

# Package: testmaker (via r-universe)

January 5, 2025

**Title** Facilitate writing testthat tests and parameter checks for data frames

**Version** 0.0.0.9000

**Description** Based on a template dataframe, generates R code for `testthat` tests or `stopifnot` checks. Intended to streamline testwriting rather than completely automate it; user will need to review the generated code and decide which comparisons make sense.

**License** MIT + file LICENSE

**Encoding** UTF-8

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.3.2

**Suggests** testthat (>= 3.0.0)

**Config/testthat/edition** 3

**Imports** cli, clipr, glue, rlang, utils

**URL** <https://framverse.r-universe.dev/testmaker>,  
<https://github.com/cbedwards-dfw/testmaker>

**Config/pak/sysreqs** libx11-dev

**Repository** <https://framverse.r-universe.dev>

**RemoteUrl** <https://github.com/cbedwards-dfw/testmaker>

**RemoteRef** HEAD

**RemoteSha** 00c82b02c9d10ec38bca75b099749df3ef56776c

## Contents

dput_to_string . . . . .	2
finish_testmaker . . . . .	3
testmaker_df_class_cli . . . . .	3
testmaker_df_class_sin . . . . .	4
testmaker_df_class_tt . . . . .	5

testmaker_df_cli . . . . .	6
testmaker_df_colcontent_sin . . . . .	7
testmaker_df_colcontent_tt . . . . .	8
testmaker_df_dim_cli . . . . .	9
testmaker_df_dim_sin . . . . .	10
testmaker_df_dim_tt . . . . .	10
testmaker_df_isit_tt . . . . .	11
testmaker_df_names_sin . . . . .	12
testmaker_df_names_sin_orderless . . . . .	13
testmaker_df_names_tt . . . . .	13
testmaker_df_sin . . . . .	14
testmaker_df_tt . . . . .	15
testmaker_df_validator . . . . .	16
validate_testmaker . . . . .	17

**Index** **19**

---

dput_to_string	<i>Helper function to convert string to R code to regenerate that string.</i>
----------------	---

---

**Description**

Helper function to convert string to R code to regenerate that string.

**Usage**

```
dput_to_string(object)
```

**Arguments**

object            R object, expecting a vector

**Value**

Character atomic of R code to recreate the object. Plays well with glue functions and cat().

**Examples**

```
## Not run:
temp = dput_to_string(rownames(mtcars))
cat(temp)

## End(Not run)
```

---

finish_testmaker	<i>Handle the standardized return options for all stopifnotmaker and testmaker functions</i>
------------------	--

---

**Description**

Internal helper function.

**Usage**

```
finish_testmaker(test.text, return.style, silent)
```

**Arguments**

test.text	Character vector of the finished lines of codes to display or provide as appropriate.
return.style	Designation for what to return. "clip" returns nothing, but loads the text into the clipboard. "text" returns the text as a character vector, "none" returns nothing
silent	Suppress printing text to console? logical; default of false.

**Value**

Either nothing or a character vector of lines of R code for writing a testthat test, depending on return.style.

---

testmaker_df_class_cli	<i>Generate cli_abort code for dataframe column classes</i>
------------------------	---

---

**Description**

Generate cli\_abort code for dataframe column classes

**Usage**

```
testmaker_df_class_cli(  
  x,  
  return.style = c("clip", "text", "none"),  
  silent = FALSE,  
  object.name = "res",  
  for.fun = FALSE  
)
```

**Arguments**

<code>x</code>	Template dataframe of target object (e.g. we expect our test object to share properties with this)
<code>return.style</code>	Designation for what to return. "clip" returns nothing, but loads the text into the clipboard. "text" returns the text as a character vector, "none" returns nothing
<code>silent</code>	Suppress printing text to console? logical; default of false.
<code>object.name</code>	Name of the object to apply the stopifnot to; presumably the name of the dataframe argument
<code>for.fun</code>	Is the resulting text meant to be inserted into a helper function? If TRUE, adds in support for correctly identifying the argument name and function name based on <code>have.arguments(arg = rlang::caller_arg(x) and call = rlang::caller_env()</code> in the helper function.

**Value**

Either nothing or a character vector of lines of R code for writing a stopifnot test.

**Examples**

```
testmaker_df_class_cli(cars, return.style = "text")
```

---

```
testmaker_df_class_sin
```

**DEPRECATED** *Generate stopifnot code for dataframe column classes*

---

**Description**

**DEPRECATED** Generate stopifnot code for dataframe column classes

**Usage**

```
testmaker_df_class_sin(
  x,
  return.style = c("clip", "text", "none"),
  silent = FALSE,
  object.name = "res"
)
```

**Arguments**

<code>x</code>	Template dataframe of target object (e.g. we expect our test object to share properties with this)
<code>return.style</code>	Designation for what to return. "clip" returns nothing, but loads the text into the clipboard. "text" returns the text as a character vector, "none" returns nothing

<code>silent</code>	Suppress printing text to console? logical; default of false.
<code>object.name</code>	Name of the object to apply the stopifnot to; presumably the name of the dataframe argument

**Value**

Either nothing or a character vector of lines of R code for writing a stopifnot test.

---

`testmaker_df_class_tt` *Generate testthat code for dataframe column classes*

---

**Description**

Generate testthat code for dataframe column classes

**Usage**

```
testmaker_df_class_tt(
  x,
  return.style = c("clip", "text", "none"),
  silent = FALSE
)
```

**Arguments**

<code>x</code>	Template dataframe of target object (e.g. we expect our test object to share properties with this)
<code>return.style</code>	Designation for what to return. "clip" returns nothing, but loads the text into the clipboard. "text" returns the text as a character vector, "none" returns nothing
<code>silent</code>	Suppress printing text to console? logical; default of false.

**Value**

Either nothing or a character vector of lines of R code for writing a testthat test, depending on `return.style`.

**Examples**

```
testmaker_df_class_tt(cars, return.style = "text")
```

---

testmaker_df_cli	<i>Workhorse function to generate cli_abort tests for dataframe inputs</i>
------------------	--

---

## Description

Generates R code to test expectations based on a template database. Intended workflow: when writing functions to work with a dataframe with a specific expected structure, load an example of that dataframe into the environment and then call `testmaker_df(exampledataframe)`. This will print if statements and `cli::cli_abort(...)` function calls based on the dimensions of the example data frame, the classes of the columns, and the names of the columns. Note that this is not intended to replace decision-making, but rather to streamline the process of generating the relevant code. In particular, testing the number of rows is often not appropriate (e.g., in many use cases the input data frame is expected to have unknown nrow).

## Usage

```
testmaker_df_cli(
  x,
  return.style = c("clip", "text", "none"),
  silent = FALSE,
  object.name = "res",
  col.order.matters = TRUE
)
```

## Arguments

<code>x</code>	Template dataframe of target object (e.g. we expect our test object to share properties with this)
<code>return.style</code>	Designation for what to return. "clip" returns nothing, but loads the text into the clipboard. "text" returns the text as a character vector, "none" returns nothing
<code>silent</code>	Suppress printing text to console? logical; default of false.
<code>object.name</code>	Name of the object to apply the stopifnot to; presumably the name of the dataframe argument
<code>col.order.matters</code>	When we generate code to test for column names, should we test for exact equivalence (TRUE), or just the presence of all the same column names (FALSE)?. Logical, defaults to TRUE. FALSE LOGIC IS NOT CURRENTLY IMPLEMENTED!

## Value

Either nothing or a character vector of lines of R code for writing a testthat test, depending on `return.style`.

**Examples**

```
## Not run:
stopifnotmaker(cars)

## End(Not run)
```

---

```
testmaker_df_colcontent_sin
```

**DEPRECATED** *Generate stopifnot code to check contents of column(s) against template*

---

**Description**

**DEPRECATED** Generate stopifnot code to check contents of column(s) against template

**Usage**

```
testmaker_df_colcontent_sin(
  x,
  cols,
  return.style = c("clip", "text", "none"),
  silent = FALSE,
  object.name = "res",
  for.fun = FALSE
)
```

**Arguments**

x	Template dataframe of target object (e.g. we expect our test object to share properties with this)
cols	character or character vector of columns in x.
return.style	Designation for what to return. "clip" returns nothing, but loads the text into the clipboard. "text" returns the text as a character vector, "none" returns nothing
silent	Suppress printing text to console? logical; default of false.
object.name	Name of the object to apply the stopifnot to; presumably the name of the dataframe argument

**Value**

Either nothing or a character vector of lines of R code for writing a testthat test, depending on return.style.

---

```
testmaker_df_colcontent_tt
```

*Generate testthat code to check contents of column(s) against template*

---

## Description

Useful for QA/QC when (a) all entries of a column should fall within a contained set (e.g. fishery columns in FRAM model outputs or manipulations should only contain fishery IDs present in the FRAM database), and/or (b) when all entries of some template column should present in the output template (e.g., if processing of a FRAM model table should result in a dataframe with all fishery ids present in the FRAM database). For example, in the `framrsquared` package, we may have functions in which we expect the `fishery_id` column of the function output to contain only `fishery_ids` present in the FRAM database. Similarly, we may expect that the output `fishery_id` column contains ALL of the `fishery_ids` present in the FRAM database.

## Usage

```
testmaker_df_colcontent_tt(
  x,
  cols,
  return.style = c("clip", "text", "none"),
  silent = FALSE
)
```

## Arguments

<code>x</code>	Template dataframe of target object (e.g. we expect our test object to share properties with this)
<code>cols</code>	character or character vector of columns in <code>x</code> .
<code>return.style</code>	Designation for what to return. "clip" returns nothing, but loads the text into the clipboard. "text" returns the text as a character vector, "none" returns nothing
<code>silent</code>	Suppress printing text to console? logical; default of false.

## Details

Depending on the number of unique variables in the column(s), this can generate very long code. This function includes comments to distinguish between checking against unexpected entries and checking against missing entries.

## Value

Either nothing or a character vector of lines of R code for writing a `testthat` test, depending on `return.style`.



**Examples**

```
dat = data.frame(state.x77, state = rownames(state.x77))
dat$category = sample(letters[1:5], size = nrow(dat), replace = TRUE)
testmaker_df_colcontent_tt(dat, c("state", "category"), return.style = "none")
```

---

```
testmaker_df_dim_cli  Generate cli_abort code for dataframe dimensions
```

---

**Description**

Generate `cli_abort` code for dataframe dimensions

**Usage**

```
testmaker_df_dim_cli(
  x,
  return.style = c("clip", "text", "none"),
  silent = FALSE,
  object.name = "res",
  for.fun = FALSE
)
```

**Arguments**

<code>x</code>	Template dataframe of target object (e.g. we expect our test object to share properties with this)
<code>return.style</code>	Designation for what to return. "clip" returns nothing, but loads the text into the clipboard. "text" returns the text as a character vector, "none" returns nothing
<code>silent</code>	Suppress printing text to console? logical; default of false.
<code>object.name</code>	Name of the object to apply the stopifnot to; presumably the name of the dataframe argument
<code>for.fun</code>	Is the resulting text meant to be inserted into a helper function? If TRUE, adds in support for correctly identifying the argument name and function name based on <code>have.arguments arg = rlang::caller_arg(x)</code> and <code>call = rlang::caller_env()</code> in the helper function.

**Value**

Either nothing or a character vector of lines of R code for writing a stopifnot test.

**Examples**

```
testmaker_df_dim_cli(cars, return.style = "text")
```

---

testmaker\_df\_dim\_sin    **Deprecated** *Generate stopifnot code for dataframe dimensions*

---

### Description

**Deprecated** Generate stopifnot code for dataframe dimensions

### Usage

```
testmaker_df_dim_sin(
  x,
  return.style = c("clip", "text", "none"),
  silent = FALSE,
  object.name = "res"
)
```

### Arguments

x	Template dataframe of target object (e.g. we expect our test object to share properties with this)
return.style	Designation for what to return. "clip" returns nothing, but loads the text into the clipboard. "text" returns the text as a character vector, "none" returns nothing
silent	Suppress printing text to console? logical; default of false.
object.name	Name of the object to apply the stopifnot to; presumably the name of the dataframe argument

### Value

Either nothing or a character vector of lines of R code for writing a stopifnot test.

---

testmaker\_df\_dim\_tt    *Generate testthat code for dataframe dimensions*

---

### Description

From a provided template data frame, generates testthat code to check that the nrow and ncols of the res dataframe match the dimensions of the template data frame. Default behavior loads the code into the clipboard for easy pasting into code.

### Usage

```
testmaker_df_dim_tt(
  x,
  return.style = c("clip", "text", "none"),
  silent = FALSE
)
```

**Arguments**

x	Template dataframe of target object (e.g. we expect our test object to share properties with this)
return.style	Designation for what to return. "clip" returns nothing, but loads the text into the clipboard. "text" returns the text as a character vector, "none" returns nothing
silent	Suppress printing text to console? logical; default of false.

**Value**

Either nothing or a character vector of lines of R code for writing a testthat test, depending on return.style.

**Examples**

```
testmaker_df_dim_tt(cars, return = "none")
```

---

testmaker\_df\_isit\_tt *Generate testthat code for dataframe dimensions*

---

**Description**

From a provided template data frame, generates testthat code to check that the nrow and ncol of the res dataframe match the dimensions of the template data frame. Default behavior loads the code into the clipboard for easy pasting into code.

**Usage**

```
testmaker_df_isit_tt(  
  x,  
  return.style = c("clip", "text", "none"),  
  silent = FALSE  
)
```

**Arguments**

x	Template dataframe of target object (e.g. we expect our test object to share properties with this)
return.style	Designation for what to return. "clip" returns nothing, but loads the text into the clipboard. "text" returns the text as a character vector, "none" returns nothing
silent	Suppress printing text to console? logical; default of false.

**Value**

Either nothing or a character vector of lines of R code for writing a testthat test, depending on return.style.

## Examples

```
testmaker_df_dim_tt(cars, return = "none")
```

---

```
testmaker_df_names_sin
```

*Generate stopifnot code for dataframe column names*

---

## Description

Generate stopifnot code for dataframe column names

## Usage

```
testmaker_df_names_sin(  
  x,  
  return.style = c("clip", "text", "none"),  
  silent = FALSE,  
  object.name = "res"  
)
```

## Arguments

x	Template dataframe of target object (e.g. we expect our test object to share properties with this)
return.style	Designation for what to return. "clip" returns nothing, but loads the text into the clipboard. "text" returns the text as a character vector, "none" returns nothing
silent	Suppress printing text to console? logical; default of false.
object.name	Name of the object to apply the stopifnot to; presumably the name of the dataframe argument

## Value

Either nothing or a character vector of lines of R code for writing a stopifnot test.

## Examples

```
testmaker_df_names_sin(cars, return.style = "text")
```

---

```
testmaker_df_names_sin_orderless
```

**DEPRECATED** *Generate stopifnot code for dataframe column names disregarding order*

---

### Description

**DEPRECATED** Generate stopifnot code for dataframe column names disregarding order

### Usage

```
testmaker_df_names_sin_orderless(
  x,
  return.style = c("clip", "text", "none"),
  silent = FALSE,
  object.name = "res"
)
```

### Arguments

x	Template dataframe of target object (e.g. we expect our test object to share properties with this)
return.style	Designation for what to return. "clip" returns nothing, but loads the text into the clipboard. "text" returns the text as a character vector, "none" returns nothing
silent	Suppress printing text to console? logical; default of false.
object.name	Name of the object to apply the stopifnot to; presumably the name of the dataframe argument

### Value

Either nothing or a character vector of lines of R code for writing a stopifnot test.

### Examples

```
testmaker_df_names_sin(cars, return.style = "text")
```

---

```
testmaker_df_names_tt Generate testthat code for dataframe column names
```

---

### Description

Generate testthat code for dataframe column names

**Usage**

```
testmaker_df_names_tt(
  x,
  return.style = c("clip", "text", "none"),
  silent = FALSE
)
```

**Arguments**

x	Template dataframe of target object (e.g. we expect our test object to share properties with this)
return.style	Designation for what to return. "clip" returns nothing, but loads the text into the clipboard. "text" returns the text as a character vector, "none" returns nothing
silent	Suppress printing text to console? logical; default of false.

**Value**

Either nothing or a character vector of lines of R code for writing a testthat test, depending on return.style.

**Examples**

```
testmaker_df_names_tt(cars, return.style = "text")
```

---

testmaker_df_sin	<b>Deprecated</b> <i>Workhorse function to generate stopifnot tests for dataframe inputs</i>
------------------	--

---

**Description**

Generates R code to test expectations based on a template database. Intended workflow: when writing functions to work with a dataframe with a specific expected structure, load an example of that dataframe into the environment and then call `testmaker_df_sin(exampledataframe)`. This will print `stopifnot(...)` function calls based on the dimensions of the example data frame, the classes of the columns, and the names of the columns. Note that this is not intended to replace decision-making, but rather to streamline the process of generating the relevant code. In particular, testing the number of rows is often not appropriate (e.g., in many use cases the input data frame is expected to have unknown nrow).

**Usage**

```
testmaker_df_sin(
  x,
  return.style = c("clip", "text", "none"),
  silent = FALSE,
  object.name = "res",
  col.order.matters = TRUE
)
```

**Arguments**

<code>x</code>	Template dataframe of target object (e.g. we expect our test object to share properties with this)
<code>return.style</code>	Designation for what to return. "clip" returns nothing, but loads the text into the clipboard. "text" returns the text as a character vector, "none" returns nothing
<code>silent</code>	Suppress printing text to console? logical; default of false.
<code>object.name</code>	Name of the object to apply the stopifnot to; presumably the name of the dataframe argument
<code>col.order.matters</code>	When we generate code to test for column names, should we test for exact equivalence (TRUE), or just the presence of all the same column names (FALSE)?. Logical, defaults to TRUE

**Value**

Either nothing or a character vector of lines of R code for writing a testthat test, depending on `return.style`.

---

<code>testmaker_df_tt</code>	<i>Workhorse function to generate testthat tests for dataframe inputs</i>
------------------------------	---

---

**Description**

Generates R code to test expectations based on a template database. Intended workflow: when writing functions to work with a dataframe with a specific expected structure, load an example of that dataframe into the environment and then call `testmaker_df_tt(exampledataframe)`. This will print `testthat::expect_*()` function calls based on the dimensions of the example data frame, the classes of the columns, and the names of the columns. Note that this is not intended to replace decision-making, but rather to streamline the process of generating the relevant code. In particular, testing the number of rows is often not appropriate (e.g., in many use cases the input data frame is expected to have unknown `nrow`).

**Usage**

```
testmaker_df_tt(x, return.style = c("clip", "text", "none"), silent = FALSE)
```

**Arguments**

<code>x</code>	Template dataframe of target object (e.g. we expect our test object to share properties with this)
<code>return.style</code>	Designation for what to return. "clip" returns nothing, but loads the text into the clipboard. "text" returns the text as a character vector, "none" returns nothing
<code>silent</code>	Suppress printing text to console? logical; default of false.

**Value**

Either nothing or a character vector of lines of R code for writing a testthat test, depending on `return.style`.

**Examples**

```
## Not run:
testmaker_df(cars)

## End(Not run)
```

---

```
testmaker_df_validator
```

*Generate code for input-checking function based on cli\_abort framework*

---

**Description**

This function streamlines creating helper functions to check dataframe inputs, writing code based on the characteristics of a template dataframe. Intended use: when writing a function that takes a dataframe as an input, and you want to include an input check to confirm the input has the same # of columns, column names, and column classes as the dataframe you're using to develop your code. Providing informative error messages requires several lines of code, so it can be cleaner to write this input check as a separate function.

**Usage**

```
testmaker_df_validator(
  x,
  return.style = c("clip", "text", "none"),
  silent = FALSE,
  object.name = "x",
  col.order.matters = TRUE
)
```

**Arguments**

<code>x</code>	Template dataframe of target object (e.g. we expect our test object to share properties with this)
<code>return.style</code>	Designation for what to return. "clip" returns nothing, but loads the text into the clipboard. "text" returns the text as a character vector, "none" returns nothing
<code>silent</code>	Suppress printing text to console? logical; default of false.
<code>object.name</code>	Name of the object to apply the stopifnot to; presumably the name of the dataframe argument
<code>col.order.matters</code>	When we generate code to test for column names, should we test for exact equivalence (TRUE), or just the presence of all the same column names (FALSE)?. Logical, defaults to TRUE. FALSE LOGIC IS NOT CURRENTLY IMPLEMENTED!



**Details**

By default, this function copies the appropriate text to the system clipboard; all that is needed after running the function is to paste into the appropriate R file and provide a function name.

Note that `testmaker_df_cli` generates all the likely/common attribute checks, not all of which may be relevant for any specific use case(e.g., the generated code includes a check that the input dataframe contains the same number of rows as the template data frame; this is often not the expectation). Modify the resulting function template to match your specific needs. checking that the input has the ).

**Value**

Either nothing (when `return.style` is "clip" or "none"), or character string of R code.

**Examples**

```
testmaker_df_validator(cars, return.style = "none")
## Not run:
##Workflow version, which loads text into clipboard for easy pasting:
testmaker_df_validator(cars)

## End(Not run)
```

---

<code>validate_testmaker</code>	<i>Validation function for testmaker functions</i>
---------------------------------	--

---

**Description**

Internal helper function to streamline validation

**Usage**

```
validate_testmaker(
  x,
  return.style,
  silent,
  object.name = NULL,
  call = rlang::caller_env()
)
```

**Arguments**

<code>x</code>	Template dataframe of target object (e.g. we expect our test object to share properties with this)
<code>return.style</code>	Defines what is returned by testmaker functions. SHOULD be "clip", "text", or "none"
<code>silent</code>	Suppress printing text to console? logical; default of false.

<code>object.name</code>	Name of the object to apply the stopifnot to; presumably the name of the dataframe argument in the function the test is being written for. Defaults to NULL; provide when validating a <code>stopifnot_*</code> function.
<code>call</code>	Identifies the function calling this helper function, allowing more informative error messages

# Index

dput\_to\_string, [2](#)

finish\_testmaker, [3](#)

testmaker\_df\_class\_cli, [3](#)

testmaker\_df\_class\_sin, [4](#)

testmaker\_df\_class\_tt, [5](#)

testmaker\_df\_cli, [6](#)

testmaker\_df\_colcontent\_sin, [7](#)

testmaker\_df\_colcontent\_tt, [8](#)

testmaker\_df\_dim\_cli, [9](#)

testmaker\_df\_dim\_sin, [10](#)

testmaker\_df\_dim\_tt, [10](#)

testmaker\_df\_isit\_tt, [11](#)

testmaker\_df\_names\_sin, [12](#)

testmaker\_df\_names\_sin\_orderless, [13](#)

testmaker\_df\_names\_tt, [13](#)

testmaker\_df\_sin, [14](#)

testmaker\_df\_tt, [15](#)

testmaker\_df\_validator, [16](#)

validate\_testmaker, [17](#)