

Package: validatesuggest (via r-universe)

July 3, 2024

Title Generate Suggestions for Validation Rules

Version 0.3.2

Description Generate suggestions for validation rules from a reference data set, which can be used as a starting point for domain specific rules to be checked with package 'validate'.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.3

Imports validate, whisker, rpart

URL <https://github.com/data-cleaning/validatesuggest>

BugReports <https://github.com/data-cleaning/validatesuggest/issues>

Depends R (>= 2.10)

Suggests knitr, rmarkdown, tinytest

VignetteBuilder knitr

Repository <https://data-cleaning.r-universe.dev>

RemoteUrl <https://github.com/data-cleaning/validatesuggest>

RemoteRef HEAD

RemoteSha ad6bc2affbe406af4f59932e892ea0eb7debb9f6

Contents

car_owner	2
suggest_rules	3
task2	4
write_cond_rule	5
write_domain_check	6
write_na_check	6

write_pos_check	7
write_range_check	8
write_ratio_check	9
write_type_check	10
write_unique_check	10

Index	12
--------------	-----------

car_owner	<i>Car owners data set (fictitious).</i>
-----------	--

Description

A constructed data set useful for detecting conditinal dependencies.

Usage

```
car_owner
```

Format

A data frame with 200 rows and 4 variables. Each row is a person with:

age age of person

driver_license has a driver license, only persons older then 17 can have a license in this data set

income monthly income

owns_car only persons with a drivers license , and a monthly income > 1500 can own a car

car_color NA when there is no car

Examples

```
data("car_owner")

rules <- suggest_cond_rule(car_owner)
rules$rules
```

suggest_rules	<i>Suggest rules</i>
---------------	----------------------

Description

Suggests rules using the various suggestion checks. Use the more specific suggest functions for more control.

Usage

```
suggest_rules(  
  d,  
  vars = names(d),  
  domain_check = TRUE,  
  range_check = TRUE,  
  pos_check = TRUE,  
  type_check = TRUE,  
  na_check = TRUE,  
  unique_check = TRUE,  
  ratio_check = TRUE,  
  conditional_rule = TRUE  
)
```

```
suggest_all(  
  d,  
  vars = names(d),  
  domain_check = TRUE,  
  range_check = TRUE,  
  pos_check = TRUE,  
  type_check = TRUE,  
  na_check = TRUE,  
  unique_check = TRUE,  
  ratio_check = TRUE,  
  conditional_rule = TRUE  
)
```

```
write_all_suggestions(  
  d,  
  vars = names(d),  
  file = stdout(),  
  domain_check = TRUE,  
  range_check = TRUE,  
  type_check = TRUE,  
  pos_check = TRUE,  
  na_check = TRUE,  
  unique_check = TRUE,  
  ratio_check = TRUE,
```

```

    conditional_rule = TRUE
  )

```

Arguments

<code>d</code>	data.frame, used to generate the checks
<code>vars</code>	character optionally the subset of variables to be used.
<code>domain_check</code>	if TRUE include domain_check
<code>range_check</code>	if TRUE include range_check
<code>pos_check</code>	if TRUE include pos_check
<code>type_check</code>	if TRUE include type_check
<code>na_check</code>	if TRUE include na_check
<code>unique_check</code>	if TRUE include unique_check
<code>ratio_check</code>	if TRUE include ratio_check
<code>conditional_rule</code>	if TRUE include cond_rule
<code>file</code>	file to which the checks will be written to.

Value

returns `validate::validator()` object with the suggested rules. `write_all_suggestions` write the rules to file and returns invisibly a named list of ranges for each variable.

task2

task2 dataset

Description

Fictitious test data set from European (ESSnet) project on validation 2017.

Usage

```
task2
```

Format

ID ID
Age Age of person
Married Marital status
Employed Employed or not
Working_hours Working hours

References

European (ESSnet) project on validation 2017

write_cond_rule	<i>Suggest a conditional rule</i>
-----------------	-----------------------------------

Description

Suggest a conditional rule based on a association rule. This functions derives conditional rules based on the non-existence of combinations of categories in pairs of variables. For each numerical variable a logical variable is derived that tests for positivity. It generates IF THEN rules based on two variables.

Usage

```
write_cond_rule(d, vars = names(d), file = stdout())  
suggest_cond_rule(d, vars = names(d))
```

Arguments

d	data.frame, used to generate the checks
vars	character optionally the subset of variables to be used.
file	file to which the checks will be written to.

Value

suggest_cond_rule returns `validate::validator()` object with the suggested rules. write_cond_rule returns invisibly a named list of ranges for each variable.

Examples

```
data(retailers, package="validate")  
  
# will generate check for all columns in retailers that are  
# complete.  
suggest_na_check(retailers)  
data("car_owner")  
  
rules <- suggest_cond_rule(car_owner)  
rules$rules
```

write_domain_check *Suggest a range check*

Description

Suggest a range check

Usage

```
write_domain_check(d, vars = names(d), only_positive = TRUE, file = stdout())
```

```
suggest_domain_check(d, vars = names(d), only_positive = TRUE)
```

Arguments

`d` `data.frame`, used to generate the checks

`vars` character optionally the subset of variables to be used.

`only_positive` if TRUE only numerical values for positive values are included

`file` file to which the checks will be written to.

Value

`suggest_domain_check` returns `validate::validator()` object with the suggested rules. `write_domain_check` returns invisibly a named list of checks for each variable.

Examples

```
data(SBS2000, package="validate")

suggest_range_check(SBS2000)

# checks the ranges of each variable
suggest_range_check(SBS2000[-1], min=TRUE, max=TRUE)

# checks the ranges of each variable
suggest_range_check(SBS2000, vars=c("turnover", "other.rev"), min=FALSE, max=TRUE)
```

write_na_check *Suggest a check for completeness.*

Description

Suggest a check for completeness.

Usage

```
write_na_check(d, vars = names(d), file = stdout())

suggest_na_check(d, vars = names(d))
```

Arguments

`d` data.frame, used to generate the checks

`vars` character optionally the subset of variables to be used.

`file` file to which the checks will be written to.

Value

`suggest_na_check` returns `validate::validator()` object with the suggested rules. `write_na_check` write the rules to file and returns invisibly a named list of ranges for each variable.

Examples

```
data(retailers, package="validate")

# will generate check for all columns in retailers that are
# complete.
suggest_na_check(retailers)
```

write_pos_check	<i>Suggest a range check</i>
-----------------	------------------------------

Description

Suggest a range check

Usage

```
write_pos_check(d, vars = names(d), only_positive = TRUE, file = stdout())

suggest_pos_check(d, vars = names(d), only_positive = TRUE)
```

Arguments

`d` data.frame, used to generate the checks

`vars` character optionally the subset of variables to be used.

`only_positive` if TRUE only numerical values for positive values are included

`file` file to which the checks will be written to.

Value

suggest_pos_check returns `validate::validator()` object with the suggested rules. `write_pos_check` write the rules to file and returns invisibly a named list of checks for each variable.

Examples

```
data(SBS2000, package="validate")

suggest_range_check(SBS2000)

# checks the ranges of each variable
suggest_range_check(SBS2000[-1], min=TRUE, max=TRUE)

# checks the ranges of each variable
suggest_range_check(SBS2000, vars=c("turnover", "other.rev"), min=FALSE, max=TRUE)
```

write_range_check	<i>Suggest a range check</i>
-------------------	------------------------------

Description

Suggest a range check

Usage

```
write_range_check(d, vars = names(d), min = TRUE, max = FALSE, file = stdout())

suggest_range_check(d, vars = names(d), min = TRUE, max = FALSE)
```

Arguments

d	data.frame, used to generate the checks
vars	character optionally the subset of variables to be used.
min	TRUE or FALSE, should the minimum value be checked?
max	TRUE or FALSE, should the maximum value be checked?
file	file to which the checks will be written to.

Value

suggest_range_check returns `validate::validator()` object with the suggested rules. `write_range_check` write the rules to file and returns invisibly a named list of ranges for each variable.

Examples

```

data(SBS2000, package="validate")

suggest_range_check(SBS2000)

# checks the ranges of each variable
suggest_range_check(SBS2000[-1], min=TRUE, max=TRUE)

# checks the ranges of each variable
suggest_range_check(SBS2000, vars=c("turnover", "other.rev"), min=FALSE, max=TRUE)

```

write_ratio_check	<i>Suggest ratio checks</i>
-------------------	-----------------------------

Description

Suggest ratio checks

Usage

```

write_ratio_check(
  d,
  vars = names(d),
  file = stdout(),
  lin_cor = 0.95,
  digits = 2
)

suggest_ratio_check(d, vars = names(d), lin_cor = 0.95, digits = 2)

```

Arguments

d	data.frame, used to generate the checks
vars	character optionally the subset of variables to be used.
file	file to which the checks will be written to.
lin_cor	threshold for abs correlation to be included (details)
digits	number of digits for rounding

Value

suggest_ratio_check returns `validate::validator()` object with the suggested rules. write_ratio_check write the rules to file and returns invisibly a named list of check for each variable.

Examples

```
data(SBS2000, package="validate")

# generates upper and lower checks for the
# ratio of two variables if their correlation is
# bigger than `lin_cor`
suggest_ratio_check(SBS2000, lin_cor=0.98)
```

write_type_check	<i>suggest type check</i>
------------------	---------------------------

Description

suggest type check

Usage

```
write_type_check(d, vars = names(d), file = stdout())

suggest_type_check(d, vars = names(d))
```

Arguments

d	data.frame, used to generate the checks
vars	character optionally the subset of variables to be used.
file	file to which the checks will be written to.

Value

suggest_type_check returns `validate::validator()` object with the suggested rules. write_type_check write the rules to file and returns invisibly a named list of types for each variable.

write_unique_check	<i>Suggest range checks</i>
--------------------	-----------------------------

Description

Suggest range checks

Usage

```
write_unique_check(d, vars = names(d), file = stdout(), fraction = 0.95)

suggest_unique_check(d, vars = names(d), fraction = 0.95)
```

Arguments

<code>d</code>	data.frame, used to generate the checks
<code>vars</code>	character optionally the subset of variables to be used.
<code>file</code>	file to which the checks will be written to.
<code>fraction</code>	if values in a column > fraction unique, the check will be generated.

Value

`suggest_unique_check` returns `validate::validator()` object with the suggested rules. `write_unique_check` write the rules to file and returns invisibly a named list of checks for each variable.

Index

* datasets

car_owner, 2

task2, 4

car_owner, 2

suggest_all (suggest_rules), 3

suggest_cond_rule (write_cond_rule), 5

suggest_domain_check

(write_domain_check), 6

suggest_na_check (write_na_check), 6

suggest_pos_check (write_pos_check), 7

suggest_range_check

(write_range_check), 8

suggest_ratio_check

(write_ratio_check), 9

suggest_rules, 3

suggest_type_check (write_type_check),

10

suggest_unique_check

(write_unique_check), 10

task2, 4

validate::validator(), 4–11

write_all_suggestions (suggest_rules), 3

write_cond_rule, 5

write_domain_check, 6

write_na_check, 6

write_pos_check, 7

write_range_check, 8

write_ratio_check, 9

write_type_check, 10

write_unique_check, 10