

Data Validation

Mark van der Loo and Edwin de Jonge

Statistics Netherlands Research & Development
Twitter: @markvdloo @edwindjonge

ENBES|EESW 2019



Try the code

03valid/check_validity.R



Data validation

Verify that data satisfy technical restrictions and does not contradict expert knowledge.

Examples of technical demands

- Number of records must equal 60
- Financial variables are numeric
- Records have a unique id
- Zipcode consists of 4 numbers followed by 2 letters

Examples of domain knowledge demands

- turnover is nonnegative
- turnover – costs = profit
- profit not larger than 60% of turnover
- average profit is larger than 0
- average profit differs less than 10% from last year's average



Data validation rules

A domain specific language to express demands.

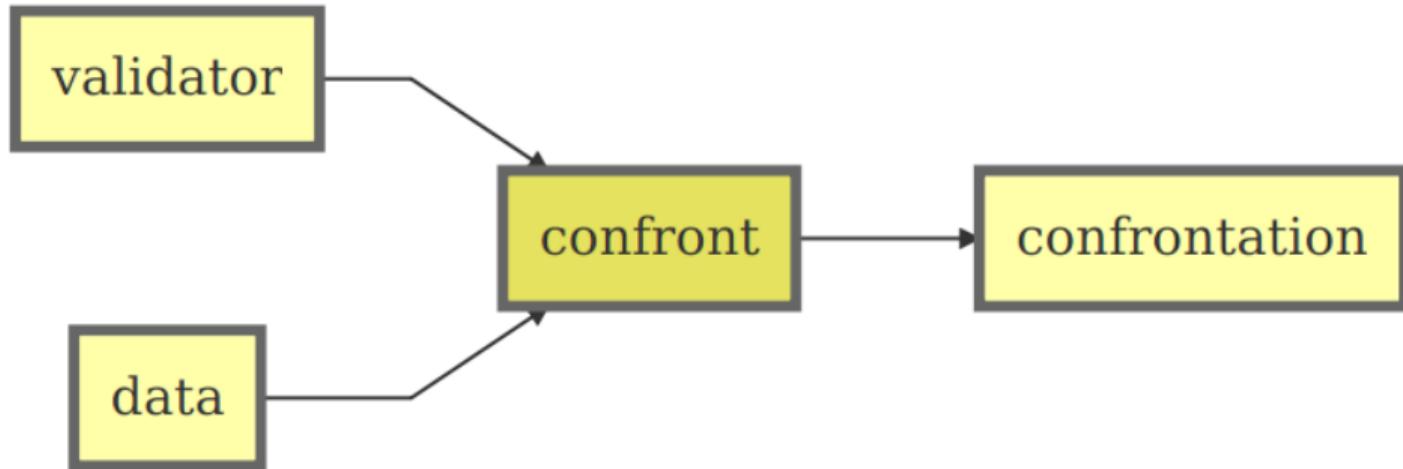
Why?

- Communicate data quality without ambiguities
- Make knowledge explicit and organize it
- Create custom data quality reports
- Reuse ruleset for data cleaning purposes

How?

```
library(validate)
companies <- read.csv("02input/input.csv",stringsAsFactors = FALSE)
rules      <- validator(.file="02input/rules.R")
result     <- confront(companies, rules)
```

Core concepts of the validate package



Comparing numbers

```
rules <- validator(turnover + costs == profit)
summary( confront(companies, rules) )
```

```
##      name items passes fails nNA error warning
## 1    V1      0       0     0   0  TRUE  FALSE
##                               expression
## 1 abs(turnover + costs - profit) < 1e-08
```



Data validation: informal definitions

Data validation

Check if a value, or combination of values is in a certain set of valid values or valid value combinations.

Data validation language in validate

Any R expression that results in a logical.



Expressions that are validation rules

Basic syntax

- Any type check: `is.numeric`, `is.character`,...
- Any comparison: `<`, `<=`, `==`, `identical`, `!=`, `%in%`, `>=`, `>`
- Logical operators `|`, `&`, `if`, `!`, `all`, `any`
- Pattern match: `grepl`

Sugar

Dot `."` stands for the whole data set:

```
nrow(.) >= 50      # at least 40 rows
"id" %in% names(.) # 'id' must be present
```

More, see `?syntax` or `vignette("introduction", package="validate")`



Challenges

1. Express the following restrictions on companies. Then confront and summary
 - Profit does not exceed 60% of turnover
 - turnover less costs equals profit
 - Average profit is larger than zero
 - correlation (corr) between total cost and staff exceeds 0.5
 - zipcode is 4 numbers followed by two upper case letters (you need to know regex)
2. Read the rules in rules.R. Then, confront, and summary.



More on validation

- Precise definition
- Classification of validation rules



Data Validation



Some examples from a survey amongst the ESS member states

- If a respondent has *income from other activities*, fields under *other activities* must be filled.
- *Yield per area* must be between 40 and 60 metric tons
- A person of age under 15 cannot *take part in an economic activity*
- The field *type of ownership* (of a building) may not be empty
- The *regional code* must be in the code list.
- The *current average price* divided by *last period's average price* must lie between 0.9 and 1.1.



Specification of allowed (valid) data

By extension

Marital status must be in

{never married, married, not married, divorced, widowed}

By intension

- Age is a *number* which is not negative and less than or equal to 120.
- (Age, Has_Job) is a pair from $\mathbb{R} \times \{\text{yes}, \text{no}\}$, satisfying the implication
 $Age < 15 \Rightarrow Has_Job = \text{no}$.



Questions

- Can we properly *define* the concept of data validation?
- If so, is it possible to *classify* validation activities?



Definition (European Statistical System)

Definition

Data Validation is an activity verifying whether or not a combination of values is a member of a set of acceptable combinations.

Methodology of Data Validation (ESS Handbook, 2016)



Combinations of values

Single variable; multiple variables

$\text{Age} \geq 0; \text{Age} < 15 \Rightarrow \text{Has_Job} = \text{no}$

Multiple entities

$\text{mean}(\text{Profit}) \geq 10$

Multiple times or domains

$0.9 < \text{mean}(\text{Profit}_{2018})/\text{mean}(\text{Profit}_{2017}) < 1.1$



Conclusion

Intuitively:

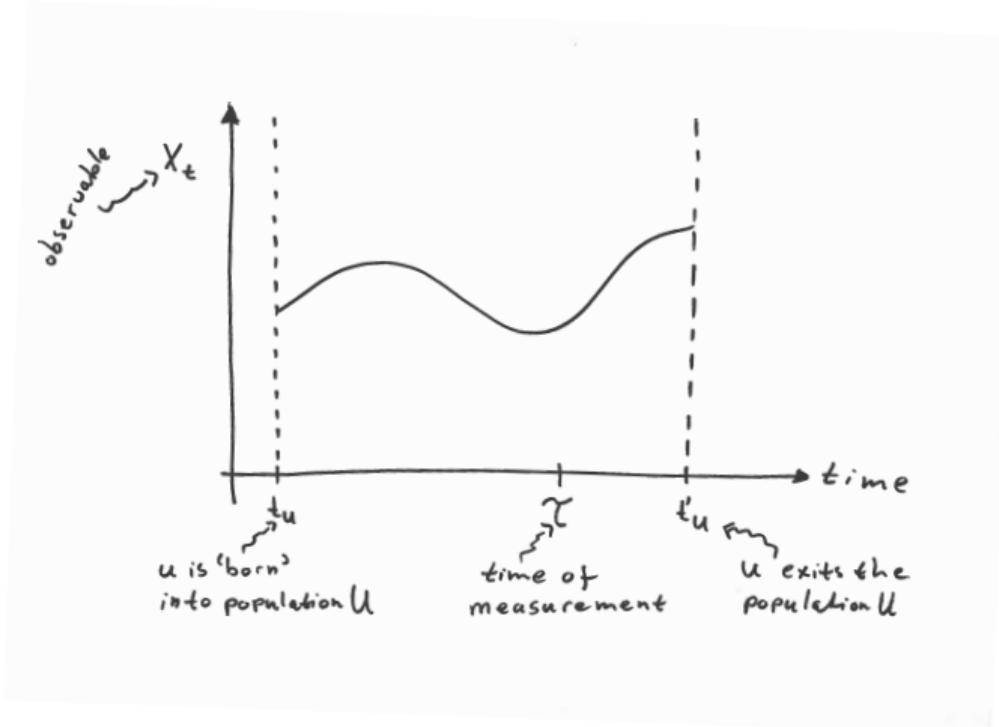
'Data validation is a function that accepts (some set of) value(s) and returns TRUE (valid) or FALSE (invalid).'

The catch

To make this precise we must define 'some set of values'.



What characterizes a data point?



What is a data point?

Definition

A *data point* consists of a pair (k, x) where

- x is a *value* (number, text, category, date/time, NA)
- k is a *key* (or list of keys) identifying at least:
 - population U
 - time of measurement τ (or: the measurement event)
 - element of the population u
 - property being measured X
- For formal reasons, we demand that there are only a finite number of possible keys k , coming from a set K .
- We say that x comes from a domain D .



What is a data set?

Definition

A *data set* S is a finite set of key-value pairs

$$S = \{(k_1, x_1), (k_2, x_2), \dots, (k_{|K|}, x_{|K|})\}$$

where all k_i are different.

Note

- The k_i are often referred to as *metadata*
- The x_i may be of different type
- Given a set of keys K and a domain D . The set of all data sets is denoted D^K .



Example

In 2017 we asked the Dutch company ‘Piet’s Bakery’ for its turnover and whether it owns the building it works in.

Domain D

Numbers or yes/no: $D = \mathbb{R} \cup \{\text{yes}, \text{no}\}$

Example data points ($k = [U, \tau, u, X], x$)

- ([Dutch Companies, 2017, Piet’s Bakery, *turnover*], 50.000)
- ([Dutch Companies, 2017, Piet’s Bakery, *owns_building*], no)



Quizz

In September 2018 we ask the two Dutch citizens **Alice** and **Bob**:

1. X : Do you have a job? (yes, no)
2. Y : What is your age? (under-aged, adult, retired)

Questions

1. Describe D
2. Give all values of k (this constitutes K)
3. How many data sets are possible?



Answers (1)

Each data point is either in $\{\text{yes}, \text{no}\}$ or in $\{\text{under-aged}, \text{adult}, \text{retired}\}$, so

$$\begin{aligned}D &= \{\text{yes}, \text{no}\} \cup \{\text{under-aged}, \text{adult}, \text{retired}\} \\&= \{\text{yes}, \text{no}, \text{under-aged}, \text{adult}, \text{retired}\}\end{aligned}$$



Answers (2)

- U : Dutch citizens (same for all data points)
- τ : 2017 (same for all data points)
- Values for k :
 - $[U, \tau, \text{Alice}, \text{job}]$
 - $[U, \tau, \text{Alice}, \text{age}]$
 - $[U, \tau, \text{Bob}, \text{job}]$
 - $[U, \tau, \text{Bob}, \text{age}]$



Number of data sets: unrestricted

- There are 4 unique keys in K
- For each key in K there are 5 options.
- Number of data sets: $5^4 = 625$.

Note

This includes cases where values are swapped (e.g. $age = no$ and $job = under-aged$)



Number of data sets: with restrictions

Restrictions

- $job \in \{\text{yes}, \text{no}\}$
- $age \in \{\text{under-aged}, \text{adult}, \text{retired}\}$
- $job = \text{yes} \Rightarrow age = \text{adult}$

Number of ways for (job, age) pairs to be valid equals 4:

	under-aged	adult	retired
yes	invalid	valid	invalid
no	valid	valid	valid

There are two such pairs in a data set so there are $4^2 = 16$ valid data sets.



What is data validation?

Definition

A *data validation function* is a surjective function v that accepts a data set in D^K and returns a value in $\{\text{FALSE}, \text{TRUE}\}$.

- If $v(S) = \text{FALSE}$ then S violates v
- If $v(S) = \text{TRUE}$ then S satisfies v
- Surjective means that if we compute v for every possible dataset S , both FALSE and TRUE have to occur at least once.

Note

Such a function is (almost) always stated as a *rule* stating a condition that data must satisfy.



Validation rule complexity

Observation

Depending on the rule, we may need to compare data points against

- A constant,
- Other data points, coming from other
 - variables,
 - measurement times,
 - statistical units,
 - populations.

Idea

Use the 'amount of extra information necessary' to classify the complexity of validation rules.



Classifying validation rules

- Recall the $U\tau uX$ notation
- A rule is labeled with a sequence of four characters $cccc$, where each character is either s (single) or m (multi).

Example

IF $age < 15$ **THEN** $job = \text{FALSE}$

- We see that
 - single population U
 - single measurement time τ
 - single statistical unit u
 - multiple (2) variables X
- Hence, the complexity class is $sssm$



Possible classes

- In principle there are $2^4 = 16$ classes
- However,
 - Given U , the possible u are known
 - Given U , the possible X are known
- This limits the classification to 10 possible options

$ssss$ $sssm$ $ssms$ $ssmm$ $smss$
 $smsm$ $smms$ $smmm$ $msmm$ $mmmm.$

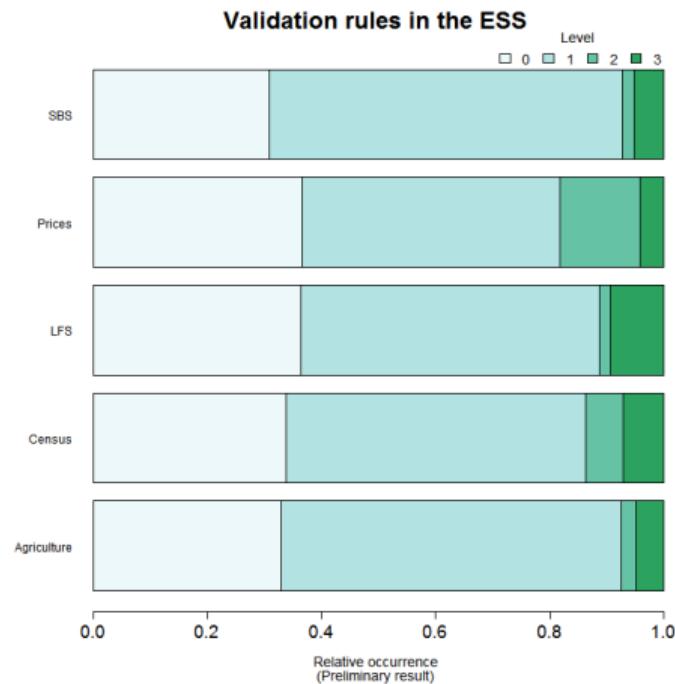


Validation rule classification

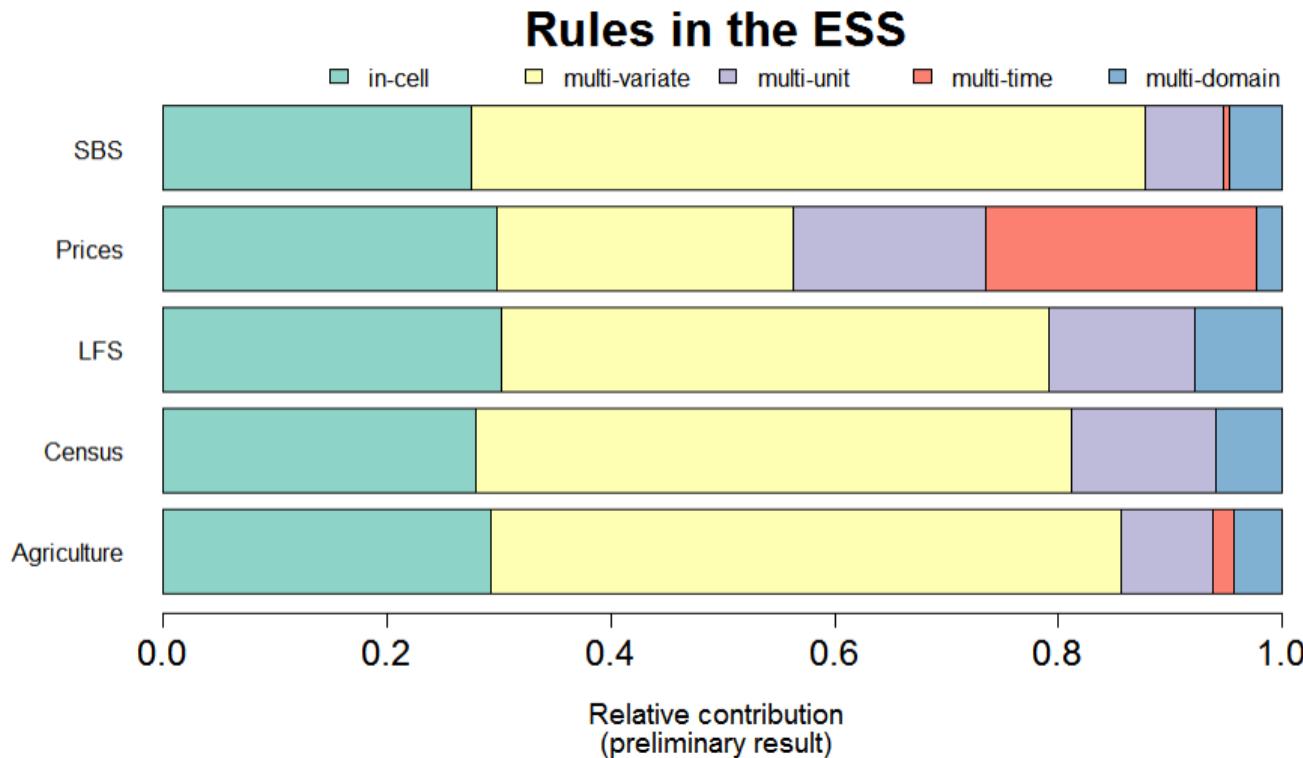
Validation level				
0	1	2	3	4
ssss	sssm	ssmm	smmm	mmmm
	ssms	smsm	msmm	
	smss	smms		



Validation rules in the ESS (1/3)

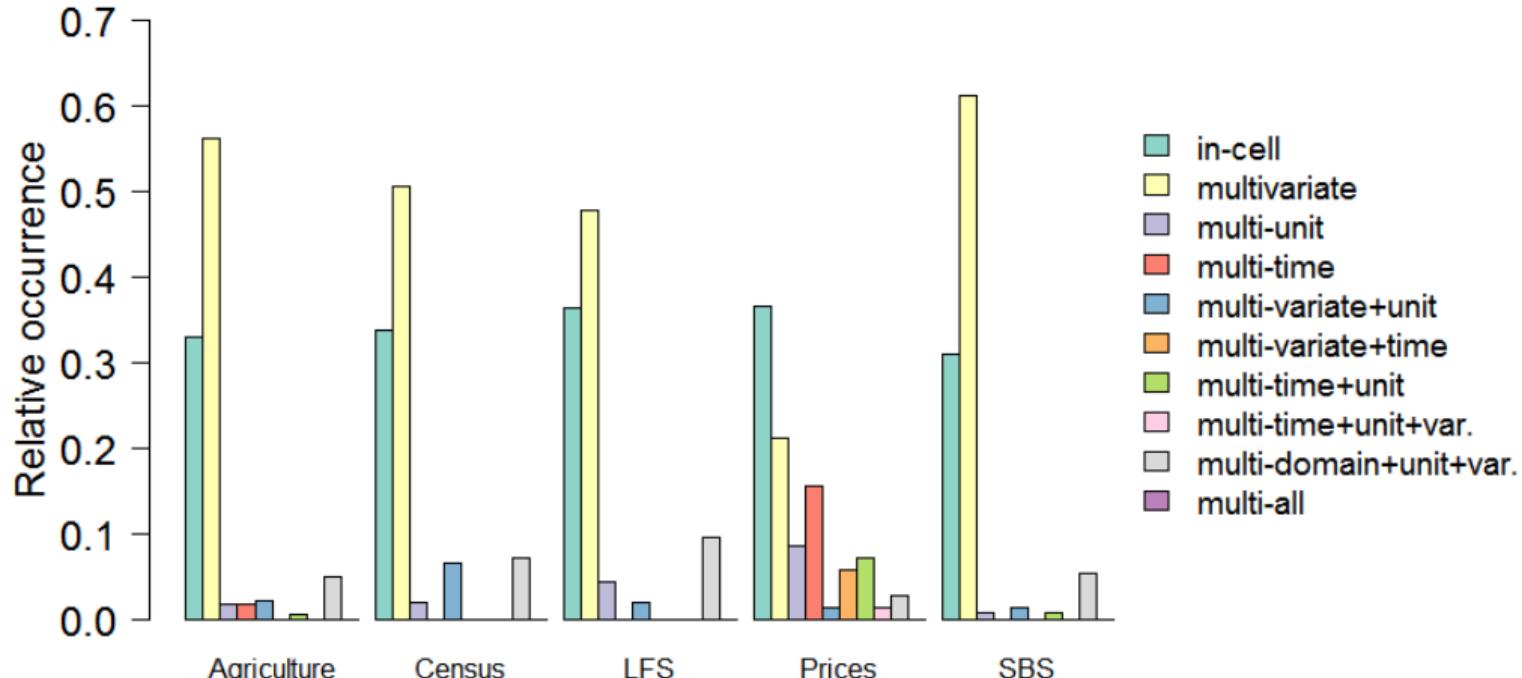


Validation rules in the ESS (2/3)



Validation rules in the ESS (3/3)

Validation rules in the ESS by type



Quizz (1)

What is the $U\tau uX$ single/multi classification of the following rule?

$$\text{mean}(\textit{price}) \geq 1$$



Quizz (2)

What is the $U\tau uX$ single/multi classification of the following rule?

$$\frac{\text{mean}(\text{price}_{2018})}{\text{mean}(\text{price}_{2017})} \leq 1.1$$



Quizz (3)

What is the $U\tau uX$ single/multi classification of the following rule?

$$\max \left(\frac{x}{\text{median}(X)}, \frac{\text{median}(X)}{x} \right) < 10$$



Quizz (4)

What is the $U\tau uX$ single/multi classification of the following rule?

$$\underbrace{COE + GOS + GMI + T_{P\&M} - S_{P\&M}}_{\text{GDP, Income approach}} = \underbrace{C + G + I + (X - M)}_{\text{GDP, expenditure approach}}$$

- COE : Compensation of employees
- GOS : Gross operating surplus
- GMI : Gross mixed income
- $T_{P\&M} - S_{P\&M}$: Taxes minus subsidies on production and import
- C : Consumption by households
- G : Government consumption & investment
- I : Gross private domestic investment
- $X - M$: Export minus Imports of goods and services

