Assignment 4: Rule management

Edwin de Jonge and Mark van der Loo

useR!2021

Exercise 1, rule management

- What is maximum number of columns in a dataset you encountered in your work?
- Can you give an indication of the maximum number of data validity rules that are checked in production process ?
- How many persons are involved in checking and maintaining the rules?

Exercise 2 check boundaries

Load the R package `validatetools`.

a) Look at the file `code/rules.yml`

```
rules:
- expr: age >= 18
  name: is_adult
  label: Not a child
- expr: if (job == TRUE) age <= 70
  name: retirement
  label: 'retirement'
- expr: if (income > 0) job == TRUE
  name: has_job
  label: 'Has a job'
- expr: income >= 0
  name: income
  label: 'income'
```

and load the rules into R variable `rules` with the help of validator

```
rules <- validator(.file = "rules.yml")
```

b) What are the allowed values for `age` and `income`?

c) Check this with `validatetools::detect_boundary_num`.

Excercise 3, simplify

Simplify:

a)

```
validator( if (income > 0) age >= 16
         , age < 12
        )
```

```
## Object of class 'validator' with 2 elements:
```

```
##  V1: income <= 0 | (age >= 16)
##  V2: age < 12
```

with `simplify_conditional`

b)

```
rules <- validator( r1 = if (income > 0) age >= 16
                  , r2 = age < 12
                  )
```

c)

```
rules <- validator( r1 = job %in% c("yes", "no")
                  , r2 = if (job == "yes") income > 0
                  , r3 = if (age < 16) income == 0
                  )
```

with Apply `simplify_rules(rules, job = "yes")`

d) Can you reproduce c) with the other simplifying functions?

Excercise 4, find the conflicting rules

a) Open the file "infeasible_rules.txt" (e.g. `file.edit("infeasible_rules.txt")`). Can you see which rules are in conflict?

b) Find which two rule(s) are causing the infeasibility in file "infeasible_rules.txt". Look into the help file of `validatetools`.

```
rules <- validator(.file = "infeasible_rules.txt")
is_infeasible(rules)
```

```
## [1] TRUE
```

```
# do your thing...
```