Journal of Statistical Software
August 2009, Volume 31, Code Snippet 1. http://www.jstatsoft.org/

reporttools: R Functions to Generate LaTeX Tables of Descriptive Statistics

Kaspar Rufibach
University of Zurich

Abstract

In statistical analysis reports, tables with descriptive statistics are routinely presented. We introduce the R package **reporttools** containing functions to efficiently generate such tables when compiling statistical analyses reports by combining LaTeX and R via `Sweave`.

Keywords: reporting, descriptive statistics, `Sweave`.

1. Introduction

In statistical analysis reports and medical publications it is common practice to start statistical analyses with displays of descriptive statistics of patient characteristics and further important variables. The purpose of these tables is (1) to get an idea about basic features of the data and (2) especially in analysis reports, data checking. Tables of descriptive statistics may take different formats, depending on the type of variables to be displayed, which descriptive statistics are to be reported, and whether statistics should be given for all observations jointly or separately for the levels of a given factor, such as e.g. treatment arm. To be able to efficiently generate these recurring parts of analyses when combining LaTeX (Knuth 1984; Lamport 1994) with R (R Development Core Team 2009) code via `Sweave` (Leisch 2002), the R package **reporttools** provides functions to generate LaTeX tables of descriptive statistics for nominal, date, and continuous variables. The tables are set up as data frames in R, then translated into LaTeX code using the standard R package **xtable** (Dahl 2009). Using `Sweave`, these tables can be directly generated in LaTeX documents by invoking basically only one line of R code. The package is available from the Comprehensive R Archive Network at `http://CRAN.R-project.org/package=reporttools`.

A package that offers somewhat related functionality is **r2lUniv** (Genolini 2009). In that package, each variable gets a separate section providing descriptive statistics and corresponding

plots, depending on the type of variable that is analyzed. The default functions in **r2lUniv** directly generate an entire LaTeX document, thereby reporting the descriptive statistics of only about three variables on one single page. Set up this way, it seems difficult to efficiently merge plain text and data analysis using **r2lUniv**. Additionally, we are not aware of a possibility in **r2lUniv** to compare a given variable between different groups in an easy way. The functions introduced in **reporttools** aim at closing these gaps. The tabulating functions in **reporttools** can be applied inside a `.Rnw` document combining LaTeX plain text and data analyses in R.

A common feature in data analysis is, that one needs to provide descriptive statistics of a large set of variables thereby generating tables that are larger than one page. The functions described here have by default the `tabular.environment`-option in the implicitly used R function `print.xtable` set to `longtable`. Together with the LaTeX package **longtable**, setting this option generates tables that may range over more than a page, without additional specification or programming effort.

The package **reporttools** contains several additional functions useful in setting up analyses and writing reports. However, the emphasis in this article is on the tabulating functions. For a more detailed description of further elements of the package we refer to its R help files.

In Section 2 we introduce the dataset that is used to illustrate the tabulating functions in Section 3. Some conclusions are drawn in Section 4.

2. Stanford heart transplantation data

We illustrate our new functions using the Stanford heart transplantation dataset `jasa` in the standard R package **survival** (Therneau and Lumley 2009). This dataset provides some patient characteristics and the survival of patients on the waiting list for the Stanford heart transplantation program, see Crowley and Hu (1977) and the corresponding R help file for details.

3. Descriptive statistics for heart transplantation data

To demonstrate the entire flexibility of **reporttools** we use the package to describe the Stanford heart transplantation data.

3.1. Nominal variables (factors)

The following code lines invoke **reporttools** and prepare the variables from `jasa` for later use.

```
R> library("reporttools")
R> data("heart", package = "survival")
R> vars0 <- with(jasa, data.frame(
+    "Transplantation" = factor(jasa$transplant, levels = 0:1, labels =
+      c("no", "yes")),
+    "Age" = jasa$age,
+    "Surgery" = factor(jasa$surgery, levels = 0:1, labels = c("no", "yes")),
+    "Survival status" = factor(jasa$fustat, levels = 0:1,
+      labels = c("alive", "dead")),
```

Variable	Levels	n	%	$\sum$ %
Surgery	no	87	84.5	84.5
	yes	16	15.5	100.0
	all	103	100.0	
Survival status	alive	28	27.2	27.2
	dead	75	72.8	100.0
	all	103	100.0	
HLA A2 score	0	48	73.8	73.8
	1	17	26.1	100.0
	all	65	100.0	

Table 1: Patient characteristics: nominal variables.

```
+    "HLA A2 score" = jasa$hla.a2,
+    "Birthday" = jasa$birth.dt,
+    "Acceptance into program" = jasa$accept.dt,
+    "End of follow up" = jasa$fu.date,
+    "Follow up time" = futime,
+    "Mismatch score" = mscore,
+    check.names = FALSE))
R> attach(vars0, warn.conflicts = FALSE)
```

To generate Table 1, a summary of some nominal variables of the heart transplantation data, simply invoke:

```
R> vars1 <- vars0[, c("Surgery", "Survival status", "HLA A2 score")]
R> cap1 <- "Patient characteristics: nominal variables."
R> tableNominal(vars = vars1, cap = cap1, vertical = FALSE, lab =
+    "tab: nominal1", longtable = FALSE)
```

If one wants to inspect differences in the distributions of patient characteristics between patients finally receiving or not receiving a transplantation, one simply needs to specify the option `group = Transplantation` to get absolute and relative frequencies per transplantation status, and for all patients jointly.

Furthermore, by specifying either `print.pval = "fisher"` or `print.pval = "chi2"` we can ask for a p value of either a Fisher's exact or a χ^2 test comparing the distributions between the groups defined by `group`. If interested in frequencies for patients not older than 50 years, we can set the option `subset = (Age <= 50)`. Then, for the HLA A2 score, we consider missing values a category in computation of percentages. To this end, we assign `miss.cat` a vector containing the indices of the factor(s) in `vars` we want the percentages to be computed this way. All these options are implemented in Table 2 via the following code lines:

```
R> cap2 <- "Patient characteristics: nominal variables, by transplantation,
+    patients not older than 50, missings as a separate category for the
+    3rd factor, $p$~values of Fisher's exact test added."
R> tableNominal(vars = vars1, group = Transplantation, subset = (Age <= 50),
```

Variable	**Levels**	n_{no}	$\%_{no}$	$\sum \%_{no}$	n_{yes}	$\%_{yes}$	$\sum \%_{yes}$	n_{all}	$\%_{\text{all}}$	$\sum \%_{\text{all}}$
Surgery	no	21	91.3	91.3	35	76.1	76.1	56	81.2	81.2
	yes	2	8.7	100.0	11	23.9	100.0	13	18.8	100.0
p = 0.19	all	23	100.0		46	100.0		69	100.0	
Survival status	alive	4	17.4	17.4	21	45.6	45.6	25	36.2	36.2
	dead	19	82.6	100.0	25	54.4	100.0	44	63.8	100.0
p = 0.033	all	23	100.0		46	100.0		69	100.0	
HLA A2 score	0	0	0.0	0.0	35	76.1	76.1	35	50.7	50.7
	1	0	0.0	0.0	8	17.4	93.5	8	11.6	62.3
	missing	23	100.0	100.0	3	6.5	100.0	26	37.7	100.0
p = 5e-04	all	23	100.0		46	100.0		69	100.0	

Table 2: Patient characteristics: nominal variables, by transplantation, patients not older than 50, missings as a separate category for the 3rd factor, p values of Fisher's exact test added.

```
+    miss.cat = 3, print.pval = "fisher", cap = cap2, lab = "tab: nominal2",
+    longtable = FALSE)
```

It may happen that one prefers a different ordering or naming of the levels of a factor in a table. For example, suppose that in Table 1 frequencies of the patients that died should precede those who are alive. We suggest to reverse this ordering via re-coding of the corresponding factor variable, by appropriately specifying the options `levels` and `labels` in R's command `factor`. Recoding factors is also recommended if one wants to change factor labels in a table.

By default, vertical lines are added to the plot. These can be omitted by specifying `vertical = FALSE` in the function call, as is done in the call for Table 1. Finally, a vector that attaches a weight to each observation can be assigned to the option `weights`.

3.2. Date variables

The primary purpose to report descriptive statistics for date variables is data checking. The implemented function `tableDate` provides number of observations (n), minimum (Min), first quartile (q_1), median ($\widetilde{x}$), mean ($\bar{x}$), third quartile (q_3), maximum (Max), and number of missing values (#NA). If not all of these statistics need to be reported, the desired columns can be specified via `stats`. Simply provide a sub-vector of `c("n", "min", "q1", "median", "mean", "q3", "max", "na")` giving the statistics you want to be displayed, in the order they should appear in the table. As an illustration, we only display n, Min, Max, and #NA for the variables birth date, acceptance date, and end of follow up date of the heart trial in Table 3, via the following code:

```
R> vars3 <- vars0[, c("Birthday", "Acceptance into program",
+    "End of follow up")]
R> cap3 <- "Patient characteristics: date variables, by transplantation
+    status."
R> tableDate(vars = vars3, group = Transplantation, stats =
+    c("n", "min", "max", "na"), print.pval = TRUE, cap = cap3, lab =
+    "tab: date1", longtable = FALSE)
```

Variable	Levels	n	Min	Max	#NA
Birthday	no	34	1909-10-04	1960-07-21	0
	yes	69	1905-04-02	1952-09-03	0
p = 0.47	all	103	1905-04-02	1960-07-21	0
Acceptance into program	no	34	1967-09-13	1974-03-22	0
	yes	69	1968-01-06	1974-02-22	0
p = 0.06	all	103	1967-09-13	1974-03-22	0
End of follow up	no	34	1967-09-18	1974-04-01	0
	yes	69	1968-01-21	1974-04-01	0
p = 0.00016	all	103	1967-09-18	1974-04-01	0

Table 3: Patient characteristics: date variables, by transplantation status.

As for `tableNominal`, the `vars` argument of the function is a data frame containing the variables of interest, but here each of R class `Date`. One can ask for a p value of a Mann-Whitney test (or Kruskal-Wallis test, depending on the number of levels of the grouping variable), to assess whether distributions are different between the groups defined by a given factor. This is especially relevant when analyzing data of a randomized trial, to verify whether patient characteristics are indeed equally distributed between randomized treatment arms.

3.3. Continuous variables

Compared to `tableDate`, the function `tableContinuous` to display continuous variables additionally provides standard deviation (s) and interquartile range (IQR) as default statistics. When using `tableContinuous`, all variables in the data frame given as the `vars` argument to this function must be `numeric`. Again, via `stats` one can choose which statistics to display, out of the options `c("n", "min", "q1", "median", "mean", "q3", "max", "s", "iqr", "na")`. To provide even more flexibility in the choice of descriptive statistics, user-defined functions can be supplied to `stats` in `tableContinuous`. Such a user-defined function must take a vector as an argument and return a single number (the desired statistic). Missing values are removed by default. For illustration, we add a trimmed mean and the coefficient of variation c_v to the chosen default statistics in Table 5. The number of decimal places the descriptive statistics are displayed with can be set using `prec`. Note that this option does not affect the columns `n` and `#NA`, their entries are always given as whole numbers. Finally, by specifying `print.pval` one can ask for a p value of an F, t, Kruskal-Wallis, or Mann-Whitney test, as appropriate, to compare the variable under consideration between the groups given by `group`.

We give two examples summarizing descriptive statistics of some continuous patient characteristics of the Stanford heart transplantation data. Table 4 is generated via

```
R> vars4 <- vars0[, c("Age", "Follow up time", "Mismatch score")]
R> cap4 <- "Patient characteristics: continuous variables."
R> tableContinuous(vars = vars4, cap = cap4, lab = "tab: cont1",
+    longtable = FALSE)
```

Variable	n	Min	q_1	$\widetilde{x}$	$\bar{x}$	q_3	Max	s	IQR	#NA
Age	103	8.8	41.2	47.8	45.2	52.1	64.4	9.8	10.9	0
Follow up time	103	0.0	32.5	89.0	309.2	411.5	1799.0	428.3	379.0	0
Mismatch score	65	0.0	0.8	1.1	1.2	1.6	3.0	0.6	0.8	38

Table 4: Patient characteristics: continuous variables.

Variable	**Levels**	n	Min	$\widetilde{x}$	$\bar{x}_{\text{trim}}$	Max	IQR	c_v	s	#NA
Age	no	34	8.8	46.3	44.0	59.1	12.5	0.3	11.4	0
	yes	69	19.6	48.0	46.4	64.4	9.5	0.2	8.9	0
p = 0.39	all	103	8.8	47.8	45.8	64.4	10.9	0.2	9.8	0
Follow up time	no	34	0.0	20.0	57.9	1400.0	41.5	2.6	250.5	0
	yes	69	4.0	206.0	374.7	1799.0	548.0	1.1	458.9	0
p = 1.1e-08	all	103	0.0	89.0	259.0	1799.0	379.0	1.4	428.3	0
Mismatch score	no	NA	NA	NA	NA	NA	NA	NA	NA	NA
	yes	65	0.0	1.1	1.1	3.0	0.8	0.5	0.6	4
	all	65	0.0	1.1	1.1	3.0	0.8	0.5	0.6	38

Table 5: Patient characteristics, by transplantation: continuous variables, user-defined functions supplied.

Finally, Table 5 can be produced via:

```
R> cap5 <- "Patient characteristics, by transplantation: continuous
+    variables, user-defined functions supplied."
R> stats <- list("n", "min", "median", "$\\bar{x}_{\\mathrm{trim}}$" =
+    function(x){return(mean(x, trim = .05))}, "max", "iqr",
+    "c$_{\\mathrm{v}}$" = function(x){return(sd(x) / mean(x))}, "s", "na")
R> tableContinuous(vars = vars4, group = Transplantation, stats = stats,
+    print.pval = "kruskal", cap = cap5, lab = "tab: cont2", longtable =
+    FALSE)
```

4. Conclusions

In this article, we present some functions in the **reporttools** package for R that facilitate the presentation of descriptive statistics of nominal, date, and continuous variables when writing reports using `Sweave`. The package is available from CRAN.

Acknowledgments

I thank the editor and two referees for constructive comments that led to an improvement of **reporttools** and the presentation of the paper. I also thank Leo Held and Philipp Muri for helpful discussions.

References

Crowley J, Hu M (1977). "Covariance Analysis of Heart Transplant Survival Data." *Journal of the American Statistical Association*, **72**, 27–36.

Dahl DB (2009). ***xtable**: Export Tables to LaTeX or HTML*. R package version 1.5-5, URL http://CRAN.R-project.org/package=xtable.

Genolini C (2009). ***r2lUniv**: R to LaTeX, Univariate Analysis*. R package version 0.9.2, URL http://CRAN.R-project.org/package=r2lUniv.

Knuth DE (1984). *The TeXbook*, volume A of *Computers and Typesetting*. Addison-Wesley, Reading, Massachusetts.

Lamport L (1994). *LaTeX: A Document Preparation System*. 2nd edition. Addison-Wesley, Reading, Massachusetts.

Leisch F (2002). "Dynamic Generation of Statistical Reports Using Literate Data Analysis." In W Härdle, B Rönz (eds.), *COMPSTAT 2002 – Proceedings in Computational Statistics*, pp. 575–580. Physica Verlag, Heidelberg.

R Development Core Team (2009). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org.

Therneau T, Lumley T (2009). ***survival**: Survival Analysis, Including Penalised Likelihood*. R package version 2.35-4, URL http://CRAN.R-project.org/package=survival.

Affiliation:

Kaspar Rufibach
Biostatistics Unit
Institute for Social and Preventive Medicine
University of Zurich
8001 Zurich, Switzerland
Telephone: +41/0/44634-4643
Fax: +41/0/44634-4386
E-mail: kaspar.rufibach@ifspm.uzh.ch
URL: http://www.biostat.uzh.ch/aboutus/people/rufibach.html

Journal of Statistical Software http://www.jstatsoft.org/
published by the American Statistical Association http://www.amstat.org/
Volume 31, Code Snippet 1 *Submitted:* 2008-11-13
August 2009 *Accepted:* 2009-05-05