Package etable

Andreas Schulz

October 21, 2011

Some examples and explanations for this package.

1 Random Data

At first we need data to work with. Therefore we create a data.frame with random data.

```
> set.seed(31415)
> age <- round(runif(10000, 20, 80))
> dec <- cut(age, c(20, 30, 40, 50, 60, 70, 80))
> weight <- rnorm(10000, mean = 80, sd = 10)
> height <- rnorm(10000, mean = 1.6, sd = 0.1)
> bmi <- weight/height^2
> bmi_q <- cut(bmi, quantile(bmi, c(0, 0.25, 0.5, 0.75, 1)))
> sex <- factor(as.factor(rbinom(10000, 1, 0.5)), labels = c("Men", "Women"))
> ethnic <- factor(as.factor(rbinom(10000, 1, 0.75)), labels = c("Other", "Caucasian"))
> stage <- as.factor(rbinom(10000, 2, 0.3) + 1)
> disease <- factor(as.factor(rbinom(10000, 1, 0.1)), labels = c("no", "yes"))
> treat <- factor(as.factor(rbinom(10000, 1, 0.2)), labels = c("no", "yes"))
> ws <- abs(rnorm(10000))
> d <- data.frame(sex, age, dec, ethnic, weight, height, bmi, bmi_q, stage,
+     disease, treat, ws)
> print(str(d))
```

```
'data.frame':        10000 obs. of  12 variables:
 $ sex    : Factor w/ 2 levels "Men","Women": 1 1 1 1 2 2 1 1 2 1 ...
 $ age    : num  77 40 28 49 24 41 34 44 76 70 ...
 $ dec    : Factor w/ 6 levels "(20,30]","(30,40]",..: 6 2 1 3 1 3 2 3 6 5 ...
 $ ethnic : Factor w/ 2 levels "Other","Caucasian": 2 2 2 2 2 2 2 2 2 2 ...
 $ weight : num  95.8 90 92.3 72.6 91.9 ...
 $ height : num  1.46 1.61 1.69 1.5 1.67 ...
 $ bmi    : num  44.7 34.7 32.2 32.2 33.1 ...
 $ bmi_q  : Factor w/ 4 levels "(14.8,27.7]",..: 4 3 3 3 3 3 2 3 3 4 ...
 $ stage  : Factor w/ 3 levels "1","2","3": 1 2 1 2 2 2 2 2 3 2 ...
 $ disease: Factor w/ 2 levels "no","yes": 1 1 1 1 1 2 1 1 2 1 ...
 $ treat  : Factor w/ 2 levels "no","yes": 1 2 1 1 1 1 1 2 2 1 ...
 $ ws     : num  0.808 0.458 0.284 0.683 0.649 ...
NULL
```

2 Simple tables

2.1 2 × 1 table.

Here is a simple example for a table with predefined cell-function iqr_cell. We want to calculate median, Q1 and Q3 for bmi variable in data.frame d. We select a variable to calculate with, `x_vars='bmi'`. With `data=d` we set the data.frame. With `rows='sex'` we set the factor to separate the table in rows. Parameter `rnames='Sex'` gives only the label for the row factor.

```
> Tab <- tabular.ade(x_vars = "bmi", rows = "sex", rnames = "Sex", data = d,
+     FUN = iqr_cell)
> print(xtable(Tab, caption = "Median (Q1/Q3) of BMI"), include.rownames = F,
+     include.colnames = F, caption.placement = "top")
```

Table 1: Median (Q1/Q3) of BMI

Sex	
Men	31.2 (27.6/35.2)
Women	31.4 (27.8/35.2)

2.2 2 × 2 table.

For creating a 2 × 2 table, we need to specify a second factor for columns. Let it be ethnicity `cols='ethnic'` and set the label to `cnames='Ethnicity'`.

```
> Tab <- tabular.ade(x_vars = "bmi", rows = "sex", rnames = "Sex", cols = "ethnic",
+     cnames = "Ethnicity", data = d, FUN = iqr_cell)
> print(xtable(Tab, caption = "Median (Q1/Q3) of BMI"), include.rownames = F,
+     include.colnames = F, caption.placement = "top")
```

Table 2: Median (Q1/Q3) of BMI

	Sex		
Ethnicity		Other	Caucasian
	Men	31.2 (27.6/35.2)	31.3 (27.6/35.2)
	Women	31.5 (27.8/35.2)	31.3 (27.9/35.2)

2.3 n(nested) $\times$ 2 table.

We can use more then one factor for rows(cols) to create nested rows(columns). Use `rows=c('sex', 'dec'), rnames=c('Sex', 'Decades')` to make more complicated table.

```
> Tab <- tabular.ade(x_vars = "bmi", rows = c("sex", "dec"), rnames = c("Sex",
+     "Decades"), cols = "ethnic", cnames = "Ethnicity", data = d, FUN = iqr_cell)
> print(xtable(Tab, caption = "Median (Q1/Q3) of BMI"), include.rownames = F,
+     include.colnames = F, caption.placement = "top")
```

Table 3: Median (Q1/Q3) of BMI

	Sex	Decades		
Ethnicity			Other	Caucasian
	Men	(20,30]	31.5 (28.1/35.5)	30.9 (27.5/35.1)
		(30,40]	31.4 (27.4/35.3)	31.3 (27.5/34.9)
		(40,50]	31.1 (27.7/35.4)	31.1 (27.9/35.1)
		(50,60]	31.9 (27.7/35.2)	31.3 (27.7/34.9)
		(60,70]	31.2 (27.6/35.1)	31.4 (27.5/35.7)
		(70,80]	30.1 (26.7/34.4)	31.4 (27.7/35.2)
	Women	(20,30]	30.9 (26.8/35.8)	31.1 (27.6/35.8)
		(30,40]	31.6 (27.5/34.9)	31.1 (27.7/34.8)
		(40,50]	31.9 (27.8/36.0)	31.6 (28.0/35.5)
		(50,60]	31.0 (27.9/34.7)	31.3 (28.2/35.5)
		(60,70]	31.8 (28.2/35.0)	31.4 (27.8/34.9)
		(70,80]	31.7 (28.8/35.6)	31.2 (27.9/34.6)

2.4 n × n nested table.

Now we use an other cell function n_cell to save space for a big table. This function returns only the number of observation in each cell. Furthermore we use the factors for rows and cols in other way.

```
> Tab <- tabular.ade(x_vars = "bmi", rows = c("dec", "bmi_q"), rnames = c("Decades",
+     "BMI Quantiles"), cols = c("sex", "ethnic"), cnames = c("Sex", "Ethnicity"),
+     data = d, FUN = n_cell)
> print(xtable(Tab, caption = "N of Obs."), include.rownames = F, include.colnames = F,
+     caption.placement = "top")
```

Table 4: N of Obs.

	Decades	BMI Quantiles				
Sex			Men		Women	
Ethnicity			Other	Caucasian	Other	Caucasian
	(20,30]	(14.8,27.7]	47	173	64	155
		(27.7,31.3]	64	167	45	150
		(31.3,35.2]	57	150	42	125
		(35.2,65.2]	58	162	53	164
	(30,40]	(14.8,27.7]	50	172	57	157
		(27.7,31.3]	37	144	48	169
		(31.3,35.2]	48	164	67	161
		(35.2,65.2]	46	149	50	146
	(40,50]	(14.8,27.7]	63	156	53	145
		(27.7,31.3]	65	180	42	158
		(31.3,35.2]	55	153	56	157
		(35.2,65.2]	64	161	63	173
	(50,60]	(14.8,27.7]	55	137	47	144
		(27.7,31.3]	48	139	55	171
		(31.3,35.2]	63	142	57	152
		(35.2,65.2]	56	131	44	162
	(60,70]	(14.8,27.7]	49	167	47	151
		(27.7,31.3]	52	135	55	160
		(31.3,35.2]	46	145	58	177
		(35.2,65.2]	48	168	50	148
	(70,80]	(14.8,27.7]	60	150	39	141
		(27.7,31.3]	50	140	54	160
		(31.3,35.2]	33	154	57	163
		(35.2,65.2]	44	154	56	132

2.5 n × 1 table.

With cell function quantile_cell we can calculate any quantiles. Let's make a big table with only one column. With additional parameter `probs = 0.95` we choose the 95th quantile.

```
> Tab <- tabular.ade(x_vars = "bmi", xname = "BMI", rows = c("sex", "ethnic",
+     "disease", "treat"), rnames = c("Sex", "Ethnicity", "Disease", "Treatment"),
+     data = d, FUN = quantile_cell, probs = 0.95)
> print(xtable(Tab, caption = "95th quantile of BMI"), include.rownames = F,
+     include.colnames = F, caption.placement = "top")
```

Table 5: 95th quantile of BMI

Sex	Ethnicity	Disease	Treatment	
Men	Other	no	no	42.0
			yes	41.9
		yes	no	43.0
			yes	44.8
	Caucasian	no	no	41.4
			yes	42.8
		yes	no	43.2
			yes	42.0
Women	Other	no	no	41.3
			yes	41.7
		yes	no	42.4
			yes	39.0
	Caucasian	no	no	41.8
			yes	41.2
		yes	no	40.4
			yes	43.7

3 Predefined cell functions

There are several predefined cell functions in this package. Here is a list. See the help pages from the package for more information.

- **stat_cell**(*basic parameters*, digits=3, digits2=1)
- **combi_cell**(*basic parameters*, digits=3, style=1)
- **n_cell**(*basic parameters*, digits=0, type="n")
- **mean_sd_cell**(*basic parameters*, digits=3, style=1, nsd=1)
- **iqr_cell**(*basic parameters*, digits=3, add_n=FALSE)
- **quantile_cell**(*basic parameters*, digits=3, probs=0.5, plabels=FALSE)
- **eventpct_cell**(*basic parameters*, digits=1, digits2=0, event=2, type=1)
- **miss_cell**(*basic parameters*, pct=FALSE, digits=0, prefix="", suffix="")
- **corr_p_cell**(*basic parameters*, digits=3)
- **mode_cell**(*basic parameters*, digits=3)

  basic parameters are `x, y, z, w, cell_ids, row_ids, col_ids, vnames, vars, n_min` each cell function must take this parameters. They will be automatically passed from tabular.and function. Most of the functions uses only the `x` variable for calculations and w for weighted calculations. Only **corr_p_cell** and **stat_p_cell** uses `y` variable.
  Additional parameters like `digits = 3` can be used in **tabular.ade**`( ,...)` instead of the points.

4 Writing custom cell function

There is a possibilityl to write your own cell-function. It allows all possible designs of the cell and a lot more.

4.1 A example of an own custom cell function.

```
my_cell<- function(x, y, z, w, cell_ids, row_ids, col_ids, vnames, vars, n_min)
{
  out<- format(mean(x[cell_ids], na.rm=T), digits = 3)
  return(out)
}

tab<-tabular.ade(x_vars='age', rows='sex', rnames='Sex', cols='dec',
cnames='Decades', data=d, FUN=my_cell)
```

It must take the *basic parameters*: `x, y, z, w, cell_ids, row_ids, col_ids, vnames, vars, n_min` , but it can take more own parameters after the basic parameters.

Table 6: Mean Age

	Sex						
Decades		(20,30]	(30,40]	(40,50]	(50,60]	(60,70]	(70,80]
	Men	25.4	35.6	45.5	55.6	65.7	75.5
	Women	25.4	35.4	45.4	55.2	65.5	75.2

4.2 An other simple example of a own custom cell function.

```
my_cell<- function(x, y, z, w, cell_ids, row_ids, col_ids, vnames, vars, n_min)
{
  out<- NULL
  tab<-table(x[cell_ids])
  for(i in 1:length(tab)){
    out<- paste(out, levels(x)[i],': ' ,tab[i], sep='')
    if(i<length(tab)) out<- paste(out, ', ', sep='')
  }
  return(out)
}

tab<-tabular.ade(x_vars='sex', rows='dec', rnames='Decades', cols='stage',
cnames='Stage',  data=d, FUN=my_cell)
```

Table 7: Frequencies

	Decades			
Stage		1	2	3
	(20,30]	Men: 444, Women: 408	Men: 341, Women: 307	Men: 93, Women: 83
	(30,40]	Men: 404, Women: 418	Men: 341, Women: 357	Men: 65, Women: 81
	(40,50]	Men: 441, Women: 432	Men: 373, Women: 331	Men: 83, Women: 84
	(50,60]	Men: 369, Women: 427	Men: 330, Women: 330	Men: 72, Women: 75
	(60,70]	Men: 400, Women: 426	Men: 323, Women: 339	Men: 87, Women: 81
	(70,80]	Men: 391, Women: 365	Men: 321, Women: 347	Men: 73, Women: 90

4.3 More complicated cell function.

```
b_cell<- function(x, y, z, w, cell_ids, row_ids, col_ids, vnames, vars, n_min){
out<- NULL
if(length(unique(x))==2){
lv<-levels(x)
n <-sum(x[cell_ids]==lv[2])
N <-sum(table(x[cell_ids]))
out<- paste(levels(x)[2],': ' , format((n/N)*100, digits=3), '% (N:',n,')',sep='')
}
if(!is.factor(x) & length(unique(x))> 2){
quant <- format(quantile(x[cell_ids], c(0.25, 0.5, 0.75), na.rm=TRUE), digits=3)
out<- paste(quant[1], ' (',quant[2],'/',quant[3],')', sep='')
}
if(is.factor(x) & length(unique(x))> 2){
lv<-levels(x)
n <-table(x[cell_ids])
N <-sum(table(x[cell_ids]))
out<- paste(lv,': ', format((n/N)*100,  digits=3), '%', collapse=' | ', sep='')
}
return(out)
}

tab<-tabular.ade(x_vars=c('bmi','ethnic','stage'), xname=c('BMI','Ethnicity','Stages'
 cols='sex', cnames='Sex', data=d, FUN=base_cell)
```

Table 8: Diverse variables

Sex		Men	Women
	BMI	27.6 (31.2/35.2)	27.8 (31.4/35.2)
	Ethnicity	Caucasian: 74.5% (N:3713)	Caucasian: 74.8% (N:3754)
	Stages	1: 49.41% | 2: 41.04% | 3: 9.56%	1: 49.67% | 2: 40.45% | 3: 9.88%

5 Multivariable tables

5.1 T-test. Usage of x and y variables.

```
t_test_cell<- function(x, y, z, w, cell_ids, row_ids, col_ids, vnames, vars, n_min)
{
v <- x[cell_ids]
group <- y[cell_ids]
test<-t.test(v[which(group==levels(group)[1])], v[which(group==levels(group)[2])])
mdiff<- format(diff(test$estimate), digits=3)
p<- base:::format.pval(test$p.value, digits=2, eps=0.0001)
out<- paste('Diff: ', mdiff, ', p-value: ', p, sep='')
return(out)
}

tab<-tabular.ade(x_vars='bmi', xname='BMI', y_vars='ethnic', yname='Ethnicity',
rows='dec', rnames='Decades', cols='sex', cnames='Sex', data=d, FUN=t_test_cell)
```

Table 9: T test for BMI means between Ethnicity

	Decades		
Sex		Men	Women
	(20,30]	Diff: -0.483, p-value: 0.24	Diff: 0.302, p-value: 0.53
	(30,40]	Diff: -0.194, p-value: 0.71	Diff: -0.305, p-value: 0.47
	(40,50]	Diff: 0.171, p-value: 0.7	Diff: -0.0821, p-value: 0.86
	(50,60]	Diff: -0.442, p-value: 0.33	Diff: 0.185, p-value: 0.69
	(60,70]	Diff: -0.15, p-value: 0.76	Diff: -0.0931, p-value: 0.83
	(70,80]	Diff: 0.769, p-value: 0.13	Diff: -0.594, p-value: 0.19

5.2 Multiple x or y variables.

There's a possibility to pass more than one variable to `x_vars` or `x_vars`.
In this way we can create for example a correlation matrix.

```
> vars <- c("age", "weight", "height", "bmi")
> vlabels <- c("Age", "Weight", "Height", "BMI")
> tab <- tabular.ade(x_vars = vars, xname = vlabels, y_vars = vars, yname = vlabels,
+     data = d, FUN = corr_p_cell, digits = 2)
> print(xtable(tab, caption = "Pearson correlation"), include.rownames = F,
+     include.colnames = F, caption.placement = "top")
```

Table 10: Pearson correlation

	Age	Weight	Height	BMI
Age	1.0	0.0080	-0.0039	0.0084
Weight	0.0080	1.0	0.0058	0.69
Height	-0.0039	0.0058	1.0	-0.71
BMI	0.0084	0.69	-0.71	1.0

5.3 Multiple x with nested columns.

Or just multiple x variable. Then they will be listed line by line.

```
> vars <- c("age", "weight", "height", "bmi")
> vlabels <- c("Age", "Weight", "Height", "BMI")
> tab <- tabular.ade(x_vars = vars, xname = vlabels, cols = c("sex", "stage"),
+     cnames = c("Sex", "Stage"), data = d, FUN = quantile_cell)
> print(xtable(tab, caption = "Medians"), include.rownames = F, include.colnames = F,
+     caption.placement = "top")
```

Table 11: Medians

Sex		Men			Women		
Stage		1	2	3	1	2	3
	Age	49.0	49.0	50.0	50.0	51.0	50.0
	Weight	80.1	79.7	80.4	80.2	80.5	80.5
	Height	1.60	1.60	1.59	1.60	1.60	1.60
	BMI	31.2	31.3	31.4	31.3	31.4	31.5

6 Complex tables

6.1 The 'ALL' keyword.

The ALL keyword after a factor in rows or cols, adds additional group without separating in levels of this factor.

```
tab<-tabular.ade(x_vars='sex', rows=c('treat', 'ALL'), rnames=c('Treatment'),
cols=c('disease', 'ALL'), cnames=c('Disease'), data=d, FUN=n_cell, alllabel='both')
```

Table 12: Contingency table

	Treatment			
Disease		no	yes	both
	no	7159	833	7992
	yes	1798	210	2008
	both	8957	1043	10000

6.2 Weighted tables.

Most of predefined cell functions support weighting with `w=weights`.

```
tab<-tabular.ade(x_vars='sex', rows=c('sex', 'ALL', 'ethnic', 'stage'),
rnames=c('Sex','Ethnicity', 'Stage'), w='ws', data=d, FUN=n_cell, digits=1)
```

Table 13: weighted N

Sex	Ethnicity	Stage	
Men	Other	1	486.1
		2	408.0
		3	95.8
	Caucasian	1	1459.6
		2	1224.7
		3	283.6
Women	Other	1	504.5
		2	413.3
		3	107.0
	Caucasian	1	1504.3
		2	1212.1
		3	295.2
All	Other	1	990.5
		2	821.2
		3	202.7
	Caucasian	1	2963.9
		2	2436.7
		3	578.8

6.3 Various statistics in a table.

The predefined cell functions **stat_cell** can calculate several statistics. You can choose the statistics with keywords in `x_vars` or `y_vars`.

```
vars    <-c('age', 'weight', 'height', 'bmi')
vlabels <-c('Age', 'Weight', 'Height', 'BMI')

keywords  <-c('MIN', 'MAX', 'MEAN', 'SD', 'CV', 'SKEW',     'KURT')
keylabels <-c('Min', 'Max', 'Mean', 'SD', 'CV', 'Skewness', 'Kurtosis')

tab<-tabular.ade(x_vars=vars, xname=vlabels, y_vars=keywords, yname=keylabels,
     data=d, FUN=stat_cell)
```

Table 14: Various statistics

	Min	Max	Mean	SD	CV	Skewness	Kurtosis
Age	20.0	80.0	49.9	17.4	0.348	0.0150	-1.20
Weight	35.2	115	80.0	10.0	0.125	-0.0475	0.0314
Height	1.19	2.02	1.60	0.101	0.0630	0.0107	0.0500
BMI	14.8	65.2	31.7	5.69	0.180	0.484	0.671

Or combined with `rows` parameter.

```
keywords  <-c('N', 'MIN', 'MAX', 'MEAN', 'SD')
keylabels <-c('N', 'Min', 'Max', 'Mean', 'SD')

tab<-tabular.ade(x_vars=vars, xname=vlabels, y_vars=keywords, yname=keylabels,
     rows=c('sex', 'ALL', 'ethnic'), rnames=c('Sex', 'Ethnicity'),
     data=d, FUN=stat_cell)
```

Table 15: Various statistics

	Sex	Ethnicity					
			N	Min	Max	Mean	SD
Age	Men	Other	1268	20.0	80.0	49.7	17.2
		Caucasian	3713	20.0	80.0	49.8	17.6
	Women	Other	1265	20.0	80.0	50.1	17.3
		Caucasian	3754	20.0	80.0	50.1	17.2
	All	Other	2533	20.0	80.0	49.9	17.3
		Caucasian	7467	20.0	80.0	50.0	17.4
Weight	Men	Other	1268	44.2	110	80.0	9.98
		Caucasian	3713	41.1	114	79.8	10.1
	Women	Other	1265	43.1	111	80.5	10.0
		Caucasian	3754	35.2	115	80.0	9.97
	All	Other	2533	43.1	111	80.2	10.0
		Caucasian	7467	35.2	115	79.9	10.0
Height	Men	Other	1268	1.19	1.98	1.60	0.103
		Caucasian	3713	1.23	2.02	1.60	0.101
	Women	Other	1265	1.24	1.98	1.60	0.0989
		Caucasian	3754	1.21	1.95	1.60	0.101
	All	Other	2533	1.19	1.98	1.60	0.101
		Caucasian	7467	1.21	2.02	1.60	0.101
BMI	Men	Other	1268	15.0	58.7	31.7	5.88
		Caucasian	3713	15.8	65.2	31.6	5.68
	Women	Other	1265	15.5	64.9	31.8	5.64
		Caucasian	3754	14.8	58.6	31.7	5.66
	All	Other	2533	15.0	64.9	31.7	5.76
		Caucasian	7467	14.8	65.2	31.7	5.67

Now using the keywords in `x_vars`.

```
keywords  <-c('N', 'MIN', 'MAX', 'MEAN', 'SD')
keylabels <-c('N', 'Min', 'Max', 'Mean', 'SD')

tab<-tabular.ade(x_vars=keywords, xname=keylabels, y_vars=vars, yname=vlabels,
     rows=c('sex', 'ALL'), rnames=c('Sex'),
     data=d, FUN=stat_cell)
```

Table 16: Various statistics

	Sex				
		Age	Weight	Height	BMI
N	Men	4981	4981	4981	4981
	Women	5019	5019	5019	5019
	All	10000	10000	10000	10000
Min	Men	20.0	41.1	1.19	15.0
	Women	20.0	35.2	1.21	14.8
	All	20.0	35.2	1.19	14.8
Max	Men	80.0	114	2.02	65.2
	Women	80.0	115	1.98	64.9
	All	80.0	115	2.02	65.2
Mean	Men	49.8	79.9	1.60	31.6
	Women	50.1	80.1	1.60	31.7
	All	49.9	80.0	1.60	31.7
SD	Men	17.5	10.0	0.101	5.73
	Women	17.3	9.98	0.100	5.66
	All	17.4	10.0	0.101	5.69

6.4 Now all together.

A weighted, multivariable, nested table with several statistics.

```
vars    <-c('age', 'weight', 'height', 'bmi')
vlabels <-c('Age', 'Weight', 'Height', 'BMI')

keywords  <-c('N', 'MEDIAN', 'IQR')
keylabels <-c('N', 'Median', 'IQR')

tab<-tabular.ade(x_vars=vars, xname=vlabels, y_vars=keywords, yname=keylabels,
     rows=c('sex', 'ALL'), rnames=c('Sex'),
     cols=c('ethnic'), cnames=c('Ethnicity'),
     w='ws', data=d, FUN=stat_cell)
```

Table 17: Various statistics

		Sex						
			N		Median		IQR	
Ethnicity			Other	Caucasian	Other	Caucasian	Other	Caucasian
	Age	Men	990	2968	49.0	49.0	29.0	31.0
		Women	1025	3012	50.0	50.0	30.0	29.0
		All	2015	5979	50.0	49.0	30.0	30.0
	Weight	Men	990	2968	80.3	80.1	13.2	13.3
		Women	1025	3012	80.6	80.2	12.9	13.3
		All	2015	5979	80.5	80.2	13.2	13.4
	Height	Men	990	2968	1.60	1.60	0.140	0.137
		Women	1025	3012	1.60	1.60	0.122	0.139
		All	2015	5979	1.60	1.60	0.132	0.138
	BMI	Men	990	2968	31.2	31.3	7.99	7.50
		Women	1025	3012	31.4	31.3	7.53	7.18
		All	2015	5979	31.3	31.3	7.77	7.35