Package: RTutorSAGI (via r-universe)

August 27, 2024

Type Package

Title Submission Analysis for Grading and Improvement

Version 2020.11.27

Date 2022-11-27

Author Sebastian Kranz

Maintainer Sebastian Kranz <sebastian.kranz@uni-ulm.de>

Description Tools for analysising submissions of solved RTutor problem sets. The functions help automatic bulk grading for several problem sets. Another feature is a shiny app that helps to analyse where students got stuck and correspondingly improve the problem set.

License Programm code: GPL >= 2.0 Contributed problem sets: Creative Commons (CY)

Depends RTutor

Suggests digest

RoxygenNote 6.0.1

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

RemoteUrl https://github.com/skranz/RTutorSAGI

RemoteRef master

RemoteSha 2f473fd792b005e333c1f35970f5dfa605cd6f06

Contents

`analyse.subs` *Summarize number of errors and hints by looking at the submission logs*

Description

See README.md for usage

Usage

```
analyse.subs(sub.li, rps.dir = "org_ps", just.summary = FALSE,
  no.summary = FALSE, protracted.minutes = 30)
```

Arguments

`sub.li`	A list of loaded submission files. See `load.subs`
`rps.dir`	The directory in which you have all rps files of the problem sets (original versions)
`just.summary`	if TRUE just return the chunk summary data frame. If FALSE (default) return a list that also contains several intermediate data frames.
`no.summary`	if TRUE just return list without summary data frame (used by write.chunk.logs)
`protracted.minutes`	Used to classify whether it took a long time, possible due to a break when gotten stuck for a student to solve the chunk. We measure the duration between the first failed attempt and the time the chunk is first solved. If this is longer (in minutes) than `protracted.minutes` (default = 30), we classify that it was protracted for the student to solve the chunk. The summary in sum.df counts for how many students it was protracted to solve the chunk.

`err.sol.table` *Create an err.sol.table as input for hint.stud.table*

Description

Create an err.sol.table as input for hint.stud.table

Usage

```
err.sol.table(err.df, rps.dir = getwd())
```

Arguments

`err.df`	If res is the the returned list from a call to analyse.subs() it is the elemet res$err.df
`rps.dir`	The directory in which all problem set rps files can be found of the problem sets that have been analysed by analyse.subs.

err.sol.table.for.ps *Create an err.sol.table from a single problem set*

Description

see err.sol.table for a basic description

Usage

```
err.sol.table.for.ps(err.df, ps.name, rps.dir = getwd(),
  rps = load.rps(file.path(rps.dir, ps.name)))
```

grade.subs *Combine for each student the points from all problem sets and create csv files with the total points.*

Description

See README.md for usage

Usage

```
grade.subs(sub.li, grade.dir = "grades")
```

hint.stud.table *Creates a table with hint.stud templates from an err.sol.table*

Description

Creates a table with hint.stud templates from an err.sol.table

Usage

```
hint.stud.table(es, min.users = 1, default.filters = TRUE)
```

Arguments

es	a data frame returned by err.sol.table, err.sol.table.for.ps or err.sol.table.for.chunk
min.users	only keep wrong calls that at least min.users made
default.filters	if TRUE remove some rows that are unlikely to be used as custom hints (e.g. if in assignment both the assigned variable and the rhs differ).

load.moodle.subs *Load submission files that follow Moodle's naming convention*

Description

Load submission files that follow Moodle's naming convention

Usage

```
load.moodle.subs(sub.dir = "sub", stud.name.fun = moodle.stud.name.fun, ...)
```

load.sub *Load a single submission file*

Description

See README.md for usage

Usage

```
load.sub(file, stud.name.fun = NULL)
```

load.subs *Load all submissions from a directory*

Description

See README.md for usage

Usage

```
load.subs(sub.dir = "sub", files = NULL, stud.name.fun = NULL,
  warn = TRUE, max.files = NA)
```

Arguments

sub.dir	The directory in which all submission files can be found.

`unpack.moodle.sub.zips` *Takes assignment ZIPs with all students' solutions and unpacks them into separate folders for each assignment*

Description

Takes assignment ZIPs with all students' solutions and unpacks them into separate folders for each assignment

Usage

```
unpack.moodle.sub.zips(zip.dir = "moodle_zip", sub.dir = "sub",
  prefix = "", postfix = ".zip")
```

Arguments

`zip.dir`	directory with big ZIP files from Moodle. Each ZIP file contains all submissions of one problem set
`sub.dir`	directory into which sub files shall be extracted

`write.chunk.logs` *Creates for each problem set and each chunk an R file that protocols the solution attempts by students.*

Description

See README.md for details

Usage

```
write.chunk.logs(sub.li, logs.dir = "chunk_logs", rps.dir = "org_ps")
```

Arguments

`sub.li`	A list of loaded submission files. See `load.subs`
`logs.dir`	Directory into which log files shall be written
`rps.dir`	The directory in which you have all rps files of the problem sets (original versions)

Index