The **xtable** Gallery

Jonathan Swinton and others

August 30, 2018

Contents

1 Introduction

This document gives a gallery of tables which can be made using the **xtable** package to create LaTeX output. It doubles as a regression check for the package.

The first step is to load the package and set an option for this document.

```
library(xtable)
options(xtable.floating = FALSE)
options(xtable.timestamp = "")
```

2 Gallery

2.1 Data frame

```
data(tli)
xtable(tli[1:10, ])
```

	grade	sex	disadvg	ethnicty	tlimth
1	6	M	YES	HISPANIC	43
2	7	M	NO	BLACK	88
3	5	F	YES	HISPANIC	34
4	3	M	YES	HISPANIC	65
5	8	M	YES	WHITE	75
6	5	M	NO	BLACK	74
7	8	F	YES	HISPANIC	72
8	4	M	YES	BLACK	79
9	6	M	NO	WHITE	88
10	7	M	YES	HISPANIC	87

2.2 Matrix

```
design.matrix <- model.matrix(~ sex*grade, data = tli[1:10, ])
xtable(design.matrix, digits = 0)
```

	(Intercept)	sexM	grade	sexM:grade
1	1	1	6	6
2	1	1	7	7
3	1	0	5	0
4	1	1	3	3
5	1	1	8	8
6	1	1	5	5
7	1	0	8	0
8	1	1	4	4
9	1	1	6	6
10	1	1	7	7

2.3 aov

```
fm1 <- aov(tlimth ~ sex + ethnicty + grade + disadvg, data = tli)
xtable(fm1)
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
sex	1	75.37	75.37	0.38	0.5417
ethnicty	3	2572.15	857.38	4.27	0.0072
grade	1	36.31	36.31	0.18	0.6717
disadvg	1	59.30	59.30	0.30	0.5882
Residuals	93	18682.87	200.89		

2.4 lm

```
fm2 <- lm(tlimth ~ sex*ethnicty, data = tli)
xtable(fm2)
```

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	73.6364	4.2502	17.33	0.0000
sexM	-1.6364	5.8842	-0.28	0.7816
ethnictyHISPANIC	-9.7614	6.5501	-1.49	0.1395
ethnictyOTHER	15.8636	10.8360	1.46	0.1466
ethnictyWHITE	4.7970	4.9687	0.97	0.3368
sexM:ethnictyHISPANIC	10.6780	8.7190	1.22	0.2238
sexM:ethnictyWHITE	5.1230	7.0140	0.73	0.4670

Anova table (one model)

```
xtable(anova(fm2))
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
sex	1	75.37	75.37	0.38	0.5395
ethnicty	3	2572.15	857.38	4.31	0.0068
sex:ethnicty	2	298.43	149.22	0.75	0.4748
Residuals	93	18480.04	198.71		

Anova table (two models)

```
fm2b <- lm(tlimth ~ ethnicty, data = tli)
xtable(anova(fm2b, fm2))
```

	Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
1	96	19053.59				
2	93	18480.04	3	573.55	0.96	0.4141

Anova list

```
Block <- gl(8, 4)
A <- factor(c(0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,
              0,1,0,1,0,1,0,1,0,1,0,1))
B <- factor(c(0,0,1,1,0,0,1,1,0,1,0,1,1,0,1,0,0,0,1,1,
              0,0,1,1,0,0,1,1,0,0,1,1))
C <- factor(c(0,1,1,0,1,0,0,1,0,0,1,1,0,0,1,1,0,1,0,1,
              1,0,1,0,0,0,1,1,1,1,0,0))
Yield <- c(101, 373, 398, 291, 312, 106, 265, 450, 106, 306, 324, 449,
```

```
          272, 89, 407, 338, 87, 324, 279, 471, 323, 128, 423, 334,
          131, 103, 445, 437, 324, 361, 302, 272)
aovdat <- data.frame(Block, A, B, C, Yield)

old <- getOption("contrasts")
options(contrasts = c("contr.helmert", "contr.poly"))
(fit <- aov(Yield ~ A*B*C + Error(Block), data = aovdat))
```

```
##
## Call:
## aov(formula = Yield ~ A * B * C + Error(Block), data = aovdat)
##
## Grand Mean: 291.5938
##
## Stratum 1: Block
##
## Terms:
##                        A:B       A:C       B:C     A:B:C Residuals
## Sum of Squares   780.1250  276.1250 2556.1250  112.5000  774.0938
## Deg. of Freedom         1         1         1         1         3
##
## Residual standard error: 16.06335
## Estimated effects are balanced
##
## Stratum 2: Within
##
## Terms:
##                           A         B         C       A:B       A:C
## Sum of Squares    3465.28 161170.03 278817.78     28.17   1802.67
## Deg. of Freedom         1         1         1         1         1
##                        B:C     A:B:C Residuals
## Sum of Squares   11528.17     45.37   5423.28
## Deg. of Freedom         1         1        17
##
## Residual standard error: 17.86103
## Estimated effects are balanced
```

```
class(fit)
```

```
## [1] "aovlist" "listof"
```

```
summary(fit)
```

```
##
## Error: Block
##           Df Sum Sq Mean Sq F value Pr(>F)
## A:B        1  780.1   780.1   3.023 0.1805
## A:C        1  276.1   276.1   1.070 0.3770
## B:C        1 2556.1  2556.1   9.906 0.0514 .
## A:B:C      1  112.5   112.5   0.436 0.5562
## Residuals  3  774.1   258.0
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Error: Within
##           Df Sum Sq Mean Sq F value   Pr(>F)
## A          1   3465    3465  10.862  0.00427 **
## B          1 161170  161170 505.209 4.40e-14 ***
## C          1 278818  278818 873.992 4.67e-16 ***
```

```
## A:B         1     28      28   0.088  0.76996
## A:C         1   1803    1803   5.651  0.02946 *
## B:C         1  11528   11528  36.137 1.40e-05 ***
## A:B:C       1     45      45   0.142  0.71074
## Residuals 17   5423     319
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
options(contrasts = old)
```

```
xtable(fit)
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
A:B	1	780.12	780.12	3.02	0.1805
A:C	1	276.12	276.12	1.07	0.3770
B:C	1	2556.12	2556.12	9.91	0.0514
A:B:C	1	112.50	112.50	0.44	0.5562
Residuals	3	774.09	258.03		
A	1	3465.28	3465.28	10.86	0.0043
B	1	161170.03	161170.03	505.21	0.0000
C	1	278817.78	278817.78	873.99	0.0000
A:B 1	1	28.17	28.17	0.09	0.7700
A:C 1	1	1802.67	1802.67	5.65	0.0295
B:C 1	1	11528.17	11528.17	36.14	0.0000
A:B:C 1	1	45.37	45.37	0.14	0.7107
Residuals1	17	5423.28	319.02		

2.5 glm

```
fm3 <- glm(disadvg ~ ethnicty*grade, data = tli, family = binomial)
xtable(fm3)
```

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	3.1888	1.5966	2.00	0.0458
ethnictyHISPANIC	-0.2848	2.4808	-0.11	0.9086
ethnictyOTHER	212.1701	22122.7093	0.01	0.9923
ethnictyWHITE	-8.8150	3.3355	-2.64	0.0082
grade	-0.5308	0.2892	-1.84	0.0665
ethnictyHISPANIC:grade	0.2448	0.4357	0.56	0.5742
ethnictyOTHER:grade	-32.6014	3393.4687	-0.01	0.9923
ethnictyWHITE:grade	1.0171	0.5185	1.96	0.0498

Analysis of deviance

```
xtable(anova(fm3))
```

	Df	Deviance	Resid. Df	Resid. Dev
NULL			99	129.49
ethnicty	3	47.24	96	82.25
grade	1	1.73	95	80.52
ethnicty:grade	3	7.20	92	73.32

2.6 prcomp

```
pr1 <- prcomp(USArrests)
xtable(pr1)
```

	PC1	PC2	PC3	PC4
Murder	0.0417	-0.0448	0.0799	-0.9949
Assault	0.9952	-0.0588	-0.0676	0.0389
UrbanPop	0.0463	0.9769	-0.2005	-0.0582
Rape	0.0752	0.2007	0.9741	0.0723

```
xtable(summary(pr1))
```

	PC1	PC2	PC3	PC4
Standard deviation	83.7324	14.2124	6.4894	2.4828
Proportion of Variance	0.9655	0.0278	0.0058	0.0008
Cumulative Proportion	0.9655	0.9933	0.9991	1.0000

2.7 Time series

```
temp.ts <- ts(cumsum(1 + round(rnorm(100), 0)),
              start = c(1954, 7), frequency = 12)
temp.table <- xtable(temp.ts, digits = 0)
temp.table
```

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
1954							0	1	3	2	3	5
1955	5	5	5	5	6	6	6	7	9	10	10	10
1956	10	13	14	15	16	17	17	17	19	19	20	20
1957	22	23	23	23	22	22	21	21	22	23	25	25
1958	25	26	26	26	26	26	26	27	26	26	26	26
1959	27	29	32	32	35	35	37	41	42	42	43	46
1960	46	48	50	51	52	53	54	56	59	60	60	60
1961	61	62	63	64	64	65	67	69	71	72	73	73
1962	74	75	78	80	81	82	82	84	86	89		

2.8 Flat tables

See the **Details** section of the help for `ftable` for a description of these tables, which are flat versions of multi-dimensional contingency tables. They require special methods to enable them to be printed using **xtable**

```
data(mtcars)
mtcars$cyl <- factor(mtcars$cyl, levels = c("4","6","8"),
                     labels = c("four","six","eight"))
tbl <- ftable(mtcars$cyl, mtcars$vs, mtcars$am, mtcars$gear,
              row.vars = c(2, 4),
              dnn = c("Cylinders", "V/S", "Transmission", "Gears"))
tbl
```

```
##           Cylinders    four    six    eight
##           Transmission    0  1   0  1     0  1
## V/S Gears
## 0   3                     0  0   0  0    12  0
##     4                     0  0   0  2     0  0
##     5                     0  1   0  1     0  2
## 1   3                     1  0   2  0     0  0
##     4                     2  6   2  0     0  0
##     5                     0  1   0  0     0  0
```

Here is the LaTeX produced:

```
xftbl <- xtableFtable(tbl, method = "compact")
print.xtableFtable(xftbl, booktabs = TRUE)
```

```
## % latex table generated in R 3.4.3 by xtable 1.8-3 package
## %
## \begin{tabular}{ll rrrrrr}
##   \toprule
##      & Cylinders                    & \multicolumn{1}{l}{ four} & \multicolumn{1}{l}{   } & \mu
##   V/S & Gears $\vert$ Transmission & \multicolumn{1}{l}{    0} & \multicolumn{1}{l}{  1} & \mu
##    \midrule
## 0   & 3                          &     0 &  0 &   0 &  0 &    12 &  0 \\
##       & 4                          &     0 &  0 &   0 &  2 &     0 &  0 \\
##       & 5                          &     0 &  1 &   0 &  1 &     0 &  2 \\
##   1   & 3                          &     1 &  0 &   2 &  0 &     0 &  0 \\
```

```
##       & 4                              &    2 &  6 &   2 &  0 &     0 &  0 \\
##       & 5                              &    0 &  1 &   0 &  0 &     0 &  0 \\
##    \bottomrule
## \end{tabular}
```

And here is a basic flat table:

```
xftbl <- xtableFtable(tbl)
print.xtableFtable(xftbl)
```

V/S	Gears	Cylinders / Transmission	four 0	four 1	six 0	six 1	eight 0	eight 1
0	3		0	0	0	0	12	0
	4		0	0	0	2	0	0
	5		0	1	0	1	0	2
1	3		1	0	2	0	0	0
	4		2	6	2	0	0	0
	5		0	1	0	0	0	0

This illustrates the `method` argument:

```
xftbl <- xtableFtable(tbl, method = "col.compact")
print.xtableFtable(xftbl, rotate.rownames = TRUE)
```

V/S	Gears	Transmission / Cylinders	four 0	four 1	six 0	six 1	eight 0	eight 1
0	3		0	0	0	0	12	0
	4		0	0	0	2	0	0
	5		0	1	0	1	0	2
1	3		1	0	2	0	0	0
	4		2	6	2	0	0	0
	5		0	1	0	0	0	0

Booktabs is incompatible with vertical lines in tables, so the vertical dividing line is removed.

```
xftbl <- xtableFtable(tbl, method = "compact")
print.xtableFtable(xftbl, booktabs = TRUE)
```

V/S	Gears | Transmission / Cylinders	four 0	four 1	six 0	six 1	eight 0	eight 1
0	3	0	0	0	0	12	0
	4	0	0	0	2	0	0
	5	0	1	0	1	0	2
1	3	1	0	2	0	0	0
	4	2	6	2	0	0	0
	5	0	1	0	0	0	0

Row and column variable names can be formatted specially using sanitization, and row and column variable names and labels can be rotated.

If special formatting is required for row and column labels, that can be done as a workaround by redefining the data and associated labels.

```
italic <- function(x){
  paste0('{\\emph{', x, '}}')
}
mtcars$cyl <- factor(mtcars$cyl, levels = c("four","six","eight"),
                     labels = c("four",italic("six"),"eight"))
large <- function(x){
  paste0('{\\Large ', x, '}')
}
bold <- function(x){
  paste0('{\\bfseries ', x, '}')
}
tbl <- ftable(mtcars$cyl, mtcars$vs, mtcars$am, mtcars$gear,
              row.vars = c(2, 4),
              dnn = c("Cylinders", "V/S", "Transmission", "Gears"))
xftbl <- xtableFtable(tbl, method = "row.compact")
print.xtableFtable(xftbl,
                   sanitize.rownames.function = large,
                   sanitize.colnames.function = bold,
                   rotate.colnames = TRUE,
                   rotate.rownames = TRUE)
```

		Cylinders	four		*six*		eight	
V/S	Gears	**Transmission**	0	1	0	1	0	1
0	3		0	0	0	0	12	0
	4		0	0	0	2	0	0
	5		0	1	0	1	0	2
1	3		1	0	2	0	0	0
	4		2	6	2	0	0	0
	5		0	1	0	0	0	0

3 Automatic formatting

3.1 Suggest alignment, digits, and display

The functions `xalign`, `xdigits`, and `xdisplay` are useful for formatting tables in a sensible way. Consider the output produced by the default formatting.

```
data(mtcars)
dat <- mtcars[1:3, 1:6]
x <- xtable(dat)
x
```

	mpg	cyl	disp	hp	drat	wt
Mazda RX4	21.00	6.00	160.00	110.00	3.90	2.62
Mazda RX4 Wag	21.00	6.00	160.00	110.00	3.90	2.88
Datsun 710	22.80	4.00	108.00	93.00	3.85	2.32

Now change the default alignment, digits and display using helper functions `xalign`, `xdigits`, and `xdisplay`. This produces a better format as shown below.

```
align(x) <- xalign(x)
digits(x) <- xdigits(x)
display(x) <- xdisplay(x)
x
```

	mpg	cyl	disp	hp	drat	wt
Mazda RX4	21.0	6	160	110	3.90	2.620
Mazda RX4 Wag	21.0	6	160	110	3.90	2.875
Datsun 710	22.8	4	108	93	3.85	2.320

3.2 Shorthand notation

For convenience, the three 'autoformat' functions (`xalign`, `xdigits`, and `xdisplay`) can be applied together when an `xtable` is created, using the `auto` argument:

```
xtable(dat, auto = TRUE)
```

	mpg	cyl	disp	hp	drat	wt
Mazda RX4	21.0	6	160	110	3.90	2.620
Mazda RX4 Wag	21.0	6	160	110	3.90	2.875
Datsun 710	22.8	4	108	93	3.85	2.320

Similarly, the `autoformat` function can be used to postprocess an existing `xtable`:

```
x <- xtable(dat)
autoformat(x)
```

	mpg	cyl	disp	hp	drat	wt
Mazda RX4	21.0	6	160	110	3.90	2.620
Mazda RX4 Wag	21.0	6	160	110	3.90	2.875
Datsun 710	22.8	4	108	93	3.85	2.320

3.3 Math-Style Exponents

If you prefer 5×10^5 in your tables to 5e5, the `math.style.exponents` option to `print.xtable` is useful:

```
print(xtable(data.frame(text = c("foo","bar"),
                        googols = c(10e10,50e10),
                        small = c(8e-24,7e-5),
                        row.names = c("A","B")),
             display = c("s","s","g","g")),
      math.style.exponents = TRUE)
```

	text	googols	small
A	foo	1×10^{11}	8×10^{-24}
B	bar	5×10^{11}	7×10^{-5}

this option also supports the values `ensuremath` which uses `\ensuremath` instead of `$$` and `UTF-8` which uses UTF-8 to approximate the LaTeXtypesetting.

4 Sanitization

```
insane <- data.frame(Name = c("Ampersand","Greater than","Less than",
                              "Underscore","Per cent","Dollar",
                              "Backslash","Hash","Caret","Tilde",
                              "Left brace","Right brace"),
                     Character = I(c("&",">","<","_","%","$",
                                     "\\","#","^","~","{","}")))
colnames(insane)[2] <- paste(insane[, 2], collapse = "")
xtable(insane)
```

	Name	&><_%$\#^~{}
1	Ampersand	&
2	Greater than	>
3	Less than	<
4	Underscore	_
5	Per cent	%
6	Dollar	$
7	Backslash	\
8	Hash	#
9	Caret	^
10	Tilde	~
11	Left brace	{
12	Right brace	}

Sometimes you might want to have your own sanitization function.

```
wanttex <- xtable(data.frame(Column =
                             paste("Value_is $10^{-",1:3,"}$", sep = "")))
print(wanttex, sanitize.text.function =
      function(str) gsub("_", "\\_", str, fixed = TRUE))
```

	Column
1	Value_is 10^{-1}
2	Value_is 10^{-2}
3	Value_is 10^{-3}

Sanitization can be useful in formatting column headings and row names:

```r
dat <- mtcars[1:3, 1:6]
large <- function(x){
  paste0('{\\Large{\\bfseries ', x, '}}')
}
italic <- function(x){
  paste0('{\\emph{ ', x, '}}')
}
```

```r
print(xtable(dat),
      sanitize.rownames.function = italic,
      sanitize.colnames.function = large,
      booktabs = TRUE)
```

	mpg	**cyl**	**disp**	**hp**	**drat**	**wt**
Mazda RX4	21.00	6.00	160.00	110.00	3.90	2.62
Mazda RX4 Wag	21.00	6.00	160.00	110.00	3.90	2.88
Datsun 710	22.80	4.00	108.00	93.00	3.85	2.32

4.1 Markup in tables

Markup can be included in tables, including in column and row names, by using a custom `sanitize.text.function`.

```
mat <- round(matrix(c(0.9, 0.89, 200, 0.045, 2.0), c(1, 5)), 4)
rownames(mat) <- "$y_{t-1}$"
colnames(mat) <- c("$R^2$", "$\\bar{x}$", "F-stat", "S.E.E", "DW")
mat <- xtable(mat)
print(mat, sanitize.text.function = function(x) {x})
```

	R^2	$\bar{x}$	F-stat	S.E.E	DW
y_{t-1}	0.90	0.89	200.00	0.04	2.00

You can also have sanitize functions that are specific to column or row names. In the table below, the row name is not sanitized but column names and table elements are.

```
money <- matrix(c("$1,000","$900","$100"), ncol = 3,
                dimnames = list("$\\alpha$",
                                c("Income (US$)","Expenses (US$)",
                                  "Profit (US$)")))
print(xtable(money), sanitize.rownames.function = function(x) {x})
```

	Income (US$)	Expenses (US$)	Profit (US$)
α	$1,000	$900	$100

5 Format examples

5.1 Adding a centering environment

```
print(xtable(anova(fm3), caption = "\\tt latex.environments = \"\""),
      floating = TRUE, latex.environments = "")
```

	Df	Deviance	Resid. Df	Resid. Dev
NULL			99	129.49
ethnicty	3	47.24	96	82.25
grade	1	1.73	95	80.52
ethnicty:grade	3	7.20	92	73.32

Table 1: `latex.environments = ""`

```
print(xtable(anova(fm3), caption = "\\tt latex.environments = \"center\""),
      floating = TRUE, latex.environments = "center")
```

	Df	Deviance	Resid. Df	Resid. Dev
NULL			99	129.49
ethnicty	3	47.24	96	82.25
grade	1	1.73	95	80.52
ethnicty:grade	3	7.20	92	73.32

Table 2: `latex.environments = "center"`

5.2 Column alignment

```
tli.table <- xtable(tli[1:10, ])
align(tli.table) <- rep("r", 6)
tli.table
```

	grade	sex	disadvg	ethnicty	tlimth
1	6	M	YES	HISPANIC	43
2	7	M	NO	BLACK	88
3	5	F	YES	HISPANIC	34
4	3	M	YES	HISPANIC	65
5	8	M	YES	WHITE	75
6	5	M	NO	BLACK	74
7	8	F	YES	HISPANIC	72
8	4	M	YES	BLACK	79
9	6	M	NO	WHITE	88
10	7	M	YES	HISPANIC	87

Left aligned strings with column lines

```
align(tli.table) <- "|rrl|l|lr|"
tli.table
```

	grade	sex	disadvg	ethnicty	tlimth
1	6	M	YES	HISPANIC	43
2	7	M	NO	BLACK	88
3	5	F	YES	HISPANIC	34
4	3	M	YES	HISPANIC	65
5	8	M	YES	WHITE	75
6	5	M	NO	BLACK	74
7	8	F	YES	HISPANIC	72
8	4	M	YES	BLACK	79
9	6	M	NO	WHITE	88
10	7	M	YES	HISPANIC	87

Fixed width columns

```
align(tli.table) <- "|rr|lp{3cm}l|r|"
tli.table
```

	grade	sex	disadvg	ethnicty	tlimth
1	6	M	YES	HISPANIC	43
2	7	M	NO	BLACK	88
3	5	F	YES	HISPANIC	34
4	3	M	YES	HISPANIC	65
5	8	M	YES	WHITE	75
6	5	M	NO	BLACK	74
7	8	F	YES	HISPANIC	72
8	4	M	YES	BLACK	79
9	6	M	NO	WHITE	88
10	7	M	YES	HISPANIC	87

5.3 Number of digits

One number for all columns,

```
display(tli.table)[c(2,6)] <- "f"
digits(tli.table) <- 3
tli.table
```

	grade	sex	disadvg	ethnicty	tlimth
1	6.000	M	YES	HISPANIC	43.000
2	7.000	M	NO	BLACK	88.000
3	5.000	F	YES	HISPANIC	34.000
4	3.000	M	YES	HISPANIC	65.000
5	8.000	M	YES	WHITE	75.000
6	5.000	M	NO	BLACK	74.000
7	8.000	F	YES	HISPANIC	72.000
8	4.000	M	YES	BLACK	79.000
9	6.000	M	NO	WHITE	88.000
10	7.000	M	YES	HISPANIC	87.000

or one for each column, including the row names,

```
digits(tli.table) <- 1:(ncol(tli)+1)
tli.table
```

	grade	sex	disadvg	ethnicty	tlimth
1	6.00	M	YES	HISPANIC	43.000000
2	7.00	M	NO	BLACK	88.000000
3	5.00	F	YES	HISPANIC	34.000000
4	3.00	M	YES	HISPANIC	65.000000
5	8.00	M	YES	WHITE	75.000000
6	5.00	M	NO	BLACK	74.000000
7	8.00	F	YES	HISPANIC	72.000000
8	4.00	M	YES	BLACK	79.000000
9	6.00	M	NO	WHITE	88.000000
10	7.00	M	YES	HISPANIC	87.000000

or as a full matrix.

```
digits(tli.table) <- matrix(0:4, nrow = 10, ncol = ncol(tli)+1)
tli.table
```

	grade	sex	disadvg	ethnicty	tlimth
1	6	M	YES	HISPANIC	43
2	7.0	M	NO	BLACK	88.0
3	5.00	F	YES	HISPANIC	34.00
4	3.000	M	YES	HISPANIC	65.000
5	8.0000	M	YES	WHITE	75.0000
6	5	M	NO	BLACK	74
7	8.0	F	YES	HISPANIC	72.0
8	4.00	M	YES	BLACK	79.00
9	6.000	M	NO	WHITE	88.000
10	7.0000	M	YES	HISPANIC	87.0000

5.4 Suppress row/column names

Suppress row names

```
tli.table <- xtable(tli[1:10, ])
print(tli.table, include.rownames = FALSE)
```

grade	sex	disadvg	ethnicty	tlimth
6	M	YES	HISPANIC	43
7	M	NO	BLACK	88
5	F	YES	HISPANIC	34
3	M	YES	HISPANIC	65
8	M	YES	WHITE	75
5	M	NO	BLACK	74
8	F	YES	HISPANIC	72
4	M	YES	BLACK	79
6	M	NO	WHITE	88
7	M	YES	HISPANIC	87

If you want a vertical line on the left, you need to change the `align` attribute.

```
align(tli.table) <- "|r|r|lp{3cm}l|r|"
print(tli.table, include.rownames = FALSE)
```

grade	sex	disadvg	ethnicty	tlimth
6	M	YES	HISPANIC	43
7	M	NO	BLACK	88
5	F	YES	HISPANIC	34
3	M	YES	HISPANIC	65
8	M	YES	WHITE	75
5	M	NO	BLACK	74
8	F	YES	HISPANIC	72
4	M	YES	BLACK	79
6	M	NO	WHITE	88
7	M	YES	HISPANIC	87

Revert the alignment to what is was before.

```
align(tli.table) <- "|rr|lp{3cm}l|r|"
```

Suppress column names

```
print(tli.table, include.colnames = FALSE)
```

1	6	M	YES	HISPANIC	43
2	7	M	NO	BLACK	88
3	5	F	YES	HISPANIC	34
4	3	M	YES	HISPANIC	65
5	8	M	YES	WHITE	75
6	5	M	NO	BLACK	74
7	8	F	YES	HISPANIC	72
8	4	M	YES	BLACK	79
9	6	M	NO	WHITE	88
10	7	M	YES	HISPANIC	87

Note the doubled header lines which can be suppressed.

```
print(tli.table, include.colnames = FALSE,
      hline.after = c(0,nrow(tli.table)))
```

1	6	M	YES	HISPANIC	43
2	7	M	NO	BLACK	88
3	5	F	YES	HISPANIC	34
4	3	M	YES	HISPANIC	65
5	8	M	YES	WHITE	75
6	5	M	NO	BLACK	74
7	8	F	YES	HISPANIC	72
8	4	M	YES	BLACK	79
9	6	M	NO	WHITE	88
10	7	M	YES	HISPANIC	87

Suppress row and column names

```
print(tli.table, include.colnames = FALSE, include.rownames = FALSE)
```

6	M	YES	HISPANIC	43
7	M	NO	BLACK	88
5	F	YES	HISPANIC	34
3	M	YES	HISPANIC	65
8	M	YES	WHITE	75
5	M	NO	BLACK	74
8	F	YES	HISPANIC	72
4	M	YES	BLACK	79
6	M	NO	WHITE	88
7	M	YES	HISPANIC	87

5.5 Rotate row/column names

The `rotate.rownames` and `rotate.colnames` arguments can be used to rotate the row and/or column names. This requires `\usepackage{rotating}` in the LaTeX preamble.

```
print(tli.table, rotate.rownames = TRUE, rotate.colnames = TRUE)
```

	grade	sex	disadvg	ethnicty	tlimth
1	6	M	YES	HISPANIC	43
2	7	M	NO	BLACK	88
3	5	F	YES	HISPANIC	34
4	3	M	YES	HISPANIC	65
5	8	M	YES	WHITE	75
6	5	M	NO	BLACK	74
7	8	F	YES	HISPANIC	72
8	4	M	YES	BLACK	79
9	6	M	NO	WHITE	88
10	7	M	YES	HISPANIC	87

5.6 Horizontal lines

Line locations

Use the `hline.after` argument to specify the position of the horizontal lines.

```
print(xtable(anova(fm3)), hline.after = c(1))
```

	Df	Deviance	Resid. Df	Resid. Dev
NULL			99	129.49
ethnicty	3	47.24	96	82.25
grade	1	1.73	95	80.52
ethnicty:grade	3	7.20	92	73.32

Line styles

Specifying `booktabs = TRUE` will generate three line types. By default, when no value is given for `hline.after`, a `\toprule` will be drawn above the table, a `\midrule` after the table headings and a `\bottomrule` below the table. This requires `\usepackage{booktabs}` in the LaTeX preamble.

The top and bottom rules are slightly thicker than the mid rule. The thickness of the lines can be set via the LaTeX lengths `\heavyrulewidth` and `\lightrulewidth`.

```
tli.table <- xtable(tli[1:10, ])
print(tli.table, include.rownames = FALSE, booktabs = TRUE)
```

grade	sex	disadvg	ethnicty	tlimth
6	M	YES	HISPANIC	43
7	M	NO	BLACK	88
5	F	YES	HISPANIC	34
3	M	YES	HISPANIC	65
8	M	YES	WHITE	75
5	M	NO	BLACK	74
8	F	YES	HISPANIC	72
4	M	YES	BLACK	79
6	M	NO	WHITE	88
7	M	YES	HISPANIC	87

If `hline.after` includes `-1`, a `\toprule` will be drawn above the table. If `hline.after` includes the number of rows in the table, a `\bottomrule` will be drawn below the table. For any other values specified in `hline.after`, a `\midrule` will be drawn after that line of the table.

The following table has more than one `\midrule`.

```
bktbs <- xtable(matrix(1:10, ncol = 2))
hlines <- c(-1, 0, 1, nrow(bktbs))
print(bktbs, booktabs = TRUE, hline.after = hlines)
```

	1	2
1	1	6
2	2	7
3	3	8
4	4	9
5	5	10

5.7 Table level commands

```
print(xtable(anova(fm3)), size = "large")
```

	Df	Deviance	Resid. Df	Resid. Dev
NULL			99	129.49
ethnicty	3	47.24	96	82.25
grade	1	1.73	95	80.52
ethnicty:grade	3	7.20	92	73.32

```
print(xtable(anova(fm3)), size = "\\setlength{\\tabcolsep}{12pt}")
```

	Df	Deviance	Resid. Df	Resid. Dev
NULL			99	129.49
ethnicty	3	47.24	96	82.25
grade	1	1.73	95	80.52
ethnicty:grade	3	7.20	92	73.32

5.8 Long tables

Requires `\usepackage{longtable}` in the LaTeX preamble.

```
x <- matrix(rnorm(1000), ncol = 10)
x.big <- xtable(x, caption = "A \\code{longtable} spanning several pages")
print(x.big, hline.after=c(-1, 0), tabular.environment = "longtable")
```

	1	2	3	4	5	6	7	8	9	10
1	0.41	0.49	-0.58	-1.23	0.98	0.31	0.45	-1.02	-2.87	-1.21
2	-0.47	0.70	-0.95	0.04	-1.22	0.61	1.43	-1.39	-0.43	0.30
3	0.07	0.19	-0.18	-0.42	0.71	-1.69	-0.21	-0.05	-1.49	-1.54
4	-0.50	0.70	1.01	-0.90	-0.11	0.78	-1.23	1.81	0.59	0.64
5	-0.83	0.31	0.02	0.42	1.78	0.01	-0.36	-0.10	-0.44	0.70
6	0.17	0.76	-0.65	0.15	-0.24	-0.18	-0.55	0.78	0.90	-1.91
7	-0.90	1.84	-0.50	1.46	-1.53	1.11	0.33	-1.10	0.47	0.94
8	0.17	1.11	1.61	-1.12	0.49	1.48	-0.93	-0.22	0.02	-0.22
9	0.35	0.03	-0.45	-0.52	0.35	-1.15	0.68	0.57	-0.09	-0.67
10	-0.05	-1.11	0.76	-0.07	-0.02	1.01	-1.16	-0.35	-0.36	0.45
11	-0.20	0.42	1.47	-1.41	-1.06	-0.63	1.76	0.79	0.20	1.28
12	-0.65	-0.40	0.44	-0.28	-0.84	0.13	1.45	0.69	-1.11	1.57
13	-1.11	1.49	-0.42	-0.71	-0.01	0.48	-0.45	-0.63	0.67	-1.20
14	0.85	-1.61	-0.04	-2.15	1.04	0.13	0.14	-0.77	-0.39	-0.44
15	0.02	-0.42	-0.49	-0.28	-0.36	-1.46	-0.37	2.36	-0.74	0.15
16	0.83	0.42	1.23	-0.53	-0.87	-0.51	-1.21	-0.19	-0.33	0.07
17	-1.24	-0.15	-0.15	1.13	-2.70	1.54	-2.66	0.01	1.87	-0.45
18	0.17	-0.61	1.55	-0.60	0.49	0.17	-0.03	-0.05	-1.76	-2.34
19	0.67	-0.30	-0.56	0.56	0.43	-0.84	-0.03	-0.57	-0.35	-0.28
20	-0.03	0.63	-0.65	0.14	1.31	-0.66	1.64	-0.93	0.11	-1.18
21	-0.19	0.90	0.14	-1.24	1.47	1.06	-2.00	-1.14	-0.54	0.03
22	-0.78	0.66	0.02	0.37	-0.14	0.35	-0.66	0.91	-0.41	-0.84
23	2.06	2.27	-0.50	-0.11	1.30	-0.90	0.88	-0.99	0.38	-0.12
24	0.75	1.17	-1.58	0.17	0.00	0.01	0.34	0.67	0.22	-0.74
25	1.82	0.29	0.03	0.66	-0.68	0.60	0.12	1.41	-0.67	-0.41
26	0.08	-0.66	-0.72	0.11	-0.85	1.72	-0.62	1.16	0.57	-0.38
27	-0.63	2.92	1.08	1.59	-0.35	-0.88	0.20	-0.14	0.18	1.33
28	-1.51	0.68	-0.95	0.12	0.23	-0.38	-0.24	-0.54	0.69	0.72
29	-0.64	-0.68	1.13	-1.25	-0.03	1.00	0.11	-1.08	-1.33	-0.23

30	0.23	0.19	-0.65	0.74	0.20	-0.46	-0.58	-0.02	-0.77	-1.23
31	1.01	-0.32	0.29	-1.22	0.20	1.46	1.19	-0.23	-0.19	1.05
32	0.25	-0.27	0.90	-1.92	0.71	1.04	1.06	0.16	0.96	-1.44
33	-1.17	-0.93	-0.52	-0.81	-0.27	-1.66	1.34	-0.60	-0.53	-1.18
34	0.67	0.12	0.55	-0.74	1.64	0.42	-0.37	0.63	-1.03	-0.78
35	-1.65	0.32	-0.09	-0.61	0.44	-1.42	-0.30	0.71	0.77	-0.97
36	-0.37	-1.08	-1.14	1.61	-1.19	-1.46	-0.35	1.08	0.47	-0.90
37	-0.32	-3.23	-0.27	1.64	-0.28	0.07	1.37	2.25	-1.23	1.29
38	-1.95	-0.25	1.62	0.73	0.52	0.06	-0.30	0.20	2.05	-0.11
39	0.92	0.03	-0.21	-1.27	-0.41	-0.03	0.10	-0.58	1.50	-0.72
40	-0.62	0.59	-0.82	0.02	0.34	1.26	0.38	-0.08	-1.39	0.67
41	-0.33	0.06	-0.05	0.49	0.43	0.29	0.68	-1.63	2.15	0.20
42	1.40	0.41	0.33	-0.14	1.30	-0.67	-0.27	0.13	-1.73	-1.22
43	0.64	-1.10	0.96	0.17	-0.65	0.12	2.35	-0.43	0.10	-0.56
44	-0.11	0.71	1.14	-0.94	-0.02	-0.51	-0.93	0.41	-1.14	0.53
45	0.51	0.72	0.10	-1.29	-0.57	-0.07	-1.43	-0.27	-0.38	1.95
46	0.40	0.25	1.16	1.41	-1.13	-1.15	0.04	-1.38	-0.35	1.58
47	1.66	1.36	-0.76	0.27	-1.85	-0.12	1.77	0.04	0.32	0.84
48	0.28	0.40	-2.34	-0.41	0.20	0.10	0.05	-1.35	-1.12	-0.01
49	0.51	0.26	-0.47	0.33	0.60	-0.70	0.11	0.90	-2.76	-1.73
50	0.35	0.27	-0.52	2.06	0.64	0.04	0.86	-1.83	-0.10	-0.73
51	-0.38	0.44	-2.32	0.36	-1.39	-0.65	-0.10	-0.14	-0.75	-0.22
52	0.10	1.06	0.56	1.41	-1.13	-0.64	0.10	0.46	0.21	0.49
53	1.64	0.45	-0.78	1.37	-1.14	-0.20	1.21	-1.52	-0.85	0.06
54	-0.88	0.66	-0.23	-0.41	-0.69	0.49	0.09	1.40	0.79	0.06
55	0.12	-1.14	-1.59	0.76	0.17	-0.21	-0.75	-0.89	0.03	1.84
56	1.36	-0.37	0.55	-0.65	0.59	0.19	0.39	-0.51	0.60	0.39
57	-0.23	1.48	1.89	-1.47	-0.82	0.97	-1.09	0.16	-0.99	-0.28
58	-1.05	-1.22	-0.88	-1.20	-2.86	-0.29	-1.46	-0.34	-0.21	-0.43
59	-0.87	0.26	-0.11	-0.15	0.95	-0.10	-0.12	-1.04	0.34	0.10
60	-0.39	0.41	1.95	1.80	0.62	0.51	-1.10	0.41	-0.09	-0.81
61	-0.85	0.98	0.93	0.10	-0.70	-1.26	0.58	-0.52	1.65	1.37
62	-0.26	-0.35	1.91	-0.80	-1.07	0.54	-0.15	0.51	-1.96	-0.29
63	-0.41	0.16	-0.01	0.23	-0.30	-0.53	-0.77	-0.50	0.76	-0.53
64	-0.18	-1.76	-0.15	0.70	0.11	-1.14	1.62	-0.96	0.01	-0.87
65	0.41	0.34	-0.51	-1.30	0.72	-1.77	-0.11	-0.09	0.95	0.94
66	0.62	-0.67	1.43	-1.05	1.39	-0.71	1.42	0.65	-0.10	-1.28
67	1.68	-0.24	-1.29	-1.94	-0.85	-1.17	-0.11	0.24	-2.74	-0.47
68	-0.07	-1.19	0.31	-1.27	0.29	-1.18	-0.33	-0.06	0.42	0.31
69	-0.32	0.38	-0.05	-0.89	1.33	1.63	0.37	-1.95	-0.22	-0.11
70	1.47	0.67	2.25	-0.29	-0.90	-1.21	1.03	1.45	-0.99	0.25
71	1.70	-0.30	-0.61	1.41	-0.22	1.00	2.71	-0.04	-0.66	-1.32
72	0.04	1.83	-1.51	0.89	-1.77	0.55	-1.03	-0.86	0.06	-0.43
73	-0.33	0.67	0.23	0.27	0.29	-0.88	-0.18	-0.20	2.45	-1.70
74	-1.82	0.95	-0.04	-0.57	-0.64	-0.61	1.08	0.26	0.19	-0.84
75	1.41	2.05	-0.84	-0.04	-0.92	-0.41	0.34	-1.52	-0.98	1.15
76	-0.84	-0.65	0.13	0.55	-0.78	0.19	-0.19	0.72	1.90	0.25
77	-1.12	0.81	-0.28	-0.50	0.44	-0.35	-1.30	0.25	-0.44	0.09
78	3.04	0.99	-0.68	1.06	-2.16	0.83	-0.28	1.20	-0.83	-0.93
79	0.24	-0.01	0.50	1.10	-0.43	-0.29	-0.18	0.99	-1.25	-0.56
80	-0.03	0.32	-0.33	-1.17	-1.01	-2.17	-0.08	-0.54	-0.67	-1.02
81	-2.73	-1.01	-1.83	-0.75	-0.96	-0.10	1.61	-0.49	1.94	0.05
82	-0.10	0.47	-2.65	1.21	0.47	-1.85	-0.46	-1.00	0.91	0.77
83	0.98	-0.70	-0.58	-1.69	0.35	0.94	-1.58	-0.36	0.74	0.30
84	0.41	0.81	1.45	0.42	-0.11	-0.75	0.50	0.94	0.15	-0.62
85	0.91	-0.81	0.84	0.23	0.54	-1.12	-0.11	1.79	0.28	0.55
86	1.98	0.32	1.22	3.20	-0.60	-0.59	-0.20	0.93	0.59	0.39

87	1.17	-0.85	0.98	-2.73	-2.02	0.59	0.64	-1.83	0.22	0.84
88	-0.51	-0.25	0.32	-0.84	1.15	-1.92	-2.91	0.82	0.16	-0.44
89	0.70	-1.55	-1.51	0.67	-0.23	0.25	-0.55	1.04	0.95	-1.18
90	-0.20	0.13	0.21	1.67	-0.37	-0.97	-0.15	0.62	-0.17	-0.08
91	-0.54	0.99	1.60	0.84	0.16	-0.01	0.61	-0.85	0.26	0.30
92	-2.86	0.18	-3.40	-1.43	0.77	0.68	-0.26	1.06	-0.90	1.56
93	-0.79	-1.77	-0.78	2.24	0.43	0.57	-1.01	0.73	1.58	-0.07
94	0.49	-0.62	1.10	-1.76	0.93	1.50	0.37	0.66	1.39	0.43
95	2.17	1.66	0.53	-1.11	0.05	0.13	0.02	-0.38	-1.81	-0.08
96	0.50	1.81	0.79	-0.04	-0.50	-0.70	0.11	0.85	-0.22	2.16
97	0.62	-1.18	0.46	2.23	-0.30	2.15	1.16	0.11	0.55	1.34
98	-0.97	-0.37	0.54	0.51	0.82	-0.30	-0.54	-0.07	0.48	-0.23
99	0.16	0.35	0.01	0.73	-0.85	0.81	-1.25	1.49	0.76	-0.13
100	-2.08	0.32	-0.92	1.73	-0.95	0.93	-1.28	0.76	-0.45	-0.02

Table 3: A `longtable` spanning several pages

Extra features of the **longtable** LaTeX package can typically be activated using `add.to.row`, as shown below.

```
add.to.row <- list(pos = list(0), command = NULL)
command <- paste0("\\hline\n\\endhead\n",
                  "\\hline\n",
                  "\\multicolumn{", dim(x)[2] + 1, "}{l}",
                  "{\\footnotesize Continued on next page}\n",
                  "\\endfoot\n",
                  "\\endlastfoot\n")
add.to.row$command <- command
print(x.big, hline.after=c(-1), add.to.row = add.to.row,
      tabular.environment = "longtable")
```

	1	2	3	4	5	6	7	8	9	10
1	0.41	0.49	-0.58	-1.23	0.98	0.31	0.45	-1.02	-2.87	-1.21
2	-0.47	0.70	-0.95	0.04	-1.22	0.61	1.43	-1.39	-0.43	0.30
3	0.07	0.19	-0.18	-0.42	0.71	-1.69	-0.21	-0.05	-1.49	-1.54
4	-0.50	0.70	1.01	-0.90	-0.11	0.78	-1.23	1.81	0.59	0.64
5	-0.83	0.31	0.02	0.42	1.78	0.01	-0.36	-0.10	-0.44	0.70
6	0.17	0.76	-0.65	0.15	-0.24	-0.18	-0.55	0.78	0.90	-1.91
7	-0.90	1.84	-0.50	1.46	-1.53	1.11	0.33	-1.10	0.47	0.94
8	0.17	1.11	1.61	-1.12	0.49	1.48	-0.93	-0.22	0.02	-0.22
9	0.35	0.03	-0.45	-0.52	0.35	-1.15	0.68	0.57	-0.09	-0.67
10	-0.05	-1.11	0.76	-0.07	-0.02	1.01	-1.16	-0.35	-0.36	0.45
11	-0.20	0.42	1.47	-1.41	-1.06	-0.63	1.76	0.79	0.20	1.28
12	-0.65	-0.40	0.44	-0.28	-0.84	0.13	1.45	0.69	-1.11	1.57
13	-1.11	1.49	-0.42	-0.71	-0.01	0.48	-0.45	-0.63	0.67	-1.20
14	0.85	-1.61	-0.04	-2.15	1.04	0.13	0.14	-0.77	-0.39	-0.44
15	0.02	-0.42	-0.49	-0.28	-0.36	-1.46	-0.37	2.36	-0.74	0.15
16	0.83	0.42	1.23	-0.53	-0.87	-0.51	-1.21	-0.19	-0.33	0.07
17	-1.24	-0.15	-0.15	1.13	-2.70	1.54	-2.66	0.01	1.87	-0.45
18	0.17	-0.61	1.55	-0.60	0.49	0.17	-0.03	-0.05	-1.76	-2.34
19	0.67	-0.30	-0.56	0.56	0.43	-0.84	-0.03	-0.57	-0.35	-0.28
20	-0.03	0.63	-0.65	0.14	1.31	-0.66	1.64	-0.93	0.11	-1.18
21	-0.19	0.90	0.14	-1.24	1.47	1.06	-2.00	-1.14	-0.54	0.03
22	-0.78	0.66	0.02	0.37	-0.14	0.35	-0.66	0.91	-0.41	-0.84
23	2.06	2.27	-0.50	-0.11	1.30	-0.90	0.88	-0.99	0.38	-0.12
24	0.75	1.17	-1.58	0.17	0.00	0.01	0.34	0.67	0.22	-0.74

Continued on next page

	1	2	3	4	5	6	7	8	9	10
25	1.82	0.29	0.03	0.66	-0.68	0.60	0.12	1.41	-0.67	-0.41
26	0.08	-0.66	-0.72	0.11	-0.85	1.72	-0.62	1.16	0.57	-0.38
27	-0.63	2.92	1.08	1.59	-0.35	-0.88	0.20	-0.14	0.18	1.33
28	-1.51	0.68	-0.95	0.12	0.23	-0.38	-0.24	-0.54	0.69	0.72
29	-0.64	-0.68	1.13	-1.25	-0.03	1.00	0.11	-1.08	-1.33	-0.23
30	0.23	0.19	-0.65	0.74	0.20	-0.46	-0.58	-0.02	-0.77	-1.23
31	1.01	-0.32	0.29	-1.22	0.20	1.46	1.19	-0.23	-0.19	1.05
32	0.25	-0.27	0.90	-1.92	0.71	1.04	1.06	0.16	0.96	-1.44
33	-1.17	-0.93	-0.52	-0.81	-0.27	-1.66	1.34	-0.60	-0.53	-1.18
34	0.67	0.12	0.55	-0.74	1.64	0.42	-0.37	0.63	-1.03	-0.78
35	-1.65	0.32	-0.09	-0.61	0.44	-1.42	-0.30	0.71	0.77	-0.97
36	-0.37	-1.08	-1.14	1.61	-1.19	-1.46	-0.35	1.08	0.47	-0.90
37	-0.32	-3.23	-0.27	1.64	-0.28	0.07	1.37	2.25	-1.23	1.29
38	-1.95	-0.25	1.62	0.73	0.52	0.06	-0.30	0.20	2.05	-0.11
39	0.92	0.03	-0.21	-1.27	-0.41	-0.03	0.10	-0.58	1.50	-0.72
40	-0.62	0.59	-0.82	0.02	0.34	1.26	0.38	-0.08	-1.39	0.67
41	-0.33	0.06	-0.05	0.49	0.43	0.29	0.68	-1.63	2.15	0.20
42	1.40	0.41	0.33	-0.14	1.30	-0.67	-0.27	0.13	-1.73	-1.22
43	0.64	-1.10	0.96	0.17	-0.65	0.12	2.35	-0.43	0.10	-0.56
44	-0.11	0.71	1.14	-0.94	-0.02	-0.51	-0.93	0.41	-1.14	0.53
45	0.51	0.72	0.10	-1.29	-0.57	-0.07	-1.43	-0.27	-0.38	1.95
46	0.40	0.25	1.16	1.41	-1.13	-1.15	0.04	-1.38	-0.35	1.58
47	1.66	1.36	-0.76	0.27	-1.85	-0.12	1.77	0.04	0.32	0.84
48	0.28	0.40	-2.34	-0.41	0.20	0.10	0.05	-1.35	-1.12	-0.01
49	0.51	0.26	-0.47	0.33	0.60	-0.70	0.11	0.90	-2.76	-1.73
50	0.35	0.27	-0.52	2.06	0.64	0.04	0.86	-1.83	-0.10	-0.73
51	-0.38	0.44	-2.32	0.36	-1.39	-0.65	-0.10	-0.14	-0.75	-0.22
52	0.10	1.06	0.56	1.41	-1.13	-0.64	0.10	0.46	0.21	0.49
53	1.64	0.45	-0.78	1.37	-1.14	-0.20	1.21	-1.52	-0.85	0.06
54	-0.88	0.66	-0.23	-0.41	-0.69	0.49	0.09	1.40	0.79	0.06
55	0.12	-1.14	-1.59	0.76	0.17	-0.21	-0.75	-0.89	0.03	1.84
56	1.36	-0.37	0.55	-0.65	0.59	0.19	0.39	-0.51	0.60	0.39
57	-0.23	1.48	1.89	-1.47	-0.82	0.97	-1.09	0.16	-0.99	-0.28
58	-1.05	-1.22	-0.88	-1.20	-2.86	-0.29	-1.46	-0.34	-0.21	-0.43
59	-0.87	0.26	-0.11	-0.15	0.95	-0.10	-0.12	-1.04	0.34	0.10
60	-0.39	0.41	1.95	1.80	0.62	0.51	-1.10	0.41	-0.09	-0.81
61	-0.85	0.98	0.93	0.10	-0.70	-1.26	0.58	-0.52	1.65	1.37
62	-0.26	-0.35	1.91	-0.80	-1.07	0.54	-0.15	0.51	-1.96	-0.29
63	-0.41	0.16	-0.01	0.23	-0.30	-0.53	-0.77	-0.50	0.76	-0.53
64	-0.18	-1.76	-0.15	0.70	0.11	-1.14	1.62	-0.96	0.01	-0.87
65	0.41	0.34	-0.51	-1.30	0.72	-1.77	-0.11	-0.09	0.95	0.94
66	0.62	-0.67	1.43	-1.05	1.39	-0.71	1.42	0.65	-0.10	-1.28
67	1.68	-0.24	-1.29	-1.94	-0.85	-1.17	-0.11	0.24	-2.74	-0.47
68	-0.07	-1.19	0.31	-1.27	0.29	-1.18	-0.33	-0.06	0.42	0.31
69	-0.32	0.38	-0.05	-0.89	1.33	1.63	0.37	-1.95	-0.22	-0.11
70	1.47	0.67	2.25	-0.29	-0.90	-1.21	1.03	1.45	-0.99	0.25
71	1.70	-0.30	-0.61	1.41	-0.22	1.00	2.71	-0.04	-0.66	-1.32
72	0.04	1.83	-1.51	0.89	-1.77	0.55	-1.03	-0.86	0.06	-0.43
73	-0.33	0.67	0.23	0.27	0.29	-0.88	-0.18	-0.20	2.45	-1.70
74	-1.82	0.95	-0.04	-0.57	-0.64	-0.61	1.08	0.26	0.19	-0.84
75	1.41	2.05	-0.84	-0.04	-0.92	-0.41	0.34	-1.52	-0.98	1.15
76	-0.84	-0.65	0.13	0.55	-0.78	0.19	-0.19	0.72	1.90	0.25
77	-1.12	0.81	-0.28	-0.50	0.44	-0.35	-1.30	0.25	-0.44	0.09
78	3.04	0.99	-0.68	1.06	-2.16	0.83	-0.28	1.20	-0.83	-0.93
79	0.24	-0.01	0.50	1.10	-0.43	-0.29	-0.18	0.99	-1.25	-0.56

Continued on next page

	1	2	3	4	5	6	7	8	9	10
80	-0.03	0.32	-0.33	-1.17	-1.01	-2.17	-0.08	-0.54	-0.67	-1.02
81	-2.73	-1.01	-1.83	-0.75	-0.96	-0.10	1.61	-0.49	1.94	0.05
82	-0.10	0.47	-2.65	1.21	0.47	-1.85	-0.46	-1.00	0.91	0.77
83	0.98	-0.70	-0.58	-1.69	0.35	0.94	-1.58	-0.36	0.74	0.30
84	0.41	0.81	1.45	0.42	-0.11	-0.75	0.50	0.94	0.15	-0.62
85	0.91	-0.81	0.84	0.23	0.54	-1.12	-0.11	1.79	0.28	0.55
86	1.98	0.32	1.22	3.20	-0.60	-0.59	-0.20	0.93	0.59	0.39
87	1.17	-0.85	0.98	-2.73	-2.02	0.59	0.64	-1.83	0.22	0.84
88	-0.51	-0.25	0.32	-0.84	1.15	-1.92	-2.91	0.82	0.16	-0.44
89	0.70	-1.55	-1.51	0.67	-0.23	0.25	-0.55	1.04	0.95	-1.18
90	-0.20	0.13	0.21	1.67	-0.37	-0.97	-0.15	0.62	-0.17	-0.08
91	-0.54	0.99	1.60	0.84	0.16	-0.01	0.61	-0.85	0.26	0.30
92	-2.86	0.18	-3.40	-1.43	0.77	0.68	-0.26	1.06	-0.90	1.56
93	-0.79	-1.77	-0.78	2.24	0.43	0.57	-1.01	0.73	1.58	-0.07
94	0.49	-0.62	1.10	-1.76	0.93	1.50	0.37	0.66	1.39	0.43
95	2.17	1.66	0.53	-1.11	0.05	0.13	0.02	-0.38	-1.81	-0.08
96	0.50	1.81	0.79	-0.04	-0.50	-0.70	0.11	0.85	-0.22	2.16
97	0.62	-1.18	0.46	2.23	-0.30	2.15	1.16	0.11	0.55	1.34
98	-0.97	-0.37	0.54	0.51	0.82	-0.30	-0.54	-0.07	0.48	-0.23
99	0.16	0.35	0.01	0.73	-0.85	0.81	-1.25	1.49	0.76	-0.13
100	-2.08	0.32	-0.92	1.73	-0.95	0.93	-1.28	0.76	-0.45	-0.02

Table 4: A `longtable` spanning several pages

5.9 Use of `add.to.row` argument

The following frequency table has outer dimnames: `Grade3` and `Grade6`.

```
Grade3 <- c("A","B","B","A","B","C","C","D","A","B",
            "C","C","C","D","B","B","D","C","C","D")
Grade6 <- c("A","A","A","B","B","B","B","B","C","C",
            "A","C","C","C","D","D","D","D","D","D")
Cohort <- table(Grade3, Grade6)
Cohort
```

```
##       Grade6
## Grade3 A B C D
##      A 1 1 1 0
##      B 2 1 1 2
##      C 1 2 2 2
##      D 0 1 1 2
```

The default behavior of `print.xtable` is to strip outer dimnames.

```
xtable(Cohort)
```

	A	B	C	D
A	1	1	1	0
B	2	1	1	2
C	1	2	2	2
D	0	1	1	2

The desired column labels can be created using `add.to.row`, in this case applying two commands to "row number zero" while suppressing the basic column names.

```
addtorow <- list()
addtorow$pos <- list(0, 0)
addtorow$command <- c("& \\multicolumn{4}{c}{Grade 6} \\\\\n",
                      "Grade 3 & A & B & C & D \\\\\n")
print(xtable(Cohort), add.to.row = addtorow, include.colnames = FALSE)
```

	Grade 6			
Grade 3	A	B	C	D
A	1	1	1	0
B	2	1	1	2
C	1	2	2	2
D	0	1	1	2

5.10 Sideways tables

Requires `\usepackage{rotating}` in the LaTeX preamble. Sideways tables can't be forced in place with the `[H]` specifier, but you can use the `\clearpage` command to get them fairly nearby.

```
x <- x[1:30, ]
x.side <- xtable(x, caption = "A sideways table")
print(x.side, floating = TRUE, floating.environment = "sidewaystable")
```

	1	2	3	4	5	6	7	8	9	10
1	0.41	0.49	-0.58	-1.23	0.98	0.31	0.45	-1.02	-2.87	-1.21
2	-0.47	0.70	-0.95	0.04	-1.22	0.61	1.43	-1.39	-0.43	0.30
3	0.07	0.19	-0.18	-0.42	0.71	-1.69	-0.21	-0.05	-1.49	-1.54
4	-0.50	0.70	1.01	-0.90	-0.11	0.78	-1.23	1.81	0.59	0.64
5	-0.83	0.31	0.02	0.42	1.78	0.01	-0.36	-0.10	-0.44	0.70
6	0.17	0.76	-0.65	0.15	-0.24	-0.18	-0.55	0.78	0.90	-1.91
7	-0.90	1.84	-0.50	1.46	-1.53	1.11	0.33	-1.10	0.47	0.94
8	0.17	1.11	1.61	-1.12	0.49	1.48	-0.93	-0.22	0.02	-0.22
9	0.35	0.03	-0.45	-0.52	0.35	-1.15	0.68	0.57	-0.09	-0.67
10	-0.05	-1.11	0.76	-0.07	-0.02	1.01	-1.16	-0.35	-0.36	0.45
11	-0.20	0.42	1.47	-1.41	-1.06	-0.63	1.76	0.79	0.20	1.28
12	-0.65	-0.40	0.44	-0.28	-0.84	0.13	1.45	0.69	-1.11	1.57
13	-1.11	1.49	-0.42	-0.71	-0.01	0.48	-0.45	-0.63	0.67	-1.20
14	0.85	-1.61	-0.04	-2.15	1.04	0.13	0.14	-0.77	-0.39	-0.44
15	0.02	-0.42	-0.49	-0.28	-0.36	-1.46	-0.37	2.36	-0.74	0.15
16	0.83	0.42	1.23	-0.53	-0.87	-0.51	-1.21	-0.19	-0.33	0.07
17	-1.24	-0.15	-0.15	1.13	-2.70	1.54	-2.66	0.01	1.87	-0.45
18	0.17	-0.61	1.55	-0.60	0.49	0.17	-0.03	-0.05	-1.76	-2.34
19	0.67	-0.30	-0.56	0.56	0.43	-0.84	-0.03	-0.57	-0.35	-0.28
20	-0.03	0.63	-0.65	0.14	1.31	-0.66	1.64	-0.93	0.11	-1.18
21	-0.19	0.90	0.14	-1.24	1.47	1.06	-2.00	-1.14	-0.54	0.03
22	-0.78	0.66	0.02	0.37	-0.14	0.35	-0.66	0.91	-0.41	-0.84
23	2.06	2.27	-0.50	-0.11	1.30	-0.90	0.88	-0.99	0.38	-0.12
24	0.75	1.17	-1.58	0.17	0.00	0.01	0.34	0.67	0.22	-0.74
25	1.82	0.29	0.03	0.66	-0.68	0.60	0.12	1.41	-0.67	-0.41
26	0.08	-0.66	-0.72	0.11	-0.85	1.72	-0.62	1.16	0.57	-0.38
27	-0.63	2.92	1.08	1.59	-0.35	-0.88	0.20	-0.14	0.18	1.33
28	-1.51	0.68	-0.95	0.12	0.23	-0.38	-0.24	-0.54	0.69	0.72
29	-0.64	-0.68	1.13	-1.25	-0.03	1.00	0.11	-1.08	-1.33	-0.23
30	0.23	0.19	-0.65	0.74	0.20	-0.46	-0.58	-0.02	-0.77	-1.23

Table 5: A sideways table

5.11 Rescaled tables

Specify a `scalebox` value to rescale the table.

```
x <- x[1:20, ]
x.rescale <- xtable(x)
print(x.rescale, scalebox = 0.7)
```

	1	2	3	4	5	6	7	8	9	10
1	0.41	0.49	-0.58	-1.23	0.98	0.31	0.45	-1.02	-2.87	-1.21
2	-0.47	0.70	-0.95	0.04	-1.22	0.61	1.43	-1.39	-0.43	0.30
3	0.07	0.19	-0.18	-0.42	0.71	-1.69	-0.21	-0.05	-1.49	-1.54
4	-0.50	0.70	1.01	-0.90	-0.11	0.78	-1.23	1.81	0.59	0.64
5	-0.83	0.31	0.02	0.42	1.78	0.01	-0.36	-0.10	-0.44	0.70
6	0.17	0.76	-0.65	0.15	-0.24	-0.18	-0.55	0.78	0.90	-1.91
7	-0.90	1.84	-0.50	1.46	-1.53	1.11	0.33	-1.10	0.47	0.94
8	0.17	1.11	1.61	-1.12	0.49	1.48	-0.93	-0.22	0.02	-0.22
9	0.35	0.03	-0.45	-0.52	0.35	-1.15	0.68	0.57	-0.09	-0.67
10	-0.05	-1.11	0.76	-0.07	-0.02	1.01	-1.16	-0.35	-0.36	0.45
11	-0.20	0.42	1.47	-1.41	-1.06	-0.63	1.76	0.79	0.20	1.28
12	-0.65	-0.40	0.44	-0.28	-0.84	0.13	1.45	0.69	-1.11	1.57
13	-1.11	1.49	-0.42	-0.71	-0.01	0.48	-0.45	-0.63	0.67	-1.20
14	0.85	-1.61	-0.04	-2.15	1.04	0.13	0.14	-0.77	-0.39	-0.44
15	0.02	-0.42	-0.49	-0.28	-0.36	-1.46	-0.37	2.36	-0.74	0.15
16	0.83	0.42	1.23	-0.53	-0.87	-0.51	-1.21	-0.19	-0.33	0.07
17	-1.24	-0.15	-0.15	1.13	-2.70	1.54	-2.66	0.01	1.87	-0.45
18	0.17	-0.61	1.55	-0.60	0.49	0.17	-0.03	-0.05	-1.76	-2.34
19	0.67	-0.30	-0.56	0.56	0.43	-0.84	-0.03	-0.57	-0.35	-0.28
20	-0.03	0.63	-0.65	0.14	1.31	-0.66	1.64	-0.93	0.11	-1.18

5.12 Aligning fixed width columns

Note that using specifications such as `p{2cm}` always produces a **left aligned** column. What if some other alignment is desired?

This is not really a problem with **xtable** but with the formatting of tables with fixed width columns and different alignments using standard LaTeX.

One solution is to use the `array` package, defining new column formats.

```
\newcolumntype{L}[1]{>{\raggedright\let\newline\\
    \arraybackslash\hspace{0pt}}m{#1}}
\newcolumntype{C}[1]{>{\centering\let\newline\\
    \arraybackslash\hspace{0pt}}m{#1}}
\newcolumntype{R}[1]{>{\raggedleft\let\newline\\
    \arraybackslash\hspace{0pt}}m{#1}}
\newcolumntype{P}[1]{>{\raggedright\tabularxbackslash}p{#1}}
```

These allow for very sophisticated cell formatting, namely left-aligned, centred, or right-aligned text, with recognition of line breaks for the first three new column types. If these lines are included along with `\usepackage{array}`, then the following is possible.

```
df <- data.frame(name = c("A","B"), right = c(1.4, 34.6),
                 left = c(1.4, 34.6), text = c("txt1","txt2"))
print(xtable(df, align = c("l", "|c", "|R{3cm}", "|L{3cm}", "| p{3cm}|")),
      floating = FALSE, include.rownames = FALSE)
```

name	right	left	text
A	1.40	1.40	txt1
B	34.60	34.60	txt2

5.13 Table width

The `tabularx` environment is for typesetting tables whose overall width is fixed. The column alignment code `X` denotes columns that will be stretched to achieve the desired table width. Requires `\usepackage{tabularx}` in the LaTeX preamble.

```
df.width <- data.frame(One = c("item 1", "A"), Two = c("item 2", "B"),
                       Three = c("item 3", "C"), Four = c("item 4", "D"))
x.width <- xtable(df.width)
align(x.width) <- "|l|X|l|l|l|"
print(x.width, tabular.environment = "tabularx", width = "\\textwidth")
```

	One	Two	Three	Four
1	item 1	item 2	item 3	item 4
2	A	B	C	D

6 Suppressing printing

By default the `print` method will print the LaTeX or HTML to standard output and also return the character strings invisibly. The printing to standard output can be suppressed by specifying `print.results = FALSE`.

```
x.out <- print(tli.table, print.results = FALSE)
```

Formatted output can also be captured without printing with the `toLatex` method. This function returns an object of class `"Latex"`.

```
x.ltx <- toLatex(tli.table)
class(x.ltx)
```

```
## [1] "Latex"
```

```
x.ltx
```

```
## % latex table generated in R 3.4.3 by xtable 1.8-3 package
## %
## \begin{tabular}{rrlllr}
##   \hline
##  & grade & sex & disadvg & ethnicty & tlimth \\
##   \hline
## 1 &   6 & M & YES & HISPANIC &  43 \\
##   2 &   7 & M & NO & BLACK &  88 \\
##   3 &   5 & F & YES & HISPANIC &  34 \\
##   4 &   3 & M & YES & HISPANIC &  65 \\
##   5 &   8 & M & YES & WHITE &  75 \\
##   6 &   5 & M & NO & BLACK &  74 \\
##   7 &   8 & F & YES & HISPANIC &  72 \\
##   8 &   4 & M & YES & BLACK &  79 \\
##   9 &   6 & M & NO & WHITE &  88 \\
##   10 &   7 & M & YES & HISPANIC &  87 \\
##    \hline
## \end{tabular}
```

7 Acknowledgements

Most of the examples in this gallery are taken from the **xtable** documentation. Two examples (`add.to.row` and 'Aligning fixed width columns') are from Stack Exchange.

8 Session information

```
toLatex(sessionInfo())
```

- R version 3.4.3 (2017-11-30), `x86_64-w64-mingw32`
- Locale: `LC_COLLATE=C`, `LC_CTYPE=English_New Zealand.1252`, `LC_MONETARY=English_New Zealand.1252`, `LC_NUMERIC=C`, `LC_TIME=English_New Zealand.1252`
- Running under: `Windows 7 x64 (build 7601) Service Pack 1`
- Matrix products: default
- Base packages: base, datasets, grDevices, graphics, methods, stats, utils
- Other packages: Matrix 1.2-12, knitr 1.20, sp 1.3-1, spData 0.2.8.3, spdep 0.7-8, sphet 1.7, splm 1.4-10, survival 2.42-3, xtable 1.8-3, zoo 1.8-1
- Loaded via a namespace (and not attached): Formula 1.2-3, LearnBayes 2.15.1, MASS 7.3-48, bdsmatrix 1.3-3, boot 1.3-20, coda 0.19-1, compiler 3.4.3, deldir 0.1-15, dotCall64 0.9-5.2, evaluate 0.10.1, expm 0.999-2, gdata 2.18.0, gmodels 2.16.2, grid 3.4.3, gtools 3.5.0, highr 0.6, ibdreg 0.2.5, lattice 0.20-35, lmtest 0.9-36, magrittr 1.5, maxLik 1.3-4, miscTools 0.6-22, nlme 3.1-137, parallel 3.4.3, plm 1.6-6, sandwich 2.4-0, spam 2.1-4, splines 3.4.3, stringi 1.1.7, stringr 1.3.1, tools 3.4.3