Systematic Data Cleaning for Official Statistics with R

Mark van der Loo

CBS, Department of Methodology

May 8 2019 | INE Spain

Instituto Nacional de Estadística

Agenda

- R at the office
- Data processing
- R tools for data cleaning
- Detailed examples + conclusions
- Awesome official statistics software

R at the office

Instituto Nacional de Estadística

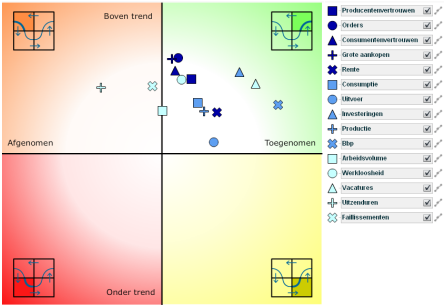
Introduction of R at statistics netherlands

- S+ and R First Used as research tool at dpt of methodology.
- R introduced as strategic tool in 2010
  - Introductory course (methods dpt)
  - User meetings
  - Wiki
  - Code standars, IT standards
- Currently 100+ users, course by stats dpts.

Use of R at Statistics Netherlands

Throughout the office.

- Deriving the business cycle indicator
- Analyzing supply-and-demand tables
- Tourism statistics
- Computing HSMR
- Editing health care institute data
- Animal population statistics
- ⋮



Instituto Nacional de Estadística

Installation and infrastructure

Current situation

- Central installation of R and RStudio, accessible to all in P-environment.
- 8-core/32G Remote Desktop Servers for heavier interactive work
- Batch environment for scheduled tasks

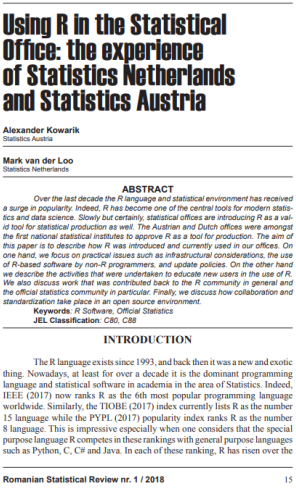
Future situation

- RStudio Server Pro

In both cases

- Curated set of pre-installed packages
- Extra packages installable via local CRAN repository
- Older versions of R remain accessible after upgrade

More information



Using R in the Statistical Office: the experience of Statistics Netherlands and Statistics Austria

Alexander Kowarik
Statistics Austria

Mark van der Loo
Statistics Netherlands

ABSTRACT

Over the last decade the R language and statistical environment has received a surge in popularity. Indeed, R has become one of the central tools for modern statistics and data science. Slowly but certainly, statistical offices are introducing R as a valid tool for statistical production as well. The Austrian and Dutch offices were amongst the first national statistical institutes to approve R as a tool for production. The aim of this paper is to describe how R was introduced and currently used in our offices. On one hand, we focus on practical issues such as infrastructural considerations, the use of R-based software by non-R programmers, and update policies. On the other hand we describe the activities that were undertaken to educate new users in the use of R. We also discuss work that was contributed back to the R community in general and the official statistics community in particular. Finally, we discuss how collaboration and standardization take place in an open source environment.

Keywords: *R Software, Official Statistics*
JEL Classification: *C80, C88*

INTRODUCTION

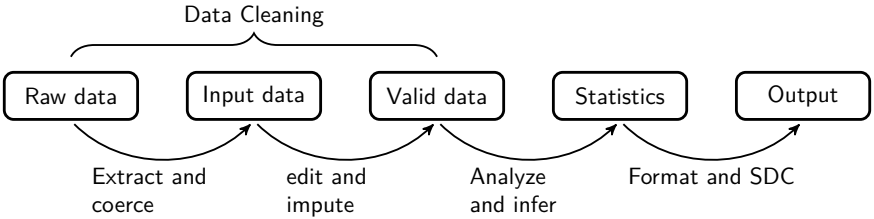
The R language exists since 1993, and back then it was a new and exotic thing. Nowadays, at least for over a decade it is the dominant programming language and statistical software in academia in the area of Statistics. Indeed, IEEE (2017) now ranks R as the 6th most popular programming language worldwide. Similarly, the TIOBE (2017) index currently lists R as the number 15 language while the PYPL (2017) popularity index ranks R as the number 8 language. This is impressive especially when one considers that the special purpose language R competes in these rankings with general purpose languages such as Python, C, C# and Java. In each of these ranking, R has risen over the

A. Kowarik and M. van der Loo. *Romanian Statistical Review* **1/2018** 15–29.

Instituto Nacional de Estadística

Data processing

(part of the) Statistical Value Chain

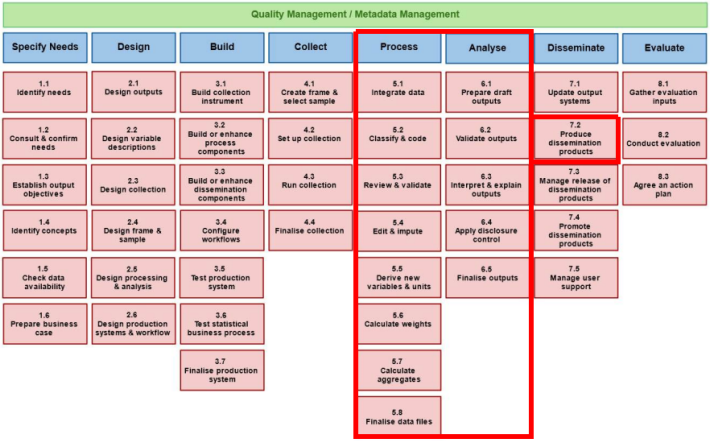


Notes

- This part only pertains to the data processing stage. Collection, design, dissemination is not included.
- The fixed points are well-defined statistical products.

Instituto Nacional de Estadística

(part of the) SVC and GSBPM



Instituto Nacional de Estadística

The SVC: Remarks

- Actual data processing is not necessarily linear across the chain
- In production architectures a more flexible model is often used where the definition of interfaces between processing steps play a crucial role. The chain shown here is a general example covering most steps in some way.
- The general idea scales really well.

Instituto Nacional de Estadística

More information

Journal of Official Statistics, Vol. 29, No. 1, 2013, pp. 49–71, DOI: 10.2478/jos-2013-0004

Redesign of Statistics Production within an Architectural Framework: The Dutch Experience

Peter Struijs[1], *Astrea Camstra*[1], *Robbert Renssen*[1], *and Barteld Braaksma*[1]

In response to a changing environment, Statistics Netherlands has embarked on a large-scale redesign of the way statistics are produced. The aim is to increase the capability to respond to changing information demand, to lower the response burden for surveys, especially for businesses, and to improve efficiency, while preserving the overall quality level. The redesign is carried out within the framework of a so-called enterprise architecture, which gives overall guidance when structuring the processes of the organisation, including statistical methods and IT tools used. The article describes the redesign approach and explains the main features of the architecture. The emphasis is on experiences that may be relevant to other national statistical institutes operating in a similar environment.

Key words: Process design; enterprise architecture; IT expenditure; standardisation; GSBPM.

1. Introduction

The context in which official statistics are produced by national statistical institutes (NSIs) has profoundly changed in the last decades (Van der Veen 2007). Many NSIs are experiencing budgetary tightening and political pressure to reduce the administrative burden of survey participation, especially in the business sector (Ypma and Zeelenberg 2007). At the same time, NSIs are facing an ever-growing demand for statistical information. This relates to new and changing user needs for relevant statistics, for example about the financial downturn or immigration, as well as to the timeliness and amount of detail with which statistics are to be released. In the European Union (EU), the number of regulations requiring EU member states to produce statistics has multiplied in the last two decades, for instance in the area of short-term statistics and annual business statistics.

In addition, technological advances have resulted in more possibilities for survey and automated data collection, and administrative registers have increasingly become available for statistical purposes. The amount of data available for use as input for statistics has grown significantly and requires integrating different data sources in an intelligent way.

[1] Statistics Netherlands, Postbus 24500, 2490 HA, Den Haag, The Netherlands. Website www.cbs.nl, for all authors. Emails: psjs@cbs.nl, acma@cbs.nl, rmsn@cbs.nl, and bbka@cbs.nl, respectively.
Acknowledgments: The authors thank Geert Bruinooge, Dirkjan Benkenhorst, Frank Hofman and Kees Zeelenberg, Statistics Netherlands, for their contribution to the discussion and their comments on drafts of the article. Comments by the guest editors of *JOS* were especially valuable. The views expressed in this article are

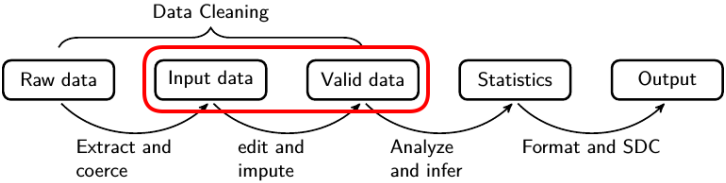
P. Struijs, A. Camstra, R. Renssen and B. Braaksma. *Journal of Official Statistics* **29/1** 2013 49–71.

Instituto Nacional de Estadística

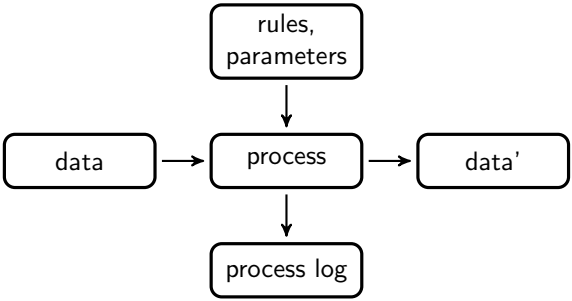
R tools for data cleaning

Instituto Nacional de Estadística

Data cleaning

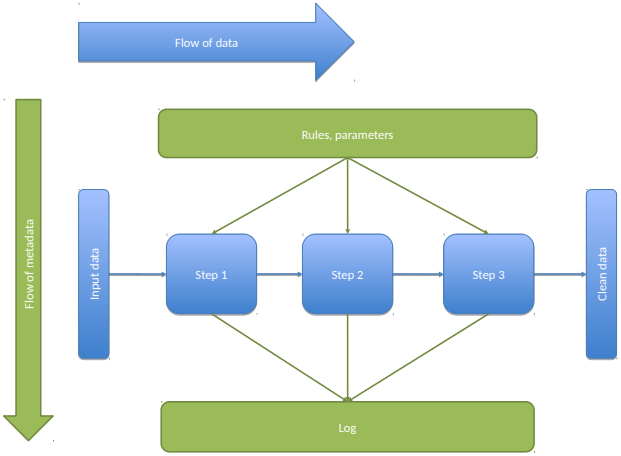


High-level process view (CSPA, GSIM)



Separation of concerns + Modular approach

Slightly more realistic process view



What does that mean for implementation?

Our approach

- Separation of concerns → DSL
- Modularity == **UNIX philosophy**
  - each tool doing one thing, really well.
  - tools should integrate with ease
- Design for **humans**
- In **open source**
- **Reuse** existing tools where possible

R

- Free/Libre and open source
- Well-established, many users
- Very flexible (functional PL)
- Best package system

By the way

R is probably the most thoroughly validated statistics software on Earth.

Uwe Ligges, CRAN maintainer (useR!2017)

R tools, available on CRAN

R> dcmodify
User-defined data cleaning

R> rspa
Alter data to pass validation rules

R> simputation
Missing data imputation

R> deductive
Use validation rules to repair data

R> errorlocate
Localize erroneous fields

R> lumberjack
Track changes in data

R> validate
Define rules, measure data quality

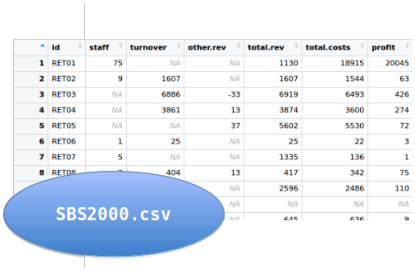
R> validatetools
Maintain and investigate rules

R> validatereport
Validation reports in ESS standard

Instituto Nacional de Estadística

Implementation: preparation

```
dat    <- read.csv("data/SBS2000.csv")
rules  <- validate::validator(.file="data/rules.R")
logger <- validate::lbj_rules(rules)
```



	id	staff	turnover	other.rev	total.rev	total.costs	profit
1	RET01	75	NA	NA	1130	18915	20045
2	RET02	9	1607	NA	1607	1544	63
3	RET03	NA	6886	-33	6919	6493	426
4	RET04	NA	3861	13	3874	3600	274
5	RET05	NA	NA	37	5602	5530	72
6	RET06	1	25	NA	25	22	3
7	RET07	5	NA	NA	1335	136	1
8	RET08		404	13	417	342	75
				NA	2596	2486	110
				NA	NA	NA	NA
				NA	645	636	9

SBS2000.csv

```
1
2 # range resitrictions
3 staff >= 0
4 turnover >= 0
5 other.rev >= 0
6
7
8 # balance restrictions
9 turnover + other.rev == total.rev
0 total.rev - total.costs == profit
1 profit <= 0.6 * total.rev
2
```

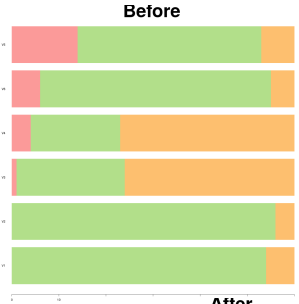
rules.R

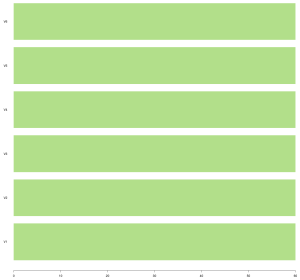
Implementation: data cleaning

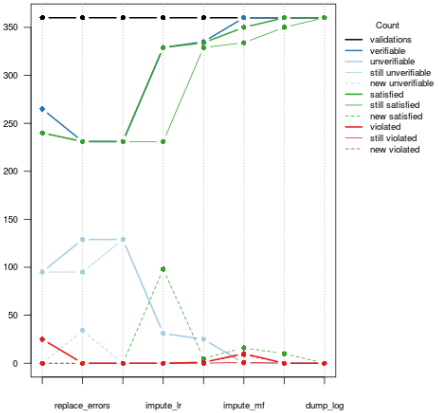
```
dat %L>%
    lumberjack::start_log(logger) %L>%
    errorlocate::replace_errors(rules) %L>%
    tag_missing() %>%
    simputation::impute_mf(. - id ~ . - id) %L>%
    rspa::match_restrictions(rules, eps=1E-8) %L>%
    dump_log() ->
    clean_data
```

Instituto Nacional de Estadística

Results







Instituto Nacional de Estadística

Detailed examples

The `validate` package

Purpose: data checking (validation)

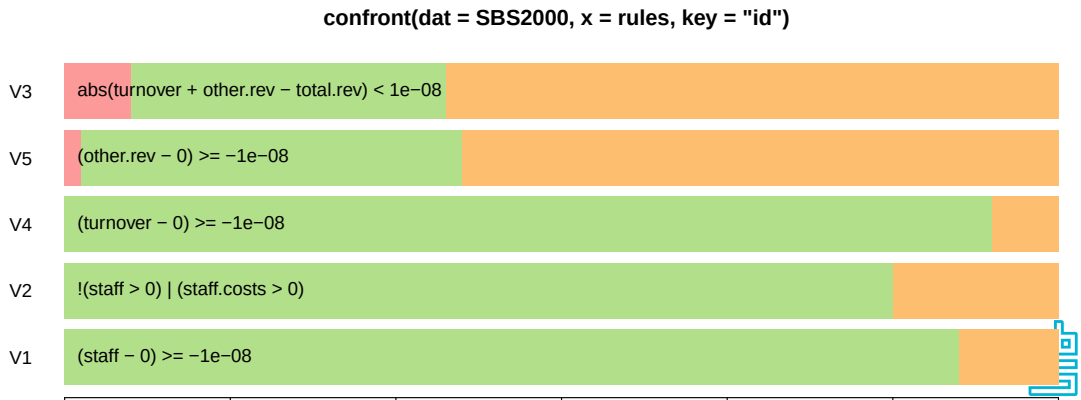
- Define, apply, and maintain data validation rules.
- Summarize, investigate, export validation results.
- Separate domain knowledge from programming skills.

Example data

```
library(validate)
SBS2000 <- read.csv("data/SBS2000.csv")
rules <- validator(
  staff >= 0
  , if (staff > 0) staff.costs > 0
  , turnover + other.rev == total.rev
  , turnover >= 0, other.rev >= 0
)
```

Confronting data with rules; plotting, summarizing

```
result <- confront(SBS2000, rules, key="id")
plot(result)
```



Confronting data with rules; plotting, summarizing

```
summary(result)
```

```
##   name items passes fails nNA error warning
## 1   V1    60     54     0   6 FALSE   FALSE
## 2   V2    60     50     0  10 FALSE   FALSE
## 3   V3    60     19     4  37 FALSE   FALSE
## 4   V4    60     56     0   4 FALSE   FALSE
## 5   V5    60     23     1  36 FALSE   FALSE
##                                     expression
## 1                          (staff - 0) >= -1e-08
## 2               !(staff > 0) | (staff.costs > 0)
## 3 abs(turnover + other.rev - total.rev) < 1e-08
## 4                        (turnover - 0) >= -1e-08
## 5                      (other.rev - 0) >= -1e-08
```

Validation results are data

```
head( as.data.frame(result) )
```

```
##      id name value            expression
## 1 RET01   V1  TRUE (staff - 0) >= -1e-08
## 2 RET02   V1  TRUE (staff - 0) >= -1e-08
## 3 RET03   V1    NA (staff - 0) >= -1e-08
## 4 RET04   V1    NA (staff - 0) >= -1e-08
## 5 RET05   V1    NA (staff - 0) >= -1e-08
## 6 RET06   V1  TRUE (staff - 0) >= -1e-08
```

Validation rules are also data

```
rules
```

```
## Object of class 'validator' with 5 elements:
##  V1: staff >= 0
##  V2: !(staff > 0) | (staff.costs > 0)
##  V3: turnover + other.rev == total.rev
##  V4: turnover >= 0
##  V5: other.rev >= 0
```

```
# rules can have rich metadata
rules[[2]]
```

```
##
## Object of class rule.
##  expr        : if (staff > 0) staff.costs > 0
##  name        : V2
##  label       :
##  description:
##  origin      : command-line
##  created     : 2019-05-23 09:46:04
##  meta        : language<chr>, severity<chr>
```

Validation rules can be investigated

```
variables(rules)
```

```
## [1] "staff"      "staff.costs" "turnover"    "other.rev"   "total.rev"
```

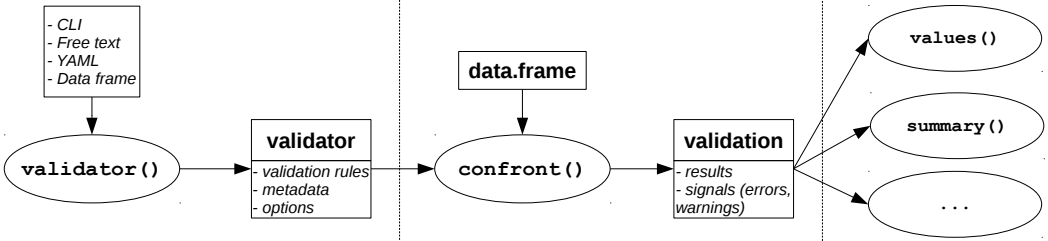
Are all variables in the data covered by the rules?

```
all( variables(SBS2000) %in% variables(rules) )
```

```
## [1] FALSE
```

Instituto Nacional de Estadística

Package overview



Example: rules with rich metadata (YAML)

```yaml
---
options:
  raise: none
  lin.eq.eps: 1.0e-08
  lin.ineq.eps: 1.0e-08
---
rules:
- expr: turnover >= 0
  name: 'rule-01'
  label: 'nonnegative income'
  description: |
    'Income cannot be negative (unlike in the
     definition of the tax office)'
  created: 2018-06-05 14:44:06
  origin: rules.txt
  meta: []
```

Imputation

Imputation in R

Specialized packages

- Many available (VIM, mice, Amelia, mi, ...)
- Interfaces vary (a lot)

DIY with model/predict

```
m <- lm(Y ~ X, data=mydata)
ina <- is.na(mydata$Y)
mydata[ina, "Y"] <- predict(m, newdata = mydata[ina,])
```

- Code duplication, doesn't always work

Goal of the `simputation` package

Provide

- a *uniform interface*,
- with *consistent behaviour*,
- across *commonly used methodologies*

To facilitate

- experimentation
- configuration for production

A first example

```
dat <- SBS2000[4:7]
dat %>% head(3)
```

```
##   staff turnover other.rev total.rev
## 1    75       NA        NA      1130
## 2     9     1607        NA      1607
## 3    NA     6886       -33      6919
```

```
dat %>% impute_lm(other.rev ~ turnover) %>% head(3)
```

```
##   staff turnover other.rev total.rev
## 1    75       NA        NA      1130
## 2     9     1607  5427.113      1607
## 3    NA     6886   -33.000      6919
```

The `simputation` package

An imputation prodedure is specified by

1. The variable to impute
2. An imputation model
3. Predictor variables

The simputation interface

```
impute_<model>(data
  , <imputed vars> ~ <predictor vars>
  , [options])
```

Experimenting with methods

```
# linear model
dat %>% impute_lm(other.rev ~ turnover) %>% head(3)
```

```
##   staff turnover other.rev total.rev
## 1    75       NA        NA      1130
## 2     9     1607  5427.113      1607
## 3    NA     6886   -33.000      6919
```

```
# robost linear model (M-estimator)
dat %>% impute_rlm(other.rev ~ turnover) %>% head(3)
```

```
##   staff turnover other.rev total.rev
## 1    75       NA        NA      1130
## 2     9     1607  17.25247      1607
## 3    NA     6886 -33.00000      6919
```

Example (ct'd): chaining methods

```
dat %>%
  impute_rlm( other.rev ~ turnover ) %>%
  impute_rlm( other.rev ~ staff    ) %>%
  head(3)
```

```
##   staff turnover other.rev total.rev
## 1    75       NA  64.88174      1130
## 2     9     1607  17.25247      1607
## 3    NA     6886 -33.00000      6919
```

Example: grouping

```
SBS2000 %>% impute_rlm(total.rev ~ turnover | size)

# or, using dplyr::group_by
SBS2000 %>%
  group_by(size) %>%
  impute_rlm(total.rev ~ turnover)
```

Example: add random residual

```
SBS2000 %>% impute_rlm(total.rev ~ turnover | size,
                       add_residual="observed")

SBS2000 %>% impute_rlm(total.rev ~ turnover | size,
                       add_residual="normal")
```

Instituto Nacional de Estadística

Example: proxy imputation

```
# impute VAT value where turnover is unavailable
SBS2000 %>% impute_proxy(turnover ~ vat)
```

```
# transformations are also allowed
SBS2000 %>% impute_proxy(other.rev ~ total.rev - turnover)
```

Instituto Nacional de Estadística

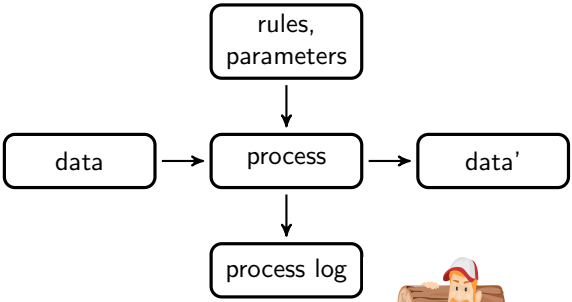
Example: train on A, apply to B

```
m <- MASS::rlm(other.rev ~ turnover + staff
               , data=SBS1999)
impute(SBS2000, other.rev ~ m)
```

Currently available methods

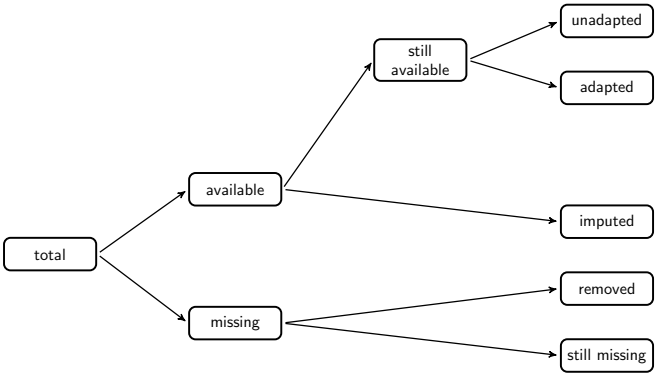
- Model based (optional random residual):
  - standard/M/elasticnet regression
  - CART models and Random forest
- Multivariate
  - EM-based imputation
  - missForest (=iterative random forest)
- Donor imputation (including various donor pool specifications)
  - k-nearest neigbour (based on gower's distance)
  - sequential, random hotdeck
  - Predictive mean matching
- Other
  - (groupwise) median imputation (optional random residual)
  - Proxy imputation: copy another variable or use a simple transformation to compute imputed values.

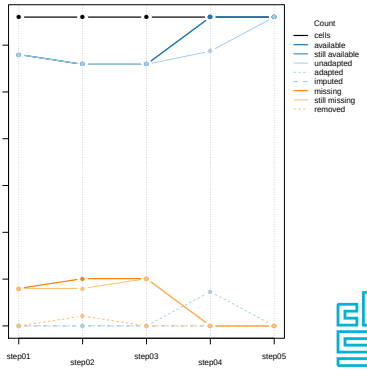
Tracking changes in data with `lumberjack`



How to compare two versions of a dataset? (1)

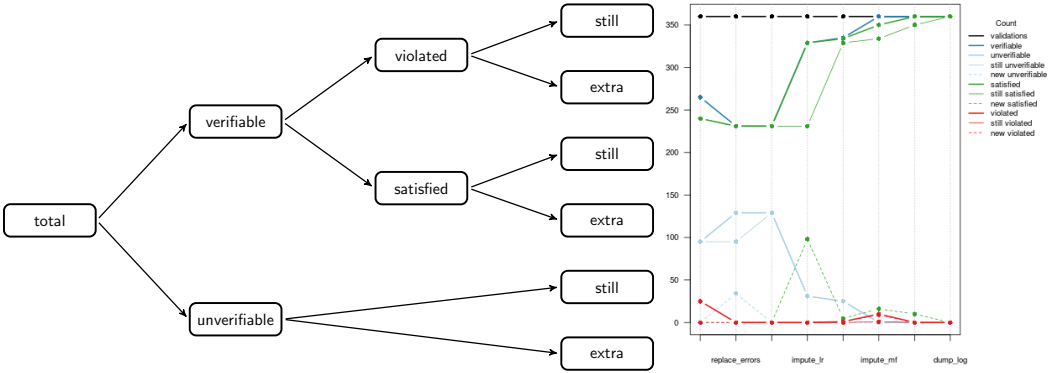
Count changes in cells





How to compare two versions of a dataset? (2)

Count changes in rule violations



How to compare two versions of a dataset? (3)

There are more ways

- Comparing cell-by cell
- Following summary statistics (e.g. means)
- ⋮

The `lumberjack` package

- Works for *any* data transformation function (no special protocols required)
- What to log is *user-definable*, and *extensible*
- Near-zero change in common R workflow

Simple example

```
SBS2000 %L>%
  start_log(cellwise$new(key="id")) %L>%
  impute_const(other.rev ~ 0) %L>%
  dump_log() -> imputed_data
```

```
## Dumped a log at cellwise.csv
```

```
read.csv("cellwise.csv") %>% head(3)
```

```
##   step                      time                   expression   key
## 1    1 2019-05-23 09:46:05 CEST impute_const(other.rev ~ 0) RET01
## 2    1 2019-05-23 09:46:05 CEST impute_const(other.rev ~ 0) RET02
## 3    1 2019-05-23 09:46:05 CEST impute_const(other.rev ~ 0) RET06
##    variable old new
## 1 other.rev  NA   0
## 2 other.rev  NA   0
## 3 other.rev  NA   0
```

Conclusions

Lessons learned

(1) Modularity

Requires careful research. Basic actions (like data validation, logging) preferably defined with mathemtical precision.

(2) Separation of concerns, building for humans

- Follows, if (1) has been thought through well enough.
- Building something that *lasts* still requires careful thinking, and sometimes new ideas.

(3) Implementing

- Using FOSS tools (here: R) makes the design → build → test quick and easy.
- The fact that R is a functional programming language offers abstractions that are much harder to build in non-functional languages.
- Still needs deep thinking sometimes.

The Dolly Parton principle

Instituto Nacional de Estadística

The awesomelist of official statistics software

Instituto Nacional de Estadística

What are awesomelists?

Awesomlist

- A *curated* list of things, considered *AWESOME* by the curator(s)
- Hosted on github

Awesome official statistics software

Official statistics software is *AWESOME* if and only if

1. It is free, open source and available for download
2. It is confirmed to be used in the production of official statistics by at least one institute *or*
3. it provides access to official statistics publications

Also

We prefer packages that are reasonably easy to install and use, that have at least one stable version, and that are actively maintained.

O. ten Bosch and M. van der Loo, www.awesomeofficialstatistics.org

Awesome official statistics software

Design frame and sample (GSBPM 2.1)

- R package SamplingStrata. Optimal Stratification of Sampling Frames for Multipurpose Sampling Surveys.

Sampling (GSBPM 4.1)

- R package sampling. Several algorithms for drawing (complex) survey samples and calibrating design weights.
- R package surveyplanning. Tools for sample survey planning, including sample size calculation, estimation of expected precision for the estimates of totals, and calculation of optimal sample size allocation.

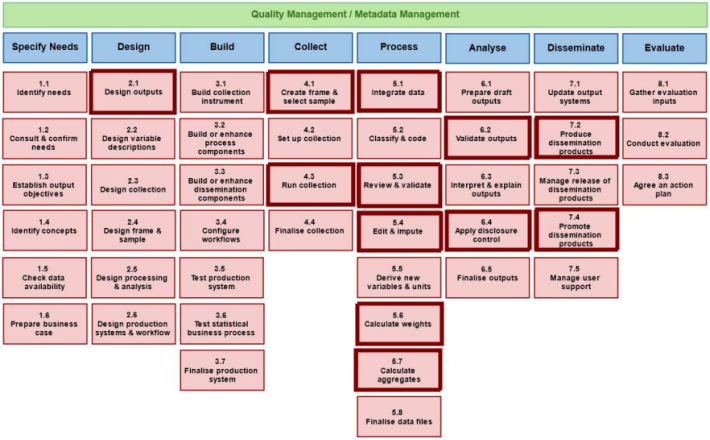
Scraping for Statistics (GSBPM 4.3)

- Java application URLSearcher. An application for searching Urls. Can be used to find websites of enterprise. By ISTAT.
- Java application URLScorer. Gives a rule based score to scraped documents in a Solr database. By ISTAT.
- node.js tool RobotTool. A tool for checking (price) changes on the web. By Statistics Netherlands.
- Python Social-Media-Presence. A script for detecting social media presence on enterprises websites. By Statistics Poland.
- Python Sustainability Reporting. A script for measuring sustainability reporting from enterprises websites. By ONS.
- node.js package S4Sroboto. A crawler framework, derived from the general package roboto extended with some functionalities for statistical scraping. By Statistics Netherlands

Process (GSBPM 5)

- Java application Java-VTL. A partial implementation of the Validation Transformation Language, based on the VTL 1.1 draft specification. By Statistics Norway.
- Java application ADaMSoft implements procedures for data analysis, data, web and text mining. Also contains procedures for data validation and imputation, based on the principle of Fellegi and Holt

Parts of GSBPM covered



Instituto Nacional de Estadística

If you are awesome, you get to wear the *BADGE*

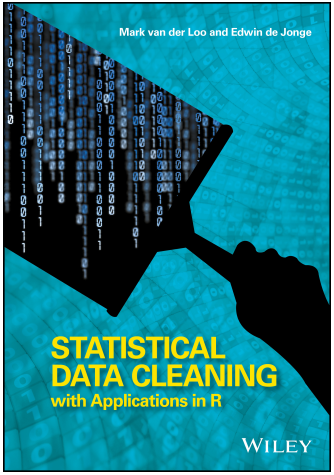
README.md

VIM

This package introduces new tools for the visualization of missing and/or imputed values, which can be used for exploring the data and the structure of the missing and/or imputed values. Depending on this structure of the missing values, the corresponding methods may help to identify the mechanism generating the missings and allows to explore the data including missing values. In addition, the quality of imputation can be visually explored using various univariate, bivariate, multiple and multivariate plot methods.

M. Temple *et al.* https://github.com/statistikat/VIM

More information



Contact

- @markvdloo
- github.com/markvanderloo
- mplo@cbs.nl

Book MPJ van der Loo and E de Jonge (2018) John Wiley & Sons.

More FOSS

www.awesomeofficialstatistics.org

Instituto Nacional de Estadística