Package: rexer (via r-universe)

October 15, 2026

Type Package

Title Random Exercises and Exams Generator

Version 1.0.0.9000

Description The main purpose of this package is to streamline the generation of exams that include random elements in exercises. Exercises can be defined in a table, based on text and figures, and may contain gaps to be filled with provided options. Exam documents can be generated in various formats. It allows us to generate a version for conducting the assessment and another version that facilitates correction, linked through a code.

License MIT + file LICENSE

URL https://josesamos.github.io/rexer/, https://github.com/josesamos/rexer

BugReports https://github.com/josesamos/rexer/issues

Imports readr, readxl, rmarkdown, snakecase, stringr, tibble, tidyr, xlsx

Suggests knitr, pander, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

Encoding UTF-8

LazyData true

RoxygenNote 7.3.1

Config/pak/sysreqs cmake make default-jdk libicu-dev libuv1-dev libx11-dev

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

Date/Publication 2024-03-11 05:36:26 UTC

RemoteUrl https://github.com/josesamos/rexer

RemoteRef HEAD

RemoteSha 2f3059ad9d47a8ad63c4fd4d8f93e3b707acf46e

Contents

create_exercise_csv 2
create_exercise_data_frame 3
create_exercise_excel 3
define_an_exercise 4
define_exercises 5
define_exercises_from_csv 7
define_exercises_from_excel 8
exam 9
generate_correction_document 10
generate_document 12
read_exercise_csv 13
read_exercise_excel 14
set_pending_answers 15
vector_to_string 15
write_exercise_csv 16

Index **18**

`create_exercise_csv` *Create an exercise csv file*

Description

Creates an empty exercise csv file.

Usage

```
create_exercise_csv(file, sep = ",")
```

Arguments

`file`	A string, name of a text file.
`sep`	Column separator character.

Value

A string.

See Also

Other support functions: `create_exercise_data_frame()`, `create_exercise_excel()`, `read_exercise_csv()`, `read_exercise_excel()`, `set_pending_answers()`, `vector_to_string()`, `write_exercise_csv()`

Examples

```
file <- create_exercise_csv(file = tempfile(fileext = '.csv'))
```

create_exercise_data_frame *Create an exercise data frame*

Description

Creates an empty exercise data frame.

Usage

```
create_exercise_data_frame()
```

Value

A data frame.

See Also

Other support functions: `create_exercise_csv()`, `create_exercise_excel()`, `read_exercise_csv()`, `read_exercise_excel()`, `set_pending_answers()`, `vector_to_string()`, `write_exercise_csv()`

Examples

```
df <- create_exercise_data_frame()
```

create_exercise_excel *Create an exercise Excel file*

Description

Creates an empty exercise Excel file.

Usage

```
create_exercise_excel(file)
```

Arguments

`file`	A string, name of a text file.

Value

A string.

See Also

Other support functions: create_exercise_csv(), create_exercise_data_frame(), read_exercise_csv(), read_exercise_excel(), set_pending_answers(), vector_to_string(), write_exercise_csv()

Examples

```
file <- create_exercise_excel(file = tempfile(fileext = '.xlsx'))
```

---

define_an_exercise *Define an exercise*

Description

Defines an exercise with random components.

Usage

```
define_an_exercise(ex, type, statement, image, image_alt, answer, ...)

## S3 method for class 'exam'
define_an_exercise(
  ex,
  type = "",
  statement = "",
  image = "",
  image_alt = "",
  answer = "",
  ...
)
```

Arguments

ex	An ‘exam‘ object.
type	A character, ’p’ indicates whether the exercise starts on a new page.
statement	A string, statement of the exercise.
image	A string, optional, image file to include in the exercise.
image_alt	A string, description of the image to include in the exercise.
answer	A string, correct answer to the exercise.
...	A string, options for the gaps in the exercise.

Details

If an image is included in the exercise, text in the ‘image_alt‘ field associated with it must also be included.

Following the answer, options can be defined to fill in the gaps that have been specified in the exercise.

Both the answer and the options are formed by a vector of strings, from which one is chosen to formulate the exercise and select the answer. To represent a vector of strings in a cell, the function ‘vector_to_string()‘ is used, which includes a separator ("<|>") between the vector elements to generate a string.

Value

An ‘exam‘.

See Also

exam, vector_to_string

Other exercise definition: define_exercises(), define_exercises_from_csv(), define_exercises_from_excel()

Examples

```
rmd <- system.file("extdata/template01.Rmd", package = "rexer")
ex <- exam(
  rmd = rmd,
  examinees = NULL,
  instances_num = 10,
  random = TRUE,
  reorder_exercises = TRUE,
  select_n_exercises = NULL
) |>
  define_an_exercise(
    type = 'p',
    statement = 'What is the three-letter country code (ISO 3166-1 alpha-3) for
    the country represented in the figure below?',
    image = paste0(system.file("extdata/figures", package = "rexer"), "/", '[[1]]'),
    image_alt = 'Country outline.',
    answer = c('ESP', 'CHL', 'NZL', 'ITA'),
    c('spain.png', 'chile.png', 'new_zealand.png', 'italy.png')
  )
```

---

define_exercises *Define exercises from a data frame*

Description

Each row in the text data frame is interpreted as an exercise. We only need to define the columns that we are going to use; the rest of the columns are taken by default.

Usage

```
define_exercises(ex, df)

## S3 method for class 'exam'
define_exercises(ex, df)
```

Arguments

ex	An ‘exam‘ object.
df	A data frame containing exercises.

Details

Both the answer and the options are formed by a vector of strings, from which one is chosen to formulate the exercise and select the answer. To represent a vector of strings in a cell, the function ‘vector_to_string()‘ is used, which includes a separator ("<|>") between the vector elements to generate a string.

Value

An ‘exam‘.

See Also

exam, vector_to_string

Other exercise definition: define_an_exercise(), define_exercises_from_csv(), define_exercises_from_excel()

Examples

```
rmd <- system.file("extdata/template01.Rmd", package = "rexer")
exercises <- system.file("extdata/exercises.csv", package = "rexer")
q <- read_exercise_csv(exercises)
ex <- exam(
  rmd = rmd,
  examinees = NULL,
  instances_num = 10,
  random = TRUE,
  reorder_exercises = TRUE,
  select_n_exercises = NULL
) |>
  define_exercises(q)
```

`define_exercises_from_csv` *Define exercises from a csv file*

Description

Each row in the text file is interpreted as an exercise. We only need to define the columns that we are going to use; the rest of the columns are taken by default.

Usage

```
define_exercises_from_csv(ex, file, sep)

## S3 method for class 'exam'
define_exercises_from_csv(ex, file, sep = ",")
```

Arguments

`ex`	An ‘exam‘ object.
`file`	A string, name of a text file.
`sep`	Column separator character ("," or ";").

Details

Both the answer and the options are formed by a vector of strings, from which one is chosen to formulate the exercise and select the answer. To represent a vector of strings in a cell, the function ‘vector_to_string()‘ is used, which includes a separator ("<|>") between the vector elements to generate a string.

Value

An ‘exam‘.

See Also

`exam`, `vector_to_string`

Other exercise definition: `define_an_exercise()`, `define_exercises()`, `define_exercises_from_excel()`

Examples

```
rmd <- system.file("extdata/template01.Rmd", package = "rexer")
exercises <- system.file("extdata/exercises.csv", package = "rexer")
ex <- exam(
  rmd = rmd,
  examinees = NULL,
  instances_num = 10,
  random = TRUE,
```

```
  reorder_exercises = TRUE,
  select_n_exercises = NULL
) |>
  define_exercises_from_csv(exercises)
```

`define_exercises_from_excel` *Define exercises from a Excel file*

Description

Each row in the Excel file is interpreted as an exercise. We only need to define the columns that we are going to use; the rest of the columns are taken by default.

Usage

```
define_exercises_from_excel(ex, file, sheet_index, sheet_name)

## S3 method for class 'exam'
define_exercises_from_excel(ex, file, sheet_index = NULL, sheet_name = NULL)
```

Arguments

`ex`	An ‘exam‘ object.
`file`	A string, name of an Excel file.
`sheet_index`	A number, sheet index in the workbook.
`sheet_name`	A string, sheet name.

Details

In addition to the file, we can indicate the sheet by its name or index. If we do not indicate anything, it considers the first sheet.

Both the answer and the options are formed by a vector of strings, from which one is chosen to formulate the exercise and select the answer. To represent a vector of strings in a cell, the function ‘vector_to_string()‘ is used, which includes a separator ("<|>") between the vector elements to generate a string.

Value

An ‘exam‘.

See Also

`exam`, `vector_to_string`

Other exercise definition: `define_an_exercise()`, `define_exercises()`, `define_exercises_from_csv()`

Examples

```
rmd <- system.file("extdata/template01.Rmd", package = "rexer")
exercises <- system.file("extdata/exercises.xlsx", package = "rexer")
ex <- exam(
  rmd = rmd,
  examinees = NULL,
  instances_num = 10,
  random = TRUE,
  reorder_exercises = TRUE,
  select_n_exercises = NULL
) |>
  define_exercises_from_excel(exercises)
```

`exam` *'exam' S3 class*

Description

Creates an 'exam' object.

Usage

```
exam(
  rmd = NULL,
  examinees = NULL,
  instances_num = 1,
  random = TRUE,
  reorder_exercises = TRUE,
  select_n_exercises = NULL
)
```

Arguments

`rmd`	A string representing the path to the Rmd file, the exam template.
`examinees`	A vector of strings, representing the names of instances to generate.
`instances_num`	An integer, representing the number of instances to generate if the examinee names are not provided.
`random`	A boolean, indicating whether to generate instances randomly or sequentially.
`reorder_exercises`	A boolean, indicating whether to reorder exercises in the exam.
`select_n_exercises`	An integer, representing the number of exercises to include.

Details

A test is based on an Rmd template that has predefined parameters whose values are filled in using the functions of this object. In the ‘rmd‘ parameter, we specify the template file.

From the template, we generate multiple instances of the exam. We can specify the instances to generate in two ways: by indicating a vector of examinee names (using the ‘examinees‘ parameter) or by specifying the number of instances to generate (using the ‘instances_num‘ parameter). If both are indicated, the examinee names take precedence.

We can generate the instances either randomly or sequentially, depending on the instance number we generate. This is controlled by the ‘random‘ parameter.

Additionally, in each test, we can include the exercises in the same order as they are defined or in random order. This is indicated by the ‘reorder_exercises‘ parameter.

Finally, using the ‘select_n_exercises‘ parameter, we can specify the number of exercises to include in each test. From all available exercises, the quantity specified in this parameter will be randomly selected. By default, all defined exercises are included.

Value

An ‘exam‘ object.

See Also

`define_an_exercise`

Other exam definition: `generate_correction_document()`, `generate_document()`

Examples

```
rmd <- system.file("extdata/template01.Rmd", package = "rexer")
ex <- exam(
  rmd = rmd,
  examinees = NULL,
  instances_num = 10,
  random = TRUE,
  reorder_exercises = TRUE,
  select_n_exercises = NULL
)
```

---

generate_correction_document

Generate the support document for exam correction

Description

From an exam object, we can generate instances that serve as support for the correction of the exam. Each instance will include the answers, if they are indicated, associated with the exercises. In any case, the randomly included part of the exam will be highlighted.

Usage

```
generate_correction_document(ex, out_dir, output_format, encoding, new_pages)

## S3 method for class 'exam'
generate_correction_document(
  ex,
  out_dir = NULL,
  output_format = "pdf_document",
  encoding = "UTF-8",
  new_pages = NULL
)
```

Arguments

ex	An ‘exam‘ object.
out_dir	A string indicating the output folder.
output_format	A vector of strings specifying the desired output formats.
encoding	A string specifying the encoding.
new_pages	A string with the values ’all’, ’none’, or ’NULL’.

Details

To do this, we specify the folder where the documents will be generated (using parameter ‘out_dir‘), the output format (using parameter ‘output_format‘), the encoding (using parameter ‘encoding‘), and whether we want each exercise to start on a new page, include exercises until the pages are filled, or preserve the definition of the exercise in this regard (using parameter ‘new_pages‘).

Value

An ‘exam‘ object.

See Also

Other exam definition: exam(), generate_document()

Examples

```
## Not run:
rmd <- system.file("extdata/template01.Rmd", package = "rexer")
exercises <- system.file("extdata/exercises.csv", package = "rexer")
ex <- exam(
  rmd = rmd,
  examinees = NULL,
  instances_num = 1,
  random = TRUE,
  reorder_exercises = TRUE
) |>
  define_exercises_from_csv(exercises) |>
  generate_correction_document(out_dir = tempdir())
```

```
## End(Not run)
```

`generate_document` *Generate the exam document*

Description

From an exam object, we generate different instances of the exam to deliver to the individuals being examined. To do this, we need to specify the folder where they will be generated (using parameter 'out_dir'), the output format (using parameter 'output_format'), the encoding (using parameter 'encoding'), and whether we want each exercise to start on a new page, include exercises until the pages are filled, or preserve the definition of the exercise in this regard (using parameter 'new_pages' with the values 'all', 'none', or NULL).

Usage

```
generate_document(ex, out_dir, output_format, encoding, new_pages)

## S3 method for class 'exam'
generate_document(
  ex,
  out_dir = NULL,
  output_format = "pdf_document",
  encoding = "UTF-8",
  new_pages = NULL
)
```

Arguments

`ex`	An 'exam' object.
`out_dir`	A string indicating the output folder.
`output_format`	A vector of strings specifying the desired output formats.
`encoding`	A string specifying the encoding.
`new_pages`	A string with the values 'all', 'none', or NULL.

Value

An 'exam' object.

See Also

Other exam definition: `exam()`, `generate_correction_document()`

Examples

```
## Not run:
rmd <- system.file("extdata/template01.Rmd", package = "rexer")
exercises <- system.file("extdata/exercises.csv", package = "rexer")
ex <- exam(
  rmd = rmd,
  examinees = NULL,
  instances_num = 1,
  random = TRUE,
  reorder_exercises = TRUE
) |>
  define_exercises_from_csv(exercises) |>
  generate_document(out_dir = tempdir(), new_pages = 'all')

## End(Not run)
```

---

read_exercise_csv *Read an exercise csv file*

Description

Reads a csv file of exercises and returns a data frame.

Usage

```
read_exercise_csv(file, sep = ",")
```

Arguments

file	A string, name of a text file.
sep	Column separator character.

Value

A data frame.

See Also

Other support functions: create_exercise_csv(), create_exercise_data_frame(), create_exercise_excel(), read_exercise_excel(), set_pending_answers(), vector_to_string(), write_exercise_csv()

Examples

```
file <- system.file("extdata/exercises.csv", package = "rexer")
df <- read_exercise_csv(file)
```

`read_exercise_excel` *Read an exercise Excel file*

Description

Reads an Excel file of exercises and returns a data frame.

Usage

```
read_exercise_excel(file, sheet_index = NULL, sheet_name = NULL)
```

Arguments

`file`	A string, name of a text file.
`sheet_index`	A number, sheet index in the workbook.
`sheet_name`	A string, sheet name.

Details

In addition to the file, we can indicate the sheet by its name or index. If we do not indicate anything, it considers the first sheet.

Value

A data frame.

See Also

Other support functions: `create_exercise_csv()`, `create_exercise_data_frame()`, `create_exercise_excel()`, `read_exercise_csv()`, `set_pending_answers()`, `vector_to_string()`, `write_exercise_csv()`

Examples

```
file <- system.file("extdata/exercises.xlsx", package = "rexer")
df <- read_exercise_excel(file)
```

set_pending_answers *Set pending answers*

Description

The answers we want to process are indicated by the character "?". Fill in the specified answers by generating combinations of the available options to fill the gaps in order, from value 1 to the maximum number of options available: the first option for each of the gaps, the second for each of the gaps, and so on.

Usage

```
set_pending_answers(df)
```

Arguments

df	A data frame of exercises.

Value

A data frame.

See Also

Other support functions: create_exercise_csv(), create_exercise_data_frame(), create_exercise_excel(), read_exercise_csv(), read_exercise_excel(), vector_to_string(), write_exercise_csv()

Examples

```
file <- system.file("extdata/example.csv", package = "rexer")
df <- read_exercise_csv(file)
df <- set_pending_answers(df)
```

vector_to_string *Transforms a vector of strings into a string*

Description

Insert the separator ("<|>") to later facilitate the reverse operation.

Usage

```
vector_to_string(vector)
```

Arguments

vector	A vector of strings.

Value

A string.

See Also

Other support functions: create_exercise_csv(), create_exercise_data_frame(), create_exercise_excel(), read_exercise_csv(), read_exercise_excel(), set_pending_answers(), write_exercise_csv()

Examples

```
s <- vector_to_string(c('Addition', '+'))
```

---

write_exercise_csv *Write an exercise csv file*

Description

Writes an exercise data frame in a csv file of exercises.

Usage

```
write_exercise_csv(df, file, sep = ",")
```

Arguments

df	A data frame.
file	A string, name of a text file.
sep	Column separator character.

Value

A string.

See Also

Other support functions: create_exercise_csv(), create_exercise_data_frame(), create_exercise_excel(), read_exercise_csv(), read_exercise_excel(), set_pending_answers(), vector_to_string()

Examples

```
file <- system.file("extdata/example.csv", package = "rexer")
df <- read_exercise_csv(file)
df <- set_pending_answers(df)

write_exercise_csv(df, file = tempfile(fileext = '.csv'))
```

Index

∗ **exam definition**
exam, 9
generate_correction_document, 10
generate_document, 12
∗ **exercise definition**
define_an_exercise, 4
define_exercises, 5
define_exercises_from_csv, 7
define_exercises_from_excel, 8
∗ **support functions**
create_exercise_csv, 2
create_exercise_data_frame, 3
create_exercise_excel, 3
read_exercise_csv, 13
read_exercise_excel, 14
set_pending_answers, 15
vector_to_string, 15
write_exercise_csv, 16

create_exercise_csv, 2, *3, 4*, *13–16*
create_exercise_data_frame, *2*, 3, *4*, *13–16*
create_exercise_excel, *2*, 3, *3*, *13–16*

define_an_exercise, 4, *6–8*, *10*
define_exercises, 5, *5*, *7, 8*
define_exercises_from_csv, *5, 6*, 7, *8*
define_exercises_from_excel, *5–7*, 8

exam, *5–8*, 9, *11, 12*

generate_correction_document, 10, *10*, *12*
generate_document, *10, 11*, 12

read_exercise_csv, *2–4*, 13, *14–16*
read_exercise_excel, *2–4*, *13*, 14, *15, 16*

set_pending_answers, *2–4*, *13, 14*, 15, *16*

vector_to_string, *2–8*, *13, 14*, 15, *15, 16*

write_exercise_csv, *2–4*, *13–15*, 16, *16*