Package 'uwIntroStats'

July 13, 2015

Type Package

Title Descriptive Statistics, Inference, Regression, and Plotting in an Introductory Statistics Course

Version 0.0.1

Date 2015-07-13

Maintainer Brian D. Williamson <brianw26@uw.edu>

Description

A set of tools designed to facilitate easy adoption of R for students in introductory classes with little programming experience. Compiles output from existing routines together in an intuitive format, and adds functionality to existing functions. For instance, the regression function can perform linear models, generalized linear models, Cox models, or generalized estimating equations. The user can also specify multiple-partial F-tests to print out with the model coefficients. We also give many routines for descriptive statistics and plotting.

Imports Exact, geepack, plyr, survival, sandwich

License GPL-2

NeedsCompilation no

Author Scott S. Emerson [aut],
Andrew J. Spieker [aut],
Brian D. Williamson [aut, cre],
R Core Team [ctb],
Terry M Therneu [ctb],
Thomas Lumley [ctb]

R topics documented:

`uwIntroStats-package` *Descriptive Statistics, One Sample Inference, Regression, and Plotting in an Introductory Statistics Course*

Description

Developed by Scott S. Emerson, M.D., Ph.D., Andrew J. Spieker, and Brian D. Williamson in the University of Washington Department of Biostatistics. Aims to facilitate more widespread use of R by implementing more intuitive layout and functionality for existing R functions.

Details

Package:	uwIntroStats
Type:	Package
Version:	3.3
Date:	2015-06-04
License:	GPL-2

A set of tools designed to facilitate easy adoption of R for students in introductory classes with little programming experience. Compiles output from existing routines together in an intuitive format, and adds functionality to existing functions. For instance, the regression function can perform linear models, generalized linear models, Cox models, or generalized estimating equations. The user can also specify multiple-partial F-tests to print out with the model coefficients. We also give many routines for descriptive statistics and plotting.

Author(s)

Scott S. Emerson, M.D., Ph.D, Andrew J. Spieker, and Brian D. Williamson

Maintainer: Brian D. Williamson <brianw26@uw.edu>

`bplot` *Boxplot with Lowess Curves, Jittered Data, Overlaid Mean and Standard Deviation, for an Arbitrary Number of Strata*

Description

This function adds functionality to the base R boxplot function. Now it is straightforward to add jittered data to the plot, and to overlay information about the sample mean and standard deviation. The function also supports stratification.

Usage

```
bplot(y, x=rep(1, length(y)), data = NA, strata = NULL, xjitter = TRUE, yjitter = TRUE, range = 0, s
      log = FALSE, cex = c(meancex = 1, jittercex = 0.8),
     col = c(sdcol = "dodgerblue3", jittercol = "gray30"), main = NULL, xlab = NULL,
      ylab = NULL, names = NULL, ylim = NULL, legend=FALSE)
```

Arguments

y	dependent variable.
x	independent variable. Must by the same length as y.
data	if intered, must contain both y and x.
strata	strata variable(s), used to stratify the plot.
xjitter	a logical specifying if the jittered data (jittered on x-axis) are to be displayed or not. Default is TRUE.
yjitter	a logical specifying if the jittered data (jittered on y-axis) are to be displayed or not. Default is TRUE.
range	passed to the boxplot() function. This determines how far the plot whiskers extend out from the box. If range is positive, the whiskers extend to the most extreme data point which is no more than range times the interquartile range from the box. A value of zero causes the whiskers to extend to the data extremes.
sd	a logical specifying if the standard devation of y should be overlaid on the plot. Default value is TRUE.
sdx	a logical specifying if the standard deviation of x should be overlaid on the plot. Default value is TRUE.
log	a logical specifying if the data are to be displayed on a log scale. Passed to boxplot(). Default value is FALSE.
cex	passed to boxplot().
col	passed to boxplot().
main	passed to plot(), the main title of the plot.
xlab	passed to plot(), the x-axis label.
ylab	passed to plot(), the y-axis label.
names	names (if any) of x.
ylim	the range for plotting the y-axis, passed to plot.
legend	a logical value. If TRUE, (and the means and standard deviations have been overlaid on the graph) displays a legend next to the first boxplot plotted denoting the max, mean+sd, mean, mean-sd, and min values.

Value

Produces a plot. No value is returned.

Author(s)

Scott S. Emerson, M.D., Ph.D., Andrew J. Spieker, Brian D. Williamson

See Also

boxplot

Examples

```
#- Read in and attach the data -#
fev <- read.table("http://courses.washington.edu/b511/Data/FEV1ClinTrial.dat", sep="")

#- Change the names of the fev data -#
names(fev) <- c("V1", "FEV", "SMOKE")
attach(fev)

#- Produce box plot with jittered data, sample mean, and sd -#
bplot(y=FEV, x=SMOKE, xlab="Smoking Group", ylab="FEV")
```

---

clusterStats *Summary Measures within Clusters*

Description

Produces a vector containing summary measures computed within clusters.

Usage

```
clusterStats(y, cluster = NULL, stat = "count", subset = NULL, x = NULL, ...,
             version = FALSE)
```

Arguments

y	a vector, Date, or Surv object for which within cluster summary statistics are desired.
cluster	vector, matrix, or list of variables defining clusters. Descriptive statistics will be computed within strata defined by each unique combination of the cluster variables.
stat	a character string indicating the descriptive statistic(s) to be returned for each cluster. See the documentation for tableStat() for a full description, although only single statistics can be specified in this function. If either "probabilities" or "quantiles" are specified, only the first such quantity is returned. In addition to the summary statistics allowed by tableStat(), a user can also specify within cluster least squares slopes (stat="slope") of y on x.
subset	a logical vector indicating a subset to be used for all descriptive statistics.
x	a numeric vector to be used as regression predictor for least squares slopes.
...	optional arguments specifying quantiles or thresholds for probabilities to be used in calculating summary statistics. See arguments for descrip().
version	if TRUE, the version of the function will be returned. No other computations will be performed.

Details

This function uses `tableStat()` to compute stratified statistics for each cluster. However, only single summary measures can be used in this function. See examples.

Value

A vector is returned that contains the summary statistic relevant for the cluster to which each observation in y belings. Although only the cases indicated by `subset` are used to calculate the summary statistics, values are expanded out to cases beyond those indicated by `subset`.

Author(s)

Scott S. Emerson, M.D., Ph.D., Andrew J. Spieker, Brian D. Williamson

Examples

```
# Load required libraries
library(survival)

# Reading in a dataset
mtx <- read.table("http://www.emersonstatistics.com/datasets/mtxlabs.txt",header=TRUE)

# Generating average bilirubin for each subject
mbili <- clusterStats (mtx$bili, mtx$ptid, "mean")
descrip(mbili,strata=mtx$tx)

# Generating average bilirubin for each subject while taking study drug
mdrugbili <- clusterStats (mtx$bili, mtx$ptid, "mean", subset=mtx$ondrug==1)
descrip(mdrugbili,strata=mtx$tx)

# Reading in a dataset
audio <- read.csv("http://www.emersonstatistics.com/datasets/audio.csv",header=TRUE)

# Generating counts for each subject
counts <- clusterStats (audio$R4000, audio$Subject, "count")
table(counts,strata=audio$Dose)

# Generating average R4000 for each subject
mR4000 <- clusterStats (audio$R4000, audio$Subject, "mean")
descrip(mR4000,strata=audio$Dose)

# Generating average R4000 for each subject after visit 0
mtxR4000 <- clusterStats (audio$R4000, audio$Subject, "mean", subset=audio$Visit>0)
descrip(mtxR4000,strata=audio$Dose)
```

---

correlate *Correlation*

Description

Computes correlation matrix for an arbitrary number of numeric variables, optionally within strata.

Usage

```
correlate(..., strata = NULL, subset = NULL, conf.level = 0.95,
          use = "pairwise.complete.obs", method = "pearson",
          stat = "cor", byStratum = TRUE, version = FALSE)
```

Arguments

`...`	an arbitrary number of variables for which a correlation matrix is desired. The arguments can be vectors, matrices, or lists. Individual columns of a matrix or elements of a list that are not of class `numeric`, `factor`, or `Date` will be omitted. Factor and Date variables are converted to integers. Character vectors will be coerced to numeric. Variables must all be of the same lengths.
`strata`	vector, matrix, or list of stratification variables. Descriptive statistics will be computed within strata defined by each unique combination of the stratification variables, as well as in the combined sample. If `strata` is supplied, all variables must be of that same length.
`subset`	vector indicating a subset to be used for all descriptive statistics. If `subset` is supplied, all variables must be of that same length.
`conf.level`	a numeric scalar between 0 and 1 denoting the confidence level to be used in constructing confidence intervals for the correlation.
`use`	character string denoting the cases to use: `"everything"` uses all cases (and causes NA when any needed variable is missing), `"complete.obs"` uses only those rows with no missing data for any variable, and `"pairwise.complete.obs"` computes pairwise correlations using all cases that are not missing data for the relevant variables.
`method`	character string denoting the correlation method to use: `"pearson"` denotes Pearson's correlation coefficient and `"spearman"` denotes Spearman's rank correlation.
`stat`	a vector of character strings indicating the descriptive statistic(s) to be tabulated. Possibilities include any statistic as specified by one or more of `"cor"`, `"n"`, `"t.stat"`, `"pval"`, `"loCI"`, or `"hiCI"`. Only enough of the string needs to be specified to disambiguate the choice. Alternatively (and more usefully), a single special format character string can be specified as described in the Details below.
`byStratum`	a logical scalar indicating whether statistics should be grouped by pair of variables. If `TRUE`, the results will be displayed in a series of tables where each table correspond to a single variable, with rows corresponding to different strata and columns reflecting all other variables. If `FALSE`, the results will be displayed in a series of tables where each table corresponds to a single stratum and rows and columns reflect the variables.
`version`	if `TRUE`, the version of the function will be returned. No other computations will be performed.

Value

An object of class uCorrelate is returned, which consists of a list of correlation estimates and inference for each specified stratum and for the combined dataset. Each element of the list has six arrays:

`cormtx`	the correlation matrix, printed.
`n`	matrix of sample sizes used to compute each correlation

t.stat	matrix of t-statistics, testing a correlation of 0.
pval	matrix of two-sided p-values for the t-test.
lo95%CI	lower bound of the 95% confidence interval.
hi95%CI	upper bound of the 95% confidence interval.

Author(s)

Scott S. Emerson, M.D., Ph.D., Andrew J. Spieker, Brian D. Williamson

Examples

```
# Load required libraries
library(survival)

# Reading in a dataset
mri <- read.table("http://www.emersonstatistics.com/datasets/mri.txt",header=TRUE)

# Estimated correlation matrix using all data, complete cases, or pairwise complete (the default)
with (mri, correlate(age,weight,ldl,use="everything"))
with (mri, correlate(age,weight,ldl,use="complete"))
with (mri, correlate(age,weight,ldl))

# Correlation matrices for each stratum
with (mri, correlate(age,weight,ldl,strata=male))

# Correlations grouped by variable
with (mri, correlate(age,weight,ldl,strata=male,byStratum=FALSE))

# Special formatting of inference for correlations within strata
with (mri, correlate(age,weight,ldl,strata=male,stat="@cor@ (@lo@, @hi@); P @p@; n= @n@"))

# Special formatting of inference for correlations grouped by variable
with (mri, correlate(age,weight,ldl,strata=male,stat="@cor@ (@lo@, @hi@); P @p@; n= @n@",
      byStratum=FALSE))
```

---

descrip *Descriptive Statistics*

Description

Produces table of relevant descriptive statistics for an arbitrary number of variables of class `integer`, `numeric`, `Surv`, `Date`, or `factor`. Descriptive statistics can be obtained within strata, and the user can specify that only a subset of the data be used. Descriptive statistics include the count of observations, the count of cases with missing values, the mean, standard deviation, geometric mean, minimum, and maximum. The user can specify arbitrary quantiles to be estimated, as well as specifying the estimation of proportions of observations within specified ranges.

Usage

```
descrip(..., strata = NULL, subset = NULL, probs = c(0.25, 0.5, 0.75),
       geomInclude = FALSE, replaceZeroes = FALSE, restriction = Inf, above = NULL,
       below = NULL, labove = NULL, rbelow = NULL, lbetween = NULL, rbetween = NULL,
         interval = NULL, linterval = NULL, rinterval = NULL, lrinterval = NULL,
         version = FALSE)
```

Arguments

...	an arbitrary number of variables for which descriptive statistics are desired. The arguments can be vectors, matrices, or lists. Individual columns of a matrix or elements of a list may be of class `numeric`, `factor`, `Surv`, or `Date`. Factor variables are converted to integers. Character vectors will be coerced to numeric. Variables may be of different lengths, unless `strata` or `subset` are non-`NULL`.
`strata`	vector, matrix, or list of stratification variables. Descriptive statistics will be computed within strata defined by each unique combination of the stratification variables, as well as in the combined sample. If `strata` is supplied, all variables must be of that same length.
`subset`	vector indicating a subset to be used for all descriptive statistics. If `subset` is supplied, all variables must be of that same length.
`probs`	a vector of probabilities between 0 and 1 indicating quantile estimates to be included in the descriptive statistics. Default is to compute 25th, 50th (median) and 75th percentiles.
`geomInclude`	If not `FALSE`, includes the geometric mean in the descriptive statistics. Default is `FALSE`.
`replaceZeroes`	if not `FALSE`, this indicates a value to be used in place of zeroes when computing a geometric mean. If `TRUE`, a value equal to one-half the lowest nonzero value is used. If a numeric value is supplied, that value is used for all variables.
`restriction`	a value used for computing restricted means, standard deviations, and geometric means with censored time to event data. The default value of `Inf` will cause restrictions at the highest observation. Note that the same value is used for all variables of class `Surv`.
`above`	a vector of values used to dichotomize variables. The descriptive statistics will include an estimate for each variable of the proportion of measurements with values greater than each element of `above`.
`below`	a vector of values used to dichotomize variables. The descriptive statistics will include an estimate for each variable of the proportion of measurements with values less than each element of `below`.
`labove`	a vector of values used to dichotomize variables. The descriptive statistics will include an estimate for each variable of the proportion of measurements with values greater than or equal to each element of `labove`.
`rbelow`	a vector of values used to dichotomize variables. The descriptive statistics will include an estimate for each variable of the proportion of measurements with values less than or equal to each element of `rbelow`.
`lbetween`	a vector of values with `-Inf` and `Inf` appended is used as cutpoints to categorize variables. The descriptive statistics will include an estimate for each variable of the proportion of measurements with values between successive elements of `lbetween`, with the left hand endpoint included in each interval.
`rbetween`	a vector of values with `-Inf` and `Inf` appended is used as cutpoints to categorize variables. The descriptive statistics will include an estimate for each variable of the proportion of measurements with values between successive elements of `rbetween`, with the right hand endpoint included in each interval.
`interval`	a two column matrix of values in which each row is used to define intervals of interest to categorize variables. The descriptive statistics will include an estimate for each variable of the proportion of measurements with values between two elements in a row, with neither endpoint included in each interval.

linterval	a two column matrix of values in which each row is used to define intervals of interest to categorize variables. The descriptive statistics will include an estimate for each variable of the proportion of measurements with values between two elements in a row, with the left hand endpoint included in each interval.
rinterval	a two column matrix of values in which each row is used to define intervals of interest to categorize variables. The descriptive statistics will include an estimate for each variable of the proportion of measurements with values between two elements in a row, with the right hand endpoint included in each interval.
lrinterval	a two column matrix of values in which each row is used to define intervals of interest to categorize variables. The descriptive statistics will include an estimate for each variable of the proportion of measurements with values between two elements in a row, with both endpoints included in each interval.
version	If TRUE, the version of the function will be returned. No other computations will be performed.

Details

This function depends on the survival R package. You should execute library(survival) if that library has not been previously installed. Quantiles are computed for uncensored data using the default method in quantile(). For variables of class factor, descriptive statistics will be computed using the integer coding for factors. For variables of class Surv, estimated proportions and quantiles will be computed from Kaplan-Meier estimates, as will be restricted means, restricted standard deviations, and restricted geometric means. For variables of class Date, estimated proportions will be labeled using the Julian date since January 1, 1970.

Value

An object of class uDescriptives is returned. Descriptive statistics for each variable in the entire subsetted sample, as well as within each stratum if any is defined, are contained in a matrix with rows corresponding to variables and strata and columns corresponding to the descriptive statistics.

N	the number of observations.
Msng	the number of observations with missing values.
Mean	the mean of the nonmissing observations (this is potentially a restricted mean for right censored time to event data).
Std Dev	the standard deviation of the nonmissing observations (this is potentially a restricted standard deviation for right censored time to event data).
Geom Mn	the geometric mean of the nonmissing observations (this is potentially a restricted geometric mean for right censored time to event data). Nonpositive values in the variable will generate NA, unless replaceZeroes was specified.
Min	the minimum value of the nonmissing observations (this is potentially restricted for right censored time to event data).
-	columns corresponding to the quantiles specified by probs
Max	the maximum value of the nonmissing observations (this is potentially restricted for right censored time to event data).
-	columns corresponding to the proportions as specified by above, below, labove, rbelow, lbetween, rbetween, interval, linterval, rinterval, lrinterval.
restriction	the threshold for restricted means, standard deviations, and geometric means.
FirstEvent	the time of the first event for censored time to event variables.
LastEvent	the time of the last event for censored time to event variables.
isDate	an indicator that the variable is a Date object.

Author(s)

Scott S. Emerson, M.D., Ph.D., Andrew J. Spieker, Brian D. Williamson

Examples

```
#- Load the data -#
clin.trial <- read.table("http://courses.washington.edu/b511/Data/FEV1ClinTrial.dat",
                         header=TRUE, sep="")
attach(clin.trial)

#- Load libraries -#
library(survival)

#- Create the table -#
descrip(clin.trial)
```

---

dummy *Create Dummy Variables*

Description

Creates dummy variables.

Usage

```
dummy(x,subset=rep(T,length(x)),
      reference=sort(unique(x[!is.na(x)])),includeAll=F, version=F)
```

Arguments

x	variable used to create the dummy variables.
subset	a subset of the data, if desired.
reference	the reference value for the dummy variables to compare to.
includeAll	logical value indicating whether all of the dummy variables should be returned (including the reference).
version	if TRUE, returns the version of the function and nothing else.

Value

A matrix containing the linear splines.

Author(s)

Scott S. Emerson, M.D., Ph.D., Andrew J. Spieker, Brian D. Williamson

Examples

```
# Reading in a dataset
mri <- read.table("http://www.emersonstatistics.com/datasets/mri.txt", header=TRUE)
attach(mri)
# Create a dummy variable for race
dummy(race)
```

lincom *Tests of Linear Combinations of Regression Coefficients*

Description

Produces point estimates, interval estimates, and p values for linear combinations of regression coefficients using a uRegress object.

Usage

```
lincom(reg, comb, hyp=0, conf.level=.95, robustSE = TRUE, eform=reg$fnctl!="mean")
```

Arguments

reg	an object of class uRegress.
comb	a vector or matrix containing the values of the constants which create the linear combination of the form $$c_0 + c_1\beta_1 + \ldots$$.
hyp	the null hypothesis to compare the linear combination of coefficients against. The default value is 0. An error will be thrown if the number of columns of this matrix are not equal to the number of coefficients in the model.
conf.level	a number between 0 and 1, indicating the desired confidence level for intervals.
robustSE	a logical value indicating whether or not to use robust standard errors in calculation. If TRUE, then robustSE must have been TRUE when reg was created.
eform	a logical value indicating whether or not to exponentiate the estimated coefficient. By default this is performed based on the type of regression used.

Value

Prints a matrix with the point estimate of the linear combination of coefficients, a p-value, and confidence interval.

Author(s)

Scott S. Emerson, M.D., Ph.D., Andrew J. Spieker, Brian D. Williamson

Examples

```
# Loading required libraries
library(survival)
library(sandwich)

# Reading in a dataset
mri <- read.table("http://www.emersonstatistics.com/datasets/mri.txt",header=TRUE)
attach(mri)

# Linear regression of LDL on age (with robust SE by default)
testReg <- regress ("mean", ldl~age+stroke)
```

```
# Testing coefficient created by .5*age - stroke (the first 1 comes from including the intercept)
testC <- c(1, .5, -1)
lincom(testReg, testC)
```

`lspline` *Create Linear Splines*

Description

Creates linear splines, mostly for use in regression.

Usage

```
lspline(x, knots, lbl=NULL,
        parameterization="absolute", version=FALSE)
lsplineD(x, knots, lbl=NULL, version=FALSE)
```

Arguments

`x`	variable used to create the linear splines.
`knots`	vector of knots to create the splines.
`lbl`	a label for the splines.
`parameterization`	defaults to`"absolute"`, and provides splines based on the absolute slope between knots. If `"change"`, provides splines based on the change from knot to knot. If `lsplineD` is called, `"change"` is entered by default.
`version`	if `TRUE`, returns the version of the function and nothing else.

Value

A matrix containing the linear splines.

Author(s)

Scott S. Emerson, M.D., Ph.D., Andrew J. Spieker, Brian D. Williamson

Examples

```
# Reading in a dataset
mri <- read.table("http://www.emersonstatistics.com/datasets/mri.txt", header=TRUE)
attach(mri)
# Create a spline based on absolute
lspline(ldl, c(70, 100, 130, 160))

# Create a spline based on change
lsplineD(ldl, c(70, 100, 130, 160))
```

mri *MRI data*

Description

An MRI data set hosted on Scott Emerson's webpage. More detailed description is hosted there, at "http://www.emersonstatistics.com/datasets/mri.pdf".

Usage

```
data(mri)
```

Format

A data frame with 735 observations on the following 30 variables.

`ptid` participant identification number.

`mridate` the date on which the participant underwent MRI scan in MMDDYY format.

`age` participant age at time of MRI, in years.

`male` indicator of whether participant is male (0=female, 1=male).

`race` indicator of participant's race (1=white, 2=black, 3=Asian, 4=other).

`weight` participant's weight at time of MRI (pounds).

`height` participant's height at time of MRI (centimeters).

`packyrs` participant smoking history in pack years (1 pack year = smoking 1 pack of cigarettes per day for 1 year). A participant who has never smoked has 0 pack years.

`yrsquit` number of years since quitting smoking. A current smoker will have a nonzero packyrs and a 0 for yrsquit. A never smoker will have a zero for both variables.

`alcoh` average alcohol intake for the participant for the two weeks prior to MRI (drinks per week, where one drink is 1 oz. whiskey, 4 oz. wine, or 12 oz.beer).

`physact` physical activity of the participant for the week prior to MRI (1,000 kcal).

`chf` indicator of whether the participant had been diagnosed with congestive heart failure prior to MRI (0=no, 1=yes).

`chd` indicator of whether the participant had been diagnosed with coronary heart disease prior to MRI (0=no, 1=diagnosis of angina, 2=diagnosis of myocardial infarction).

`stroke` indicator of whether the participant had been diagnosed with a cerebrovascular event prior to MRI (0=no, 1=diagnosis of a transient ischemic attack, 2=diagnosis of stroke).

`diabetes` indicator of whether the participant had been diagnosed with diabetes prior to MRI (0=no, 1=yes).

`genhlth` an indicator of the participant's view of their own health (1=excellent, 2=very good, 3=good, 4=fair, 5=poor)

`ldl` a laboratory measure of low density lipoprotein (a kind of cholesterol) in the participant's blood at the time of MRI (mg/dL).

`alb` a laboratory measure of albumin, a kind of protein, in the participant's blood at the time of MRI (g/L).

`crt` a laboratory measure of creatinine, a waste product, in the participant's blood at the time of MRI (mg/dL).

`plt` a laboratory measure of the number of platelets circulating in the participant's blood at the time of MRI (1000 per cubic mm).

`sbp` a measurement of the participant's systolic blood pressure in their arm at the time of MRI (mm Hg).

`aai` the ratio of systolic blood pressure measured in the participant's ankle at time of MRI to the systolic blood pressure in the participant's arm.

`fev` a measure of the forced expiratory volume in the participant at the time of MRI (L/sec).

`dsst` a measure of cognitive function (Digit Symbol Substitution Test) for the participant at the time of MRI. Maximum score possible is 100.

`atrophy` a measure of global brain activity detected on MRI. Measurements range from 0 to 100, with 100 being the most severe atrophy.

`whgrd` a measure of white matter changes detected on MRI. 0 means no changes, 9 means marked changes.

`numinf` a count of the number of distinct regions identified on MRi scan which were suggestive of infarcts.

`volinf` a measure of the total volume of infarct-like lesions found on MRI scan (cubic cm).

`obstime` the total time (in days) that the participant was observed on study between the date of MRI and death or September 16, 1997, whichever came first.

`death` an indicator that the participant was observed to die while on study. If 1, the number of days recorded in `obstime` is the number of days from that participant's MRI to their death. If 0, the number of days in `obstime` is the number of days between that participant's MRI and September 16, 1997.

`oneSample` *One Sample Inferential Methods*

Description

Produces point estimates, interval estimates, and p values for an arbitrary functional (mean, geometric mean, proportion, median, quantile, odds) of a variable of class `integer`, `numeric`, `Surv`, or `Date`. A variety of inferential methods are provided, with the choices depending on the functional and the data type.

Usage

```
oneSample(fnctl, y, null.hypothesis = NA, test.type = "two.sided", subset = rep(TRUE, N),
          conf.level = 0.95, na.rm = TRUE, probs = 0.5, replaceZeroes = NULL,
        restriction = Inf, subjTime = rep(1, length(y)), method = NULL, above = NULL,
        below = NULL, labove = NULL, rbelow = NULL, interval = NULL, linterval = NULL,
          rinterval = NULL, lrinterval = NULL, g1 = 1, g2 = 0, dispersion = 1,
          nbstrap = 10000, resample = "pairs", seed = 0, ..., version = FALSE)
```

Arguments

`fnctl` a character string indicating the functional (summary measure of the distribution) for which inference is desired. Choices include `"mean"`, `"geometric mean"`, `"proportion"`, `"median"`, `"quantile"`, `"odds"`, `"rate"`. The character string may be shortened to a unique substring. Hence `"mea"` will suffice for `"mean"`.

y	a variable for which inference is desired. The variable may be of class numeric, Surv, or Date.
null.hypothesis	a numeric scalar indicating any null hypothesis to be tested.
test.type	a character string indicating whether a hypothesis test is to be of a one sided test of a lesser alternative hypothesis ("less"), a one sided test of a greater alternative hypothesis ("greater"), or a test of a two sided alternative hypothesis ("two.sided"). The default value is "two.sided".
subset	a vector indicating a subset to be used for all inference.
conf.level	a numeric scalar indicating the level of confidence to be used in computing confidence intervals. The default is 0.95.
na.rm	an indicator that missing data is to be removed prior to computation of the descriptive statistics.
probs	a vector of probabilities between 0 and 1 indicating quantile estimates to be included in the descriptive statistics. Default is to the 50th (median) percentile.
replaceZeroes	if not FALSE, this indicates a value to be used in place of zeroes when computing a geometric mean. If TRUE, a value equal to one-half the lowest nonzero value is used. If a numeric value is supplied, that value is used for all variables.
restriction	a value used for computing restricted means, standard deviations, and geometric means with censored time to event data. The default value of Inf will cause restrictions at the highest observation. Note that the same value is used for all variables of class Surv.
subjTime	a vector of values for use with rates.
method	a character string used to indicate inferential methods. Allowed choices depend on the variable type and the functional. Default values are "t.test" for means and geometric means, and "exact" for proportions of uncensored data, and "KM" for censored survival data.
above	a vector of values used to dichotomize variables. The descriptive statistics will include an estimate for each variable of the proportion of measurements with values greater than each element of above.
below	a vector of values used to dichotomize variables. The descriptive statistics will include an estimate for each variable of the proportion of measurements with values less than each element of below.
labove	a vector of values used to dichotomize variables. The descriptive statistics will include an estimate for each variable of the proportion of measurements with values greater than or equal to each element of labove.
rbelow	a vector of values used to dichotomize variables. The descriptive statistics will include an estimate for each variable of the proportion of measurements with values less than or equal to each element of rbelow.
interval	a two column matrix of values in which each row is used to define intervals of interest to categorize variables. The descriptive statistics will include an estimate for each variable of the proportion of measurements with values between two elements in a row, with neither endpoint included in each interval.
linterval	a two column matrix of values in which each row is used to define intervals of interest to categorize variables. The descriptive statistics will include an estimate for each variable of the proportion of measurements with values between two elements in a row, with the left hand endpoint included in each interval.

rinterval	a two column matrix of values in which each row is used to define intervals of interest to categorize variables. The descriptive statistics will include an estimate for each variable of the proportion of measurements with values between two elements in a row, with the right hand endpoint included in each interval.
lrinterval	a two column matrix of values in which each row is used to define intervals of interest to categorize variables. The descriptive statistics will include an estimate for each variable of the proportion of measurements with values between two elements in a row, with both endpoints included in each interval.
g1	used in method="mean-variance".
g2	used in method="mean-variance".
dispersion	dispersion, used in method="mean-variance".
nbstrap	number of bootstrap iterations to perform, used with method="bootstrap".
resample	character string specifying how the bootstrap should resample, used with method="bootstrap".
seed	sets the seed (for random number generation), used with method="bootstrap".
...	other arguments.
version	if TRUE, the version of the function will be returned. No other computations will be performed.

Details

Default values for inference correspond to the most commonly implemented methods. Additional methods are provided more for educational purposed than for purposes of statistical analysis.

Value

An object of class uOneSample is returned. Inferential statistics are contained in a vector named $Inference that includes the sample size, the point estimate, the lower and upper bounds of a confidence interval, any null hypothesis that was specified, and the p-value. Also included is a vector named $Statistics that includes more technical information. There is a print method that will format the descriptive statistics for the Date and Surv objects.

Author(s)

Scott S. Emerson, M.D., Ph.D., Andrew J. Spieker, Brian D. Williamson

Examples

```
# Load required libraries
library(survival)

# Reading in a dataset
mri <- read.table("http://www.emersonstatistics.com/datasets/mri.txt",header=TRUE)

# Creating a Surv object to reflect time to death
mri$ttodth <- Surv(mri$obstime,mri$death)

# Reformatting an integer MMDDYY representation of date to be a Date object
mri$mridate <- as.Date(paste(trunc(mri$mridate/10000),trunc((mri$mridate %% 10000)/100),
mri$mridate %% 100,sep="/"),"%m/%d/%y")

# Inference about the mean LDL: a two sample t test that mean LDL is 135 mg/dl
oneSample ("mean", mri$ldl, null.hypothesis=125)
```

```
# Inference about the mean LDL: a one sample t test of a lesser alternative
# that mean LDL is 125 mg/dl
oneSample ("mean", mri$ldl, null.hypothesis=125, test.type="less")

# Inference about the mean LDL: a one sample t test of a greater alternative
# that mean LDL is 125 mg/dl
oneSample ("mean", mri$ldl, null.hypothesis=125, test.type="greater")

# Inference about the geometric mean LDL: a one sample t test of a greater
# alternative that geometric mean LDL is 125 mg/dl
oneSample ("geom", mri$ldl, null.hypothesis=125, test.type="greater")

# Inference about the proportion of subjects with LDL greater than 128: exact binomial
# inference that 50% of subjects have LDL greater than 128 mg/dl
oneSample ("prop", mri$ldl, null.hypothesis=0.5, above=128)
oneSample ("prop",mri$ldl>128, null.hypothesis=0.5)
```

`polynomial` *Create Polynomials*

Description

Creates polynomial variables.

Usage

```
polynomial(x,degree=2,center=mean(x,na.rm=T), version=F)
```

Arguments

`x`	variable used to create the polynomials.
`degree`	the maximum degree polynomial to be returned. Polynomials of degree <= degree will be returned.
`center`	the value to center the polynomials at.
`version`	if `TRUE`, returns the version of the function and nothing else.

Value

A matrix containing the linear splines.

Author(s)

Scott S. Emerson, M.D., Ph.D., Andrew J. Spieker, Brian D. Williamson

Examples

```
# Reading in a dataset
mri <- read.table("http://www.emersonstatistics.com/datasets/mri.txt", header=TRUE)
attach(mri)
# Create a polynomial on ldl
polynomial(ldl, degree=3)
```

predict.uRegress *Prediction Intervals for* uRegress *objects*

Description

Produces prediction intervals for objects of class uRegress.

Usage

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

Arguments

object	an object of class uRegress.
...	other arguments to pass to the appropriate predict function for the class of object$fit. See predict.coxph, predict.lm, or predict.glm for more details. Predictions are not currently implemented for objects of type geeglm.

Value

Returns a matrix with the fitted value and prediction interval for the entered X.

Author(s)

Scott S. Emerson, M.D., Ph.D., Andrew J. Spieker, Brian D. Williamson

See Also

regress

Examples

```
# Loading required libraries
library(survival)
library(sandwich)

# Reading in a dataset
mri <- read.table("http://www.emersonstatistics.com/datasets/mri.txt",header=TRUE)
attach(mri)

# Linear regression of LDL on age (with robust SE by default)
testReg <- regress ("mean", ldl~age)

# 95% Prediction Interval for age 50
predict(testReg)
```

regress *General Regression for an Arbitrary Functional*

Description

Produces point estimates, interval estimates, and p values for an arbitrary functional (mean, geometric mean, proportion, median, quantile, odds) of a variable of class `integer`, `numeric`, `Surv`, when regressed on an arbitrary number of covariates. Multiple Partial F-tests can be specified using the `U` function.

Usage

```
regress(fnctl, formula, data, intercept = fnctl!="hazard",
       strata = rep(1,n),weights=rep(1,n),id=1:n,ties="efron",subset=rep(TRUE,n),
         robustSE = TRUE, conf.level = 0.95, exponentiate = fnctl!="mean",
       replaceZeroes, useFdstn = TRUE, suppress = FALSE, na.action, method = "qr",
         model.f = TRUE, model.x = FALSE, model.y = FALSE, qr = TRUE,
         singular.ok = TRUE, contrasts = NULL, offset,control = list(...),
         tt, init, ..., version=FALSE)
```

Arguments

`fnctl`	a character string indicating the functional (summary measure of the distribution) for which inference is desired. Choices include `"mean"`, `"geometric mean"`, `"odds"`, `"rate"`, `"hazard"`. The character string may be shortened to a unique substring. Hence `"mea"` will suffice for `"mean"`.
`formula`	an object of class `formula` as might be passed to `lm`, `glm`, or `coxph`.
`data`	a data frame, matrix, or other data structure with matching names to those entered in `formula`.
`intercept`	a logical value indicating whether a intercept exists or not.
`strata`	vector indicating a variable to be used for stratification in proportional hazards regression.
`weights`	vector indicating optional weights for weighted regression.
`id`	vector with ids for the variables. If any ids are repeated, runs a clustered regression.
`ties`	One of `"efron"` (by default), `"breslow"`, or `"exact"`. Determines the method used to handle ties in proportional hazard regression.
`subset`	vector indicating a subset to be used for all inference.
`robustSE`	a logical indicator that standard errors are to be computed using the Huber-White sandwich estimator.
`conf.level`	a numeric scalar indicating the level of confidence to be used in computing confidence intervals. The default is 0.95.
`exponentiate`	a logical indicator that the regression parameters should be exponentiated. This is by default true for all functionals except the mean.
`replaceZeroes`	if not `FALSE`, this indicates a value to be used in place of zeroes when computing a geometric mean. If `TRUE`, a value equal to one-half the lowest nonzero value is used. If a numeric value is supplied, that value is used.

useFdstn	a logical indicator that the F distribution should be used for test statistics instead of the chi squared distribution even in logistic and proportional hazard regression models. When using the F distribution, the degrees of freedom are taken to be the sample size minus the number of parameters, as it would be in a linear regression model.
suppress	if TRUE, and a model which requires exponentiation (for instance, regression on the geometric mean) is computed, then a table with only the exponentiated coefficients and confidence interval is returned. Otherwise, two tables are returned - one with the original unexponentiated coefficients, and one with the exponentiated coefficients.
na.action, method, model.f, model.x, model.y, qr, singular.ok, offset, contrasts, control	optional arguments that are passed to the functionality of lm or glm.
tt, init	optional arguments that are passed to the functionality of coxph.
...	other arbitrary parameters.
version	if TRUE, returns the version of the function. No other computation is performed.

Details

Regression models include linear regression (for the "mean" functional), logistic regression (for the "odds" functional), Poisson regression (for the "rate" functional), and proportional hazards regression (for the "hazard" functional). Objects created using the U function can also be passed in. If the U call involves a partial formula of the form ~ var1 + var2, then regress will return a multiple-partial F-test involving var1 and var2. The multiple partial tests must be the last terms specified in the model (i.e. no other predictors can follow them).

Value

An object of class uRegress is returned. Parameter estimates, confidence intervals, and p values are contained in a matrix $augCoefficients.

Author(s)

Scott S. Emerson, M.D., Ph.D., Andrew J. Spieker, Brian D. Williamson

See Also

Functions for fitting linear models (lm), generalized linear models (glm), proportional hazards models (coxph), and generalized estimating equations (geeglm). Also see the function to specify multiple-partial F-tests, U.

Examples

```
# Loading required libraries
library(survival)
library(sandwich)

# Reading in a dataset
mri <- read.table("http://www.emersonstatistics.com/datasets/mri.txt",header=TRUE)

# Creating a Surv object to reflect time to death
mri$ttodth <- Surv(mri$obstime,mri$death)

# Attaching the mri dataset
```

```
attach(mri)

# Linear regression of atrophy on age
regress("mean", atrophy~age, data=mri)

## Linear regression of atrophy on male and race and their interaction, with a multiple-partial F-test on the
regress("mean", atrophy~ male + U(ra=~race*age), data=mri)

## Linear regression of atrophy on age, male, race (as a dummy variable), chf,
## and diabetes. There are two multiple partial F-tests and both are named
regress("mean", atrophy~age+male+U(rc=~dummy(race)+chf)+U(md=~male+diabetes), data=mri)

## Proportional hazards regression clustered on id (here it makes no difference because the ids are unique)
regress("hazard", ttodth~ldl, id=ptid, data=mri)
```

---

scatter *Scatter Plot with Lowess Curves*

Description

Produces a scatter plot of two variables with (possibly stratified) superimposed lowess smooths and least squares fitted lines.

Usage

```
scatter(y, x, strata=rep(1,length(y)), subset= rep(TRUE,length(y)),
        reference=sort(unique(strata)), plotPoints=TRUE, plotLowess=TRUE,
       plotLSfit=FALSE, legend=0.05, colors=c("black", "blue", "orange", "pink",
        "green", "red", "cornflowerblue", "darkolivegreen", "magenta"),
        xJitter=TRUE, yJitter=FALSE, newplot=TRUE,lty=1:6, lwd=1, pch=1:25,
        ..., version=FALSE)
```

Arguments

y	a numeric vector containing the values to be plotted on the y-axis.
x	a numeric vector containing the values to be plotted on the x-axis.
strata	a vector, matrix, or list of stratification variables. Descriptive statistics will be computed within strata defined by each unique combination of the stratification variables, as well as in the combined sample. If strata is supplied, all variables must be of that same length.
subset	a vector indicating a subset to be used for all descriptive statistics. If subset is supplied, all variables must be of that same length.
reference	a list of the strata in the order they are to be plotted.
plotPoints	an indicator that points are to be plotted. A different color and point type combination will be used for each stratum. Default value is TRUE.
plotLowess	an indicator that lowess smoths are to be plotted. A different color and line type combination will be used for each stratum. Default value is TRUE.
plotLSfit	an indicator that least squares fitted lines are to be plotted. A different color and line type combination will be used for each stratum. Default value is TRUE.

legend	where to place the legend on the plot
colors	a vector of colors used in plotting strata.
xJitter	the proportion of the minimal difference between adjacent x-values divided by 8 by which plotted points are to be jittered in the x-dimension. A value of 0 implies no jittering.
yJitter	the proportion of the minimal difference between adjacent y-values divided by 8 by which plotted points are to be jittered in the y-dimension. A value of 0 implies no jittering.
newplot	logical value indicating that the graph should be plotted on a new set of axes. Default value is TRUE.
lty, lwd, pch	plotting parameters.
...	optional arguments for plotting parameters (e.g. xlab, ylab, main) that will be passed to plot().
version	if TRUE, the version of the function will be returned. No other computations will be performed.

Value

This function produces a plot. No value is returned.

Author(s)

Scott S. Emerson, M.D., Ph.D., Andrew J. Spieker, Brian D. Williamson

Examples

```
#- Read in data set -#
clin.trial <- read.table("http://courses.washington.edu/b511/Data/FEV1ClinTrial.dat",
                         header=TRUE, sep="")
#- Set the names -#
names(clin.trial) <- c("Y0", "Y1", "T")
attach(clin.trial)

#- Create a scatterplot with lowess curves and least squares fitted regression lines -#
scatter(Y1, Y0, ylab="FEV1 At 24 Weeks", xlab="FEV1 at Baseline")

#- Create a scatterplot with lowess curves and least squares fitted regression lines -#
scatter(Y1, Y0, strata=T, ylab="FEV1 At 24 Weeks", xlab="FEV1 at Baseline")
```

---

tableStat *Table of Stratified Descriptive Statistics*

Description

Produces a table of stratified descriptive statistics for a single variable of class integer, numeric, Surv, Date, or factor. Descriptive statistics are those that can be estimated using the descrip function.

Usage

```
## S3 class 'tableStat'
tableStat(variable=NULL, ..., stat="count", printer=TRUE, na.rm=TRUE,
        subset=NULL, probs= c(.25,.50,.75), replaceZeroes=FALSE,
      restriction=Inf, above=NULL, below=NULL, labove=NULL, rbelow=NULL, lbetween=NULL,
      rbetween=NULL, interval=NULL, linterval=NULL, rinterval=NULL, lrinterval=NULL,
        version=FALSE)
```

Arguments

variable	a vector or Surv object suitable for use as an argument to descrip(). If a NULL value is supplied for variable, the valid statistics returned by the function is only the cross-tabulation of counts and percentages within strata.
...	an arbitrary number of stratification variables. The arguments can be vectors, matrices, or lists. Individual columns of a matrix or elements of a list may be of class numeric, factor, or character. Stratification variables must all be the same length as each other and (if it is supplied) variable.
stat	a vector of character strings indicating the descriptive statistic(s) to be tabulated within strata. Possibilities include any statistic returned by descrip() as specified by one or more of "count", "missing", "mean", "geometric mean", "median", "sd", "variance", "minimum", "maximum", "quantiles", "probabilities", "mn(sd)", "range", "iqr", "all", "row%", "col%", or "tot%". Only enough of the string needs to be specified to disambiguate the choice. Alternatively (and more usefully), a single special format character string can be specified as described in the Details below.
printer	a logical indicating whether or not the function should return the values necessary for a print with special characters as laid out in stat.
na.rm	an indicator that missing data is to be removed prior to computation of the descriptive statistics.
subset	vector indicating a subset to be used for all descriptive statistics. If subset is supplied, all variables must be of that same length.
probs	a vector of probabilities between 0 and 1 indicating quantile estimates to be included in the descriptive statistics. Default is to compute 25th, 50th (median) and 75th percentiles.
replaceZeroes	if not FALSE, this indicates a value to be used in place of zeroes when computing a geometric mean. If TRUE, a value equal to one-half the lowest nonzero value is used. If a numeric value is supplied, that value is used for all variables.
restriction	a value used for computing restricted means, standard deviations, and geometric means with censored time to event data. The default value of Inf will cause restrictions at the highest observation. Note that the same value is used for all variables of class Surv.
above	a vector of values used to dichotomize variables. The descriptive statistics will include an estimate for each variable of the proportion of measurements with values greater than each element of above.
below	a vector of values used to dichotomize variables. The descriptive statistics will include an estimate for each variable of the proportion of measurements with values less than each element of below.
labove	a vector of values used to dichotomize variables. The descriptive statistics will include an estimate for each variable of the proportion of measurements with values greater than or equal to each element of labove.

rbelow	a vector of values used to dichotomize variables. The descriptive statistics will include an estimate for each variable of the proportion of measurements with values less than or equal to each element of rbelow.
lbetween	a vector of values with -Inf and Inf appended is used as cutpoints to categorize variables. The descriptive statistics will include an estimate for each variable of the proportion of measurements with values between successive elements of lbetween, with the left hand endpoint included in each interval.
rbetween	a vector of values with -Inf and Inf appended is used as cutpoints to categorize variables. The descriptive statistics will include an estimate for each variable of the proportion of measurements with values between successive elements of rbetween, with the right hand endpoint included in each interval.
interval	a two column matrix of values in which each row is used to define intervals of interest to categorize variables. The descriptive statistics will include an estimate for each variable of the proportion of measurements with values between two elements in a row, with neither endpoint included in each interval.
linterval	a two column matrix of values in which each row is used to define intervals of interest to categorize variables. The descriptive statistics will include an estimate for each variable of the proportion of measurements with values between two elements in a row, with the left hand endpoint included in each interval.
rinterval	a two column matrix of values in which each row is used to define intervals of interest to categorize variables. The descriptive statistics will include an estimate for each variable of the proportion of measurements with values between two elements in a row, with the right hand endpoint included in each interval.
lrinterval	a two column matrix of values in which each row is used to define intervals of interest to categorize variables. The descriptive statistics will include an estimate for each variable of the proportion of measurements with values between two elements in a row, with both endpoints included in each interval.
version	If TRUE, the version of the function will be returned. No other computations will be performed.

Details

This function uses descrip() to compute the descriptive statistics. In addition to the basic choices specified above for stat, the user can supply a special format character string. Arbitrary text can be specified to label any of the descriptive statistics, which are indicated by bracketing with a "@". All text bracketed by a "@" must refer to a descriptive statistic, and all other text is printed verbatim. For instance, a display of the mean, standard deviation, minimum, maximum, and sample size might be specified by "@mean@ (@sd@; @min@ - @max@; n=@count@)". Similarly, a cross tabulation displaying counts, row percentages, column percentages, and percentages of the total might be specified by "@count@ (r @row%@; c @col%@; t @tot%@)". See examples for more detal. Any call to tableStat() will run tableStat.default(), with user specified values in place of the appropriate defaults.

Value

An object of class tableStat is returned, which consists of a list of arrays. Each array corresponds to a table of stratified statistics for one of the possible choices of stat. The print method provides the formatted output for the choice specified in stat.

Author(s)

Scott S. Emerson, M.D., Ph.D., Andrew J. Spieker, Brian D. Williamson

Examples

```
# Load required libraries
library(survival)

# Reading in a dataset
mri <- read.table("http://www.emersonstatistics.com/datasets/mri.txt",header=TRUE)

# Creating a Surv object to reflect time to death
mri$ttodth <- Surv(mri$obstime,mri$death)

# Reformatting an integer MMDDYY representation of date to be a Date object
mri$mridate <- as.Date(paste(trunc(mri$mridate/10000),trunc((mri$mridate %% 10000)/100),
mri$mridate %% 100,sep="/"),"%m/%d/%y")

# Cross tabulation of counts with sex and race strata
with (mri, tableStat (NULL, race, male, stat= "@count@ (r @row%@; c @col%@; t @tot%@)"))

# Cross tabulation of counts with sex, race, and coronary disease strata
# (Note row and column percentages are defined within the first two strata, while overall
# percentage considers all strata)
with (mri, tableStat (NULL, race, male, chd,
stat= "@count@ (r @row%@; c @col%@; t @tot%@)"))

# Description of time to death with appropriate quantiles
with (mri, tableStat(ttodth,probs=c(0.05,0.1,0.15,0.2),
stat="mean @mean@ (q05: @q@; q10: @q@; q15: @q@; q20: @q@; max: @max@)"))

# Description of mridate with mean, range stratified by race and sex
with (mri, tableStat(mridate, race, male,
stat="mean @mean@ (range @min@ - @max@)"))

# Stratified descriptive statistics with proportions
with (mri, tableStat(age,stat=">75: @p@; >85: @p@; [-Inf,75): @p@; [75,85): @p@;
      [85,Inf): @p@"), above=c(75,85),lbetween=c(75,85))

# Descriptive statistics on a subset comprised of males
with (mri, tableStat(dsst,age,stroke,subset=male==1,
stat="@mean@ (@sd@; n= @count@/@missing@)"))
```

`tabulate` *Table Variables, with Stratification and Statistical Tests*

Description

Takes in a list of strata, and returns a stratified table of counts (can also include other descriptives as specified by the user) as well as the chi-squared summaries for each. Relies on the `Exact` package for calculating Odds Ratios and Risk Ratios, the `plyr` package for array manipulation, and the `survival` package for censored data analysis.

Usage

```
## S3 method for class 'tabulate'
tabulate(..., dispRatios=FALSE, stratified=TRUE, tests=NULL,
```

```
stat="count", na.rm=TRUE, subset=NULL, probs= c(.25,.50,.75),
replaceZeroes=FALSE, restriction=Inf, above=NULL,
below=NULL, labove=NULL, rbelow=NULL, lbetween=NULL, rbetween=NULL,
interval=NULL, linterval=NULL, rinterval=NULL, lrinterval=NULL,
version=FALSE)
```

Arguments

`...`	a list of strata variables. All variables must have the same length.
`dispRatios`	can only be used if the first two variables only take on two values. Default is `FALSE`. If `TRUE`, displays Odds Ratio and Relative Risk (the user must decide which is appropriate for inference). It is assumed that the lower value of the data is healthy and the higher value is diseased (for example `0` might be no stroke and `1` would be had stroke).
`stratified`	default is `TRUE`, displays chi-squared statistics by stratum. Also if row/column/total percentages are requested, displays stratified percentages. If `FALSE`, only the overall chi-squared statistic is displayed and row/column/total percentages are calculated over all values.
`tests`	the type of tests to include in addition to the chi-squared test. Options are `"lrchisq"` - Likelihood Ratio Chi-squared Test, `"lr"` - Likelihood Ratio Test (logistic regression), `"fisher"` - Fisher's Exact Test, `"mh"` - Mantel-Haenszel Test, `"uWald"` - Barnard's Unconditional Exact Test (Wald statistic), `"uScore"` - Barnard's Unconditional Exact Test (score statistic), `"score"` - calculate Rao's Score statistic (logistic regression), and `"wald"` - Wald Chi-square Test (logistic regression). If `"mh"` is entered, the first three strata variables must have dimension at least 2.
`stat, na.rm, subset, probs, replaceZeroes, restriction, above, below, labove, rbelow, lbetween,`	variables passed to `descrip()`. For a detailed description, read the file on `descrip()`.

Details

If `stratified=TRUE`, prints the stratified count tables and chi-squared summaries. The printed result is a matrix with columns for (if appropriate): the point estimate, test statistic, degrees of freedom, 95% confidence interval, p-value, and any warnings in computation. If requested, row/column/total percentages are also stratified based on the first two variables entered. If `stratified=FALSE`, then the overall chi-squared statistic is returned and percentages are calculated from the overall table. If specified the user can also display Odds Ratio/Risk Ratio, likelihood-ratio test, Fisher's Exact test, Wald test, Barnard's Unconditional Exact test, Rao's Score test, and Mantel-Haenszel test results. The Mantel-Haenzsel chi-squared estimate, confidence interval, and estimate of the odds ratio are only returned if the first three variables entered into `tabulate()` are 2 by 2 by K. Any call to `tabulate()` will run `tabulate.default()`, with user specified values in place of the appropriate defaults.

Value

Returns an object of class `tabulate`. Contains two values in a list:

`rslt`	the raw result, which has all of the tables of counts.
`printer`	the raw result with descriptive statistics, if specified.

Author(s)

Scott S. Emerson, M.D., Ph.D., Andrew J. Spieker, Brian D. Williamson

See Also

exact.test, fisher.test, aaply, descrip, tableStat

Examples

```
## load the necessary libraries
library(survival)
library(Exact)
library(plyr)

## read in the mri dataset
mri <- read.table("http://www.emersonstatistics.com/datasets/mri.txt",header=TRUE)

## attach the mri dataset
attach(mri)

## create a table of stroke and race
tabulate(stroke, race)

## perform a chi-squared test of stroke vs race, display the count, row percentage,
## and column percentage
tabulate(stroke, race, stat="@count@ @row%@ @col%@")

## perform chi-squared test, likelihood ratio test, and fisher's exact test
## for stroke vs race
tabulate(stroke, race, tests=c("lrchisq", "fisher"))

## for diabetes vs male by race, perform chi-squared test, display
## odds ratio/risk ratio, mantel-haenzsel, likelihood ratio chi-squared
tabulate(diabetes, male, race, dispRatios=TRUE, tests=c("lrchisq", "mh"))
```

---

ttest *T-test with Improved Layout*

Description

Produces table of relevant descriptive statistics and inference for either one- or two-sample t-test. In the two-sample case, the user can specify whether or not observations are matched, and whether or not equal variances should be presumed. Can also perform a test of equality of proportions (Wald based or exact binomial based).

Usage

```
## S3 method for class "ttest"
ttest(var1, var2 = NA, by = NA, strat=NULL, geom=FALSE, prop=FALSE,
        exact=FALSE, null.hypoth = 0, test.type = "two.sided",
        var.eq = FALSE, conf.level = 0.95, matched = FALSE, more.digits = 0)
```

Arguments

var1	a (non-empty) numeric vector of data values.
var2	an optional (non-empty) numeric vector of data.
by	a variable of equal length to that of var1 with two outcomes. This will be used to define strata for a t-test on var1.
strat	a variable to use instead of by. However, using by instead is recommended.
geom	a logical indicating whether the geometric mean should be calculated and displayed.
prop	if TRUE, performs a test of equality of proportions with Wald based confidence intervals.
exact	must be FALSE if prop=FALSE. If true, performs a test of equality of proportions with Exact Binomial based confidence intervals.
null.hypoth	a number specifying the null hypothesis for the mean (or difference in means if performing a two-sample test). Defaults to zero.
test.type	a string: one of "less", "two.sided", or "greater" specifying the form of the test. Defaults to a two-sided test.
var.eq	a logical value, either TRUE or FALSE (default), specifying whether or not equal variances should be presumed in a two-sample t-test. Also controls robust standard errors.
conf.level	confidence level of the test. Defaults to 95/100.
matched	a logical value, either TRUE or FALSE, indicating whether or not the variables of a two-sample t-test are matched. Variables must be of equal length.
more.digits	a numeric value specifying whether or not to display more or fewer digits in the output. Non-integers are automatically rounded down. Any call to ttest() will run ttest.default(), with user specified values in place of the appropriate defaults.
...	only used in the generic S3 class.

Details

Missing values must be given by "NA"s to be recognized as missing values. Any call to ttest() is run by ttest.default(), with user specified values in place of defaults in the appropriate places.

Value

Prints a summary of the data and the corresponding t-test.

Variable	the variable name supplied to the t-test function
Group	the group name: either the variable names supplied to the function or the names of the strata if the variable by was specified.
Obs	Number of observations of each variable: includes missing values.
Missing	number of missing values in each data vector.
Mean	the sample mean of each data vector; also, the estimated difference in means in a two-sample test.
Std.Err.	the estimated standard error of the mean and of the difference in the two-sample test.
Std.Dev.	standard deviation estimates from the data.

CI	a confidence interval for the means, and for the difference in the two-sample test. This is at the confidence level specified in the argument. If prop and/or exact are TRUE, also returns the appropriate confidence interval for the test of equality of proportions.
Null hypothesis	a statement of the null hypothesis.
Alternative hypothesis	a statement of the alternative hypothesis.
t	value of the t-statistic.
df	the degrees of freedom for the test.
Pr	a p-value for inference on the corresponding hypothesis test.

Author(s)

Scott S. Emerson, M.D., Ph.D., Andrew J. Spieker, Brian D. Williamson

See Also

t.test

Examples

```
#- Read in data set -#
psa <- read.table("http://www.emersonstatistics.com/datasets/psa.txt", header=TRUE)
attach(psa)

#- Perform t-test -#
ttest(pretxpsa, null.hypoth = 100, test.type = "greater", more.digits = 1)

#- Define new binary variable as indicator -#
#- of whether or not bss was worst possible -#
bssworst <- bss
bssworst[bss == 1] <- 0
bssworst[bss == 2] <- 0
bssworst[bss == 3] <- 1

#- Perform t-test allowing for unequal -#
#- variances between strata -#
ttest(pretxpsa, by = bssworst)

#- Perform matched t-test -#
ttest(pretxpsa, nadirpsa, matched = TRUE, conf.level = 99/100, more.digits = 1)
```

---

ttesti *T-test Given Descriptive Statistics with Improved Layout*

Description

Produces table of relevant descriptive statistics and inference for either one- or two-sample t-test. In the two-sample case, the user can specify whether or not equal variances should be presumed. Can also perform a test of equality of proportions, with the appropriate Wald or exact binomial based confidence intervals.

Usage

```
## Method with the appropriate defaults filled in
ttesti(obs, mean, sd, obs2=NA, mean2=NA, sd2=NA,
       null.hyp = 0, level=.95, alternative="two.sided",
       var.eq = FALSE, prop=FALSE, exact=FALSE)
```

Arguments

obs	number of observations for the first sample.
mean	the sample mean of the first sample.
sd	the sample standard deviation of the first sample.
obs2	number of observations for the second sample (this is optional).
mean2	if obs2 is supplied, then sample mean of the second sample must be supplied.
sd2	if obs2 is supplied, then sample standard deviation of the second sample must be supplied.
null.hyp	a number specifying the null hypothesis for the mean (or difference in means if performing a two-sample test). Defaults to zero.
alternative	a string: one of "less", "two.sided", or "greater" specifying the form of the test. Defaults to a two-sided test.
level	confidence level of the test. Defaults to 95/100.
var.eq	a logical value, either TRUE or FALSE (default), specifying whether or not equal variances should be presumed in a two-sample t-test. Also controls robust standard errors.
prop	if TRUE, performs a test of equality of proportions with Wald based confidence intervals.
exact	must be FALSE if prop=FALSE. If true, performs a test of equality of proportions with Exact Binomial based confidence intervals.

Details

Values must be placed in the specified spaces, in place of the defaults. If obs2, mean2, or sd2 is specified, then all three must be specified and a two-sample t-test is run.

Value

Prints a summary of the data and the corresponding t-test.

Variable	x in a one-sample test, or x and y in a two sample test. The first set of descriptives entered goes to x.
Obs	Number of observations of each variable: includes missing values.
Mean	the sample mean; also, the estimated difference in means in a two-sample test.
Std.Err.	the estimated standard error of the mean and of the difference in the two-sample test.
Std.Dev.	standard deviation estimates.
CI	a confidence interval for the means, and for the difference in the two-sample test. This is at the confidence level specified in the argument. If prop and/or exact are specified, also returns the appropriate Wald or Exact Binomial based confidence interval.

`Null hypothesis`
: a statement of the null hypothesis.

`Alternative hypothesis`
: a statement of the alternative hypothesis.

`t`
: value of the t-statistic.

`df`
: the degrees of freedom for the test.

`Pr`
: a p-value for inference on the corresponding hypothesis test.

Author(s)

Scott S. Emerson, M.D., Ph.D., Andrew J. Spieker, Brian D. Williamson

Examples

```
#- T-test given sample descriptives -#
ttesti(24, 175, 35, null.hyp=230)

#- Two-sample test -#
ttesti(10, -1.6, 1.5, 30, -.7, 2.1)
```

U *Create a Transformed Variable*

Description

Creates a transformed variable using either the natural log, a dummy transformation, linear splines, or a polynomial. Mostly for use in regression. If a partial formula of the form `~var1 + var2` is entered, returns the formula for use in regression. The partial formula can be named by adding an equals sign before the tilde.

Usage

```
U(..., type=NULL, subset=rep(T,length(x)), knots=NULL, degree=2, reference=sort(unique(x[!is.na(x
            lbl=NULL, center=mean(x,na.rm=T), includeAll=FALSE, parameterization="absolute", vrsn
```

Arguments

`...`
: variable(s) used to create the transformation.

`type`
: a character string describing the transformation. Partial matching is used, so only enough of the string to make the transformation unique is needed.

`subset`
: used in creating dummy variables. Only used if `type == "dummy"`.

`knots`
: vector of knots to create the splines. Only used if `type=="lspline"`.

`degree`
: the degree of the polynomial to be returned. Only used if `type=="polynomial"`.

`reference`
: the reference vector for levels of the dummy variable. Only used if `type=="dummy"`.

`lbl`
: a label for the splines. Only used if `type=="lspline"`

`center`
: the center of the returned polynomial. Only used if `type=="polynomial"`.

`includeAll`
: a logical value to use all values even in the presense of a subset. Only used if `type=="dummy"`.

parameterization
: defaults to"absolute", and provides splines based on the absolute slope between knots. If "change", provides splines based on the change from knot to knot. If lsplineD is called, "change" is entered by default. Only used if type=="lspline".

vrsn
: if TRUE, returns the version of the function and nothing else.

Value

A matrix or vector containing the transformations. The class of the returned value is c("transformation", y) where y is the class of the transformed variable (usually numeric). The type of transformation performed is encoded as one of the attributes of the returned value, along with the original data.

Author(s)

Scott S. Emerson, M.D., Ph.D., Andrew J. Spieker, Brian D. Williamson

See Also

regress

Examples

```
# Reading in a dataset
mri <- read.table("http://www.emersonstatistics.com/datasets/mri.txt", header=TRUE)
attach(mri)
# Create a spline based on absolute
U(ldl, type="lspline", knots=c(70, 100, 130, 160))
U(ldl, type="ls", knots=c(70,100,130,160))

# Create a spline based on change
U(ldl, type="ls", knots=c(70, 100, 130, 160), parameterization="change")

# Create a log transformed variable
U(age, type="log")

## Create a partial formula
U(ma=~male+age)
```

---

uResiduals *Extract Residuals from uRegress objects*

Description

Extracts residuals (unstandardized, standardized, studentized, or jackknife) from uRegress objects.

Usage

```
uResiduals(object, type="", version=FALSE)
```

Arguments

object	an object of class uRegress, as returned by regress.
type	denotes the type of residuals to return. Default value is "", which returns unstandardized residuals. "standardized", "studentized", and "jackknife" return the expected type of residuals.
version	if TRUE, the version of the function will be returned. No other computations will be performed.

Details

Relies on functionality from the stats package to return residuals from the uRegress object. "studentized" residuals are computed as internally studentized residuals, while "jackknife" computes the externally studentized residuals.

Value

Returns the type of residuals requested.

Author(s)

Scott S. Emerson, M.D., Ph.D., Andrew J. Spieker, Brian D. Williamson

See Also

regress, rstudent, rstandard

Examples

```
# Load required libraries
library(survival)

# Reading in a dataset
mri <- read.table("http://www.emersonstatistics.com/datasets/mri.txt",header=TRUE)

# Create a uRegress object, regressing ldl on age
ldlReg <- regress("mean", age~ldl, data=mri)

# Get the studentized residuals
uResiduals(ldlReg, "studentized")

# Get the jackknifed residuals
uResiduals(ldlReg, "jackknife")
```

---

wilcoxon *Wilcoxon Signed Rank and Mann-Whitney-Wilcoxon Rank Sum Test*

Description

Performs Wilcoxon signed rank test or Mann-Whitney-Wilcoxon rank sum test depending on data and logicals entered. Relies heavily on the function wilcox.test. Adds formatting and variances, and prints the z-score and p-value in addition to the test statistic and p-value.

Usage

```
## S3 method for class 'wilcoxon'
wilcoxon(y, x=NULL, alternative="two.sided", mu=0, paired=FALSE,
        exact=NULL, correct=FALSE, conf.int=FALSE, conf.level=0.95)
```

Arguments

y	numeric vector of data values. Non-finite (missing or infinite) values will be omitted.
x	optional numeric vector of data values. Non-finite (missing or infinite) values will be omitted.
alternative	specifies the alternative hypothesis for the test; acceptable values are "two.sided", "greater", or "less".
mu	the value of the null hypothesis.
paired	logical indicating whether the data are paired or not. Default is FALSE. If TRUE, data must be the same length.
exact	logical value indicating whether or not an exact test should be computed.
correct	logical indicating whether or not a continuity correction should be used and displayed.
conf.int	logical indicating whether or not to calculate and display a 'confidence interval' (performs a semi-parametric test on medians, and is non-robust) and point estimate.
conf.level	confidence level for the interval.
...	only used in the generic S3 class.

Details

See above.

Value

A list with class "wilcoxon" is returned. The print method lays out the information in an easy to read format.

statistic	the value of the test statistic with a name describing it.
parameter	the parameter(s) for the exact distribution of the test statistic.
p.value	the p-value for the test (calculated for the test statistic).
null.value	the parameter mu.
alternative	character string describing the alternative hypothesis.
method	the type of test applied.
data.name	a character string giving the names of the data.
conf.int	a confidence interval for the location parameter (only present if the argument conf.int=TRUE).
estimate	an estimate of the location parameter (only present if the argument conf.int=TRUE).
table	a formatted table of rank sum and number of observation values, for printing.
vars	a formatted table of variances, for printing.
hyps	a formatted table of the hypotheses, for printing.
inf	a formatted table of inference values, for printing.

Author(s)

Scott S. Emerson, M.D., Ph.D., Andrew J. Spieker, Brian D. Williamson

See Also

wilcox.test

Examples

```
#- Create the data -#
cf <- c(1153, 1132, 1165, 1460, 1162, 1493, 1358, 1453, 1185, 1824, 1793, 1930, 2075)
healthy <- c(996, 1080, 1182, 1452, 1634, 1619, 1140, 1123, 1113, 1463, 1632, 1614, 1836)

#- Perform the test -#
wilcoxon(cf, healthy, paired=TRUE)

#- Perform the test -#
wilcoxon(cf, healthy, conf.int=TRUE)
```

Index