Journal of Statistical Software

MMMMMM YYYY, Volume VV, Issue II. *doi: 10.18637/jss.v000.i00*

cleanR: Your Personal Maid for Cleaning Datasets in R

Claus Thorn Ekstrøm
Biostatistics
Department of Public Health
University of Copenhagen

Anne Helby Petersen
Biostatistics
Department of Public Health
University of Copenhagen

Abstract

Data cleaning and validation is the first step in any data analysis since the validity of the conclusions from the analysis hinged on the quality of the input data. Ideally, a human investigator should go through each variable in the dataset and look for potential errors — both in input values and coding.

We describe an R package which implements an extensive and customizeable suite of checks to be applied to the variables in a dataset in order to identify potential problems in the corresponding variables. The typical output is a stand-alone document that summarizes the variables and lists potential errors. The results are typically presented in a stand-alone document that could be perused by an investigator with an understanding of the variables in the data and the experimental design.

The **cleanR** package is designed to be easily extended with custom user-created checks that are relevant in particular situations.

Keywords: data cleaning, quality control, R.

1. Introduction

Statisticians and data analysts spend a large portion of their time on data cleaning and on data wrangling. Packages such as **data.table**, and **plyr** have made data wrangling a lot easier in R, but there are only a few options available for automated data cleaning.

Data cleaning is a time consuming endeavour and it inherently requires human interaction since every dataset is different and the variables in the dataset can only be understood in the proper context of the experiment. While each dataset is different and requires unique attention there are often a number of similar tasks that are undertaken as part of the data cleaning and quality control process. This is especially true when data are received repeatedly

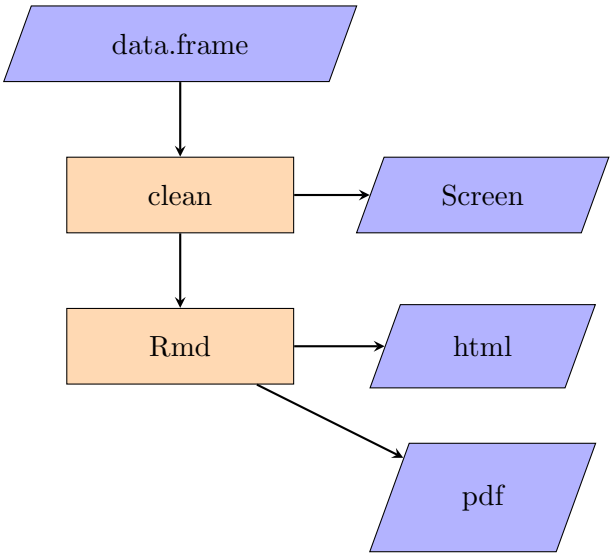

Figure 1: The process of cleaning data frames using the **cleanR** package. A series of check is applied to each variable in a `data.frame` and a summary of the result is either printed to the screen, or an R markdown file is produced which is subsequently rendered.

from the same source as data providers tend to use a stable setup (and hence introduce the same set of potential mistakes that need to be identified and corrected).

In many situations these errors are discovered in the process of the data analysis (e.g., a categorical variable with numeric labels is wrongly classified as a numeric variable), but in other cases a human with knowledge about the data context area is needed to identify possible mistakes in the data (e.g., if there are 4 categories for a variable that should only have 3). It is necessary to summarize information — both numerically and graphically — about each variable in order for an investigator to detect possible errors, and help raise flags of warning to draw the attention of investigator for the situations where there *might* be a error.

The **cleanR** package supports the automated checking of errors in a dataset and produces an output document with detailed information about each variable that can be The process is illustrated in Figure 3. A dataset is cleaned and an R markdown file is produced and possibly rendered into html or pdf.

The manusccript is organized as follows. Section 2 presents a worked example and tutorial on how to use the **cleanR** package to create a summary of potential errors. Section 3 illustrates the methodology of modeling proportional data using simplex generalized regression. The generalized estimating equations for longitudinal proportional outcomes are given in Section 4. Then we address model diagnostics in Section 5. Section 6 presents the details of the simplexreg package. Section 7 further conducts analyses based on the simplex distribution in R with real data sets. Finally, plans for extending the package are described in Section 8.

2. Checking a dataset for errors

In **cleanR** the clean function

```
> library(cleanR)
> data(testData)
> head(testData)

  charVar factorVar numVar intVar boolVar keyVar emptyVar _joeVar jack__var
1       a         a      1      1    TRUE      1        1       1         1
2       b         b      2      2   FALSE      2        1       2         2
3       c         c      3      3    TRUE      3        1       3         3
4       a         a      4      4    TRUE      4        1       4         4
5       b         b      5      5    TRUE      5        1       5         5
6       d         d      6      6   FALSE      6        1       6         6
  numOutlierVar smartNumVar      cprVar   cprKeyVar miscodedMissingVar
1             1           0 010101-1111 010101-1111                  .
2             2           0 020102-2929 020102-2929
3             3           0 121201-1902 121201-1902                nan
4             4           0 030729-2222 030729-2222                NaN
5             5           0 080909-1212 080909-1212                NAN
6             6           0 010101-1111 020202-0101                 na

>                                       # clean(testData)
> 2+3

[1] 5
```

Arguments

3. The structure of cleanR

Bør vise DF -> prechecks -> for each variable do this ...

4. Extending cleanR by adding custom error checks

Lav en situation svarende til eksemplet

```
> characterFoo <- function(v) {
+     if (substr(substitute(v), 1, 1) == "_") {
+         out <- list(problem=TRUE, message="Note that the variable name begins with \\_")
+     } else out <- list(problem=FALSE, message="")
+     out
+ }
> class(characterFoo) <- "checkFunction"
> attr(characterFoo, "description") <- "I really hate underscores"
> #clean(testData, characterChecks=c(defaultCharacterChecks(), "characterFoo"))
>
```

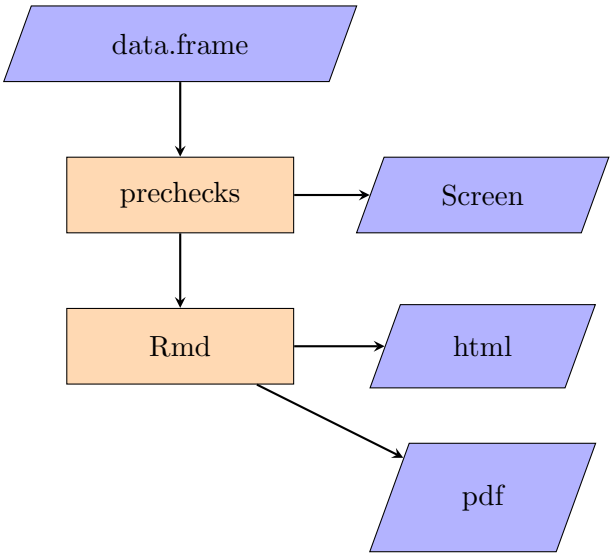

Figure 2: The process of cleaning data frames using the **cleanR** package. A series of check is applied to each variable in a `data.frame` and a summary of the result is either printed to the screen, or an R markdown file is produced which is subsequently rendered.

Lav også et eksempel med rangecheck.

5. Using the online web-app

asd asd

Affiliation:

Claus Thorn Ekstrøm
Biostatistics, Department of Public Health
University of Copenhagen
Denmark
E-mail: ekstrom@sund.ku.dk
URL: http://eeecon.uibk.ac.at/~zeileis/

Journal of Statistical Software http://www.jstatsoft.org/
published by the Foundation for Open Access Statistics http://www.foastat.org/
MMMMMM YYYY, Volume VV, Issue II *Submitted:* yyyy-mm-dd
doi:10.18637/jss.v000.i00 *Accepted:* yyyy-mm-dd