datawizard: An R Package for Easy Data Preparation 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 cynkra Analytics GmbH, Germany 2 Nanyang Technological University, Singapore 3 Ben-Gurion University of the Negev, Israel 4 Independent Researcher 5 Luxembourg Institute of Socio-Economic Research (LISER), Luxembourg 6 University Medical Center Hamburg-Eppendorf, Germany

DOI: 10.21105/joss.04684

Software
- Review
- Repository
- Archive

Editor: Øystein Sørensen

Reviewers:
- @tomfaulkenberry
- @garretrc

Submitted: 07 August 2022
Published: 09 October 2022

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

Summary
The \{datawizard\} package for the R programming language (R Core Team, 2021) provides a lightweight toolbox to assist in key steps involved in any data analysis workflow: (1) wrangling the raw data to get it in the needed form, (2) applying preprocessing steps and statistical transformations, and (3) compute statistical summaries of data properties and distributions. Therefore, it can be a valuable tool for R users and developers looking for a lightweight option for data preparation.

Statement of Need
The \{datawizard\} package is part of \{easystats\}, a collection of R packages designed to make statistical analysis easier (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)). As this ecosystem follows a “0-external-hard-dependency” policy, a data manipulation package that relies only on base R needed to be created. In effect, \{datawizard\} provides a 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)).

Because \{datawizard\} is also meant to be used and adopted easily by a wide range of users, its workflow and syntax are designed to be similar to \{tidyverse\} (Wickham et al., 2019), a widely used ecosystem of R packages. Thus, users familiar with the \{tidyverse\} can easily translate their knowledge and make full use of \{datawizard\}.

In addition to being a lightweight solution to clean messy data, \{datawizard\} also provides helpers for the other important step of data analysis: applying statistical transformations to the cleaned data while setting up statistical models. This includes various types of data standardization, normalization, rank-transformation, and adjustment. These transformations, although widely used, are not currently collectively implemented in a package in the R ecosystem, so \{datawizard\} can help new R users in finding the transformation they need.

Lastly, \{datawizard\} also provides a toolbox to create detailed summaries of data properties and distributions (e.g., tables of descriptive statistics for each variable). This is a common step in data analysis, but it is not available in base R or many modeling packages, so its inclusion makes \{datawizard\} a one-stop-shop for data preparation tasks.

\*Brenton Wiernik is currently an independent researcher and Research Scientist at Meta, Demography and Survey Science. The current work was done in an independent capacity.

Features

Data Preparation

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 or reshaped, certain portions of the data need to be filtered out, data scattered across multiple tables needs to be joined, 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/

<table>
<thead>
<tr>
<th>Function</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>data_filter()</td>
<td>to select only certain observations</td>
</tr>
<tr>
<td>data_select()</td>
<td>to select only certain variables</td>
</tr>
<tr>
<td>data_extract()</td>
<td>to extract a single variable</td>
</tr>
<tr>
<td>data_rename()</td>
<td>to rename variables</td>
</tr>
<tr>
<td>data_to_long()</td>
<td>to convert data from wide to long</td>
</tr>
<tr>
<td>data_to_wide()</td>
<td>to convert data from long to wide</td>
</tr>
<tr>
<td>data_join()</td>
<td>to join two data frames</td>
</tr>
</tbody>
</table>

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 -0.91474184 -0.5654808
#> 2 2009-01-02 1.00124785 -1.5270177
#> 3 2009-01-03 -0.05642291 -1.3700199
#> 4 2009-01-04 0.29664516 0.7341479
#> 5 2009-01-05 -2.79147086 0.3659937
```

```
data_to_long(  
  stocks,  
  select = -c("time"),  
  names_to = "stock",  
  values_to = "price"
)
```

```
data_to_long(  
  stocks,  
  select = -c("time"),  
  names_to = "stock",  
  values_to = "price"
)
```

```
#> time stock price
#> 1 2009-01-01 X -0.91474184
#> 2 2009-01-01 Y -0.56548082
#> 3 2009-01-02 X 1.00124785
#> 4 2009-01-02 Y -1.52701766
#> 5 2009-01-03 X -0.05642291
#> 6 2009-01-03 Y -1.37001987
#> 7 2009-01-04 X 0.29664516
#> 8 2009-01-04 Y 0.73414790
```

Statistical Transformations

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

{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/

<table>
<thead>
<tr>
<th>Function</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>standardize()</td>
<td>to center and scale data</td>
</tr>
<tr>
<td>normalize()</td>
<td>to scale variables to 0-1 range</td>
</tr>
<tr>
<td>adjust()</td>
<td>to adjust data for effect of other variables</td>
</tr>
<tr>
<td>slide()</td>
<td>to shift numeric value range</td>
</tr>
<tr>
<td>ranktransform()</td>
<td>to convert numeric values to integer ranks</td>
</tr>
</tbody>
</table>

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:

```r
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
```

Summaries of Data Properties and Distributions

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}.

```r
describe_distribution(mtcars)
```
<table>
<thead>
<tr>
<th>Variable</th>
<th>Mean</th>
<th>SD</th>
<th>IQR</th>
<th>Min</th>
<th>Max</th>
<th>Skewness</th>
<th>Kurtosis</th>
<th>n</th>
<th>n_Missing</th>
</tr>
</thead>
<tbody>
<tr>
<td>mpg</td>
<td>20.09</td>
<td>6.03</td>
<td>7.53</td>
<td>10.4</td>
<td>33.9</td>
<td>0.67</td>
<td>-0.02</td>
<td>32</td>
<td>0</td>
</tr>
<tr>
<td>cyl</td>
<td>6.19</td>
<td>1.79</td>
<td>4.00</td>
<td>4.0</td>
<td>8.0</td>
<td>-0.19</td>
<td>-1.76</td>
<td>32</td>
<td>0</td>
</tr>
<tr>
<td>disp</td>
<td>230.72</td>
<td>123.94</td>
<td>221.52</td>
<td>71.1</td>
<td>472.0</td>
<td>0.42</td>
<td>-0.02</td>
<td>32</td>
<td>0</td>
</tr>
<tr>
<td>hp</td>
<td>146.69</td>
<td>68.56</td>
<td>84.50</td>
<td>52.0</td>
<td>335.0</td>
<td>0.80</td>
<td>0.28</td>
<td>32</td>
<td>0</td>
</tr>
<tr>
<td>drat</td>
<td>3.22</td>
<td>0.98</td>
<td>1.19</td>
<td>1.5</td>
<td>5.4</td>
<td>0.47</td>
<td>0.42</td>
<td>32</td>
<td>0</td>
</tr>
<tr>
<td>wt</td>
<td>17.85</td>
<td>1.79</td>
<td>2.02</td>
<td>14.5</td>
<td>22.9</td>
<td>0.41</td>
<td>0.86</td>
<td>32</td>
<td>0</td>
</tr>
<tr>
<td>qsec</td>
<td>0.44</td>
<td>0.50</td>
<td>1.00</td>
<td>0.0</td>
<td>1.0</td>
<td>0.26</td>
<td>-2.06</td>
<td>32</td>
<td>0</td>
</tr>
<tr>
<td>vs</td>
<td>0.41</td>
<td>0.50</td>
<td>1.00</td>
<td>0.0</td>
<td>1.0</td>
<td>0.40</td>
<td>-1.97</td>
<td>32</td>
<td>0</td>
</tr>
<tr>
<td>gear</td>
<td>3.69</td>
<td>0.74</td>
<td>1.00</td>
<td>3.0</td>
<td>5.0</td>
<td>0.58</td>
<td>-0.90</td>
<td>32</td>
<td>0</td>
</tr>
<tr>
<td>carb</td>
<td>2.81</td>
<td>1.62</td>
<td>2.00</td>
<td>1.0</td>
<td>8.0</td>
<td>1.16</td>
<td>2.02</td>
<td>32</td>
<td>0</td>
</tr>
</tbody>
</table>

**Licensing and Availability**

{datawizard} is licensed under the GNU General Public License (v3.0), with all source code openly developed and stored on 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**


