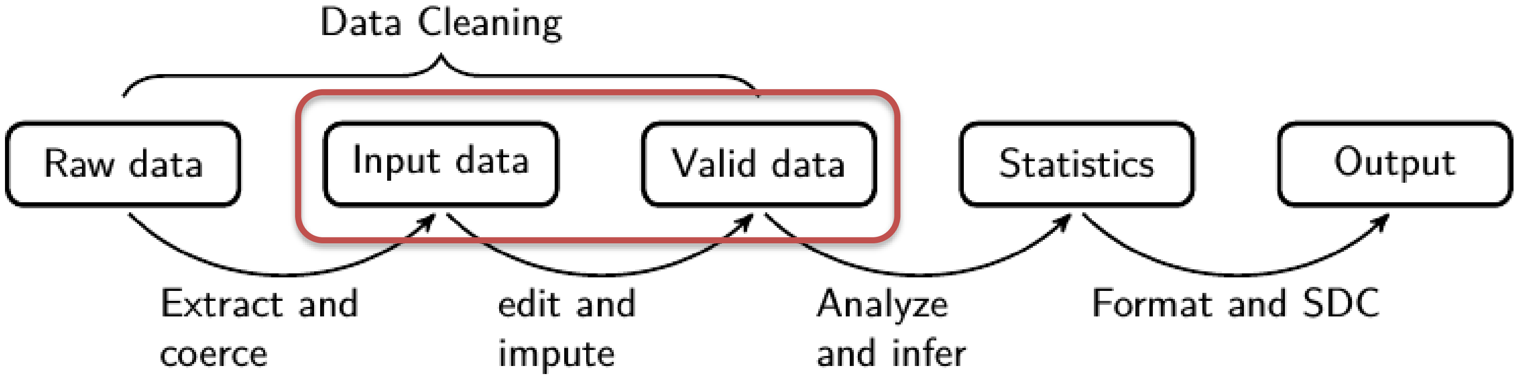
Systematic Data Cleaning using R

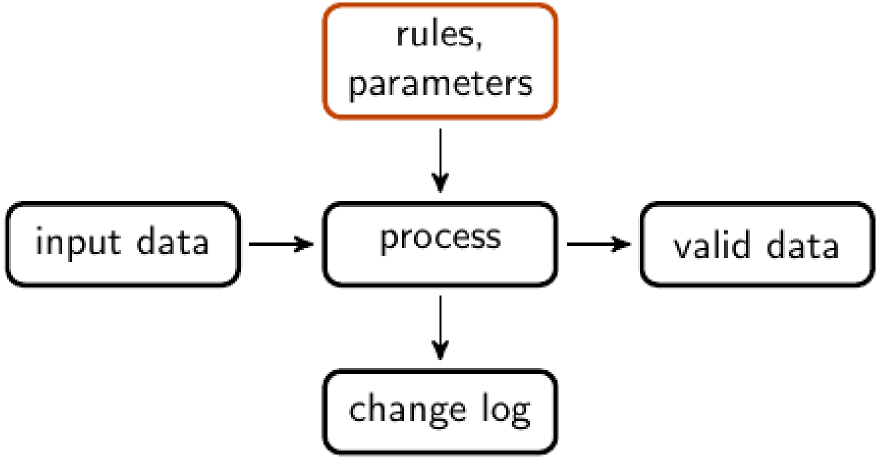
Mark van der Loo
Dpt of Methodology
NTTS 13-03-2019

Data Cleaning in the Statistical Value Chain



Ultimate goal

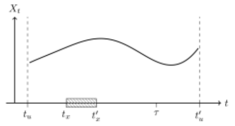
- User specifies what is ‘valid data’.
- Everything else is automated.



Approach

- Think deeply about the nature of ‘data validation’
- Design tools
  - In **open source**
  - Following the **Unix philosophy** (small, powerful, combinable)
  - With **humans** and modern **standards** in mind
  - While **re-using** existing tools where possible

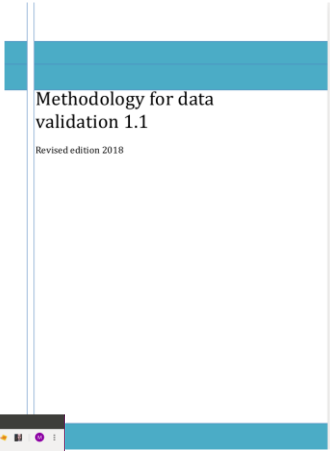
Results, ESS collaboration



$$T \xrightarrow{p} 2^U \xrightarrow{I_u} U \xrightarrow{X_\tau} D,$$

Definition 6.3.4 *Let D^K be the set of possible datasets as defined in 6.3.2. A data validation function v is a surjective function*

$$v : D^K \twoheadrightarrow \{0,1\}, \tag{6.1}$$



Methodology for data validation 1.1

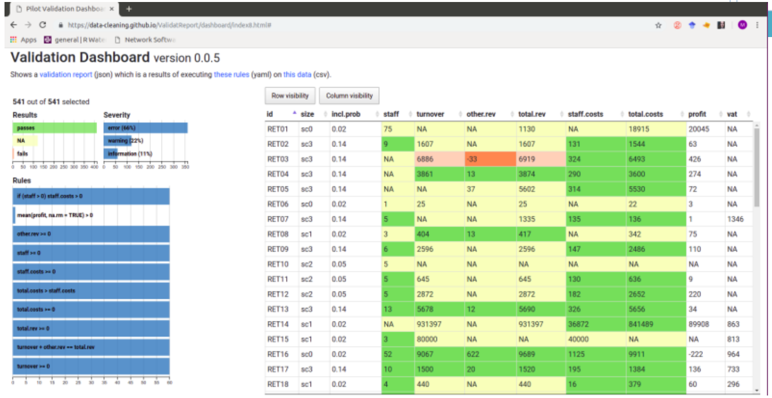
Revised edition 2018

ValiDat

Design of a generic machine-readable validation report structure

Mark van der Loo and Olav ten Bosch
Statistics Netherlands

Version 1.0.0 October 23, 2017



Validation Dashboard version 0.0.5

Shows a validation report (json) which is a results of executing these rules (yaml) on this data (csv).

541 out of **541** selected

- if (staff > 0) staff.costs > 0
- mean(profit, na.rm = TRUE) > 0
- other.rev >= 0
- staff >= 0
- staff.costs >= 0
- total.costs > staff.costs
- total.costs >= 0
- total.rev >= 0
- turnover + other.rev == total.rev
- turnover >= 0

id	size	incl.prob	staff	turnover	other.rev	total.rev	staff.costs	total.costs	profit	vat
RET01	sc0	0.02	75	NA	NA	1130	NA	18915	20045	NA
RET02	sc3	0.14	9	1607	NA	1607	131	1544	63	NA
RET03	sc3	0.14	NA	6886	-33	6919	324	6493	426	NA
RET04	sc3	0.14	NA	3861	13	3874	290	3600	274	NA
RET05	sc3	0.14	NA	NA	37	5602	314	5530	72	NA
RET06	sc0	0.02	1	25	NA	25	NA	22	3	NA
RET07	sc3	0.14	5	NA	NA	1335	135	136	1	1346
RET08	sc1	0.02	3	404	13	417	NA	342	75	NA
RET09	sc3	0.14	6	2596	NA	2596	147	2486	110	NA
RET10	sc2	0.05	5	NA	NA	NA	NA	NA	NA	NA
RET11	sc2	0.05	5	645	NA	645	130	636	9	NA
RET12	sc2	0.05	5	2872	NA	2872	182	2652	220	NA
RET13	sc3	0.14	13	5678	12	5690	326	5656	34	NA
RET14	sc1	0.02	NA	931397	NA	931397	36872	841489	89908	863
RET15	sc1	0.02	3	80000	NA	NA	40000	NA	NA	813
RET16	sc0	0.02	52	9067	622	9689	1125	9911	-222	964
RET17	sc3	0.14	10	1500	20	1520	195	1384	136	733
RET18	sc1	0.02	4	440	NA	440	16	379	60	296

R-based 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

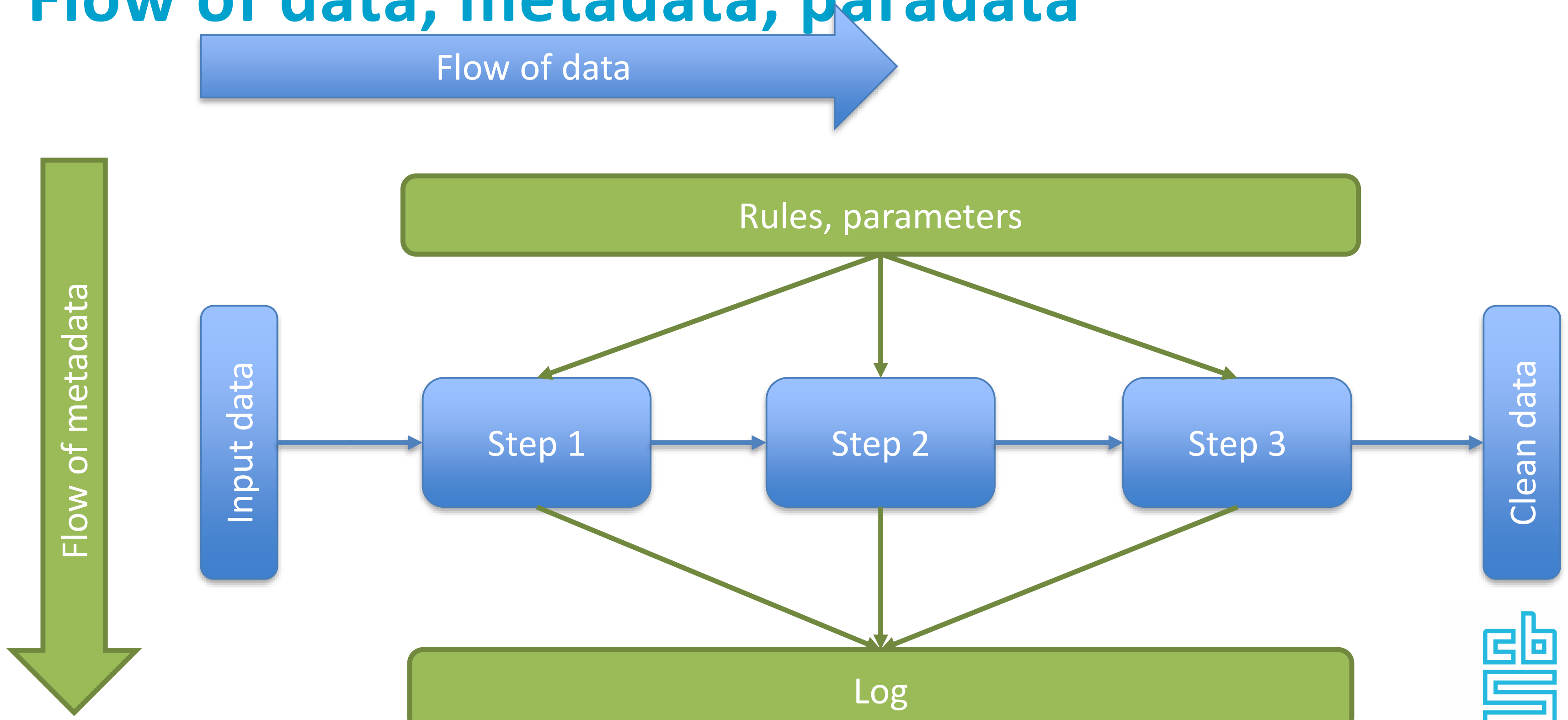
```
R> lumberjack
```
Track changes in data

```
R> validate
```
Define rules, measure data quality

```
R> validatetools
```
Maintain and investigate rules

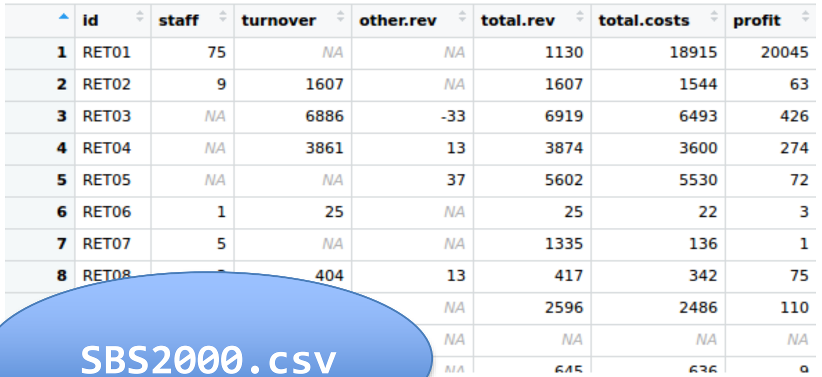
```
R> validatereport
```
Validation reports in ESS standard

Flow of data, metadata, paradata



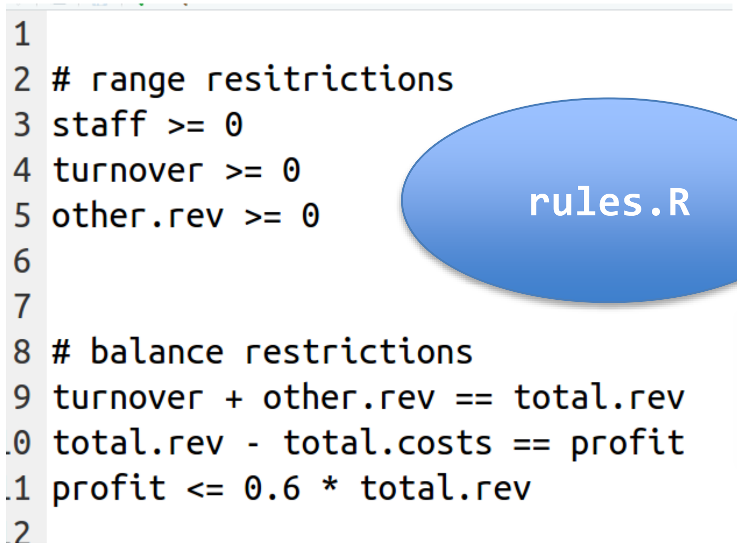
Implementation (1): preparation

```
dat    <- read.csv( “SBS2000.csv” )
rules  <- validate::validator( .file = “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	[illegible]	404	13	417	342	75
[illegible]	[illegible]	[illegible]	[illegible]	NA	2596	2486	110
[illegible]	[illegible]	[illegible]	[illegible]	NA	NA	NA	NA
[illegible]	[illegible]	[illegible]	[illegible]	NA	645	636	9

SBS2000.csv



```

# range resitrictions
staff >= 0
turnover >= 0
other.rev >= 0


# balance restrictions
turnover + other.rev == total.rev
total.rev - total.costs == profit
profit <= 0.6 * total.rev

```

rules.R

Implementation (2): execution

```
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
```

Implementation (2): execution

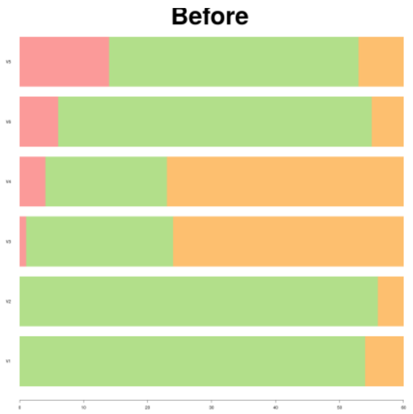
```
dat %L>%
    lumberjack::start_lo                 ) %L>%
    errorlocate::replace_errors(rules) %L>%
    tag_missing() %>%
    simputation::impute_mf(               - id) %L>%
    rspa::match_restriction               s=1E-8) %L>%
    dump_log() ->
    clean_data
```

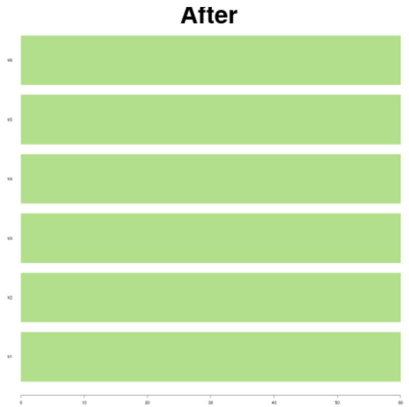
Process block

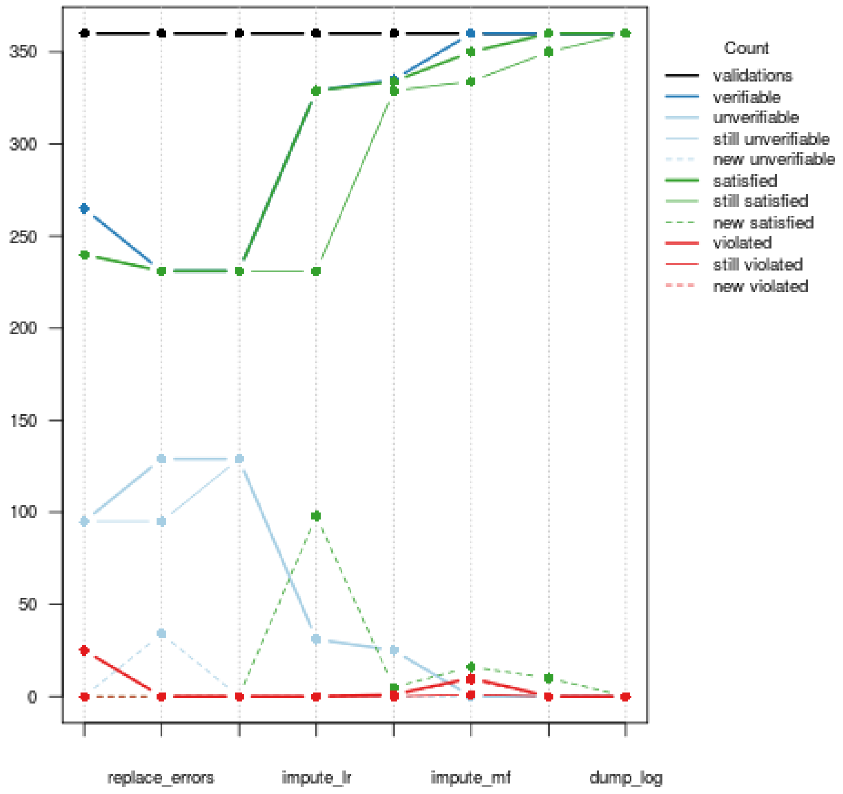
Pipe operator, controlling flow of data and meta-data

User Specification

Implementation (3) process monitoring



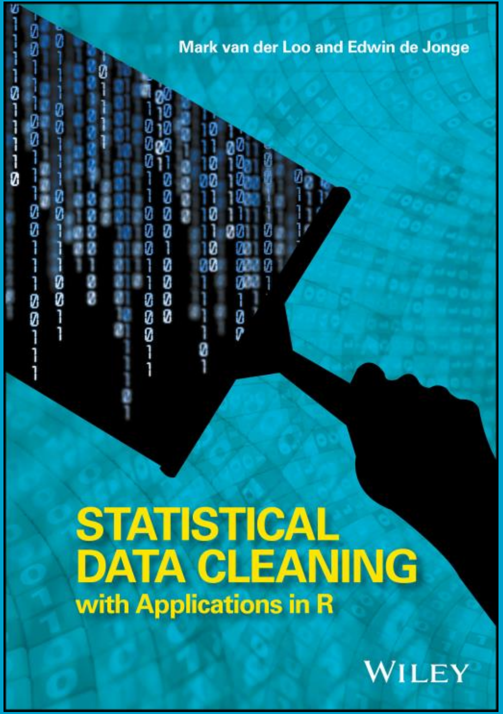




Implementation in CBS

- Used in production, e.g.
  - Health care institutes
  - Energy (currently under redesign)
  - Imputation of population registers
  - …
- How?
  - Internal courses (CBS academy)
  - Redesign of production systems

More information


Mark van der Loo and Edwin de Jonge
STATISTICAL DATA CLEANING
with Applications in R
WILEY

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

Upcoming tutorials:
- uRos2019 conference (Bucharest)
- useR2019 conference (Toulouse)
- ENBES2019 conference (BilBao)

More FOSS for official statistics:
www.awesomeofficialstatistics.org