

rempsysc: Convenience functions for psychology

Rémi Thériault ¹

¹ Département de Psychologie, 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Published: unpublished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 affords nice APA tables exportable to Word (via `{flextable}`) and easily customizable APA plots (via `{ggplot2}`). It makes it easy to run statistical tests, check assumptions, and automatize various tasks. It is a package mostly geared at researchers in the psychological sciences but people from all fields can benefit from it.

Statement of need

There are many reasons to use R ([R Core Team, 2022](#)) for analyzing and reporting data from research studies. R is more compatible with the ideals of open science ([Quintana, 2020](#)). In contrast to commercial software: (a) it is free to use; (b) it makes it easy to share a fully comprehensive analysis script; (c) it is transparent as anyone can look at the formulas or algorithms used in a given package; (d) the community can quickly contribute new packages based on current needs; (e) it generates better-looking figures; and (f) it helps reduce copy-paste errors so common in psychology. The latter point 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](#)).

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, mobilize to produce packages that make using R increasingly easier and more user-friendly (e.g., the *easystats* ecosystem [Lüdecke et al., 2019/2023](#)). The `{rempsysc}` package 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

Formatting your table properly in R is already a time-consuming task, but fortunately several packages take care of the formatting within R (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

38 export to Word (e.g., Stanley & Spence, 2018), but their formatting is often rigid especially
39 when using analyzes that are not supported by default.

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

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

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

Term	<i>b</i>	<i>SE</i>	<i>t</i>	<i>p</i>	95% CI
(Intercept)	49.49	3.66	13.51	< .001	[41.97, 57.01]
cyl	-0.37	0.51	-0.72	.479	[-1.41, 0.68]
wt	-7.63	1.52	-5.01	< .001	[-10.75, -4.51]
hp	-0.11	0.03	-3.64	.001	[-0.17, -0.05]
wt × hp	0.03	0.01	3.23	.003	[0.01, 0.04]

53
54 We can do the same with a {report} table.

```
55 library(report)
56
57 stats.table <- lm(mpg ~ cyl + wt * hp, mtcars) |>
58   report() |>
59   as.data.frame()
60
61 nice_table(stats.table)
```

Parameter	Fit	b	95% CI (b)	t	df	p	β	95% CI (β)
(Intercept)		49.49	[41.97, 57.01]	13.51	27	< .001	-0.18	[-0.36, -0.01]
cyl		-0.37	[-1.41, 0.68]	-0.72	27	.479	-0.11	[-0.42, 0.20]
wt		-7.63	[-10.75, -4.51]	-5.01	27	< .001	-0.62	[-0.85, -0.40]
hp		-0.11	[-0.17, -0.05]	-3.64	27	.001	-0.29	[-0.53, -0.04]
wt \times hp		0.03	[0.01, 0.04]	3.23	27	.003	0.29	[0.11, 0.47]
AIC	147.01							
AICc	150.37							
BIC	155.80							
R2	0.89							
R2 (adj.)	0.87							
Sigma	2.17							

62

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

```

67 my_table <- nice_table(
68   stats.table, short = TRUE, highlight = 0.001,
69   title = c("Table 1", "A Pretty Regression Model"),
70   note = c("The data was extracted from the 1974 Motor Trend US magazine.",
71            "Greyed rows represent statistically significant differences, p < .001.")
72 my_table

```

Table 1

A Pretty Regression Model

Parameter	<i>b</i>	<i>t</i>	<i>df</i>	<i>p</i>	β	95% CI (β)
(Intercept)	49.49	13.51	27	< .001	-0.18	[-0.36, -0.01]
cyl	-0.37	-0.72	27	.479	-0.11	[-0.42, 0.20]
wt	-7.63	-5.01	27	< .001	-0.62	[-0.85, -0.40]
hp	-0.11	-3.64	27	.001	-0.29	[-0.53, -0.04]
wt \times hp	0.03	3.23	27	.003	0.29	[0.11, 0.47]

Note. The data was extracted from the 1974 Motor Trend US magazine.

Greyed rows represent statistically significant differences, $p < .001$.

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

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

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

Formattting Results of Analyses

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

t tests

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

Dependent Variable	<i>t</i>	<i>df</i>	<i>p</i>	<i>d</i>	95% CI
mpg	-3.77	18.33	.001	-1.48	[-2.27, -0.67]
disp	4.20	29.26	< .001	1.45	[0.64, 2.23]
drat	-5.65	27.20	< .001	-2.00	[-2.86, -1.12]

89 Contrasts

```
90 nice_contrasts(data = mtcars,
91               response = c("mpg", "disp"),
92               group = "cyl",
93               covariates = "hp") |>
94   nice_table(highlight = .001)
```

Dependent Variable	Comparison	<i>df</i>	<i>t</i>	<i>p</i>	<i>d</i>	95% CI
mpg	4 - 8	28	3.66	.001	3.59	[2.69, 4.49]
	6 - 8	28	1.29	.207	1.44	[0.83, 2.00]
	4 - 6	28	3.64	.001	2.15	[1.40, 3.09]
disp	4 - 8	28	-6.04	<.001	-4.80	[-5.79, -3.86]
	6 - 8	28	-4.86	<.001	-3.29	[-4.29, -2.23]
	4 - 6	28	-2.70	.012	-1.51	[-2.24, -0.88]

95

96 Regressions

```
97 model1 <- lm(mpg ~ cyl + wt * hp, mtcars)
98 model2 <- lm(qsec ~ disp + drat * carb, mtcars)
99
100 nice_lm(list(model1, model2)) |>
101   nice_table(highlight = TRUE)
```

Dependent Variable	Predictor	<i>df</i>	<i>b</i>	<i>t</i>	<i>p</i>	<i>sr</i> ²	95% CI
mpg	cyl	27	-0.37	-0.72	.479	.00	[0.00, 0.01]
	wt	27	-7.63	-5.01	<.001	.11	[0.01, 0.20]
	hp	27	-0.11	-3.64	.001	.06	[0.00, 0.12]
	wt × hp	27	0.03	3.23	.003	.04	[0.00, 0.10]
qsec	disp	27	-0.01	-1.97	.059	.07	[0.00, 0.20]
	drat	27	0.23	0.20	.845	.00	[0.00, 0.01]
	carb	27	1.15	0.72	.479	.01	[0.00, 0.06]
	drat × carb	27	-0.48	-1.08	.289	.02	[0.00, 0.09]

102

103 Simple Slopes

```
104 model1 <- lm(mpg ~ gear * wt, mtcars)
105 model2 <- lm(displ ~ gear * wt, mtcars)
106 my.models <- list(model1, model2)
107
108 nice_lm_slopes(my.models, predictor = "gear", moderator = "wt") |>
109   nice_table()
```

Dependent Variable	Predictor (+/-1 SD)	df	b	t	p	sr ²	95% CI
mpg	gear (LOW-wt)	28	7.54	2.01	.054	.03	[0.00, 0.09]
	gear (MEAN-wt)	28	5.62	1.94	.062	.03	[0.00, 0.08]
	gear (HIGH-wt)	28	3.69	1.80	.083	.02	[0.00, 0.08]
displ	gear (LOW-wt)	28	50.51	0.67	.511	.00	[0.00, 0.02]
	gear (MEAN-wt)	28	35.80	0.61	.545	.00	[0.00, 0.02]
	gear (HIGH-wt)	28	21.08	0.51	.616	.00	[0.00, 0.02]

111 Correlation Matrix

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

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

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

126 When the goal is more exploratory, rather than reporting, and we have large matrices, it can
 127 be more useful to export it to Excel. In `{rempsyc}`, we combine the idea of using a coloured
 128 correlation matrix from the `{correlation}` package with the idea of exporting to Excel using
 129 `{openxlsx2}` (Barbone & Garbuszus, 2023).

130 We also provide some quality of life-improvements, like freezing the first row and column so as
 131 to be able to easily see which variables correlates with which other variable, regardless of how
 132 far or deep those variables are located within the matrix.

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

```
136 cormatrix_excel(data = infert,
137                 filename = "cormatrix1",
```

```
138 select = c("age", "parity", "induced", "case", "spontaneous",
139           "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									

	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									

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)
```



158

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

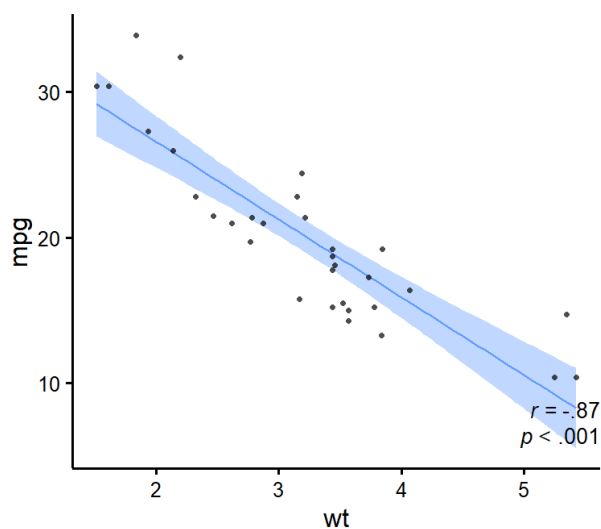
160 One can easily save the resulting figure with `ggplot2::ggsave()`, specifying the desired file
161 name, extension, and resolution.

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

164 Recommended dimensions for saving `{rempsyc}` figures is 7 inches wide and 7 inches high
165 at 300 dpi, which makes sure that the resolution is high enough even if saving to non-vector
166 graphics formats like `.png`. That said, scalable vector graphics formats like `.pdf` or `.eps` are
167 still recommended for high-resolution submissions to scientific journals. Additionally, figures
168 are `{ggplot2}` objects (Wickham, 2016), and can be modified as such.

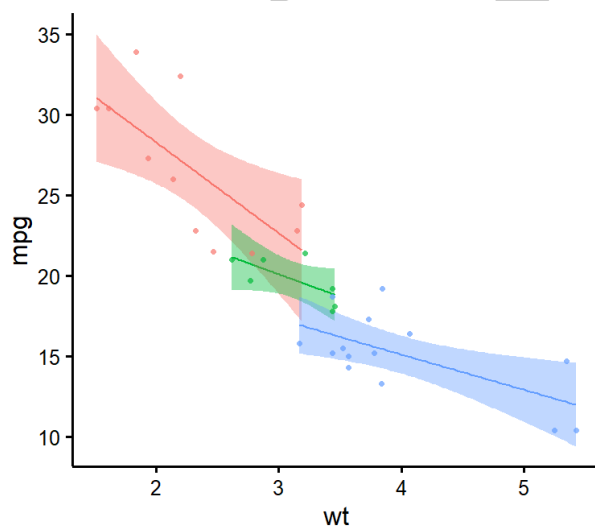
169 Scatter Plots

170 `nice_scatter(data = mtcars,`
171 `predictor = "wt",`
172 `response = "mpg",`
173 `has.confband = TRUE,`
174 `has.r = TRUE,`
175 `has.p = TRUE)`



176

```
177 nice_scatter(data = mtcars,  
178             predictor = "wt",  
179             response = "mpg",  
180             group = "cyl",  
181             has.confband = TRUE)
```



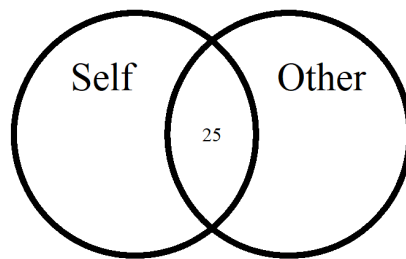
182

183 For an example of such use in publication, see Krol et al. (2020).

184 **Overlapping Circles**

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

```
188 overlap_circle(3.5)
```



189

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

191 Testing assumptions

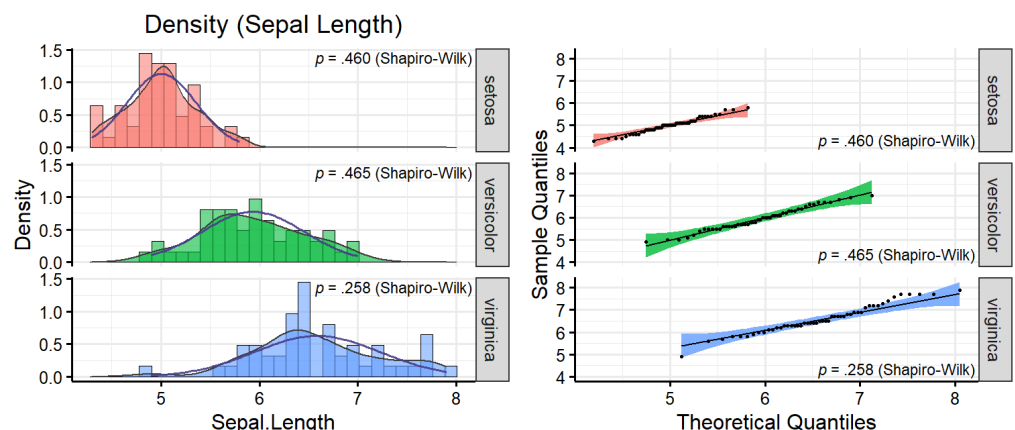
192 When comes time to test assumptions of a linear model, the best option is the `check_model()`
193 function from *easystats*' {performance} package, which allows direct visual evaluation of
194 assumptions (Lüdtke, Ben-Shachar, et al., 2021). Indeed, visual assessment of diagnostic
195 plots is recommended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

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

201 Categorical Predictors

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

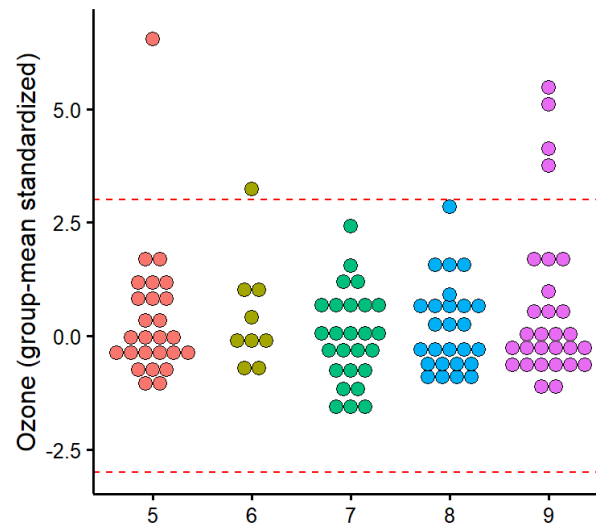
```
205 nice_normality(data = iris,  
206               variable = "Sepal.Length",  
207               group = "Species",  
208               shapiro = TRUE,  
209               histogram = TRUE,  
210               title = "Density (Sepal Length)")
```



211

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

```
213 plot_outliers(airquality,  
214               group = "Month",  
215               response = "Ozone")
```



216
217 Univariate outliers based on the MAD can also be simply requested with `find_mad()[5]`

```
218 find_mad(airquality, names(airquality), criteria = 3)  
219  
220 ## 8 outlier(s) based on 3 median absolute deviations for variable(s):  
221 ## Ozone, Solar.R, Wind, Temp, Month, Day  
222 ##  
223 ## Outliers per variable:  
224 ##  
225 ## $Ozone  
226 ##   Row Ozone_mad  
227 ## 1  30  3.218284  
228 ## 2  62  3.989131  
229 ## 3  99  3.488081  
230 ## 4 101  3.025573  
231 ## 5 117  5.261028  
232 ## 6 121  3.333911  
233 ##  
234 ## $Wind  
235 ##   Row Wind_mad  
236 ## 1   9  3.049871  
237 ## 2  48  3.225825
```

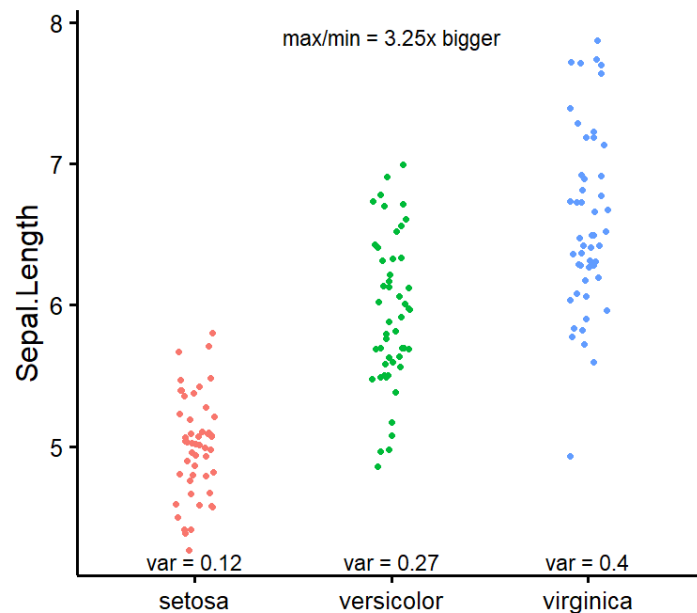
238 Homoscedasticity can also be checked numerically with `nice_var()` or visually with
239 `nice_varplot()`.

```
240 nice_var(data = iris,  
241          variable = names(iris[1:4]),  
242          group = "Species")  
243  
244 ##           Species Setosa Versicolor Virginica Variance.ratio Criteria  
245 ## 1 Sepal.Length  0.124      0.266      0.404           3.3         4  
246 ## 2 Sepal.Width  0.144      0.098      0.104           1.5         4  
247 ## 3 Petal.Length  0.030      0.221      0.305          10.2         4
```

```

248 ## 4 Petal.Width 0.011      0.039      0.075      6.8      4
249 ## Heteroscedastic
250 ## 1 FALSE
251 ## 2 FALSE
252 ## 3 TRUE
253 ## 4 TRUE
254
255 nice_varplot(data = iris,
256              variable = "Sepal.Length",
257              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.

Availability

The `{rempsyc}` package 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/>.

Acknowledgements

I would like to thank Hugues Leduc, Jay Olson, 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>
- Leys, 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>
- Lüdtke, D., Ben-Shachar, M. S., Patil, I., Waggoner, P., & Makowski, D. (2021). performance: An R package for assessment, comparison and testing of statistical models. *Journal of Open Source Software*, 6(60), 3139. <https://doi.org/10.21105/joss.03139>
- Lüdtke, D., Makowski, D., Ben-Shachar, M. S., Patil, I., Wiernik, B. M., Bacher, E., & Thériault, R. (2023). *easystats: Streamline model interpretation, visualization, and reporting*. <https://easystats.github.io/easystats/> (Original work published 2019)
- Lüdtke, D., Patil, I., Ben-Shachar, M. S., Wiernik, B. M., Waggoner, P., & Makowski, D. (2021). see: An R package for visualizing statistical models. *Journal of Open Source Software*, 6(64), 3393. <https://doi.org/10.21105/joss.03393>
- Makowski, D., Ben-Shachar, M. S., Patil, I., & Lüdtke, D. (2020). Methods and algorithms for correlation analysis in R. *Journal of Open Source Software*, 5(51), 2306. <https://doi.org/10.21105/joss.02306>
- Makowski, D., Lüdtke, D., Patil, I., Thériault, R., Ben-Shachar, M. S., & Wiernik, B. M. (2023). *report: Automated reporting of results and statistical models*. <https://easystats.github.io/report/> (Original work published 2021)
- Nuijten, M. B., Hartgerink, C. H., Van Assen, M. A., Epskamp, S., & Wicherts, J. M. (2016). The prevalence of statistical reporting errors in psychology (1985–2013). *Behavior Research Methods*, 48, 1205–1226. <https://doi.org/10.3758/s13428-015-0664-2>
- Parent, M. C. (2013). Handling item-level missing data: Simpler is just as good. *The Counseling Psychologist*, 41(4), 568–600. <https://doi.org/10.1177/0011000012445176>
- Quintana, D. S. (2020). *Five things about open and reproducible science that every early career researcher should know*. <https://osf.io/2jt9u>

- 320 R Core Team. (2022). *R: A language and environment for statistical computing*. R Foundation
321 for Statistical Computing. <https://www.R-project.org/>
- 322 Robinson, D., Hayes, A., & Couch, S. (2022). *broom: Convert statistical objects into tidy*
323 *tibbles*. <https://CRAN.R-project.org/package=broom>
- 324 Stanley, D. J., & Spence, J. R. (2018). Reproducible tables in psychology using the apaTables
325 package. *Advances in Methods and Practices in Psychological Science*, 1(3), 415–431. <https://doi.org/10.1177/2515245918773743>
- 326
- 327 Thériault, R., Olson, J. A., Krol, S. A., & Raz, A. (2021). Body swapping with a Black person
328 boosts empathy: Using virtual reality to embody another. *Quarterly Journal of Experimental*
329 *Psychology*, 74(12), 2057–2074. <https://doi.org/10.1177/17470218211024826>
- 330 Wickham, H. (2016). *ggplot2: Elegant graphics for data analysis*. Springer-Verlag New York.
331 <https://ggplot2.tidyverse.org>
- 332 [1] This argument can be used logically, as 'TRUE' or 'FALSE', but can also be provided with
333 a numeric value representing the cut-off threshold for the p value
- 334 [2] A great resource for this is the {flextable} e-book: [https://ardata-fr.github.io/](https://ardata-fr.github.io/flextable-book/)
335 [flextable-book/](https://ardata-fr.github.io/flextable-book/)
- 336 [3] Exporting the correlation matrix to an image through the {correlation} package also requires
337 the {see} package (Lüdtke, Patil, et al., 2021)
- 338 [4] For convenience, colours are only used when the corresponding p value is at least smaller
339 than .05
- 340 [5] Once one has identified outliers, it is also possible to winsorize them with the
341 `winsorize_mad()` function.