

# Package ‘RndTexExams’

January 13, 2018

**Title** Build and Grade Multiple Choice Exams with Randomized Content

**Version** 1.4.1

**Description** Using as input a 'LaTeX' file with a multiple choice exam, this package will produce several versions with randomized contents of the same exam. Functions for grading and testing for cheating are also available.

**Depends** R (>= 3.2.3), CopyDetect

**Imports** tools, stringr, stringi, data.table

**License** GPL-2

**LazyData** true

**RoxygenNote** 6.0.1

**Suggests** knitr, rmarkdown, testthat, ggplot2, irtoys

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** Marcelo Perlin [aut, cre]

**Maintainer** Marcelo Perlin <marceloperlin@gmail.com>

**Repository** CRAN

**Date/Publication** 2018-01-13 17:36:28 UTC

## R topics documented:

engine.analyze.class.exam . . . . .	2
engine.analyze.class.examdesign . . . . .	2
rte.analyze.tex.file . . . . .	3
rte.build.rdn.test . . . . .	4
rte.check.latex.flavor . . . . .	5
rte.check.my.os . . . . .	6
rte.check.pdflatex . . . . .	6
rte.compile.latex . . . . .	7
rte.get.classes.def . . . . .	8
rte.get.n.cases . . . . .	8
rte.grade.exams . . . . .	9
rte.test.cheating . . . . .	11
rte.The.Randomizer . . . . .	12

**Index****14**

---

`engine.analyze.class.exam`

*Function that breaks latex file of exam class exam into a dataframe (internal use)*

---

**Description**

Function that breaks latex file of exam class exam into a dataframe (internal use)

**Usage**

```
engine.analyze.class.exam(my.text)
```

**Arguments**

`my.text`           Text of latex file written as an exam template (UTF-8)

**Value**

A dataframe with several information (see `rte.analyze.tex.file`)

**Examples**

```
f.in <- system.file("extdata", "MyRandomTest_exam.tex", package = "RndTexExams")
my.text <- stringi::stri_read_lines(f.in)
Encoding(my.text) <- 'UTF-8'
```

```
out <- engine.analyze.class.exam(my.text)
```

---

`engine.analyze.class.examdesign`

*Function that breaks latex file of exam class examdesign into a dataframe (internal use)*

---

**Description**

Function that breaks latex file of exam class examdesign into a dataframe (internal use)

**Usage**

```
engine.analyze.class.examdesign(my.text)
```

**Arguments**

`my.text`           Text of latex file written in class examdesign (UTF-8)

**Value**

A dataframe with several information (see `rte.analyze.tex.file`)

**Examples**

```
f.in <- system.file("extdata", "MyRandomTest_examdesign.tex", package = "RndTexExams")
my.text <- stringi::stri_read_lines(f.in)
Encoding(my.text) <- 'UTF-8'

out <- engine.analyze.class.examdesign(my.text)
```

---

`rte.analyze.tex.file` *Analyze a LaTeX file and convert it into a list*

---

**Description**

This function will take as input a LaTeX file and break its components into a single R List. The class of the latex file should be either exam or examdesign. The code identifies the latex class automatically.

**Usage**

```
rte.analyze.tex.file(f.in, latex.dir.out = "latexOut",
  pdf.dir.out = "PdfOut")
```

**Arguments**

<code>f.in</code>	The latex file with the exam
<code>latex.dir.out</code>	The name of the folder where the files from the latex compilation should go (will create if not found)
<code>pdf.dir.out</code>	The name of the folder where the pdf from the latex compilation should go (will create if not found)

**Value**

A list that represents the tex file with preamble, questions, answers and more. This list is later used by function `rte.build.rdn.text`

**df.questions** A data.frame with all questions

**df.answers** A data.frame with all answers

**my.begin.mchoice.line** text with beggining of mchoice enviroment

**my.preamble** preamble of tex file, including everything before the beggining of the multiple choice enviroment

**my.last.part** All of the tex code after the end of the multiple choice enviroment

**Examples**

```

latex.dir.out <- 'latexOut' # Name of folder where latex files are going
                        #(will create if it does not exists)

pdf.dir.out <- 'PdfOut'    # Name of folder where resulting pdf files are going

# Get latex example from package
f.in <- system.file("extdata", "MyRandomTest_examdesign.tex", package = "RndTexExams")

# Break latex file into a R list
list.out <- rte.analyze.tex.file(f.in,
                                latex.dir.out = latex.dir.out,
                                pdf.dir.out = pdf.dir.out)

print(list.out)

```

---

rte.build.rdn.test      *Build random tests from LaTeX file*

---

**Description**

This function will take as input a list from `rte.analyze.tex.file` and use it to build pdf files of random exams. See the package vignette for details on how to use it.

**Usage**

```

rte.build.rdn.test(list.in, f.out, n.test, n.question,
                  latex.dir.out = "latexOut", pdf.dir.out = "PdfOut",
                  latex.compile.fct = "custom", do.randomize.questions = T,
                  do.randomize.answers = T, do.clean.up = T)

```

**Arguments**

<code>list.in</code>	A list with all the information of the LaTeX file. Usually the output from function <code>rte.analyze.tex.file()</code>
<code>f.out</code>	The name for the pdf files (e.g. using <code>f.out &lt;- 'RdnTest_'</code> , the code will create files <code>'RdnTest_1.pdf'</code> , <code>'RdnTest_2.pdf'</code> , and so on)
<code>n.test</code>	The number of random exams to be build (usually the number of students in class)
<code>n.question</code>	The number of questions in each exam (If the LaTeX file has N questions, the code will randomly select <code>n.question</code> of these)
<code>latex.dir.out</code>	The name of the folder where the files from the latex compilation should go (will create if not found)
<code>pdf.dir.out</code>	The name of the folder where the pdf files from the latex compilation should go (will create if not found)

```

latex.compile.fct      Option for defining function that calls pdflatex: 'texi2pdf' or 'custom' (default).
do.randomize.questions Do you want the order of the questions to be random? (TRUE or FALSE)
do.randomize.answers   Do you want the order of the answers to be random? (TRUE or FALSE)
do.clean.up            Should R clean up all extra files from the LaTeX compilations and leave only
                      the pdf? (select FALSE if you want see the log files from latex)

```

**Value**

A list with the following items:

**df.answer.wide** A dataframe with the tests, order of questions and correct answers

**answer.matrix** A matrix with the correct answers (rows = version, columns = questions)

**Examples**

```

# define some options
latex.dir.out = 'latexOut' # Name of folder where latex files are going (will create if not exists)
pdf.dir.out = 'PdfOut'    # Name of folder where resulting pdf files are going
f.out <- 'MyRandomTest_'  # Name of pdfs (MyRandomTest_1.pdf, MyRandomTest_2.pdf, ... )
n.test <- 1               # Number of tests to build
n.question <- 2           # Number of questions in each test

# Get latex example from package
f.in <- system.file("extdata", "MyRandomTest_examdesign.tex", package = "RndTexExams")

# Break latex file into a R list
list.out <- rte.analyze.tex.file(f.in,
                                latex.dir.out = latex.dir.out,
                                pdf.dir.out = pdf.dir.out)

# Build pdfs
result.out <- rte.build.rdn.test(list.in = list.out,
                                f.out = f.out,
                                n.test = n.test,
                                n.question = n.question,
                                latex.dir.out = latex.dir.out,
                                pdf.dir.out = pdf.dir.out)

```

---

```
rte.check.latex.flavor
```

*Function to check the distribution of LaTeX*

---

**Description**

Function to check the distribution of LaTeX

**Usage**

```
rte.check.latex.flavor()
```

**Value**

The flavor of latex instlation (e.g. miktex, texlive)

**Examples**

```
rte.check.latex.flavor()
```

---

```
rte.check.my.os
```

*Function to check operating system of user*

---

**Description**

Function to check operating system of user

**Usage**

```
rte.check.my.os()
```

**Value**

A string with the name of the operating system (e.g. Windows)

**Examples**

```
rte.check.my.os()
```

---

```
rte.check.pdflatex
```

*Function to check if system has pdflatex.exe available*

---

**Description**

Function to check if system has pdflatex.exe available

**Usage**

```
rte.check.pdflatex()
```

**Value**

TRUE if the pdflatex is available, FALSE if not

## Examples

```
rte.check.pdflatex()
```

---

rte.compile.latex	<i>Function to compile a LaTeX file</i>
-------------------	-----------------------------------------

---

## Description

This function will first check for the flavor of latex, type of OS and then use the proper command for pdflatex compilation

## Usage

```
rte.compile.latex(f.in, pdf.dir.out = "PdfOut", do.clean.up = T,  
  latex.compile.fct = "texi2pdf")
```

## Arguments

f.in	The location and name of latex file
pdf.dir.out	The name of the folder for the output pdf
do.clean.up	Clean (delete) auxiliary latex files? (TRUE or NOT)
latex.compile.fct	Option for function compiling pdf ('texi2pdf' or 'custom')

## Value

A flag, TRUE if the latex compilation was a success and FALSE if not

## Examples

```
f.in <- system.file("extdata", "MyRandomTest_examdesign.tex", package = "RndTexExams")  
pdf.dir.out <- 'PdfOut'  
  
rte.compile.latex(f.in = f.in,  
  pdf.dir.out = pdf.dir.out, latex.compile.fct = 'custom')
```

---

`rte.get.classes.def`     *Function that returns parameters of latex classes (internal use)*

---

### Description

This function outputs the main elements of each latex class such as question/choices identifiers and environmental switches.

### Usage

```
rte.get.classes.def(exam.class)
```

### Arguments

`exam.class`     The class of the exam (exam, examdesign)

### Value

A list with parameters of the class

### Examples

```
l.out <- rte.get.classes.def(exam.class = 'exam')
print(l.out)
```

---

`rte.get.n.cases`     *Function to get number of cases (textual switches) based on question text*

---

### Description

Function to get number of cases (textual switches) based on question text

### Usage

```
rte.get.n.cases(str.in)
```

### Arguments

`str.in`     The full text of the question

### Value

`n.cases` Number of cases (textual switches in string)



**Examples**

```
my.question <- 'My questions is ... @{\ver1}|{\ver2}@' ## two cases
n.cases <- rte.get.n.cases(my.question)
```

---

rte.grade.exams	<i>Grade exams built using rte.grade.exams</i>
-----------------	------------------------------------------------

---

**Description**

This function will take as input the information from the exam and grade it using the framework of **RndTexExams**

**Usage**

```
rte.grade.exams(exam.names, exam.version, exam.answer.matrix,
  list.build.rdn.exam, question.points = NULL)
```

**Arguments**

exam.names	A character vector with the names of the students, obtained from the test
exam.version	A numeric vector with the version of the exam for each student, obtained from the exam
exam.answer.matrix	A matrix with the answers of the students where the rows represent each student and the columns are the answers to each question
list.build.rdn.exam	A list with several information of the random exams (output from rte.build.rdn.text)
question.points	A numeric vector with the score for each question (default = 1/n.question)

**Value**

A list with the following items:

**df.grade** A dataframe with the partial results from grading

**df.final.score** A dataframe with the final results for each student

**Examples**

```

# define some options
latex.dir.out = 'latexOut' # Name of folder where latex files are going (will create if not exists)
pdf.dir.out = 'PdfOut'    # Name of folder where resulting pdf files are going
f.out <- 'MyRandomTest_'  # Name of pdfs (MyRandomTest_1.pdf, MyRandomTest_2.pdf, ... )
n.test <- 1               # Number of tests to build
n.question <- 2           # Number of questions in each test

# Get latex example from package
f.in <- system.file("extdata", "MyRandomTest_examdesign.tex", package = "RndTexExams")

# Break latex file into a R list
list.out <- rte.analyze.tex.file(f.in,
                                latex.dir.out = latex.dir.out,
                                pdf.dir.out = pdf.dir.out)

# Build pdfs
list.build.rdn.exam <- rte.build.rdn.test(list.in = list.out,
                                          f.out = f.out,
                                          n.test = n.test,
                                          n.question = n.question,
                                          latex.dir.out = latex.dir.out,
                                          pdf.dir.out = pdf.dir.out,
                                          do.randomize.questions=TRUE,
                                          do.randomize.answers=TRUE,
                                          do.clean.up = TRUE)

# Grade it!
#' # create some (almost) random names
my.names <- c('John', 'Max', 'Marcelo')

# version of the test for each student
ver.test <- sample(seq(n.test), size = length(my.names), replace=TRUE)

# Get the correct answer sheet from previous code
correct.answer.sheet <- list.build.rdn.exam$answer.matrix

# create simulated answers from students (cheat a little bit!)
q.to.cheat <- 1 # get at least 1 question right!
my.answers <- cbind(correct.answer.sheet[ver.test, 1:q.to.cheat],
                    matrix(sample(letters[1:5],
                                  replace = TRUE,
                                  size = length(my.names)*(n.question-q.to.cheat)),
                            ncol = n.question-q.to.cheat ))

# grade exams with rte.grade.exams
list.grade <- rte.grade.exams(exam.names = my.names,
                              exam.version = ver.test,
                              exam.answer.matrix = my.answers,
                              list.build.rdn.exam = list.build.rdn.exam)

print(list.grade$df.final.score)

```

---

rte.test.cheating      *Performs statistical tests for cheating using **CopyDetect***

---

## Description

Uses function `CopyDetect1` from package **CopyDetect** to test for visual cheating in exams based on correct answers from the students. The statistical tests performed by `CopyDetect1` are:

- Omega index (Wollack, 1996)
- Generalized Binomial Test ([GBT], van der Linden & Sotaridona (2006)
- K index (Holland, 1996)
- K1 and K2 indices (Sotaridona & Meijer, 2002)
- S1 and S2 indices (Sotaridona & Meijer, 2003)

The function `rte.test.cheating` will have as input a dataframe with the names and corrections of students and output a summary of the cheating tests as a list, including suspicious pairs.

## Usage

```
rte.test.cheating(df.grade, p.level = 0.05, print.suspects = TRUE,
  do.cheat.plot = TRUE, suspicion.threshold = 0.5)
```

## Arguments

<code>df.grade</code>	A dataframe where first column is the name of students (item <code>exam.names</code> ) and the rest of the columns are the correct (TRUE) and incorrect (FALSE) answers. Each column other than <code>exam.names</code> should be a question
<code>p.level</code>	Critical level of statistical testing
<code>print.suspects</code>	Print testing information and suspects on screen ? (TRUE or FALSE) (Default=TRUE)
<code>do.cheat.plot</code>	Print plot of cheating tests? (TRUE or FALSE) (Default=TRUE)
<code>suspicion.threshold</code>	Proportion of failed cheating tests that justify suspicion, between 0 and 1

## Details

More details regarding the tests can be found in:

Zopluoglu, C. (2013). CopyDetect An R Package for Computing Statistical Indices to Detect Answer Copying on Multiple-Choice Examinations. *Applied psychological measurement*, 37(1), 93-95.

The article can be found [here](#)

**Value**

A list with the following items:

**df.pvalue** A dataframe with the statistical results for all pairs of students from the upper triangle  
(1 test for each pair)

**df.suspects** A dataframe with the suspicious pair of students

**Examples**

```
# number of simulated questions in exam
n.sim.questions <- 10

base.names <- c('John', 'Marcelo', 'Ricardo', 'Tarcizio')
last.names <- c('Smith', 'P.')

name.grid <- expand.grid(base.names, last.names)

my.names <- paste(name.grid[,1], name.grid[,2])
# official names from the university system (will assume it is equal to my.names)
# In a practical situation, this list of official names will come from the university system
exam.names <- my.names

set.seed(10)

correction.mat <- matrix(sample(c(TRUE, FALSE),
                              size = length(exam.names)*n.sim.questions,
                              replace = TRUE), nrow = length(exam.names))

idx.cheater.1 <- 5 # std 5 and 6 have simillar correct answers
idx.cheater.2 <- 6
proportion.to.cheat <- 0.5 # proportion of same correct answers
q.to.cheat <- floor(proportion.to.cheat*n.sim.questions)
correction.mat[idx.cheater.1, ] <- c(rep(TRUE, q.to.cheat),
                                   rep(FALSE, n.sim.questions-q.to.cheat))

correction.mat[idx.cheater.2, ] <- correction.mat[idx.cheater.1, ]

df.grade <- cbind(data.frame(exam.names), correction.mat)

test.cheating.out <- rte.test.cheating(df.grade, do.cheat.plot = FALSE )
```

---

rte.The.Randomizer      *Function to randomize a question in a dataframe (internal use)*

---

**Description**

Function to randomize a question in a dataframe (internal use)

**Usage**

```
rte.The.Randomizer(q.text, q.answers, case.now, my.rdn.idx.answers)
```

**Arguments**

<code>q.text</code>	Main text of question (character)
<code>q.answers</code>	The answers of the questions as an atomic vector (each alternative as an item)
<code>case.now</code>	The random case to build the string (only used for textual switches)
<code>my.rdn.idx.answers</code>	The random index of the questions

**Value**

A list with the text of the full questions, among other things

**Examples**

```
q.text <- '\\question Whats my name \\n'  
q.answers <- c('\\choice Mario', '\\choice Roberto', '\\choice Marcelo')  
  
case.now <- 1  
my.rdn.idx.answers <- sample(seq(length(q.answers)))  
  
l.out <- rte.The.Randomizer(q.text,q.answers, case.now, my.rdn.idx.answers)
```

# Index

CopyDetect1, [11](#)

engine.analyze.class.exam, [2](#)

engine.analyze.class.examdesign, [2](#)

rte.analyze.tex.file, [3](#)

rte.build.rdn.test, [4](#)

rte.check.latex.flavor, [5](#)

rte.check.my.os, [6](#)

rte.check.pdflatex, [6](#)

rte.compile.latex, [7](#)

rte.get.classes.def, [8](#)

rte.get.n.cases, [8](#)

rte.grade.exams, [9](#)

rte.test.cheating, [11](#)

rte.The.Randomizer, [12](#)