The Journal of Open Source Software

datawizard: An R Package for Easy Data Wrangling and Statistical Transformations

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

1 esqLABS GmbH, Germany **2** Nanyang Technological University, Singapore **3** Ben-Gurion University of the Negev, Israel **4** Facebook **5** Luxembourg Institute of Socio-Economic Research, Luxembourg **6** University Medical Center Hamburg-Eppendorf, Germany

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 `{datawizard}` package in the R programming language (R Core Team, 2021) provides a lightweight toolbox to assist the following keys steps in any data analysis workflow: (*i*) to get the data in the right form, (*ii*) to modify data for statistical modeling, and (*iii*) to provide sanity checks for transformed data. Therefore, it can be a valuable tool for R users and developers looking for a lightweight option for data preprocessing.

Statement of Need

The `{datawizard}` package makes basic data wrangling easier than with base R. Its workflow and syntax are designed to be similar to `{tidyverse}` (Wickham et al. (2019)), which is a widely used ecosystem of packages for data analysis, and, therefore, users familiar with this ecosystem can easily translate their knowledge. Naturally, one might wonder why recreate data wrangling functionality already present in `{tidyverse}`.

The `{easystats}` (Ben-Shachar et al. (2020), Lüdecke et al. (2020), Lüdecke, Ben-Shachar, et al. (2021), Lüdecke, Patil, et al. (2021), Lüdecke et al. (2019), Makowski et al. (2019), Makowski et al. (2020)) is an ecosystem of packages designed to make statistical analysis easier in R. Importantly, in order to be lightweight, it follows a "0-external-hard-dependency" policy. Thus, while building this ecosystem, a new data wrangling package that relies only on base R needed to be created. In effect, this package provides the data processing backend for this entire ecosystem. In addition to its usefulness to the `{easystats}` ecosystem, it also provides *an* option for R users and package developers if they wish to keep their (recursive) dependency weight to a minimum (for other options, see Dowle & Srinivasan (2021), Eastwood (2021), etc.).

In addition to providing functions to clean messy data, `{datawizard}` also provides helpers for the other important step of data analysis: transforming the cleaned data further for setting up statistical models. For example, one may need to standardize certain variables, normalize range of some variables, adjust the data for effect of some variables, etc.

Lastly, `{datawizard}` also provides a toolbox to create a detailed profile of data properties.

Features

Data wrangling

The raw data is rarely in a state that it can be directly fed into a statistical model. It often needs to be modified in various ways. For example, columns need to be renamed and/or reordered, data scattered across multiple tables needs to be joined, certain parts of the data need to be left out, etc.

`{datawizard}` provides various functions for cleaning and preparing data (see Table 1).

Table 1: The table below lists a few key functions offered by *datawizard* for data wrangling. To see the full list, see the package website: https://easystats.github.io/datawizard/

Function	Operation
`data_filter()`	to select only certain *observations*
`data_select()`	to select only a few *attributes*
`data_extract()`	to extract a single *attribute*
`data_rename()`	to rename attributes
`reshape_longer()`	to convert data from wide to long
`reshape_wider()`	to convert data from long to wide
`data_join()`	to join two data frames
...	...

We will look at one example function that converts data in wide format to tidy/long format:

```
stocks <- data.frame(
  time = as.Date('2009-01-01') + 0:4,
  X = rnorm(5, 0, 1),
  Y = rnorm(5, 0, 2)
)

stocks
#>         time          X          Y
#> 1 2009-01-01 -0.9803102  0.3879404
#> 2 2009-01-02 -1.0518387  0.3173859
#> 3 2009-01-03  1.3891458 -1.2397131
#> 4 2009-01-04 -0.5247569 -3.7735505
#> 5 2009-01-05  0.7724189 -0.5662019

data_to_long(
  stocks,
  select = -c("time"),
  colnames_to = "stock",
  values_to = "price"
)
#>          time stock      price
#> 1  2009-01-01     X -0.9803102
#> 2  2009-01-01     Y  0.3879404
#> 3  2009-01-02     X -1.0518387
#> 4  2009-01-02     Y  0.3173859
#> 5  2009-01-03     X  1.3891458
```

```
#> 6  2009-01-03     Y -1.2397131
#> 7  2009-01-04     X -0.5247569
#> 8  2009-01-04     Y -3.7735505
#> 9  2009-01-05     X  0.7724189
#> 10 2009-01-05     Y -0.5662019
```

Statistical transformations

Even after getting the raw data in the needed format, we may further need to transform certain variables further to meet requirements imposed by the statistical model.

`{datawizard}` provides a rich collection of such functions for transforming variables (see Table 2).

Table 2: The table below lists a few key functions offered by *datawizard* for data transformations. To see the full list, see the package website: https://easystats.github.io/datawizard/

Function	Operation
`standardize()`	to center and scale data
`normalize()`	to scale variables to 0-1 range
`adjust()`	to adjust data for effect of other variables
`data_shift()`	to shift numeric value range
`ranktransform()`	to convert numeric values to integer ranks
...	...

We will look at one example function that standardizes (i.e. centers and scales) data so that it can be expressed in terms of standard deviation:

```
d <- data.frame(
  a = c(-2, -1, 0, 1, 2),
  b = c(3, 4, 5, 6, 7)
)

standardize(d, center = c(3, 4), scale = c(2, 4))
#>      a     b
#> 1 -2.5 -0.25
#> 2 -2.0  0.00
#> 3 -1.5  0.25
#> 4 -1.0  0.50
#> 5 -0.5  0.75
```

Data properties

The workhorse function to get a comprehensive summary of data properties is `describe_distribution()`, which combines a set of indices (e.g., measures of centrality, dispersion, range, skewness, kurtosis, etc.) computed by other functions in `{datawizard}`.

```
describe_distribution(mtcars$wt)
#> Mean |   SD |  IQR |        Range | Skewness | Kurtosis |  n | n_Missing
#> -----------------------------------------------------------------------
#> 3.22 | 0.98 | 1.19 | [1.51, 5.42] |     0.47 |     0.42 | 32 |         0
```

Licensing and Availability

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

datawizard 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

Dowle, M., & Srinivasan, A. (2021). *Data.table: Extension of 'data.frame'*. https://CRAN.R-project.org/package=data.table

Eastwood, N. (2021). *Poorman: A poor man's dependency free recreation of 'dplyr'*. https://CRAN.R-project.org/package=poorman

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

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

Lüdecke, 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

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. (2020). Methods and algorithms for correlation analysis in R. *Journal of Open Source Software*, *5*(51), 2306. https://doi.org/10.21105/joss.02306

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

Wickham, H., Averick, M., Bryan, J., Chang, W., McGowan, L. D., François, R., Grolemund, 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