The Journal of Open Source Software

rempsyc: Convenience functions for psychology and social sciences

Rémi Thériault [1]

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

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor: Open Journals

Reviewers:

- @openjournals

Submitted: 01 January 1970
Published: unpublished

Summary

{rempsyc} is an R package of convenience functions that make the analysis-to-publication workflow faster, easier, and less error-prone. It enables the creation of publication-ready tables exportable to Word (via {flextable}) and easily customizable APA-compliant plots (via {ggplot2}). It makes it easy to run statistical tests, check assumptions, and automate various tasks common in psychological and social scientific 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 Lüdecke et al., 2019/2023). The {rempsyc} 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. {rempsyc} does not generate publication-ready text summarizing analyses; for this, see the {report} package (Makowski et al., 2021/2023). Instead, {rempsyc} 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

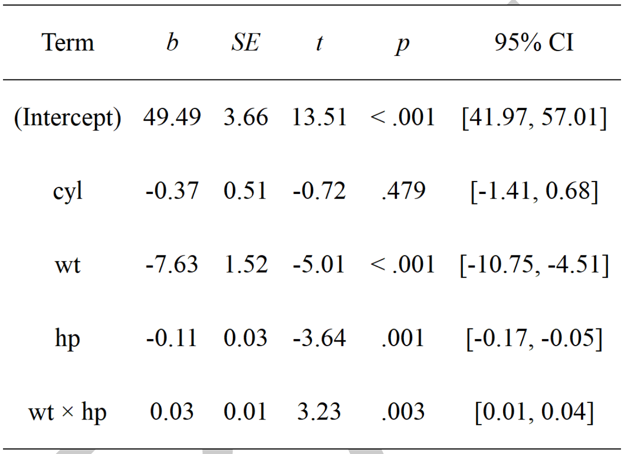
Many researchers using R still copy-paste the values from the R console to their manuscript, or retype them manually. Yet, this approach increases the risks of copy-paste and retyping errors so common in psychology. This problem is not trivial given that according to some estimates, up to 50% of articles 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 a table properly in R is already a tedious and 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.

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

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

```
library(rempsyc)

lm(mpg ~ cyl + wt * hp, mtcars) |>
  broom::tidy(conf.int = TRUE) |>
  nice_table(broom = "lm")
```



Term	*b*	*SE*	*t*	*p*	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]

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

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

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 × 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							

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

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

Table 1

A Pretty Regression Model

Parameter	*b*	*t*	*df*	*p*	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 × 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

{rempsyc} 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	*t*	*df*	*p*	*d*	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]

Contrasts

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

Dependent Variable	Comparison	*df*	*t*	*p*	*d*	95% CI
mpg	4 - 8	28	3.66	.001	3.59	[2.62, 4.50]
	6 - 8	28	1.29	.207	1.44	[0.85, 2.02]
	4 - 6	28	3.64	.001	2.15	[1.36, 3.09]
disp	**4 - 8**	**28**	**-6.04**	**< .001**	**-4.80**	**[-5.80, -3.91]**
	6 - 8	**28**	**-4.86**	**< .001**	**-3.29**	**[-4.25, -2.32]**
	4 - 6	28	-2.70	.012	-1.51	[-2.24, -0.91]

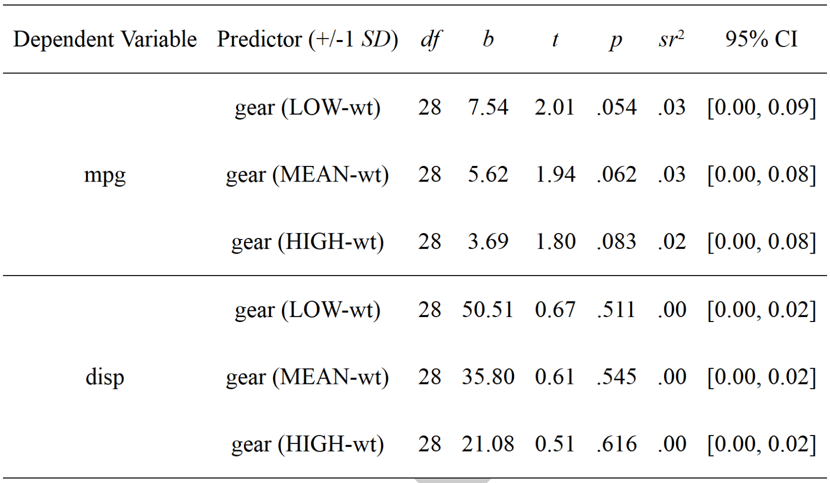
Regressions

```
model1 <- lm(mpg ~ cyl + wt * hp, mtcars)
model2 <- lm(qsec ~ disp + drat * carb, mtcars)

nice_lm(list(model1, model2)) |>
  nice_table(highlight = TRUE)
```

Dependent Variable	Predictor	*df*	*b*	*t*	*p*	*sr*^{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]

Simple Slopes

```
model1 <- lm(mpg ~ gear * wt, mtcars)
model2 <- lm(disp ~ gear * wt, mtcars)
my.models <- list(model1, model2)

nice_lm_slopes(my.models, predictor = "gear", moderator = "wt") |>
  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]
disp	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]

Correlation Matrices

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

The {apaTables} package (Stanley & Spence, 2018) allows exporting the correlation matrix to Word in an APA format, and in many cases this already meets the formal requirements of APA style. Hovever, the Word format is not suitable for large matrices, as it will often spread beyond the document's margin limits.

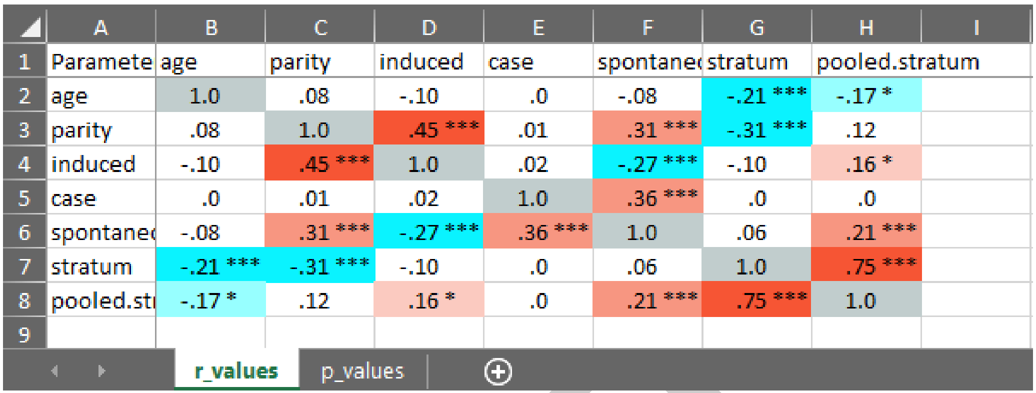
Another approach is to export the correlation matrix to an image, like the {correlation} package does (Makowski et al., 2020).[3] For very small matrices, this works extremely well, and the color is an immense help to quickly identify which correlations are strong or weak, positive or negative, and significant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

When the goal is more exploratory in nature, and one has large matrices, it can be beneficial to export them to Excel. In {rempsyc}, I combine the idea of using a colored correlation matrix from the {correlation} package with the idea of exporting to Excel using {openxlsx2} (Barbone & Garbuszus, 2023).

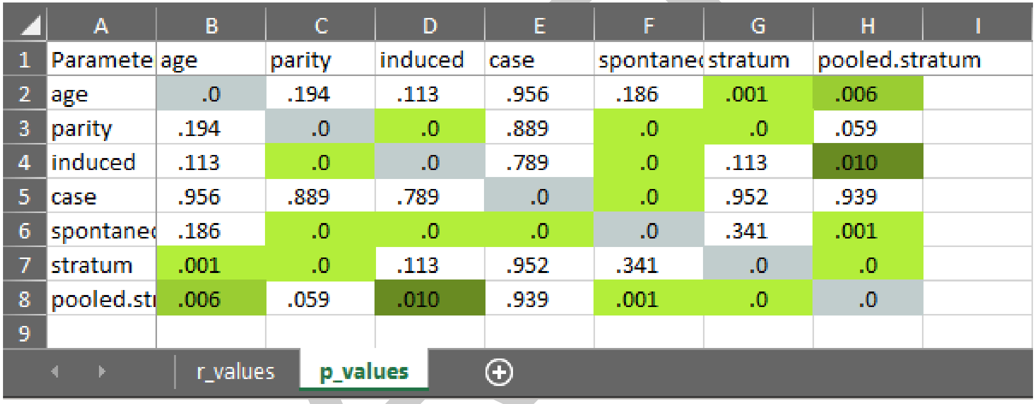
The {rempsyc} package also provides some usability improvements, like freezing the first row and column so as to be able to easily see which variables correlates with which other variable, regardless of how far or deep those variables are located within the matrix.

The color represents the strength of the correlation, whereas the stars represent different significance treshholds for the *p* value.[4] The exact *p* values are provided in a second tab for reference purposes, so all information is readily available in just one convenient function call.

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



	A	B	C	D	E	F	G	H	I
1	Paramete	age	parity	induced	case	spontane	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	spontane	-.08	.31 ***	-.27 ***	.36 ***	1.0	.06	.21 ***	
7	stratum	-.21 ***	-.31 ***	-.10	.0	.06	1.0	.75 ***	
8	pooled.st	-.17 *	.12	.16 *	.0	.21 ***	.75 ***	1.0	
9									



	A	B	C	D	E	F	G	H	I
1	Paramete	age	parity	induced	case	spontane	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	spontane	.186	.0	.0	.0	.0	.341	.001	
7	stratum	.001	.0	.113	.952	.341	.0	.0	
8	pooled.st	.006	.059	.010	.939	.001	.0	.0	
9									

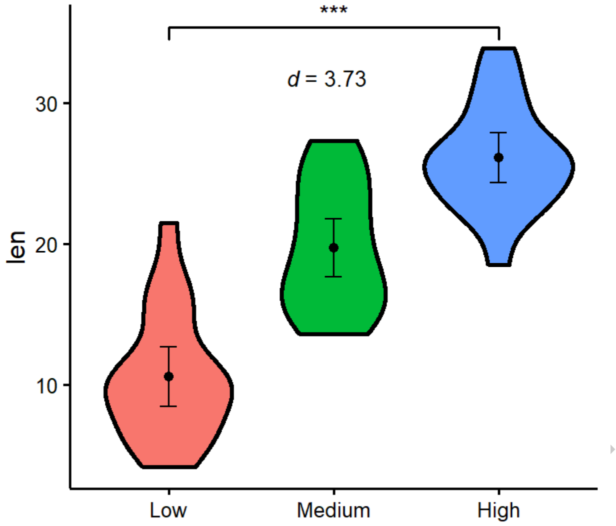
Publication-Ready Figures

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)
```



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.

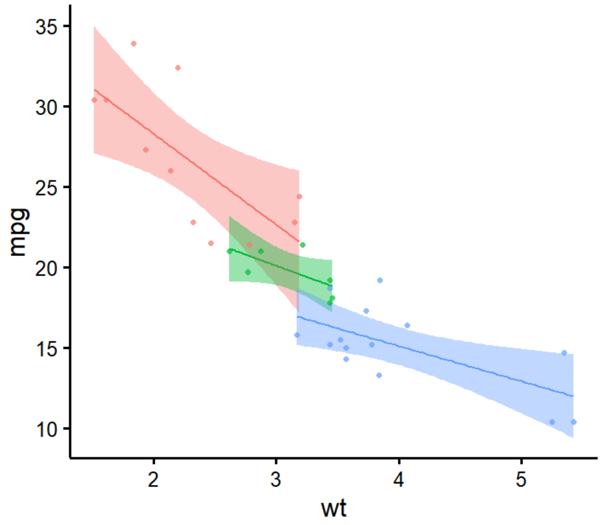
```
ggplot2::ggsave('nice_violinplothere.pdf', width = 7, height = 7,
                unit = 'in', dpi = 300)
```

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

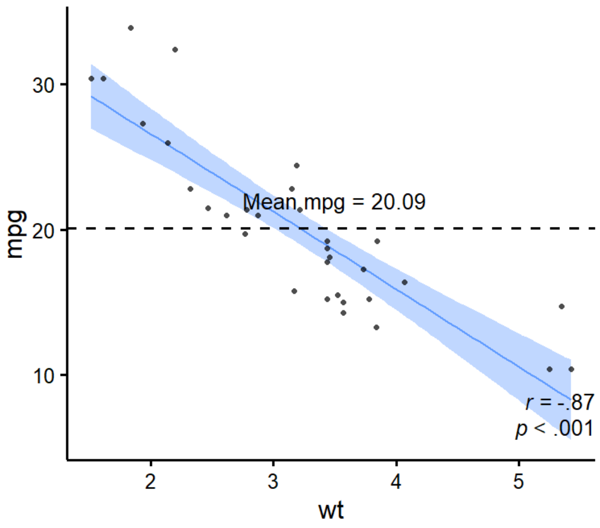
Scatter Plots

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

```
nice_scatter(data = mtcars,
             predictor = "wt",
             response = "mpg",
             group = "cyl",
             has.confband = TRUE)
```



```r
nice_scatter(data = mtcars,
             predictor = "wt",
             response = "mpg",
             has.confband = TRUE,
             has.r = TRUE,
             has.p = TRUE) +
  ggplot2::geom_hline(yintercept = mean(mtcars$mpg), colour = "black",
                      linewidth = 1.4, linetype = "dashed") +
  ggplot2::annotate("text", x = 3.5, y = 22, size = 7,
                    label = paste("Mean mpg =", round(mean(mtcars$mpg), 2)))
```

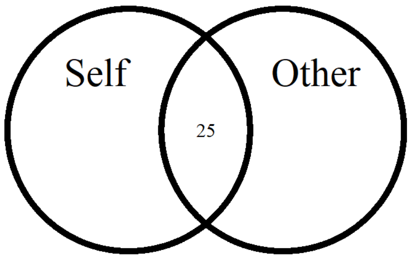


For an example of such use in publication, see Krol et al. (2020).

Overlapping Circles

For psychologists using the Inclusion of Other in the the Self Scale (Aron et al., 1992), it can be useful to interpolate the original discrete scores (1 to 7) into a group average representation of the conceptual self-other overlap. For example, assuming the group mean is 3.5 on the 1 to 7 scale, overlap_circle() will draw a 25% overlap from interpolation:

```r
overlap_circle(3.5)
```



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

Testing assumptions

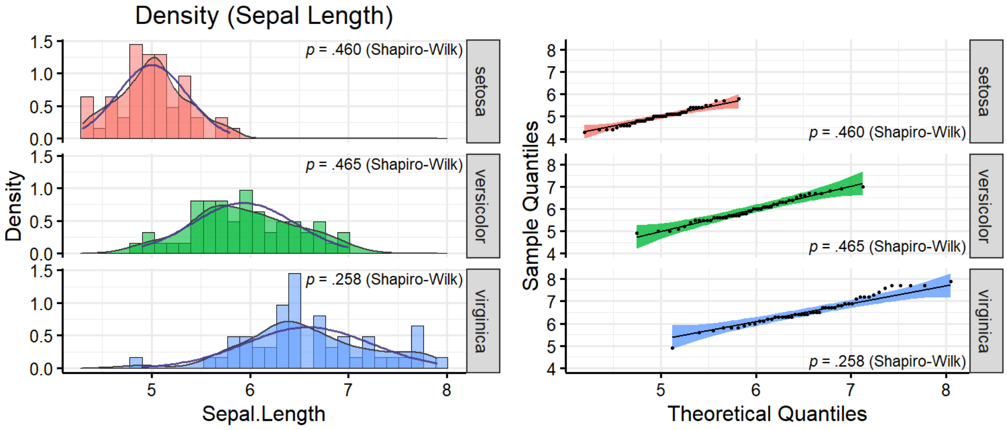
When comes time to test assumptions of a linear model, the best option is the `check_model()` function from *easystats*' {performance} package, which allows direct visual evaluation of assumptions (Lüdecke, Ben-Shachar, et al., 2021). Indeed, visual assessment of diagnostic plots is recommended over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

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

Categorical Predictors

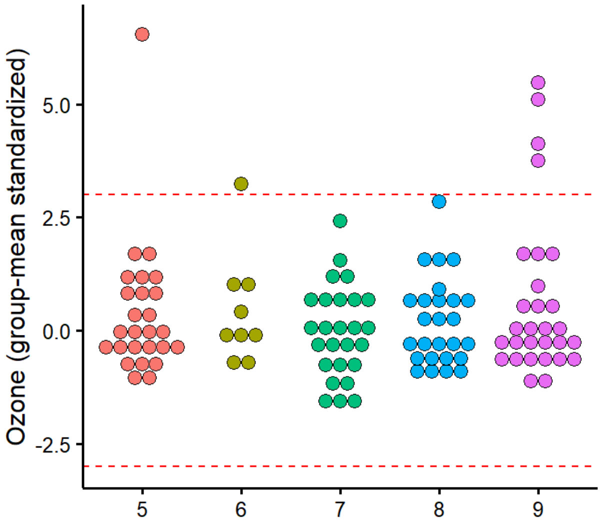
`nice_normality()` makes it easy to visually check normality in the case of categorical predictors (i.e., when using groups), through a combination of quantile-quantile plots, density plots, and histograms.

```
nice_normality(data = iris,
               variable = "Sepal.Length",
               group = "Species",
               shapiro = TRUE,
               histogram = TRUE,
               title = "Density (Sepal Length)")
```



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

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



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

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

```
## 8 outlier(s) based on 3 median absolute deviations for variable(s):
##  Ozone, Solar.R, Wind, Temp, Month, Day
##
## Outliers per variable:
##
## $Ozone
##   Row Ozone_mad
## 1  30  3.218284
## 2  62  3.989131
## 3  99  3.488081
## 4 101  3.025573
## 5 117  5.261028
## 6 121  3.333911
##
## $Wind
##   Row Wind_mad
## 1   9 3.049871
## 2  48 3.225825
```

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

```
nice_var(data = iris,
         variable = names(iris[1:4]),
         group = "Species") |>
  nice_table()
```

```
nice_varplot(data = iris,
             variable = "Sepal.Length",
             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

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

Lüdecke, 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

Lüdecke, 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)

Lüdecke, 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., & Lüdecke, 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., Lüdecke, 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/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%2F0011000012445176

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/

Robinson, D., Hayes, A., & Couch, S. (2022). *broom: Convert statistical objects into tidy tibbles*. https://CRAN.R-project.org/package=broom

Stanley, D. J., & Spence, J. R. (2018). Reproducible tables in psychology using the apaTables package. *Advances in Methods and Practices in Psychological Science*, *1*(3), 415–431. https://doi.org/10.1177/2515245918773743

Thériault, R., Olson, J. A., Krol, S. A., & Raz, A. (2021). Body swapping with a Black person boosts empathy: Using virtual reality to embody another. *Quarterly Journal of Experimental Psychology*, *74*(12), 2057–2074. https://doi.org/10.1177/17470218211024826

Wickham, H. (2016). *ggplot2: Elegant graphics for data analysis*. Springer-Verlag New York. https://ggplot2.tidyverse.org

[1] This argument can be used logically, as `'TRUE'` or `'FALSE'`, but can also be provided with a numeric value representing the cut-off threshold for the *p* value

[2] A great resource for this is the {flextable} e-book: https://ardata-fr.github.io/flextable-book/

[3] Exporting the correlation matrix to an image through the {correlation} package also requires the {see} package (Lüdecke, Patil, et al., 2021)

[4] For convenience, colours are only used when the corresponding *p* value is at least smaller than .05

[5] Once one has identified outliers, it is also possible ot winsorize them with the `winsorize_mad()` function.