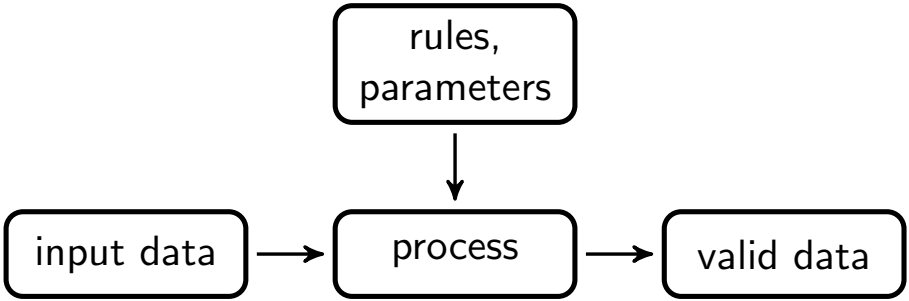
Automating data quality measurement

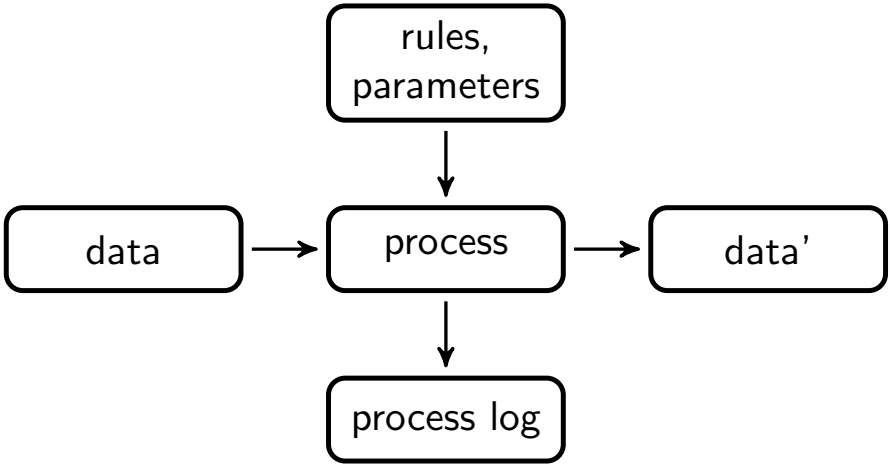
Mark van der Loo and Edwin de Jonge

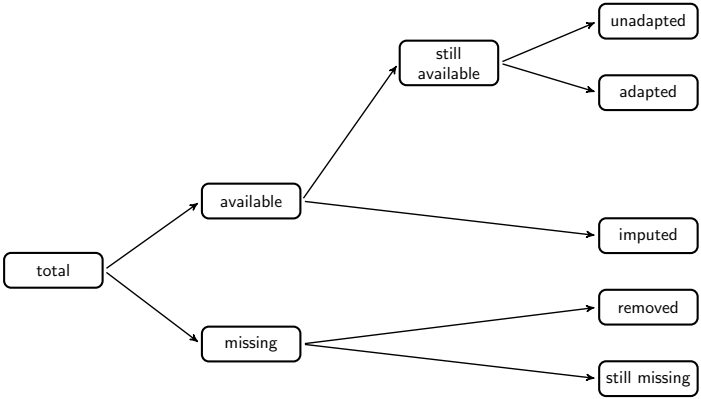
Statistics Netherlands Research & Development
@markvdloo @edwindjonge

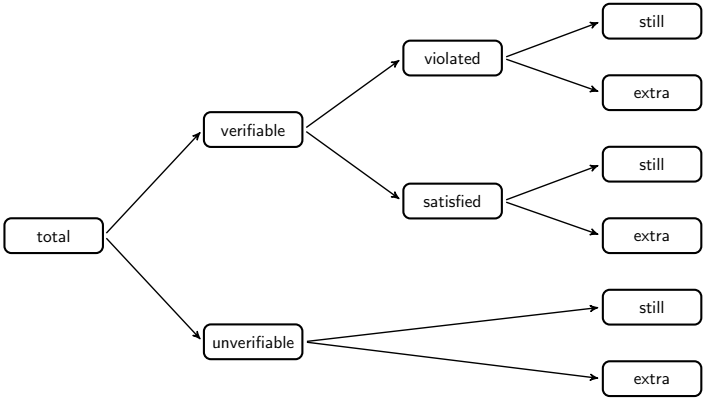
useR!2021

- Measuring differences in data
- Measuring evolution of quality


rules,
parameters
input data
process
valid data


rules,
parameters
data
process
data'
process log


total
available
missing
still available
imputed
removed
still missing
unadapted
adapted


total
verifiable
unverifiable
violated
satisfied
still
extra
still
extra
still
extra

lumberjack

Track changes in data

- Without *changing* code
- Customizable, user-defined loggers
- Track multiple data sets
- Track a data set in multiple ways

Main functions

Logger	what it does
`start_log`	Assign a logger to an `R` object.
`stop_log`	Stop logging and dump log, where dumping can be switched off.
`dump_log`	Dump logging info and stop logging, where stopping is optional
`run_file`	Execute a file, while logging, in a new `environment`.
`source_file`	Execute a file, while logging, in the global `environment`.

Built-in loggers

Logger	what it does
`expression_logger`	record result of custom `R` expressions.
`filedump`	dump a file after each operation.
`simple`	record whether anything changed (`logical`).
`cellwise`	record cell-by-cell changes.

Using lumberjack in a script

```
library(lumberjack)
# create some loggers
logger1 <- filedump$new(dir="/my/output")
logger2 <- cellwise(key="id")
# add both loggers
start_log(dat, logger1)
start_log(dat, logger2)

# Code doing stuff with 'dat'

# dump one of the loggers
dump_log(dat, logger="filedump")

# continue modifying 'dat'
#...
```

The expression logger

Track any summary statistic

```
logger <- expression_logger$new(
    av_turnover = mean(turnover, na.rm=TRUE)
  , av_staffcost = mean(staff.costs, na.rm=TRUE)
)

start_log(dat, logger)
```

Create your own logger

Loggers are reference (`R6`, or base `R`) objects with two obligatory methods

- `$add(meta, input, output)` Compute difference between in- and output
  - `meta`: R expression and source
  - `input`: data before expression evaluation
  - `output`: data after expression evaluation
- `$dump()` Make the logger dump the logging info

`validate` exports two loggers

`lbj_cells()`

Keep track of diffence in data, using the `cells()` function.

```
logger <- validate::lbj_cells()
```

`lbj_rules`

Keep track of changes in rule violations, using the `compare()` function.

```
logger <- validate::lbj_rules(rules)
```

Exercise

1. Use the `lbj_rules()` logger to track changes in rule violations for the `clean_supermarkets.R` script.
   - use the rules in `code/SBS2000_rules.R`
2. Add a logger that tracks the `mean` of variables `turnover` and `staff.costs`

Note

- `R6` loggers are initialized with `$new()` (like the expression logger)
- Base R reference class loggers are initialized with `loggername()` (like `lbj_rules()`)