

rempsysc: Convenience functions for psychology

Rémi Thériault ¹

¹ Department of Psychology, Université du Québec à Montréal, Québec, Canada

DOI: [10.xxxxxx/draft](https://doi.org/10.xxxxxx/draft)

Software

- [Review](#) 
- [Repository](#) 
- [Archive](#) 

Editor: [Open Journals](#) 

Reviewers:

- [@openjournals](#)

Submitted: 01 January 1970

Published: unpublished

License

Authors of papers retain copyright
and release the work under a
Creative Commons Attribution 4.0
International License ([CC BY 4.0](#))

Summary

`{rempsysc}` is an R package of convenience functions that make the analysis-to-publication workflow faster, easier, and less error-prone. It enables nice APA tables exportable to Word (via `{flectable}`) and easily customizable APA-compliant plots (via `{ggplot2}`). It makes it easy to run statistical tests, check assumptions, and automate various tasks common in psychology research.

Statement of need

There are many reasons to use R ([R Core Team, 2022](#)) for analyzing and reporting data from research studies, such as being compatible with the ideals of open science ([Quintana, 2020](#)). However, R has a major downside for novices: its steep learning curve due to its programmatic interface, in contrast to perhaps more user-friendly point-and-click software. Of course, this flexibility is also a strength, as the R community can and does come together to produce packages that make using R increasingly easier and more user-friendly (e.g., the *easystats* ecosystem [Lüdtke et al., 2019/2023](#)). The `{rempsysc}` package (Really Easy Methods for Psychology) contributes to this momentum by providing convenience functions that remove as much friction as possible between your script and your manuscript (in particular, if you are using Microsoft Word).

There are mainly three things that go into a manuscript: text, tables, and figures. `{rempsysc}` does not generate publication-ready text summarizing analyses; for this, see the `{report}` package ([Makowski et al., 2021/2023](#)). Instead, `{rempsysc}` focuses on the production of publication-ready tables and figures. Below, I go over a few quick examples of those.

Examples Features

Publication-Ready Tables

Many researchers using R still copy-paste the values from the R console to their manuscript, or retype them manually. Yet, this approach increases the risks of copy-paste and retyping errors so common in psychology. This problem is not trivial given that according to some estimates, up to 50% of articles in psychology have at least one statistical error ([Nuijten et al., 2016](#)). Ideally, one should be able to format the table directly in R, and to export it to Word directly.

Formatting your table properly in R is already a time-consuming task, but fortunately several packages take care of this step (e.g., the `{broom}` or `{report}` packages, [Makowski et al., 2021/2023](#); [Robinson et al., 2022](#), and there are several others). Exporting these formatted tables to Microsoft Word remains a challenge however. Some packages do export to Word (e.g., [Stanley & Spence, 2018](#)), but their formatting is often rigid especially when using analyzes or table formats that are not supported by default.

38 {rempsysc} solves this problem by allowing maximum flexibility: you manually create the data
39 frame exactly the way you want, and then only use the magical function, `nice_table()`, on
40 the resulting data frame. `nice_table()` works on any data frame, even non-statistical ones
41 like `mtcars`.

42 One of its main benefit however is the automatic formatting of statistical symbols and its
43 integration with other packages. We can for example create a {broom} table and then apply
44 `nice_table()` on it. It suits particularly well the pipe workflow.

```
45 library(rempsysc)
46
47 lm(mpg ~ cyl + wt * hp, mtcars) |>
48   broom::tidy(conf.int = TRUE) |>
49   nice_table(broom = "lm")
```

50 We can do the same with a {report} table.

```
51 stats.table <- lm(mpg ~ cyl + wt * hp, mtcars) |>
52   report::report() |>
53   as.data.frame()
54
55 nice_table(stats.table)
```

56 The {report} package provides quite comprehensive tables, so one may request an abbreviated
57 table with the 'short' argument. For convenience, it is also possible to highlight significant
58 results for better visual discrimination, using the 'highlight' argument.[1] Once satisfied
59 with the table, we can add a title and note.

```
60 my_table <- nice_table(
61   stats.table, short = TRUE, highlight = 0.001,
62   title = c("Table 1", "A Pretty Regression Model"),
63   note = c("The data was extracted from the 1974 Motor Trend US magazine.",
64            "Greyed rows represent statistically significant differences, p < .001."))
65 my_table
```

66 One can then easily save the resulting table to Word with `flextable::save_as_docx()`,
67 specifying the object name and desired path.

```
68 flextable::save_as_docx(my_table, path = "nice_tablehere.docx")
```

69 Additionally, tables created with `nice_table()` are {flextable} objects (Gohel & Skintzos,
70 2022), and can be modified as such.[2]

71 **Formattting Results of Analyses**

72 {rempsysc} also provides its own set of functions to prepare statistical tables before they can be
73 fed to `nice_table()` and saved to Word.

74 **t tests**

```
75 nice_t_test(data = mtcars,
76             response = c("mpg", "disp", "drat"),
77             group = "am",
78             warning = FALSE) |>
79   nice_table()
```

80 **Contrasts**

```
81 nice_contrasts(data = mtcars,
82                response = c("mpg", "disp"),
```

```

83         group = "cyl",
84         covariates = "hp") |>
85     nice_table(highlight = .001)

```

86 Regressions

```

87 model1 <- lm(mpg ~ cyl + wt * hp, mtcars)
88 model2 <- lm(qsec ~ disp + drat * carb, mtcars)
89
90 nice_lm(list(model1, model2)) |>
91     nice_table(highlight = TRUE)

```

92 Simple Slopes

```

93 model1 <- lm(mpg ~ gear * wt, mtcars)
94 model2 <- lm(disp ~ gear * wt, mtcars)
95 my.models <- list(model1, model2)
96
97 nice_lm_slopes(my.models, predictor = "gear", moderator = "wt") |>
98     nice_table()

```

99 Correlation Matrices

100 It is also possible to export a colour-coded correlation matrix to Microsoft Excel. The
 101 `cormatrix_excel()` function has several benefits over conventional approaches. The base R
 102 `cor()` function for example does not use rounded values and the console is impractical for
 103 large matrices. One may manually round values and export it to a .csv file, which is an
 104 improvement but still unsatisfying.

105 The `{apaTables}` package (Stanley & Spence, 2018) allows exporting the correlation matrix
 106 to Word in an APA format, and in many cases this is very satisfying for APA requirements.
 107 However, the Word format is not suitable for large matrices, as it will often spread beyond the
 108 document's margin limits.

109 Another approach is to export to an image, like the `{correlation}` package does (Makowski
 110 et al., 2020).[3] For very small matrices, this works extremely well, and the colour is an
 111 immense help to quickly identify which correlations are strong or weak, positive or negative,
 112 and significant or non-significant. Again, however, this does not work so well for large matrices
 113 because labels might overlap or navigating the large figure becomes difficult.

114 When the goal is more exploratory, rather than reporting, and we have large matrices, it can
 115 be more useful to export it to Excel. In `{rempsyc}`, we combine the idea of using a coloured
 116 correlation matrix from the `{correlation}` package with the idea of exporting to Excel using
 117 `{openxlsx}` (Barbone & Garbuszus, 2023).

118 We also provide some usability improvements, like freezing the first row and column so as to
 119 be able to easily see which variables correlates with which other variable, regardless of how far
 120 or deep those variables are located within the matrix.

121 The colour represents the strength of the correlation, whereas the stars represent how significant
 122 the p value is.[4] The exact p values are provided in a second tab for reference purposes, so all
 123 information is readily available in a convenient format.

```

124 cormatrix_excel(data = infert,
125                 filename = "cormatrix1",
126                 select = c("age", "parity", "induced", "case", "spontaneous",
127                             "stratum", "pooled.stratum"))

```

	A	B	C	D	E	F	G	H	I
1	Parameter	age	parity	induced	case	spontaneous	stratum	pooled.stratum	
2	age	1.0	.08	-.10	.0	-.08	-.21 ***	-.17 *	
3	parity	.08	1.0	.45 ***	.01	.31 ***	-.31 ***	.12	
4	induced	-.10	.45 ***	1.0	.02	-.27 ***	-.10	.16 *	
5	case	.0	.01	.02	1.0	.36 ***	.0	.0	
6	spontaneous	-.08	.31 ***	-.27 ***	.36 ***	1.0	.06	.21 ***	
7	stratum	-.21 ***	-.31 ***	-.10	.0	.06	1.0	.75 ***	
8	pooled.stratum	-.17 *	.12	.16 *	.0	.21 ***	.75 ***	1.0	
9									
128		r_values	p_values						
	A	B	C	D	E	F	G	H	I
1	Parameter	age	parity	induced	case	spontaneous	stratum	pooled.stratum	
2	age	.0	.194	.113	.956	.186	.001	.006	
3	parity	.194	.0	.0	.889	.0	.0	.059	
4	induced	.113	.0	.0	.789	.0	.113	.010	
5	case	.956	.889	.789	.0	.0	.952	.939	
6	spontaneous	.186	.0	.0	.0	.0	.341	.001	
7	stratum	.001	.0	.113	.952	.341	.0	.0	
8	pooled.stratum	.006	.059	.010	.939	.001	.0	.0	
9									
129		r_values	p_values						

Publication-Ready Figures

Preparing figures according to APA style, having them look good, and being able to save them in high-resolution with the proper ratios is often challenging. Working with `{ggplot2}` (Wickham, 2016) provides tremendous flexibility, but an unintended consequence is that doing even trivial operations can at times be daunting.

This is why `{rempsyc}` setups a few default plot types, ready to be saved to your preferred format (.pdf, .tiff, or .png).

Violin Plots

```
nice_violin(data = ToothGrowth,
            group = "dose",
            response = "len",
            xlabels = c("Low", "Medium", "High"),
            comp1 = 1,
            comp2 = 3,
            has.d = TRUE,
            d.y = 30)
```

For an example of such use in publication, see Thériault et al. (2021).

One can easily save the resulting figure with `ggplot2::ggsave()`, specifying the desired file name, extension, and resolution.

```
ggplot2::ggsave('nice_violinplotthere.pdf', width = 7, height = 7,
               unit = 'in', dpi = 300)
```

Recommended dimensions for saving `{rempsyc}` figures is 7 inches wide and 7 inches high at 300 dpi, which makes sure that the resolution is high enough even if saving to non-vector

153 graphics formats like .png. That said, scalable vector graphics formats like .pdf or .eps are
154 still recommended for high-resolution submissions to scientific journals.

155 Scatter Plots

156 Figures are {ggplot2} objects ([Wickham, 2016](#)), and can be modified as such.

```
157 nice_scatter(data = mtcars,  
158             predictor = "wt",  
159             response = "mpg",  
160             group = "cyl",  
161             has.confband = TRUE)  
  
162 nice_scatter(data = mtcars,  
163             predictor = "wt",  
164             response = "mpg",  
165             has.confband = TRUE,  
166             has.r = TRUE,  
167             has.p = TRUE) +  
168   ggplot2::geom_hline(yintercept = mean(mtcars$mpg), colour = "black",  
169                     linewidth = 1.4, linetype = "dashed") +  
170   ggplot2::annotate("text", x = 3.5, y = 22, size = 7,  
171                     label = paste("Mean mpg =", round(mean(mtcars$mpg), 2)))
```

172 For an example of such use in publication, see Krol et al. ([2020](#)).

173 Overlapping Circles

174 For psychologists using the Inclusion of Other in the the Self Scale ([Aron et al., 1992](#)), it can
175 be useful to interpolate the original discrete scores (1 to 7) into a group average representation
176 of the conceptual self-other overlap.

```
177 overlap_circle(3.5)
```

178 For an example of such use in publication, see Thériault et al. ([2021](#)).

179 Testing assumptions

180 When comes time to test assumptions of a linear model, the best option is the `check_model()`
181 function from *easystats*' {performance} package, which allows direct visual evaluation of
182 assumptions ([Lüdtke, Ben-Shachar, et al., 2021](#)). Indeed, visual assessment of diagnostic
183 plots is recommended over statistical tests since they are overpowered in large samples and
184 underpowered in small samples ([Kozak & Piepho, 2018](#)).

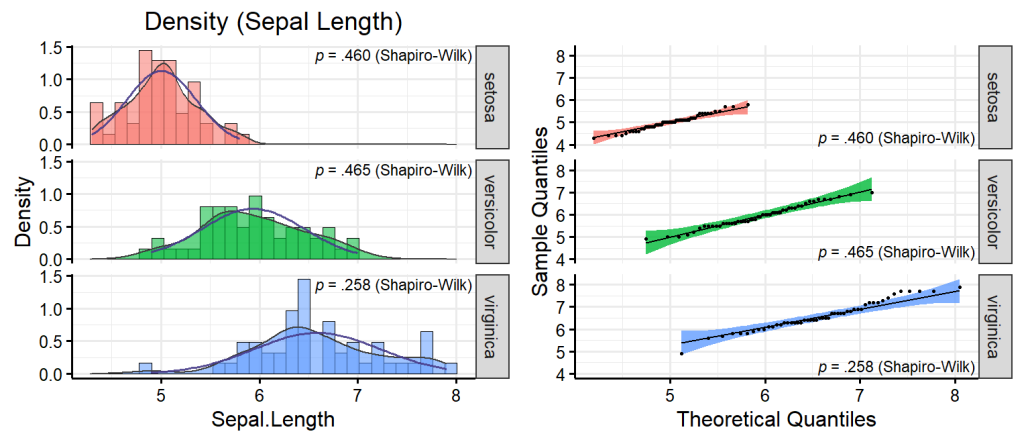
185 That said, if for whatever reason one wants to check objective assumption tests for a linear
186 model, *rempsysc* makes this easy with the `nice_assumptions()` function, which provide *p*
187 values for normality (Shapiro–Wilk), homoscedasticity (Breusch–Pagan) and autocorrelation of
188 residuals (Durbin–Watson) in one call.

189 Categorical Predictors

190 `nice_normality()` makes it easy to visually check normality in the case of categorical predictors
191 (i.e., when using groups), through a combination of quantile-quantile plots, density plots, and
192 histograms.

```
193 nice_normality(data = iris,  
194               variable = "Sepal.Length",  
195               group = "Species",  
196               shapiro = TRUE,
```

```
197 histogram = TRUE,
198 title = "Density (Sepal Length)")
```



199

200 Similarly for univariate outliers using the median absolute deviation (MAD, [Leys et al., 2013](#)).

```
201 plot_outliers(airquality,
202               group = "Month",
203               response = "Ozone")
```

204 Univariate outliers based on the MAD can also be simply requested with `find_mad()`.^[5]

```
205 find_mad(airquality, names(airquality), criteria = 3)
206
207 ## 8 outlier(s) based on 3 median absolute deviations for variable(s):
208 ##  Ozone, Solar.R, Wind, Temp, Month, Day
209 ##
210 ## Outliers per variable:
211 ##
212 ## $Ozone
213 ##   Row Ozone_mad
214 ## 1  30  3.218284
215 ## 2  62  3.989131
216 ## 3  99  3.488081
217 ## 4 101  3.025573
218 ## 5 117  5.261028
219 ## 6 121  3.333911
220 ##
221 ## $Wind
222 ##   Row Wind_mad
223 ## 1   9  3.049871
224 ## 2  48  3.225825
```

225 Homoscedasticity can also be checked numerically with `nice_var()` or visually with `nice_varplot()`.

```
227 nice_var(data = iris,
228          variable = names(iris[1:4]),
229          group = "Species") |>
230 nice_table()
231
232 nice_varplot(data = iris,
233              variable = "Sepal.Length",
234              group = "Species")
```

Utility functions

Finally, with the idea of making the analysis workflow easier in mind, `{rempsyc}` also provides a few other utility functions. `nice_na()` allows reporting item-level missing values per scale, as well as participant's maximum number of missing items by scale, as per recommendations (Parent, 2013).

`extract_duplicates()` creates a data frame of only observations with a duplicated ID or participant number, so they can be investigated more thoroughly. `best_duplicate()` allows to follow-up on this investigation and only keep the "best" duplicate, meaning those with the fewer number of missing values, and in case of ties, the first one.

`nice_reverse()` permits the automatic reverse-coding of scores so common for psychology questionnaires, provided the minimum and maximum score values are known.

There are other functions that the reader can explore at their leisure on the package official website. However, hopefully, this overview has given the reader a gentle introduction to this package.

Licensing and Availability

The `{rempsyc}` package is licensed under the GNU General Public License (GPL v3.0). It is available on CRAN, and can be installed using `install.packages("rempsyc")`. The full tutorial website can be accessed at: <https://rempsyc.remi-theriault.com/>. All code is open-source and hosted on GitHub, and bugs can be reported at <https://github.com/rempsyc/rempsyc/issues/>.

Acknowledgements

I would like to thank Jay Olson, Hugues Leduc, Charles-Étienne Lavoie, and Björn Büdenbender for statistical or technical advice that helped inform some functions of this package and/or useful feedback on this manuscript. I would also like to acknowledge funding from the Social Sciences and Humanities Research Council of Canada.

References

- Aron, A., Aron, E. N., & Smollan, D. (1992). Inclusion of Other in the Self Scale and the structure of interpersonal closeness. *Journal of Personality and Social Psychology*, 63(4), 596. <https://doi.org/10.1037/0022-3514.63.4.596>
- Barbone, J. M., & Garbuszus, J. M. (2023). *openxlsx2: Read, write and edit 'xlsx' files*. <https://github.com/JanMarvin/openxlsx2>
- Gohel, D., & Skintzos, P. (2022). *Flextable: Functions for tabular reporting*. <https://CRAN.R-project.org/package=flextable>
- Kozak, M., & Piepho, H.-P. (2018). What's normal anyway? Residual plots are more telling than significance tests when checking ANOVA assumptions. *Journal of Agronomy and Crop Science*, 204(1), 86–98. <https://doi.org/10.1111/jac.12220>
- Krol, S. A., Thériault, R., Olson, J. A., Raz, A., & Bartz, J. A. (2020). Self-concept clarity and the bodily self: Malleability across modalities. *Personality and Social Psychology Bulletin*, 46(5), 808–820. <https://doi.org/10.1177/0146167219879126>
- Lays, C., Ley, C., Klein, O., Bernard, P., & Licata, L. (2013). Detecting outliers: Do not use standard deviation around the mean, use absolute deviation around the median. *Journal of Experimental Social Psychology*, 49(4), 764–766. <https://doi.org/10.1016/j.jesp.2013.03.013>

- 275 Lüdtke, D., Ben-Shachar, M. S., Patil, I., Waggoner, P., & Makowski, D. (2021). performance:
276 An R package for assessment, comparison and testing of statistical models. *Journal of Open*
277 *Source Software*, 6(60), 3139. <https://doi.org/10.21105/joss.03139>
- 278 Lüdtke, D., Makowski, D., Ben-Shachar, M. S., Patil, I., Wiernik, B. M., Bacher, E., &
279 Thériault, R. (2023). *easystats: Streamline model interpretation, visualization, and reporting*.
280 <https://easystats.github.io/easystats/> (Original work published 2019)
- 281 Lüdtke, D., Patil, I., Ben-Shachar, M. S., Wiernik, B. M., Waggoner, P., & Makowski, D.
282 (2021). see: An R package for visualizing statistical models. *Journal of Open Source Software*,
283 6(64), 3393. <https://doi.org/10.21105/joss.03393>
- 284 Makowski, D., Ben-Shachar, M. S., Patil, I., & Lüdtke, D. (2020). Methods and algorithms
285 for correlation analysis in R. *Journal of Open Source Software*, 5(51), 2306. <https://doi.org/10.21105/joss.02306>
- 286
- 287 Makowski, D., Lüdtke, D., Patil, I., Thériault, R., Ben-Shachar, M. S., & Wiernik, B. M.
288 (2023). *report: Automated reporting of results and statistical models*. <https://easystats.github.io/report/>
289 <https://easystats.github.io/report/> (Original work published 2021)
- 290 Nuijten, M. B., Hartgerink, C. H., Van Assen, M. A., Epskamp, S., & Wicherts, J. M. (2016).
291 The prevalence of statistical reporting errors in psychology (1985–2013). *Behavior Research*
292 *Methods*, 48, 1205–1226. <https://doi.org/10.3758/s13428-015-0664-2>
- 293 Parent, M. C. (2013). Handling item-level missing data: Simpler is just as good. *The*
294 *Counseling Psychologist*, 41(4), 568–600. <https://doi.org/10.1177/0011000012445176>
- 295 Quintana, D. S. (2020). *Five things about open and reproducible science that every early*
296 *career researcher should know*. <https://osf.io/2jt9u>
- 297 R Core Team. (2022). *R: A language and environment for statistical computing*. R Foundation
298 for Statistical Computing. <https://www.R-project.org/>
- 299 Robinson, D., Hayes, A., & Couch, S. (2022). *broom: Convert statistical objects into tidy*
300 *tibbles*. <https://CRAN.R-project.org/package=broom>
- 301 Stanley, D. J., & Spence, J. R. (2018). Reproducible tables in psychology using the apaTables
302 package. *Advances in Methods and Practices in Psychological Science*, 1(3), 415–431. <https://doi.org/10.1177/2515245918773743>
- 303
- 304 Thériault, R., Olson, J. A., Krol, S. A., & Raz, A. (2021). Body swapping with a Black person
305 boosts empathy: Using virtual reality to embody another. *Quarterly Journal of Experimental*
306 *Psychology*, 74(12), 2057–2074. <https://doi.org/10.1177/17470218211024826>
- 307 Wickham, H. (2016). *ggplot2: Elegant graphics for data analysis*. Springer-Verlag New York.
308 <https://ggplot2.tidyverse.org>
- 309 [1] This argument can be used logically, as 'TRUE' or 'FALSE', but can also be provided with
310 a numeric value representing the cut-off threshold for the *p* value
- 311 [2] A great resource for this is the {flextable} e-book: <https://ardata-fr.github.io/flextable-book/>
312 <https://ardata-fr.github.io/flextable-book/>
- 313 [3] Exporting the correlation matrix to an image through the {correlation} package also requires
314 the {see} package (Lüdtke, Patil, et al., 2021)
- 315 [4] For convenience, colours are only used when the corresponding *p* value is at least smaller
316 than .05
- 317 [5] Once one has identified outliers, it is also possible to winsorize them with the
318 `winsorize_mad()` function.