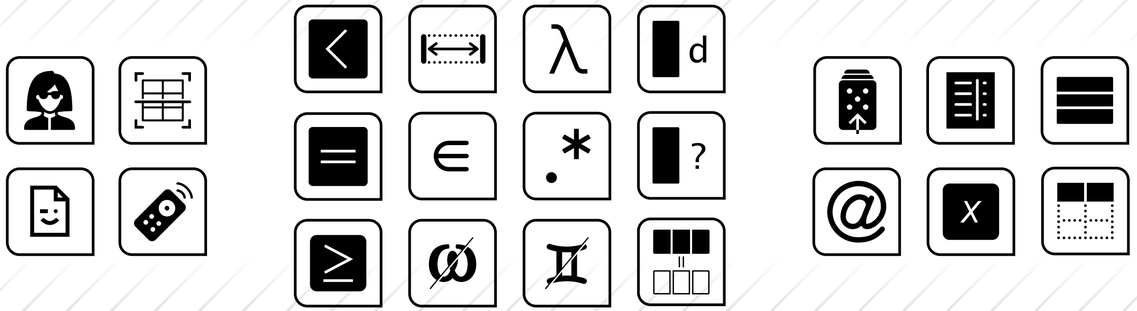
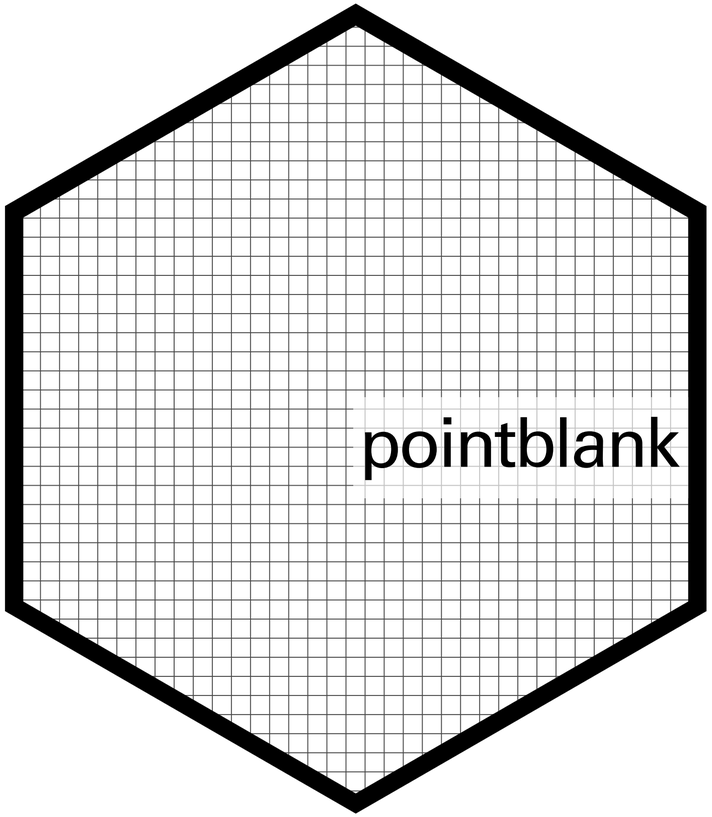
The **pointblank** R Package





rich-iannone

@riannone

rich@rstudio.com

pointblank

It’s a package for validating data tables.

Because, more often than not, they are not free of errors.

Some Design Goals of pointblank

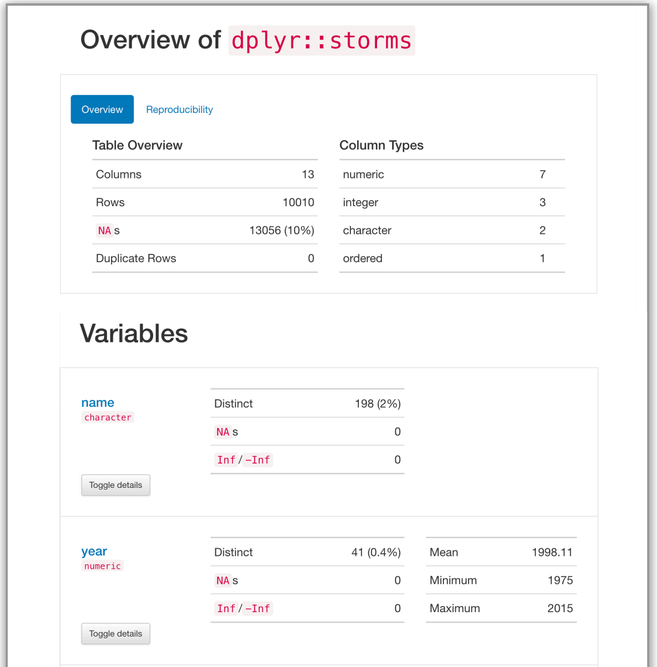
Make it work with all sorts of different tables, like **local tables** and **database tables**.

`data.frame` `tbl_dbi*`

`tbl_df` `tbl_spark`

*tested with: **MySQL**, **SQLite**, and **PostgreSQL**.

Provide a way to **understand** new datasets.



Overview of `dplyr::storms`

Overview | Reproducibility

Table Overview	
Columns	13
Rows	10010
NAs	13056 (10%)
Duplicate Rows	0

Column Types	
numeric	7
integer	3
character	2
ordered	1

Variables

name `character`

Distinct	198 (2%)
NAs	0
Inf / -Inf	0

Toggle details

year `numeric`

Distinct	41 (0.4%)	Mean	1998.11
NAs	0	Minimum	1975
Inf / -Inf	0	Maximum	2015

Toggle details

Create a toolset that can accommodate a lot of different **data validation workflows**.

1. DATA QUALITY
2. ETL or ANALYSIS SCRIPT
3. UNIT TESTING
4. CONDITIONAL CODE
5. TABLE SCAN
6. RMarkdown VALIDATION

THERE'S MORE

Keep trying to **make it easy** to use the package, with clear docs and examples.

Have reporting outputs be really nice to look at and **useful to everyone** in an organization.

Have reporting outputs translated to multiple **spoken languages**.

EN ▪ FR ▪ DE ▪ IT ▪ ES

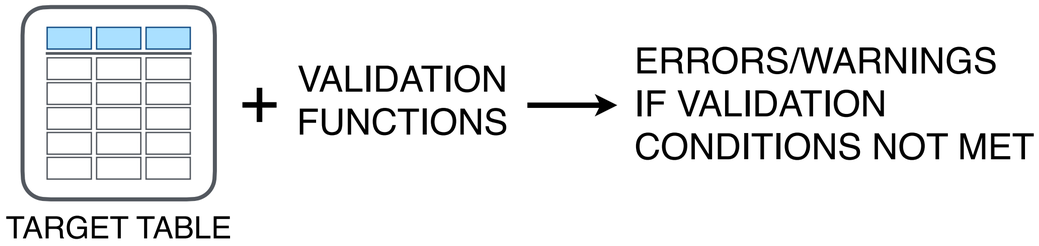
→ and more to come!

Useful Workflows in pointblank

MAIN WORKFLOWS

1 DATA QUALITY

2 ETL or ANALYSIS SCRIPT



SECONDARY WORKFLOWS

3 UNIT TESTING

You can validate your data tables in your unit tests. It's just like **testthat** actually. *Great* for packages.

4 CONDITIONAL CODE

Get `TRUE` or `FALSE` based on a data validation. This can be useful **in R code**.

The pointblank Data Quality Workflow ①

The **agent** is given the **target table**…

```
create_agent()
```

The **agent** is an integral part of the **data quality workflow**.

The pointblank Data Quality Workflow ①

The **agent** is given the **target table**…

`actions`

`end_fns`

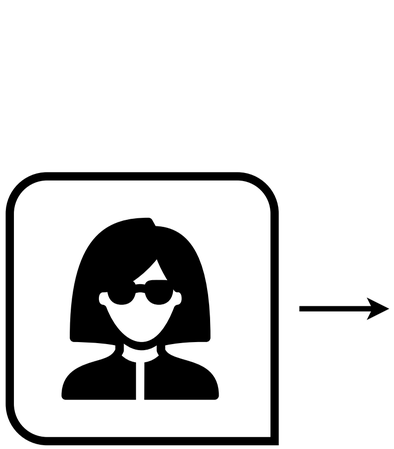
`lang`

…and some directives on interrogation.

`create_agent()`

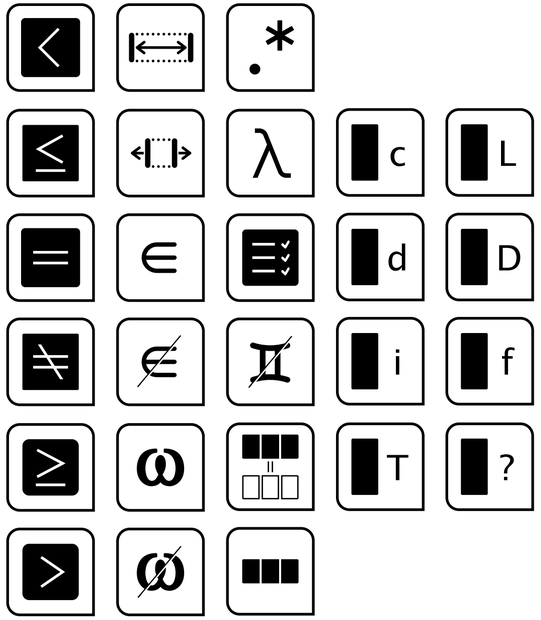
The **agent** is an integral part of the **data quality workflow**.

The pointblank Data Quality Workflow ①



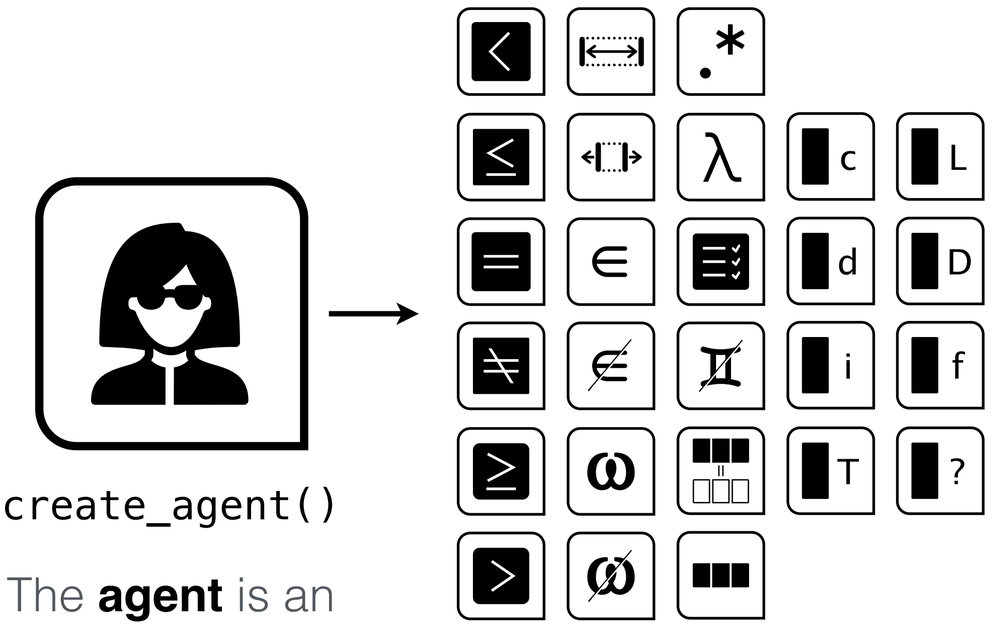
`create_agent()`

The **agent** is an integral part of the **data quality workflow**.



Use functions to generate a **validation plan**.

The pointblank Data Quality Workflow ①



`create_agent()`

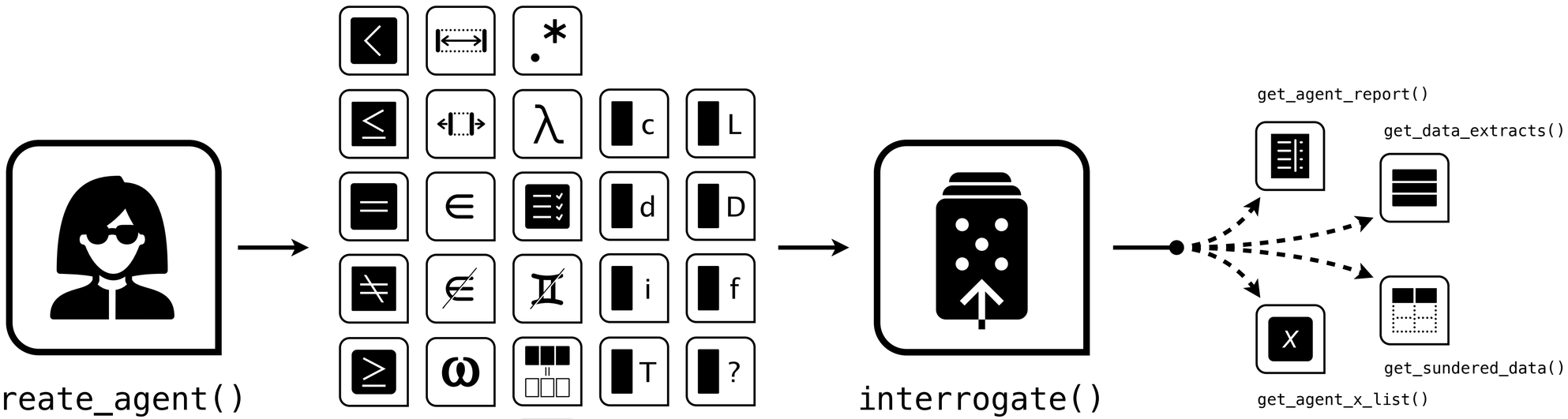
The **agent** is an integral part of the **data quality workflow**.

Use functions to generate a **validation plan**.

`interrogate()`

The **agent** performs the **interrogation**. Gathers **intel**.

The pointblank Data Quality Workflow ①



The **agent** is an integral part of the **data quality workflow**.

Use functions to generate a **validation plan**.

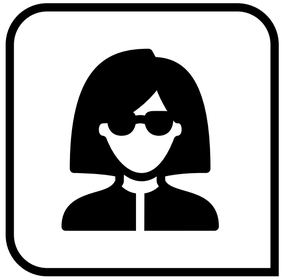
The **agent** performs the **interrogation**. Gathers **intel**.

Use **post-interrogation** functions for reporting, root-cause analysis, etc.

The pointblank Data Quality Workflow ①

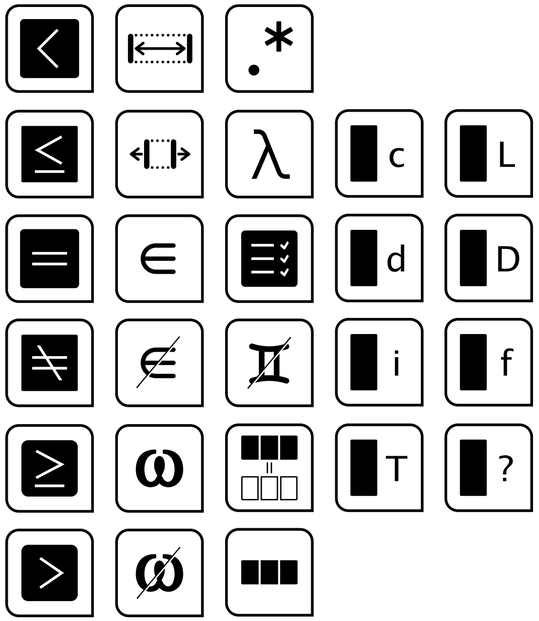
Understand your data more with a **table scan**.

`scan_data()`



`create_agent()`

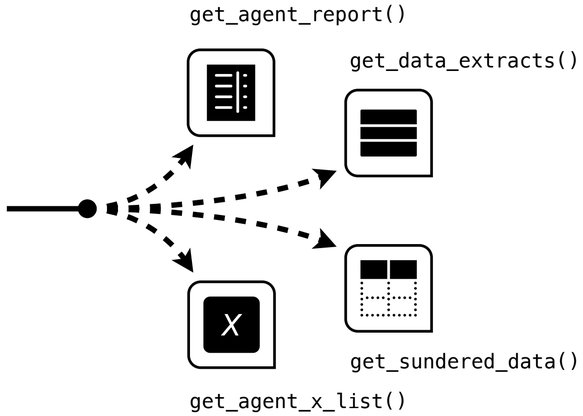
The **agent** is an integral part of the **data quality workflow**.



Use functions to generate a **validation plan**.

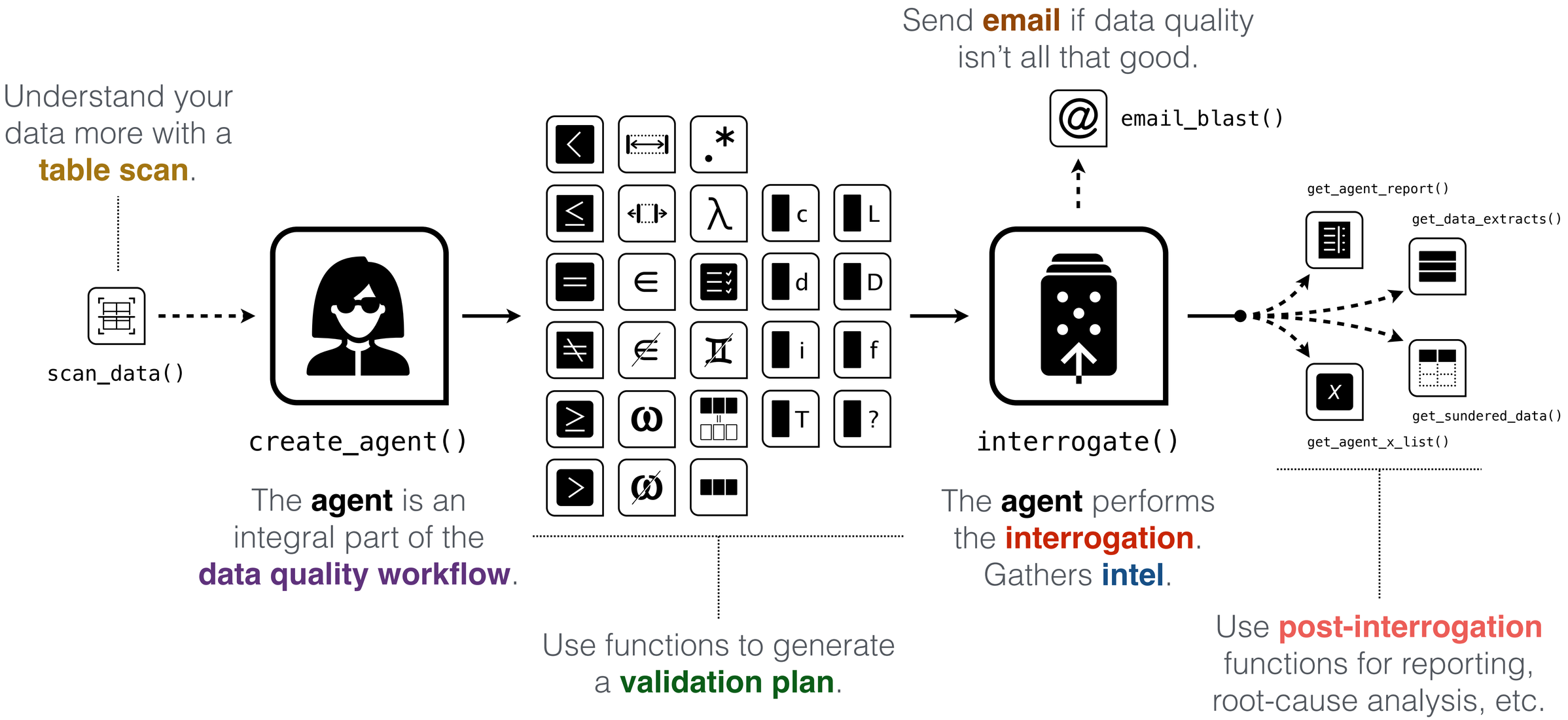
`interrogate()`

The **agent** performs the **interrogation**. Gathers **intel**.



Use **post-interrogation** functions for reporting, root-cause analysis, etc.

The pointblank Data Quality Workflow ①



Creating a Validation Plan

a	b	c
yko2	1	23.1
lju7	0	16.3
qib0	1	21.2
sd33	1	24.9
NA	2	NA

Let's start with a simple table

5 rows, 3 columns

Creating a Validation Plan

a	b	c
yko2	1	23.1
lju7	0	16.3
qib0	1	21.2
sd33	1	24.9
NA	2	NA

simple table

5 rows, 3 columns

1. All values in **c** should be greater than `15`
2. All values in **b** should be either `0` or `1`
3. All values in **a** should fit a pattern of three lowercase letters and a digit
4. Values in **c** must be `≥20` if **b** is `1`; if **b** is `0` then values in **c** must be `<20`
5. Columns **a**, **b**, and **c** should not have any missing values.

validation plan

5 steps

Creating a Validation Plan

1. All values in **c** should be greater than `15`
2. All values in **b** should be either `0` or `1`
3. All values in **a** should fit a pattern of three lowercase letters and a digit
4. Values in **c** must be `≥20` if **b** is `1`; if **b** is `0` then values in **c** must be `<20`
5. Columns **a**, **b**, and **c** should not have any missing values.

validation plan

5 steps

- ≥ `col_vals_gte()`
- ∈ `col_vals_in_set()`
- .* `col_vals_regex()`
- λ `col_vals_expr() + case_when()`
- ∅ `col_vals_not_null()`

validation functions

5 `col_vals_*()` functions

Creating a Validation Plan

1. All values in **c** should be greater than `15`
2. All values in **b** should be either `0` or `1`
3. All values in **a** should fit a pattern of three lowercase letters and a digit
4. Values in **c** must be `≥20` if **b** is `1`; if **b** is `0` then values in **c** must be `<20`
5. Columns **a**, **b**, and **c** should not have any missing values.

validation plan

5 steps

```
col_vals_gte(c, 15)
```

```
col_vals_in_set(b, c(0, 1))
```

```
col_vals_regex(a, “[a-z]{3}[0-9]")
```

```
col_vals_expr(~case_when(
  b == 1 ~ c >= 20,
  b == 0 ~ c <  20))
```

```
col_vals_not_null(vars(a, b, c))
```

validation functions

5 `col_vals_*()` functions

Interrogating the Table

a	b	c
yko2	1	23.1
lju7	0	16.3
qib0	1	21.2
sd33	1	24.9
NA	2	NA

simple table

5 rows, 3 columns

1 ≥ `col_vals_gte(c, 15)`

2 ∈ `col_vals_in_set(b, c(0, 1))`

3 .* `col_vals_regex(a, "[a-z]{3}[0-9]")`

4 λ

```
col_vals_expr(~case_when(
  b == 1 ~ c >= 20,
  b == 0 ~ c <  20))
```

5 ∅ `col_vals_not_null(vars(a, b, c))`

validation functions

5 `col_vals_*()` functions

Interrogating the Table

a	b	c
yko2	1	23.1
lju7	0	16.3
qib0	1	21.2
sd33	1	24.9
NA	2	NA

INTERROGATION

test units

1 `col_vals_gte(c, 15)`

REPORT

UNITS	PASS	FAIL
5	4 0.8	1 0.2

Interrogating the Table

a	b		c
yko2	1	☐	23.1
lju7	0	☐	16.3
qib0	1	☐	21.2
sd33	1	☐	24.9
NA	2	☐	NA

test units

2 ∈ `col_vals_in_set(b, c(0, 1))`

Interrogating the Table

a	b	c
yko2	1	23.1
lju7	0	16.3
qib0	1	21.2
sd33	1	24.9
NA	2	NA

test units

2 ∈ `col_vals_in_set(b, c(0, 1))`

REPORT

UNITS	PASS	FAIL
5	4 0.8	1 0.2

Interrogating the Table

a		b	c
yko2	■	1	23.1
lju7	■	0	16.3
qib0	■	1	21.2
sd33	■	1	24.9
NA	■	2	NA

test units

3 .* `col_vals_regex(a, "[a-z]{3}[0-9]")`

REPORT

UNITS	PASS	FAIL
5	3 0.6	2 0.4

Interrogating the Table

a	b	c
yko2	1	23.1
lju7	0	16.3
qib0	1	21.2
sd33	1	24.9
NA	2	NA

test units

4

```
col_vals_expr(~ case_when(
  b == 1 ~ c >= 20,
  b == 0 ~ c <  20))
```

REPORT

UNITS	PASS	FAIL
4	4 1.0	0 0

Interrogating the Table

a	b	c
yko2	1	23.1
lju7	0	16.3
qib0	1	21.2
sd33	1	24.9
NA	2	NA

5

```
col_vals_not_null(vars(a, b, c))
```

Interrogating the Table

a	b	c
yko2	1	23.1
lju7	0	16.3
qib0	1	21.2
sd33	1	24.9
NA	2	NA

5 `col_vals_not_null(vars(a))`

6 `col_vals_not_null(vars(b))`

7 `col_vals_not_null(vars(c))`

Interrogating the Table

a		b	c
yko2	■	1	23.1
lju7	■	0	16.3
qib0	■	1	21.2
sd33	■	1	24.9
NA	■	2	NA

test units

5 `col_vals_not_null(vars(a))`

6 `col_vals_not_null(vars(b))`

7 `col_vals_not_null(vars(c))`

REPORT

UNITS	PASS	FAIL
5	4 0.8	1 0.2

Interrogating the Table

a	b	c
yko2	1	23.1
lju7	0	16.3
qib0	1	21.2
sd33	1	24.9
NA	2	NA

test units

5 `col_vals_not_null(vars(a))`

6 `col_vals_not_null(vars(b))`

7 `col_vals_not_null(vars(c))`

REPORT

UNITS	PASS	FAIL
5	5 1.0	0 0

Interrogating the Table

a	b	c
yko2	1	23.1
lju7	0	16.3
qib0	1	21.2
sd33	1	24.9
NA	2	NA

test units

5 `col_vals_not_null(vars(a))`

6 `col_vals_not_null(vars(b))`

7 `col_vals_not_null(vars(c))`

REPORT

UNITS	PASS	FAIL
5	4 0.8	1 0.2

The pointblank Agent Report

	STEP	UNITS	PASS	FAIL
1	col_vals_gte()	5	4 0.8	1 0.2
2	col_vals_in_set()	5	4 0.8	1 0.2
3	col_vals_regex()	5	3 0.6	2 0.4
4	col_vals_expr()	4	4 1.0	0 0
5	col_vals_not_null()	5	4 0.8	1 0.2
6	col_vals_not_null()	5	5 1.0	0 0
7	col_vals_not_null()	5	4 0.8	1 0.2

The pointblank Agent Report

	STEP	UNITS	PASS	FAIL
1	col_vals_gte()	5	4 0.8	1 0.2
2	col_vals_in_set()	5	4 0.8	1 0.2
3	col_vals_regex()	5	3 0.6	2 0.4
4	col_vals_expr()	4	4 1.0	0 0
5	col_vals_not_null()	5	4 0.8	1 0.2
6	col_vals_not_null()	5	5 1.0	0 0
7	col_vals_not_null()	5	4 0.8	1 0.2

For better reporting on data quality, can set thresholds (and even use side effects).

Failure thresholds can be set for three states

W WARNING

S STOP

N NOTIFY

Let's set:

W to 1
S to 2
(**N** not set)

R CODE

```r
action_levels(
  warn_at = 1,
  stop_at = 2
)
```

The pointblank Agent Report

	STEP	UNITS	PASS	FAIL	W	S	N
1	col_vals_gte()	5	4 0.8	1 0.2	●	○	—
2	col_vals_in_set()	5	4 0.8	1 0.2	●	○	—
3	col_vals_regex()	5	3 0.6	2 0.4	●	●	—
4	col_vals_expr()	4	4 1.0	0 0	○	○	—
5	col_vals_not_null()	5	4 0.8	1 0.2	●	○	—
6	col_vals_not_null()	5	5 1.0	0 0	○	○	—
7	col_vals_not_null()	5	4 0.8	1 0.2	●	○	—

The pointblank Agent Report

Pointblank Validation

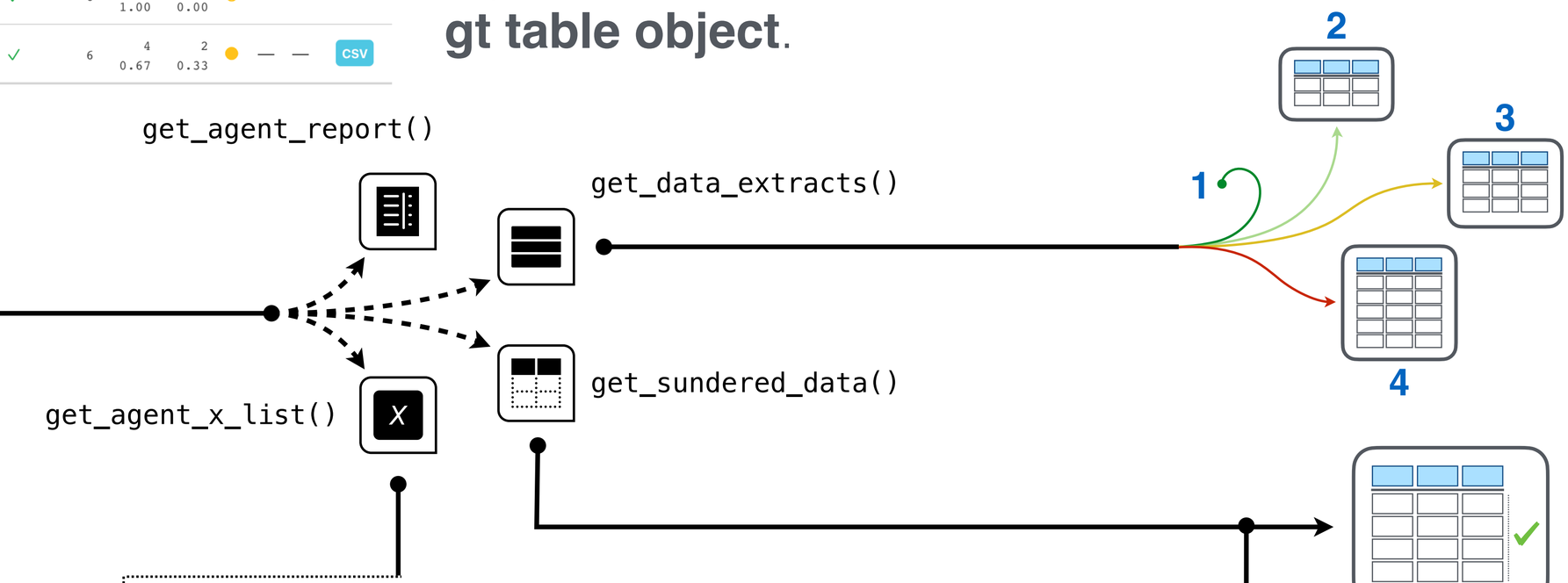
agent_2020-08-13_11:42:20 (2020-08-13 11:42:20)

	STEP	COLUMNS	VALUES	TBL	EVAL	UNITS	PASS	FAIL	W	S	N	EXT
1	col_vals_gte	c	15		✓	5	4 0.80	1 0.20	●	○	—	CSV
2	col_vals_in_set	b	0, 1		✓	5	4 0.80	1 0.20	●	○	—	CSV
3	col_vals_regex	a	[a-z]{3}[0-9]		✓	5	3 0.60	2 0.40	●	●	—	CSV
4	col_vals_expr	—	case_when(b ==…		✓	4	4 1.00	0 0.00	○	○	—	—
5	col_vals_not_null	a	—		✓	5	4 0.80	1 0.20	●	○	—	CSV
6	col_vals_not_null	b	—		✓	5	5 1.00	0 0.00	○	○	—	—
7	col_vals_not_null	c	—		✓	5	4 0.80	1 0.20	●	○	—	CSV

Other Post-Interrogation Ops

EVAL	UNITS	PASS	FAIL	W	S	N	EXT
✓	6	6 1.00	0 0.00	○	—	—	—
✓	6	4 0.67	2 0.33	●	—	—	CSV

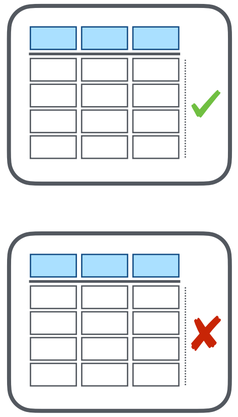
We can get the agent report as a customizable **gt table object**.



Get **table fragments** for failing test units (from row-based validations)

```
$time (POSIXct [1])
$name $tbl_name $tbl_src $tbl_src_details (chr [1])
$tbl (tbl_df, tbl, and data.frame)
$col_names $col_types (chr [1])
$i $type $columns $values $briefs (mixed [2])
$eval_error $eval_warning (lgl [2])
$capture_stack (list [2])
$n $n_passed $n_failed $f_passed $f_failed (num [2])
$warn $stop $notify (lgl [2])
$validation_set (tbl_df [2, 25])
$reporting_lang (chr [1])
$report_object (gt_tbl)
$email_object (blastula_message)
$report_html $report_html_small (chr [1])
```

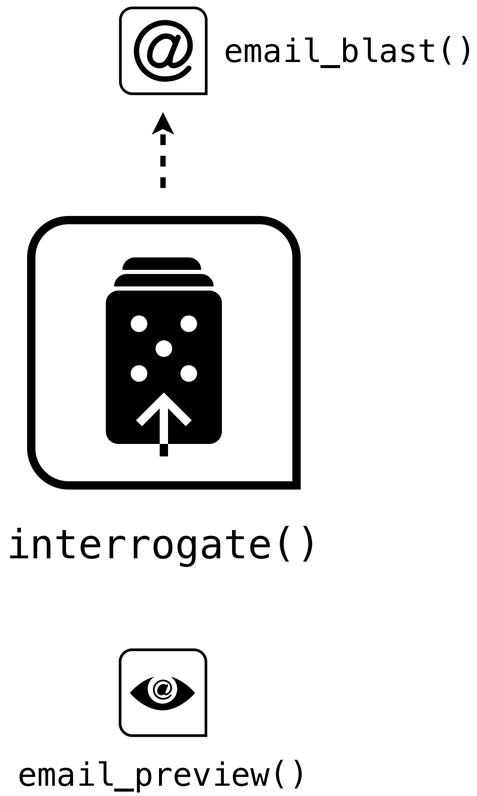
Get a giant list object containing all of the **agent intel**.



Get two table fragments split across the **passing** and **failing** test units.

Notifying with Email

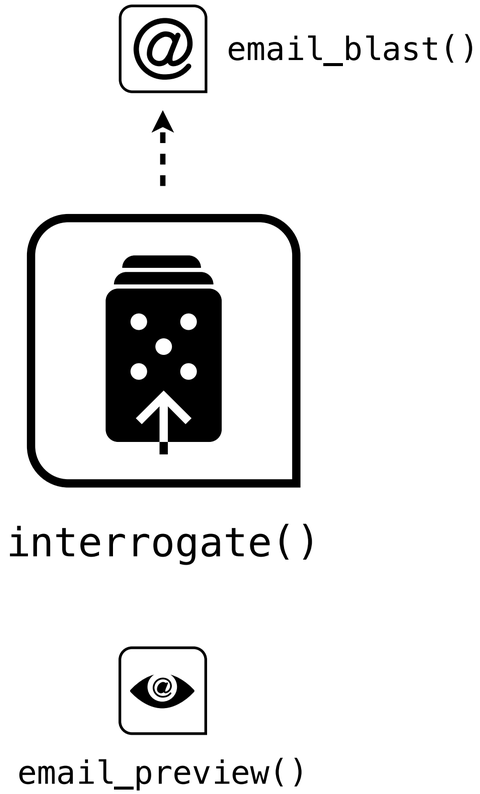
Send email (or *not*) depending on the interrogation results.



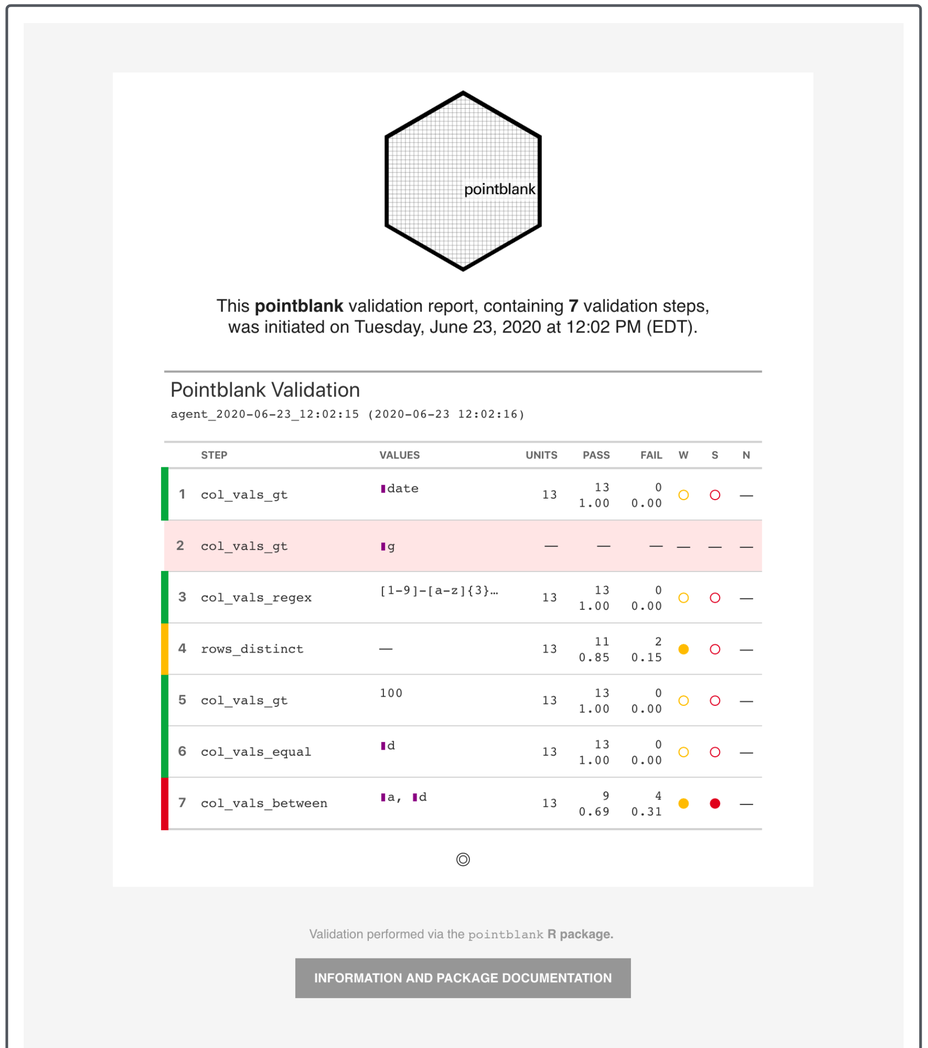
Preview a pointblank email, helpful for customization.

Notifying with Email

Send email (or *not*) depending on the interrogation results.



Preview a pointblank email, helpful for customization.



pointblank

This **pointblank** validation report, containing **7** validation steps, was initiated on Tuesday, June 23, 2020 at 12:02 PM (EDT).

Pointblank Validation

agent_2020-06-23_12:02:15 (2020-06-23 12:02:16)

	STEP	VALUES	UNITS	PASS	FAIL	W	S	N
1	col_vals_gt	date	13	13 1.00	0 0.00	○	○	—
2	col_vals_gt	g	—	—	—	—	—	—
3	col_vals_regex	[1-9]-[a-z]{3}…	13	13 1.00	0 0.00	○	○	—
4	rows_distinct	—	13	11 0.85	2 0.15	●	○	—
5	col_vals_gt	100	13	13 1.00	0 0.00	○	○	—
6	col_vals_equal	d	13	13 1.00	0 0.00	○	○	—
7	col_vals_between	a, d	13	9 0.69	4 0.31	●	●	—

Validation performed via the pointblank R package.

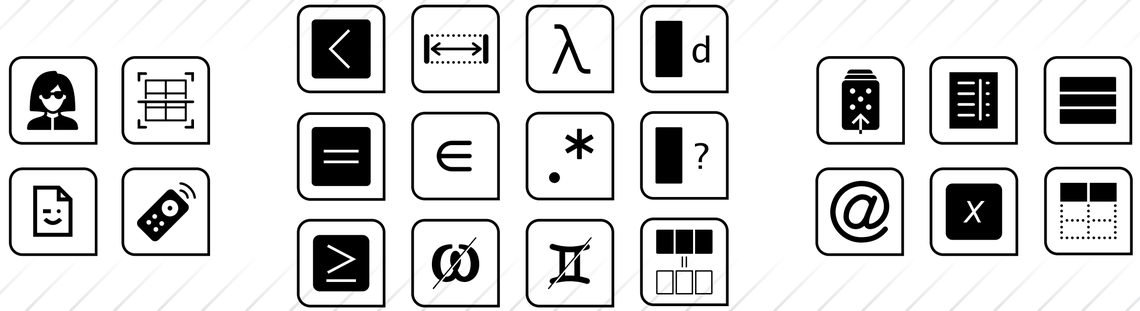
INFORMATION AND PACKAGE DOCUMENTATION

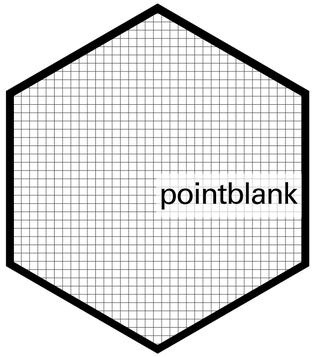
This is the stock email w/o any customization.

It's made with the **blastula** package. It's an HTML email that is well-tested in multiple email clients.

Demo

The **pointblank** R Package





github.com/rich-iannone/pointblank

github.com/rich-iannone/presentations

rich-iannone

@riannone

rich@rstudio.com