Tables That Work for HTML and Word

2024-06-03

Packages such as gt and flextable produce highly flexible and beautiful HTML tables, but using features such as math in them will not result in a properly rendered Word .docx file. Using R functions that render HTML that is simple but flexible results in well-rendered Word and HTML. Here is an example using the Hmisc package summaryM function and its print method, which uses the [htmlTable package](https://cran.r-project.org/web/packages/htmlTable) by Max Gordon, which you can use in this context to produce a wide variety of nice tables.

Quarto allows you to include Markdown inside the html as described [here](https://quarto.org/docs/authoring/tables.html#html-tables). I define a little function mk to facitate that.

require(Hmisc)  
options(prType='html')  
mk <- function(x) paste0('<span data-qmd="', x, '"></span>')  
  
getHdata(support)  
d <- support  
label(d$age) <- mk('Age \_in years\_ $\\alpha\_{3}$')  
s <- summaryM(age + sex + crea + sod ~ dzclass, test=TRUE, data=d)  
print(s, npct='both', digits=3, middle.bold=TRUE, prmsd=TRUE)

| Descriptive Statistics *(N=1000)*. | | | | | | |
| --- | --- | --- | --- | --- | --- | --- |
|  | N | ARF/MOSF *N=477* | COPD/CHF/Cirrhosis *N=314* | Coma *N=60* | Cancer *N=149* | Test Statistic |
| Age *in years* | 1000 | 47.7 63.7 73.5  (60.4 ±17.4) | 57.4 67.1 76.2  (66.0 ±14.7) | 48.6 64.9 77.8  (62.7 ±18.9) | 53.6 62.6 69.4  (61.4 ±11.6) | *F*3 996=6.64, P<0.0011 |
| sex : male | 1000 | 0.53 253⁄477 | 0.59 186⁄314 | 0.55 33⁄ 60 | 0.60 90⁄149 | χ23=4.21, P=0.2392 |
| Serum creatinine Day 3 | 997 | 0.900 1.400 2.800  (2.258 ±2.134) | 1.000 1.300 1.700  (1.475 ±0.894) | 0.900 1.300 2.025  (2.032 ±2.042) | 0.700 0.900 1.200  (0.968 ±0.312) | *F*3 993=30, P<0.0011 |
| Serum sodium Day 3 | 1000 | 133.00 137.00 142.00  (138.00 ± 6.53) | 135.00 138.00 141.00  (137.50 ± 5.36) | 135.00 138.50 143.25  (139.37 ± 8.74) | 134.00 137.00 140.00  (136.39 ± 4.71) | *F*3 996=2.27, P=0.0791 |
| *a* *b* *c* represent the lower quartile *a*, the median *b*, and the upper quartile *c* for continuous variables. *x ± s* represents X ± 1 SD.   *N* is the number of non-missing values. Tests used: 1Kruskal-Wallis test; 2Pearson test . | | | | | | |

The HTML file is perfect. pandoc (used by Quarto) rendered the .docx table quite well, just not respecting font size changes and the middle.bold argument to put the median in a larger, bold, font.

The math rendered from summaryM output uses HTML Greek characters, subscripts, superscripts, and font size changes. Better output would have been achieved by using the mk() approach except for font size.

Here’s an example from an htmlTable package [vignette](https://cran.r-project.org/web/packages/htmlTable/vignettes/complex_tables.html).

require(htmlTable)  
output <-  
 matrix(paste("Content", LETTERS[1:16]),  
 ncol = 4, byrow = TRUE)  
  
output |>  
 htmlTable(header = paste(c("1st", "2nd", "3rd", "4th"), "header"),  
 rnames = paste(c("1st", "2nd", "3rd", "4th"), "row"),  
 rgroup = c("Group A", "Group B"),  
 n.rgroup = c(2, 2),  
 cgroup = c("Cgroup 1", "Cgroup 2&dagger;"),  
 n.cgroup = c(2, 2),  
 caption = "Basic table with both column spanners (groups) and row groups",  
 tfoot = "&dagger; A table footer commment")

| Basic table with both column spanners (groups) and row groups | | | | | |
| --- | --- | --- | --- | --- | --- |
|  | Cgroup 1 | |  | Cgroup 2† | |
|  | 1st header | 2nd header |  | 3rd header | 4th header |
| Group A | | | | | |
| 1st row | Content A | Content B |  | Content C | Content D |
| 2nd row | Content E | Content F |  | Content G | Content H |
| Group B | | | | | |
| 3rd row | Content I | Content J |  | Content K | Content L |
| 4th row | Content M | Content N |  | Content O | Content P |
| † A table footer commment | | | | | |

Here’s a more advanced one from [here](https://cran.r-project.org/web/packages/htmlTable/vignettes/general.html).

mx <- matrix(ncol = 6, nrow = 8)  
rownames(mx) <- paste(c("1st", "2nd",  
 "3rd",  
 paste0(4:8, "th")),  
 "row")  
colnames(mx) <- paste(c("1st", "2nd",  
 "3rd",   
 paste0(4:6, "th")),  
 "hdr")  
  
for (nr in 1:nrow(mx)) {  
 for (nc in 1:ncol(mx)) {  
 mx[nr, nc] <-  
 paste0(nr, ":", nc)  
 }  
}  
rgroup <- c(paste("Group", LETTERS[1:2]), "")  
attr(rgroup, "add") <- list(`2` = "More")  
mx |>  
 addHtmlTableStyle(align = "rr|r",  
 align.header = "cc|c",  
 spacer.celltype = "double\_cell",  
 col.columns = c(rep("none", 2),  
 rep("#F5FBFF", 4)),  
 col.rgroup = c("none", "#F7F7F7"),  
 css.cell = "padding-left: .5em; padding-right: .2em;",  
 css.header = "font-weight: normal") |>   
 htmlTable(rgroup = rgroup,  
 n.rgroup = c(2,4),  
 tspanner = paste("Spanner", LETTERS[1:2]),  
 n.tspanner = c(1),  
 cgroup = list(c("", "Column spanners"),  
 c("", "Cgroup 1", "Cgroup 2&dagger;")),  
 n.cgroup = list(c(1,5),  
 c(2,2,2)),  
 caption = "A table with column spanners, row groups, and zebra striping",  
 tfoot = "&dagger; A table footer commment",  
 cspan.rgroup = 2)

| A table with column spanners, row groups, and zebra striping | | | | | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  | Column spanners | | | | | | | | |
|  |  | | | |  |  | Cgroup 1 | |  |  | Cgroup 2† | |
|  | 1st hdr |  |  | 2nd hdr |  |  | 3rd hdr | 4th hdr |  |  | 5th hdr | 6th hdr |
| Spanner A | | | | | | | | | | | | |
| Group A | | | |  |  |  |  |  |  |  |  |  |
| 1st row | 1:1 |  |  | 1:2 |  |  | 1:3 | 1:4 |  |  | 1:5 | 1:6 |
| 2nd row | 2:1 |  |  | 2:2 |  |  | 2:3 | 2:4 |  |  | 2:5 | 2:6 |
| Spanner B | | | | | | | | | | | | |
| Group B | | | |  |  |  |  |  |  |  |  | More |
| 3rd row | 3:1 |  |  | 3:2 |  |  | 3:3 | 3:4 |  |  | 3:5 | 3:6 |
| 4th row | 4:1 |  |  | 4:2 |  |  | 4:3 | 4:4 |  |  | 4:5 | 4:6 |
| 5th row | 5:1 |  |  | 5:2 |  |  | 5:3 | 5:4 |  |  | 5:5 | 5:6 |
| 6th row | 6:1 |  |  | 6:2 |  |  | 6:3 | 6:4 |  |  | 6:5 | 6:6 |
| 7th row | 7:1 |  |  | 7:2 |  |  | 7:3 | 7:4 |  |  | 7:5 | 7:6 |
| 8th row | 8:1 |  |  | 8:2 |  |  | 8:3 | 8:4 |  |  | 8:5 | 8:6 |
| † A table footer commment | | | | | | | | | | | | |

The Word result isn’t perfect, but not bad.

## Other R Packages for Table Making

The kableExtra package will not work for docx output. Try gt using an example from [here](https://gt.rstudio.com/articles/gt.html). Sometimes adding prefer-html: true in the yaml header will make simple html tables render properly in Word.

require(gt)  
d <- airquality[1:10,]  
d$Year <- 1973  
  
gt(d) |>  
 tab\_header(  
 title = "New York Air Quality Measurements",  
 subtitle = "Daily measurements in New York City (May 1-10, 1973)"  
 ) |>  
 tab\_spanner(  
 label = "Time",  
 columns = c(Year, Month, Day)  
 ) |>  
 tab\_spanner(  
 label = "Measurement",  
 columns = c(Ozone, Solar.R, Wind, Temp)  
 ) |>  
 cols\_move\_to\_start(  
 columns = c(Year, Month, Day)  
 ) |>  
 cols\_label(  
 Ozone = md("Ozone,<br>\_ppbV\_"),  
 Solar.R = md("Solar R.,<br>$$\\text{cal}/m^2$$"),  
 Wind = md("Wind,<br>mph"),  
 Temp = md("Temp,<br>$$^{\\circ}F$$")  
 )

Table 1: New York Air Quality Measurements

Daily measurements in New York City (May 1-10, 1973)

| Time | | | Measurement | | | |
| --- | --- | --- | --- | --- | --- | --- |
| Year | Month | Day | Ozone,*ppbV* | Solar R.,$$\text{cal}/m^2$$ | Wind,mph | Temp,$$^{\circ}F$$ |
| 1973 | 5 | 1 | 41 | 190 | 7.4 | 67 |
| 1973 | 5 | 2 | 36 | 118 | 8.0 | 72 |
| 1973 | 5 | 3 | 12 | 149 | 12.6 | 74 |
| 1973 | 5 | 4 | 18 | 313 | 11.5 | 62 |
| 1973 | 5 | 5 | NA | NA | 14.3 | 56 |
| 1973 | 5 | 6 | 28 | NA | 14.9 | 66 |
| 1973 | 5 | 7 | 23 | 299 | 8.6 | 65 |
| 1973 | 5 | 8 | 19 | 99 | 13.8 | 59 |
| 1973 | 5 | 9 | 8 | 19 | 20.1 | 61 |
| 1973 | 5 | 10 | NA | 194 | 8.6 | 69 |

Math expressions rendered fine in HTML but not in Word.

## rms Package Model Output Example

Here I fit an ordinal regression model on a continuous response variable using the orm function in rms, the use the orm print method. With options(prTYpe='html') in effect the results are rendered in HTML.

require(rms)  
set.seed(1)  
x <- runif(20)  
y <- runif(20)  
f <- orm(y ~ x)  
f

|  | Model Likelihood Ratio Test | Discrimination Indexes | Rank Discrim. Indexes |
| --- | --- | --- | --- |
| Obs 20 | LR χ2 1.06 | *R*2 0.052 | ρ 0.250 |
| Distinct *Y* 20 | d.f. 1 | *R*21,20 0.003 |  |
| *Y*0.5 0.4466773 | Pr(>χ2) 0.3042 | *R*21,20 0.003 |  |
| max |∂log *L*/∂β| 0.0008 | Score χ2 1.06 | |Pr(*Y* ≥ median)-½| 0.091 |  |
|  | Pr(>χ2) 0.3042 |  |  |

|  | β | S.E. | Wald *Z* | Pr(>|*Z*|) |
| --- | --- | --- | --- | --- |
| x | -1.5028 | 1.4753 | -1.02 | 0.3084 |

The Word result is excellent except for loss of the long overbar over and the coefficient table header being repeated.