

effectsize: Estimation of Effect Size Indices and Standardized Parameters

Mattan S. Ben-Shachar¹, Daniel Lüdtke², and Dominique Makowski³

¹ Ben-Gurion University of the Negev, Israel ² University Medical Center Hamburg-Eppendorf, Germany ³ Nanyang Technological University, Singapore

DOI: [10.21105/joss.02815](https://doi.org/10.21105/joss.02815)

Software

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

Editor: Mikkel Meyer Andersen
↗

Reviewers:

- [@trashbirdecology](#)
- [@jkarreth](#)

Submitted: 28 October 2020
Published: 23 December 2020

License

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

Aims of the Package

In both theoretical and applied research, it is often of interest to assess the strength of an observed association. This is typically done to allow the judgment of the magnitude of an effect (especially when units of measurement are not meaningful, e.g. in the use of estimated latent variables; [Bollen, 1989](#)), to facilitate comparing between predictors' importance within a given model, or both. Though some indices of effect size, such as the correlation coefficient (itself a standardized covariance coefficient) are readily available, other measures are often harder to obtain. **effectsize** is an R package ([R Core Team, 2020](#)) that fills this important gap, providing utilities for easily estimating a wide variety of standardized effect sizes (i.e., effect sizes that are not tied to the units of measurement of the variables of interest) and their confidence intervals (CIs), from a variety of statistical models. **effectsize** provides easy-to-use functions, with full documentation and explanation of the various effect sizes offered, and is also used by developers of other R packages as the back-end for effect size computation, such as **parameters** ([Lüdtke et al., 2020](#)), **ggstatsplot** ([Patil, 2018](#)), **gtsummary** ([Sjoberg et al., 2020](#)) and more.

Comparison to Other Packages

effectsize's functionality is in part comparable to packages like **lm.beta** ([Behrendt, 2014](#)), **MOTE** ([Buchanan et al., 2019](#)) or **MBESS** ([K. Kelley, 2020](#)). Yet, there are some notable differences, e.g.:

- **lm.beta** provides standardized regression coefficients for linear models, based on post-hoc model matrix standardization. However, the functionality is available only for a limited number of models (models inheriting from the `lm` class), whereas **effectsize** provides support for many types of models, including (generalized) linear mixed models, Bayesian models, and more. Additionally, in addition to post-hoc model matrix standardization, **effectsize** offers other methods of standardization (see below).
- Both **MOTE** and **MBESS** provide functions for computing effect sizes such as Cohen's *d* and effect sizes for ANOVAs ([Cohen, 1988](#)), and their confidence intervals. However, both require manual input of *F*- or *t*-statistics, *degrees of freedom*, and *sums of squares* for the computation the effect sizes, whereas **effectsize** can automatically extract this information from the provided models, thus allowing for better ease-of-use as well as reducing any potential for error.

- Finally, in **base R**, the function `scale()` can be used to standardize vectors, matrices and data frame, which can be used to standardize data prior to model fitting. The coefficients of a linear model fit on such data are in effect standardized regression coefficients. **effectsize** expands on this, allowing for robust standardization (using the median and the MAD, instead of the mean and SD), post-hoc parameter standardization, and more.

Examples of Features

effectsize provides various functions for extracting and estimating effect sizes and their confidence intervals (estimated using the noncentrality parameter method; [Steiger, 2004](#)). In this article, we provide basic usage examples for estimating some of the most common effect size. A comprehensive overview, including in-depth examples and [a full list of features and functions](#), are accessible via the dedicated website (<https://easystats.github.io/effectsize/>).

Indices of Effect Size

Standardized Differences

effectsize provides functions for estimating the common indices of standardized differences such as Cohen's d (`cohens_d()`), Hedge's g (`hedges_g()`) for both paired and independent samples ([Cohen, 1988](#); [Hedges & Olkin, 1985](#)), and Glass' Δ (`glass_delta()`) for independent samples with different variances ([Hedges & Olkin, 1985](#)).

```
cohens_d(mpg ~ am, data = mtcars)
#> Cohen's d |          95% CI
#> -----
#>    -1.48 | [-2.27, -0.67]
#>
#> - Estimated using pooled SD.
```

Contingency Tables

Pearson's ϕ (`phi()`) and Cramér's V (`cramers_v()`) can be used to estimate the strength of association between two categorical variables ([Cramér, 1946](#)), while Cohen's g (`cohens_g()`) estimates the deviance between paired categorical variables ([Cohen, 1988](#)).

```
M <- rbind(c(150, 130, 35, 55),
           c(100, 50, 10, 40),
           c(165, 65, 2, 25))

cramers_v(M)
#> Cramer's V |          95% CI
#> -----
#>    0.18 | [0.12, 0.22]
```

Parameter and Model Standardization

Standardizing parameters (i.e., coefficients) can allow for their comparison within and between models, variables and studies. To this end, two functions are available: `standardize()` which

returns an updated model, re-fit with standardized data, and `standardize_parameters()` which returns a table of standardized coefficients from a provided model (for a list of supported models, see the *insight* package; Lüdtke et al., 2019).

```
model <- lm(mpg ~ cyl * am,
            data = mtcars)

standardize(model)
#>
#> Call:
#> lm(formula = mpg ~ cyl * am, data = data_std)
#>
#> Coefficients:
#> (Intercept)          cyl          am          cyl:am
#>   -0.09777      -0.7426       0.1739      -0.1930

standardize_parameters(model)
#> Parameter | Coefficient (std.) |          95% CI
#> -----|-----|-----
#> (Intercept) |          -0.10 | [-0.30,  0.11]
#> cyl          |          -0.74 | [-0.95, -0.53]
#> am           |           0.17 | [-0.04,  0.39]
#> cyl:am       |          -0.19 | [-0.41,  0.02]
#>
#> # Standardization method: refit
```

Standardized parameters can also be produced for generalized linear models (GLMs; where only the predictors are standardized):

```
model <- glm(am ~ cyl + hp,
             family = "binomial",
             data = mtcars)

standardize_parameters(model, exponentiate = TRUE)
#> Parameter | Odds Ratio (std.) |          95% CI
#> -----|-----|-----
#> (Intercept) |           0.53 | [0.18,  1.32]
#> cyl          |           0.05 | [0.00,  0.29]
#> hp           |           6.70 | [1.32, 61.54]
#>
#> # Standardization method: refit
```

`standardize_parameters()` provides several standardization methods, such as robust standardization, or *pseudo*-standardized coefficients for (generalized) linear mixed models (Hoffman, 2015). A full review of these methods can be found in the [Parameter and Model Standardization vignette](#).

Effect Sizes for ANOVAs

Unlike standardized parameters, the effect sizes reported in the context of ANOVAs (analysis of variance) or ANOVA-like tables represent the amount of variance explained by each of the model's terms, where each term can be represented by one or more parameters. `eta_squared()` can produce such popular effect sizes as Eta-squared (η^2), its partial version (η_p^2), as well as the generalized η_G^2 (Cohen, 1988; Olejnik & Algina, 2003):

```
options(contrasts = c('contr.sum', 'contr.poly'))

data("ChickWeight")
# keep only complete cases and convert `Time` to a factor
ChickWeight <- subset(ChickWeight, ave(weight, Chick, FUN = length) == 12)
ChickWeight$Time <- factor(ChickWeight$Time)

model <- aov(weight ~ Diet * Time + Error(Chick / Time),
             data = ChickWeight)

eta_squared(model, partial = TRUE)
#> Group      / Parameter / Eta2 (partial) /      90% CI
#> -----
#> Chick      /      Diet /           0.27 | [0.06, 0.42]
#> Chick:Time /      Time /           0.87 | [0.85, 0.88]
#> Chick:Time / Diet:Time /           0.22 | [0.11, 0.23]

eta_squared(model, generalized = "Time")
#> Group      / Parameter / Eta2 (generalized) /      90% CI
#> -----
#> Chick      /      Diet /           0.04 | [0.00, 0.09]
#> Chick:Time /      Time /           0.74 | [0.71, 0.77]
#> Chick:Time / Diet:Time /           0.03 | [0.00, 0.00]
```

effectsize also offers ϵ_p^2 (`epsilon_squared()`) and ω_p^2 (`omega_squared()`), which are less biased estimates of the variance explained in the population (T. L. Kelley, 1935; Olejnik & Algina, 2003). For more details about the various effect size measures and their applications, see the [Effect sizes for ANOVAs vignette](#).

Effect Size Conversion

From Test Statistics

In many real world applications there are no straightforward ways of obtaining standardized effect sizes. However, it is possible to get approximations of most of the effect size indices (d , r , η_p^2 ...) with the use of test statistics (Friedman, 1982). These conversions are based on the idea that test statistics are a function of effect size and sample size (or more often of degrees of freedom). Thus it is possible to reverse-engineer indices of effect size from test statistics (F , t , χ^2 , and z).

```
F_to_eta2(f = c(40.72, 33.77),
          df = c(2, 1), df_error = c(18, 9))
#> Eta2 (partial) /      90% CI
#> -----
#>           0.82 | [0.66, 0.89]
#>           0.79 | [0.49, 0.89]

t_to_d(t = -5.14, df_error = 22)
#>      d /      95% CI
#> -----
#> -2.19 | [-3.23, -1.12]

t_to_r(t = -5.14, df_error = 22)
```

```
#>      r |          95% CI
#> -----
#> -0.74 | [-0.85, -0.49]
```

These functions also power the `effectsize()` convenience function for estimating effect sizes from R's `hstest`-type objects. For example:

```
aov1 <- oneway.test(salary ~ n_comps,
                    data = hardlyworking, var.equal = TRUE)
effectsize(aov1)
#> Eta2 |          90% CI
#> -----
#> 0.20 | [0.14, 0.24]

xtab <- rbind(c(762, 327, 468), c(484, 239, 477), c(484, 239, 477))
Xsq <- chisq.test(xtab)
effectsize(Xsq)
#> Cramer's V |          95% CI
#> -----
#>      0.07 | [0.05, 0.09]
```

These functions also power our *Effect Sizes From Test Statistics* shiny app (<https://easystats4u.shinyapps.io/statistic2effectsize/>).

Between Effect Sizes

For comparisons between different types of designs and analyses, it is useful to be able to convert between different types of effect sizes (d , r , Odds ratios and Risk ratios; Borenstein et al., 2009; Grant, 2014).

```
r_to_d(0.7)
#> [1] 1.960392

d_to_oddsratio(1.96)
#> [1] 34.98946

oddsratio_to_riskratio(34.99, p0 = 0.4)
#> [1] 2.397232

oddsratio_to_r(34.99)
#> [1] 0.6999301
```

Effect Size Interpretation

Finally, `effectsize` provides convenience functions to apply existing or custom interpretation rules of thumb, such as for instance Cohen's (1988). Although we strongly advocate for the cautious and parsimonious use of such judgment-replacing tools, we provide these functions to allow users and developers to explore and hopefully gain a deeper understanding of the relationship between data values and their interpretation. More information is available in the *Automated Interpretation of Indices of Effect Size* vignette.

```
interpret_d(c(0.02, 0.52, 0.86), rules = "cohen1988")
#> [1] "very small" "medium"      "large"
#> (Rules: cohen1988)
```

Licensing and Availability

effectsize is licensed under the GNU General Public License (v3.0), with all source code stored at GitHub (<https://github.com/easystats/effectsize>), and 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, by [filing an issue](#). See the package's [Contribution Guidelines](#).

Acknowledgments

effectsize is part of the [easystats](#) ecosystem, a collaborative project created to facilitate the usage of R for statistical analyses. Thus, we would like to thank the [members of easystats](#) as well as the users.

References

- Behrendt, S. (2014). *Lm.beta: Add standardized regression coefficients to lm-objects*. <https://CRAN.R-project.org/package=Lm.beta>
- Bollen, K. A. (1989). *Structural equations with latent variables*. John Wiley & Sons. <https://doi.org/10.1002/9781118619179>
- Borenstein, M., Hedges, L. V., Higgins, J. P., & Rothstein, H. R. (2009). Converting among effect sizes. In *Introduction to meta-analysis* (pp. 45–49). John Wiley & Sons. <https://doi.org/10.1002/9780470743386.ch7>
- Buchanan, E. M., Gillenwaters, A., Scofield, J. E., & Valentine, K. D. (2019). *MOTE: Measure of the Effect: Package to assist in effect size calculations and their confidence intervals*. <http://github.com/doomlab/MOTE>
- Cohen, J. (1988). *Statistical power analysis for the behavioral sciences*, 2nd ed. Routledge. ISBN: 9780805802832
- Cramér, H. (1946). *Mathematical methods of statistics*. Princeton University Press. ISBN: 9781400883868
- Friedman, H. (1982). Simplified determinations of statistical power, magnitude of effect and research sample sizes. *Educational and Psychological Measurement*, 42(2), 521–526. <https://doi.org/10.1177/001316448204200214>
- Grant, R. L. (2014). Converting an odds ratio to a range of plausible relative risks for better communication of research findings. *British Medical Journal*, 348, f7450. <https://doi.org/10.1136/bmj.f7450>
- Hedges, L., & Olkin, I. (1985). *Statistical methods for meta-analysis*. Academic Press. ISBN: 9780080570655
- Hoffman, L. (2015). *Longitudinal analysis: Modeling within-person fluctuation and change*. Routledge. ISBN: 9780415876025
- Kelley, K. (2020). *MBESS: The MBESS r package*. <https://CRAN.R-project.org/package=MBESS>
- Kelley, T. L. (1935). An unbiased correlation ratio measure. *Proceedings of the National Academy of Sciences of the United States of America*, 21(9), 554–559. <https://doi.org/10.1073/pnas.21.9.554>

- Lüdtke, D., Ben-Shachar, M. S., Patil, I., & Makowski, D. (2020). 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., Waggoner, P., & Makowski, D. (2019). Insight: A unified interface to access information from model objects in r. *Journal of Open Source Software*, 4(38), 1412. <https://doi.org/10.21105/joss.01412>
- Olejnik, S., & Algina, J. (2003). Generalized eta and omega squared statistics: Measures of effect size for some common research designs. *Psychological Methods*, 8(4), 434–447. <https://doi.org/10.1037/1082-989X.8.4.434>
- Patil, I. (2018). ggstatsplot: 'ggplot2' based plots with statistical details. CRAN. <https://doi.org/10.5281/zenodo.2074621>
- R Core Team. (2020). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing. <https://www.R-project.org/>
- Sjoberg, D. D., Curry, M., Hannum, M., Whiting, K., & Zabor, E. C. (2020). *Gtsummary: Presentation-ready data summary and analytic result tables*. <https://CRAN.R-project.org/package=gtsummary>
- Steiger, J. H. (2004). Beyond the f test: Effect size confidence intervals and tests of close fit in the analysis of variance and contrast analysis. *Psychological Methods*, 9(2), 164–182. <https://doi.org/10.1037/1082-989X.9.2.164>