

Package ‘tableone’

July 2, 2014

Type Package

Title Create “Table 1” to describe baseline characteristics

Version 0.6.2

Date 2014-06-01

Author Kazuki Yoshida, Justin Bohn.

Maintainer Kazuki Yoshida <kazukiyoshida@mail.harvard.edu>

Description This package creates “Table 1”, i.e., description of baseline patient characteristics, which is essential in every medical research. This package provides functions to create such summaries for continuous and categorical variables, optionally with subgroup comparisons. The package was inspired by and based on descriptive statistics functions in Deducer, a Java-based GUI package by Ian Fellows. This package does not require GUI or Java, and intended for command-line users.

License GPL-2

Imports e1071, gmodels

Suggests survival, testthat

URL <https://github.com/kaz-yos/tableone>

NeedsCompilation no

Repository CRAN

Date/Publication 2014-06-02 14:41:30

R topics documented:

tableone-package	2
CreateCatTable	4
CreateContTable	6
CreateTableOne	8

print.CatTable	11
print.ContTable	14
print.TableOne	16
ShowRegTable	18
summary.CatTable	20
summary.ContTable	21
summary.TableOne	22

Index	24
--------------	-----------

tableone-package	<i>Create "Table 1" to describe baseline characteristics</i>
------------------	--------------------------------------------------------------

Description

This package creates "Table 1", i.e., description of baseline patient characteristics, which is essential in every medical research. This package provides functions to create such summaries for continuous and categorical variables, optionally with subgroup comparisons. The package was inspired by and based on descriptive statistics functions in Deducer, a Java-based GUI package by Ian Fellows. This package does not require GUI or Java, and intended for command-line users.

Note

Special Thanks:

Ian Fellows for developing the Deducer package, which this package is based on.

Hadley Wickham for packaging advice and for creating tools this package was made with (roxygen2, devtools, testthat).

Yoshinobu Kanda for design advice and for integration into RcmdrPlugin.EZR.

jomuller for a bug report and fix suggestion regarding handling of ordered factors.

Raja Sriswan Mamidi for suggestions regarding alignment.

Members of the Facebook Organization of R Users for Medical Statistics in Japan (FORUMS-J) for testing pre-release versions and suggestions.

Developmental repository is on github. Your contributions are appreciated.

<https://github.com/kaz-yos/tableone>

Author(s)

Kazuki Yoshida, Justin Bohn

Maintainer: Kazuki Yoshida <kazukiyoshida@mail.harvard.edu>

See Also

[CreateTableOne](#), [print.TableOne](#), [summary.TableOne](#), [CreateContTable](#), [print.ContTable](#), [summary.ContTable](#), [CreateCatTable](#), [print.CatTable](#), [summary.CatTable](#), [ShowRegTable](#)

Examples

```
## Load
library(tableone)

## Load Mayo Clinic Primary Biliary Cirrhosis Data
library(survival)
data(pbc)
## Check variables
head(pbc)

## List numerically coded categorical variables for later conversion.
## Factor variables are automatically handled as categorical variables.
factorVars <- c("status", "trt", "ascites", "hepato", "spiders", "edema", "stage")

## Create a variable list
dput(names(pbc)) # This shows a character vector-creating syntax.
vars <- c("time", "status", "age", "sex", "ascites", "hepato",
          "spiders", "edema", "bili", "chol", "albumin",
          "copper", "alk.phos", "ast", "trig", "platelet",
          "protime", "stage")

## Create Table 1 stratified by trt. Use factorVars to convert numerically
## coded categorical variables as factors without changing the dataset.
tableOne <- CreateTableOne(vars = vars, strata = c("trt"), data = pbc,
                           factorVars = factorVars)

## Just typing the object name will invoke the print.TableOne method
tableOne

## Specifying nonnormal variables will show the variables appropriately,
## and show nonparametric test p-values. Specify variables in the exact
## argument to obtain the exact test p-values. For two-level categorical
## variables specified in cramVars, both levels are shown. Use minMax
## argument to show median [min, max] for nonnormal variables.
print(tableOne, nonnormal = c("bili", "chol", "copper", "alk.phos", "trig"),
      exact = c("status", "stage"), cramVars = "sex")

## Use the summary.TableOne method for detailed summary
summary(tableOne)

## See the categorical part only using $ operator
tableOne$CatTable
summary(tableOne$CatTable)

## See the continuous part only using $ operator
tableOne$ContTable
summary(tableOne$ContTable)

## If your work flow includes copying to Excel and Word when writing manuscripts,
## you may benefit from the quote argument. This will quote everything so that
## Excel does not mess up the cells.
print(tableOne, nonnormal = c("bili", "chol", "copper", "alk.phos", "trig"),
```

```

exact = c("status", "stage"), cramVars = "sex", quote = TRUE)

## If you want to center-align values in Word, use noSpaces option.
print(tableOne, nonnormal = c("bili", "chol", "copper", "alk.phos", "trig"),
      exact = c("status", "stage"), cramVars = "sex", quote = TRUE, noSpaces = TRUE)

```

CreateCatTable *Create an object summarizing categorical variables*

Description

Create an object summarizing categorical variables optionally stratifying by one or more stratifying variables and performing statistical tests. The object gives a table that is easy to use in medical research papers. See also [print.CatTable](#) and [summary.CatTable](#).

Usage

```

CreateCatTable(vars, strata, data, test = TRUE, testApprox = chisq.test,
  argsApprox = list(correct = TRUE), testExact = fisher.test,
  argsExact = list(workspace = 2 * 10^5))

```

Arguments

vars	Variable(s) to be summarized given as a character vector.
strata	Stratifying (grouping) variable name(s) given as a character vector. If omitted, the overall results are returned.
data	A data frame in which these variables exist. All variables (both vars and strata) must be in this data frame.
test	If TRUE, as in the default and there are more than two groups, groupwise comparisons are performed. Both tests that require the large sample approximation and exact tests are performed. Either one of the result can be obtained from the print method.
testApprox	A function used to perform the large sample approximation based tests. The default is chisq.test . This is not recommended when some of the cell have small counts like fewer than 5.
argsApprox	A named list of arguments passed to the function specified in testApprox. The default is <code>list(correct = TRUE)</code> , which turns on the continuity correction for chisq.test .
testExact	A function used to perform the exact tests. The default is <code>fisher.test</code> . If the cells have large numbers, it will fail because of memory limitation. In this situation, the large sample approximation based should suffice.
argsExact	A named list of arguments passed to the function specified in testExact. The default is <code>list(workspace = 2*10^5)</code> , which specifies the memory space allocated for fisher.test .

Value

An object of class `CatTable`, which really is a `by` object with additional attributes. Each element of the `by` part is a matrix with rows representing variables, and columns representing summary statistics.

Author(s)

Kazuki Yoshida (based on `Deducer::frequencies()`)

See Also

[CreateCatTable](#), [print.CatTable](#), [summary.CatTable](#), [CreateContTable](#), [print.ContTable](#), [summary.ContTable](#), [CreateTableOne](#), [print.TableOne](#), [summary.TableOne](#)

Examples

```
## Load
library(tableone)

## Load Mayo Clinic Primary Biliary Cirrhosis Data
library(survival)
data(pbc)
## Check variables
head(pbc)

## Create an overall table for categorical variables
catVars <- c("status", "ascites", "hepato", "spiders", "edema", "stage")
catTableOverall <- CreateCatTable(vars = catVars, data = pbc)

## Simply typing the object name will invoke the print.CatTable method,
## which will show the sample size, frequencies and percentages.
## For 2-level variables, only the higher level is shown for simplicity
## unless the variables are specified in the cramVars argument.
catTableOverall

## If you need to show both levels for some 2-level factors, use cramVars
print(catTableOverall, cramVars = "hepato")

## Use the showAllLevels argument to see all levels for all variables.
print(catTableOverall, showAllLevels = TRUE)

## You can choose form frequencies ("f") and/or percentages ("p") or both.
## "fp" frequency (percentage) is the default. Row names change accordingly.
print(catTableOverall, format = "f")
print(catTableOverall, format = "p")

## To further examine the variables, use the summary.CatTable method,
## which will show more details.
summary(catTableOverall)

## The table can be stratified by one or more variables
catTableBySexTrt <- CreateCatTable(vars = catVars,
```

```

strata = c("sex","trt"), data = pbc)

## print now includes p-values which are by default calculated by chisq.test.
## It is formatted at the decimal place specified by the pDigits argument
## (3 by default). It does <0.001 for you.
catTableBySexTrt

## The exact argument toggles the p-values to the exact test result from
## fisher.test. It will show which ones are from exact tests.
print(catTableBySexTrt, exact = "ascites")

## summary now includes both types of p-values
summary(catTableBySexTrt)

## If your work flow includes copying to Excel and Word when writing manuscripts,
## you may benefit from the quote argument. This will quote everything so that
## Excel does not mess up the cells.
print(catTableBySexTrt, exact = "ascites", quote = TRUE)

## If you want to center-align values in Word, use noSpaces option.
print(catTableBySexTrt, exact = "ascites", quote = TRUE, noSpaces = TRUE)

```

CreateContTable

Create an object summarizing continuous variables

Description

Create an object summarizing continuous variables optionally stratifying by one or more stratifying variables and performing statistical tests. The object gives a table that is easy to use in medical research papers. See also [print.ContTable](#) and [summary.ContTable](#).

Usage

```

CreateContTable(vars, strata, data, funcNames = c("n", "miss", "p.miss",
  "mean", "sd", "median", "p25", "p75", "min", "max", "skew", "kurt"),
  funcAdditional, test = TRUE, testNormal = oneway.test,
  argsNormal = list(var.equal = TRUE), testNonNormal = kruskal.test,
  argsNonNormal = list(NULL))

```

Arguments

vars	Variable(s) to be summarized given as a character vector.
strata	Stratifying (grouping) variable name(s) given as a character vector. If omitted, the overall results are returned.
data	A data frame in which these variables exist. All variables (both vars and strata) must be in this data frame.

funcNames	The functions to give the group size, number with missing values, proportion with missing values, mean, standard deviations, median, 25th percentile, 75th percentile, minimum, maximum, skewness (same definition as in SAS), kurtosis (same definition as in SAS). All of them can be seen in the summary method output. The print method uses subset of these. You can choose subset of them or reorder them. They are all configure to omit NA values (<code>na.rm = TRUE</code>).
funcAdditional	Additional functions can be given as a named list. For example, <code>list(sum = sum)</code> .
test	If TRUE, as in the default and there are more than two groups, groupwise comparisons are performed. Both tests that assume normality and tests that do not are performed. Either one of the result can be obtained from the print method.
testNormal	A function used to perform the normal assumption based tests. The default is oneway.test . This is equivalent of the t-test when there are only two groups.
argsNormal	A named list of arguments passed to the function specified in <code>testNormal</code> . The default is <code>list(var.equal = TRUE)</code> , which makes it the ordinary ANOVA that assumes equal variance across groups.
testNonNormal	A function used to perform the nonparametric tests. The default is <code>kruskal.test</code> (Kruskal-Wallis rank sum test). This is equivalent of the <code>wilcox.test</code> (Mann-Whitney U test) when there are only two groups.
argsNonNormal	A named list of arguments passed to the function specified in <code>testNonNormal</code> . The default is <code>list(NULL)</code> , which is just a placeholder.

Value

An object of class `ContTable`, which really is a `by` object with additional attributes. Each element of the `by` part is a matrix with rows representing variables, and columns representing summary statistics.

Author(s)

Kazuki Yoshida (based on `Deducer::descriptive.table()`)

See Also

[CreateContTable](#), [print.ContTable](#), [summary.ContTable](#), [CreateCatTable](#), [print.CatTable](#), [summary.CatTable](#), [CreateTableOne](#), [print.TableOne](#), [summary.TableOne](#)

Examples

```
## Load
library(tableone)

## Load Mayo Clinic Primary Biliary Cirrhosis Data
library(survival)
data(pbc)
## Check variables
head(pbc)

## Create an overall table for continuous variables
```

```

contVars <- c("time", "age", "bili", "chol", "albumin", "copper",
             "alk.phos", "ast", "trig", "platelet", "protime")
contTableOverall <- CreateContTable(vars = contVars, data = pbc)

## Simply typing the object name will invoke the print.ContTable method,
## which will show the sample size, means and standard deviations.
contTableOverall

## To further examine the variables, use the summary.ContTable method,
## which will show more details.
summary(contTableOverall)

## c("age", "chol", "copper", "alk.phos", "trig", "protime") appear highly skewed.
## Specify them in the nonnormal argument, and the display changes to the median,
## and the [25th, 75th] percentile.
nonNormalVars <- c("age", "chol", "copper", "alk.phos", "trig", "protime")
print(contTableOverall, nonnormal = nonNormalVars)

## To show median [min,max] for nonnormal variables, use minMax = TRUE
print(contTableOverall, nonnormal = nonNormalVars, minMax = TRUE)

## The table can be stratified by one or more variables
contTableBySexTrt <- CreateContTable(vars = contVars,
                                   strata = c("sex", "trt"), data = pbc)

## print now includes p-values which are by default calculated by oneway.test (t-test
## equivalent in the two group case). It is formatted at the decimal place specified
## by the pDigits argument (3 by default). It does <0.001 for you.
contTableBySexTrt

## The nonnormal argument toggles the p-values to the nonparametric result from
## kruskal.test (wilcox.test equivalent for the two group case).
print(contTableBySexTrt, nonnormal = nonNormalVars)

## summary now includes both types of p-values
summary(contTableBySexTrt)

## If your work flow includes copying to Excel and Word when writing manuscripts,
## you may benefit from the quote argument. This will quote everything so that
## Excel does not mess up the cells.
print(contTableBySexTrt, nonnormal = nonNormalVars, quote = TRUE)

## If you want to center-align values in Word, use noSpaces option.
print(contTableBySexTrt, nonnormal = nonNormalVars, quote = TRUE, noSpaces = TRUE)

```


Description

Create an object summarizing all baseline variables optionally stratifying by one or more stratifying variables and performing statistical tests. The object gives a table that is easy to use in medical research papers. See also [print.TableOne](#) and [summary.TableOne](#).

Usage

```
CreateTableOne(vars, strata, data, factorVars, test = TRUE,
  testApprox = chisq.test, argsApprox = list(correct = TRUE),
  testExact = fisher.test, argsExact = list(workspace = 2 * 10^5),
  testNormal = oneway.test, argsNormal = list(var.equal = TRUE),
  testNonNormal = kruskal.test, argsNonNormal = list(NULL))
```

Arguments

<code>vars</code>	Variables to be summarized given as a character vector. Factors are handled as categorical variables, whereas numeric variables are handled as continuous variables. If empty, all variables in the data frame specified in the data argument are used.
<code>strata</code>	Stratifying (grouping) variable name(s) given as a character vector. If omitted, the overall results are returned.
<code>data</code>	A data frame in which these variables exist. All variables (both vars and strata) must be in this data frame.
<code>factorVars</code>	Numerically coded variables that should be handled as categorical variables given as a character vector. If omitted, only factors are considered categorical variables. If all categorical variables in the dataset are already factors, this option is not necessary. The variables specified here must also be specified in the vars argument.
<code>test</code>	If TRUE, as in the default and there are more than two groups, groupwise comparisons are performed.
<code>testNormal</code>	A function used to perform the normal assumption based tests. The default is oneway.test . This is equivalent of the t-test when there are only two groups.
<code>argsNormal</code>	A named list of arguments passed to the function specified in <code>testNormal</code> . The default is <code>list(var.equal = TRUE)</code> , which makes it the ordinary ANOVA that assumes equal variance across groups.
<code>testNonNormal</code>	A function used to perform the nonparametric tests. The default is <code>kruskal.test</code> (Kruskal-Wallis Rank Sum Test). This is equivalent of the <code>wilcox.test</code> (Mann-Whitney U test) when there are only two groups.
<code>argsNonNormal</code>	A named list of arguments passed to the function specified in <code>testNonNormal</code> . The default is <code>list(NULL)</code> , which is just a placeholder.
<code>testApprox</code>	A function used to perform the large sample approximation based tests. The default is chisq.test . This is not recommended when some of the cell have small counts like fewer than 5.
<code>argsApprox</code>	A named list of arguments passed to the function specified in <code>testApprox</code> . The default is <code>list(correct = TRUE)</code> , which turns on the continuity correction for chisq.test .

testExact	A function used to perform the exact tests. The default is <code>fisher.test</code> . If the cells have large numbers, it will fail because of memory limitation. In this situation, the large sample approximation based should suffice.
argsExact	A named list of arguments passed to the function specified in <code>testExact</code> . The default is <code>list(workspace = 2*10^5)</code> , which specifies the memory space allocated for <code>fisher.test</code> .

Value

An object of class `TableOne`, which really is a list of three objects.

<code>TableOne</code>	a categorical-continuous mixture data formatted and printed by the <code>print.TableOne</code> method
<code>ContTable</code>	an object of class <code>ContTable</code> , containing continuous variables only
<code>CatTable</code>	an object of class <code>CatTable</code> , containing categorical variables only

The second and third objects can be then be examined with the `print` and `summary` method, for example, `summary(object$CatTable)` to examine the categorical variables in detail.

Author(s)

Justin Bohn, Kazuki Yoshida

See Also

[CreateTableOne](#), [print.TableOne](#), [summary.TableOne](#), [CreateCatTable](#), [print.CatTable](#), [summary.CatTable](#), [CreateContTable](#), [print.ContTable](#), [summary.ContTable](#)

Examples

```
## Load
library(tableone)

## Load Mayo Clinic Primary Biliary Cirrhosis Data
library(survival)
data(pbc)
## Check variables
head(pbc)

## Make categorical variables factors
varsToFactor <- c("status", "trt", "ascites", "hepato", "spiders", "edema", "stage")
pbc[varsToFactor] <- lapply(pbc[varsToFactor], factor)

## Create a variable list
dput(names(pbc))
vars <- c("time", "status", "age", "sex", "ascites", "hepato",
         "spiders", "edema", "bili", "chol", "albumin",
         "copper", "alk.phos", "ast", "trig", "platelet",
         "protime", "stage")

## Create Table 1 stratified by trt
```

```

tableOne <- CreateTableOne(vars = vars, strata = c("trt"), data = pbc)

## Just typing the object name will invoke the print.TableOne method
tableOne

## Specifying nonnormal variables will show the variables appropriately,
## and show nonparametric test p-values. Specify variables in the exact
## argument to obtain the exact test p-values.
print(tableOne, nonnormal = c("bili", "chol", "copper", "alk.phos", "trig"),
      exact = c("status", "stage"))

## Use the summary.TableOne method for detailed summary
summary(tableOne)

## See the categorical part only using $ operator
tableOne$CatTable
summary(tableOne$CatTable)

## See the continuous part only using $ operator
tableOne$ContTable
summary(tableOne$ContTable)

## If your work flow includes copying to Excel and Word when writing manuscripts,
## you may benefit from the quote argument. This will quote everything so that
## Excel does not mess up the cells.
print(tableOne, nonnormal = c("bili", "chol", "copper", "alk.phos", "trig"),
      exact = c("status", "stage"), quote = TRUE)

## If you want to center-align values in Word, use noSpaces option.
print(tableOne, nonnormal = c("bili", "chol", "copper", "alk.phos", "trig"),
      exact = c("status", "stage"), quote = TRUE, noSpaces = TRUE)

```

print.CatTable

Format and print the CatTable class objects

Description

This is the print method for the CatTable class objects created by [CreateCatTable](#) function.

Usage

```

## S3 method for class 'CatTable'
print(x, digits = 1, pDigits = 3, quote = FALSE,
      missing = FALSE, explain = TRUE, printToggle = TRUE, noSpaces = FALSE,
      format = c("fp", "f", "p", "pf")[1], showAllLevels = FALSE,
      cramVars = NULL, test = TRUE, exact = NULL, CrossTable = FALSE, ...)

```

Arguments

<code>x</code>	The result of a call to the CreateCatTable function.
<code>digits</code>	Number of digits to print in the table.
<code>pDigits</code>	Number of digits to print for p-values.
<code>quote</code>	Whether to show everything in quotes. The default is FALSE. If TRUE, everything including the row and column names are quoted so that you can copy it to Excel easily.
<code>missing</code>	Whether to show missing data information (not implemented yet, placeholder)
<code>explain</code>	Whether to add explanation to the variable names, i.e., (%) is added to the variable names when percentage is shown.
<code>printToggle</code>	Whether to print the output. If FALSE, no output is created, and a matrix is invisibly returned.
<code>noSpaces</code>	Whether to remove spaces added for alignment. Use this option if you prefer to align numbers yourself in other software.
<code>format</code>	The default is "fp" frequency (percentage). You can also choose from "f" frequency only, "p" percentage only, and "pf" percentage (frequency).
<code>showAllLevels</code>	Whether to show all levels. FALSE by default, i.e., for 2-level categorical variables, only the higher level is shown to avoid redundant information.
<code>cramVars</code>	A character vector to specify the two-level categorical variables, for which both levels should be shown in one row.
<code>test</code>	Whether to show the p-values. TRUE by default. If FALSE, only the numerical summaries are shown.
<code>exact</code>	A character vector to specify the variables for which the p-values should be those of exact tests. By default all p-values are from large sample approximation tests (<code>chisq.test</code>).
<code>CrossTable</code>	Whether to show the cross table objects held internally using <code>gmodels::CrossTable</code> function. This will give an output similar to the PROC FREQ in SAS.
<code>...</code>	For compatibility with generic. Ignored.

Value

It is mainly for printing the result. But this function does return a matrix containing what you see in the output invisibly. You can assign it to an object to save it.

Author(s)

Kazuki Yoshida

See Also

[CreateCatTable](#), [print.CatTable](#), [summary.CatTable](#), [CreateContTable](#), [print.ContTable](#), [summary.ContTable](#), [CreateTableOne](#), [print.TableOne](#), [summary.TableOne](#)

Examples

```
## Load
library(tableone)

## Load Mayo Clinic Primary Biliary Cirrhosis Data
library(survival)
data(pbc)
## Check variables
head(pbc)

## Create an overall table for categorical variables
catVars <- c("status", "ascites", "hepato", "spiders", "edema", "stage")
catTableOverall <- CreateCatTable(vars = catVars, data = pbc)

## Simply typing the object name will invoke the print.CatTable method,
## which will show the sample size, frequencies and percentages.
## For 2-level variables, only the higher level is shown for simplicity.
catTableOverall

## If you need to show both levels for some 2-level factors, use cramVars
print(catTableOverall, cramVars = "hepato")

## Use the showAllLevels argument to see all levels for all variables.
print(catTableOverall, showAllLevels = TRUE)

## You can choose form frequencies ("f") and/or percentages ("p") or both.
## "fp" frequency (percentage) is the default. Row names change accordingly.
print(catTableOverall, format = "f")
print(catTableOverall, format = "p")

## To further examine the variables, use the summary.CatTable method,
## which will show more details.
summary(catTableOverall)

## The table can be stratified by one or more variables
catTableBySexTrt <- CreateCatTable(vars = catVars,
                                  strata = c("sex", "trt"), data = pbc)

## print now includes p-values which are by default calculated by chisq.test.
## It is formatted at the decimal place specified by the pDigits argument
## (3 by default). It does <0.001 for you.
catTableBySexTrt

## The exact argument toggles the p-values to the exact test result from
## fisher.test. It will show which ones are from exact tests.
print(catTableBySexTrt, exact = "ascites")

## summary now includes both types of p-values
summary(catTableBySexTrt)

## If your work flow includes copying to Excel and Word when writing manuscripts,
## you may benefit from the quote argument. This will quote everything so that
```

```
## Excel does not mess up the cells.
print(catTableBySexTrt, exact = "ascites", quote = TRUE)

## If you want to center-align values in Word, use noSpaces option.
print(catTableBySexTrt, exact = "ascites", quote = TRUE, noSpaces = TRUE)
```

print.ContTable *Format and print the ContTable class objects*

Description

This is the print method for the ContTable class objects created by [CreateContTable](#) function.

Usage

```
## S3 method for class 'ContTable'
print(x, digits = 2, pDigits = 3, quote = FALSE,
      missing = FALSE, explain = TRUE, printToggle = TRUE, noSpaces = FALSE,
      nonnormal = NULL, minMax = FALSE, insertLevel = FALSE, test = TRUE,
      ...)
```

Arguments

x	The result of a call to the CreateContTable function.
digits	Number of digits to print in the table.
pDigits	Number of digits to print for p-values.
quote	Whether to show everything in quotes. The default is FALSE. If TRUE, everything including the row and column names are quoted so that you can copy it to Excel easily.
missing	Whether to show missing data information (not implemented yet, placeholder)
explain	Whether to add explanation to the variable names, i.e., (mean (sd) or median [IQR]) is added to the variable names.
printToggle	Whether to print the output. If FALSE, no output is created, and a matrix is invisibly returned.
noSpaces	Whether to remove spaces added for alignment. Use this option if you prefer to align numbers yourself in other software.
nonnormal	A character vector to specify the variables for which the p-values should be those of nonparametric tests. By default all p-values are from normal assumption-based tests (oneway.test).
minMax	Whether to use [min,max] instead of [p25,p75] for nonnormal variables. The default is FALSE.
insertLevel	Whether to add an empty level column to the left of strata.
test	Whether to show the p-values. TRUE by default. If FALSE, only the numerical summaries are shown.
...	For compatibility with generic. Ignored.

Value

It is mainly for printing the result. But this function does return a matrix containing what you see in the output invisibly. You can assign it to an object to save it.

Author(s)

Kazuki Yoshida

See Also

[CreateCatTable](#), [print.CatTable](#), [summary.CatTable](#), [CreateContTable](#), [print.ContTable](#), [summary.ContTable](#), [CreateTableOne](#), [print.TableOne](#), [summary.TableOne](#)

Examples

```
## Load
library(tableone)

## Load Mayo Clinic Primary Biliary Cirrhosis Data
library(survival)
data(pbc)
## Check variables
head(pbc)

## Create an overall table for continuous variables
contVars <- c("time", "age", "bili", "chol", "albumin", "copper",
             "alk.phos", "ast", "trig", "platelet", "protime")
contTableOverall <- CreateContTable(vars = contVars, data = pbc)

## Simply typing the object name will invoke the print.ContTable method,
## which will show the sample size, means and standard deviations.
contTableOverall

## To further examine the variables, use the summary.ContTable method,
## which will show more details.
summary(contTableOverall)

## c("age", "chol", "copper", "alk.phos", "trig", "protime") appear highly skewed.
## Specify them in the nonnormal argument, and the display changes to the median,
## and the [25th, 75th] percentile.
nonNormalVars <- c("age", "chol", "copper", "alk.phos", "trig", "protime")
print(contTableOverall, nonnormal = nonNormalVars)

## To show median [min,max] for nonnormal variables, use minMax = TRUE
print(contTableOverall, nonnormal = nonNormalVars, minMax = TRUE)

## The table can be stratified by one or more variables
contTableBySexTrt <- CreateContTable(vars = contVars,
                                   strata = c("sex", "trt"), data = pbc)

## print now includes p-values which are by default calculated by oneway.test (t-test
## equivalent in the two group case). It is formatted at the decimal place specified
```

```

## by the pDigits argument (3 by default). It does <0.001 for you.
contTableBySexTrt

## The nonnormal argument toggles the p-values to the nonparametric result from
## kruskal.test (wilcox.test equivalent for the two group case).
print(contTableBySexTrt, nonnormal = nonNormalVars)

## The minMax argument toggles whether to show median [range]
print(contTableBySexTrt, nonnormal = nonNormalVars, minMax = TRUE)

## summary now includes both types of p-values
summary(contTableBySexTrt)

## If your work flow includes copying to Excel and Word when writing manuscripts,
## you may benefit from the quote argument. This will quote everything so that
## Excel does not mess up the cells.
print(contTableBySexTrt, nonnormal = nonNormalVars, quote = TRUE)

## If you want to center-align values in Word, use noSpaces option.
print(contTableBySexTrt, nonnormal = nonNormalVars, quote = TRUE, noSpaces = TRUE)

```

print.TableOne

Format and print the TableOne class objects

Description

This is the print method for the TableOne class objects created by [CreateTableOne](#) function.

Usage

```

## S3 method for class 'TableOne'
print(x, catDigits = 1, contDigits = 2, pDigits = 3,
      quote = FALSE, missing = FALSE, explain = TRUE, printToggle = TRUE,
      test = TRUE, noSpaces = FALSE, format = c("fp", "f", "p", "pf")[1],
      showAllLevels = FALSE, cramVars = NULL, exact = NULL,
      nonnormal = NULL, minMax = FALSE, ...)

```

Arguments

x	The result of a call to the CreateTableOne function.
catDigits	Number of digits to print for proportions. Default 1.
contDigits	Number of digits to print for continuous variables. Default 2.
pDigits	Number of digits to print for p-values. Default 3.
quote	Whether to show everything in quotes. The default is FALSE. If TRUE, everything including the row and column names are quoted so that you can copy it to Excel easily.
missing	Whether to show missing data information (not implemented yet, placeholder)

explain	Whether to add explanation to the variable names, i.e., (%) is added to the variable names when percentage is shown.
printToggle	Whether to print the output. If FALSE, no output is created, and a matrix is invisibly returned.
test	Whether to show the p-values. TRUE by default. If FALSE, only the numerical summaries are shown.
noSpaces	Whether to remove spaces added for alignment. Use this option if you prefer to align numbers yourself in other software.
format	The default is "fp" frequency (percentage). You can also choose from "f" frequency only, "p" percentage only, and "pf" percentage (frequency).
showAllLevels	Whether to show all levels. FALSE by default, i.e., for 2-level categorical variables, only the higher level is shown to avoid redundant information.
cramVars	A character vector to specify the two-level categorical variables, for which both levels should be shown in one row.
exact	A character vector to specify the variables for which the p-values should be those of exact tests. By default all p-values are from large sample approximation tests (chisq.test).
nonnormal	A character vector to specify the variables for which the p-values should be those of nonparametric tests. By default all p-values are from normal assumption-based tests (oneway.test).
minMax	Whether to use [min,max] instead of [p25,p75] for nonnormal variables. The default is FALSE.
...	For compatibility with generic. Ignored.

Value

It is mainly for printing the result. But this function does return a matrix containing what you see in the output invisibly. You can assign it to an object to save it.

Author(s)

Kazuki Yoshida, Justin Bohn

See Also

[CreateTableOne](#), [print.TableOne](#), [summary.TableOne](#), [CreateCatTable](#), [print.CatTable](#), [summary.CatTable](#), [CreateContTable](#), [print.ContTable](#), [summary.ContTable](#)

Examples

```
## Load
library(tableone)

## Load Mayo Clinic Primary Biliary Cirrhosis Data
library(survival)
data(pbc)
## Check variables
```

```

head(pbc)

## Make categorical variables factors
varsToFactor <- c("status","trt","ascites","hepato","spiders","edema","stage")
pbc[varsToFactor] <- lapply(pbc[varsToFactor], factor)

## Create Table 1 stratified by sex and trt
tableOne <- CreateTableOne(vars = c("time","status","age","ascites","hepato",
                                   "spiders","edema","bili","chol","albumin",
                                   "copper","alk.phos","ast","trig","platelet",
                                   "protime","stage"),
                           strata = c("sex","trt"), data = pbc)

## Just typing the object name will invoke the print.TableOne method
tableOne

## Specifying nonnormal variables will show the variables appropriately,
## and show nonparametric test p-values. Specify variables in the exact
## argument to obtain the exact test p-values. cramVars can be used to
## show both levels for a 2-level categorical variables.
print(tableOne, nonnormal = c("bili","chol","copper","alk.phos","trig"),
       exact = c("status","stage"), cramVars = "hepato")

## Use the summary.TableOne method for detailed summary
summary(tableOne)

## See the categorical part only using $ operator
tableOne$CatTable
summary(tableOne$CatTable)

## See the continuous part only using $ operator
tableOne$ContTable
summary(tableOne$ContTable)

## If your work flow includes copying to Excel and Word when writing manuscripts,
## you may benefit from the quote argument. This will quote everything so that
## Excel does not mess up the cells.
print(tableOne, nonnormal = c("bili","chol","copper","alk.phos","trig"),
       exact = c("status","stage"), cramVars = "hepato", quote = TRUE)

## If you want to center-align values in Word, use noSpaces option.
print(tableOne, nonnormal = c("bili","chol","copper","alk.phos","trig"),
       exact = c("status","stage"), cramVars = "hepato", quote = TRUE, noSpaces = TRUE)

```

ShowRegTable

Format regression results in medically decent format

Description

It shows the regression result in the HR [95% CI] p-value format, which is usually the form used in medical research papers.

Usage

```
ShowRegTable(model, exp = TRUE, digits = 2, pDigits = 3,  
  printToggle = TRUE, quote = FALSE)
```

Arguments

model	Regression model result objects that have the summary and confint methods.
exp	TRUE by default. You need to specify exp = FALSE if your model is has the identity link function (linear regression, etc).
digits	Number of digits to print for the main part.
pDigits	Number of digits to print for the p-values.
printToggle	Whether to print the output. If FALSE, no output is created, and a matrix is invisibly returned.
quote	Whether to show everything in quotes. The default is FALSE. If TRUE, everything including the row and column names are quoted so that you can copy it to Excel easily.

Value

A matrix containing what you see is returned invisibly. You can capture it by assignment to an object.

Author(s)

Kazuki Yoshida

Examples

```
## Load  
library(tableone)  
  
## Load Mayo Clinic Primary Biliary Cirrhosis Data  
library(survival)  
data(pbc)  
## Check variables  
head(pbc)  
  
## Fit a Cox regression model  
objCoxph <- coxph(formula = Surv(time, status == 2) ~ trt + age + albumin + ascites,  
  data = pbc)  
  
## Show the simple table  
ShowRegTable(objCoxph)  
  
## Show with quote to ease copy and paste  
ShowRegTable(objCoxph, quote = TRUE)
```

summary.CatTable	<i>Shows all results in a CatTable class object</i>
------------------	-----------------------------------------------------

Description

This method shows all the data a `CatTable` class object has. This includes the (optionally stratified) part with summary statistics and , if available, p-values from the approximation method test (`chisq.test` by default) and exact method test (`fisher.test` by default).

Usage

```
## S3 method for class 'CatTable'
summary(object, digits = 1, ...)
```

Arguments

<code>object</code>	An object that has the <code>CatTable</code> class to be shown.
<code>digits</code>	Number of digits to print.
<code>...</code>	For compatibility with generic. Ignored.

Value

It will print the results.

Author(s)

Kazuki Yoshida

See Also

[CreateCatTable](#), [print.CatTable](#), [summary.CatTable](#), [CreateContTable](#), [print.ContTable](#), [summary.ContTable](#), [CreateTableOne](#), [print.TableOne](#), [summary.TableOne](#)

Examples

```
## Load
library(tableone)

## Load Mayo Clinic Primary Biliary Cirrhosis Data
library(survival)
data(pbc)
## Check variables
head(pbc)

## Create an overall table for categorical variables
catVars <- c("status", "ascites", "hepato", "spiders", "edema", "stage")
catTableOverall <- CreateCatTable(vars = catVars, data = pbc)
```

```
## Simply typing the object name will invoke the print.CatTable method,  
## which will show the sample size, frequencies and percentages.  
## For 2-level variables, only the higher level is shown for simplicity.  
catTableOverall  
  
## To further examine the variables, use the summary.CatTable method,  
## which will show more details.  
summary(catTableOverall)
```

summary.ContTable *Shows all results in a ContTable class object*

Description

This method shows all the data a ContTable class object has. This includes the (optionally stratified) part with summary statistics and, if available, p-values from the normal assumption-based test ([oneway.test](#) by default) and nonparametric test ([kruskal.test](#) by default).

Usage

```
## S3 method for class 'ContTable'  
summary(object, digits = 2, ...)
```

Arguments

object	An object that has the ContTable class to be shown.
digits	Number of digits to print.
...	For compatibility with generic. Ignored.

Value

It will print the results.

Author(s)

Kazuki Yoshida

See Also

[CreateContTable](#), [print.ContTable](#), [summary.ContTable](#), [CreateCatTable](#), [print.CatTable](#), [summary.CatTable](#), [CreateTableOne](#), [print.TableOne](#), [summary.TableOne](#)

Examples

```
## Load
library(tableone)

## Load Mayo Clinic Primary Biliary Cirrhosis Data
library(survival)
data(pbc)
## Check variables
head(pbc)

## Create an overall table for continuous variables
contVars <- c("time", "age", "bili", "chol", "albumin", "copper",
             "alk.phos", "ast", "trig", "platelet", "protime")
contTableOverall <- CreateContTable(vars = contVars, data = pbc)

## Simply typing the object name will invoke the print.ContTable method,
## which will show the sample size, means and standard deviations.
contTableOverall

## To further examine the variables, use the summary.ContTable method,
## which will show more details.
summary(contTableOverall)
```

summary.TableOne	<i>Shows all results in a TableOne class object</i>
------------------	-----------------------------------------------------

Description

This method shows all the data a CatTable class object has. This includes the (optionally stratified) part with summary statistics and p-values from the approximation method test (chisq.test by default) and exact method test (fisher.test by default).

Usage

```
## S3 method for class 'TableOne'
summary(object, digits = 1, ...)
```

Arguments

object	An object that has the CatTable class to be shown.
digits	Number of digits to print.
...	For compatibility with generic. Ignored.

Value

It will print the results.

Author(s)

Kazuki Yoshida

See Also

[CreateTableOne](#), [print.TableOne](#), [summary.TableOne](#), [CreateContTable](#), [print.ContTable](#), [summary.ContTable](#), [CreateCatTable](#), [print.CatTable](#), [summary.CatTable](#)

Examples

```
## Load
library(tableone)

## Load Mayo Clinic Primary Biliary Cirrhosis Data
library(survival)
data(pbc)
## Check variables
head(pbc)

## Make categorical variables factors
varsToFactor <- c("status","trt","ascites","hepato","spiders","edema","stage")
pbc[varsToFactor] <- lapply(pbc[varsToFactor], factor)

## Create Table 1 stratified by sex and trt
tableOne <- CreateTableOne(vars = c("time","status","age","ascites","hepato",
                                   "spiders","edema","bili","chol","albumin",
                                   "copper","alk.phos","ast","trig","platelet",
                                   "protime","stage"),
                           strata = c("sex","trt"), data = pbc)

## Just typing the object name will invoke the print.TableOne method
tableOne

## Use the summary.TableOne method for detailed summary
summary(tableOne)
```

Index

by, [5](#), [7](#)

`chisq.test`, [4](#), [9](#), [20](#)

`CreateCatTable`, [2](#), [4](#), [5](#), [7](#), [10–12](#), [15](#), [17](#), [20](#),
[21](#), [23](#)

`CreateContTable`, [2](#), [5](#), [6](#), [7](#), [10](#), [12](#), [14](#), [15](#),
[17](#), [20](#), [21](#), [23](#)

`CreateTableOne`, [2](#), [5](#), [7](#), [8](#), [10](#), [12](#), [15–17](#), [20](#),
[21](#), [23](#)

`fisher.test`, [4](#), [10](#), [20](#)

`kruskal.test`, [21](#)

`oneway.test`, [7](#), [9](#), [21](#)

`print.CatTable`, [2](#), [4](#), [5](#), [7](#), [10](#), [11](#), [12](#), [15](#), [17](#),
[20](#), [21](#), [23](#)

`print.ContTable`, [2](#), [5–7](#), [10](#), [12](#), [14](#), [15](#), [17](#),
[20](#), [21](#), [23](#)

`print.TableOne`, [2](#), [5](#), [7](#), [9](#), [10](#), [12](#), [15](#), [16](#), [17](#),
[20](#), [21](#), [23](#)

`ShowRegTable`, [2](#), [18](#)

`summary.CatTable`, [2](#), [4](#), [5](#), [7](#), [10](#), [12](#), [15](#), [17](#),
[20](#), [20](#), [21](#), [23](#)

`summary.ContTable`, [2](#), [5–7](#), [10](#), [12](#), [15](#), [17](#),
[20](#), [21](#), [21](#), [23](#)

`summary.TableOne`, [2](#), [5](#), [7](#), [9](#), [10](#), [12](#), [15](#), [17](#),
[20](#), [21](#), [22](#), [23](#)

`tableone` (`tableone-package`), [2](#)

`tableone-package`, [2](#)