2016 ■ Vol. 12 ■ no. 2

schoRsch: An R package for analyzing and reporting factorial experiments

Roland Pfister[a, ✉] and Markus Janczyk[b]

[a]Julius Maximilians University of Würzburg
[b]Eberhard Karls University of Tübingen

Abstract ■ The schoRsch package aims at improving the usability of the open-source software R for researchers in psychology and related disciplines. It provides easy-to-use functions to format the results of typical tests used in psychology and related fields according to the current style guidelines of the American Psychological Association (APA). These and several other convenience functions, allow for efficient data analysis and aim at expediting the workflow when reporting test results in scientific publications.

Keywords ■ R software, usability, APA style, effect sizes. **Tools** ■ R.

Acting Editor ■ Denis Cousineau (Université d'Ottawa)

Reviewers
■ Vincent LeBlanc (Université d'Ottawa)

✉ roland.pfister@psychologie.uni-wuerzburg.de

iD *RP*: 0000-0002-4429-1052

doi 10.20982/tqmp.12.2.p147

Introduction

The R software for statistical computing (R Core Team, 2014) has gained a large and active user base among psychologists. To customize the base software according to the needs of behavioral data analysis and cognitive modeling, these users have provided a wide array of add-on packages (for summaries and reviews, see e.g., Fox, 2014; Mair, 2015; Wickelmaier, Strobl, & Zeileis, 2012).

Such R packages are written with different applications and target audiences in mind. Some aim at presenting relatively advanced methods to sophisticated specialists. Among others, these packages include exhaustive collections of tools for personality psychology and scale construction (psy: Fallisard, 2012; psych: Revelle, 2015), as well as tools that follow recent trends toward fitting mixed-effects models to behavioral data (lme4: Bates, Maechler, Bolker, & Walker, 2015; nlme: Pinheiro, Bates, DebRoy, Sarkar, & R Core Team, 2015). Using these packages typically requires a thorough statistical background and also some knowledge of the R language.

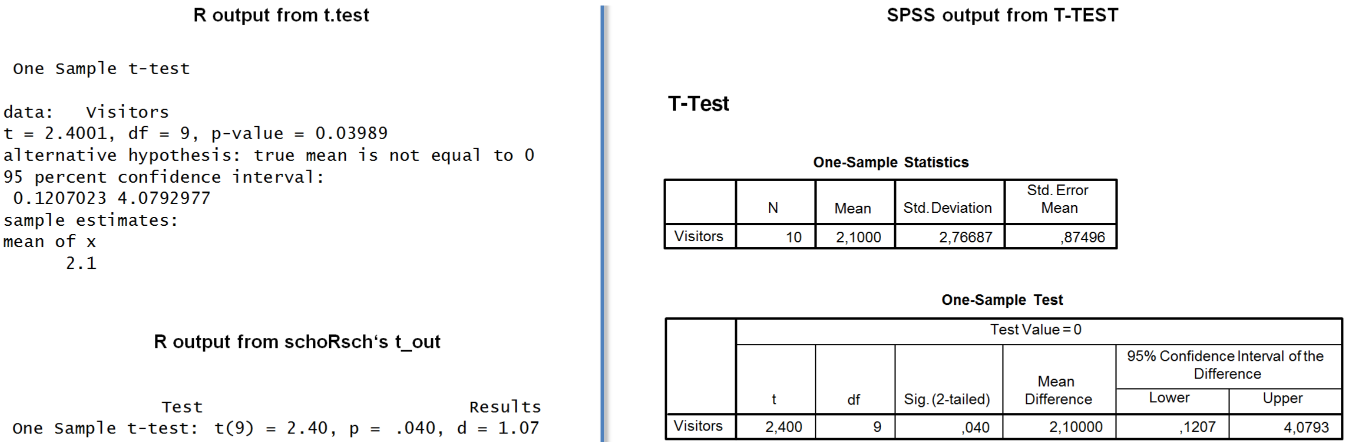
Other packages aim at increasing the usability of the software. These packages often aim at less experienced users, and provide easy-to-use functions that expedite the process of data analysis (for example, userfriendlyscience: Peters, 2015). They often come with wrapper functions that intend to provide an intuitive interface to native functions of the R language which are expected to be difficult to master otherwise. Prominent examples for this type of functionality are packages that facilitate the analysis of factorial experiments via analysis of variance (ANOVA), without having to specify the corresponding models manually (afex: Singmann et al., 2015; ez: Lawrence, 2013). Next to facilitating the access to functions – i.e., facilitating the input into R – it is also possible to optimize the output provided by R. This latter type of functionality is more rarely seen in published packages, though exceptions certainly exist (e.g., in terms of the functions ezDesign or ezPlot of the ez package).

Yet, the need for optimizing the output provided by R becomes apparent when comparing R to commercial software commonly used by psychologists such as SPSS/PASW (Fig. 1). Whereas R tends to come with a seemingly unstructured assortment of information, the structured tables provided by SPSS/PASW and other programs might look more appealing to many users. The package schoRsch (Pfister & Janczyk, 2015) presented here mainly aims at doing precisely that: providing a more accessible and intuitive output for common statistical tests in order to enhance R's usability for psychological scientists.

schoRsch

The package schoRsch – pronounced "shorsh" – lends its name to Georg "Schorsch" Schuessler, ingenious former

Figure 1 ■ Output provided by R's `t.test` function (top left-hand side) and SPSS's `T-TEST` function (right-hand side). Screenshots were taken from RStudio (www.rstudio.org) running R 3.1.1 and from SPSS 22 (IBM Corp.), respectively. The bottom left-hand screenshot shows the formatting provided by `schoRsch` and its `t_out` function, yielding a structured format following APA style that can be directly pasted to any type of document.



R output from t.test

```
 One Sample t-test

data:   Visitors
t = 2.4001, df = 9, p-value = 0.03989
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 0.1207023 4.0792977
sample estimates:
mean of x
      2.1
```

R output from schoRsch's t_out

```
               Test                          Results
 One Sample t-test:  t(9) = 2.40, p = .040, d = 1.07
```

SPSS output from T-TEST

T-Test

One-Sample Statistics

	N	Mean	Std. Deviation	Std. Error Mean
Visitors	10	2,1000	2,76687	,87496

One-Sample Test

	Test Value = 0					
					95% Confidence Interval of the Difference	
	t	df	Sig. (2-tailed)	Mean Difference	Lower	Upper
Visitors	2,400	9	,040	2,10000	,1207	4,0793

technician at the Department of Psychology III, University of Wuerzburg. As noted above, a main aim of the package was to increase the usability of R's output of different statistical tests. To this end, we set ourselves two goals: First, the output of commonly used tests such as `t.test` or `cor.test` should allow a more structured, at-a-glance assessment of the results. Second, the output should allow for direct pasting into a manuscript that uses APA style (American Psychological Association, 2009) to report statistical tests.

To meet the first objective, we decided to concentrate on inferential statistics in terms of (1) the test statistic and the associated degrees of freedom, (2) the p-value, and (3) an appropriate measure of effect size. Following APA style, these statistics should be presented with two decimal places in case of the test statistic and the effect size, and with three decimal places (while omitting the trailing 0) for p-values. The text-based command line interface of R further lends itself ideally to copying these numbers into a possible manuscript.[1]

In the following, we first describe how to use the function `t_out` to format the output of R's `t.test` as an example on how to use the format functions available within the `schoRsch` package.[2] We then take a closer look at the detailed mechanics of this function for users who are interested in embedding this functionality for other tests.

Using the output functions

Once `schoRsch` is installed and loaded via `library(schoRsch)`, it provides the functions `t_out` to format the results of a call to `t.test`, `chi_out` for formatting the results of `chisq.test`, `cor_out` for formatting the results of `cor.test`, and `anova_out` that allows formatting the output from `ezANOVA` (Lawrence, 2013).

As a hands-on example of how to use these functions, we will describe a call of `t_out` with a fictitious data set that is evaluated with a one-sample t-test (the function also supports t-tests for paired samples and for independent samples). Suppose, for instance, you are monitoring 10 different homepages on the internet and you are tracking the number of users on your pages. For each of the sites, you have a number representing the current trend in visitor numbers, with positive numbers representing an increase in visitors (in arbitrary units). To figure out whether your sites generally show an upward trend in visitor numbers, you perform a t-test against a reference value of 0. In R, such a test could look like this:

```
> Visitors <- c(-1,4,2,3,0,6,-1,-1,3,6)
> t.test(Visitors)
```

[1] When mentioning manuscripts, we have a user of MS Word or Libre Office Writer in mind. Users of LaTeX (possibly using `Sweave`; Leisch, 2002) might still find the formatted output useful.

[2] The `schoRsch` package also provides a number of additional convenience functions, e.g., for outlier correction based on z-scores or absolute cut-offs. These functions are documented in the package's reference manual. Contributions of additional functions by any user of the R language are welcome.

```
        One Sample t-test

data:  Visitors
t = 2.4001, df = 9, p-value = 0.03989
alternative hypothesis: true mean is not
    equal to 0
95 percent confidence interval:
 0.1207023 4.0792977
sample estimates:
mean of x
      2.1
```

The original output obviously comes with a test statistic, the associated degrees of freedom, and the corresponding p-value. However, a measure of effect size is missing to properly report this test according to APA style.[3] To get an appropriate measure for this test, such as Cohen's d based on the sample mean m and standard deviation sd, and using a correction factor of $\sqrt{2}$ for the one-sample case

$$d = \sqrt{2} \times \frac{m}{sd}, \tag{1}$$

one either has to do the necessary computations manually or resort to packages that come with appropriate functionality (e.g., MBESS: Kelley & Lai, 2012; or compute.es: Del Re, 2014, 2015). The function t_out uses the information provided in the fields $statistic and $parameter of the result of each call to t.test to compute the effect size according to the formula

$$d = \sqrt{2} \times \frac{t}{\sqrt{n}}, \tag{2}$$

where

$$t = \frac{m}{se}, \tag{3}$$

and therefore provides an integrated summary of the results that can be easily copied to a word processor:

```
> t_out(t.test(Visitors))

             Test
1 One Sample t-test:
                       Results
t(9) = 2.40, p = .040, d = 1.07
```

Additional options allow for customizing this format. For instance, adding the option d.corr = FALSE to the call to t_out will return an uncorrected effect size according to the formula

$$d = \frac{t}{\sqrt{n}}. \tag{4}$$

Other options, especially relating to t-tests for independent samples, are documented in the schoRsch package and are thus available via calling ?t_out in R. Documentation for the output functions chi_out, cor_out, and anova_out can be accessed in the same way.

Mechanics of t_out

Like most test functions in R, t.test returns a named list that assembles all relevant statistics produced by the call to t.test. The components of the list can therefore be accessed by using the dollar operator ($name). Assuming we had saved the output of t.test to the list results, we can view the available components via the names function:

```
> results <- t.test(Visitors)
> names(results)

[1] "statistic"   "parameter"   "p.value"
[4] "conf.int"    "estimate"    "null.value"
[7] "alternative"  "method"      "data.name"
```

Of these components, t_out makes use of results$statistic, containing the t-value, results$parameter, containing the degrees of freedom, and results$p.value, containing the associated p-value. The t-value and the degrees of freedom can be used to compute an estimate of the effect size d according to the formula given in the previous section:

```
> d <- results$statistic / sqrt(results$
+ parameter +1) * sqrt(2)
> d

       t
1.073361
```

After this computation, all information is available to report the test according to APA standards (note that R further adds the headline t to the above output which is a remnant of the component results$statistic and will be replaced later). To comply with APA formatting, the results can be formatted using the round function to crop the number to the appropriate number of decimal places, and by using the format function to show trailing zeros. The format functions further converts the numbers to strings, i.e., character vectors, which will become handy later on. For example:

```
> format(round(results$statistic,2),
+ nsmall=2)

     t
"2.40"
```

Such formatting is applied to the t-statistic, the p-value,

[3] Note that this lack of an effect size is also the case for current versions of SPSS

and the effect size, and the corresponding data are assembled in a data frame:

```
> formatted_results <- data.frame(
+   t=format(round(results$statistic,2),
+     nsmall=2),
+   df = results $parameter,
+   p = format(round(results$p.value,3),
+     nsmall=3),
+   d = format(round(d,2),nsmall=2))
> formatted_results

     t df      p    d
t 2.40  9 0.040 1.07
```

The resulting data frame has four variables: t, df, p, and d, and it has one case that is currently labelled t. The variables already contain the appropriate formatting in terms of decimal places, but are still missing mathematical markup (parentheses, equals signs, etc.). The p-value further is still denoted with a leading zero that needs to be replaced. We begin with the p-value and concatenate a string by combining “, p = ” with the current content of formatted_results$p:

```
> pcorr <- paste(", p = ",
+   formatted_results\$p, sep="")
> pcorr

[1] ", p = 0.040"
```

The trailing zero is then removed via the gsub function that allows performing string replacement for a character vector. This function is also used to replace mathematical errors due to rounding ($p = 1.000$, and $p = 0.000$, respectively).

```
> pcorr <- gsub("p = 1.000","p > .999",
+   pcorr, fixed=TRUE)
> pcorr <- gsub("p = 0.000","p < .001",
+   pcorr, fixed=TRUE)
> pcorr <- gsub("p = 0","p = ", pcorr,
+   fixed=TRUE)
```

The final character vector can then be constructed in the following way:

```
> formatted_final <- paste("t(",
+   formatted_results$df,") = " ,
+   formatted_results$t, pcorr,
+   ", d = ", formatted_results$d,sep=""
+ )
> formatted_final

[1] "t(9) = 2.40, p = .040, d = 1.07"
```

The information is now ready to be copied to a manuscript. To further increase readability, schoRsch adds this formatted character vector to a data frame that also contains a reference to the test in question:

```
> outtext <- data.frame(
+   Test=paste(results$method,":",
+   sep=""),
+   Results=formatted_final)
> outtext

             Test
1 One Sample t-test:
                        Results
t(9) = 2.40, p = .040, d = 1.07
```

The remaining output functions, chi_out, cor_out, and anova_out operate in a similar way to distill relevant output from named lists that are produced by the corresponding test functions. All return data frames as described above, stating the test in question, followed by a manuscript-ready line for copy and pasting.

Summary

The schoRsch package improves the usability of R’s most common statistical tests for researchers in psychology and related disciplines. It provides easy-to-use functions that format the results of these tests according to APA style, thus allowing an efficient workflow when intending to report these tests in publications.

Authors’ note

Work of MJ is supported by the Institutional Strategy of the University of Tübingen (Deutsche Forschungsgemeinschaft [German Research Foundation], ZUK 63).

References

American Psychological Association. (2009). *Publication manual of the American Psychological Association*. Washington, DC: APA.

Bates, D., Maechler, M., Bolker, B., & Walker, S. (2015). lme4: Linear mixed-effects models using Eigen and S4 (Version 1.1-8). Retrieved from http://cran.r-project.org/web/packages/lme4

Del Re, A. C. (2014). compute.es: Compute effect sizes (Version 0.2-4). Retrieved from http://cran.r-project.org/web/packages/compute.es

Del Re, A. C. (2015). A practical tutorial on conducting meta-analysis in R. *The Quantitative Methods for Psychology*, *11*(1), 37–50.

Fallisard, B. (2012). psy: Various procedures used in psychometry (Version 1.1). Retrieved from http://cran.r-project.org/web/packages/psy

Fox, J. (2014). CRAN Task View: Statistics for the social sciences. Retrieved from http://cran.r-project.org/web/views/SocialSciences.html

Kelley, K. & Lai, K. (2012). MBESS (Version 3.3). Retrieved from http://cran.r-project.org/web/packages/MBESS

Lawrence, M. A. (2013). ez: Easy analysis and visualization of factorial experiments. Retrieved from http://cran.r-project.org/web/packages/ez

Leisch, F. (2002). sweave: Dynamic generation of statistical reports using literate data analysis. In W. Härdle & B. Rönz (Eds.), *Compstat 2002 - Proceedings in computational statistics* (pp. 575–580). Heidelberg: Physica.

Mair, P. (2015). CRAN Task View: Psychometric models and methods. Retrieved from http://cran.r-project.org/web/views/Psychometrics.html

Peters, G.-J. (2015). userfriendlyscience: Quantitative analysis made accessible (Version 0.3.0). Retrieved from http://cran.r-project.org/web/packages/userfriendlyscience

Pfister, R. & Janczyk, M. (2015). schoRsch: Tools for analyzing factorial experiments. Retrieved from http://CRAN.R-project.org/package=schoRsch

Pinheiro, J., Bates, D., DebRoy, S., Sarkar, D., & R Core Team. (2015). nlme: Linear and nonlinear mixed effects models. Retrieved from http://cran.r-project.org/web/packages/nlme

R Core Team. (2014). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria. Retrieved from http://www.R-project.org/

Revelle, W. (2015). psych: Procedures for personality and psychological research (Version 1.5.4). Retrieved from http://cran.r-project.org/web/packages/psych

Singmann, H., Bolker, B., Westfall, J., Højsgaard, S., Fox, J., Lawrence, M. J., & Mertens, U. (2015). afex: Analysis of factorial experiments. Retrieved from http://cran.r-project.org/web/packages/afex

Wickelmaier, F., Strobl, C., & Zeileis, A. (2012). psychoco: Psychometric computing in R. *Journal of Statistical Software*, *48*(1), 1–5. doi:10.18637/jss.v048.i01

Citation

Pfister, R. & Janczyk, M. (2016). schoRsch: An R package for analyzing and reporting factorial experiments. *The Quantitative Methods for Psychology*, *12*(2), 147–151. doi:10.20982/tqmp.12.2.p147

Received: 12/02/2016 ∼ Accepted: 21/05/2016