The Journal of Open Source Software

# see: An R Package for Visualizing Statistical Models

**Daniel Lüdecke**[1], **Mattan S. Ben-Shachar**[2], **Indrajeet Patil**[3], **Brenton M. Wiernik**[4], **Philip Waggoner**[5], **and Dominique Makowski**[6]

**1** University Medical Center Hamburg-Eppendorf, Germany **2** Ben-Gurion University of the Negev, Israel **3** Center for Humans and Machines, Max Planck Institute for Human Development, Berlin, Germany **4** Department of Psychology, University of South Florida, USA **5** University of Chicago, USA **6** Nanyang Technological University, Singapore

**DOI:**

**Software**

- Review
- Repository
- Archive

**Submitted:**
**Published:**

## Summary

The *easystats* ecosystem refers to a collection of packages that operate in synergy to provide a consistent and intuitive syntax when working with statistical models within the R programming language (R Core Team, 2021). Though most of the packages return comprehensive numeric summaries of model parameters and performance, the ability to visualize model output leads to more informative, communicable and well-rounded scientific reporting. As a core pillar of *easystats*, the *see* package offers a host of functions and tools for a range of publication-ready visualizations.

## Statement of Need

The grammar of graphics (Wilkinson, 2012), largely due to its implementation in the *ggplot2* (Wickham, 2016) package, has become the dominant approach to visualization in R. As a result, a number of packages for visually exploring statistical models have been built on top of *ggplot2*.[1] A hallmark of many of these packages is the provision of ready-made plots or geometric layers for simply preparing common visualizations, but without linking them to any particular statistical analyses (e.g., *ggrepel*, *ggalluvial*, *ggridges*, *ggdist*, etc.). Even still, a few packages, though typically more specialized, are interested in linking visualization and statistical analysis (e.g., *ggpubr*, *tidymv*, *survminer*). For example, the *ggstatsplot* package (Patil, 2021) offers visualizations for statistical analysis of one-way factorial designs, and the *plotmm* package (Waggoner, 2020) supports specific types of mixture model objects.

The *see* package, on the other hand, is designed to work with objects created by the other *easystats* packages, such as *parameters* tables, *performance* tables, *correlation* matrices, and so on. While these and other *easystats* packages support a wide range of statistical models, the *see* package acts as a visual support to the entire *easystats* ecosystem. As such, visualizations corresponding to all stages of statistical analysis, from model fitting to reporting, can be easily created using *see*. Additionally, *see* contains many aesthetic utilities to embellish non-*easystats* plots. The result is a package that minimizes the barrier to producing high-quality statistical visualizations in R.

The central goal of *easystats* is to make the task of doing statistics in R as easy as possible. This goal is operationalized through intuitive and consistent syntax, consistent and transparent argument names, comprehensive documentation, informative warnings

[1] For a sampling of these packages, visit https://exts.ggplot2.tidyverse.org/gallery/

and error messages, and smart functions with sensible default parameter values. The *see* package follows this philosophy by *not* introducing any new function for most of its features. As a result, *see* typically relies on a generic `plot()` method to which an *easystats* object is passed. And as demonstrated in this paper, plots are fully customizable by adding *ggplot2* geometric layers in a traditional way (e.g., `ggtitle()` for a plot title).

## Features

Though we introduce here only one plotting method for each *easystats* package, many other methods are available. Interested readers are encouraged to explore the range of examples on the package website, https://easystats.github.io/see/.

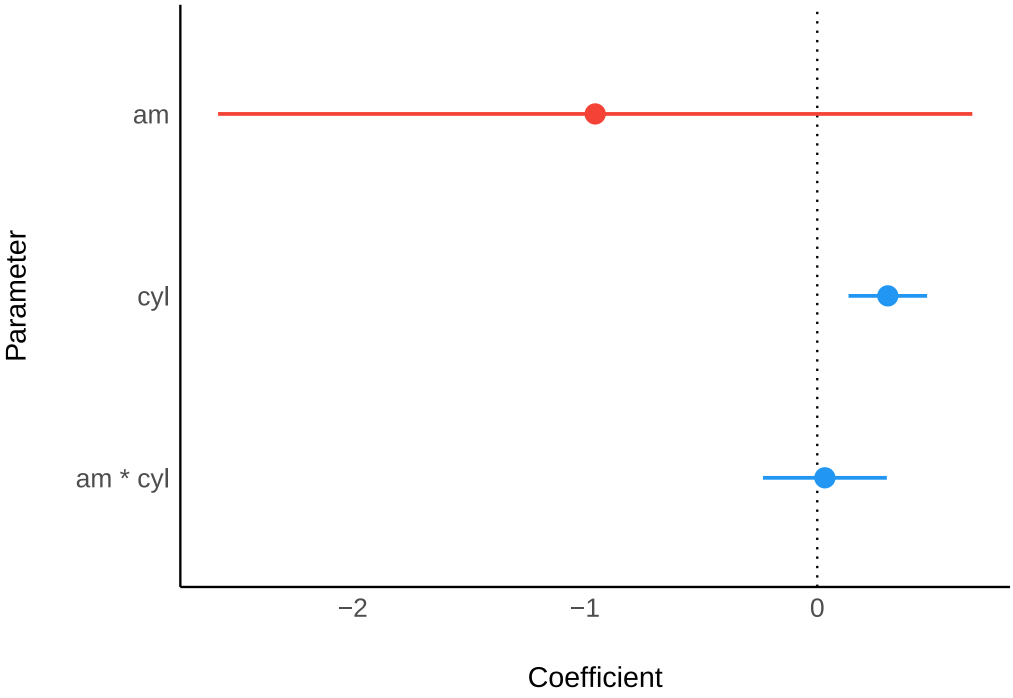
### Visualizing Model Parameters

The *parameters* package converts summaries of regression model objects into dataframes (Lüdecke et al., 2020). The *see* package can take this transformed object and, for example, create a dot-and-whisker plot for the extracted regression estimates simply by passing the `parameters` class object to `plot()`.

```
library(parameters)
library(see)

model <- lm(wt ~ am * cyl, mtcars)

plot(parameters(model))
```

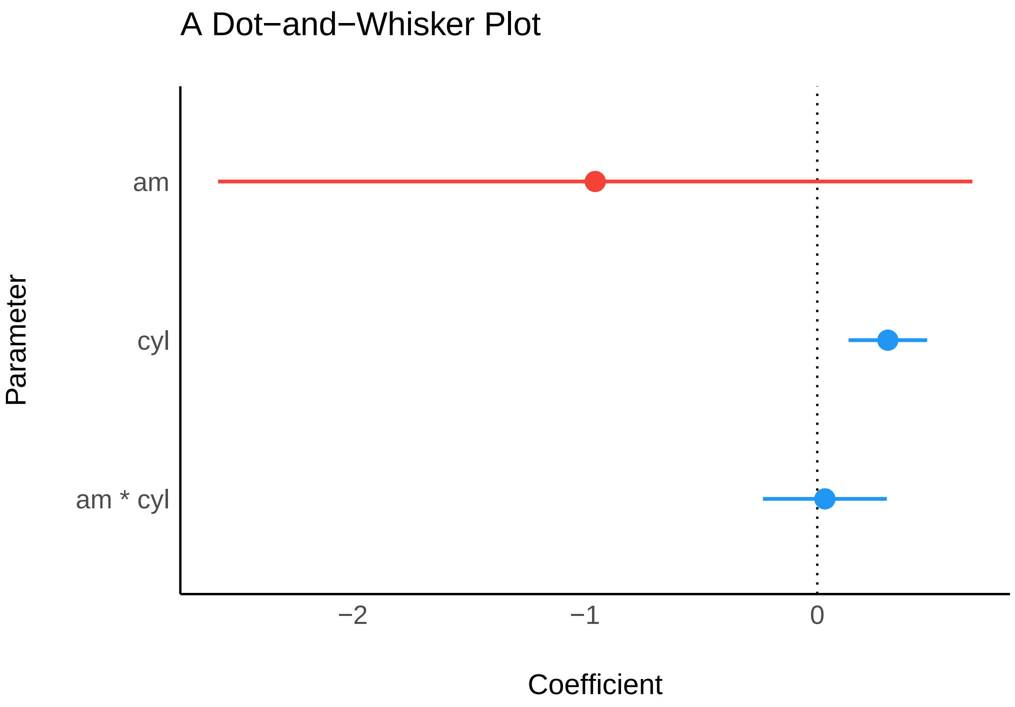


As *see* outputs objects of class `ggplot`, *ggplot2* functions can be added as layers to the plot as with all other *ggplot2* visualizations. For example, we might add a title using `labs()` from *ggplot2*.

```r
library(parameters)
library(see)

model <- lm(wt ~ am * cyl, mtcars)

plot(parameters(model)) +
  ggplot2::labs(title = "A Dot-and-Whisker Plot")
```

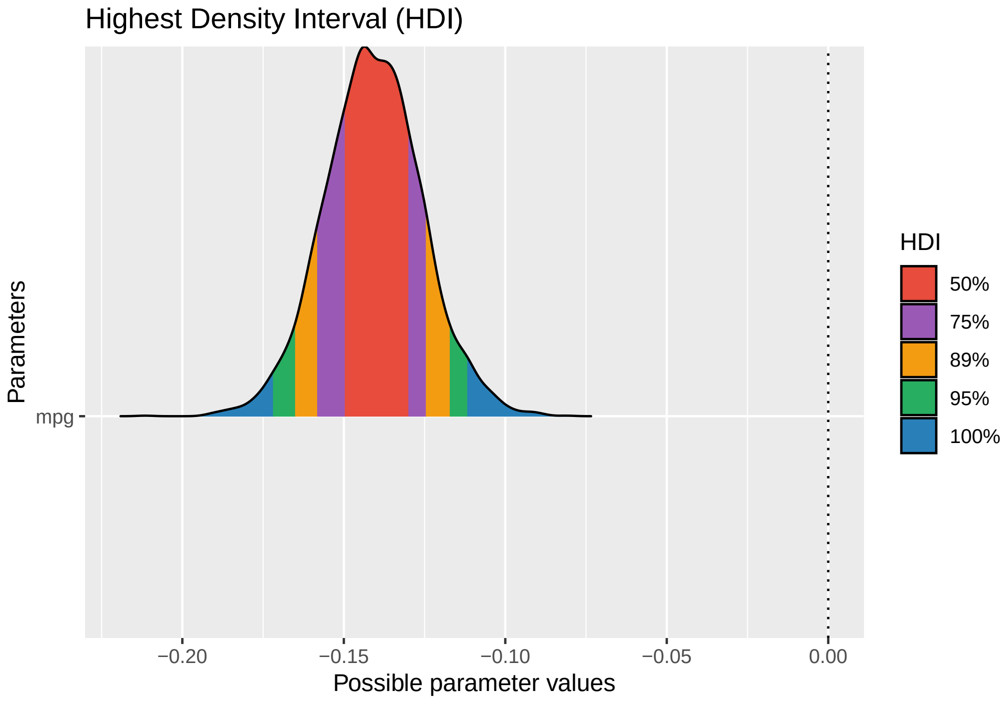


Similarly, for Bayesian regression model objects, which are handled by the *bayestestR* package (Makowski et al., 2019), the *see* package can also provide special plotting methods relevant only for Bayesian models (e.g., Highest Density Interval, or *HDI*). Similarly, users can fit the model and pass the model results, extracted via *bayestestR*, to `plot()`.

```r
library(bayestestR)
library(rstanarm)
library(see)

model <- stan_glm(wt ~ mpg, data = mtcars, refresh = 0)
result <- hdi(model, ci = c(0.5, 0.75, 0.89, 0.95))

plot(result)
```



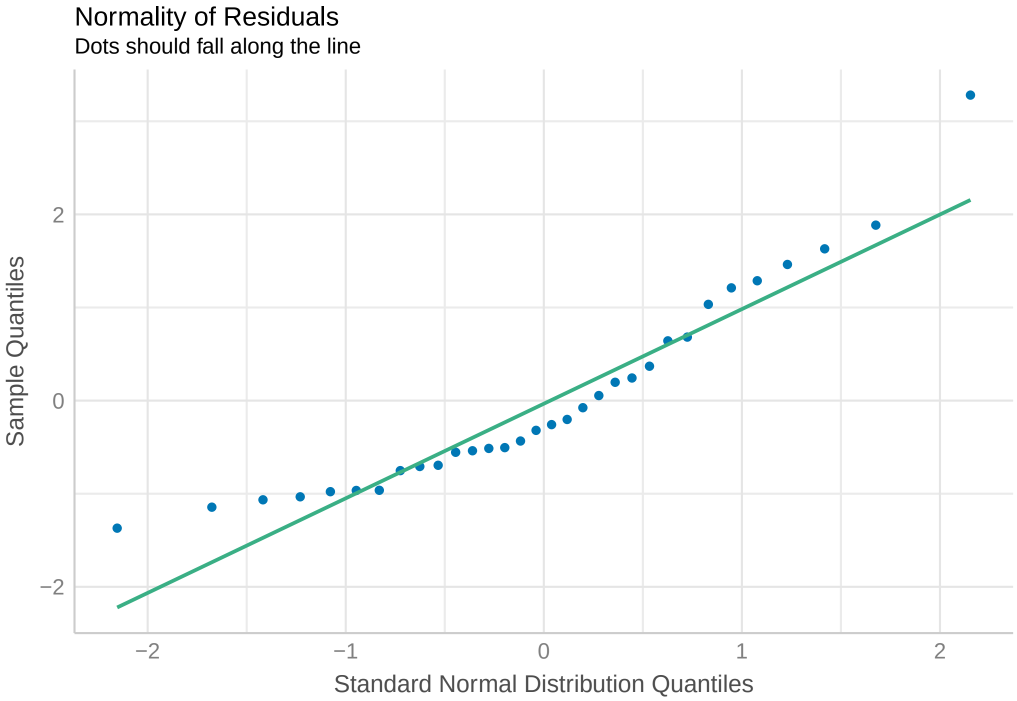
## Visualizing Model Performance

The *performance* package is primarily concerned with checking regression model assumptions (Lüdecke et al., 2021). The *see* package offers a number of tools to visualize these assumption checks, such as the normality of residuals. Similar to uses of *see* with *parameters* introduced in the previous section, users simply pass the fit model object to the relevant *performance* function (`check_normality()` in the example below). Then, this result can be passed to `plot()` to render a *ggplot2* visualization of the check on the normality of the residuals.

```r
library(performance)
library(see)

model <- lm(wt ~ mpg, mtcars)
check <- check_normality(model)
#> Warning: Non-normality of residuals detected (p = 0.016).

plot(check, type = "qq")
```



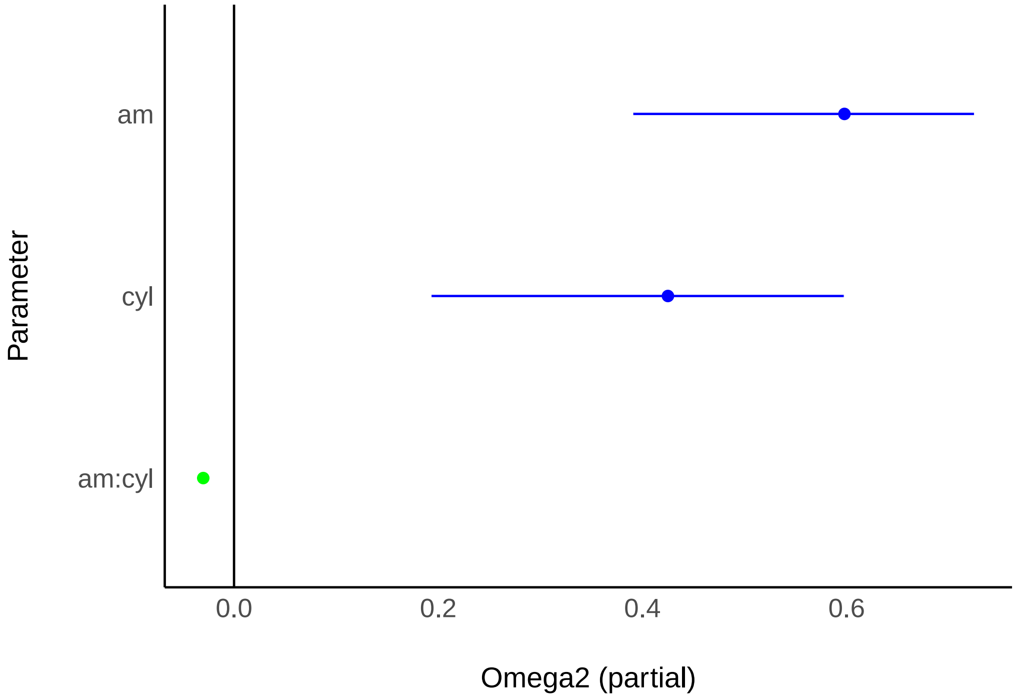
## Visualizing Effect Sizes

In addition to providing tabular summaries of regression model objects, the *easystats* ecosystem also provides several packages to assess various aspects of statistical models. Of note, the *effectsize* package assesses the practical importance of observed effects by computing appropriate effect size measures (Ben-Shachar et al., 2020). In conjunction with *see*, users are able to visualize the magnitude and uncertainty of effect sizes by passing the model object to the relevant `effectsize` function (`omega_squared()` in the following example), and then to `plot()` as before.

```r
library(effectsize)
library(see)

model <- aov(wt ~ am * cyl, mtcars)

plot(omega_squared(model))
```



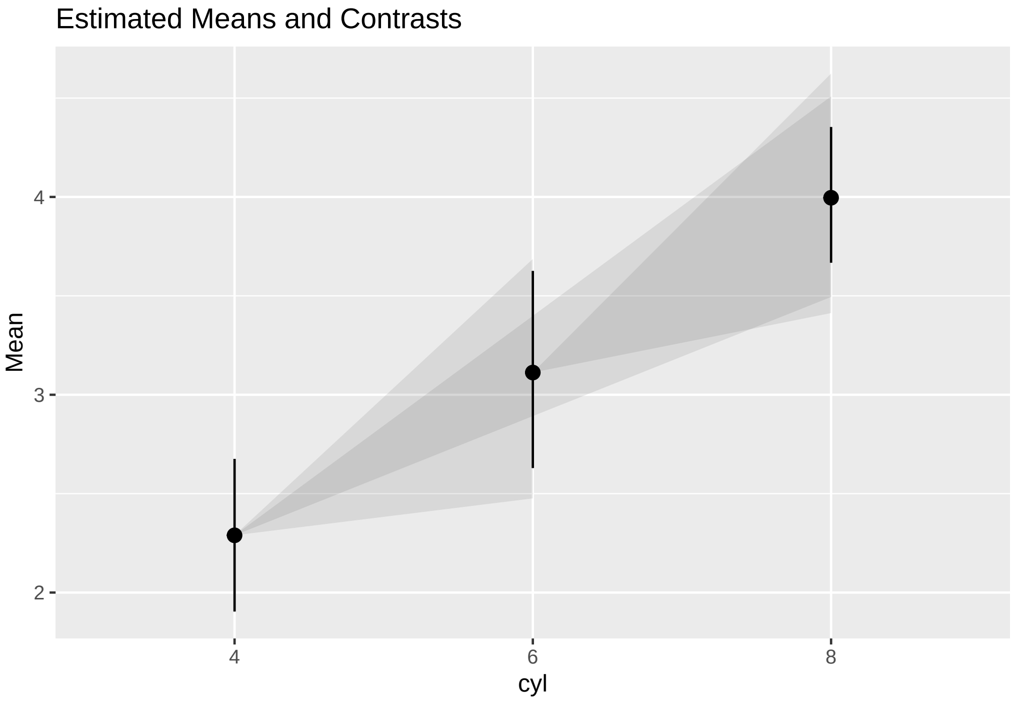
## Visualizing Marginal Effects

The *modelbased* package computes a range of quantities from fit regression models (Makowski et al., 2020a). With minimal and human-readable code, users start by passing the model object to the relevant *modelbased* function(s) (`estimate_contrasts()` and `estimate_means()` in the example to follow). Then, these new objects containing the *modelbased* quantities are passed to `plot()` from the *see* package. Of note, in the following example *two* quantities (objects `contrasts` and `means`) are passed to `plot()`, demonstrating and deepening the flexibility of the package.

```r
library(modelbased)
library(rstanarm)
library(see)

model <- stan_glm(wt ~ as.factor(cyl), data = mtcars, refresh = 0)
contrasts <- estimate_contrasts(model)
means <- estimate_means(model)

plot(contrasts, means)
```



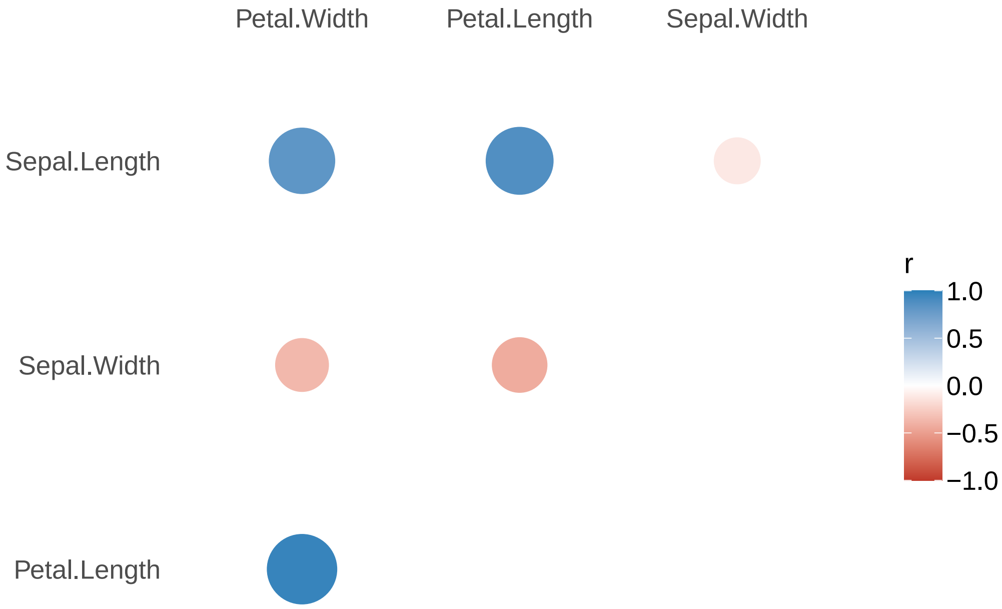
## Visualizing Correlation Matrices

The *correlation* package provides a unified syntax and human-readable code to carry out many flavors of correlation analysis (Makowski et al., 2020b). Results from the `correlation()` function call of class `easycorrelation` are passed to `plot()` from *see* to render these correlations in a matrix.

```
library(correlation)
library(see)

results <- correlation(iris, type = "percentage")

plot(summary(results))
```



## Licensing and Availability

*see* is licensed under the GNU General Public License (v3.0), with all source code openly developed and stored at GitHub (https://github.com/easystats/see), along 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

*see* is part of the collaborative *easystats* ecosystem. Thus, we thank the members of easystats as well as the users.

## References

Ben-Shachar, M. S., Lüdecke, 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

Lüdecke, 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

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

Makowski, D., Ben-Shachar, M. S., & Lüdecke, 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., Patil, I., & Lüdecke, D. (2020a). Estimation of model-based predictions, contrasts and means. *CRAN*. https://github.com/easystats/modelbased

Makowski, D., Ben-Shachar, M. S., Patil, I., & Lüdecke, D. (2020b). Methods and algorithms for correlation analysis in R. *Journal of Open Source Software*, *5*(51), 2306. https://doi.org/10.21105/joss.02306

Patil, I. (2021). Visualizations with statistical details: The 'ggstatsplot' approach. *Journal of Open Source Software*, *6*(61), 3167. https://doi.org/10.21105/joss.03167

R Core Team. (2021). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing. https://www.R-project.org/

Waggoner, P. D. (2020). *plotmm: Tidy tools for visualizing mixture models*. https://CRAN.R-project.org/package=plotmm

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

Wilkinson, L. (2012). The Grammar of Graphics. In *Handbook of computational statistics* (pp. 375–414). Springer.