Package ‘crsra’

October 12, 2022

Title Tidying and Analyzing ‘Coursera’ Research Export Data

Version 0.2.3

Description Tidies and performs preliminary analysis of ‘Coursera’ research export data. These export data can be downloaded by anyone who has classes on Coursera and wants to analyze the data. Coursera is one of the leading providers of MOOCs and was launched in January 2012. With over 25 million learners, Coursera is the most popular provider in the world being followed by EdX, the MOOC provider that was a result of a collaboration between Harvard University and MIT, with over 10 million users. Coursera has over 150 university partners from 29 countries and offers a total of 2000+ courses from computer science to philosophy. Besides, Coursera offers 180+ specialization, Coursera's credential system, and four fully online Masters degrees. For more information about Coursera check Coursera's About page on <https://blog.coursera.org/about/>.

Depends R (>= 2.10)

License GPL-2

Encoding UTF-8

LazyData true

Suggests utils, testthat, rmarkdown

Imports dplyr, purrr, readr, digest, tidytext, tibble, rcorpora, knitr

BugReports https://github.com/jhudsl/crsra/issues

RooxygenNote 6.0.1.9000

VignetteBuilder knitr

NeedsCompilation no

Author Aboozar Hadavand [aut, cre], Jeff Leek [aut], John Muschelli [aut]

Maintainer Aboozar Hadavand <hadavand.a@gmail.com>

Repository CRAN

Date/Publication 2018-05-05 06:25:58 UTC


**R topics documented:**

- `crsra_anonymize` .................................................. 2
- `crsra_assessmentskips` ......................................... 3
- `crsra_delete_user` .................................................. 4
- `crsra_gradesummary` ............................................... 4
- `crsra_import` ......................................................... 5
- `crsra_import_as_course` .......................................... 6
- `crsra_import_course` .............................................. 6
- `crsra_membershares` ............................................... 7
- `crsra_progress` ...................................................... 8
- `crsra_tabledesc` ...................................................... 8
- `crsra_timetofinish` ............................................... 9
- `crsra_whichtable` .................................................. 9
- `example_course_import` ........................................... 10
- `tabdesc` ................................................................. 10

**Index**

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>crsra_anonymize</code></td>
<td>2</td>
</tr>
<tr>
<td><code>crsra_assessmentskips</code></td>
<td>3</td>
</tr>
<tr>
<td><code>crsra_delete_user</code></td>
<td>4</td>
</tr>
<tr>
<td><code>crsra_gradesummary</code></td>
<td>4</td>
</tr>
<tr>
<td><code>crsra_import</code></td>
<td>5</td>
</tr>
<tr>
<td><code>crsra_import_as_course</code></td>
<td>6</td>
</tr>
<tr>
<td><code>crsra_import_course</code></td>
<td>6</td>
</tr>
<tr>
<td><code>crsra_membershares</code></td>
<td>7</td>
</tr>
<tr>
<td><code>crsra_progress</code></td>
<td>8</td>
</tr>
<tr>
<td><code>crsra_tabledesc</code></td>
<td>8</td>
</tr>
<tr>
<td><code>crsra_timetofinish</code></td>
<td>9</td>
</tr>
<tr>
<td><code>crsra_whichtable</code></td>
<td>9</td>
</tr>
<tr>
<td><code>example_course_import</code></td>
<td>10</td>
</tr>
<tr>
<td><code>tabdesc</code></td>
<td>10</td>
</tr>
</tbody>
</table>

---

**Description**

This function will still keep the relationship between tables, i.e. it will change a specific id across all tables to the same id.

**Usage**

```r
crsra_anonymize(all_tables,
col_to_mask = attributes(all_tables)$partner_user_id, algorithm = "crc32")
```

**Arguments**

- `all_tables` A list from `crsra_import_course` or `crsra_import`
- `col_to_mask` The name of id column to mask.
- `algorithm` The algorithms to be used for anonymization; for currently available choices, see `digest`.

**Value**

A list that contains all the tables within each course.
Examples

```r
res = crsra_anonymize(example_course_import,
                      col_to_mask = "jhu_user_id",
                      algorithm = "crc32")
```

---

**crsra_assessmentskips**  
*Frequencies of skipping an peer-assessed submission*

### Description

Frequencies of skipping an peer-assessed submission

### Usage

```r
crsra_assessmentskips(all_tables, bygender = FALSE, wordcount = TRUE,
                       n = 20)
```

### Arguments

- `all_tables`: A list from `crsra_import_course` or `crsra_import`
- `bygender`: A logical value indicating whether results should be broken down by gender
- `wordcount`: A logical value indicating whether word count should be shown in the results; default is true
- `n`: An integer indicating the number of rows for the word count

### Value

The outputs are frequency tables (tibble) and are shown for each specific course

### Examples

```r
crsra_assessmentskips(example_course_import)
crsra_assessmentskips(example_course_import, bygender = TRUE, n = 10)
```
### crsra_delete_user

*Deletes a specific user from all tables in the data in case Coursera data privacy laws require you to delete a specific (or set of) user(s) from your data.*

**Description**

Deletes a specific user from all tables in the data in case Coursera data privacy laws require you to delete a specific (or set of) user(s) from your data.

**Usage**

```r
crsra_delete_user(all_tables, users)
```

**Arguments**

- **all_tables**
  
  A list from `crsra_import_course` or `crsra_import`

- **users**
  
  A vector of user ids to delete

**Value**

A list that contains all the tables within each course.

**Examples**

```r
del_user = example_course_import$users$jhu_user_id[1]
del_user %in% example_course_import$users$jhu_user_id
res = crsra_delete_user(example_course_import, users = del_user)
del_user %in% res$users$jhu_user_id
```

### crsra_gradesummary

*The average course grade across different groups*

**Description**

The average course grade across different groups

**Usage**

```r
crsra_gradesummary(all_tables, groupby = c("total", "country", "language", "gender", "empstatus", "education", "stustatus"))
```
Arguments

all_tables A list from crsra_import_course or crsra_import


groupby A character string indicating the how to break down grades. The default is set to total and returns the grade summary for each course. Other values are gender (for grouping by gender), education (for grouping by education level), stustatus (for grouping by student status), empstatus (for grouping by employment status), and country (for grouping by country). Note that this grouping uses the entries in the table users that is not fully populated so by grouping you lose some observations.

Value

A table which indicates the average grade across specified groups for each course

Examples

```r
crsra_gradesummary(example_course_import)
crsra_gradesummary(example_course_import, groupby = "education")
```

---

crsra_import Imports all the .csv files into one list consisting of all the courses and all the tables within each course.

Description

Imports all the .csv files into one list consisting of all the courses and all the tables within each course.

Usage

```r
crsra_import(workdir = ".", ...)
```

Arguments

workdir A character string vector indicating the directory where all the unzipped course directories are stored.

... Additional arguments to pass to crsra_import_course

Examples

```r
zip_file = system.file("extdata", "fake_course_7051862327916.zip", package = "crsra")
bn = basename(zip_file)
bz = sub("\[.]zip\$", "", bn)
res = unzip(zip_file, exdir = tempdir(), overwrite = TRUE)
example_import = crsra_import(workdir = tempdir(), check_problems = FALSE)
```
crsra_import_as_course

Convert a Coursera Course to Coursera Import

Description

Convert a Coursera Course to Coursera Import

Usage

crsra_import_as_course(x)

Arguments

x object of class coursera_import or coursera_course_import

Value

object of class coursera_import

crsra_import_course

Imports all the .csv files into one list consisting of all the tables within the course.

Description

Imports all the .csv files into one list consisting of all the tables within the course.

Usage

crsra_import_course(workdir = ".", add_course_name = FALSE, change_pid_column = FALSE, check_problems = TRUE)

Arguments

workdir A character string vector indicating the directory where the unzipped course is stored.
add_course_name Should a column of the course name be added to all the data.frames
change_pid_column Should the partner_user_id column be changed to simply say "partner_user_id"?
check_problems Should problems with reading in the data be checked?
Examples

```r
zip_file = system.file("extdata", "fake_course_7051862327916.zip", 
    package = "crsra")
bn = basename(zip_file)
bn = sub("[.]zip$", "", bn)
res = unzip(zip_file, exdir = tempdir(), overwrite = TRUE)
workdir = file.path(tempdir(), bn)
course_tables = crsra_import_course(workdir,
    check_problems = FALSE)
```

Description

The share of learners in each course based on specific characteristics.

Usage

```r
crsra_membershares(all_tables, groupby = c("roles", "country", "language", 
    "gender", "empstatus", "education", "stustatus"), remove_missing = TRUE)
```

Arguments

- `all_tables`: A list from `crsra_import_course` or `crsra_import`
- `groupby`: A character string indicating the how to break down learners in each course. The default is set to `roles` and returns the share of students in each category such as Learner, Not Enrolled, Pre-Enrolled Learner, Mentor, Browser, and Instructor. Other values are `country` (for grouping based on country), `language` (for grouping based on language), `gender` (for grouping by gender), `education` (for grouping by education level), `stustatus` (for grouping by student status), `empstatus` (for grouping by employment status), and `country` (for grouping by country). Note that this grouping uses the entries in the table `users` that is not fully populated so by grouping you lose some observations.
- `remove_missing`: Should the NA be removed from the `groupby` column?

Value

A table which indicates the total number and the share of students in each group for each course.

Examples

```r
crsra_membershares(
    example_course_import, 
    groupby = "country")
crsra_membershares(
    example_course_import, 
    groupby = "roles", remove_missing = FALSE)
```
crsra_membershares(
example_course_import,
groupby = "roles", remove_missing = TRUE)

---

**crsra_progress**

Ordered list of course items and the number and share of learners who have completed the item

**Description**

Ordered list of course items and the number and share of learners who have completed the item

**Usage**

```r
crsra_progress(all_tables)
```

**Arguments**

- `all_tables` A list from `crsra_import_course` or `crsra_import`

**Value**

A table which lists all the item within a course and the total number of learners and the share of learners who have completed the item.

**Examples**

```r
crsra_progress(example_course_import)
```

---

**crsra_tabledesc**

Returns description for a table

**Description**

Returns description for a table

**Usage**

```r
crsra_tabledesc(x)
```

**Arguments**

- `x` Name of the table to get the description

**Value**

The description for a table based on the description provided by Coursera in the data exports
**Examples**

```r
crsra_tabledesc("assessments")
```

---

**crsra_timetofinish**

*Time that took each learner (in days) to finish a course*

---

**Description**

Time that took each learner (in days) to finish a course

**Usage**

```r
crsra_timetofinish(all_tables)
```

**Arguments**

- `all_tables`:
  A list from `crsra_import_course` or `crsra_import`

**Value**

A table containing `hashed_user_ids` with a column indicating the time (in days) that took each user to complete a course. The time is calculated as the difference between the last and first activity in the course.

**Examples**

```r
crsra_timetofinish(example_course_import)
```

---

**crsra_whichtable**

*Returns a list of tables a variable appears in*

---

**Description**

Returns a list of tables a variable appears in

**Usage**

```r
crsra_whichtable(all_tables, col_name)
```

**Arguments**

- `all_tables`:
  A list from `crsra_import_course` or `crsra_import`
- `col_name`:
  The name of the column/variable to look for
Value

A list of tables that a specific variable appears in

Examples

crsra_whichtable(example_course_import, "assessment_id")

description

valid

Usage
description

Format

A list with 100 elements, which are data.frames imported from a fake Coursera class:

description

Format

A vector table descriptions, where the names of the table descriptions is the name of the tables in an import.
Index

* datasets
  example_course_import, 10
  tabdesc, 10

  crsra_anonymize, 2
  crsra_assessmentskips, 3
  crsra_delete_user, 4
  crsra_gradesummary, 4
  crsra_import, 2–5, 5, 7–9
  crsra_import_as_course, 6
  crsra_import_course, 2–5, 6, 7–9
  crsra_membershares, 7
  crsra_progress, 8
  crsra_tabledesc, 8
  crsra_timetofinish, 9
  crsra_whichtable, 9

digest, 2

example_course_import, 10

tabdesc, 10