**A Lazy Function**

***CillianMacAodh (Cillian McHugh)***

***20 October 2018***

It has been quite a while since I posted, but I haven’t been idle, I completed my PhD since the last post, and I’m due to graduate next Thursday.

**A Lazy Function**

I have already written 2 posts about writing functions, and I will try to diversify my content. That said, I won’t refrain from sharing something that has been helpful to me. The function(s) I describe in this post is an artefact left over from before I started using R Markdown. It is a product of its time but may still be of use to people who haven’t switched to R Markdown yet. It is lazy (and quite imperfect) solution to a tedious task.

**The Problem**

At the time I wrote this function I was using R for my statistics and [Libreoffice](https://www.libreoffice.org/) for writing. I would run a test in R and then write it up in Libreoffice. Each value that needed reporting had to be transferred from my R output to Libreoffice - and for each test there are a number of values that need reporting. Writing up these tests is pretty formulaic. There’s a set structure to the sentence, for example writing up a t-test with a significant result nearly always looks something like this:

An independent samples t-test revealed a significant difference in X between the Y sample, (*M* = [ ], *SD* = [ ]), and the Z sample, (*M* = [ ], SD = [ ]), *t*([df]) = [ ], *p* = [ ].

And the write up of a non-significant result looks something like this:

An independent samples t-test revealed no significant difference in X between the Y sample, (*M* = [ ], *SD* = [ ]), and the Z sample, (*M* = [ ], SD = [ ]), *t*([df]) = [ ], *p* = [ ].

Seven values (the square [ ] brackets) need to be reported for this single test. Whether you copy and paste or type each value, the reporting of such tests can be very tedious, and leave you prone to errors in reporting.

**The Solution**

In order to make reporting values easier (and more accurate) I wrote the t\_paragraph() function (and the related t\_paired\_paragraph() function). This provided an output that I could copy and paste into a Word (Libreoffice) document. This function is part of the desnum package (McHugh, 2017).

**The t\_parapgraph() Function**

The t\_parapgraph() function runs a t-test and generates an output that can be copied and pasted into a word document. The code for the function is as follows:

# Create the function t\_paragraph with arguments x, y, and measure  
# x is the dependent variable  
# y is the independent (grouping) variable  
# measure is the name of dependent variable inputted as string  
  
t\_paragraph <- function (x, y, measure){  
   
 # Run a t-test and store it as an object t  
   
 t <- t.test(x ~ y)  
   
   
 # If your grouping variable has labelled levels, the next line will store them for reporting at a later stage  
   
 labels <- levels(y)  
   
 # Create an object for each value to be reported  
   
 tsl <- as.vector(t$statistic)  
 ts <- round(tsl, digits = 3)  
 tpl <- as.vector(t$p.value)  
 tp <- round(tpl, digits = 3)  
 d\_fl <- as.vector(t$parameter)  
 d\_f <- round(d\_fl, digits = 2)  
 ml <- as.vector(tapply(x, y, mean))  
 m <- round(ml, digits = 2)  
 sdl <- as.vector(tapply(x, y, sd))  
 sd <- round(sdl, digits = 2)  
   
 # Use print(paste0()) to combine the objects above and create two potential outputs  
 # The output that is generated will depend on the result of the test  
   
   
 # wording if significant difference is observed  
   
 if (tp < 0.05)   
 print(paste0("An independent samples t-test revealed a significant difference in ",   
 measure, " between the ", labels[1], " sample, (M = ",   
 m[1], ", SD = ", sd[1], "), and the ", labels[2],   
 " sample, (M =", m[2], ", SD =", sd[2], "), t(",   
 d\_f, ") = ", ts, ", p = ", tp, "."), quote = FALSE,   
 digits = 2)  
   
 # wording if no significant difference is observed   
   
 if (tp > 0.05)   
 print(paste0("An independent samples t-test revealed no difference in ",   
 measure, " between the ", labels[1], " sample, (M = ",   
 m[1], ", SD = ", sd[1], "), and the ", labels[2],   
 " sample, (M = ", m[2], ", SD =", sd[2], "), t(",   
 d\_f, ") = ", ts, ", p = ", tp, "."), quote = FALSE,   
 digits = 2)  
}

When using t\_paragraph(), x is your DV, y is your grouping variable while measure is a string value that the name of the dependent variable. To illustrate the function I’ll use the mtcars dataset.

**Applications of the t\_parapgraph() Function**

The mtcars dataset is comes with R. For information on it simply type help(mtcars). The variables of interest here are am (transmission; 0 = automatic, 1 = manual), mpg (miles per gallon), qsec (1/4 mile time). The two questions I’m going to look at are:

1. Is there a difference in miles per gallon depending on transmission?
2. Is there a difference in 1/4 mile time depending on transmission?

Before running the test it is a good idea to look at the data[2](http://feedproxy.google.com/~r/Cillianmacaodh/~3/wxbIdWh9RV8/a-lazy-function.html#fn2). Because we’re going to look at differences between groups we want to run descriptives for each group separately. To do this I’m going to combine the the descriptives() function which I previously covered (also part of the desnum package) and the tapply() function.

The tapply() function allows you to run a function on subsets of a dataset using a grouping variable (or index). The arguments are as follows tapply(vector, index, function). vector is the variable you want to pass through function; and index is the grouping variable. The examples below will make this clearer.

We want to run descriptives on mtcars$mpg and on mtcars$qsec and for each we want to group by transmission (mtcars$am). This can be done using tapply() and descriptives() together as follows:

tapply(mtcars$mpg, mtcars$am, descriptives)

## $`0`  
## mean sd min max len  
## 1 17.14737 3.833966 10.4 24.4 19  
##   
## $`1`  
## mean sd min max len  
## 1 24.39231 6.166504 15 33.9 13

Recall that 0 = automatic, and 1 = manual. Replace mpg with qsec and run again:

tapply(mtcars$qsec, mtcars$am, descriptives)

## $`0`  
## mean sd min max len  
## 1 18.18316 1.751308 15.41 22.9 19  
##   
## $`1`  
## mean sd min max len  
## 1 17.36 1.792359 14.5 19.9 13

**Running t\_paragraph()**

Now that we know the values for automatic vs manual cars we can run our t-tests using t\_paragraph(). Our first question:

Is there a difference in miles per gallon depeding on transmission?

t\_paragraph(mtcars$mpg, mtcars$am, "miles per gallon")

## [1] An independent samples t-test revealed a significant difference in miles per gallon between the sample, (M = 17.15, SD = 3.83), and the sample, (M =24.39, SD =6.17), t(18.33) = -3.767, p = 0.001.

There is a difference, and the output above can be copied and pasted into a word document with minimal changes required.

Our second question was:

Is there a difference in 1/4 mile time depending on transmission?

t\_paragraph(mtcars$qsec, mtcars$am, "quarter-mile time")

## [1] An independent samples t-test revealed no difference in quarter-mile time between the sample, (M = 18.18, SD = 1.75), and the sample, (M = 17.36, SD =1.79), t(25.53) = 1.288, p = 0.209.

This time there was no significant difference, and again the output can be copied and pasted into word with minimal changes.

**Limitations**

The function described was written a long time ago, and could be updated. However I no longer copy and paste into word (having switched to R markdown instead). The reporting of the p value is not always to APA standards. If p is < .001 this is what should be reported. The code for t\_paragraph() could be updated to include the p\_report function, which would address this. Another limitation is that the formatting of the text isn’t perfect, the letters (N,M,SD,t,p) should all be italicised, but having to manually fix this formatting is still easier than manually transferring individual values.

**Conclusion**

Despite the limitations the functions t\_paragraph() and t\_paired\_paragraph()have made my life easier. I still use them occasionally. I hope they can be of use to anyone who is using R but has not switched to R Markdown yet.

**References**

McHugh, C. (2017). *Desnum: Creates some useful functions*.