Package: DataFakeR (via r-universe)

March 21, 2025

Type Package

Title Generate Fake Data for Relational Databases

Version 0.1.3

Maintainer Krystian Igras <krystian8207@gmail.com>

Description Based on provided database description and/or database connection generate data sample preserving source structure.

License MIT + file LICENSE

Encoding UTF-8

Imports yaml, purrr, tidygraph, dplyr (>= 1.0.0), tibble, magrittr, glue, R6

Suggests DBI, charlatan, stringr, stringi, rmarkdown, knitr, covr, lintr, httr, mockery, testthat (>= 3.0.0), rcmdcheck

Config/testthat/edition 3

RoxygenNote 7.2.3

VignetteBuilder knitr

Collate 'DataFaker-package.R' 'simulate_char_col.R' 'simulate_num_col.R' 'simulate_int_col.R' 'simulate_lgl_col.R' 'simulate_dat_col.R' 'simulate_cols.R' 'simulate_tables.R' 'schema_utils.R' 'schema_deps.R' 'schema_conf.R' 'schema_source.R' 'schema_from_db.R' 'schema_from_list.R'

URL https://github.com/openpharma/DataFakeR

NeedsCompilation no

Author Krystian Igras [aut, cre], Kamil Wais [ctb], Adam Foryś [ctb], Adam Leśniewski [ctb], Paweł Kawski [ctb]

Date/Publication 2023-02-12 14:22:11 UTC

Additional_repositories https://cranhaven.r-universe.dev

Config/pak/sysreqs libglpk-dev libicu-dev libxml2-dev

Repository https://cranhaven.r-universe.dev

RemoteUrl https://github.com/cranhaven/cranhaven.r-universe.dev

RemoteRef package/DataFakeR

RemoteSha 797cbf8cc2d6600c439e6e218b1ccace9eaccc27

RemoteSubdir DataFakeR

Contents

`default_simulation_params`

Setup default column type parameters

Description

All the parameters (excluding `regexp`) are attached to column definition when the ones are not specified in configuration YAML file. All the functions are used to specify default configuration (see: `default_faker_opts`).

Usage

```
opt_default_character(
  regexp = "text|char|factor",
  nchar = 10,
  na_ratio = 0.05,
  not_null = FALSE,
  unique = FALSE,
  default = "",
  levels_ratio = 1,
  ...
)
```

```
  opt_default_numeric(
    regexp = "^decimal|^numeric|real|double precision",
    na_ratio = 0.05,
    not_null = FALSE,
    unique = FALSE,
    default = 0,
    precision = 7,
    scale = 2,
    levels_ratio = 1,
    ...
  )

  opt_default_integer(
    regexp = "smallint|integer|bigint|smallserial|serial|bigserial",
    na_ratio = 0.05,
    not_null = FALSE,
    unique = FALSE,
    default = "",
    levels_ratio = 1,
    ...
  )

  opt_default_logical(
    regexp = "boolean|logical",
    na_ratio = 0.05,
    not_null = FALSE,
    unique = FALSE,
    default = FALSE,
    levels_ratio = 1,
    ...
  )

  opt_default_date(
    regexp = "date|Date",
    na_ratio = 0.05,
    not_null = FALSE,
    unique = FALSE,
    default = Sys.Date(),
    format = "%Y-%m-%d",
    min_date = as.Date("1970-01-01"),
    max_date = Sys.Date(),
    levels_ratio = 1,
    ...
  )
```

Arguments

`regexp`	Regular expression that allows mapping YAML configuration column type to desired R class.
`nchar`	Maximum number of characters when simulating character values. When source column is of type `char(n)` the parameter is ignored.
`na_ratio`	Ratio of NA values returned in simulated sample.
`not_null`	Should the column allow to simulate NA values?
`unique`	Should column values be unique?
`default`	Default column value. Ignored during simulation.
`levels_ratio`	Ratio of unique values (in terms of sample length) simulated in the sample.
`...`	Other default parameters attached to the column definition.
`precision`	Precision of numeric column value when simulating numeric values. When source column is of type e.g. `numeric(precision)` the parameter is ignored.
`scale`	Precision of numeric column value when simulating numeric values. When source column is of type e.g. `numeric(precision, scale)` the parameter is ignored.
`format`	Format of date used when simulating Date columns.
`min_date, max_date`	Minimum and maximum date used when simulating Date columns.

`faker_configuration` *Default options for pulling metadata and data simulation*

Description

Generated with the set of configuration functions: default_simulation_params, opt_default_table, special_simulation, restricted_simulation, sourcing_metadata.

Usage

```
default_faker_opts

set_faker_opts(
  opt_pull_character,
  opt_pull_numeric,
  opt_pull_integer,
  opt_pull_logical,
  opt_pull_date,
  opt_pull_table,
  opt_default_character,
  opt_simul_spec_character,
  opt_simul_restricted_character,
  opt_simul_default_fun_character,
```

```
  opt_default_numeric,
  opt_simul_spec_numeric,
  opt_simul_restricted_numeric,
  opt_simul_default_fun_numeric,
  opt_default_integer,
  opt_simul_spec_integer,
  opt_simul_restricted_integer,
  opt_simul_default_fun_integer,
  opt_default_logical,
  opt_simul_spec_logical,
  opt_simul_restricted_logical,
  opt_simul_default_fun_logical,
  opt_default_date,
  opt_simul_spec_date,
  opt_simul_restricted_date,
  opt_simul_default_fun_date,
  opt_default_table,
  global = TRUE
)

get_faker_opts()
```

Arguments

```
opt_pull_character,       opt_pull_numeric,       opt_pull_integer,
opt_pull_logical,           opt_pull_date,           opt_pull_table,
opt_default_character,                     opt_simul_spec_character,
opt_simul_restricted_character,    opt_simul_default_fun_character,
opt_default_numeric,                         opt_simul_spec_numeric,
opt_simul_restricted_numeric,        opt_simul_default_fun_numeric,
opt_default_integer,                         opt_simul_spec_integer,
opt_simul_restricted_integer,        opt_simul_default_fun_integer,
opt_default_logical,                         opt_simul_spec_logical,
opt_simul_restricted_logical,        opt_simul_default_fun_logical,
opt_default_date,  opt_simul_spec_date,  opt_simul_restricted_date,
opt_simul_default_fun_date, opt_default_table
```

Parameters defined in default configuration that can be modified by using `set_faker_opts` function. Please make sure each parameter is specified by method designed to it.

`global` If TRUE, default the configuration will be set up globally (no need to pass it as a `faker_opts` parameter for schema_source and schema_methods).

Format

An object of class `list` of length 27.

Details

`set_faker_opts` allows to overwrite selected options. `get_faker_opts` lists the current options

configuration.

`number_of_rows` *Methods for extracting number of target rows in simulation*

Description

Each method returns function of list of tables. The value of such function is named list being mapping between tables (names of list) and target number of rows (values of list). Such methods can be passed as `nrows` parameter of opt_default_table.

Usage

```
nrows_simul_constant(n, force = FALSE)

nrows_simul_ratio(ratio, total, force = FALSE)
```

Arguments

`n`	Default number of rows for each table when not defined in configuration file.
`force`	Should specified parameters overwrite related configuration parameters?
`ratio, total`	The parameters multiplications results with defining target number of rows for simulated table. See details section.

Details

Currently supported methods are:

- nrows_simul_constant Returns `n` rows for each table when not defined in YAML parameter `nrows`
- nrows_simul_ratio Returns `nrows * ratio` when `nrows` defined as YAML parameter and is integer. Returns `nrows` when `nrows` defined as YAML parameter and id fraction, Returns `n * ratio` otherwise.

`opt_default_table` *Configure data simulation options*

Description

The parameters affect high level (not column type related) simulation settings such as target number of rows for each table. Currently only number of simulated rows is supported.

Usage

```
opt_default_table(nrows = nrows_simul_constant(10))
```

Arguments

nrows	Integer or function. When nrows is precised as an integer value, all the tables will have the same number of rows. In case of function, the should take tables configuration (list of tables section from configuration YSML file) and return named list of table with rows values. See nrows_simul_constant and nrows_simul_ratio for more details.

---

restricted_simulation *Simulate data restricted by extra column parameters*

Description

The functions allow to define a set of methods for simulating data using additional column-based parameters such as range or values.

Usage

```
opt_simul_restricted_character(
  f_key = simul_restricted_character_fkey,
  ...,
  in_set = simul_restricted_character_in_set
)

opt_simul_restricted_numeric(
  f_key = simul_restricted_numeric_fkey,
  ...,
  in_set = simul_restricted_numeric_in_set,
  range = simul_restricted_numeric_range
)

opt_simul_restricted_integer(
  f_key = simul_restricted_integer_fkey,
  ...,
  in_set = simul_restricted_integer_in_set,
  range = simul_restricted_integer_range
)

opt_simul_restricted_logical(f_key = simul_restricted_integer_fkey, ...)

opt_simul_restricted_date(
  f_key = simul_restricted_integer_fkey,
  ...,
  range = simul_restricted_date_range
)
```

Arguments

f_key	Method for simulating foreign key columns. The values parameter of the function, receives all the unique values from parent primary key column.
...	Other methods that can be defined to handle extra parameters.
in_set	Method for simulating columns from defined set of values. The values parameter of the function, take all the values defined in YAML column definition as values parameter.
range	Method for simulating columns fitting inside defined range. It takes special parameter range 2-length vector minimum and maximum value for simulated data.

Details

Except for the standard column parameters, that are now:

- type
- unique
- not_null
- default
- nchar
- min_date
- max_date
- precision
- scale

it is also allowed to add custom ones (either directly in YAML configuration file, or in opt_default_<column_type> functions).

In order to respect simulation using such parameters, we may want to define our custom simulation functions.

Such functions should be defined as a parameters of opt_simul_restricted_<column_type> functions, and each of them should take special parameter as its own one.

When the parameter condition is not met (for example the parameter is missing) such function should return NULL value. This allows the simulation workflow to move to the next defined method. The order of methods execution is followed by the order of defined parameters in the below methods.

That means, the highest priority always have f_key - a special method that is used for foreign key columns, and simulates only from values received from parent primary key.

The second priority method for character type columns is in_set, that seeks for values column parameter, and when such exists it simulates the data from defined set of values. See simul_restricted_character_in_set definition to check details.

sample_modifiers *Modify sample with desired condition*

Description

The set of function that allows to perform most common operations ion data sample.

Usage

```
unique_sample(sim_expr, ..., unique = TRUE, n_name = "n", n_iter = 10)

na_rand(sample_vec, na_ratio, not_null = FALSE)

levels_rand(sample_vec, levels_ratio, unique)
```

Arguments

sim_expr	Expression to be evaluated in order to get column sample.
...	Parameters and their values that are used in sim_expr.
unique	If TRUE the function will try to simulate unique values.
n_name	Name of the parameter providing sample length (for example 'n' for rnorm and 'size' for sample).
n_iter	Number of iteration to make to assure the returned values are unique.
sample_vec	Vector to which NA values should be injected.
na_ratio	Ratio (in terms of column length) of NA values to attach to the sample.
not_null	Information whether NA's are allowed.
levels_ratio	Ratio of unique levels in terms of whole sample length.

Details

unique_sample - takes simulation expression and assures the expression will be executed as many times as needed to return unique result sample. na_rand - attaches NA values to the sample according to provided NA's ratio. levels_rand - takes provided number of sample levels, and assures the returned sample have as many levels as requested.

Examples

```
unique_sample(rnorm(n, mean = my_mean), n = 10, my_mean = 2)
unique_sample(sample(values, size, replace = TRUE), size = 10, values = 1:10, n_name = "size")

## Not run:
  ## In 10 iterations it was not possible to simulate 6 unique values from the vector 1:5
 unique_sample(sample(values, size, replace = TRUE), size = 6, values = 1:5, n_name = "size")

## End(Not run)
```

```
na_rand(1:10, na_ratio = 0.5)
```

---

schema_methods *Schema object methods*

Description

The set of methods that can be used on schema object returned by schema_source function.

Usage

```
schema_update_source(
  schema,
  file,
  faker_opts = getOption("dfkr_options", default_faker_opts)
)

schema_get_table(schema, table_name)

schema_plot_deps(schema, table_name)

schema_simulate(schema)
```

Arguments

schema	Schema object keeping table dependency graph.
file	Path to schema configuration yaml file.
faker_opts	Structure sourcing and columns simulation config.
table_name	Name of the table.

Details

The methods are:

- schema_update_source Update schema dependency graph based on provided file.
- schema_simulate Run data simulation process.
- schema_get_table Get simulated table value.
- schema_plot_deps Plot inter or inner table dependecies.

schema_source *Source schema file into dependency graph object*

Description

The functions parses table schema (from database) and saves its structure yaml format. The defined structure is then used to prepare schema dependency graph, that is:

- dependencies between tablesBased on foreign key definitions
- inner table column dependenciesBased on defined dependencies by various methods. See `vignette('todo')`.

Usage

```
schema_source(
  source,
  schema = "public",
  file = if (is.character(source)) source else file.path(getwd(), "schema.yml"),
  faker_opts = getOption("dfkr_options", default_faker_opts)
)
```

Arguments

`source`	Connection to Redshift or Postgres database or path to YAML configuration file from which schema metadata should be sourced. When missing `file` defined file will be sourced if existing.
`schema`	Schema name from which the structure should be sourced.
`file`	Path to yaml file describing database schema, or target file when schema should be saved (when db_conn not mising). See `vignette('todo')`.
`faker_opts`	Structure sourcing and columns simulation config.

Details

Detected dependencies are then saved in R6Class object that is returned and possible to pass for further methods. See schema_methods.

Keeping the schema as a graph allows to perform simulation process in proper order, preserving table dependencies and constraints.

simulation_methods_character *Character type simulation methods*

Description

Character type simulation methods

Usage

```
simul_spec_character_name(
  n,
  not_null,
  unique,
  default,
  spec_params,
  na_ratio,
  levels_ratio,
  ...
)

simul_default_character(
  n,
  not_null,
  unique,
  default,
  nchar,
  type,
  na_ratio,
  levels_ratio,
  ...
)

simul_restricted_character_in_set(
  n,
  not_null,
  unique,
  default,
  nchar,
  type,
  values,
  na_ratio,
  levels_ratio,
  ...
)

simul_restricted_character_fkey(
```

```
  n,
  not_null,
  unique,
  default,
  nchar,
  type,
  values,
  na_ratio,
  levels_ratio,
  ...
)
```

Arguments

n	Number of values to simulate.
not_null	Should NA values be forbidden?
unique	Should duplicated values be allowed?
default	Default column value.
spec_params	Set of parameters passed to special method.
na_ratio	Ratio of NA values (in terms of sample length) the sample should have.
levels_ratio	Fraction of levels (in terms of sample length) the sample should have.
...	Other parameters passed to column configuration in YAML file.
nchar	Maximum number of characters for each value.
type	Column raw type (sourced from configuration file).
values	Possible values from which to perform simulation.

`simulation_methods_date`

Date type simulation methods

Description

Date type simulation methods

Usage

```
simul_spec_date_distr(
  n,
  not_null,
  unique,
  default,
  spec_params,
  na_ratio,
  levels_ratio,
```

```
  ...
)

simul_default_date(
  n,
  not_null,
  unique,
  default,
  type,
  min_date,
  max_date,
  format,
  na_ratio,
  levels_ratio,
  ...
)

simul_restricted_date_range(
  n,
  not_null,
  unique,
  default,
  type,
  range,
  format,
  na_ratio,
  levels_ratio,
  ...
)

simul_restricted_date_fkey(
  n,
  not_null,
  unique,
  default,
  type,
  values,
  na_ratio,
  levels_ratio,
  ...
)
```

Arguments

n	Number of values to simulate.
not_null	Should NA values be forbidden?
unique	Should duplicated values be allowed?
default	Default column value.

spec_params	Set of parameters passed to special method.
na_ratio	Ratio of NA values (in terms of sample length) the sample should have.
levels_ratio	Fraction of levels (in terms of sample length) the sample should have.
...	Other parameters passed to column configuration in YAML file.
type	Column raw type (sourced from configuration file).
format	Date format used to store dates.
range, min_date, max_date	Date range or minimum and maximum date from which to simulate data.
values	Possible values from which to perform simulation.

`simulation_methods_integer`

Integer type simulation methods

Description

Integer type simulation methods

Usage

```
simul_spec_integer_distr(
  n,
  not_null,
  unique,
  default,
  spec_params,
  na_ratio,
  levels_ratio,
  ...
)

simul_default_integer(
  n,
  not_null,
  unique,
  default,
  type,
  na_ratio,
  levels_ratio,
  ...
)

simul_restricted_integer_range(
  n,
  not_null,
```

```
  unique,
  default,
  type,
  range,
  na_ratio,
  levels_ratio,
  ...
)

simul_restricted_integer_in_set(
  n,
  not_null,
  unique,
  default,
  type,
  values,
  na_ratio,
  levels_ratio,
  ...
)

simul_restricted_integer_fkey(
  n,
  not_null,
  unique,
  default,
  type,
  values,
  na_ratio,
  levels_ratio,
  ...
)
```

Arguments

n	Number of values to simulate.
not_null	Should NA values be forbidden?
unique	Should duplicated values be allowed?
default	Default column value.
spec_params	Set of parameters passed to special method.
na_ratio	Ratio of NA values (in terms of sample length) the sample should have.
levels_ratio	Fraction of levels (in terms of sample length) the sample should have.
...	Other parameters passed to column configuration in YAML file.
type	Column raw type (sourced from configuration file).
range	Possible range of values from which to perform simulation.
values	Possible values from which to perform simulation.

simulation_methods_logical *Logical type simulation methods*

Description

Logical type simulation methods

Usage

```
simul_spec_logical_distr(
  n,
  not_null,
  unique,
  default,
  spec_params,
  na_ratio,
  levels_ratio,
  ...
)

simul_default_logical(
  n,
  not_null,
  unique,
  default,
  type,
  na_ratio,
  levels_ratio,
  ...
)

simul_restricted_logical_fkey(
  n,
  not_null,
  unique,
  default,
  type,
  values,
  na_ratio,
  levels_ratio,
  ...
)
```

Arguments

n	Number of values to simulate.

not_null	Should NA values be forbidden?
unique	Should duplicated values be allowed?
default	Default column value.
spec_params	Set of parameters passed to special method.
na_ratio	Ratio of NA values (in terms of sample length) the sample should have.
levels_ratio	Fraction of levels (in terms of sample length) the sample should have.
...	Other parameters passed to column configuration in YAML file.
type	Column raw type (sourced from configuration file).
values	Possible values from which to perform simulation.

`simulation_methods_numeric` *Numeric type simulation methods*

Description

Numeric type simulation methods

Usage

```
simul_spec_numeric_distr(
  n,
  not_null,
  unique,
  default,
  spec_params,
  na_ratio,
  levels_ratio,
  ...
)

simul_default_numeric(
  n,
  not_null,
  unique,
  default,
  type,
  na_ratio,
  levels_ratio,
  ...
)

simul_restricted_numeric_range(
  n,
```

```
  not_null,
  unique,
  default,
  type,
  range,
  na_ratio,
  levels_ratio,
  ...
)

simul_restricted_numeric_in_set(
  n,
  not_null,
  unique,
  default,
  type,
  values,
  na_ratio,
  levels_ratio,
  ...
)

simul_restricted_numeric_fkey(
  n,
  not_null,
  unique,
  default,
  type,
  values,
  na_ratio,
  levels_ratio,
  ...
)
```

Arguments

n	Number of values to simulate.
not_null	Should NA values be forbidden?
unique	Should duplicated values be allowed?
default	Default column value.
spec_params	Set of parameters passed to special method.
na_ratio	Ratio of NA values (in terms of sample length) the sample should have.
levels_ratio	Fraction of levels (in terms of sample length) the sample should have.
...	Other parameters passed to column configuration in YAML file.
type	Column raw type (sourced from configuration file).
range	Possible range of values from which to perform simulation.

values Possible values from which to perform simulation.

---

sourcing_metadata *Specify YAML configuration options while pulling the schema from DB*

Description

The set of function allows to configure which data information should be saved to configuration YAML file when such configuration is sourced directly from database schema.

Usage

```
opt_pull_character(
  values = TRUE,
  max_uniq_to_pull = 10,
  nchar = TRUE,
  na_ratio = TRUE,
  levels_ratio = TRUE,
  ...
)

opt_pull_numeric(
  values = TRUE,
  max_uniq_to_pull = 10,
  range = TRUE,
  precision = TRUE,
  scale = TRUE,
  na_ratio = TRUE,
  levels_ratio = FALSE,
  ...
)

opt_pull_integer(
  values = TRUE,
  max_uniq_to_pull = 10,
  range = TRUE,
  na_ratio = TRUE,
  levels_ratio = FALSE,
  ...
)

opt_pull_date(range = TRUE, na_ratio = TRUE, levels_ratio = FALSE, ...)

opt_pull_logical(na_ratio = TRUE, levels_ratio = FALSE, ...)

opt_pull_table(nrows = "exact", ...)
```

Arguments

values	Should column unique values be sourced? If so the ones are stored as an array withing values parameter.
max_uniq_to_pull	Pull unique values only when the distinct number of them is less than provided value. The parameter prevents for sourcing large amount of values to configuration file for example when dealing with ids column.
nchar	Should maximum number of characters in column be pulled? Is so stored as nchar parameter in configuration YAML file.
na_ratio	Should ratio of NA values existing in column be sourced?
levels_ratio	Should ratio of unique column values be sourced?
...	Other parameters defining column metadata source. Currently unsupported.
range	Should column range be sourced? Is so stored as range parameter in configuration YAML file.
precision	Currently unused.
scale	Currently unused.
nrows	Should number of original columns be sourced? When 'exact' stored as a nrows parameter for each table in YAML configuration file. When 'ratio' stored as a fraction of original columns (based on all tables) and saved as nrows configuration parameter. When 'none' tables rows information will not be saved.

---

special_simulation *Set of functions defining special simulation methods for column and its type*

Description

Whenever there's a need to simulate column using specific function (as a spec parameter in YAML configuration file), such method should be defined in one of opt_simul_spec_<column_type> functions.

Usage

```
opt_simul_spec_character(name = simul_spec_character_name, ...)

opt_simul_spec_numeric(distr = simul_spec_numeric_distr, ...)

opt_simul_spec_integer(distr = simul_spec_integer_distr, ...)

opt_simul_spec_logical(distr = simul_spec_logical_distr, ...)

opt_simul_spec_date(distr = simul_spec_date_distr, ...)
```

Arguments

name	Function for simulating personal names.
...	Other custom special methods.
distr	Function for simulating data from desired distribution.

Details

Currently defined special methods are:

- name For character column, that allows to simulate character reflecting real names and surnames
- distr For all the remaining column types. The method allows to simulate data with specified distribution generator, such as rnorm, rbinom etc.

Each 'spec' method receives n parameter (the desired number of rows to simulate), all the default column-based parameters (type, unique, not_null, etc.) but also a special one named spec_params that are applied to selected distribution simulation method.

See for example simul_spec_character_name definition.

Index