Discover validation rules in data with ‘validatesuggest’

Olav ten Bosch, Edwin de Jonge, Mark van der Loo
Statistics Netherlands

UNECE Workshop on Statistical Data Editing (SDE), Oct. 2022

Contents

- Need and use of Data validation
- An ecosystem for (inter)national data validation
- A data-driven approach
- Implementation in ‘validatesuggest’
- Wrap up

Data validation: the problem

- ***Data quality*** in official statistics tends to vary over time
- Data, organisations, processes and systems change over time and may encounter ***unexpected*** data dynamics
- If not detected in time this may lead to costly ***recalculations***, ***retransmissions*** (data ping pong) or – worse – undetected ***errors***
- Problem grows with new, more ***volatile*** data sources
- Data needs to be ***validated*** before being used

Data validation: the (theoretical) solution

- ***Agree*** on rules between data producer and consumer
- ***ESS***: agree in statistical working groups
- ***Validate*** data against rules on both sides

But:

- How to define meaningful rules?
- How to maintain them?

ESS Validation principles:

1. *The sooner, the better*
2. *Trust but verify*
3. *Well-documented and appropriately communicated validation rules*
4. *Well-documented and appropriately communicated validation errors*
5. *Comply or explain*
6. *Good enough is the new perfect*

https://ec.europa.eu/eurostat/cros/content/principles_en

Validation rule ecosystem (1)

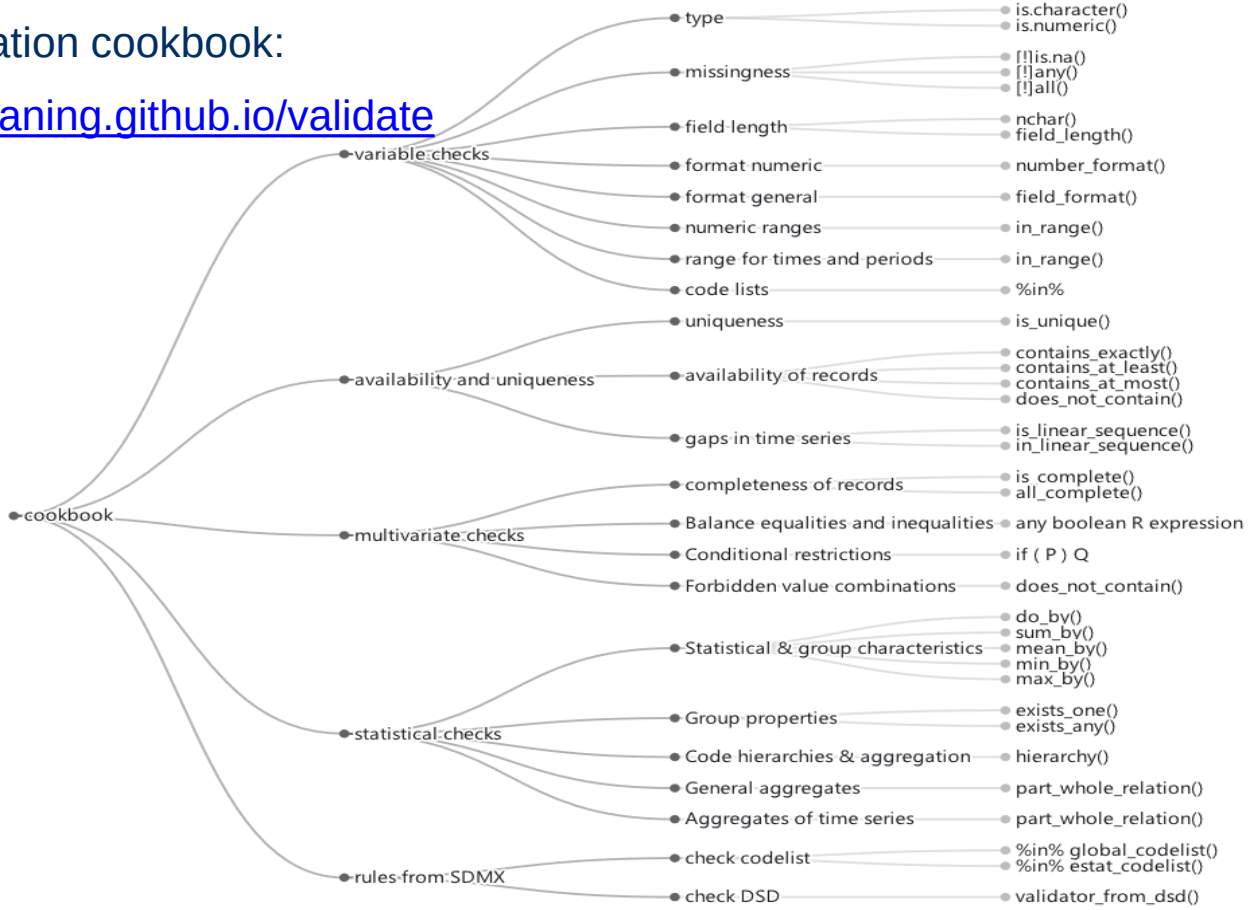
The 20 main types of validation rules in the ESS and their characteristics

Rule type	Mandatory	Default	Validation level						SDMX	Micro data	Severity level		
			0	1	2	3	4	5			E	W	I
(EVA) Envelope is Acceptable	X		X						X	X	X		
(FLF) File Format	X		X						X	X	X		
(FDD) Fields Delimiter	(X)	";"	X						X	X	X		
(DES) Decimals Separator	(X)	"."	X						X	X	X		
(FDT) Field Type	X		X	(X)					(X)	X	X		
(FDL) Field Length	X		X						X	X	X		
(FDM) Field is Mandatory or empty			X	(X)					(X)	X	X	(X)	
(COV) Codes are Valid	(X)			X					(X)	X	X	(X)	
(RWD) Records are Without Duplicates	(X)	Key		X					X	(X)	X		
(REP) Records Expected are Provided				X	X					X	X	(X)	
(RNR) Records' Number is in a Range	X	>=1		X	(X)					X	X	(X)	(X)
(COC) Codes are Consistent				X	X				(X)	X	X	(X)	
(VIR) Values are in Range		>=0		X	X				(X)	X	X	(X)	(X)
(VCO) Values are Consistent				X	X	X	X	X		X	X	(X)	(X)
(VAD) Values for Aggregates are consistent with Details	(X)	=		X	X						X	(X)	(X)
(VNO) Values are Not Outliers				X	X						(X)	X	(X)
(VSA) Values for Seasonally Adjusted data are plausible				X	X						X	(X)	(X)
(RRL) Records Revised are Limited					X					(X)	(X)	X	(X)
(VRT) Values are Revised within a Tolerance level					X					(X)	(X)	X	(X)
(VMP) Values for Mirror data are Plausible						X					(X)	X	(X)

Validation rule ecosystem (2)

The data validation cookbook:

https://data-cleaning.github.io/validate



How to start a set of validation rules?

Traditionally extracting rules from:

- Domain experts: ***implicit*** domain knowledge
- Production systems: ***solidified*** implicit knowledge (indirectly from domain experts)

However:

- Can we expand and complement current rule sets?
- What if we have a new data source?

Idea: use the data as a starting point

A data-driven approach to data validation (1)

- **Infer** rules from the data: data ‘suggests’ rules
- Start from a ***clean*** or slightly dirty dataset
- More data ~=> more knowledge ~=> better rules
- Suggest rules ***typical*** for official statistics, taken from existing validation rule ecosystem (ESS and cookbook)
- Suggest rules that are ***human-readable***
- Suggest rules to the rule developer / maintainer, to be ***interpreted*** before used in production

A data-driven approach to data validation (2)

- For timeseries data rules may depend on the ***time period*** in data (e.g. growth rate)
- Rules suggestions should allow for ***tolerances***
- Advantage over ML approaches: rules are ***explicit***, ***understandable***, ***explainable*** and use existing rule ***terminologies***

The R-package validate

Manually defined rules

```
# Range limits:
Age >= 0
Age <= 120
Working_hours >= 0
Working_hours <= 100

# Some checks between variables:
if (Married > 0) Age > 18
if (Working_hours > 0) Employed > 0

#Such a rule depends on country legislation:
if (Age > 65) Working_hours = 0

# ID must be unique
any(duplicated(ID)) == FALSE
```

Data

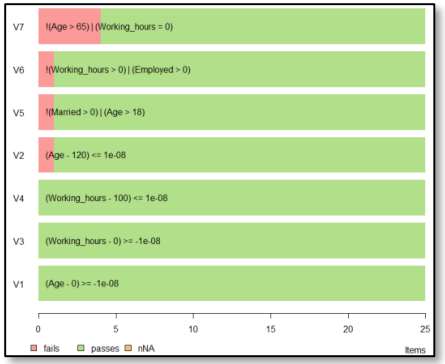
	ID	Age	Married	Employed	Working_hours
1	1	36	FALSE	TRUE	40
2	2	40	TRUE	TRUE	40
3	3	25	FALSE	FALSE	0
4	4	31	FALSE	TRUE	20
5	5	62	TRUE	TRUE	43
6	6	55	TRUE	TRUE	41

confront

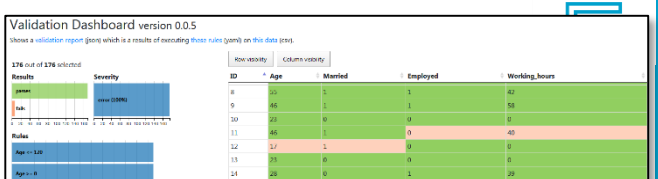
Summary

```
> summary(validation)
  name items passes fails nNA error warning                                expression
1   V1    25     25     0   0 FALSE   FALSE                          (Age - 0) >= -1e-08
2   V2    25     24     1   0 FALSE   FALSE                        (Age - 120) <= 1e-08
3   V3    25     25     0   0 FALSE   FALSE                (Working_hours - 0) >= -1e-08
4   V4    25     25     0   0 FALSE   FALSE              (Working_hours - 100) <= 1e-08
5   V5    25     24     1   0 FALSE   FALSE                 !(Married > 0) | (Age > 18)
6   V6    25     24     1   0 FALSE   FALSE !(Working_hours > 0) | (Employed > 0)
7   V7    25     21     4   0 FALSE   FALSE     !(Age > 65) | (working_hours = 0)
8   V8     1      0     1   0 FALSE   FALSE          any(duplicated(ID)) == FALSE
```

Per rule



Dashboard: data & results



The R-package validatesuggest

https://github.com/data-cleaning/validatesuggest

(a) Positivity checks
(b) Range checks
(c) Checks on na: whether a variable may contain nas
(d) Checks on uniqueness
(e) Type checks
(f) Ratio checks:
(g) Discovery of conditional rules

Manually defined rules

```
# Range limits:
Age >= 0
Age <= 120
Working_hours >= 0
Working_hours <= 100

# Some checks between variables:
if (Married > 0) Age > 18
if (Working_hours > 0) Employed > 0

#Such a rule depends on country legislation:
if (Age > 65) Working_hours = 0

# ID must be unique
any(duplicated(ID)) == FALSE
```

Compare and improve

Data

	ID	Age	Married	Employed	Working_hours
1	1	36	FALSE	TRUE	40
2	2	40	TRUE	TRUE	40
3	3	25	FALSE	FALSE	0
4	4	31	FALSE	TRUE	20
5	5	62	TRUE	TRUE	43

suggest_all()

Suggested rules

```
# check for positivity
ID >= 0
Age >= 0
Working_hours >= 0

# check the range of variables
in_range(ID, 1, 25)
in_range(Age, 17, 125)
Married %in% c(TRUE, FALSE)
Employed %in% c(TRUE, FALSE)
in_range(Working_hours, 0, 58)

# check the type of variables
is.complete(ID)
is.complete(Age)
is.complete(Married)
is.complete(Employed)
is.complete(Working_hours)
```

Ratio checks and conditional rules

Ratio checks:

- Only variables that are (enough) correlated are considered (threshold)

Conditional rules:

- Unsupervised ML: association rules and CART
- Checks co-occurrence frequency of values
- Direction of causality derived from occurrence of other values

Retailer dataset (fictitious)

ize	incl.prob	staff	turnover	other.rev	total.rev	staff.costs	total.costs	profit	vat
c0	0.02	75	NA	NA	1130	NA	18915	20045	NA
c3	0.14	9	1607	NA	1607	131	1544	63	NA
c3	0.14	NA	6886	-33	6919	324	6493	426	NA
c3	0.14	NA	3861	13	3874	290	3600	274	NA
c3	0.14	NA	NA	37	5602	314	5530	72	NA
c0	0.02	1	25	NA	25	NA	22	3	NA
c3	0.14	5	NA	NA	1335	135	136	1	1346

```
suggest_ratio_check(retailers)
#> Object of class 'validator' with 10 elements:
#>  RC1 : turnover >= 0 * total.rev
#>  RC2 : turnover <= 9.07 * total.rev
#>  RC3 : other.rev >= -0.1 * staff.costs
#>  RC4 : other.rev <= 34.55 * staff.costs
#>  RC5 : other.rev >= -0.01 * total.costs
#>  RC6 : other.rev <= 1.27 * total.costs
#>  RC7 : staff.costs >= 0 * total.costs
#>  RC8 : staff.costs <= 0.99 * total.costs
#>  RC9 : other.rev >= -2.8 * profit
#>  RC10: other.rev <= 4.72 * profit

write_cond_rule(retailers)
#>
#> # Conditional checks
#> if (staff > 0) other.rev > 0
#> if (other.rev <= 0) profit > 0
#> if (other.rev <= 0) vat <= 0
```

Wrap-up

- Rule disovery and maintenance is a problem
- A data-driven approach to validation rule management
- Uses existing (inter)national rule ecosystems
- Supports rule developer with explainable and interpretable rules derived from data
- ‘Validatesuggest’: positivity checks, range checks, NA, uniqueness, type checks, ratio checks, conditional rules
- Concept can be extended to support more rules

  => Demo……

Questions, ideas, suggestions

?

Olav ten Bosch	o.tenbosch@cbs.nl	@olavtenbosch
Edwin de Jonge	e.dejonge@cbs.nl	@edwindjonge
Mark van der Loo	mpj.vanderloo@cbs.nl	@markvdloo

and keep an eye on:

awesomeofficialstatistics.org