transformations.

Usage

```
optimBoxCox(X, start = NULL)
```

Arguments

`X`	a data frame or matrix of the data to find the optimized Box-Cox transforms to produce multivariate normality. Can also be a numeric vector for a simple Box-Cox transform to normality.
`start`	a numeric vector of length matching the number of columns in `X` to provide starting values for the Box-Cox transforms.

Value

An object of class "optimBoxCox" having these components:

`start`	the starting values for the Box-Cox transformations.
`criterion`	the log-likelihood of the Box-Cox transformations.
`names`	the names of the columns.
`lambda`	the values of the Box-Cox transformations.
`stderr`	the standard errors of the Box-Cox transformations.
`return.code`	the convergence value returned by `optim`.
`gm`	the geometric means of the data in `X`.
`data`	the data in `X` with missing values removed.

Note

The maximum likelihood estimate of the Box-Cox transformations corresponds to the minimum determinant of the variance-covariance matrix of the transformed `X`. The methodology is described in Andrews and others (1971).

References

Andrews, D.F., Gnanadesikan, R., and Warner, J.L., 1971, Transformations of multivariate data: Biometrics, v. 27, p. 825–840.

See Also

`boxCox`, `optim`

`percentile` *Empirical Cumulative Percent*

Description

Computes the empirical cumulative percent or percent exceedance of observed data for specific values.

Usage

```
percentile(x, q, test = ">=", na.rm = TRUE, percent = TRUE, ...)

## Default S3 method:
percentile(x, q, test = ">=", na.rm = TRUE,
  percent = TRUE, ...)
```

Arguments

x	a numeric vector representing the observed data.
q	a vector of quantiles for which the cumulative percent or percent exceedence is desired.
test	a character string indicating the test. The default value, '>=,' is the percent equalling or exceeding the quantile and '<' would return the cumulative percent below the quantile.
na.rm	a logical value indication whether missing values (NAs) should be removed or not. If na.rm is FALSE and there are missing values in x, then the result will be NA. The default value is TRUE.
percent	a logical value indicating whether the result should be expressed as a percent or proportion. The default value, TRUE, will express the result as a percent.
...	not used, required for method function

Value

A named vector as long as q corresponding to the requested value.

Note

The stats package contains the ecdf function that performs a similar function when test is "<=."

See Also

ecdf

Examples

```
set.seed(2342)
Xr <- rlnorm(24)
# The percentage of the observarions greater than or equal to 2
percentile(Xr, 1:5)
```

---

plot.ancovaReg *Diagnostic Plots*

Description

Produce a series of diagnostic plots for statistical analyses.

Usage

```
## S3 method for class 'ancovaReg'
plot(x, which = "All", set.up = TRUE, span = 0.8, ...)

## S3 method for class 'binaryreg'
plot(x, which = 2:5, set.up = TRUE, bandw = 0.3, ...)

## S3 method for class 'cor.all'
plot(x, which = "All", set.up = TRUE, ...)

## S3 method for class 'lecessie'
plot(x, which = "All", set.up = TRUE, ...)

## S3 method for class 'move.1'
plot(x, which = "All", set.up = TRUE, span = 0.8, ...)

## S3 method for class 'move.2'
plot(x, which = "All", set.up = TRUE, span = 0.8, ...)

## S3 method for class 'multReg'
plot(x, which = "All", set.up = TRUE, span = 1, ...)

## S3 method for class 'senSlope'
plot(x, which = "All", set.up = TRUE, span = 0.8, ...)

## S3 method for class 'roc'
plot(x, which = "All", set.up = TRUE, ...)
```

Arguments

x	the object to be plotted.
which	a character string or sequence of integers indicating which diagnostic plots to produce. See **Details**.
set.up	set up the graphics page? Set to FALSE if the graphics page has been set up with a call to setPage.
span	the smoothing parameter for loess.smooth.
bandw	the bandwidth for kernel smoothing for the Hosmer-Lemeshow plot.
...	not used, required for method function. These diagnostic plots have fixed graphics output.

Details

For objects of class "ancovaReg" and "multReg," the argument which can be a character string, "All" or any of a sequence of numbers from 1 to 8. If it is "All," then all plots are produced. For class "multReg," can also be the name of an explanatory variable so that a residual dependence plot is created for a single variable.
Numeric values for which:

1. Fitted with separate factor levels vs. Observed
2. Fitted vs. Residual
3. S-L plot
4. A correlogram if dates are available in the model or in the data set
5. Q-normal and Tukey boxplots for each factor level
6. Influence plot
7. Outliers plot
8. Residual dependence plots for each explanatory variable

For objects of class "binaryreg," the argument `which` can be a character string, "All" or any of a sequence of numbers from 1 to 5. If it is "All," then all plots are produced. By default, the le Cessie-van Houwelingen diagnotic plot is not shown as it can be difficult to interpret. Numeric values for `which`:

1. Le Cessie-van Houwelingen overall fit
2. Overall fit
3. Classification plot
4. ROC plot
5. Lin-Wei-Ying partial residual plots

For objects of class "cor.all," the argument `which` must be either "All," "Lower," or the name of one of the variables. If `which` is "All", then the full scatter plot matrix is shown if there are 4 or fewer variables, otherwise indivdual paired plots are shown. If `which` is "Lower", then the lower part of the scatter plot matrix is shown if there are 5 or fewer variables, otherwise indivdual paired plots are shown. If `which` is the name of a variable, then only the scatter plots of that variable and all others are shown.

For objects of class "lecessie," the argument `which` must be either "All," "First," the name of one or more of the variables, or a number indicating which variable to plot. If `which` is "All", then the fitted vs. Residual and all partial residual plots are created. If `which` is "First", then only the fitted vs. Residual plot is created. If `which` is one or more of the variable names, then those partial residual plots are created. If `which` is numeric, then the fitted vs. Residual (1) sequentially numbered partial residual plots are created.

For objects of class "move.1" or "move.2," the argument `which` can be a character string, "All" or any of a sequence of numbers from 1 to 3. If it is "All," then all plots are produced. Numeric values for `which`:

1. x on y and y on x
2. The LOC regression line with ellipse
3. Q-Q plot of x and y

For objects of class "roc," the argument `which` can be a character string, "All" or 1. The only graph is the receiver operating characteristics curve for a logistic regression.

For objects of class "senSlope," the argument `which` can be a character string, "All" or 1. The only graph avaialble is a scatter plot of the data with the regression line in green and a smoothed line in cyan.

Value

The object x is returned invisibly.

References

le Cessie, S. and van Houwelingen, H.C., 1995, Testing the fit of a regression model via score tests in random effects models: Biometrics, v. 51, p. 600–614.
Lin, D.Y., Wei, L.J., and Ying, Z., 2002, Model-checking techniques based on cumulative residuals: Biometrics, v. 58, p. 1–12.

See Also

ancovaReg, binaryReg, cor.all, leCessie.test, move.1, move.2, multReg, roc, senSlope, loess.smooth, setPage

---

plot.htest *Diagnostic Plots*

Description

Creates diagnostic plots for selected hypothesis tests.

Usage

```
## S3 method for class 'htest'
plot(x, which = "All", set.up = TRUE, ...)

## S3 method for class 'kensen'
plot(x, which = "All", set.up = TRUE, ...)

## S3 method for class 'ppcc'
plot(x, which = "All", set.up = TRUE, ...)
```

Arguments

x	an object having classes "htest" and some other class for which a plot method exists.
which	either "All" or a number indicating which plot, see **Details**.
set.up	set up the graphics page?
...	not used, required for other methods.

Details

The kensen method for plot graphs the y and t data with the best fit line.

The ppcc method creates a single graph that shows the q-normal plot with a line showing the fit.

Value

The object x is returned invisibly.

---

ppcc.test *Test for Normality*

Description

Computes the probability plot correlation coefficient test for departures from normality.

Usage

```
ppcc.test(x)
```

Arguments

x	a vector of numeric values. Missing values are allowed, but are ignored in the calculation.

Value

An object of class "htest" having the following components:

statistic	the value of the test statistic.
p.value	the attained p-value for the test.
data.name	a character string describing the name of the data used in the test.
method	a description of the method.

Note

The PPCC test is attractive because it has a simple, graphical interpretation: it is a measure of the correlation in a Q-normal plot of the data. As such, it is related to the Shapiro-Wilk test (Shapiro and Wilk, 1965) for normality.

The distribution function of the test statistic is empirical. This application uses the "pocket calculator" approximation for computing the p-value of the observed statistic (Royston, 1992).

References

Filliben, 1975, The PPCC test for normality: Technometrics, v. 17, no. 1, p. 111–117.

Looney, S.W., and Gulledge, T.R., 1985, Use of the correlation coefficient with normal probability plots: The American Statistician, v. 39, p. 75–79.

Royston, J.P., 1992, A pocket-calculator algorithm for the Shapiro-Francia test of non-normality–an application to medicine: Statistics in Medicine, v. 12, p. 181–184.

Shapiro, S.S., and Wilk, M.B., 1965, An analysis of variance test for normality (complete samples): Biometrika, v. 52, p. 591–611.

See Also

shapiro.test

Examples

```
## These data should produce an attained p-value less than 0.001
set.seed(45)
ppcc.test.data <- rnorm(32)
qqnorm(ppcc.test.data)
abline(mean(ppcc.test.data), sd(ppcc.test.data))
ppcc.test(ppcc.test.data)
```

---

predict.move.1 *Predict Maintenance of Variance Extension, Type 1*

Description

Predicts new values from a maintenance of variance extension, type 1 (MOVE.1) model fit.

Usage

```
## S3 method for class 'move.1'
predict(object, newdata, type = c("response", "link"),
  var.fit = FALSE, ...)
```

Arguments

object	an object of class "move.1" on which to base the predicted values.
newdata	an optional data.frame in which to look for variables with which to predict. If omitted, then the fitted values are used.
type	the type of prediction ("response" or "link"). See **Details**.
var.fit	logical if TRUE, then compute the variance of the predicted values. If FALSE, then the varainces are not computed.
...	not used, required for method function.

Details

If type is "response," then the predicted values are back-transformed. Otherwise, the predicted values are computed directly from the model equation.

Value

If `var.fit` is `FALSE`, then a vector of predictions matching `newdata` or the model data. If `var.fit` is `TRUE`, then a data frame containing the columns:

`fit`	the predicted values
`var.fit`	the variance of the predicted values

References

Lorenz, D.L., 2015, smwrStats-an R package for analyzing hydrologic data, version 0.7.0: U.S. Geological Survey Open-File Report 2015-XXXX, XX p.

See Also

`move.1`, `jackknifeMove.1`

Examples

```
library(smwrData)
data(IonBalance)
# Build model for non missing Alkalinity
IB.move <- move.1(Anion_sum ~ Cation_sum, data=IonBalance, subset=abs(Pct_Diff) < 10)
print(IB.move)
# Predict Anion_sum for missing Alkalinity
predict(IB.move, IonBalance[1, ])
```

---

predict.move.2 *Predict Maintenance of Variance Extension, Type 2*

Description

Predicts new values from a maintenance of variance extension, type 2 (MOVE.2) model fit.

Usage

```
## S3 method for class 'move.2'
predict(object, newdata, type = c("response", "link"), ...)
```

Arguments

`object`	an object of class "move.2" on which to base the predicted values.
`newdata`	an optional data.frame in which to look for variables with which to predict. If omitted, then the fitted values are used.
`type`	the type of prediction ("response" or "link"). See **Details**.
`...`	not used, required for method function.

Details

If type is "response," then the predicted values are back-transformed. Otherwise, the predicted values are computed directly from the model equation.

Value

A vector of predictions matching newdata or the model data.

Note

If lag was set to a non-zero value in the call to move.2, then the explanatory variable is lagged only when predictiong values from the calibration data (newdata is not supplied.) This facilitates prediction of selected statistics at the response site rather than the complete record.

See Also

move.2

Examples

```
## Not run:
# See the vignette:
vignette("RecordExtension", package="smwrStats")

## End(Not run)
```

---

predict.senSlope *Predict Values.*

Description

Predicts new values from Sen slope (senSlope) model fit.

Usage

```
## S3 method for class 'senSlope'
predict(object, newdata, ...)
```

Arguments

object	an object of class "senSlope" on which to base the predicted values.
newdata	an optional data.frame in which to look for variables with which to predict. If omitted, then the fitted values are used.
...	not used, required for method function.

Value

A vector of predictions matching newdata or the model data.

See Also

senSlope

---

predictDuan *Bias Corrected Predictions*

Description

Predicts bias-corrected expected mean response values from a log-transformed regression model, using either the minimum variance unbiased estimate(MVUE), Duan's smoothing estimate, or Ferguson's maximum likelihood estimate.

Usage

```
predictDuan(object, newdata, back.trans = exp)

predictFerguson(object, newdata, Log10 = FALSE)

predictMVUE(object, newdata, Log10 = FALSE)
```

Arguments

object	an object of class "lm" on which to base the predicted values.
newdata	an optional data.frame in which to look for variables with which to predict. If omitted, then the fitted values are used.
back.trans	the back-transformation function. For common log transforms, use function(x) 10^x.
Log10	is the transform of the response variable the common log?

Value

A vector of predictions matching newdata or the model data.

References

Bradu, D. and Mundlak, Y., 1970, Estimation in the lognormal linear models: Journal of the American Statistical Association, v. 65, no. 329, p. 198–211.

Duan, N., 1983, Smearing estimate: a nonparametric retransformation method: Journal of the American Statistical Association, v. 78, p. 159–178.

Ferguson, R.I. 1986, River loads underestimated by rating curves: Water Resources Research, v. 22, p 74–76.

Helsel, D.R. and Hirsch, R.M., 2002, Statistical methods in water resources: U.S. Geological Survey Techniques of Water-Resources Investigations, book 4, chap. A3, 522 p.

See Also

lm

Examples

```
## Generate random log-normal data and build the regression model
set.seed(111)
XX.df <- data.frame(x=sort(runif(32, 1, 5)), y=rlnorm(32, seq(1,2, length.out=32)))
XX.lm <- lm(log(y) ~ x, data=XX.df)
## Compare the results for x=1:5
## The simple back-transformed estimates
exp(predict(XX.lm, newdata=data.frame(x=1:5)))
## The bias corrected estimates of the mean response
predictFerguson(XX.lm, newdata=data.frame(x=1:5))
predictDuan(XX.lm, newdata=data.frame(x=1:5))
predictMVUE(XX.lm, newdata=data.frame(x=1:5))
```

---

press *Prediction Error Sum of Squares*

Description

Computes the prediction error sum of squares statistic (PRESS) (Helsel and Hirsch, 2002) for a linear regression model.

Usage

```
press(model)
```

Arguments

model	an object of class "lm" or the output from lsfit.

Value

The prediction error sum of squares statistic.

References

Helsel, D.R. and Hirsch, R.M., 2002, Statistical methods in water resources: U.S. Geological Survey Techniques of Water-Resources Investigations, book 4, chap. A3, 522 p.

See Also

multReg, lm

print.ancovaReg *Print Objects*

Description

Prints the results of a analysis of covariance (ancovaReg).

Usage

```
## S3 method for class 'ancovaReg'
print(x, digits = 3, ...)
```

Arguments

x	an object of class "ancovaReg" from ancovaReg.
digits	the number of significant digits to print numeric data.
...	not used for method, required for other methods.

Value

The object x is returned invisibly.

Note

The printed output contains the ANOVA table for the orignal models, the regression summary for the final model, variance inflation factors for each explanatory variable in the final model, and selected test criteria with observations that exceed one of more of the criteria.

print.binaryreg *Print Objects*

Description

Prints the results of a logistic regression diagnotic analysis (binaryReg).

Usage

```
## S3 method for class 'binaryreg'
print(x, digits = 4, ...)
```

Arguments

x	an object of class "binaryreg" from binaryReg.
digits	the number of significant digits to print numeric data.
...	not used for method, required for other methods.

Details

The original regression model should be the output from `glm` with `binomial` as the value for the `family` argument.

Value

The object `x` is returned invisibly.comp2 Description of 'comp2'

Note

The printed output contains the original call; a summary of the deviance residuals; the coefficients with their estimates, standard errors, z scores, and attained p-values; a summary of the deviance comparison between the model and null model (no explanatory varaibles), also known as the G-squared or -2 log-likelihood; the response profile, which matches the value of the response variable to 0 or 1 and the number of observations in each group; the le Cessie-van Houwelingen and Hosmer-Lemeshow goodness of fit tests; three assessments of the predictinve power: the MaFadden R-squared and the adjusted R-squared, the classification table, the concordance index, and the area under the reciever operating characteristics curve; and selected test criteria with observations that exceed one of more of the criteria.

`print.cor.all` *Print Objects*

Description

Prints the results of a correlation test (`cor.all`).

Usage

```
## S3 method for class 'cor.all'
print(x, digits = 4, lower = TRUE, ...)
```

Arguments

`x`	an object of class "cor.all" from `cor.all`.
`digits`	the number of significant digits to print numeric data.
`lower`	logical, print only the lower triangular matrix? Otherwise, print the full, square matrix.
`...`	not used for method, required for other methods.

Value

The object `x` is returned invisibly.

Note

The printed output contains a description of the test; and 3 lines for each comparison that is printed: the correlation statistic, the attained p-value, and the number of pairs.

See Also

cor.all, printCor

---

print.lecessie *Print Objects*

Description

Prints the results of a le Cessie-van Houwelingen test (leCessie.test).

Usage

```
## S3 method for class 'lecessie'
print(x, digits = 4, ...)
```

Arguments

x	an object of class "lecessie" from leCessie.test.
digits	the number of significant digits to print numeric data.
...	not used for method, required for other methods.

Value

The object x is returned invisibly.

Note

The printed ouput is very similar to the printed ouput for class "htest," the output from a hypothesis test, but includes the additional output summarizing the distance between observations, which can be useful for comparing the reults with different bamdwidth settings.

---

print.MCT *Print Objects*

Description

Prints the results of a multiple comparison test (multicomp.test).

Usage

```
## S3 method for class 'MCT'
print(x, digits = 4, ...)
```

Arguments

x	an object of class "MCT" from multicomp.test.
digits	the number of significant digits to print numeric data.
...	not used for method, required for other methods.

Value

The object x is returned invisibly.

Note

The printed output contains a description of the test, critical values, the variables in the test, and two tables: the paired comparisons and associations among the groups. The table of the paried comparisons shows the groups in the comparison, the estimate of the difference between the group means, the standard error of the difference, lower and upper confidence intervals, and a flag that indicates if the confidence interval excludes 0, which indicates wheter the difference is significantly different from 0 at the user-specified value. The table of asociations shows the group, the mean value of the response, the number of observations in the group, and any number of columns names "A," "B," and so forth that represent possible associations of the groups whare an "X" is present in the group.

---

print.move.1 *Print Objects*

Description

Prints the results of a move.1 analysis (move.1).

Usage

```
## S3 method for class 'move.1'
print(x, digits = 4, ...)
```

Arguments

x	an object of class "move.1" from move.1.
digits	the number of significant digits to print numeric data.
...	additonal arguments for printing numeric values.

Value

The object x is returned invisibly.

Note

The printed output contains the call, the regression coefficients, and selected statistics of the variables.

print.move.2 *Print Objects*

Description

Prints the results of a MOVE.2 analysis (`move.2`).

Usage

```
## S3 method for class 'move.2'
print(x, digits = 4, ...)
```

Arguments

`x`	an object of class "move.2" from `move.2`.
`digits`	the number of significant digits to print numeric data.
`...`	additonal arguments for printing numeric values.

Value

The object `x` is returned invisibly.

Note

The printed output contains the call, the regression coefficients, and selected statistics of the variables.

print.multReg *Print Objects*

Description

Prints the results of a multiple regression diagnostic analysis (`multReg`).

Usage

```
## S3 method for class 'multReg'
print(x, digits = 3, ...)
```

Arguments

`x`	an object of class "multReg" from `multReg`.
`digits`	the number of significant digits to print numeric data.
`...`	not used for method, required for other methods.

Value

The object x is returned invisibly.

Note

The printed output contains the regression model call; a summary of the residuals; A table of the coefficients with their estiamtes, standard errors, t vlaues, and attained probability levels; the residual standard error; R-squared and F-statistical summaries; Model comparison statistics; if more than one explanatory variable a type II sum-of-squares analysis of variance table and variance inflation factors; and selected test criteria with observations that exceed one of more of the criteria.

---

print.optimBoxCox *Print Object*

Description

Prints the results of a multivariate unconditional Box-Cox transformation.

Usage

```
## S3 method for class 'optimBoxCox'
print(x, digits = 4, ...)
```

Arguments

x	an object of class "optimBoxCox" from optimBoxCox.
digits	the number of significant digits to print numeric data.
...	not used for method, required for other methods.

Value

The object x is returned invisibly.

Note

The printed output contains a table showing the power transformation values and their standard errors.

See Also

boxCox

print.roc *Print Object*

Description

Prints the results of a receiver operator characteristics (ROC) for a logistic regression model.

Usage

```
## S3 method for class 'roc'
print(x, digits = 3, ...)
```

Arguments

x	an object of class "roc" from roc.
digits	the number of significant digits to print numeric data.
...	not used for method, required for other methods.

Value

The object x is returned invisibly.

Note

The printed output contains the area under to ROC curve.

print.seasonalPeak *Print Objects*

Description

Prints the results of a seasonal peak analysis (seasonalPeak).

Usage

```
## S3 method for class 'seasonalPeak'
print(x, digits = 3, details = FALSE, ...)
```

Arguments

x	an object of class "seasonalPeak" from seasonalpeak.
digits	the number of significant digits to print numeric data.
details	logical, print the model details?
...	not used for method, required for other methods.

Value

The object x is returned invisibly.

Note

The printed output contains the default time-of-peak value and potential alternate values for unconfirmed peaks and the number of peaks, timing of the primary peak, and optionally the model information for confirmed peaks.

`print.senSlope` *Print Objects*

Description

Prints the results of a Sen slope analysis (`senSlope`).

Usage

```
## S3 method for class 'senSlope'
print(x, digits = 4, ...)
```

Arguments

`x`	an object of class "senSlope" from `senSlope`.
`digits`	the number of significant digits to print numeric data.
`...`	additonal arguments for printing numeric values.

Value

The object x is returned invisibly.

Note

The printed output contains the call, the smalles and largest residuals, the regression coefficients, and the confidence limits of the Sen slope.

printCor *Print Objects*

Description

Prints the results of a correlation test (`cor.all`) or correlation matrix in a compact form indicating positive or negative correlation.

Usage

```
printCor(x, criterion = 0.75)
```

Arguments

`x`	an object of class "cor.all" from `cor.all` or the output from `cor`.
`criterion`	a numeric value indicating the test criterion for showing a value.

Value

The object `x` is returned invisibly.

Note

The printed output is a compressed table showing "+" where the value in `x` is greater than `criterion`, "-" where the value in `x` is less than -`criterion`, "\" on the diagonal, if `x` is symmetric, and "." where `x` has a missing value, and " " otherwise.

See Also

`cor.all`

Examples

```
## Not run:
library(smwrData)
data(TNLoads)
printCor(cor(TNLoads, method="spearman"), .5)

## End(Not run)
```

qgrubbs *The Grubbs Distribution*

Description

Computes the cumulative probability and quantiles for the Grubbs distribution describing an outlier in a sample from Normal distribution.

Usage

```
qgrubbs(alpha, N)

pgrubbs(G, N)
```

Arguments

alpha	the significance level.
N	the number of values in the sample.
G	the maximum or minimum scaled difference from the mean.

Value

The function pgrubbs returns the attained p-value, not the cumulative distribution, for the maximum scaled distance from the mean. the funciton qgrubbs returns the one-sided critical value for the maximum scaled difference from the mean.

References

Grubbs, F., 1969, Procedures for Detecting Outlying Observations in Samples, Technometrics, v. 11, no. 1, pp. 1-21.

See Also

grubbs.test

Examples

```
# The difference is due to rounding errors
pgrubbs(c(.9, .95, .99), 32)
qgrubbs(c(5.905348, 5.483234, 5.159097), 32)
```

qtiles.CI *Quantiles with Confidence Limits*

Description

Computes sample quantiles and confidence limts for specified probabilities.

Usage

```
qtiles.CI(x, probs = 0.5, CI = 0.9, bound = c("two.sided", "upper",
  "lower"), na.rm = TRUE)
```

Arguments

x	numeric vector to compute the sample quantiles.
probs	numeric vector of desired probabilities with values between 0 and 1.
CI	the minimum desired confidence interval for each level specifed in probs.
bound	a character string indicating the desired bounds, "two.sided" means the two-sided interval, "upper" means the upper bound of the interval, and "lower" means the lower bound of the interval. Only a single character is needed. The lower confidence limit is -Inf when bound is "upper" and the upper confidence limit is Inf when bound is "lower."
na.rm	logical; if TRUE, then missing values are removed before computation.

Value

A matrix of sample quantiles, the lower confidence limit, the upper confidence limit, and the probability represented by the confidence interval corresponding to the probs levels in the sorted x data.

References

Helsel, D.R. and Hirsch, R.M., 2002, Statistical methods in water resources: U.S. Geological Survey Techniques of Water-Resources Investigations, book 4, chap. A3, 522 p.

See Also

quantile

Examples

```
## Generate a random sample
set.seed(222)
XX.rn <- rexp(32)
qtiles.CI(XX.rn, probs=c(.25, .5, .75))
```

quantile.numeric *Sample Quantiles*

Description

Computes sample quantiles corresponding to the given probabilities: method for "numeric" data. The smallest observation corresponds to a probability of 0 and the largest to a probability of 1. This method function is a simple wrapper for the default function that sets the default type to 2 for numeric data.

Usage

```
## S3 method for class 'numeric'
quantile(x, probs = seq(0, 1, 0.25), na.rm = FALSE,
  names = TRUE, type = 2, ...)
```

Arguments

x	numeric vector whose sample quantiles are wanted.
probs	numeric vector of probabilities with values in the range from 0 to 1.
na.rm	remove missing values NAs before computation?
names	include names of the probabilities, expressed as percentages?
type	an interger between 1 and 9 that selects the method for computing the quantile. See **Note**.
...	further arguments passed to or from other methods.

Value

An optionally named vector corresponding to the quantiles of x for the selected probabilities.

Note

Helsel and Hirsh (2002) define the 75th percentile as "a value which exceeds no more than 75 percent of the data and is exceeded by no more than 25 percent of the data." This rule can be easily extended to other percentiles. The selection of type equal to 2 ensures that this rule is met for all data. The rule stated by Helsel and Hirsch is very useful for an emprical description of the data, but Hyndman and Fan (1996) describe the slection of type for other applications.

References

Helsel, D.R. and Hirsch, R.M., 2002, Statistical methods in water resources: U.S. Geological Survey Techniques of Water-Resources Investigations, book 4, chap. A3, 522 p.

Hyndman, R.J. and Fan, Y. 1996, Sample quantiles in statistical packages: American Statistician, v. 50, p. 361-365.

See Also

quantile.default

Examples

```
# The default value for type (7) will compute a value that is exceeded by 4 values
#  for a sample of size 14
quantile(seq(14), type=7)
# But 4/14 is greater than 25 percent. But setting type to 2 will result in
#  only 3 values that are larger than the computed 75th percentile.
quantile(seq(14))
```

---

regken *Trend Test*

Description

Computes the regional Kendall trend test with Sen slope estimator.

Usage

```
regken(series, correct = nrow(series) > 9)
```

Arguments

series	a numeric matrix with rows representing the annual observations and columns representing the sites. Missing values are permitted.
correct	logical, if TRUE, then apply the correction for cross correlation among sites. if FALSE, then do not apply the correction.

Value

An object of class "htest" also inhereting class "seaken" containing the following components:

method	a description of the method.
statistic	the value of Kendall's tau.
p.value	the p-value. See **Note**.
p.value.raw	the p-value computed without correction for cross correlation. See **Note**.
p.value.corrected	the p-value computed with correction for cross correlation. See **Note**.
estimate	a named vector containing the Sen estimate of the slope in units per year, the median value of the data, and the median value of time.
data.name	a string containing the actual name of the input series with the number of years and sites
alternative	a character string describing alternative to the test ("two.sided").

null.value	the value for the hypothesized slope (0).
nyears	the number of years.
nseasons	the number of sites
series	the data that was analyzed.
seasonnames	the names of the sites

Note

The value of p.value is p.value.raw if there are fewer than 10 years of data and is p.value.corrected otherwise.

The regional Kendall is described in Douglas and others (2000) and Helsel and others (2006). The adjustment for spatial correlation used in regken is based on equation 13 in Douglas and others (2000) and uses the Spearman correlation to account for monotonic correlations.

References

Douglas, E.M., Vogel, R.M., and Kroll, C.N., 2000, Trends in floods and low flows in the United States: impact of spatial correlation: Journal of Hydrology, v. 240, p. 90–105.

Helsel, D.R., Mueller, D.K., and Slack, J.R., 2006, Computer program for the Kendall family of trend tests: U.S. Geological Survey Scientific Investigations Report 2005–5275, 4 p.

See Also

seaken

Examples

```
# Need example?
```

---

rmse *Root-Mean-Squared and Relative Differences*

Description

Computes the root-mean-squared error (RMSE) of the difference between observed values and the predicted values or the RMSE or relative percent differences (RPD) between samples and duplicates.

Usage

```
rmse(x, ...)

## Default S3 method:
rmse(x, y, ...)

## S3 method for class 'lm'
```

```
rmse(x, ...)

rpd(x, y, plotit = FALSE)
```

Arguments

x	either a random vector an object for which a method exists.
y	duplicate samples paired with x.
plotit	logical, if TRUE, then create a Bland-Altman mean-difference plot (banld and Altman, 1986); otherwise no plot is created.
...	arguments to be passed to or from methods.

Value

For the rmse functions, a single value representing the estimated RMSE. For rpd, the relative percent differences for each paired sample and duplicate.

Note

The definition for the RMSE of paired water-quality duplicates is

$$RMSE = \sqrt{\frac{\sum (x_i - y_i)^2}{2n}}$$

The definition for RPD for paired water-quality duplicates is

$$RPD = abs(x - y)/(x + y)/2 * 100$$

Other disciplines may use different equations.

References

Bland J.M. and Altman D.G., 1986 Statistical methods for assessing agreement between two methods of clinical measurement: Lancet, i, p. 307–310.

Clesceri, L.S., Greenberg, A.E., and Eaton, A.D., 1998, Standard methods for the examination of water and wastewater, 20th edition: Baltimore, Md, United Book Press, Inc., 1162 p.

Harvey, D., undated, Analytical chemistry 2.0: Analytical Sciences Digital Library: online at URL: http://www.asdlib.org/onlineArticles/ecourseware/Analytical

Helsel, D.R., and Hirsch, R.M., 2002, Statistical methods in water resources: U.S. Geological Survey Techniques of Water-Resources Investigations, book 4, chap. A3, 522 p.

Examples

```
# Example 15.2 from Harvey.
dupX1 <- c(160, 196, 207, 185, 172, 133)
dupX2 <- c(147, 202, 196, 193, 188, 119)
rmse(dupX1, dupX2)
rpd(dupX1, dupX2)
```

roc *Receiver Operator Characteristics for Logistic Regression*

Description

Computes the receiver operator characteristics (ROC) for a logistic regression model.

Usage

```
roc(object)
```

Arguments

object	an object of class "glm."

Value

An object of class "roc" haveing these components:

c.val	the area under the ROC curve
table	a 3-column matrix of the fitted values, sensitivity and specificity

Note

Objects of class "roc" have `print` and `plot` methods.

References

Gonen, M., 2006, Receiver Operating Characteristics (ROC) Curves: SAS Users Group International Conference, Paper 210-31 18 p., available at http://www2.sas.com/proceedings/sugi31/210-31.pdf, last accessed Octocer 26, 2011.

seaken *Trend Test*

Description

Computes the seasonal Kendall trend test with Sen slope estimator.

Usage

```
seaken(series, nseas = 12)
```

Arguments

series	a regularly spaced numeric vector to test for trend. Missing values are permitted.
nseas	the number of seasons per year. Must not exceed 52. Can also be a character vector of the names of the seasons. The length of the character vector determines the number of seasons.

Value

An object of class "htest" also inhereting class "seaken" containing the following components:

method	a description of the method.
statistic	the value of Kendall's tau.
p.value	the p-value. See **Note**.
p.value.raw	the p-value computed without correction for serial correlation. See **Note**.
p.value.corrected	the p-value computed with correction for serial correlation. See **Note**.
estimate	a named vector containing the Sen estimate of the slope in units per year, the median value of the data, and the median value of time.
data.name	a string containing the actual name of the input series with the number of years and seasons.
alternative	a character string describing alternative to the test ("two.sided").
null.value	the value for the hypothesized slope (0).
nyears	the number of years.
nseasons	the number of seasons.
series	the data that was analyzed.
seasonnames	the names of the seasons.

Note

The value of `p.value` is `p.value.raw` if there are fewer than 10 years of data and is `p.value.corrected` otherwise.

References

Hirsch, R.M., Alexander, R.B., and Smith, R.A., 1991, Selection of methods for the detection and estimation of trends in water quality: Water Resources Research, v. 27, p. 803–813.

Hirsch, R.M., Slack, J.R., and Smith, R.A., 1982, Techniques of trend analysis for monthly water quality data: Water Resources Research, v. 18, p. 107–121.

Hirsch, R.M., and Slack, J.R., 1984, A nonparametric trend test for seasonal data with serial dependence: Water Resources Research, v. 20, p. 727–732.

Kendall, M.G., 1938, A new measure of rank correlation: Biometrika v. 30, p. 81–89.

Kendall, M.G., 1976, Rank correlation methods (4th ed.): London, Griffin, 202 p.

Sen, P.K., 1968, Estimates of regression coefficient based on Kendall's tau: Journal of the American Statisical Association, v. 63, p. 1379–1389.

See Also

kensen.test, regularSeries

Examples

```
## Not run:
library(smwrData)
library(smwrBase)
data(KlamathTP)
RegTP <- with(KlamathTP, regularSeries(TP_ss, sample_dt))
# The warning generated is expected and acceptable for these data
seaken(RegTP$Value, 12)
# Manaus river data is in package boot
library(boot)
data(manaus)
manaus.sk <- seaken(manaus, 12)
print(manaus.sk)
# Note for these data the large difference between the raw and corrected p-values.
#  p-value (raw) is << 0.001
manaus.sk$p.value.raw
#  p-value (with correlation correction) is = 0.10
manaus.sk$p.value.corrected
# Hence, it may be concluded that these particular data show substantial serial correlation
#  as seen with see with acf(manaus).

## End(Not run)
```

---

seasonalPeak *Seasonal Peak Timing*

Description

Computes the timing of the seasonal peak value. The timing of the seasonal peak is needed for the seasonalWave model (Vecchia and others, 2008).

Usage

```
seasonalPeak(x, y)
```

Arguments

x	a vector of decimal time representing dates and times. Missing values are permitted and are removed before analysis.
y	a vector of the data for which the peak is needed. Missing values are permitted and are removed before analysis.

Details

The timing of the peak of the data is computed by identifying the largest value produced by smoothing that data with supsmu. The remaining data in the attributes are used by using the seasonalPeak method of confirm.

Value

An object of class seasonalPeak. The unconfirmed object is a single value that represents the estimate of the timing of the peak and five additional attributes.

Data: a list of the x and y values where x is the fractional part of the original decimal time data. Missing values have been removed.
Smooth: a list of the x and y smoothed values.
Points: a list of 361 evenly spaced xout and yout values.
Extra: pointers to all the peaks in Points.
Confirmed: logical indicating that the object has not been confirmed.

Note

The generic functions print and confirm have methods for object of class seasonalPeak.

References

Vecchia, A.V., Martin, J.D., and Gilliom, R.J., 2008, Modeling variability and trends in pesticide concentrations in streams: Journal of the American Water Resources Association, v. 44, no. 5, p. 1308-1324

See Also

seasonalWave, confirm.seasonalPeak, supsmu, print.seasonalPeak

Examples

```
library(smwrData)
data(QW05078470)
with(QW05078470, seasonalPeak(dectime(DATES), P00665))
## Should be:
# Default value: 0.499
# Alternate values: 0.497
```

`seasonalWave` *Seasonal Wave*

Description

Computes a seasonal wave model to describe the variation of a constituent over the course of a year. This model is particularly well suited to describe the concentration of pesticides.

Usage

```
seasonalWave(x, cmax, loading, hlife, second.peak = NULL)
```

Arguments

`x`	a vector of decimal time representing dates. Missing values (`NA`s) are permitted.
`cmax`	the time of the greatest peak value, expressed as a fraction of the year.
`loading`	the number of months of contituent loading for the primary peak.
`hlife`	the half life of the decay rate, expressed in units of months. This should be in the range of 1 though 4.
`second.peak`	a list of the parameters for the second peak. See **Details**.

Details

The seasonal wave model $W(t)$ is expressed as a differential equation

$$\frac{d}{dt}W(t) = \lambda(t) - \phi W(t)[0 \leq t \leq 1,$$
$$W(0) = W(1)]$$
$$\lambda(t) = \sum_{k=1}^{1} 2\omega_k I(\frac{k-1}{12} \leq t <$$
$$\frac{k}{12})$$

where ϕ is a specified constant, which represents the rate of removal or chemical decay; $[\omega_k, k = 1, 2, ..., 12]$ are specified nonnegative constants, which represent monthly input amounts; $I(.)$ is the indicator function with $I(.) = 1$ if t lies in the given interval and $I(.) = 0$ otherwise; and $\lambda(t)$ is the input amount at time t.

The value of hlife is used to define the decay rate ϕ in the differential equation. The value of ϕ is 12/`hlife` and $W(t)$ decays at a a rate of $exp(-\phi/12)$ per month.

The list for `second.peak` must contain these components:
la, the lag from the primary peak to the second peak, in months. This is always the time from the primary peak to the second peak, even if the second peak occurs earlier in the year.
lo, the loading duration in months, this value must be leas than `la`. w, the load scaling factor relative to the primary peak. Must be greater than 0 and less than or equal to 1. For practical pruposes, it should be greater than 0.5.

Value

A vector expressing the expected variation of concentration for each value in x; the values are scaled to a range of -0.5 to 0.5.

Author(s)

Dave Lorenz, original coding by Aldo Vecchia.

References

Vecchia, A.V., Martin, J.D., and Gilliom, R.J., 2008, Modeling variability and trends in pesticide concentrations in streams: Journal of the American Water Resources Association, v. 44, no. 5, p. 1308-1324.

See Also

seasonalPeak, confirm.seasonalPeak

Examples

```
## Not run:
# Selected single peak models
# 1 month loading, 1 month half-life
curve(seasonalWave(x, 3/12, 1, 1), 0, 1, n=361, xlab='fraction of year', ylab="W")
# 1 month loading, 3 month half-life
curve(seasonalWave(x, 3/12, 1, 3), 0, 1, n=361, add=TRUE, col="blue")
# 3 month loading, 2 month half-life
curve(seasonalWave(x, 3/12, 3, 2), 0, 1, n=361, add=TRUE, col="green")
# Add a second peak model
curve(seasonalWave(x, 3/12, 3, 2, second.peak=list(la=6, lo=2, w=.75) ), 0, 1,
  n=361, add=TRUE, col="red")

## End(Not run)
```

---

seasonalWave.fit *Compute Seasonal Wave Model*

Description

This is a support function for seasonalWave model (Vecchia and others, 2008).

Usage

```
seasonalWave.fit(cmax, wtx, pkt, phi)
```

Arguments

cmax	the time of the greatest peak value, expressed as a fraction of the year.
wtx	a vector of 12 monthly application rates.
pkt	the numeric month of peak expected from the data in wtx, expressed as a fraction of the year.
phi	the recession rate, in 1/years.

Value

A vector of 361 values describing the annual seasonal wave.

Note

This function is support for the seasonalWave function and is not intended to be called by the user.

Author(s)

Dave Lorenz, original coding by Aldo Vecchia.

References

Vecchia, A.V., Martin, J.D., and Gilliom, R.J., 2008, Modeling variability and trends in pesticide concentrations in streams: Journal of the American Water Resources Association, v. 44, no. 5, p. 1308-1324.

See Also

seasonalWave

---

seasonalWave.wt *Compute Seasonal Wave Model*

Description

This is a support function for seasonalWave model (Vecchia and others, 2008).

Usage

```
seasonalWave.wt(loading, second.peak)
```

Arguments

loading	the number of months of primary loading for the seasonal wave model.
second.peak	a list describing the loading characteristics of the second peak. See **Details**. If NULL, then no second peak is added to the seasonal wave model.

Details

If second.peak is supplied, then it must be a list with these components:
la, the lag (number of months) between the primary and second peak;
lo, the loading duration (number of months) for the second peak (must be less than la;
w, the weigthing factor for the second peak loading, relative to the primary peak. Must be greater than 0 and less than or equal to 1.

Value

The weighting vector for the seasaon wave model. A vector of length 12 that represents the relative loading to the system.

Note

This function is support for the seasonalWave function and is not intended to be called by the user.

References

Vecchia, A.V., Martin, J.D., and Gilliom, R.J., 2008, Modeling variability and trends in pesticide concentrations in streams: Journal of the American Water Resources Association, v. 44, no. 5, p. 1308-1324.

See Also

seasonalWave

---

selBestWave *Select the "Best" Seasonal Wave*

Description

Selects the "best" parameters for a seasonal wave fit.

Usage

```
selBestWave(formula, data, dec.time, wave.list, exhaustive = FALSE,
  Regression = lm, Test = AIC, ...)
```

Arguments

formula	the formula describing the model without a seasonalWave term.
data	the data.frame that contians the veriable specified in the formula.
dec.time	a character string of the name of the column in data in decimal time format.
wave.list	an object of class "seasonalPeak" (confirmed) describing the timing of the peak and potential candidate models.
exhaustive	logical; if TRUE, then do a fairly complete search for the timing of the peak, otherwise accept the timing specified in wave.list.

Regression	the regression function.
Test	the function to perform the comparison test among all of the candidate models.
...	any additional arguments to Regression.

Details

For logistic regression, use Regression=glm, Test=deviance, family=binomial(link="logit").

Value

A 4- or 7-column matrix of the "best" models. The columns are the timing of the peak (Cmax), the primary peak loading (Loading), the half life (Hlife), and the test score (Test). If the model is a two-peak model, then additional columns (la, lo, and w) describeing the second peak are included.

References

Vecchia, A.V., Martin, J.D., and Gilliom, R.J., 2008, Modeling variability and trends in pesticide concentrations in streams: Journal of the American Water Resources Association, v. 44, no. 5, p.1308–1324.

Examples

```
## See the SeasonalWave demo
```

---

senSlope *Compute the Sen Slope*

Description

Computes the Sen slope with confidence interval and an intercept for paired data.

Usage

```
senSlope(formula, data, subset, na.action, intercept = "Ac", CI = 0.95)
```

Arguments

formula	a model formula with exactly one explanatory variable.
data	the data.
subset	any descriptiopn to subset data.
na.action	the function to handle missing values.
intercept	a character string indicating the method to compute the intercept. See **Details**.
CI	the desired confidence interval for the slope.

Details

The argument `intercept` may be either "Ac" or "A1m." If it is "Ac," then the intercept is computed from the median of `y` and `x`, also known as the Conover method. If it is "A1m," then the intercept is chosen so that the median of the residuals is zero.

Value

An object of class "senSlope" with these components:

`call`	the matched call.
`coefficients`	the intercept ans Sen slope.
`slope.CI`	the lower and upper confidence limits of the Sen slope.
`residuals`	the residuals of the regression.
`fitted.values`	the fitted values.
`na.action`	information about any missing values.
`x`	the explanatory variable.
`y`	the response variable.
`var.names`	the response and explanatory variable names.
`model`	the model frame.

References

Dietz, E.J., 1989, Teaching regression in a nonparametric statistics course: The American Statstician, v. 43, p. 35–40

Helsel, D.R., and Hirsch, R.M., 2002, Statistical methods in water resources: U.S. Geological Survey Techniques of Water-Resources Investigations, book 4, chap. A3, 522 p.

Sen, P.K., 1968, Estimates of the regression coefficient based on Kendall's tau: Journal of the American Statistical Association, v. 63 p. 1379–1389

See Also

`kensen.test`, `serial.test`

Examples

```
## Not run:
library(smwrData)
data(SaddlePeaks)
senSlope(Flow ~ Year, data=SaddlePeaks)

## End(Not run)
```

`serial.test` *Test for Serial Correlation*

Description

Performs either of two nonparametric tests (Wilcoxon Rank-Sum test or runs test) for serial correlation. Both tests compute Z, which is the product of sequential values of `x` after removing any trend and normalizing so that the median of Z values is 0.

Usage

```
serial.test(x, method = "wilcoxon")
```

Arguments

`x`	the numeric vector of observations. Missing values (NAs) are not allowed.
`method`	the string "wilcoxon" or "runs," depending on which test should be used. Only the first character is necessary. See **Details**.

Details

If the method is "wilcoxon," then the Wilcoxon Rank-Sum test (Wilcoxon, 1945) is performed to compare the distribution of positive Z values to the distribution of negative Z values. Excessive numbers of positive Z values suggests positive serial correlation and excessive negative Z values suggests negative serial correlation.

If the method is "runs," then the runs test (Wald and Wolfowitz, 1940) is performed on sequences of positive and negative values of Z. If there is a larger than expected number of runs, then that suggest negative serial correlation and if fewer than the expected number of runs, then positive serial correlation.

Value

An object of class htest containing the following components:

`statistic`	the score for either test.
`p.value`	the two-sided p-value of the statistic.
`Z`	the raw data of the analysis.
`alternative`	a string "two.sided" indicating the hypothesis.
`method`	a description of the method.
`data.name`	a string containing the actual name of the input data.

Note

The null hypothesis is that the data are uncorrelated.

References

Dufour, J.M., 1981, Rank test for serial dependence: Journal of the Time Series Analysis, v. 2, no. 3, p. 117–128.

Wald, A., and Wolfowitz, J., 1940, On a test whether two samples are from the same population: Annals of Mathematical Statistics, v. 11, p. 147–162.

Wilcoxon, F., 1945, Individual comparisons by ranking methods: Biometrics, v. 1, p. 80–83.

`seriesPlot.seaken` *Series Plot*

Description

Produces a plot of time-series data by season that shows seasonal and overall trends: method for "seaken" object.

Usage

```
## S3 method for class 'seaken'
seriesPlot(x, SeasonLine = list(name = "", what = "vertical",
  color = "black"), SeasonPoint = list(name = "", what = "points", symbol =
  "circle", filled = TRUE, size = 0.09, color = "black"), yaxis.log = FALSE,
  yaxis.range = c(NA, NA), ylabels = 7, xlabels = x$seasonames,
  xtitle = "", ytitle = deparse(substitute(x)), caption = "",
  margin = c(NA, NA, NA, NA), SeasonTrend = list(name = "", what = "lines",
  color = "brown"), TrendLine = list(name = "", what = "lines", color =
  "blue"), ...)
```

Arguments

`x`	data that can be treated as a regularly-spaced time series.
`SeasonLine`	control parameters of the lines in the plot. See **Details**.
`SeasonPoint`	control parameters of the points in the plot. See **Details**.
`yaxis.log`	log-transform the y axis?
`yaxis.range`	set the range of the y axis.
`ylabels`	set the y-axis labels. See `linearPretty` for details.
`xlabels`	set the x-axis labels, may be a single numeric value indicating the number of season in `x`. See `namePretty` for details.
`xtitle`	the x-axis title.
`ytitle`	the y-axis title.
`caption`	the figure caption.
`margin`	the parameters of the margin of the plot area.
`SeasonTrend`	control parameters of the trend line for each season. See **Details**.
`TrendLine`	control parameters of the overall trend line compute by `seaken`. See **Details**.
`...`	any additional arguments required for specific methods.

Details

The argument what for SeasonLine must be either "lines" or "vertical." See monthplot for more information.
The argument what for either SeasonPoint, SeasonTrend, or trendLine may be set to "none" to suppress drawing of that feature.

Value

Information about the graph.

See Also

monthplot, seasonPlot, seaken

---

skew *Skewness*

Description

Computes the skewness statistic.

Usage

```
skew(x, na.rm = TRUE, method = "fisher")
```

Arguments

x	any numeric vector.
na.rm	logical; if TRUE, then remove missing value before computation.
method	the method to use for computing the skew, must be "fisher" or "moments."

Value

a single value representing the skewness of the data in x.

References

Helsel, D.R. and Hirsch, R.M., 2002, Statistical methods in water resources: U.S. Geological Survey Techniques of Water-Resources Investigations, book 4, chap. A3, 522 p.

Examples

```
skew(c(1.0, 1.2, 1.5, 1.9, 2.5))
```

summary.cor.all *Summarize Correlations*

Description

Summarizes the correlations: method for "cor.all" object.

Usage

```
## S3 method for class 'cor.all'
summary(object, p.adjust = "holm", variable = NULL, ...)
```

Arguments

object	an object created by the cor.all function.
p.adjust	a character string describing the method to use to adjust the p-values to account for m ultiple comparisons. See p.adjust for the options and details.
variable	if the name of a variable, then summarize only those correlations of this variable with all others, otherwise summarize all combinations.
...	further arguments passed to or from other methods.

Value

A data frame containing columns of the paired variables, the correlation, the adjusted p-value, and the number of observations in the correlation.

See Also

cor.all

Examples

```
library(smwrData)
data(TNLoads)
# Extract only the correlations with the log of total nitrogen
summary(cor.all(TNLoads[, 1:5]), variable="LOGTN")
```

`sumStats` *Compute Summary Statistics*

Description

Creates a dataset of specified summary statistics from a collection of data.

Usage

```
sumStats(..., group = NULL, Num = "Num", Stats = list(Mean = mean, StdDev
  = sd), Probs = (0:4)/4, na.rm = TRUE)
```

Arguments

`group`	list whose components are interpreted as categories, each of the same length as the objects in ... The e lements of the categories define the position in a multi-way array corresponding to each observation. Missing values (NAs) are allowed. The names of group are used as the names of the columns in the output dataset. If a vector is given, it will be treated as a list with one component. The default is NULL, which indicates no grouping variable.
`Num`	a character string indicating the name of the column that contains the number of nonmissing observations.
`Stats`	a tagged list. An element in the list should have the name of the target variable in the output data set, a nd it should be a function that accepts the na.rm argument and computes a single value. See **Notes** for commonly used functions. The default is the target columns Mean and StdDev, which are computing using functions mean and stdev.
`Probs`	vector of desired probability levels. Values must be between 0 and 1. Minimum returned for probs=0 and maximum returned for probs=1. Default is c(0.0, 0.25, 0.50, 0.75, 1.0).
`na.rm`	logical; if TRUE, then missing values are removed from each column in ... before computing the statistics
`...`	any number of arguments, which can be of many different forms: a dataset, selected columns of a dataset, vectors, and matrices. If a dataset is supplied, then nonnumeric columns are removed before computing statistics.

Value

A data frame containing columns identifying each variable in ..., any grouping variables, and the requested statistics as named in the call.

Note

Commonly used functions referenced in the Stats arguments include mean, sd, skew and var.

The statistics requested by the Probs argument are computed by the quantile function using `type=2`.

References

Helsel, D.R. and Hirsch, R.M., 2002, Statistical methods in water resources: U.S. Geological Survey Techniques of Water-Resources Investigations, book 4, chap. A3, 522 p.

See Also

mean, sd, skew, var, quantile,

Examples

```
## Generate a random sample
set.seed(222)
XX.rn <- rexp(32)
sumStats(XX.rn)
```

---

timeWeightedMean *Time-Weighted Mean*

Description

Computes the mean weighted by the duration between values.

Usage

```
timeWeightedMean(x, time, na.rm = TRUE, fill = TRUE, by.period = FALSE,
  excessive = 2.5)
```

Arguments

x	a sequnece of numeric values whose mean is to be computed.
time	the times associated with x. See **Details**.
na.rm	remove missing values before computing the mean?
fill	expand the weights for the first and last observation to the end of the period? Otherwise, the weight is based on the distance to the second or next-to-last observation.
by.period	computes wights by the periods defined by time.
excessive	a warning is issued if the largest weight for any observation exceeds excessive times the average weight.

Details

The values of time are expected to be in decimal format, where the integer part indicates the period and the fractional part linearly distributed through the period. For example, the year 2000 begins at 2,000.0 and July 2, 2000 is 2,000.5. The function dectime can be used to convert Date or POSIX to decimal format. If time is of class "Date," or any "POSIX" classs, it is converted using dectime. If na.rm is FALSE and byeriod is TRUE, then missing values are returned only for periods that contain a missing value (NA).

Value

If by.period is TRUE, then a vector with one entry per period, otherwise the mean for the entire data set.

References

Crawford,C.G., 2004, Sampling strategies for estimating acute and chronic exposures of pesticides in streams: Journal of the American Water Resources Association, v. 40, n. 2, p 485-502.

See Also

weighted.mean, dectime

Examples

```
## Not run:
library(smwrData)
data(QW05078470)
with(QW05078470, timeWeightedMean(P00665, dectime(DATES)))

## End(Not run)
```

---

trends *Basis for Piecewise Linear Trends*

Description

Generates a basis matrix for piecewise linear modeling of trends.

Usage

```
trends(x, breaks, boundary.breaks = c(FALSE, FALSE), steps)
```

Arguments

x	a vector of dates/times, assumed to be in dectime format. Missing values are permitted and result in corresponding missing values in the output.
breaks	a vector of breakpoints in the linear trends.
boundary.breaks	a logical vector of length 2, indicating whether the breaks include the range of the data. If the first is TRUE, then the first break denotes the beginning of a trend. If the last is TRUE, then the last break denotes the end of a trend. Otherwise, the trends begin at the floor of the first value of x and end at the ceiling of the last value of x.
steps	a vector indicating any step trends.

Value

A matrix of dimension length of x by number of linear trend and step trends. The breaks are included as an attribute.

Note

Each trend is 0 prior to the break, and then increases at the rate of 1 per unit and maintain that maximum value after the break. The regression coefficient then reprents the trend as a rate. A step trend is 0 before the step and 1 after it.

See Also

floor, ceiling, curvi

Examples

```
# model two piecewise linear trends from 2000 to 2004, with a break at 2001 and 2003
trends(2000 + seq(0,20)/5, breaks=c(2001, 2003))
```

---

vif *Variance Inflation Factors*

Description

Computes the variance inflation factor (Helsel and Hirsch, 2002) for each variable in a linear regression fit.

Usage

```
vif(model, ...)

## S3 method for class 'lm'
vif(model, ...)
```

Arguments

model	an object of class "lm." There is no default method for vif.
...	further arguments passed to or from other methods.

Value

A named numeric vector containing the variance inflation factors for each variable.

References

Helsel, D.R. and Hirsch, R.M., 2002, Statistical methods in water resources: U.S. Geological Survey Techniques of Water-Resources Investigations, book 4, chap. A3, 522 p.

Index

∗ **distribution**
qgrubbs, 48
∗ **hplot**
plot.ancovaReg, 29
plot.htest, 32
seriesPlot.seaken, 65
∗ **htest**
cor.all, 11
grubbs.test, 14
hosmerLemeshow.test, 15
kensen.test, 17
leCessie.test, 19
multicomp.test, 25
ppcc.test, 33
regken, 51
seaken, 54
serial.test, 64
summary.cor.all, 67
∗ **manip**
confirm, 9
confirm.seasonalPeak, 10
percentile, 28
seasonalPeak, 56
seasonalWave, 58
seasonalWave.fit, 59
seasonalWave.wt, 60
∗ **math**
percentile, 28
∗ **models**
allReg, 5
ancovaReg, 7
binaryReg, 8
jackknifeMove.1, 16
move.1, 21
move.2, 23
multReg, 26
predict.move.1, 34
predict.move.2, 35
predict.senSlope, 36
predictDuan, 37
selBestWave, 61
∗ **model**
curvi, 13
trends, 70
vif, 71
∗ **nonparametric**
cor.all, 11
multicomp.test, 25
senSlope, 62
serial.test, 64
∗ **package**
smwrStats-package, 3
∗ **regression**
allReg, 5
ancovaReg, 7
binaryReg, 8
jackknifeMove.1, 16
move.1, 21
move.2, 23
multReg, 26
predict.move.1, 34
predict.move.2, 35
predict.senSlope, 36
predictDuan, 37
press, 38
roc, 54
selBestWave, 61
senSlope, 62
∗ **robust**
senSlope, 62
∗ **univariate**
skew, 66
∗ **univar**
percentile, 28
qtiles.CI, 49
quantile.numeric, 50
rmse, 52
sumStats, 68

timeWeightedMean, 69

allReg, 5
ancovaReg, 7, *32*

binaryReg, 8, *16*, *20*, *32*
boxCox, *28*, *44*

ceiling, *71*
confirm, 9
confirm.seasonalPeak, 10, *57*, *59*
cor.all, 11, *32*, *41*, *47*, *67*
cor.test, *13*
curvi, 13, *71*

dectime, *19*, *70*

ecdf, *29*

floor, *71*

Grubbs *(*qgrubbs*)*, 48
grubbs.test, 14, *48*

hosmerLemeshow.test, *9*, 15

jackknifeMove.1, 16, *35*

kensen.test, 17, *56*, *63*

leCessie.test, *9*, 19, *32*
linearPretty, *65*
lm, *8*, *27*, *38*
loess.smooth, *32*

makepredictcall.trends, 20
mean, *69*
model.frame.default, *20*
monthplot, *66*
move.1, *17*, 21, *32*, *35*
move.2, 23, *32*, *36*
multicomp.test, 25
multReg, *8*, 26, *32*, *38*

namePretty, *65*

optim, *28*
optimBoxCox, *24*, 27

p.adjust, *67*
percentile, 28

pgrubbs, *15*
pgrubbs *(*qgrubbs*)*, 48
plot.ancovaReg, *8*, 29
plot.binaryreg *(*plot.ancovaReg*)*, 29
plot.cor.all, *13*
plot.cor.all *(*plot.ancovaReg*)*, 29
plot.htest, 32
plot.kensen *(*plot.htest*)*, 32
plot.lecessie *(*plot.ancovaReg*)*, 29
plot.move.1, *22*
plot.move.1 *(*plot.ancovaReg*)*, 29
plot.move.2, *24*
plot.move.2 *(*plot.ancovaReg*)*, 29
plot.multReg, *27*
plot.multReg *(*plot.ancovaReg*)*, 29
plot.ppcc *(*plot.htest*)*, 32
plot.roc *(*plot.ancovaReg*)*, 29
plot.senSlope *(*plot.ancovaReg*)*, 29
plot.Stats *(*plot.ancovaReg*)*, 29
ppcc.test, 33
predict.move.1, *17*, *22*, 34
predict.move.2, *24*, 35
predict.senSlope, 36
predictDuan, 37
predictFerguson *(*predictDuan*)*, 37
predictMVUE *(*predictDuan*)*, 37
press, 38
print.ancovaReg, 39
print.binaryreg, 39
print.cor.all, 40
print.lecessie, 41
print.MCT, 41
print.move.1, 42
print.move.2, 43
print.multReg, 43
print.optimBoxCox, 44
print.roc, 45
print.seasonalPeak, 45, *57*
print.senSlope, 46
printCor, *41*, 47

qgrubbs, *15*, 48
qtiles.CI, 49
quantile, *49*, *69*
quantile.default, *51*
quantile.numeric, 50

regken, 51
regsubsets, *6*

regularSeries, *56*
rmse, 52
roc, *9*, *32*, 54
rpd *(*rmse*)*, 52

sd, *69*
seaken, *19*, *52*, 54, *66*
seasonalPeak, *11*, 56, *59*
seasonalWave, *11*, *57*, 58, *60, 61*
seasonalWave.fit, 59
seasonalWave.wt, 60
seasonPlot, *66*
selBestWave, 61
senSlope, *32*, *37*, 62
serial.test, *63*, 64
seriesPlot *(*seriesPlot.seaken*)*, 65
seriesPlot.seaken, 65
setPage, *32*
shapiro.test, *34*
skew, 66, *69*
smwrStats *(*smwrStats-package*)*, 3
smwrStats-package, 3
summary.cor.all, *13*, 67
sumStats, 68
supsmu, *57*

timeWeightedMean, 69
trends, *14*, 70

var, *69*
vif, 71

weighted.mean, *70*