

easystats: A Unified Framework for Statistical Analysis in R

Daniel Lüdecke¹, Mattan S. Ben-Shachar², Indrajeet Patil³, Brenton M. Wiernik⁴, Etienne Bacher⁵, Rémi Thériault⁶, Philip D. Waggoner^{7, 8}, and Dominique Makowski^{9, 10}

1 University Medical Center Hamburg-Eppendorf, Germany **2** Independent Researcher, Ramat Gan, Israel **3** Carl Zeiss AG, Munich, Germany **4** Independent Researcher, Tampa, FL, USA **5** Luxembourg Institute of Socio-Economic Research (LISER), Luxembourg **6** Department of Psychology, New York University, New York, NY, USA **7** Department of Applied Mathematics and Statistics, Colorado School of Mines, Golden, CO, USA **8** School of Medicine, Stanford University, Stanford, CA, USA **9** School of Psychology, University of Sussex, Brighton, UK **10** Sussex Centre for Consciousness Science, University of Sussex, Brighton, UK

DOI:

Software

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

Submitted:

Published:

License

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

Summary

The **easystats** project is a collection of R packages that provides a unified and intuitive framework for data wrangling and statistical analysis, with *accessibility* as its core driving principle. In order to fulfil this vision, the packages of the ecosystem implement functionalities designed to be **user-friendly** (e.g., transparent names of functions and arguments, comprehensive documentation, sensible defaults), **consistent** (similar syntax and design principles across packages), and **interoperable** (seamless integration between packages).

The packages are specialized for different stages of the statistical workflow, such as data wrangling (`{datawizard}`, Patil et al., 2022), model assessment (`{performance}`, Lüdecke, Ben-Shachar, et al., 2021), understanding and describing model parameters (`{parameters}`, Lüdecke et al., 2020), including Bayesian models (`{bayestestR}`, Makowski, Ben-Shachar, & Lüdecke, 2019; Makowski, Ben-Shachar, Chen, et al., 2019), computation of effect sizes (`{effectsize}`, Ben-Shachar et al., 2020), calculating and visualizing marginal effects (`{modelbased}`, Makowski et al., 2025), and generating publication-ready figures (`{see}`, Lüdecke, Patil, et al., 2021) or reports of statistical models (`{report}`, Makowski et al., 2023).

Statement of Need

R is a powerful language for statistical computing, but its capabilities are scattered across a fragmented landscape of packages. Conducting a full analysis consisting of data manipulation, modeling, diagnostics, interpretation, and visualization, often requires juggling multiple tools with different syntax, design principles, outputs, and classes. This creates barriers for newcomers and inefficiencies even for experienced users.

The **easystats** ecosystem addresses this challenge by enabling a seamless workflow from data exploration to result communication, while nudging users toward good, reproducible and transparent statistical practices with sensible defaults and clear documentation. The packages in this ecosystem share consistent syntax and integrate seamlessly, making robust

analysis more accessible while reducing cognitive load for novice and experienced R users alike.

The modular and lightweight nature of the **easystats** ecosystem enables developers to use and integrate in other packages only the necessary components. For example, `{insight}`, a dependency-free package for retrieving model information, is utilized by 45 other CRAN packages, such as `{marginaleffects}` (Arel-Bundock et al., 2024) and `{gtsummary}` (Sjoberg et al., 2021), and `{parameters}` is used by 22. In contrast, the `{easystats}` meta-package provides users with a cohesive experience, granting access to the entire ecosystem and its consistent design principles without needing to know the specific package of each function.

Other R ecosystems or packages often serve different purposes. The `{tidyverse}` (Wickham et al., 2019), for example, provides foundational framework for data manipulation and general-purpose visualisation but does not focus on the intricacies of statistical model interpretation and reporting. Specialist packages like `{lme4}` (Bates et al., 2015) for mixed-effects models or `{brms}` (Bürkner, 2017) for Bayesian analysis are essential tools, but **easystats** serves as a complementary meta-layer that provides a single, easy-to-learn interface for interacting with the outputs from these and many other modeling packages. This allows analysts and researchers to focus on scientific questions rather than the technical idiosyncrasies of software implementations. **easystats** meets a critical need when doing statistics in R by delivering a coherent, intuitive suite of tools that span the statistical modeling pipeline.

A Harmonized and Integrated Workflow

A key design principle of the **easystats** ecosystem is the harmonization and integration of different packages into a simple, sequential workflow. The typical workflow for a statistical analysis using `{easystats}` starts with importing data and bringing the data into shape for the next step—fitting a model—and then sequentially using different functions to obtain a comprehensive understanding of the model. This can include checking the model’s parameters, performance metrics, specific effect sizes (e.g., Ben-Shachar et al., 2023), and statistical outliers (Thériault et al., 2024), as well as obtaining publication-ready figures and written summaries of the results.

Let’s demonstrate this with an example, where the user wants to fit a logistic mixed effects:

```
# we don't load each package individually,  
# but rather the entire ecosystem  
library(easystats)  
data(coffee_data, package = "modelbased")  
  
# dichotomize outcome variable  
coffee_data$alertness <- categorize(coffee_data$alertness, lowest = 0)  
  
# rename variable  
coffee_data <- data_rename(coffee_data, select = c(treatment = "coffee"))  
  
# fit mixed model  
model <- glmmTMB::glmmTMB(  
  alertness ~ treatment + (1 | ID),  
  data = coffee_data,  
  family = binomial()  
)
```

The `model` object can then be passed to functions from different **easystats** packages. For instance, the user can get a summary of the model parameters using the `{parameters}` package:

```
model_parameters(model)
#> # Fixed Effects
#>
#> Parameter          / Log-Odds /   SE /          95% CI /    z /    p
#> -----
#> (Intercept)          /    0.13 / 0.26 / [-0.37, 0.64] /  0.52 / 0.606
#> treatment [control] /   -0.27 / 0.37 / [-0.99, 0.45] / -0.73 / 0.466
#>
#> # Random Effects
#>
#> Parameter          / Coefficient /          95% CI
#> -----
#> SD (Intercept: ID) /        0.05 / [0.00, 9.86e+28]
```

Then, the performance of the model can be assessed with the `{performance}` package:

```
model_performance(
  model,
  metrics = c("AIC", "BIC", "R2")
)
#> # Indices of model performance
#>
#> AIC   /   BIC / R2 (cond.) / R2 (marg.)
#> -----
#> 171.8 / 180.2 /    0.006 /    0.005
```

The results can be visualized using the `{see}` package by, for example, plotting the model's predictions from `{modelbased}` (Figure 1):

```
predictions <- estimate_means(model, "treatment")
plot(predictions) + theme_modern(show.ticks = TRUE) # add nice theme
```

Finally, a full report of the analysis can be generated with the `{report}` package:

```
report(model)
#> We fitted a logistic mixed model (estimated using ML and nlminb
#> optimizer) to predict alertness with treatment (formula: alertness ~
#> treatment). The model included ID as random effect (formula: ~1 |
#> ID). The model's total explanatory power is very weak (conditional
#> R2 = 6.35e-03) and the part related to the fixed effects alone
#> (marginal R2) is of 5.44e-03. The model's intercept, corresponding
#> to treatment = coffee, is at 0.13 (95% CI [-0.37, 0.64], p = 0.606).
#> Within this model:
#>
#> - The effect of treatment [control] is statistically non-significant
#> and negative (beta = -0.27, 95% CI [-0.99, 0.45], p = 0.466)
#>
#> Standardized parameters were obtained by fitting the model on a
#> standardized version of the dataset. 95% Confidence Intervals (CIs)
#> and p-values were computed using a Wald z-distribution
#> approximation.
```

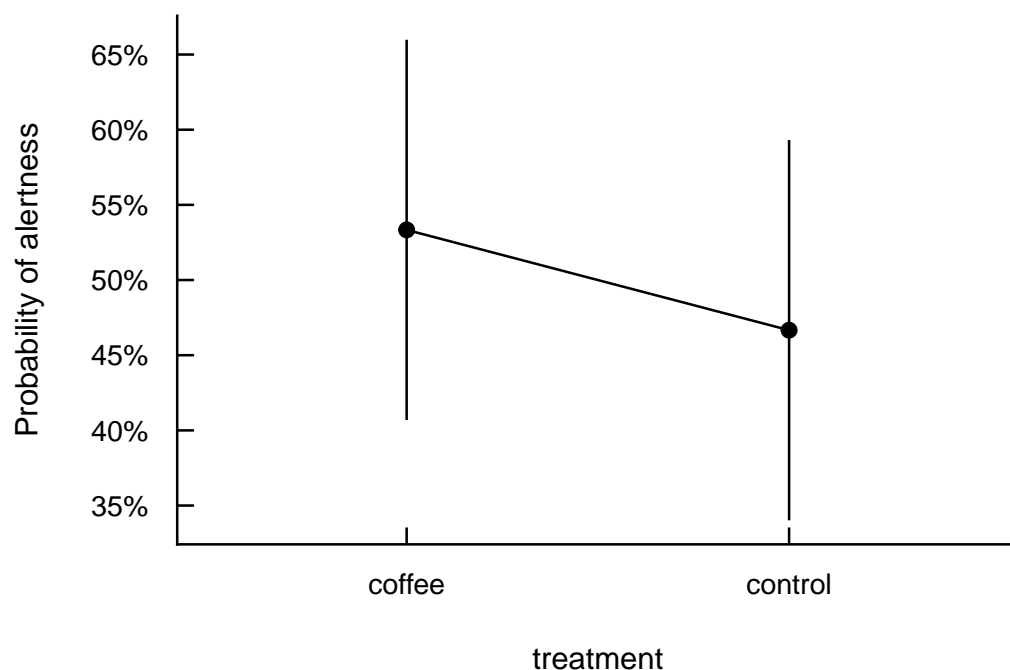


Figure 1: Predicted probability of alertness by treatment group.

The `model_dashboard()` function generates an interactive model summary, integrating several **easystats** steps into a single command. Since this paper is a static document, Figure 2 provides a static screenshot of the interactive report.

This seamless integration between **easystats** packages allows users to move from model fitting and interpretation to visualization and reporting in a fluid and intuitive way, without having to learn different syntaxes, data structures, or idiosyncratic software design choices.

To get the newest features from the rapidly developing **easystats** ecosystem, users can run `easystats::install_latest()`. It conveniently installs the latest development version of every package in the project.

Licensing and Availability

The release version of `{easystats}` is available for download from [CRAN](https://cran.r-project.org/web/packages/easystats/index.html), whereas development versions are available from the [R-universe](https://github.com/easystats/easystats) and [GitHub](https://github.com/easystats/easystats). The package is licensed under the MIT-License, with all source code stored at GitHub (<https://github.com/easystats/easystats>), with a corresponding issue tracker for bug reporting and feature enhancements. In the spirit of honest and open science, we encourage requests, tips for fixes, feature updates, as well as general questions and concerns via direct interaction with contributors and developers.

Acknowledgments

`{easystats}` is part of the collaborative **easystats** ecosystem. Thus, we thank all [members of easystats](#), contributors, and users alike.

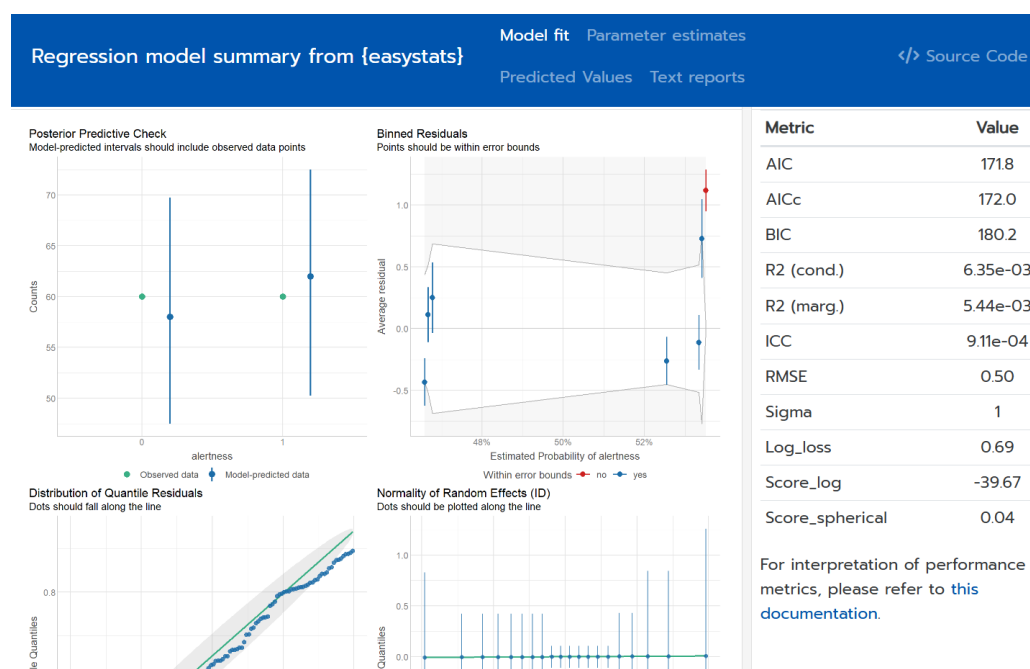


Figure 2: Screenshot of the “Model fit” tab from the interactive dashboard

References

- Arel-Bundock, V., Greifer, N., & Heiss, A. (2024). How to interpret statistical models using `marginalEffects` for R and Python. *Journal of Statistical Software*, 111(9). <https://doi.org/10.18637/jss.v111.i09>
- Bates, D., Mächler, M., Bolker, B., & Walker, S. (2015). Fitting linear mixed-effects models using `lme4`. *Journal of Statistical Software*, 67(1), 1–48. <https://doi.org/10.18637/jss.v067.i01>
- Ben-Shachar, M. S., Lüdtke, D., & Makowski, D. (2020). `effectsize`: Estimation of effect size indices and standardized parameters. *Journal of Open Source Software*, 5(56), 2815. <https://doi.org/10.21105/joss.02815>
- Ben-Shachar, M. S., Patil, I., Thériault, R., Wiernik, B. M., & Lüdtke, D. (2023). Phi, fei, fo, fum: Effect sizes for categorical data that use the chi-squared statistic. *Mathematics*, 11(9), 1982. <https://doi.org/10.3390/math11091982>
- Bürkner, P.-C. (2017). `Brms`: An R package for Bayesian multilevel models using `stan`. *Journal of Statistical Software*, 80, 1–28.
- Lüdtke, D., Ben-Shachar, M. S., Patil, I., & Makowski, D. (2020). `parameters`: Extracting, computing and exploring the parameters of statistical models using R. *Journal of Open Source Software*, 5(53), 2445. <https://doi.org/10.21105/joss.02445>
- 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., Patil, I., Ben-Shachar, M., Wiernik, B., 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., Chen, S. A., & Lüdtke, D. (2019). Indices of effect existence and significance in the Bayesian framework. *Frontiers in Psychology*, 10, 2767. <https://doi.org/10.3389/fpsyg.2019.02767>
- Makowski, D., Ben-Shachar, M. S., & Lüdtke, D. (2019). `bayestestR`: Describing effects and their uncertainty, existence and significance within the Bayesian framework.

- Journal of Open Source Software*, 4(40), 1541. <https://doi.org/10.21105/joss.01541>
- Makowski, D., Ben-Shachar, M. S., Wiernik, B. M., Patil, I., Thériault, R., & Lüdecke, D. (2025). modelbased: An R package to make the most out of your statistical models through marginal means, marginal effects, and model predictions. *Journal of Open Source Software*, 10(109), 7969. <https://doi.org/10.21105/joss.07969>
- Makowski, D., Lüdecke, D., Patil, I., Thériault, R., Ben-Shachar, M. S., & Wiernik, B. M. (2023). report: Automated results reporting as a practical tool to improve reproducibility and methodological best practices adoption. *CRAN*. <https://doi.org/10.32614/CRAN.package.report>
- Patil, I., Makowski, D., Ben-Shachar, M. S., Wiernik, B. M., Bacher, E., & Lüdecke, D. (2022). datawizard: An R package for easy data preparation and statistical transformations. *Journal of Open Source Software*, 7(78), 4684. <https://doi.org/10.21105/joss.04684>
- Sjoberg, D. D., Whiting, K., Curry, M., Lavery, J. A., & Larmarange, J. (2021). Reproducible Summary Tables with the gtsummary Package. *The R Journal*, 13(1), 570–580. <https://doi.org/10.32614/RJ-2021-053>
- Thériault, R., Ben-Shachar, M. S., Patil, I., Lüdecke, D., Wiernik, B. M., & Makowski, D. (2024). Check your outliers! An introduction to identifying statistical outliers in R with easystats. *Behavior Research Methods*, 56(4), 4162–4172. <https://doi.org/10.3758/s13428-024-02356-w>
- Wickham, H., Averick, M., Bryan, J., Chang, W., McGowan, L. D., François, R., Grolemond, G., Hayes, A., Henry, L., Hester, J., Kuhn, M., Pedersen, T. L., Miller, E., Bache, S. M., Müller, K., Ooms, J., Robinson, D., Seidel, D. P., Spinu, V., ... Yutani, H. (2019). Welcome to the tidyverse. *Journal of Open Source Software*, 4(43), 1686. <https://doi.org/10.21105/joss.01686>