Data errors, how to find them?

Edwin de Jonge, Statistics Netherlands

@edwindjonge | github.com/edwindj


CAUTION: BAD DATA
BAD DATA QUALITY
MAY RESULT IN
FRUSTRATION AND
LEAD TO DROP
KICKING YOUR
COMPUTER

Data cleaning...

A large part of your job is spent in data-cleaning:

- getting your data in the right shape (e.g. `tidyverse`)
- assessing missing data (e.g. `VIM`)
- checking validity (e.g. `validate`)
- locating and removing errors: **`errorlocate`**!
- impute values for missing or erroneous data (e.g. `simputation`)

Centraal Bureau
voor de Statistiek

Validation rules?

Package `validate` allows to:

- formulate explicit data rule that data must conform to:

```
library(validate)
check_that( data.frame(age=160, driver_license=TRUE),
  age >= 0,
  age < 150,
  if (driver_license == TRUE) age >= 16
)
```

Explicit validation rules:

- Give a clear overview what the data must conform to.
- Can be used to reason about.
- Can be used to fix/correct data!
- Find error, and when found correct it.

Note:

- Manual fix is error prone, not reproducible and not feasible for large data sets.
- Large rule set have (very) complex behavior, e.g. entangled rules: adjusting one value may invalidate other rules.

Error localization

Error localization is a procedure that points out fields in a data set that can be altered or imputed in such a way that all validation rules can be satisfied.

Find the error:

```
library(validate)
check_that( data.frame(age=160, driver_license=TRUE),
  age >= 0,
  age < 150,
  if (driver_license == TRUE) age >= 16
)
```

It is clear that age has an erroneous value, but for more complex rule sets it is less clear.

Multivariate example:

```
check_that( data.frame( age      = 3
                      , married = TRUE
                      , attends = "kindergarten"
                      )
          , if (married == TRUE) age >= 16
          , if (attends == "kindergarten") age <= 6
          )
```

Ok, clear that this is a faulty record, but what is the error?

Feligi Holt formalism:

> *Find the minimal (weighted) number of variables that cause the invalidation of the data rules.*

Makes sense! (But there are exceptions. . .)

Implemented in `errorlocate` (second generation of `editrules`).

errorlocate::locate_errors

```
locate_errors( data.frame( age      = 3
                         , married = TRUE
                         , attends = "kindergarten"
                         )
     , validator( if (married == TRUE) age >= 16
                , if (attends == "kindergarten") age <= 6
                )
     )$errors
```

```
##         age married attends
## [1,] FALSE    TRUE   FALSE
```

errorlocate::replace_errors

```
replace_errors(
    data.frame( age      = 3
              , married = TRUE
              , attends = "kindergarten"
              )
  , validator( if (married == TRUE) age >= 16
             , if (attends == "kindergarten") age <= 6
             )
)
```

```
##   age married      attends
## 1   3      NA kindergarten
```

Internal workings:

`errorlocate`:

- translates error localization problem into a **mixed integer problem**, which is solved with `lpsolveAPI`.
- contains a small framework for implementing your own error localization algorithms.

Pipe friendly

The `replace_errors` function is pipe friendly:

```
rules <- validator(age < 150)

data_noerrors <-
  data.frame(age=160, driver_license = TRUE) %>%
  replace_errors(rules)

errors_removed(data_noerrors) # contains errors removed
```

Thank you!

Interested?

```
install.packages("errorlocate")
```

Or visit:

http://github.com/data-cleaning/errorlocate