

rempsysc: Convenience functions for psychology

Rémi Thériault  ¹

¹ Departement 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Published: unpublished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 affords easily customizable APA plots (via `{ggplot2}`) and nice APA tables exportable to Word (via `{flectable}`). 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 a substantial one because 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 R 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 increasingly does, mobilize to produce packages that make using R as easy as possible (e.g., the *easystats* ecosystem [Lüdtke 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, Robinson et al. (2022); Makowski et al. (2021/2023); 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 model <- lm(mpg ~ cyl + wt * hp, mtcars)
50 tidy(model, conf.int = TRUE) |>
51   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]

52
53 We can do the same with a {report} table.

```
54 library(report)
55 model <- lm(mpg ~ cyl + wt * hp, mtcars)
56 stats.table <- as.data.frame(report(model))
57
58 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							

59

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

```
64 my_table <- nice_table(  
65   stats.table, short = TRUE, highlight = 0.001,  
66   title = c("Table 1", "A Pretty Regression Model"),  
67   note = c("The data was extracted from the 1974 Motor Trend US magazine.",  
68            "* p < .05, ** p < .01, *** p < .001"))  
69 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.

* $p < .05$, ** $p < .01$, *** $p < .001$

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]

86 Contrasts

```
87 nice_contrasts(data = mtcars,
88               response = c("mpg", "disp"),
89               group = "cyl",
90               covariates = "hp") |>
91   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	[2.75, 4.50]
	6 - 8	28	1.29	.207	1.44	[0.83, 1.98]
	4 - 6	28	3.64	.001	2.15	[1.29, 3.10]
disp	4 - 8	28	-6.04	<.001	-4.80	[-5.82, -3.81]
	6 - 8	28	-4.86	<.001	-3.29	[-4.34, -2.19]
	4 - 6	28	-2.70	.012	-1.51	[-2.23, -0.87]

92

93 Regressions

```
94 model1 <- lm(mpg ~ cyl + wt * hp, mtcars)
95 model2 <- lm(qsec ~ disp + drat * carb, mtcars)
96
97 nice_lm(list(model1, model2)) |>
98   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]

99

100 Simple Slopes

```
101 model1 <- lm(mpg ~ gear * wt, mtcars)
102 model2 <- lm(displ ~ gear * wt, mtcars)
103 my.models <- list(model1, model2)
104
105 nice_lm_slopes(my.models, predictor = "gear", moderator = "wt") |>
106   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]

108 Correlation Matrix

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

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

118 Another approach is to export to an image, like {correlation} package does (Makowski et al.,
 119 2020). For very small matrices, this works extremely well, and the colour is an immense help
 120 to quickly identify which correlations are strong or weak, positive or negative. Again, however,
 121 this does not work so well for large matrices because labels might overlap or navigating the
 122 large figure becomes difficult.

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

127 We also provide some quality of life-improvements, like freezing the first row and column so as
 128 to be able to easily see to which variables the correlations relate, regardless of how far or deep
 129 we are within the large correlation matrix.

130 The colour represents the strength of the correlation, whereas the stars represent how significant
 131 the *p* value is.[3] The exact *p* values are provided in a second tab for reference purposes, so all
 132 information is readily available in a convenient format.

```
133 cormatrix_excel(data = infert,
134                 filename = "cormatrix1",
```

```
135 select = c("age", "parity", "induced", "case", "spontaneous",
136           "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									

139 Publication-Ready Figures

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

144 This is why {rempsyc} prepares a few plot types for you, so they are ready to be saved to your
 145 preferred format (.pdf, .tiff, or .png).

146 Violin Plots

```
147 nice_violin(data = ToothGrowth,
148             group = "dose",
149             response = "len",
150             xlabels = c("Low", "Medium", "High"),
151             comp1 = 1,
152             comp2 = 3,
153             has.d = TRUE,
154             d.y = 30)
```



155

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

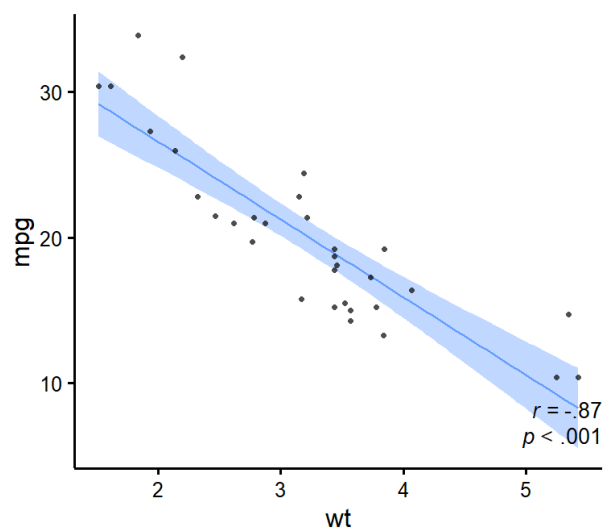
157 One can easily save the resulting figure with `ggplot2::ggsave()`, specifying the desired file
158 name, extension, and resolution.

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

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

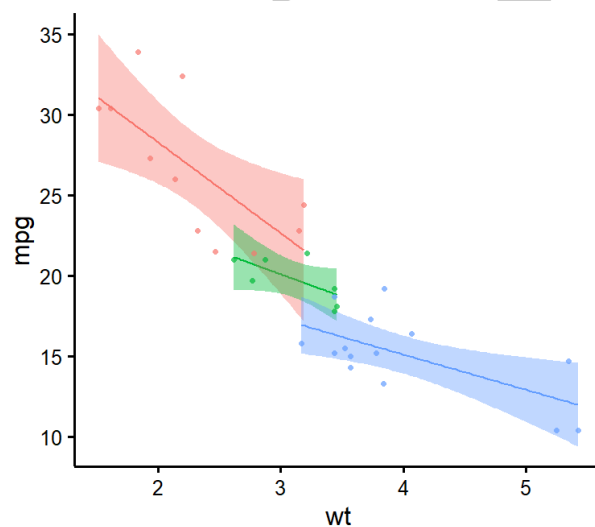
166 Scatter Plots

167 `nice_scatter(data = mtcars,`
168 `predictor = "wt",`
169 `response = "mpg",`
170 `has.confband = TRUE,`
171 `has.r = TRUE,`
172 `has.p = TRUE)`



173

```
174 nice_scatter(data = mtcars,  
175               predictor = "wt",  
176               response = "mpg",  
177               group = "cyl",  
178               has.confband = TRUE)
```



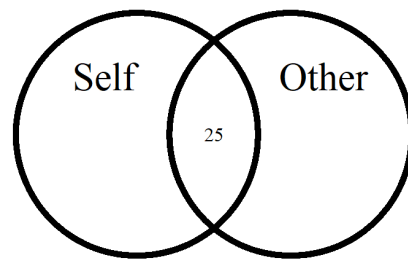
179

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

181 Overlapping Circles

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

```
185 overlap_circle(3.5)
```



186

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

188 Testing assumptions

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

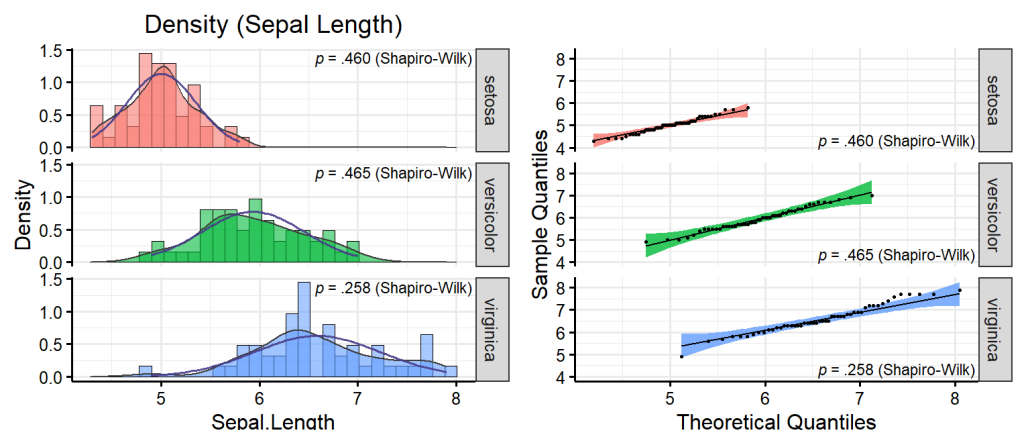
194 That said, if for whatever reason one wants to check objective assumption tests for a linear
195 model, `rempsysc` makes this easy with the `nice_assumptions()` function, which provide p
196 values for normality (Shapiro-Wilk), homoscedasticity (Breusch-Pagan) and autocorrelation of
197 residuals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

198 Categorical Predictors

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

```

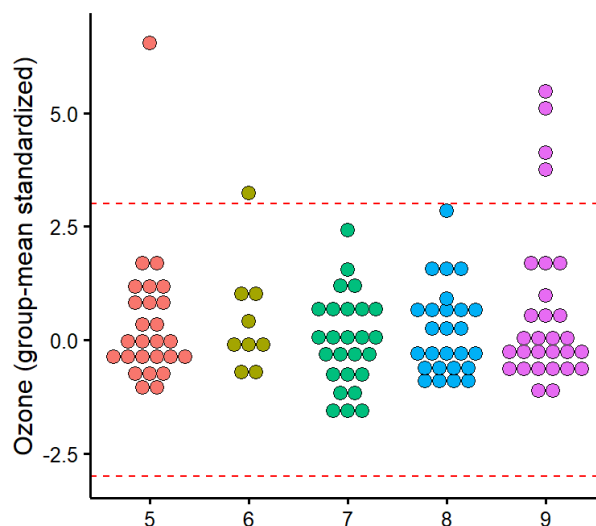
202 nice_normality(data = iris,
203               variable = "Sepal.Length",
204               group = "Species",
205               shapiro = TRUE,
206               histogram = TRUE,
207               title = "Density (Sepal Length)")
  
```



208

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

```
210 plot_outliers(airquality,  
211               group = "Month",  
212               response = "Ozone")
```



213

214 Univariate outliers based on the MAD can also be simply requested with `find_mad()`.^[4]

```
215 find_mad(airquality, names(airquality), criteria = 3)
```

216

```
217 ## 8 outlier(s) based on 3 median absolute deviations for variable(s):
```

```
218 ## Ozone, Solar.R, Wind, Temp, Month, Day
```

```
219 ##
```

```
220 ## Outliers per variable:
```

```
221 ##
```

```
222 ## $Ozone
```

```
223 ##   Row Ozone_mad
```

```
224 ## 1  30  3.218284
```

```
225 ## 2  62  3.989131
```

```
226 ## 3  99  3.488081
```

```
227 ## 4 101  3.025573
```

```
228 ## 5 117  5.261028
```

```
229 ## 6 121  3.333911
```

```
230 ##
```

```
231 ## $Wind
```

```
232 ##   Row Wind_mad
```

```
233 ## 1   9  3.049871
```

```
234 ## 2  48  3.225825
```

235 Homoscedasticity can also be checked numerically with `nice_var()` or visually with

236 `nice_varplot()`.

```
237 nice_var(data = iris,  
238          variable = names(iris[1:4]),  
239          group = "Species")
```

240

```
241 ##           Species Setosa Versicolor Virginica Variance.ratio Criteria
```

```
242 ## 1 Sepal.Length  0.124      0.266      0.404           3.3         4
```

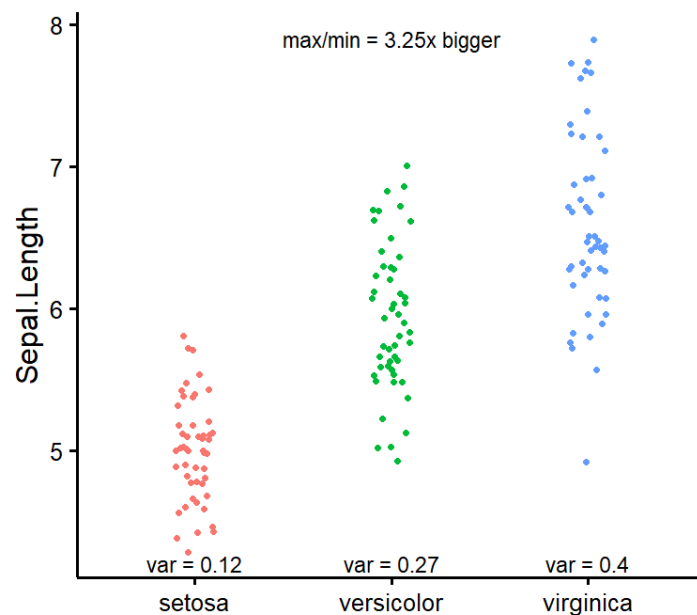
```
243 ## 2 Sepal.Width  0.144      0.098      0.104           1.5         4
```

```
244 ## 3 Petal.Length 0.030      0.221      0.305          10.2         4
```

```

245 ## 4 Petal.Width 0.011      0.039      0.075      6.8      4
246 ## Heteroscedastic
247 ## 1 FALSE
248 ## 2 FALSE
249 ## 3 TRUE
250 ## 4 TRUE
251
252 nice_varplot(data = iris,
253              variable = "Sepal.Length",
254              group = "Species")

```



Utility functions

Finally, with the idea of making the analysis workflow easier in mind, `{rempsys}` also has 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 `{rempsys}` package is available on CRAN, and can be installed using `install.packages("rempsys")`. The full tutorial website can be accessed at: <https://rempsys.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)
- 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>
- R Core Team. (2022). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing. <https://www.R-project.org/>

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

DRAFT