

Package ‘GIFTr’

January 20, 2025

Type Package

Title GIFT Questions Format Generator from Dataframes

Version 0.1.0

Author Omar I. Elashkar

Maintainer Omar I. Elashkar <omar.ibrahim@miuegypt.edu.eg>

Depends R (>= 3.5.0)

Imports stringr, glue

Description

A framework and functions to create 'MOODLE' quizzes. 'GIFTr' takes dataframe of questions of four types: multiple choices, numerical, true or false and short answer questions, and exports a text file formatted in 'MOODLE' GIFT format. You can prepare a spreadsheet in any software and import it into R to generate any number of questions with 'HTML', 'markdown' and 'LaTeX' support.

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

URL <https://github.com/omarelashkar/GIFTr>

Suggests knitr, rmarkdown, kableExtra,

VignetteBuilder knitr

NeedsCompilation no

Repository CRAN

Date/Publication 2019-10-20 11:30:05 UTC

Contents

GIFTr	2
GIFTrData	5
GIFTrData_2	6
mcq	7

num_q	9
short_ans	12
tf_q	14

Index	17
--------------	-----------

GIFTr *Generate A GIFT File of Different Question Types From a Spreadsheet*

Description

GIFTr function is the main function of 'GIFTr' package. You provide it dataframe and select question_type column where a more specialized functions can process the file. See Details and Examples

Usage

```
GIFTr(data, output, questions, answers, question_type, categories = NULL,
       question_names = NULL, mcq_answer_column = FALSE, verbose = TRUE)
```

Arguments

data	dataframe or tibble of the questions data
output	directory of .txt file name the questions will be exported to.
questions	name(string) or index(integer) of the questions column
answers	a vector of names(strings) or indices of answers column(s)
question_type	name(string) or index(integer) of the questions type column.
categories	name(string) or index(integer) of categories column if available, Default: NULL
question_names	name(string) or index(integer) of the questions names column. If NULL, it will be the first 40 letters of the question title, Default: NULL
mcq_answer_column	If TRUE, the first column of answers columns will be set as the right answer, Default: FALSE
verbose	If TRUE, the functions will print to the console the statistics of writing the output, Default: TRUE

Details

'GIFTr' package is intended to reduce the time a course creator would take to make input question on 'MOODLE' without the pain of writing markup.

'GIFTr' package is build on '[MOODLE' guidelines](#). The idea is simple, you create a spreadsheet in a special format, call GIFTr function, get a GIFT formatted file that can be imported by 'MOODLE' and other LMS systems.

GIFTr function is unique in that it gives you detailed *statistics* and can work with *the 4 question types* supported by 'GIFTr' package. question_type argument is unique for GIFTr function in

this package and must be passed to map the questions. The current supported question types are `mcq`{multiple choices question}, `num_q`{numeric entry}, `tf_q`{true or false}, and `short_ans`{short answer} questions. You can find more details on the individual functions details.

'GIFTr' supports basic markdown and GIFT syntax. See the vignette and sections below for complete documentation of formatting your data.

Value

None

Formatting Your Data

A guideline for creating you questions data can be found below. Check the data `GIFTrData` and `GIFTrData_2` as example for formatted questions.

Markdown, HTML Support and LATEX support: 'MOODLE' itself supports basic markdown and HTML for questions formatting. So when formatting your data you can use HTML tags like `<sub>` and `<sup>`. Also you can use markdown **`**bold**`** or `__bold__` and *`*italic*`* or `_italic_` ...etc. However it is better a better practice to avoid using asterisk to avoid confusion with MCQ format. For more about the supported markdown see '**MOODLE**' [documentation](#).

LATEX Support: 'MOODLE' also supports inline and block LATEX equations through `math-jax`, however you have to be careful with the **special characters** like curly brackets `//` and equal sign `=`, so you have to use back slash before those to ensure you can import correctly.

Note that if you see thee data in console, you will find it with 2 backslash `\`, however that's the escaping of the backslash in R and you write with with single slash normally. Check `GIFTrData` `GIFTrData[11, 3]` for an example. For further details on LATEX, check '**MOODLE**' [docs](#).

Answer Feedback: You can easily choose to enter a feedback by using `#sign` after the answer you want to specify a feedback on. Check `GIFTrData` for examples.

Data columns: The data passed would differ slightly according to question type, however generally you need to have:

1. a column contains question. This cannot contain empty values
2. answer(s) column(s). This may be multiple columns if you have multiple answers. If you mix single and multiple answer it is better to write the single answer in the first column. If you have **only single answer MCQ and NO multiple answer MCQ**, you can set the answers without asterisk in the first column of the answers columns. More details in the vignette and below.
3. a column specifying the type of question. The current supported questions are multiple choices, numerical entry, true or false and short answer questions. The should be named `'mcq'`, `'num_q'`, `'tf_q'` and `'short_ans'` respectively.
4. a column specifying categories and subcategories in the 'MOODLE' categories are important when you are preparing a quiz as you may want to specify certain proportions of inclusions. Categories and subcategories are spaced by forward slash like `Categ1/subcateg1/subsubcateg1` ...etc.

5. a column specifying question names. So you can easily enter a certain question names by like certain ID or keyword to make the questions easily on the system. however you don't need to worry about that as automatically if not set, the first 40 letter are set a question name.

Formatting Multiple Choices Questions

Specifying mcq answer: You can specify answers simply using asterisk in a the start on the answer. If you choose more than one answer, the function will generate a multiple answers mcq with every answer holds partial even credit. See [GIFTrData](#) for this usage.

If you intend to **use single answer only**, you might specify 'make_answer' or 'mcq_answer_column' to TRUE, this will consider the first answer column to be the answer. For example, if the answers are in columns c(5,9), the answers will be listed in the first column, 5. See [GIFTrData_2](#) for this usage.

Numeric Entry Questions Formatting

Numeric Answer can be in single column or multiple columns. You can also format it as range or interval limit for the answer and partial credit. For further illustration, check the vignette.

Numeric Range and Intervals: If you want the answer to be between 1 and 2, you can enter you data as '1..2'. If you want to set an acceptance limit, for example your answer is '158' and you want to set limit plus or minus 2, you can write it as '158:2'. Check the [GIFTrData](#) for examples.

Multiple Numeric Answers with partial credits: You enter multiple numeric answers with the same concepts above, but if you did not enter the credits for an answer, 'MOODLE' will consider it '100%'. So you have to specify the credit at the start of an answer. For example '122' and '%50%122:5'.

True or False Questions Formatting

True or False answers is to be set in 1 column with letter 'T' or 'F' insensitive to the case.

For further illustration, check [GIFTrData](#).

Short Answer Questions Formatting

Short answer question answers can be in single column or multiple columns. If an answer has not credit it will be given 100% credit automatically. For example if the answer is 'statistics', it will be equivalent to '%100%statistics'. While '%80%Data Science' answer will take 80% of the credit

For further illustration, check [GIFTrData](#).

See Also

[mcq](#), [num_q](#), [tf_q](#), [short_ans](#)

Examples

```

#' load Data and Check structure
data(GIFTrData)
str(GIFTrData)

GIFTr::GIFTr(data = GIFTrData, questions = 3,
  answers = c(4:8), categories = 1,
  question_type = 9,
  output = file.path(tempdir(), "quiz.txt"))
#write file"quiz.txt" in tempdir()

GIFTr::GIFTr(data = GIFTrData, question_names = 2,
  questions = 3, answers = c(4:8),
  categories = 1, question_type = 9,
  output = file.path(tempdir(), "quiz2.txt"))
#write file"quiz2.txt" in tempdir()

```

GIFTrData

Questions Data with asterisk and multiple answer mcq

Description

DATASET_DESCRIPTION

Usage

data(GIFTrData)

Format

A data frame with 10 rows and 9 variables:

categories character categories of questions to be imported to 'MOODLE' to hierarchical order

names character names of the questions

question character question text

A character first answer option

B character second answer option

C character third answer option

D character fourth answer option

E character fifth answer option

question_type character question type value should be either 'mcq', 'num_q', 'tf_q', 'short_ans'

Details

This data is a demo on a typical structure of data passed in 'GIFTr' package. This data mcq question contains asterisk in the first 2 questions, one with single answer and the other is multiple answers question.

Formatting Tips

markdown: Markdown is supported in GIFT format and example for it's usage in first and second question text.

HTML: Simple HTML syntax is supported as in question 10

LATEX: LATEX is supported in GIFT but you have to escape any curly brackets as in question 11

Partial Credits: Partial Credits for a question can be specified if like in question 9, you could also use them in short answer question. If you have a short answer question or numeric question with no specified credits, this answer will take "100%" credit

Answer Feedback: Every answer can be accompanied with feedback using # as in questions 1 and 9

See Also

[GIFTrData_2 GIFTr mcq](#)

GIFTrData_2

Questions Data without asterisk

Description

This data is a manipulated copy of [GIFTrData](#) data with removal of the multiple answers mcq and removal of asterisk from the single answer mcq.

Usage

```
data(GIFTrData_2)
```

Format

A data frame with 10 rows and 9 variables:

categories character categories of questions to be imported to 'MOODLE' to hierarchical order

names character names of the questions

question character question text

A character first answer option, this column must be the answer column for mcq

B character second answer option

C character third answer option

D character fourth answer option

E character fifth answer option

question_type character question type value should be either 'mcq', 'num_q', 'tf_q', 'short_ans'

Details

This data is an altered form of [GIFTrData](#). Check its usage in [mcq](#). For further details on the data structure, check [GIFTr](#) documentation.

See Also

[GIFTrData](#) [GIFTr](#) [mcq](#)

mcq

Generate GIFT MCQs From Spreadsheet

Description

Create GIFT file with single & multiple answers MCQs from a spreadsheet to be exported to LMS.

Usage

```
mcq(data, questions, answers, categories = NULL, question_names = NULL,
     output, make_answer = FALSE, verbose = TRUE)
```

Arguments

data	dataframe or tibble of mcq questions data
questions	name(string) or index(integer) of the questions column
answers	a vector of names(strings) or indices of answers column(s)
categories	name(string) or index(integer) of categories column if available, Default: NULL
question_names	name(string) or index(integer) of the questions names column. If NULL, it will be the first 40 letters of the question title, Default: NULL
output	string of .txt file name and path where the questions will be exported to.
make_answer	If TRUE, the first column of answers columns will be set as the right answer, Default: FALSE.
verbose	If TRUE, the functions will print to the console the statistics of writing the output, Default: TRUE

Details

mcq function takes a dataframe with multiple choices questions(mcq) and export a text file in 'MOODLE' GIFT format. The function automatically makes an mcq a single answer or multiple answers depends on asterisks present in the answers column. If you have additional column of question_type set to 'mcq' you can also use [GIFTr](#) function which wraps all question generating functions.

See Vignette and [GIFTrData](#) for demos.

Value

None

Formatting Multiple Choices Questions

Specifying mcq answer: You can specify answers simply using asterisk in the start on the answer. If you choose more than one answer, the function will generate a multiple answers mcq with every answer holds partial even credit. See [GIFTrData](#) for this usage.

If you intend to **use single answer only**, you might specify 'make_answer' or 'mcq_answer_column' to TRUE, this will consider the first answer column to be the answer. For example, if the answers are in columns c(5,9), the answers will be listed in the first column, 5. See [GIFTrData_2](#) for this usage.

Formatting Your Data

A guideline for creating your questions data can be found below. Check the data [GIFTrData](#) and [GIFTrData_2](#) as example for formatted questions.

Markdown, HTML Support and LATEX support: 'MOODLE' itself supports basic markdown and HTML for questions formatting. So when formatting your data you can use HTML tags like <sub> and <sup>. Also you can use markdown ****bold**** or **__bold__** and ***italic*** or **_italic_** ...etc. However it is better a better practice to avoid using asterisk to avoid confusion with MCQ format. For more about the supported markdown see '[MOODLE](#)' [documentation](#).

LATEX Support: 'MOODLE' also supports inline and block LATEX equations through [mathjax](#), however you have to be careful with the [special characters](#) like curly brackets // and equal sign = , so you have to use back slash before those to ensure you can import correctly.

Note that if you see thee data in console, you will find it with 2 backslash \, however that's the escaping of the backslash in R and you write with with single slash normally. Check [GIFTrData](#) [GIFTrData\[11, 3\]](#)for an example. For further details on LATEX, check '[MOODLE](#)' [docs](#).

Answer Feedback: You can easily choose to enter a feedback by using #sign after the answer you want to specify a feedback on. Check [GIFTrData](#) for examples.

Data columns: The data passed would differ slightly according to question type, however generally you need to have:

1. a column contains question. This cannot contain empty values

2. answer(s) column(s). This may be multiple columns if you have multiple answers. If you mix single and multiple answer it is better to write the single answer in the first column. If you have **only single answer MCQ and NO multiple answer MCQ**, you can set the answers without asterisk in the first column of the answers columns. More details in the vignette and below.
3. a column specifying the type of question. The current supported questions are multiple choices, numerical entry, true or false and short answer questions. The should be named 'mcq', 'num_q', 'tf_q' and 'short_ans' respectively.
4. a column specifying categories and subcategories in the 'MOODLE' categories are important when you are preparing a quiz as you may want to specify certain proportions of inclusions. Categories and subcategories are spaced by forward slash like Categ1/subcateg1/subsubcateg1...etc.
5. a column specifying question names. So you can easily enter a certain question names by like certain ID or keyword to make the questions easily on the system. however you don't need to worry about that as automatically if not set, the first 40 letter are set a question name.

See Also

[GIFTr](#)

Examples

```
#data with asterisk and multiple answer mcq(Q2 question)
data(GIFTrData)
mcqdata <- GIFTrData[which(GIFTrData$question_type == "mcq"),]

mcq(data = mcqdata, questions = 3,
     answers = c(4:8), categories = 1,
     question_names = 2, output = file.path(tempdir(), "mcq.txt"))
#No make_answer argument, question_name specified
#write file "mcq.txt" in tempdir()

#data with no atrisk and no multiple answer mcq
mcqdata_2 <- GIFTrData_2[which(GIFTrData_2$question_type == "mcq"),]

mcq(data = mcqdata_2, questions = 3,
     answers = c(4:8), categories = 1,
     question_names = 2, make_answer = TRUE,
     output = file.path(tempdir(), "mcq_1.txt")) #Answer is in column 4
#write file "mcq_1.txt" in tempdir()
```

num_q

Generate GIFT Numeric Questions From Spreadsheet

Description

Create GIFT file with numeric entry questions from a spreadsheet to be exported to LMS.

Usage

```
num_q(data, questions, answers, categories, question_names = NULL,
      output, verbose = TRUE)
```

Arguments

data	dataframe or tibble of numerical entry questions data
questions	name(string) or index(integer) of the questions column
answers	a vector of names(strings) or indices of answers column(s)
categories	name(string) or index(integer) of categories column if available, Default: NULL
question_names	name(string) or index(integer) of the questions names column. If NULL, it will be the first 40 letters of the question title, Default: NULL
output	string of .txt file name and path where the questions will be exported to.
verbose	If TRUE, the functions will print to the console the statistics of writing the output, Default: TRUE

Details

num_q function takes a dataframe with numeric entry questions and export a text file in 'MOODLE' GIFT format. The function automatically makes an numeric entry question either single or multiple according to your data format(check numeric questions formatting below). If you have additional column of question_type set to 'num_q' you can also use [GIFTr](#) function which wraps all question generating functions.

See Vignette and [GIFTrData](#) for demos.

Value

None

Numeric Entry Questions Formatting

Numeric Answer can be in single column or multiple columns. You can also format it as range or interval limit for the answer and partial credit. For further illustration, check the vignette.

Numeric Range and Intervals: If you want the answer to be between 1 and 2, you can enter you data as '1..2'. If you want to set an acceptance limit, for example your answer is '158' and you want to set limit plus or minus 2, you can write it as '158:2'. Check the [GIFTrData](#) for examples.

Multiple Numeric Answers with partial credits: You enter multiple numeric answers with the same concepts above, but if you did not enter the credits for an answer, 'MOODLE' will consider it '100%'. So you have to specify the credit at the start of an answer. For example '122' and '%50%122:5'.

Formatting Your Data

A guideline for creating your questions data can be found below. Check the data [GIFTrData](#) and [GIFTrData_2](#) as example for formatted questions.

Markdown, HTML Support and LATEX support: 'MOODLE' itself supports basic markdown and HTML for questions formatting. So when formatting your data you can use HTML tags like `<sub>` and `<sup>`. Also you can use markdown **bold** or `__bold__` and *italic* or `_italic_` ...etc. However it is better a better practice to avoid using asterisk to avoid confusion with MCQ format. For more about the supported markdown see '[MOODLE](#)' [documentation](#).

LATEX Support: 'MOODLE' also supports inline and block LATEX equations through [mathjax](#), however you have to be careful with the [special characters](#) like curly brackets `//` and equal sign `=`, so you have to use back slash before those to ensure you can import correctly.

Note that if you see thee data in console, you will find it with 2 backslash `\`, however that's the escaping of the backslash in R and you write with with single slash normally. Check [GIFTrData](#) `GIFTrData[11,3]` for an example. For further details on LATEX, check '[MOODLE](#)' [docs](#).

Answer Feedback: You can easily choose to enter a feedback by using `#sign` after the answer you want to specify a feedback on. Check [GIFTrData](#) for examples.

Data columns: The data passed would differ slightly according to question type, however generally you need to have:

1. a column contains question. This cannot contain empty values
2. answer(s) column(s). This may be multiple columns if you have multiple answers. If you mix single and multiple answer it is better to write the single answer in the first column. If you have **only single answer MCQ and NO multiple answer MCQ**, you can set the answers without asterisk in the first column of the answers columns. More details in the vignette and below.
3. a column specifying the type of question. The current supported questions are multiple choices, numerical entry, true or false and short answer questions. The should be named `'mcq'`, `'num_q'`, `'tf_q'` and `'short_ans'` respectively.
4. a column specifying categories and subcategories in the 'MOODLE' categories are important when you are preparing a quiz as you may want to specify certain proportions of inclusions. Categories and subcategories are spaced by forward slash like `Categ1/subcateg1/subsubcateg1` ...etc.
5. a column specifying question names. So you can easily enter a certain question names by like certain ID or keyword to make the questions easily on the system. however you don't need to worry about that as automatically if not set, the first 40 letter are set a question name.

See Also

[GIFTr](#)

Examples

```
data(GIFTrData)
#data with numeric entry questions
numq_data <- GIFTrData[which(GIFTrData$question_type == "num_q"),]
```

```
num_q(data = numq_data, questions = 3,
      answers = c(4:8), categories = 1,
      question_names = 2, output = file.path(tempdir(), "numq.txt"))
#write file "numq.txt" in tempdir()
```

 short_ans

Generate 'GIFT' Short Answer From Spreadsheet

Description

Create 'GIFT' file with short answer questions from a spreadsheet to be exported to LMS.

Usage

```
short_ans(data, questions, answers, categories, question_names = NULL,
          output, verbose = TRUE)
```

Arguments

data	dataframe or tibble of short answer questions data
questions	name(string) or index(integer) of the questions column
answers	a vector of names(strings) or indices of answers column(s)
categories	name(string) or index(integer) of categories column if available, Default: NULL
question_names	name(string) or index(integer) of the questions names column. If NULL, it will be the first 40 letters of the question title, Default: NULL
output	string of .txt file name and path where the questions will be exported to.
verbose	If TRUE, the functions will print to the console the statistics of writing the output, Default: TRUE

Details

short_ans function takes a dataframe with short answer questions and export a text file in 'MOODLE' GIFT format. The function automatically makes a short answer question either single or multiple with or without different credit weight according to your data format(check short answer questions formatting below). If you have additional column of question_type set to 'short_ans' you can also use [GIFTr](#) function which wraps all question generating functions.

See Vignette and [GIFTrData](#) for demos.

Value

None

Short Answer Questions Formatting

Short answer question answers can be in single column or multiple columns. If an answer has not credit it will be given 100% credit automatically. For example if the answer is 'statistics', it will be equivalent to '%100%statistics'. While '%80%Data Science' answer will take 80% of the credit

For further illustration, check [GIFTrData](#).

Formatting Your Data

A guideline for creating you questions data can be found below. Check the data [GIFTrData](#) and [GIFTrData_2](#) as example for formatted questions.

Markdown, HTML Support and LATEX support: 'MOODLE' itself supports basic mark-down and HTML for questions formatting. So when formatting your data you can use HTML tags like <sub> and <sup>. Also you can use markdown ****bold**** or `__bold__` and **italic** or `_italic_` ...etc. However it is better a better practice to avoid using asterisk to avoid confusion with MCQ format. For more about the supported markdown see '[MOODLE](#)' [documentation](#).

LATEX Support: 'MOODLE' also supports inline and block LATEX equations through [math-jax](#), however you have to be careful with the [special characters](#) like curly brackets // and equal sign = , so you have to use back slash before those to ensure you can import correctly.

Note that if you see thee data in console, you will find it with 2 backslash \, however that's the escaping of the backslash in R and you write with with single slash normally. Check [GIFTrData](#) `GIFTrData[11,3]`for an example. For further details on LATEX, check '[MOODLE](#)' [docs](#).

Answer Feedback: You can easily choose to enter a feedback by using #sign after the answer you want to specify a feedback on. Check [GIFTrData](#) for examples.

Data columns: The data passed would differ slightly according to question type, however generally you need to have:

1. a column contains question. This cannot contain empty values
2. answer(s) column(s). This may be multiple columns if you have multiple answers. If you mix single and multiple answer it is better to write the single answer in the first column. If you have **only single answer MCQ and NO multiple answer MCQ**, you can set the answers without asterisk in the first column of the answers columns. More details in the vignette and below.
3. a column specifying the type of question. The current supported questions are multiple choices, numerical entry, true or false and short answer questions. The should be named 'mcq', 'num_q', 'tf_q' and 'short_ans' respectively.
4. a column specifying categories and subcategories in the 'MOODLE' categories are important when you are preparing a quiz as you may want to specify certain proportions of inclusions. Categories and subcategories are spaced by forward slash like `Categ1/subcateg1/subsubcateg1` ...etc.
5. a column specifying question names. So you can easily enter a certain question names by like certain ID or keyword to make the questions easily on the system. however you don't need to worry about that as automatically if not set, the first 40 letter are set a question name.

See Also[GIFTr](#)**Examples**

```

data(GIFTrData)
#data with short answer question type
shortans_data <- GIFTrData[which(GIFTrData$question_type == "short_ans"),]

short_ans(data = shortans_data, questions = 3,
  answers = c(4:8), categories = 1,
  question_names = 2, output = file.path(tempdir(), "shortq.txt"))
#write file "shortq.txt" in tempdir()

```

tf_q

*Generate GIFT True or False Questions From Spreadsheet***Description**

Create GIFT file with true or false questions from a spreadsheet to be exported to LMS.

Usage

```
tf_q(data, questions, answers, categories, question_names = NULL, output,
  verbose = TRUE)
```

Arguments

data	dataframe or tibble of true or false questions data
questions	name(string) or index(integer) of the questions column
answers	a vector of names(strings) or indices of answers column(s)
categories	name(string) or index(integer) of categories column if available, Default: NULL
question_names	name(string) or index(integer) of the questions names column. If NULL, it will be the first 40 letters of the question title, Default: NULL
output	string of .txt file name and path where the questions will be exported to.
verbose	If TRUE, the functions will print to the console the statistics of writing the output, Default: TRUE

Details

tf_q function takes a dataframe with true or false questions and export a text file in 'MOODLE' GIFT format. The function automatically makes a true or false when it detect 'T' or 'F' in the answers vector regardless of case. If you have additional column of question_type set to 'tf_q' you can also use [GIFTr](#) function which wraps all question generating functions.

See Vignette and [GIFTrData](#) for demos.

Value

None

True or False Questions Formatting

True or False answers is to be set in 1 column with letter 'T' or 'F' insensitive to the case.

For further illustration, check [GIFTrData](#).

Formatting Your Data

A guideline for creating you questions data can be found below. Check the data [GIFTrData](#) and [GIFTrData_2](#) as example for formatted questions.

Markdown, HTML Support and LATEX support: 'MOODLE' itself supports basic mark-down and HTML for questions formatting. So when formatting your data you can use HTML tags like <sub> and <sup>. Also you can use markdown ****bold**** or `__bold__` and **italic** or `_italic_` ...etc. However it is better a better practice to avoid using asterisk to avoid confusion with MCQ format. For more about the supported markdown see '[MOODLE](#)' [documentation](#).

LATEX Support: 'MOODLE' also supports inline and block LATEX equations through [math-jax](#), however you have to be careful with the [special characters](#) like curly brackets // and equal sign = , so you have to use back slash before those to ensure you can import correctly.

Note that if you see thee data in console, you will find it with 2 backslash \, however that's the escaping of the backslash in R and you write with with single slash normally. Check [GIFTrData](#) `GIFTrData[11,3]`for an example. For further details on LATEX, check '[MOODLE](#)' [docs](#).

Answer Feedback: You can easily choose to enter a feedback by using #sign after the answer you want to specify a feedback on. Check [GIFTrData](#) for examples.

Data columns: The data passed would differ slightly according to question type, however generally you need to have:

1. a column contains question. This cannot contain empty values
2. answer(s) column(s). This may be multiple columns if you have multiple answers. If you mix single and multiple answer it is better to write the single answer in the first column. If you have **only single answer MCQ and NO multiple answer MCQ**, you can set the answers without asterisk in the first column of the answers columns. More details in the vignette and below.
3. a column specifying the type of question. The current supported questions are multiple choices, numerical entry, true or false and short answer questions. The should be named 'mcq', 'num_q', 'tf_q' and 'short_ans' respectively.
4. a column specifying categories and subcategories in the 'MOODLE' categories are important when you are preparing a quiz as you may want to specify certain proportions of inclusions. Categories and subcategories are spaced by forward slash like `Categ1/subcateg1/subsubcateg1` ...etc.
5. a column specifying question names. So you can easily enter a certain question names by like certain ID or keyword to make the questions easily on the system. however you don't need to worry about that as automatically if not set, the first 40 letter are set a question name.

See Also

[GIFTr](#)

Index

* datasets

GIFTrData, [5](#)

GIFTrData_2, [6](#)

GIFTr, [2](#), [6–12](#), [14](#), [16](#)

GIFTrData, [3](#), [4](#), [5](#), [6–8](#), [10–15](#)

GIFTrData_2, [3](#), [4](#), [6](#), [6](#), [8](#), [11](#), [13](#), [15](#)

mcq, [3](#), [4](#), [6](#), [7](#), [7](#)

num_q, [3](#), [4](#), [9](#)

short_ans, [3](#), [4](#), [12](#)

tf_q, [3](#), [4](#), [14](#)