

# The **xtable** Gallery

Jonathan Swinton and others

October 21, 2015

## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Gallery</b>	<b>2</b>
2.1	Data frame . . . . .	2
2.2	Matrix . . . . .	2
2.3	aov . . . . .	3
2.4	lm . . . . .	3
2.5	glm . . . . .	4
2.6	precomp . . . . .	4
2.7	Time series . . . . .	5
<b>3</b>	<b>Automatic formatting</b>	<b>6</b>
3.1	Suggest alignment, digits, and display . . . . .	6
3.2	Shorthand notation . . . . .	6
<b>4</b>	<b>Sanitization</b>	<b>7</b>
4.1	Markup in tables . . . . .	8
<b>5</b>	<b>Format examples</b>	<b>8</b>
5.1	Adding a centering environment . . . . .	8
5.2	Column alignment . . . . .	9
5.3	Number of digits . . . . .	10
5.4	Suppress row/column names . . . . .	11
5.5	Rotate row/column names . . . . .	13
5.6	Horizontal lines . . . . .	14
5.7	Table level commands . . . . .	15
5.8	Long tables . . . . .	15
5.9	Use of <code>add.to.row</code> argument . . . . .	18
5.10	Sideways tables . . . . .	18
5.11	Rescaled tables . . . . .	20
5.12	Aligning fixed width columns . . . . .	20
5.13	Table width . . . . .	21
<b>6</b>	<b>Suppressing printing</b>	<b>21</b>
<b>7</b>	<b>Acknowledgements</b>	<b>22</b>
<b>8</b>	<b>Session information</b>	<b>22</b>

# 1 Introduction

This document gives a gallery of tables which can be made using the **xtable** package to create L<sup>A</sup>T<sub>E</sub>X 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)
```

## 2 Gallery

### 2.1 Data frame

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

	grade	sex	disadv	ethnicity	timth
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 + ethnicity + grade + disadv, data = tli)
xtable(fm1)
```

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

## 2.4 lm

```
fm2 <- lm(tlimth ~ sex*ethnicity, 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
ethnicityHISPANIC	-9.7614	6.5501	-1.49	0.1395
ethnicityOTHER	15.8636	10.8360	1.46	0.1466
ethnicityWHITE	4.7970	4.9687	0.97	0.3368
sexM:ethnicityHISPANIC	10.6780	8.7190	1.22	0.2238
sexM:ethnicityWHITE	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
ethnicity	3	2572.15	857.38	4.31	0.0068
sex:ethnicity	2	298.43	149.22	0.75	0.4748
Residuals	93	18480.04	198.71		

### Anova table (two models)

```
fm2b <- lm(tlimth ~ ethnicity, 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

## 2.5 glm

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

	Estimate	Std. Error	z value	Pr(> z )
(Intercept)	3.1888	1.5966	2.00	0.0458
ethnicityHISPANIC	-0.2848	2.4808	-0.11	0.9086
ethnicityOTHER	212.1701	22122.7093	0.01	0.9923
ethnicityWHITE	-8.8150	3.3355	-2.64	0.0082
grade	-0.5308	0.2892	-1.84	0.0665
ethnicityHISPANIC:grade	0.2448	0.4357	0.56	0.5742
ethnicityOTHER:grade	-32.6014	3393.4687	-0.01	0.9923
ethnicityWHITE: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
ethnicity	3	47.24	96	82.25
grade	1	1.73	95	80.52
ethnicity: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							1	2	4	5	7	9
1955	10	11	13	14	15	16	17	17	17	18	20	22
1956	22	23	25	26	28	30	30	30	31	31	31	32
1957	35	37	37	38	37	37	39	42	45	46	45	46
1958	45	48	49	49	51	50	51	53	55	56	56	56
1959	59	60	59	58	58	58	60	59	60	61	63	63
1960	65	64	63	65	65	67	68	68	69	70	72	72
1961	73	73	73	74	75	76	79	81	83	83	85	84
1962	83	85	87	88	88	90	90	91	93	95		

## 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.

```
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

## 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 <sup>-1</sup>
2	Value_is 10 <sup>-2</sup>
3	Value_is 10 <sup>-3</sup>

## 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\$)
$\alpha$	\$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
ethnicity	3	47.24	96	82.25
grade	1	1.73	95	80.52
ethnicity: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
ethnicity	3	47.24	96	82.25
grade	1	1.73	95	80.52
ethnicity: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	disadv	ethnicity	timth
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	disadv	ethnicity	timth
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	disadv	ethnicity	timth
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	disadv	ethnicity	timth
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	disadv	ethnicity	timth
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	disadv	ethnicity	timth
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	disadv	ethnicity	timth
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	disadv	ethnicity	timth
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 it 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 L<sup>A</sup>T<sub>E</sub>X preamble.

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

	grade	sex	disadv	ethnicity	timth
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
ethnicity	3	47.24	96	82.25
grade	1	1.73	95	80.52
ethnicity: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 L<sup>A</sup>T<sub>E</sub>X preamble.

The top and bottom rules are slightly thicker than the mid rule. The thickness of the lines can be set via the L<sup>A</sup>T<sub>E</sub>X lengths `\heavyrulewidth` and `\lightrulewidth`.

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

grade	sex	disadv	ethnicity	tlmth
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
2	2
3	3
4	4
5	5

## 5.7 Table level commands

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

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

## 5.8 Long tables

Requires `\usepackage{longtable}` in the L<sup>A</sup>T<sub>E</sub>X 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	-1.81	-0.38	0.26	0.53	-1.68	1.12	0.13	0.93	-0.55	0.87
2	-1.91	0.01	0.70	0.72	-0.15	0.45	0.36	0.31	0.52	0.96
3	-1.08	-0.99	-1.10	-0.12	-0.69	1.04	-1.91	0.47	-1.68	0.12
4	1.48	-0.64	1.13	-1.95	0.30	-0.68	1.25	0.48	-0.07	0.40
5	0.10	-0.14	-0.88	0.95	0.19	-0.09	-0.30	0.64	1.36	1.41
6	-0.55	1.60	-1.26	0.60	-0.16	-0.23	-3.23	-0.66	-1.00	-0.58
7	0.55	0.29	1.52	0.92	-0.81	0.15	-1.17	-0.24	0.49	-0.11
8	-0.48	2.19	-0.00	-0.26	-0.98	0.71	-1.70	0.53	0.57	-2.21
9	-0.41	-0.89	-0.87	-0.50	0.66	-1.47	-0.98	0.28	-0.80	0.11
10	1.06	-1.06	0.15	1.33	-1.14	0.22	1.00	-0.72	-0.94	-1.08
11	0.44	-0.14	-0.08	-1.06	-0.37	0.64	-1.25	0.27	-0.44	-1.86
12	0.92	0.82	0.21	-0.43	0.34	2.27	-1.36	1.08	1.62	0.38
13	-1.08	1.18	0.09	-0.64	-1.00	0.93	-0.17	-0.99	1.22	-0.42
14	-0.28	-0.67	1.30	1.00	-1.40	-0.59	-2.37	0.76	2.18	-1.46
15	-0.74	0.25	0.45	-1.96	-0.22	-0.44	-0.45	-0.66	0.01	0.42
16	0.32	1.87	-1.32	0.02	0.61	1.55	-0.90	-1.24	0.19	1.01
17	1.25	-1.40	1.24	-0.09	-1.73	-0.22	-0.89	0.26	0.33	-1.21
18	-0.06	-0.54	-1.13	0.80	-0.75	0.11	-1.24	0.27	-0.43	-0.78
19	-1.47	-0.11	-1.01	0.72	0.36	0.39	-0.37	-1.22	1.46	-1.13
20	1.23	0.06	-0.31	-1.34	-1.68	1.02	-1.27	-0.64	0.28	0.76
21	-0.24	0.92	2.20	-0.06	-0.99	0.89	0.47	0.22	-1.47	-0.97
22	-1.69	-0.08	0.60	-0.82	-0.57	-0.29	1.83	-1.96	0.22	-0.75
23	-1.64	-1.43	0.31	0.07	0.24	1.04	0.77	-1.04	-2.22	1.93
24	-0.08	1.07	0.85	1.04	-0.97	-0.09	-0.54	0.54	0.14	-1.34
25	0.59	1.48	-2.20	1.31	0.69	0.86	0.50	-0.15	0.42	-0.77
26	-0.75	-1.84	-0.42	-0.13	-1.02	0.42	-0.06	-1.12	-0.83	-0.67
27	1.32	-1.66	-0.68	-1.82	-0.56	-1.75	-1.68	-1.40	0.55	0.68
28	0.37	-0.16	0.76	0.19	0.26	-0.43	-0.97	-0.64	0.58	0.74
29	0.59	-0.14	-0.22	0.06	-0.14	1.85	1.83	0.05	-0.67	-0.82

30	-1.60	1.93	0.96	1.35	0.88	0.62	-0.86	-0.49	-0.96	0.67
31	0.88	0.73	-0.06	-0.88	0.68	0.06	1.39	1.29	-0.63	-0.54
32	-0.97	0.89	-0.02	0.64	-0.84	-0.52	1.29	0.97	-0.53	0.60
33	-0.07	1.89	0.38	2.08	-0.48	-0.90	0.72	-0.62	-0.54	0.95
34	0.48	-1.91	0.77	-0.19	-0.18	0.05	0.64	-0.91	-0.85	-1.82
35	0.79	-0.68	0.39	2.36	-0.72	-1.62	0.21	1.33	-0.14	-0.09
36	-0.86	1.31	0.44	-0.45	2.16	-0.42	-1.13	-2.79	0.09	-1.22
37	-2.35	0.24	-0.83	-0.41	1.55	-0.63	-0.27	-0.49	-0.12	0.71
38	2.01	1.96	1.50	0.37	0.11	-1.25	-2.07	-0.68	-0.08	0.81
39	0.12	-0.43	-0.57	1.24	0.37	0.13	-2.15	-0.17	1.05	-1.62
40	-0.77	-2.57	0.73	1.06	-0.05	1.65	1.86	-1.58	0.82	0.23
41	0.77	0.18	-1.25	0.10	-0.01	-0.61	-1.04	1.16	-0.99	1.33
42	-1.10	0.97	0.96	0.31	0.09	0.29	0.10	1.62	0.29	0.09
43	-0.40	-0.49	2.14	-0.37	0.89	1.04	1.71	-0.15	-0.05	-1.25
44	1.29	0.14	-1.29	-0.49	-2.16	-0.43	-0.07	-0.22	0.77	-0.12
45	0.87	-0.80	-0.50	-0.63	-0.65	0.60	0.02	0.07	1.47	-1.84
46	0.55	0.96	-0.59	0.57	0.11	0.32	-1.91	-0.08	0.79	-0.18
47	-0.63	-0.58	0.39	-0.67	1.20	2.12	-1.93	-2.50	-0.84	1.07
48	0.20	-0.02	-1.69	-1.51	-0.96	-0.32	0.72	-1.88	0.43	0.19
49	0.89	-1.87	0.28	2.28	1.83	0.04	0.26	-1.26	-1.82	-0.21
50	0.72	0.34	1.68	-0.97	-1.12	1.82	0.03	-0.07	-0.78	-0.80
51	-1.67	0.22	1.43	0.69	0.15	0.17	-1.66	-0.41	-0.25	1.20
52	0.75	0.05	-1.23	-1.28	0.40	-0.31	-3.50	-0.13	1.08	-0.30
53	-0.44	0.43	-0.66	-0.28	1.10	-0.55	0.00	1.64	-0.87	-1.79
54	0.25	-0.55	-0.62	0.49	-1.52	0.66	0.42	-0.20	-0.04	-0.01
55	-0.66	0.97	-1.50	-0.85	1.31	-0.62	-0.67	0.03	1.52	-0.96
56	0.50	0.26	-0.33	1.32	0.07	-1.26	0.65	-1.36	-0.13	0.23
57	-0.31	-0.25	-1.37	-2.10	1.71	-1.04	-0.75	-0.82	0.49	-1.33
58	0.54	0.01	0.29	-0.57	-0.03	-0.25	-1.55	-0.43	-0.07	-0.92
59	0.21	-0.26	1.29	0.53	1.89	0.52	-1.72	-1.19	-0.49	1.64
60	0.84	1.51	-0.16	-0.01	-0.13	-0.56	2.02	0.83	1.37	-1.19
61	-0.39	-0.38	-0.37	1.16	1.09	0.40	1.24	0.34	-0.00	-0.83
62	-0.79	-0.23	-0.24	-0.40	-1.17	-0.47	-0.24	-0.68	0.60	-0.97
63	-0.65	-0.56	0.46	-0.42	-0.22	1.55	-0.57	-1.92	-1.65	-0.06
64	1.04	-0.51	1.29	-0.47	0.43	-2.36	0.79	-0.21	0.73	-0.66
65	-1.14	-0.92	1.67	2.28	-0.54	1.77	-0.89	-0.34	-1.05	-0.64
66	0.05	-0.73	-0.12	-0.54	0.82	-0.83	-0.82	0.24	0.50	1.95
67	1.12	0.96	-0.16	0.35	0.74	0.12	-2.22	-0.11	-0.91	0.18
68	0.44	-0.68	-1.02	0.19	0.51	1.40	0.32	-0.25	-0.04	-0.21
69	0.55	-0.58	0.30	-0.42	1.17	-1.57	0.79	0.77	-1.53	-1.09
70	0.00	-0.01	0.06	0.27	0.82	-1.56	-0.40	-1.47	0.61	0.41
71	0.26	-1.66	-1.07	-1.11	-0.08	-1.49	-2.21	-0.17	2.70	0.04
72	0.39	0.25	-0.57	1.51	-0.43	0.46	1.83	-1.27	0.41	-0.23
73	-1.89	1.38	-0.08	-1.51	1.05	-0.49	-1.15	1.44	-0.20	-0.72
74	1.12	1.80	-0.70	-0.10	0.70	-0.00	-1.67	-0.08	-0.05	0.50
75	-1.75	-0.44	-0.18	0.01	-2.18	0.04	0.74	-0.02	0.80	-0.57
76	-1.76	0.88	-0.14	-1.92	-1.47	-0.96	-0.26	0.36	0.32	-1.00
77	0.35	-0.31	0.58	-0.40	1.23	-2.21	0.56	-0.03	-1.23	-1.69
78	0.30	1.10	-0.01	-0.16	-0.10	-1.44	0.58	-0.20	-0.54	0.21
79	0.45	0.35	0.12	0.07	1.60	-0.10	0.63	-0.52	0.27	0.84
80	1.00	0.14	-0.26	-0.75	0.72	0.57	1.01	1.50	0.04	-0.32
81	-0.48	1.04	0.23	-0.74	-1.02	-0.82	-0.44	0.54	-2.41	-1.94
82	-0.62	-0.46	-1.05	0.05	-0.84	-0.87	1.09	-0.62	1.76	1.29
83	-0.95	0.61	0.29	1.78	0.36	-1.99	-0.11	1.38	-0.56	-1.67
84	0.50	0.26	1.61	-1.31	-0.07	-1.12	1.08	0.33	-1.46	-2.16
85	-0.10	-1.05	-1.43	-0.96	0.14	0.38	1.33	-1.02	-1.02	1.50
86	-0.98	0.22	0.85	1.45	1.58	0.51	0.36	0.83	1.15	-0.78



87	-1.62	0.02	-0.46	1.24	-1.10	-0.22	1.03	1.09	-0.20	0.17
88	1.60	1.36	-1.94	-0.95	0.93	0.52	-2.63	0.01	1.42	0.23
89	-0.17	2.01	0.14	-0.58	-1.20	-0.03	0.32	-0.55	1.16	0.32
90	-0.45	0.49	0.44	0.90	0.89	-0.43	-1.87	-0.41	-0.08	-0.43
91	0.74	0.80	0.04	0.92	-1.16	0.52	-0.12	-0.00	-1.41	0.71
92	1.04	-0.34	-0.29	-1.17	0.10	-0.12	-1.11	0.32	0.79	0.24
93	0.42	-0.04	0.02	-0.20	-1.52	-1.04	-0.35	-1.71	0.75	1.46
94	0.38	-0.97	2.26	0.35	-0.40	1.88	0.06	0.47	0.19	-0.48
95	-0.34	-0.48	-0.05	-0.44	0.68	0.39	0.35	1.32	-0.97	-0.11
96	-1.80	0.05	0.28	0.10	-0.63	0.68	1.63	0.25	1.32	-0.73
97	-0.21	-0.98	-0.80	-0.61	0.63	-1.37	1.19	1.68	-0.51	-0.68
98	-1.33	-0.58	1.25	-1.34	0.79	-0.92	0.22	0.81	-1.79	0.45
99	-0.89	-0.85	-0.41	0.52	0.42	1.19	2.08	0.32	-0.50	-0.88
100	-0.96	-0.87	0.85	1.31	-1.39	0.48	2.21	0.70	-0.87	0.02

Table 3: 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	-1.81	-0.38	0.26	0.53	-1.68	1.12	0.13	0.93	-0.55	0.87
2	-1.91	0.01	0.70	0.72	-0.15	0.45	0.36	0.31	0.52	0.96
3	-1.08	-0.99	-1.10	-0.12	-0.69	1.04	-1.91	0.47	-1.68	0.12
4	1.48	-0.64	1.13	-1.95	0.30	-0.68	1.25	0.48	-0.07	0.40
5	0.10	-0.14	-0.88	0.95	0.19	-0.09	-0.30	0.64	1.36	1.41
6	-0.55	1.60	-1.26	0.60	-0.16	-0.23	-3.23	-0.66	-1.00	-0.58
7	0.55	0.29	1.52	0.92	-0.81	0.15	-1.17	-0.24	0.49	-0.11
8	-0.48	2.19	-0.00	-0.26	-0.98	0.71	-1.70	0.53	0.57	-2.21
9	-0.41	-0.89	-0.87	-0.50	0.66	-1.47	-0.98	0.28	-0.80	0.11
10	1.06	-1.06	0.15	1.33	-1.14	0.22	1.00	-0.72	-0.94	-1.08
11	0.44	-0.14	-0.08	-1.06	-0.37	0.64	-1.25	0.27	-0.44	-1.86
12	0.92	0.82	0.21	-0.43	0.34	2.27	-1.36	1.08	1.62	0.38
13	-1.08	1.18	0.09	-0.64	-1.00	0.93	-0.17	-0.99	1.22	-0.42
14	-0.28	-0.67	1.30	1.00	-1.40	-0.59	-2.37	0.76	2.18	-1.46
15	-0.74	0.25	0.45	-1.96	-0.22	-0.44	-0.45	-0.66	0.01	0.42
16	0.32	1.87	-1.32	0.02	0.61	1.55	-0.90	-1.24	0.19	1.01
17	1.25	-1.40	1.24	-0.09	-1.73	-0.22	-0.89	0.26	0.33	-1.21
18	-0.06	-0.54	-1.13	0.80	-0.75	0.11	-1.24	0.27	-0.43	-0.78
19	-1.47	-0.11	-1.01	0.72	0.36	0.39	-0.37	-1.22	1.46	-1.13
20	1.23	0.06	-0.31	-1.34	-1.68	1.02	-1.27	-0.64	0.28	0.76
21	-0.24	0.92	2.20	-0.06	-0.99	0.89	0.47	0.22	-1.47	-0.97
22	-1.69	-0.08	0.60	-0.82	-0.57	-0.29	1.83	-1.96	0.22	-0.75
23	-1.64	-1.43	0.31	0.07	0.24	1.04	0.77	-1.04	-2.22	1.93
24	-0.08	1.07	0.85	1.04	-0.97	-0.09	-0.54	0.54	0.14	-1.34
25	0.59	1.48	-2.20	1.31	0.69	0.86	0.50	-0.15	0.42	-0.77
26	-0.75	-1.84	-0.42	-0.13	-1.02	0.42	-0.06	-1.12	-0.83	-0.67
27	1.32	-1.66	-0.68	-1.82	-0.56	-1.75	-1.68	-1.40	0.55	0.68
28	0.37	-0.16	0.76	0.19	0.26	-0.43	-0.97	-0.64	0.58	0.74
29	0.59	-0.14	-0.22	0.06	-0.14	1.85	1.83	0.05	-0.67	-0.82
30	-1.60	1.93	0.96	1.35	0.88	0.62	-0.86	-0.49	-0.96	0.67

Table 4: 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	-1.81	-0.38	0.26	0.53	-1.68	1.12	0.13	0.93	-0.55	0.87
2	-1.91	0.01	0.70	0.72	-0.15	0.45	0.36	0.31	0.52	0.96
3	-1.08	-0.99	-1.10	-0.12	-0.69	1.04	-1.91	0.47	-1.68	0.12
4	1.48	-0.64	1.13	-1.95	0.30	-0.68	1.25	0.48	-0.07	0.40
5	0.10	-0.14	-0.88	0.95	0.19	-0.09	-0.30	0.64	1.36	1.41
6	-0.55	1.60	-1.26	0.60	-0.16	-0.23	-3.23	-0.66	-1.00	-0.58
7	0.55	0.29	1.52	0.92	-0.81	0.15	-1.17	-0.24	0.49	-0.11
8	-0.48	2.19	-0.00	-0.26	-0.98	0.71	-1.70	0.53	0.57	-2.21
9	-0.41	-0.89	-0.87	-0.50	0.66	-1.47	-0.98	0.28	-0.80	0.11
10	1.06	-1.06	0.15	1.33	-1.14	0.22	1.00	-0.72	-0.94	-1.08
11	0.44	-0.14	-0.08	-1.06	-0.37	0.64	-1.25	0.27	-0.44	-1.86
12	0.92	0.82	0.21	-0.43	0.34	2.27	-1.36	1.08	1.62	0.38
13	-1.08	1.18	0.09	-0.64	-1.00	0.93	-0.17	-0.99	1.22	-0.42
14	-0.28	-0.67	1.30	1.00	-1.40	-0.59	-2.37	0.76	2.18	-1.46
15	-0.74	0.25	0.45	-1.96	-0.22	-0.44	-0.45	-0.66	0.01	0.42
16	0.32	1.87	-1.32	0.02	0.61	1.55	-0.90	-1.24	0.19	1.01
17	1.25	-1.40	1.24	-0.09	-1.73	-0.22	-0.89	0.26	0.33	-1.21
18	-0.06	-0.54	-1.13	0.80	-0.75	0.11	-1.24	0.27	-0.43	-0.78
19	-1.47	-0.11	-1.01	0.72	0.36	0.39	-0.37	-1.22	1.46	-1.13
20	1.23	0.06	-0.31	-1.34	-1.68	1.02	-1.27	-0.64	0.28	0.76

## 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 L<sup>A</sup>T<sub>E</sub>X.

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 L<sup>A</sup>T<sub>E</sub>X 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 L<sup>A</sup>T<sub>E</sub>X 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 3.2.2 by xtable 1.8-0 package
## % Wed Oct 21 23:30:41 2015
## \begin{tabular}{rrlllr}
##   \hline
##   & grade & sex & disadvg & ethnicity & 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 3.2.2 (2015-08-14), 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
- Base packages: base, datasets, grDevices, graphics, methods, stats, utils
- Other packages: knitr 1.11, xtable 1.8-0
- Loaded via a namespace (and not attached): evaluate 0.8, formatR 1.2.1, highr 0.5.1, magrittr 1.5, stringi 0.5-5, stringr 1.0.0, tools 3.2.2