Package 'easystats'

May 24, 2026

Type Package

Title Framework for Easy Statistical Modeling, Visualization, and Reporting

Version 0.7.6

Maintainer Daniel Lüdecke <d.luedecke@uke.de>

Description A meta-package that installs and loads a set of packages from 'easystats' ecosystem in a single step. This collection of packages provide a unifying and consistent framework for statistical modeling, visualization, and reporting. Additionally, it provides articles targeted at instructors for teaching 'easystats', and a dashboard targeted at new R users for easily conducting statistical analysis by accessing summary results, model fit indices, and visualizations with minimal programming.

License MIT + file LICENSE

URL https://easystats.github.io/easystats/

BugReports https://github.com/easystats/easystats/issues

Depends R (>= 3.6)

Imports bayestestR (>= 0.16.1), correlation (>= 0.8.8), datawizard (>= 1.1.0), effectsize (>= 1.0.1), insight (>= 1.3.1), modelbased (>= 0.12.0), parameters (>= 0.27.0), performance (>= 0.15.0), report (>= 0.6.1), see (>= 0.11.0), tools, utils

Suggests callr, collapse, cranlogs, DHARMa, DT, flexdashboard (>= 0.6.2), Formula, ggplot2, glmmTMB, httr, jsonlite, knitr, marginaleffects (>= 0.25.0), mockery, pak, patchwork, scholar, rmarkdown, testthat (>= 3.2.1), withr, xml2

VignetteBuilder knitr

Encoding UTF-8

Language en-US

RoxygenNote 7.3.3

Config/testthat/edition 3

Config/Needs/website easystats/easystatstemplate

NeedsCompilation no

Author Daniel Lüdecke [aut, cre] (ORCID:
<https://orcid.org/0000-0002-8895-3206>),
Dominique Makowski [aut] (ORCID:
<https://orcid.org/0000-0001-5375-9967>),
Mattan S. Ben-Shachar [aut] (ORCID:
<https://orcid.org/0000-0002-4287-4801>),
Indrajeet Patil [aut] (ORCID: <https://orcid.org/0000-0003-1995-6531>),
Brenton M. Wiernik [aut] (ORCID:
<https://orcid.org/0000-0001-9560-6336>),
Etienne Bacher [aut] (ORCID: <https://orcid.org/0000-0002-9271-5075>),
Rémi Thériault [aut] (ORCID: <https://orcid.org/0000-0003-4315-6788>)

Repository CRAN

Date/Publication 2026-05-24 19:10:02 UTC

Contents

`easystats_citations` *Reports citations for easystats publications*

Description

This function reports the total number of Google Scholar citations for `easystats` publications through the `scholar` package.

Usage

```
easystats_citations(sort_by = "year", length = 30)
```

Arguments

`sort_by`	Name of the column that should be used for sorting. Can be `"title"`, `"journal"`, `"year"`, or `"cites"`. Use `NULL` for no sorting and to return data as retrieved by Google scholar.

length — Numeric, maximum length of the returned string. If not NULL, will shorten the string to a maximum length, however, it will not truncate inside words. I.e. if the string length happens to be inside a word, this word is removed from the returned string, so the returned string has a *maximum* length of length, but might be shorter.

Value

A data frame of four columns: title, journal, year, and cites.

Examples

```
## Not run:
easystats_citations()

## End(Not run)
```

---

easystats_downloads *Reports download statistics for easystats packages*

Description

This function reports download statistics for easystats packages from CRAN through the cranlogs package.

Usage

```
easystats_downloads(from = "2019-02-26", sort_by = "total", length = 30)
```

Arguments

from — Starting date for the download statistics. Defaults to "2019-02-26", which is when the first easystats package was released on CRAN.

sort_by — Name of the column that should be used for sorting. Can be "package", "total", or "monthly". Use NULL for no sorting. Defaults to "total" (total downloads).

length — Numeric, maximum length of the returned string. If not NULL, will shorten the string to a maximum length, however, it will not truncate inside words. I.e. if the string length happens to be inside a word, this word is removed from the returned string, so the returned string has a *maximum* length of length, but might be shorter.

Value

A data frame with three columns: package, total downloads, and average monthly downloads.

Examples

```
## Not run:
easystats_downloads()

## End(Not run)
```

`easystats_packages` *List all packages in the easystats ecosystem*

Description

List all packages in the easystats ecosystem

Usage

```
easystats_packages()
```

Value

A character vector

Examples

```
easystats_packages()
```

`easystats_update` *Update* easystats*-packages and its dependencies from CRAN, if necessary.*

Description

Update *easystats*-packages and its dependencies from CRAN, if necessary.

Usage

```
easystats_update(which = "all")
```

Arguments

`which`	String, indicates whether easystats-packages (`which = "core"`), dependencies (`which = "deps"`) or both (`which = "all"`) should be checked for available updates.

Details

If package {pak} is installed, pak::pkg_install() will be used to install packages. Else, utils::install.packages() is used.

Value

Invisible NULL.

Examples

```
# check which local easystats-packages (and their dependencies)
# are out of date and install updates from CRAN
easystats_update()

# update only easystats core-packages
easystats_update("core")
```

---

easystats_zen *Welcome to the easyverse*

Description

Welcome to the easyverse

Usage

```
easystats_zen()
```

Value

A reassuring message.

Examples

```
easystats_zen()
```

`install_latest` *Install the easystats suite from R-universe (GitHub) or CRAN*

Description

This function can be used to install all the **easystats** packages, either latest development versions (from R-universe/GitHub) or the current versions from CRAN. If the development versions are installed, packages will be installed from the stable branch (master/main) for each package.

Usage

```
install_latest(
  source = "development",
  packages = "all",
  force = FALSE,
  verbose = TRUE
)
```

Arguments

`source`	Character. Either `"development"`, `"cran"` or `"github"`. If `"cran"`, packages will be installed from the default CRAN mirror returned by `getOption("repos")['CRAN']`. If `"development"` (the default), packages are installed from the R-universe repository (https://easystats.r-universe.dev/). `"github"` installs the latest version from the GitHub-repositories main-branch. However, this option is usually not needed, because R-universe provides latest binaries. Use the `"github"` option only when R-universe servers are unavailable.
`packages`	Character vector, indicating which packages to be installed. By default, the option `"all"` will install all **easystats** packages.
`force`	Logical, if `FALSE`, only those packages with a newer version number will be installed. Use `force=TRUE` to force installation of all packages, even if the version number for the locally installed packages is identical to the latest development-versions. Only applies when `source="development"`.
`verbose`	Toggle messages.

Value

Invisible `NULL`.

Examples

```
# install latest development-version of easystats packages from
# the r-universe repository, but only those packages that have newer
# versions available
install_latest()

# install all latest development-version of easystats packages from
```

```
# the r-universe repository, no matter whether local installations
# are up to date or not.
install_latest(force = TRUE)
```

`install_suggested` *Download all suggested packages*

Description

In `easystats`, we have a 0-dependency policy, which makes our packages fairly light and fast to install. However, we rely on many many (many) packages for testing (at least all the packages for functions that we support) and some specific features. These "soft dependencies" can be downloaded at once using this function. This will allow you to fully utilize all of easystats' functionalities without errors.

Usage

```
install_suggested(package = "easystats")

show_suggested(package = "easystats")

show_reverse_dependencies(package = "easystats")
```

Arguments

`package`	If `NULL` or `"easystats"` (the default), all suggested packages for all 'easystats' packages will be installed. If specific packages are specified, only suggested packages for those packages will be installed.

Details

To reduce the dependency load, 'easystats' packages by default will not download all internally needed packages. It will ask the user to download them only if they are needed. The current function can help install all packages a given 'easystats' package might need. For example, `install_suggested("see")`. `show_suggested()` is a convenient helper to show the current list of suggested packages for each 'easystats' package.

If package {pak} is installed, `pak::pkg_install()` will be used to install packages. Else, `utils::install.packages()` is used.

Value

Useful only for its side-effect of installing the needed packages.

Examples

```
# download all suggested packages
if (FALSE) {
  install_suggested("easystats")
}

# listing all reverse dependencies of easystats packages
show_reverse_dependencies()

# listing all soft/weak dependencies of easystats packages
show_suggested()
```

---

model_dashboard *Generate a regression model summary from* easystats

---

Description

A dashboard containing the following details for the entered regression model:

- tabular summary of parameter estimates
- dot-and-whisker plot for parameter estimates
- tabular summary of indices for the quality of model fit
- collection of models for checking model assumptions
- text report
- model information table

Usage

```
model_dashboard(
  model,
  check_model_args = NULL,
  parameters_args = NULL,
  performance_args = NULL,
  output_file = "easydashboard.html",
  output_dir = getwd(),
  rmd_dir = system.file("templates/easydashboard.Rmd", package = "easystats"),
  quiet = FALSE,
  browse_html = interactive()
)
```

Arguments

model	A regression model object.

check_model_args
: A list of named arguments that are passed down to performance::check_model(). For further documentation and details about the arguments, see this website. See also 'Examples'.

parameters_args
: A list of named arguments that are passed down to parameters::model_parameters(). For further documentation and details about the arguments, see this website. See also 'Examples'.

performance_args
: A list of named arguments that are passed down to performance::model_performance(). For further documentation and details about the arguments, see this website. See also 'Examples'.

output_file
: A string specifying the file name in rmarkdown::render(). Default is "easydashboard.html".

output_dir
: A string specifying the path to the output directory for report in rmarkdown::render(). Default is to use the working directory.

rmd_dir
: A string specifying the path to the directory containing the RMarkdown template file. By default, package uses the template shipped with the package installation (inst/templates/easydashboard.Rmd).

quiet
: An option to suppress printing during rendering from knitr, pandoc command line and others. To only suppress printing of the last "Output created: " message, you can set rmarkdown.render.message to FALSE

browse_html
: A logical deciding if the rendered HTML should be opened in the browser. Defaults to interactive().

Value

An HTML dashboard.

Troubleshooting

For models with many observations, or for more complex models in general, generating the model assumptions plot might become very slow. One reason is that the underlying graphic engine becomes slow for plotting many data points. In such cases, setting the argument check_model_args = list(show_dots = FALSE) might help. Furthermore, look at other arguments of ?performance::check_model, which can be set using check_model_args, to increase performance (in particular the check-argument can help, to skip some unnecessary checks).

Examples

```
# define a regression model
mod <- lm(wt ~ mpg, mtcars)

# with default options
model_dashboard(mod)

# customizing 'parameters' output: standardize coefficients
model_dashboard(mod, parameters_args = list(standardize = "refit"))
```

```
# customizing 'performance' output: only show selected performance metrics
model_dashboard(mod, performance_args = list(metrics = c("AIC", "RMSE")))

# customizing output of model assumptions plot: don't show dots (faster plot)
model_dashboard(mod, check_model_args = list(show_dots = FALSE))
```

Index