Package 'canvasquizzeR'

August 22, 2023

Title Tools for converting quizzes CSV files to QTI files or export QTI files to HTML files

Version 0.0.0.9000

Description This packages contains functions for converting quizzes CSV files to QTI files or export QTI files to HTML files.

License `use_gpl3_license()`

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.3

Imports readr,
stringr,
utils,
XML,
dplyr,
readxl,
tibble,
magrittr,
knitr,
rmarkdown,
googlesheets4,
googledrive,
docxtractr

R topics documented:

`create_groupxml` *Create the XML text for a group of quiz questions*

Description

This function creates the XML code for a group of questions, including the XML creating the group and the XML for each individual question.

Usage

```
create_groupxml(df, groupi, respids)
```

Arguments

`df`	data.frame where each row is a question. The column names should be the following: • G: Question group • 'Question Type': Character string identifying the type of question, should equal either 'MC' for a multiple choice question or 'Essay' for an essay/short-answer question • 'Text Type': Character string identifying whether the question text is plain text or html, should equal either 'html' or 'plain' • Points: Number of points for the question • Question: Text of the question • A: Correct answer, a number 1 thru 4 • 'Choice 1': Choice 1 • 'Choice 2': Choice 2 • 'Choice 3': Choice 3 • 'Choice 4': Choice 4 • Feedback: General feedback given to students after they complete the quiz and answers are shown
`groupi`	Character string of the group to create, which must be exactly one of the strings in column 'G' in the data frame `df`
`respids`	Vector of unique four-digit integers with length equal to the number of choices in the whole quiz. For example, if there are 10 questions in the quiz and each question has 4 answer choices, `respids` will be a vector of length 10*4 = 40.

Value

Character string of the XML code for a group of multiple-choice quiz questions

create_longid *Create an identity for a QTI item.*

Description

Each item in a QTI file contains a unique 33 digit hex code. This function just creates a random 33 character hex code. There are 5.4 duodecillion unique 33-digit hex codes, so it's extremely unlikely two will be the same.

Usage

```
create_longid(size = 33)
```

Arguments

size	Number of digits for the hex code. Defaults to 33.

Value

Character string of lenghth 34, starting with the letter 'g' and followed by the 33-digit hex code, because that's what Canvas wants?

create_questionxml *Create the XML text for a single quiz question*

Description

Create the XML text for a single quiz question. The quiz question may be a multiple-choice question (default if not specified) or essay question. Takes as an argument a data frame with all the multiple choice questions of the quiz, all the response IDs for every response on the quiz, and an integer identifying which row in the data frame that the question appears. It is necessary to include information for the whole quiz, because some questions may need response IDs that are unique to the entire quiz.

Usage

```
create_questionxml(df, qn, respids)
```

Arguments

df	data.frame where each row is a question. The column names should be the following: • G: Question group • 'Question Type': Character string identifying the type of question, should equal either 'MC' for a multiple choice question or 'Essay' for an essay/short-answer question • 'Text Type': Character string identifying whether the question text is plain text or html, should equal either 'html' or 'plain' • Points: Number of points for the question

- Question: Text of the question
- A: Correct answer, a number 1 thru 4
- 'Choice 1': Choice 1
- 'Choice 2': Choice 2
- 'Choice 3': Choice 3
- 'Choice 4': Choice 4
- Feedback: General feedback given to students after they complete the quiz and answers are shown

`qn` integer identifying the row in `df` that the individual question of interest appears

`respids` Vector of unique four-digit integers with length equal to the number of choices in the whole quiz. For example, if there are 10 questions in the quiz and each question has 4 answer choices, `respids` will be a vector of length 10*4 = 40.

Value

Character string of the XML code for a single multiple-choice quiz question

---

`create_questionxml_essay` *Create the XML text for a single essage question*

Description

Create the XML text for a single essay question. Takes as an argument a data frame with all the questions of the quiz and an integer identifying which row in the data frame that the question appears. It is necessary to include information for the whole quiz, because some questions may need response IDs that are unique to the entire quiz.

Usage

```
create_questionxml_essay(df, qn)
```

Arguments

`df` data.frame where each row is a question. The column names should include the following:

- G: Question group
- Question: Text of the question
- Feedback: General feedback given to students after they complete the quiz and answers are shown

`qn` integer identifying the row in `df` that the individual question of interest appears

Value

Character string of the XML code for a single multiple-choice quiz question

`create_questionxml_mc` *Create the XML text for a single multiple-choice question*

Description

Create the XML text for a single multiple-choice question. Takes as an argument a data frame with all the multiple choice questions of the quiz, all the response IDs for every response on the quiz, and an integer identifying which row in the data frame that the question appears. It is necessary to include information for the whole quiz, because each question needs response IDs that are unique to the entire quiz.

Usage

```
create_questionxml_mc(df, qn, respids)
```

Arguments

`df`	data.frame where each row is a question. The column names should be the following: • G: Question group • Question: Text of the question • A: Correct answer, a number 1 thru 4 • 'Choice 1': Choice 1 • 'Choice 2': Choice 2 • 'Choice 3': Choice 3 • 'Choice 4': Choice 4 • Feedback: General feedback given to students after they complete the quiz and answers are shown
`qn`	integer identifying the row in `df` that the individual question of interest appears
`respids`	Vector of unique four-digit integers with length equal to the number of choices in the whole quiz. For example, if there are 10 multilpe-choice questions in the quiz and each question has 4 answer choices, `respids` will be a vector of length 10*4 = 40.

Value

Character string of the XML code for a single multiple-choice quiz question

`create_quizdf_zip` *Extract all the questions from a quiz from a QTI .zip file*

Description

Extract all the questions from a quiz given in a QTI .zip file and return in a tibble

Usage

```
create_quizdf_zip(filename, folder)
```

Arguments

`filename`	Character string of the filename for the .zip file. Do not include the folder/path
`folder`	Character string of the folder for the .zip file. The contents of the zip file will be extracted to the same folder.

Value

Returns a tibble with all the questions in the quiz, where each row is a question, and the tibble has the following columns:

- G: Question group
- 'Question Type': Character string identifying the type of question, should equal either 'MC' for a multiple choice question or 'Essay' for an essay/short-answer question
- 'Text Type': Character string identifying whether the question text is plain text or html, should equal either 'html' or 'plain'
- Points: Number of points for the question
- Question: Text of the question
- A: Correct answer, a number 1 thru 4
- 'Choice 1': Choice 1
- 'Choice 2': Choice 2
- 'Choice 3': Choice 3
- 'Choice 4': Choice 4
- Feedback: General feedback given to students after they complete the quiz and answers are shown

`extract_group_questions` *Extract information for a group of quiz questions*

Description

Extract information for all the questions in a group of questions from an XML node associated with a question group and return in a single-row tibble

Usage

```
extract_group_questions(group.xml)
```

Arguments

`group.xml`	The XMLNode object associated with a group of quiz questions

Value

Returns a tibble with all the questions in the group, where each row is a question, and the tibble has the following columns:

- G: Question group
- 'Question Type': Character string identifying the type of question, should equal either 'MC' for a multiple choice question or 'Essay' for an essay/short-answer question
- 'Text Type': Character string identifying whether the question text is plain text or html, should equal either 'html' or 'plain'
- Points: Number of points for the question
- Question: Text of the question
- A: Correct answer, a number 1 thru 4
- 'Choice 1': Choice 1
- 'Choice 2': Choice 2
- 'Choice 3': Choice 3
- 'Choice 4': Choice 4
- Feedback: General feedback given to students after they complete the quiz and answers are shown

extract_qtizip — *Unzip the QTI file and return the filepath for the XML file containing the assessment*

Description

This function unzips the QTI .zip file that is in `folder`, extracting the contents to the same folder, and returns the filepath for the XML file that contains the quiz questions

Usage

```
extract_qtizip(filename, folder)
```

Arguments

`filename`	Character string of the filename for the .zip file. Do not include the folder/path
`folder`	Character string of the folder for the .zip file. The contents of the zip file will be extracted to the same folder.

Value

Character string of the filepath to the XML file containing the assessment tag (i.e. the XML file with the quiz questions)

extract_question — *Extract information for a single quiz question from XML*

Description

Extract information for a single quiz question from an XML node and return in a single-row tibble

Usage

```
extract_question(question.xml)
```

Arguments

`question.xml`	The XMLNode object associated with a quiz question

Value

Returns a single-row tibble with the following columns:

- G: Question group
- 'Question Type': Character string identifying the type of question, should equal either 'MC' for a multiple choice question or 'Essay' for an essay/short-answer question
- 'Text Type': Character string identifying whether the question text is plain text or html, should equal either 'html' or 'plain'
- Points: Number of points for the question

- Question: Text of the question
- A: Correct answer, a number 1 thru 4
- 'Choice 1': Choice 1
- 'Choice 2': Choice 2
- 'Choice 3': Choice 3
- 'Choice 4': Choice 4
- Feedback: General feedback given to students after they complete the quiz and answers are shown

---

`extract_quiz_xml` *Extract all the questions from a quiz given in an XML file*

Description

Extract all the questions from a quiz given in an XML file and return in a tibble

Usage

```
extract_quiz_xml(xml_filepath)
```

Arguments

`xml_filepath` Character string equal to the file path for the XML file containing the assessment

Value

Returns a tibble with all the questions in the group, where each row is a question, and the tibble has the following columns:

- G: Question group
- 'Question Type': Character string identifying the type of question, should equal either 'MC' for a multiple choice question or 'Essay' for an essay/short-answer question
- 'Text Type': Character string identifying whether the question text is plain text or html, should equal either 'html' or 'plain'
- Points: Number of points for the question
- Question: Text of the question
- A: Correct answer, a number 1 thru 4
- 'Choice 1': Choice 1
- 'Choice 2': Choice 2
- 'Choice 3': Choice 3
- 'Choice 4': Choice 4
- Feedback: General feedback given to students after they complete the quiz and answers are shown

generateQTI — *Generate QTI quiz file from data frame*

Description

This function generates a zipped QTI file for a multiple-choice test given by the data frame, df, and saves it in the folder `outfolder`

Usage

```
generateQTI(df, outfolder, quiztitle, quizfilename)
```

Arguments

`df`	This is a tibble (i.e. data.frame) where each row is a question. The column names should be the following: • G: Question group • 'Question Type': Character string identifying the type of question, should equal either 'MC' for a multiple choice question or 'Essay' for an essay/short-answer question • 'Text Type': Character string identifying whether the question text is plain text or html, should equal either 'html' or 'plain' • Points: Number of points for the question • Question: Text of the question • A: Correct answer, a number 1 thru 4 • 'Choice 1': Choice 1 • 'Choice 2': Choice 2 • 'Choice 3': Choice 3 • 'Choice 4': Choice 4 • Feedback: General feedback given to students after they complete the quiz and answers are shown
`outfolder`	Character string with path to the output folder for the QTI quiz file
`quiztitle`	Character string with the title for the quiz
`quizfilename`	Character string with the filename for the quiz

quiz_format — *Check formatting and set default values for quiz data frame, df*

Description

This function adjusts the format of the quiz to conform to the following:

- G: Question group
- Question: Text of the question
- Question type: Either MC or Essay, for multiple-choice or short-answer. The default is based on whether there is an answer given in column `A`, 'MC' if there is, 'Essay' otherwise.

- Text Type: The type of text in all text columns, 'html' or 'plain', for html-formatted text or plain text. The default is 'html'.
- Points: Numeric, number of points for the problem. The default is 1.
- A: Correct answer for a multiple choice question, a number 1 thru 4. Default is NA.
- 'Choice 1': Choice 1, default is empty character string ""
- 'Choice 2': Choice 2, default is empty character string ""
- 'Choice 3': Choice 3, default is empty character string ""
- 'Choice 4': Choice 4, default is empty character string ""
- Feedback: General feedback given to students after they complete the quiz and answers are shown. Default is empty character string "".

The only required column of the input data frame is `Question`. Any other missing columns will be created with default values

Usage

```
quiz_format(df)
```

Arguments

`df`	data frame containing a quiz, hopefully mostly conforming to the above.

Value

Quiz data frame with the columns above

`quiz_tibble` *Return a quiz tibble*

Description

Return an empty tibble with the columns associated with quiz data frames used in this package

Usage

```
quiz_tibble(nrows = 0)
```

Arguments

`nrows`	Number of rows for the tibble. Default is 0.

Value

Returns a tibble (i.e. data.frame) where each row can hold a quiz question. The column names are those that are used in this package for quizzes

- G: Question group
- 'Question Type': Character string identifying the type of question, should equal either 'MC' for a multiple choice question or 'Essay' for an essay/short-answer question

- 'Text Type': Character string identifying whether the question text is plain text or html, should equal either 'html' or 'plain'
- Points: Number of points for the question
- Question: Text of the question
- A: Correct answer, a number 1 thru 4
- 'Choice 1': Choice 1
- 'Choice 2': Choice 2
- 'Choice 3': Choice 3
- 'Choice 4': Choice 4
- Feedback: General feedback given to students after they complete the quiz and answers are shown

---

`read_quiz_csv` *Read a quiz in from a CSV file and return a quiz data frame*

Description

This function reads a quiz in from a CSV file and return a quiz data frame that conforms to the following:

- G: Question group
- Question: Text of the question
- Question type: Either MC or Essay, for multiple-choice or short-answer. The default is based on whether there is an answer given in column `A`, 'MC' if there is, 'Essay' otherwise.
- Text Type: The type of text in all text columns, 'html' or 'plain', for html-formatted text or plain text. The default is 'html'.
- Points: Numeric, number of points for the problem. The default is 1.
- A: Correct answer for a multiple choice question, a number 1 thru 4. Default is NA.
- 'Choice 1': Choice 1, default is empty character string ""
- 'Choice 2': Choice 2, default is empty character string ""
- 'Choice 3': Choice 3, default is empty character string ""
- 'Choice 4': Choice 4, default is empty character string ""
- Feedback: General feedback given to students after they complete the quiz and answers are shown. Default is empty character string "".

The only required column of the input data frame is `Question`. Any other missing columns will be created with default values

Usage

```
read_quiz_csv(filepath)
```

Arguments

`filepath`	Path to the CSV file, hopefully mostly conforming to the above.

Value

Quiz data frame with the columns above

`read_quiz_docx` *Read a quiz in from a Word file and return a quiz data frame.*

Description

This function reads a quiz in from a Word file. The quiz must be in a table in the document. The functions returns a quiz data frame that conforms to the following:

- G: Question group
- Question: Text of the question
- Question type: Either MC or Essay, for multiple-choice or short-answer. The default is based on whether there is an answer given in column `A`, 'MC' if there is, 'Essay' otherwise.
- Text Type: The type of text in all text columns, 'html' or 'plain', for html-formatted text or plain text. The default is 'html'.
- Points: Numeric, number of points for the problem. The default is 1.
- A: Correct answer for a multiple choice question, a number 1 thru 4. Default is NA.
- 'Choice 1': Choice 1, default is empty character string ""
- 'Choice 2': Choice 2, default is empty character string ""
- 'Choice 3': Choice 3, default is empty character string ""
- 'Choice 4': Choice 4, default is empty character string ""
- Feedback: General feedback given to students after they complete the quiz and answers are shown. Default is empty character string "".

The only required column of the input data frame is `Question`. Any other missing columns will be created with default values

Usage

```
read_quiz_docx(filepath, tbl_number = 1)
```

Arguments

`filepath`	Path to the Word file, with a table hopefully mostly conforming to the above.
`tbl_number`	If there is more than one table in the file, specify the table number you would like to import. Default is 1.

Value

Quiz data frame with the columns above

`read_quiz_excel` *Read a quiz in from an Excel file and return a quiz data frame*

Description

This function reads a quiz in from an Excel file and return a quiz data frame that conforms to the following:

- G: Question group
- Question: Text of the question
- Question type: Either MC or Essay, for multiple-choice or short-answer. The default is based on whether there is an answer given in column `A`, 'MC' if there is, 'Essay' otherwise.
- Text Type: The type of text in all text columns, 'html' or 'plain', for html-formatted text or plain text. The default is 'html'.
- Points: Numeric, number of points for the problem. The default is 1.
- A: Correct answer for a multiple choice question, a number 1 thru 4. Default is NA.
- 'Choice 1': Choice 1, default is empty character string ""
- 'Choice 2': Choice 2, default is empty character string ""
- 'Choice 3': Choice 3, default is empty character string ""
- 'Choice 4': Choice 4, default is empty character string ""
- Feedback: General feedback given to students after they complete the quiz and answers are shown. Default is empty character string "".

The only required column of the input data frame is `Question`. Any other missing columns will be created with default values

Usage

```
read_quiz_excel(filepath)
```

Arguments

`filepath`	Path to the Excel file, hopefully mostly conforming to the above.

Value

Quiz data frame with the columns above

`read_quiz_googledoc` *Read a quiz in from a Google Docs file and return a quiz data frame.*

Description

This function reads a quiz in from a Google Docs file. The quiz must be in a table in the document. The function has the side effect of saving the Google Doc as a Word .docx file. The functions returns a quiz data frame that conforms to the following:

- G: Question group
- Question: Text of the question
- Question type: Either MC or Essay, for multiple-choice or short-answer. The default is based on whether there is an answer given in column `A`, 'MC' if there is, 'Essay' otherwise.
- Text Type: The type of text in all text columns, 'html' or 'plain', for html-formatted text or plain text. The default is 'html'.
- Points: Numeric, number of points for the problem. The default is 1.
- A: Correct answer for a multiple choice question, a number 1 thru 4. Default is NA.
- 'Choice 1': Choice 1, default is empty character string ""
- 'Choice 2': Choice 2, default is empty character string ""
- 'Choice 3': Choice 3, default is empty character string ""
- 'Choice 4': Choice 4, default is empty character string ""
- Feedback: General feedback given to students after they complete the quiz and answers are shown. Default is empty character string "".

The only required column of the input data frame is `Question`. Any other missing columns will be created with default values

The function will also save a .docx version of the Google Doc in the default file path.

Usage

```
read_quiz_googledoc(doc_url, noauth = FALSE, tbl_number = 1, overwrite = TRUE)
```

Arguments

`doc_url`	Path to the Google Docs file, with a table hopefully mostly conforming to the above.
`noauth`	Logical, default is FALSE. Set equal to TRUE if the spreadsheet is accessible without Google login authorization and you would like to skip the login. Turns off authorization using `googledrive::deauth()`.
`tbl_number`	If there is more than one table in the file, specify the table number you would like to import. Default is 1.
`overwrite`	The function has the side effect of saving the Google Doc as a Word .docx file. If the .docx file already exists, set to TRUE or FALSE to overwrite the .docx file. The default is equal to TRUE.

Value

Quiz data frame with the columns above

`read_quiz_googlesheet` *Read a quiz in from a Google Sheet and return a quiz data frame*

Description

This function reads a quiz in from a Google Sheet and return a quiz data frame that conforms to the following:

- G: Question group
- Question: Text of the question
- Question type: Either MC or Essay, for multiple-choice or short-answer. The default is based on whether there is an answer given in column `A`, 'MC' if there is, 'Essay' otherwise.
- Text Type: The type of text in all text columns, 'html' or 'plain', for html-formatted text or plain text. The default is 'html'.
- Points: Numeric, number of points for the problem. The default is 1.
- A: Correct answer for a multiple choice question, a number 1 thru 4. Default is NA.
- 'Choice 1': Choice 1, default is empty character string ""
- 'Choice 2': Choice 2, default is empty character string ""
- 'Choice 3': Choice 3, default is empty character string ""
- 'Choice 4': Choice 4, default is empty character string ""
- Feedback: General feedback given to students after they complete the quiz and answers are shown. Default is empty character string "".

The only required column of the input data frame is `Question`. Any other missing columns will be created with default values

Usage

```
read_quiz_googlesheet(
  sheet_url,
  noauth = FALSE,
  sheet = NULL,
  range = NULL,
  na = "",
  trim_ws = TRUE,
  skip = 0
)
```

Arguments

`sheet_url`	URL to a Google Sheet, hopefully mostly conforming to the above.
`noauth`	Logical, default is FALSE. Set equal to TRUE if the spreadsheet is accessible without Google login authorization and you would like to skip the login. Turns off authorization using `googlesheets4::gs4_deauth()`.

sheet	Same as the sheet parameter passed to googlesheets4::read_sheet(). Default is NULL. Sheet to read, in the sense of "worksheet" or "tab". You can identify a sheet by name, with a string, or by position, with a number. Ignored if the sheet is specified via range. If neither argument specifies the sheet, defaults to the first visible sheet.
range	Same as the range parameter passed to googlesheets4::read_sheet(). Default is NULL. A cell range to read from. If NULL, all non-empty cells are read. Otherwise specify range as described in Sheets A1 notation or using the helpers documented in cell-specification. Sheets uses fairly standard spreadsheet range notation, although a bit different from Excel. Examples of valid ranges: "Sheet1!A1:B2", "Sheet1!A:A", "Sheet1!1:2", "Sheet1!A5:A", "A1:B2", "Sheet1". Interpreted strictly, even if the range forces the inclusion of leading, trailing, or embedded empty rows or columns. Takes precedence over skip, n_max and sheet. Note range can be a named range, like "sales_data", without any cell reference.
na	Same as the na parameter passed to googlesheets4::read_sheet(). Default is "". Character vector of strings to interpret as missing values. By default, blank cells are treated as missing data.
trim_ws	Same as the trim_ws parameter passed to googlesheets4::read_sheet(). Logical, default is TRUE. Should leading and trailing whitespace be trimmed from cell contents?
skip	Same as the skip parameter passed to googlesheets4::read_sheet(). Default is 0. Minimum number of rows to skip before reading anything, be it column names or data. Leading empty rows are automatically skipped, so this is a lower bound. Ignored if range is given.

Value

Quiz data frame with the columns above

---

render_pdf_quiz *Render a TeX and PDF file of a quiz*

Description

This function takes a dataframe with a quiz and creates Markdown file for the quiz, and renders to TeX and PDF files

Usage

```
render_pdf_quiz(
  quiz.df,
  outfile,
  outfolder,
  quiz_title,
  quiz_subtitle = "",
  instructor = "",
  includeanswers = TRUE,
  version = "a",
  seed = 1
)
```

Arguments

`quiz.df`	data.frame where each row is a question. The column names should be the following: • G: Question group • Question: Text of the question • A: Correct answer, a number 1 thru 4 • 'Choice 1': Choice 1 • 'Choice 2': Choice 2 • 'Choice 3': Choice 3 • 'Choice 4': Choice 4 • Feedback: General feedback given to students after they complete the quiz and answers are shown
`outfile`	String with the name of the output file, but do not include the extension. File names with extensions .md, .tex, and .pdf will be created
`outfolder`	String with the name of the folder to save the output
`quiz_title`	String with the title of the quiz
`quiz_subtitle`	String with a subtitle for the quiz, default is ""
`instructor`	String with the instructor name, default is ""
`includeanswers`	Boolean, set equal to TRUE to include answers in the output, FALSE to not include answers in the output. Default value is TRUE.
`version`	Character, set to "a" or "b" for a version number. If there are more than two questions in any group of questions, for the name random number seed, the two versions will have two different questions. Default value is "a".

Index