Package: c403 (via r-universe)

September 12, 2024

Version 0.9-3

Date 2024-08-04

Title Exam Tools for Department of Statistics (c403), Uni Innsbruck

Description Support tools for managing lectures and exams at Uni Innsbruck, specifically for automatic generation and evaluation of mathematics and statistics exams.

Depends R (>= 4.0.0), exams (>= 2.3-1)

Suggests base64enc, openxlsx, psychotools, tth, xlsx, XML, knitr, rmarkdown

Imports graphics, grDevices, stats, tools, utils, xml2

SystemRequirements pandoc (>= 2.0)

License GPL-2 | GPL-3

RoxygenNote 7.2.3

VignetteBuilder knitr

Encoding UTF-8

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

RemoteUrl https://github.com/r-forge/exams

RemoteRef HEAD

RemoteSha 184988d103e30da57ecbfefc840d107576c1cc99

Contents

exams2nops *Deprecated: Generation of NOPS Exams (Uni Innsbruck)*

Description

Unexported legacy interface to `exams2nops` with different default values as used at Uni Innsbruck. Instead it is recommended to use `exams2nops` directly.

Usage

```
exams2nops(
  file,
  n = 1L,
  dir = NULL,
  name = NULL,
  language = "de",
  title = "Klausur",
  course = "",
  institution = "Universit\\\"at Innsbruck",
  logo = "uibk-logo-bw.png",
  date = Sys.Date(),
  replacement = TRUE,
  intro = NULL,
  blank = NULL,
  duplex = TRUE,
  pages = NULL,
  usepackage = NULL,
  encoding = "",
  startid = 1L,
  points = NULL,
  showpoints = FALSE,
  reglength = 8L,
  ...
)
```

Arguments

`file` character. A specification of a (list of) exercise files.

`n` integer. The number of copies to be compiled from `file`.

`dir` character. The default is either display on the screen or the current working directory.

`name` character. A name prefix for resulting exercises and RDS file.

`language` character. Path to a DCF file with a language specification. Currently, `"en"` and `"de"` are shipped with the **exams** package.

`title` character. Title of the exam, e.g., `"Statistische Datenanalyse"`.

`course` character. Course number, e.g., `"403001"`.

`institution` character. Name of the institution at which the exam is conducted.

`logo` character. Path to a logo image. If the logo is not found, it is simply omitted.

`date` character or `"Date"` object specifying the date of the exam.

`replacement` logical. Should a replacement exam sheet be included?

`intro` character with LaTeX code for introduction text at the beginning of the exam.

`blank` integer. Number of blank pages to be added at the end. (Default is chosen to be half of the number of exercises.)

`duplex` logical. Should blank pages be added after the title page (for duplex printing)?

`pages` character. Path(s) to additional PDF pages to be included at the end of the exam.

`usepackage` character. Names of additional LaTeX packages to be included.

`encoding` character, passed to `Sweave`.

`startid` integer. Starting ID for the exam numbers (defaults to 1).

`points` integer. How many points should be assigned to each exercise? Note that this argument overules any exercise points that are provided within the `expoints` tags of the exercise files (if any). The vector of points supplied should either have length 1 or the number of exercises in the exam.

`showpoints` logical. Should the PDF show the number of points associated with each exercise (if specified in the Rnw/Rmd exercise or in `points`)?

`reglength` integer. Number of digits in the registration ID. The default is 8 and it can be increased up to 10.

`...` arguments passed on to `exams2pdf`.

Details

`exams2nops` is a convenience interface for `exams2nops` with somewhat different defaults: German titles/descriptions, Uni Innsbruck logo, replacement sheets enabled, duplex printing enabled.

Value

A list of exams as generated by `xexams` is returned invisibly.

See Also

nops_eval nops_register

Examples

```
## load package and enforce par(ask = FALSE)
library("exams")
options(device.ask.default = FALSE)

## define an exams (= list of exercises)
myexam <- list(
  "boxplots",
  c("tstat", "ttest", "confint"),
  c("regression", "anova"),
  c("scatterplot", "boxhist"),
  "relfreq"
)

if(interactive()) {
  ## compile a single random exam (displayed on screen)
  exams2nops(c("tstat2", "anova", "boxplots"))
}
```

---

exams2olat *Legacy Generation of Exams for OpenOlat (Uni Innsbruck)*

Description

Old legacy interface for producing QTI 1.2 (rather than QTI 2.1) exams for OpenOlat at Uni Innsbruck. By now superseded by exams2openolat.

Usage

```
exams2olat(
  file,
  n = 1L,
  dir = ".",
  name = "olattest",
  maxattempts = 1L,
  cutvalue = 1000,
  solutionswitch = FALSE,
  stitle = "Aufgabe",
  ititle = "Frage",
  adescription = "Bitte bearbeiten Sie folgende Aufgaben.",
  sdescription = "Bitte beantworten Sie folgende Frage.",
  eval = list(partial = FALSE, negative = FALSE),
  ...
)
```

Arguments

`file` list, of `md`/`Rmd` files to be used

`n` integer, number of randomized tests to be created (default `1L`)

`dir` character, where to store the resulting file(s) (default `.`)

`name` character name of the test/quiz

`maxattempts` integer, he maximum attempts for one question (must be smaller than `100000L`).

`cutvalue` numeric, the cutvalue at which the exam is passed

`solutionswitch` logical Should the question/item solutionswitch be enabled? In OLAT this means that the correct solution is shown after an incorrect solution was entered by an examinee (i.e., this is typically only useful if `maxattempts = 1`).

`stitle` character A title that should be used for the sections. May be a vector of length 1 to use the same title for each section, or a vector containing different section titles.

`ititle` character A title that should be used for the assessment items. May be a vector of length 1 to use the same title for each item, or a vector containing different item titles. Note that the maximum of different item titles is the number of sections/questions that are used for the exam.

`adescription` character Description (of length 1) for the overall assessment (i.e., exam).

`sdescription` character Vector of descriptions for each section.

`eval` named list, specifies the settings for the evaluation policy, see function `exams_eval`

`...` forwarded to `exams2qti12`

Details

`exams2olat` is the old convenience interface to produce QTI 1.2 tests/exams for OpenOlat. It has been superseded by `exams2openolat` which offers more options and flexibility.

exams2openolat *Deprecated: Generation of Exams for OpenOlat (Uni Innsbruck)*

Description

Unexported legacy interface to `exams2openolat` with slightly different default values as used at the Department of Statistics, Uni Innsbruck. Instead it is recommended to use `exams2openolat` directly.

Usage

```
exams2openolat(
  file,
  n = 1L,
  dir = ".",
  name = "olattest",
  maxattempts = 1,
  cutvalue = 1000,
  solutionswitch = FALSE,
  qti = "2.1",
  stitle = "Aufgabe",
  ititle = "Frage",
  adescription = "",
  sdescription = "",
  eval = list(partial = FALSE, negative = FALSE),
  template = "qti21",
  ...
)
```

Arguments

`file` character. A specification of a (list of) exercise files.

`n` integer. The number of copies to be compiled from `file`.

`dir` character. The default is either display on the screen or the current working directory.

`name` character. A name prefix for resulting ZIP and RDS file.

`maxattempts` integer. The maximum attempts for one question, may also be set to `Inf`.

`cutvalue` numeric. The cutvalue at which the exam is passed.

`solutionswitch` logical. Should the question/item solutionswitch be enabled?

`qti` character indicating whether QTI `"1.2"` or `"2.1"` (default) should be generated.

`stitle`, `ititle`, `adescription`, `sdescription`
character. Descriptions for various titles/descriptions. Defaults to generic German titles.

`eval` named list. Specifies the settings for the evaluation policy, see function `exams_eval`. The default is not to partial credits for multiple-choice exercises.

`template` character. The IMS QTI 2.1 template that should be used. In addition to the default template this can be set to `"qti21test"` which uses almost the same template but with a PASS variable. Thus, no `cutvalue` is needed and the online tests just generate a SCORE.

`...` arguments passed on to `exams2openolat`.

Details

`exams2openolat` is a convenience interface for `exams2openolat` with somewhat different defaults: German titles/descriptions, partial credits disabled, solution switch turned off, large cut value (so that the test cannot be passed), and with fancy quotes turned off in verbatim R output. Finally, an RDS file is stored as a by-product containing the `xexams` list. This enables extracting and displaying specific exercises from an online test in R.

Value

A list of exams as generated by `xexams` is returned invisibly.

See Also

olat_eval olat_exercise

Examples

```
## load package and enforce par(ask = FALSE)
library("exams")
options(device.ask.default = FALSE)

## define an exams (= list of exercises)
myexam <- list(
  "boxplots",
  c("tstat", "ttest", "confint"),
  c("regression", "anova"),
  c("scatterplot", "boxhist"),
  "relfreq"
)

## output directory
mydir <- tempdir()

## generate .zip with German OpenOlat exam in temporary directory
## using a few customization options
exams2openolat(myexam, n = 3, dir = mydir, maxattempts = 2)
dir(mydir)
```

nops_eval *Deprecated: Evaluate NOPS Exams (Uni Innsbruck)*

Description

Unexported legacy interface to evaluate NOPS exams produced with exams2nops, and scanned by nops_scan. Instead it is recommended to use nops_scan directly.

Usage

```
nops_eval(
  register = Sys.glob("*.csv"),
  solutions = Sys.glob("*.rds"),
  scans = Sys.glob("nops_scan_*.zip"),
  points = NULL,
  eval = exams_eval(partial = FALSE, negative = 0.25),
  mark = c(0.5, 0.6, 0.75, 0.85),
  dir = ".",
  results = "nops_eval",
  html = NULL,
  col = hcl(c(0, 0, 60, 120), c(70, 0, 70, 70), 90),
  language = "de",
  module = NULL,
  interactive = TRUE,
  string_scans = Sys.glob("nops_string_scan_*.zip"),
  string_points = seq(0, 1, 0.25)
)

nops_eval_write_uibk(
  results = "nops_eval.csv",
  file = "exam_eval",
  dir = ".",
  language = "en",
  ...
)

nops_register(
  file = Sys.glob("*.xls*"),
  startid = 1L,
  tab = !identical(startid, FALSE),
  pdf = !identical(startid, FALSE),
  split = NULL,
  info = NULL,
  verbose = TRUE,
  ...
)
```

Arguments

`register` character. File name of a CSV file (semicolon-separated) of the registered students, e.g., as produced by `nops_register` based on the VIS registration lists. Must contain columns `"Matrikelnr"`, `"Name"`, `"Account"` (and `"LV"`, in case `module` marks should be computed). The file name should not contain spaces, umlaut or other special characters (something like `"GP-2014-02.csv"` is recommended).

`solutions` character. File name of the RDS exercise file produced by `exams2nops`.

`scans` character. File name of the ZIP file with scanning results (containing Daten.txt and PNG files) as produced by `nops_scan` (or the FSS).

`points` numeric. Vector of points per exercise. By default read from `solutions`.

`eval` list specification of evaluation policy as computed by `exams_eval`.

`mark` logical or numeric. If `mark = FALSE`, no marks are computed. Otherwise `mark` needs to be a numeric vector with four threshold values to compute marks from 5 to 1. The thresholds can either be relative (all lower than 1) or absolute. In case results exactly matching a threshold, the better mark is used.

`dir` character. File path to the output directory (the default being the current working directory).

`results` character. Prefix for output files.

html character. File name for individual HTML files, by default the same as register with suffix .html.

col character. Hex color codes used for exercises with negative, neutral, positive, full solution.

language character. Path to a DCF file with a language specification. Currently, "en" and "de" are shipped with the package.

module logical or numeric. Should module marks (in addition to the exam marks) be computed? If this is numeric, this can be a vector of two ECTS weights for the written exam and seminar, respectively (by default equal weights of 0.5 and 0.5 are used). If module is not FALSE, register needs to contain a column "LV" with the seminar marks.

interactive logical. Should possible errors in the Daten.txt file by corrected interactively? Requires the **png** package for full interactivity.

string_scans character. Optional file name of the ZIP file with scanning results of string exercise sheets (if any) containing Daten2.txt and PNG files as produced by nops_scan.

string_points numeric. Vector of length 5 with points assigned to string results.

file character. Name of the VIS file with the registration list.

... Further arguments passed on to nops_eval_write.

startid integer or logical, default 1L. If FALSE no random seats are generated.

tab logical. Should a tab-separated file with the seat information be generated for OpenOlat? Defaults to TRUE if random seats are generated.

pdf logical. Should PDF files with participant lists be generated for printing? Defaults to TRUE if random seats are generated.

split integer. Number of participant lists ordered by seat.

info character. Vector of length 4 with information about the exam: (1) type of exam (GP, LVP, VO, ...), (2) title of exam, (3) date/time (YYYY-MM-DD HH:MM), (4) location/room. By default the information is inferred from the VIS file.

verbose logical. Should information about the registrations be printed to the screen?

... further arguments passed to read_vis (e.g., subset).

Details

nops_eval is a companion function for exams2nops and nops_scan. It calls nops_eval from the **exams** package which evaluates the scanned exams by computing the sums of the points achived and (if desired) mapping them to marks (and to module marks). Furthermore HTML reports for each individual student are generated for upload into OpenOlat. In addition to this function from the **exams** package, the function adds the marks in the Uni Innsbruck-specifc format in an Excel spreadsheet.

nops_register is another companion function for preprocessing the registration lists that are provided by VIS. The function assigns random seats for every student and saves the result in both CSV and XLSX format as well as a tab-separated text file with the seat numbers for import into OLAT. The underlying workhorse function is read_vis.

Value

A `data.frame` with the detailed exam results is returned invisibly. It is also written to a CSV file in the current directory, along with a ZIP file containing the HTML reports (for upload into OLAT), and an XLSX file (for importing the marks into VIS).

See Also

`exams2nops`, `nops_scan`, `nops_eval`, `read_vis`

`nops_feedback` *Generate nops test feedback files (answer/solution)*

Description

Similar to what `olat_eval` does but in a more detailed way. Creates html files for each participant with feedback about his/her test results (including questions and solutions).

Usage

```
nops_feedback(res, xexam, name = "nops_feedback")
```

Arguments

`res`	`data.frame`, content of the csv file created by `nops_eval`.
`xexam`	list as returned from reading the rds file
`name`	character, name of the test, will be used to name the zip archive file and the html files

Value

Returns the name of the zip file created.

Author(s)

Reto Stauffer

nops_itemresp *Extract Item-Response Data from NOPS Exams*

Description

Process data from NOPS evaluation results (via nops_eval) for subsequent IRT (item response theory) modeling.

Usage

```
nops_itemresp(
  eval = "nops_eval.csv",
  exam = Sys.glob("*.rds"),
  psychotools = NULL,
  labels = NULL,
  ...
)
```

Arguments

eval	character. File name of CSV output from nops_eval.
exam	character. File name of RDS output from exams2nops.
psychotools	logical. Should itemresp from **psychotools** be used as the class for item response columns? By default, this is used if the **psychotools** package is available. If FALSE, matrices with dummy codings are used.
labels	function for extracting exercise labels from each $metainfo. By default the original file name $metainfo$file is used.
...	additional arguments (such as colClasses) passed to read.csv2(eval, dec = ".", ...).

Details

nops_itemresp returns a data frame with several item response outcomes for each student: solved indicates whether or not an exercise was fully solved, partial whether or not it was at least partially solved. points gives the points achieved for each exercise. The corresponding nsolved, npartial, and npoints are the sums of these for each student. Moreover, solved2, partial2, and points2 distinguish not only the exercises within the exam but also the actual source template within each exercise.

Value

A data.frame.

See Also

nops_eval

olat_eval *Evaluate OLAT Exams*

Description

Evaluate OLAT exams produced with exams2olat, and carried out and exported in (Open)OLAT.

Usage

```
olat_eval(file, plot = TRUE, export = FALSE)
```

Arguments

file	character. Base file name of RDS and XLS file with exam generated by exams2olat and exported OLAT results, respectively.
plot	logical. Should barplots with the aggregated results be displayed on the screen?
export	logical. Should detailed questions along with individual results be exported to HTML files in a ZIP archive for convenient import into OLAT?

Details

olat_eval is a companion function for exams2olat. It evaluates the exams carried out in OLAT for further processing outside of OLAT (in CSV format) and optionally exports detailed individual HTML reports (in a ZIP archive) for reimport into OLAT.

Value

A data.frame with the detailed exam results is returned invisibly. It is also written to a CSV file in the current directory, along with a ZIP file containing the HTML reports (for upload into OLAT).

See Also

exams2olat

olat_eval_adjust_lang *Adjust language of variables/columns*

Description

Modifies the names of the variables in the dat.frame as read from the xlsx file (OpenOLAT). Converts the variable names to English such that we do no longer have to care about language in all other methods/functions.

Usage

```
olat_eval_adjust_lang(x)
```

Arguments

x	`data.frame` read from the xlsx file

Details

Input 'x' is the data.frame as read from the xlsx file which contains user meta information and the detailed information about the individual questions of the test. Problem: depending on the user language settings of OLAT the names and order of the columns differs. This function takes input 'x' and manipulates the variable or column names in a way that the rest of the code (`olat_eval`) is not language dependent anymore.

- tries to guess the language by calling `olat_eval_guess_lang`
- loads the search-replace-data.frame (internally)
- search and replace variable names
- return input object `x` with new varaible names

Note: Even English to English will rename some of the variables.

The function uses the data set `olat_eval_lang` which is shipped with the package (see `data("olat_eval_lang")`).

Value

Returns the same `data.frame` (same dimension and data) with adjusted names.

Author(s)

Reto

`olat_eval_export` *Olat eval export*

Description

Takes the results from `read_olat_results` and the information from the `rds` file with the individual questions/answers to generate a zip archive file with individual test results (html file). This zip file can be used to upload to OLAT.

Usage

```
olat_eval_export(
  results,
  xexam,
  file = "olat_eval.zip",
  html = "Testergebnisse.html",
  col = hcl(c(0, 0, 60, 120), c(70, 0, 70, 70), 90)
)
```

Arguments

`results` data.frame, results from `read_olat_results`

`xexam` list the object loaded from the rds file which contains the individual questions/answers. The length of the list corresponds to the number of randomized tests, each list element contains N elements (N = number of questions) with all the information required to generate the output.

`file` character, name of the zip flie, the final archive file where to store the exported html files

`html` character, name of the output files (html files)

`col` character vector of length `4L` with hex colors, default is `hcl(c(0, 0, 60, 120), c(70, 0, 70, 70), 90)`

Value

Return of the `zip()` call.

`olat_eval_guess_lang` *Guess language of the OLAT results file*

Description

Given a set of character vectors this function tries to gess the language of the imported file. This allows to evaluate exams from OpenOLAT in different languages. Used by `olat_eval` to rename the columns to make the evaluation independent from the language used when exporting the results via OLAT.

Usage

```
olat_eval_guess_lang(x)
```

Arguments

`x` character vector with column names

Value

Returns the language name (`"en"`, `"de"`) if the function is able to guess the language. Else the script will stop.

Author(s)

Reto

`olat_eval_lang` *Olat evaluation language data set*

Description

Depending on the user profile language settings OLAT exports test results (xlsx file) in differnet languages. To make things easier in the **c403** package the variable names are modified using this language search-and-replace data set.

Usage

```
olat_eval_lang
```

Format

A data frame with a language flag (given your export was in German, only rows with `lang == "de"` will be considered). The two columns `search` and `replace` are used for `gsub()` (regular expression string replacement).

Author(s)

Reto Stauffer

`olat_exercise` *Extract (and Display) Exercises from OpenOlat Exams*

Description

Extract (and display) selected exercises from OpenOlat exams produced with `exams2openolat` in order to see both question and solution.

Usage

```
olat_exercise(x, ..., fixed = TRUE, show = TRUE, mathjax = TRUE)
```

Arguments

`x`	character or list. Either an OpenOlat exam list as produced by `exams2openolat` or a character with the file path to an `.rds` file containing such an exam.
`...`	character. Either a single numeric index of the exam to be selected. Or, alternatively, patterns to be searched for in the question text of the exams in `x`.
`fixed`	logical. Should the search pattern(s) be matched as is?
`show`	logical. Should the exercise(s) found be shown in the browser?
`mathjax`	logical. Should the JavaScript from `http://www.MathJax.org/` be included for rendering mathematical formulas?

Details

olat_exercise is a companion function for exams2openolat. As OpenOlat has no option during an exam to look at the precise question of a particular student – and more importantly the corresponding solution – one strategy is to search for particular words, numbers, or other strings in the database of all questions from an OpenOlat exam.

olat_exercise goes through all questions in the exam and selects those question(s) that match(es) the given search patterns. By default the question(s)/solution(s) are displayed in the browser and returned invisibly.

Value

A list containing either a single exercise or a list of such exercises (in case the search patterns do not yield a unique question).

See Also

exams2openolat

olat_feedback *Generate OLAT test feedback files (answer/solution)*

Description

Similar to what olat_eval does but in a more detailed way. Creates html files for each participant with feedback about his/her test results (including questions and solutions). For OLAT tests use olat_feedback, for nops tests (written tests) use nops_feedback.

Usage

```
olat_feedback(res, xexam, name = "olat_feedback")

olat_feedback_render_one(res, xexam, i, htmlfile = "Result.html", show = FALSE)
```

Arguments

res	data.frame, result from olat_eval
xexam	list as returned from reading the rds file
name	character, name of the test, will be used to name the zip archive file and the html files
i	integer, row index (which row in 'x' to render)
htmlfile	character, name of the output file
show	logical, if set to TRUE the html file will be opened in a browser

Value

Returns the name of the zip file created.

Author(s)

Reto Stauffer

`read_olat_results` *Read OLAT Results from XLSX file*

Description

Get OLAT test results (FIXME: eval support and cloze evaluation). TODO: Write proper documentation.

Usage

```
read_olat_results(file, xexam = NULL)
```

Arguments

`file`	name of the file ...
`xexam`	...

Value

...

`read_vis` *Reading VIS Registrations*

Description

Read registration lists (for exams or courses) from the Excel export of VIS (which actually may or may not be XLS or HTML files).

Usage

```
read_vis(file, ...)

vis_register(file = Sys.glob("*.xls"), subset = TRUE)
```

Arguments

`file`	character with file name of an XLS file from VIS.
`...`	additional arguments passed to `read.xlsx`.
`subset`	logical. Should students without confirmed registration be omitted?

Details

VIS offers Excel exports but in case of registration lists these are actually HTML files containing an HTML table. (Note that as of 2021 VIS offers an additional “real Excel” export.) HTMLtables are read using the XML package. However, some exports are also converted to actual Excel files which are read using the xlsx package. In either case some basic cleaning is done and additional meta-information is extracted.

The `vis_register` function loops over reading several VIS exports and then consolidates the resulting data frames.

Value

A `data.frame` with an additional attribute `"info"` providing details about the type of course (`"LV"`) or exam (`"GP"`).

uibkmark *Auxiliary Formatting Functions*

Description

Auxiliary functions for formatting elements of exams.

Usage

```
uibkmark(x, factor = TRUE)

mchoice2text(
  x,
  true = "\\\\textbf{Richtig}",
  false = "\\\\textbf{Falsch}"
)
```

Arguments

`x`	numeric (`uibkmark`) or logical (`mchoice2text`) vector.
`factor`	logical. Should the result be a factor or a character?
`true`	character. Text for true results.
`false`	character. Text for false results.

Details

The function `uibkmark` maps the numbers 1 to 5 to the mark labels SGT1, GUT2, etc. as used by UIBK.

The function `mchoice2text` masks the exams function of the same name in order to show German text.

Examples

```
uibkmark(1:5)
mchoice2text(c(TRUE, FALSE))
```

`vis_groups` *Randomly Assign VIS Participants into Groups (as HTML Table)*

Description

Randomly assign a vector of names (typically obtained from a VIS registration) into groups and display the result as an HTML table.

Usage

```
vis_groups(x, nrow = 5L, ncol = 2L, ...)
```

Arguments

`x` character. Either a vector of names or a file name (csv or xls/xlsx from VIS) from which the `Name` column can be extracted.

`nrow, ncol` numeric. Number of rows and columns into which the students should be assigned.

`...` additional arguments passed to `read_vis` or `read.csv2`.

Value

A character vector with the HTML code is returned invisibly.

Index