

Package ‘grade’

July 22, 2025

Version 0.2-1

Date 2013-11-15

Title Binary Grading functions for R.

Author Leif Johnson <leif.t.johnson@gmail.com>

Maintainer Leif Johnson <leif.t.johnson@gmail.com>

URL <https://github.com/ltjohnson/grade>

Depends R (>= 2.4.1)

Description

Provides functions for matching student-answers to teacher answers for a variety of data types.

License GPL-2

NeedsCompilation no

Repository CRAN

Date/Publication 2013-11-16 08:07:28

Contents

grade-package	2
grade.discreteprobability	3
grade.interval	5
grade.negative	6
grade.number	7
grade.parse	8
grade.set	10
grade.truefalse	11
Index	13

grade-package

Grade

Description

Binary Grading functions for R.

Details

```

Package:    grade
Version:    0.2
Date:       2009-02-20
Title:      Grade
Author:     Leif Johnson <leif.t.johnson@gmail.com>
Maintainer: Leif Johnson <leif.t.johnson@gmail.com>
URL:        http://www.stat.umn.edu/~leif/software/grade/
Depends:    R (>= 2.4.1)
Description: Binary Grading functions for R.
License:    GPL-2
Packaged:   Fri Feb 20 10:28:59 2009; leif

```

Index:

```

\link{grade.discreteprobability}    Grade Discrete Probability Sets
\link{grade.interval}               Grade Intervals
\link{grade.isscalar}               Check if an object is a scalar
\link{grade.negative}               Check the Sign of a Number
\link{grade.number}                 Grade Single Numbers
\link{grade.orderedset}             Grade Ordered Sets
\link{grade.parse}                  Parse input
\link{grade.parsechunk}
\link{grade.parseset}
\link{grade.set}                    Grade Sets
\link{grade.truefalse}              True/False answers

```

Note

There are some common arguments across all of the grade functions. These are:

- `correctansInput` to be the *correct* answer. May be a string or a vector. Checks are likely to be more stringent on this component and result in more errors. E.g. `grade.interval` requires that `correctans` have length 2.
- `studentansInput` to check for correctness. May be a string or a vector. Most of the grade functions check it against `correctans`

- useevalTRUE or FALSE. If TRUE `eval` is used to evaluate text elements. If FALSE `as.numeric` is used to evaluate text elements. The advantage of using `eval` is more forgiveness for input, e.g. `eval` of "pi" returns 3.1415, or `eval` of "1/2" returns 0.5, but `as.numeric` returns NA in each case. The disadvantage is that `eval` could be abused to run arbitrary code leading to a security issue. However, the `grade` package does not submit any text to either `eval` or `as.numeric` that contains any of the characters '[', ']', '(', ')', '<', '>', '=' or ','. It is unlikely that code containing function calls could be inserted. So `useeval` defaults to TRUE. If there are problems, or you are worried, you can always set `useeval=FALSE`.
- usenaTRUE or FALSE. If TRUE, NA is considered to be a valid number. If FALSE, NA is considered to be invalid. Default is `usena=FALSE`.
- useinfTRUE or FALSE. If TRUE, Inf and -Inf are considered to be valid numbers. If FALSE, Inf and -Inf are considered to be invalid. Default is `useinf=FALSE`.
- quietTRUE or FALSE. If FALSE, errors or bad input result in more warning messages. Default is `quiet=TRUE`.

Author(s)

Leif Johnson <leif.t.johnson@gmail.com>

Maintainer: Leif Johnson <leif.t.johnson@gmail.com>

References

<http://www.stat.umn.edu/~leif/software/grade>

grade.discreteprobability

Grade Discrete Probability Sets

Description

Checks a students probability distribution, makes sure that (1) It sums to 1 (2) All elements are ≥ 0

Optionally, it compares the students to a correct one. Order is optionally enforced.

Usage

```
grade.discreteprobability(correctans, studentans, tolerance=.01,
                           useeval=TRUE, usena=FALSE, useinf=FALSE,
                           quiet=TRUE, ordered=FALSE, checkcorrect=TRUE)
```

Arguments

correctans	a vector of type numeric or a string
studentans	a vector of type numeric or a string
tolerance	a string or numeric representing the accepted component wise tolerance

useeval	TRUE or FALSE indicates whether or not to use 'eval' on strings
usena	TRUE or FALSE indicating whether or not NA is an accepted value
useinf	TRUE or FALSE indicating whether or not Inf and -Inf are accepted values
quiet	TRUE or FALSE. If TRUE there are more warning messages when checks fail. Can be helpful for debugging.
ordered	TRUE or FALSE. If TRUE studentans order must match correctans order to be considered correct. If FALSE, order does not matter (so both are sorted and then checked)
checkcorrect	TRUE or FALSE. if TRUE studentans needs to match correctans. If FALSE studentans only needs to qualify as a discrete probability distribution.

Details

If checkcorrect=FALSE, grade.discreteprobability does not do any checks on correctans. In this case to be correct, studentans needs to satisfy discrete probability constraints – all elements ≥ 0 and sums to 1.

If checkcorrect=TRUE discrete probability constraints are enforced on correctans. studentans needs to match correctans in this case. Order is only enforced if ordered=TRUE.

grade.discreteprobability does not use NA. If usena=TRUE grade.discreteprobability sets it to FALSE and issues a warning message.

Value

TRUE or FALSE indicating match success or failure respectively. FALSE is also returned if studentans does not look like a set.

Note

The [grade](#) main page contains a discussion of the common parameters correctans, studentans, useeval, usena, useinf, quiet.

See Also

[grade grade.set](#)

Examples

```
# TRUE
grade.discreteprobability(c(1/2,1/2), "[.5, .5]")
# TRUE
grade.discreteprobability(NULL, "[0, .33, .17, .5]", checkcorrect=FALSE)

# FALSE
grade.discreteprobability(NULL, "[-1, 0, 0, 1, 1]", checkcorrect=FALSE)

# TRUE
grade.discreteprobability(c(0, 1/2, 1/4, 1/4), "[0, 1/2, 1/4, 1/4]")
# FALSE
```

```

grade.discreteprobability(c(0, 1/2, 1/4, 1/4), "[0, .25, .25, .5]",
                          ordered=TRUE)

# TRUE
grade.discreteprobability(c(0, 1/2, 1/4, 1/4), "[0, .5, .25, .25]",
                          ordered=TRUE)

```

grade.interval

Grade Intervals

Description

Checks a students interval against a correct one.

Usage

```

grade.interval(correctans, studentans, tolerance=0.01, useeval=TRUE,
               usena=FALSE, useinf=FALSE, quiet=TRUE)

```

Arguments

correctans	a vector of type numeric or a string
studentans	a vector of type numeric or a string
tolerance	a string or numeric representing the accepted component wise tolerance
useeval	TRUE or FALSE indicates whether or not to use 'eval' on strings
usena	usena is ignored in grade.interval. Setting to TRUE results in a warning message.
useinf	TRUE or FALSE indicating whether or not Inf and -Inf are accepted values
quiet	TRUE or FALSE. If TRUE there are more warning messages when checks fail. Can be helpful for debugging.

Details

usena is ignored in this function. If set to true, grade.interval sets it back to false and produces a warning message. grade.interval expects correctans to be a vector of length 2, if not it errors out. If correctans is in reverse order and quiet=FALSE, grade.interval issues a warning, but continues grading.

Value

TRUE or FALSE indicating match success or failure respectively. FALSE is also returned if studentans does not look like an interval.

Note

The [grade](#) main page contains a discussion of the common parameters correctans, studentans, useeval, usena, useinf, quiet.

See Also

[grade](#) [grade.set](#) [grade.number](#)

Examples

```
grade.interval(c(1,2), "[1,2]") # TRUE
grade.interval(c(1,2), "[1.1,2]", tolerance=".01") # FALSE

grade.interval(c(1,pi), "(1,3.142)", tolerance=".001") # TRUE
```

grade.negative	<i>Check the Sign of a Number</i>
----------------	-----------------------------------

Description

Sees if studentans is negative, correctans is ignored.

Usage

```
grade.negative(correctans=NULL, studentans, tolerance=0.01,
               useeval=TRUE, usena=FALSE, useinf=FALSE, quiet=TRUE)
```

Arguments

correctans	not used in this function, no restrictions are enforced.
studentans	a vector of type numeric or a string
tolerance	a string or numeric representing the accepted component wise tolerance
useeval	TRUE or FALSE indicates whether or not to use 'eval' on strings
usena	TRUE or FALSE indicating whether or not NA is an accepted value
useinf	TRUE or FALSE indicating whether or not Inf and -Inf are accepted values
quiet	TRUE or FALSE. If TRUE there are more warning messages when checks fail. Can be helpful for debugging.

Value

TRUE if (studentans < -tolerance) FALSE otherwise.

Note

The [grade](#) main page contains a discussion of the common parameters correctans, studentans, useeval, usena, useinf, quiet.

See Also

[grade](#) [grade.set](#) [grade.number](#)

Examples

```
grade.negative(studentans=0, "1") # FALSE
grade.negative(NULL, "1.1", tolerance=".01") # FALSE

grade.negative("soup", "-.1", tolerance=.05) # TRUE
```

grade.number

Grade Single Numbers

Description

Checks studentans against correctans. For scalars only.

Usage

```
grade.number(correctans, studentans, tolerance=0.01,
             useeval=TRUE, usena=FALSE, useinf=FALSE, quiet=TRUE)
```

Arguments

correctans	a vector of type numeric or a string
studentans	a vector of type numeric or a string
tolerance	a string or numeric representing the accepted component wise tolerance
useeval	TRUE or FALSE indicates whether or not to use 'eval' on strings
usena	TRUE or FALSE indicating whether or not NA is an accepted value
useinf	TRUE or FALSE indicating whether or not Inf and -Inf are accepted values
quiet	TRUE or FALSE. If TRUE there are more warning messages when checks fail. Can be helpful for debugging.

Value

TRUE if studentans is within tolerance of correctans. FALSE otherwise.

Note

The [grade](#) main page contains a discussion of the common parameters correctans, studentans, useeval, usena, useinf, quiet.

See Also

[grade](#) [grade.set](#) [grade.negative](#)

Examples

```
grade.number(1, "1") # TRUE
grade.number(1, "1.1", tolerance=".01") # FALSE

grade.number(pi, "3.142", tolerance=".001") # TRUE

grade.number(1, "[1]") # TRUE
```

grade.parse

Parse Input

Description

Parse input, returning either NULL or a vector of the values.

Usage

```
grade.isscalar(x, usena=FALSE, useinf=FALSE, quiet=TRUE)

grade.parse(ans, useeval=TRUE, usena=FALSE, useinf=FALSE, quiet=TRUE)
grade.parseset(ans, useeval=TRUE, usena=FALSE, useinf=FALSE, quiet=TRUE)
grade.parsechunk(ans, useeval=TRUE, usena=FALSE, useinf=FALSE, quiet=TRUE)
```

Arguments

x	argument for grade.isscalar to check
ans	input to parse. Can be a string or a vector
useeval	TRUE or FALSE indicates whether or not to use 'eval' on strings
usena	TRUE or FALSE indicating whether or not NA is an accepted value
useinf	TRUE or FALSE indicating whether or not Inf and -Inf are accepted values
quiet	TRUE or FALSE. If TRUE there are more warning messages when checks fail. Can be helpful for debugging.

Details

grade.isscalar checks to see if x is a finite numeric scalar (vector of length 1). If usena=TRUE, NA is also accepted. If useinf=TRUE, Inf and -Inf are also accepted.

Input to the grade.parse functions can be a string or a vector. grade.parsechunk will only return scalars, the other two will return a vector. All three check return values using grade.isscalar on each element.

grade.parse delegates character types to either grade.parsechunk or grade.parseset. If the string contains any of the characters '[', ']', '(', ')', or ',', the string is sent to grade.parseset. Otherwise it is sent to grade.parsechunk.

If x is a character, grade.parsechunk checks for any of the forbidden characters '[', ']', '(', ')', or ','. If any are found grade.parsechunk refuses to evaluate the string.

If `x` is a character, `grade.parsechunk` makes sure that it *looks* like a vector or set. I.e. it starts with an open bracket or parenthesis and ends with a close bracket or parenthesis. No other brackets or parenthesis are allowed. The middle is expected to be a comma delimited list. See the examples for more clarification.

If `useeval=TRUE`, text elements are evaluated using `eval`. If `useeval=FALSE` text elements are coerced using `as.numeric`. `eval` is more forgiving to input, i.e. `eval` of text input `'1/2'` returns `.25`, but `as.numeric` of text `'1/2'` returns `NA`. However, `eval` does leave an opening for unchecked code to be run in R. Text containing parenthesis or brackets is not put into either `eval` or `as.numeric` by the `grade.parse` functions, but there is still a risk. If you are concerned, set `useeval=FALSE`.

Value

`grade.parse` and `grade.parseset` returns either a vector of the values, or `NULL` if the input was not valid.

`grade.parsechunk` returns either a single value, or `NULL` if the input was not valid.

`grade.isscalar` returns `TRUE` if `x` is a scalar (vector of length 1), `FALSE` otherwise.

Note

The [grade](#) main page contains a discussion of the common parameters `correctans`, `studentans`, `useeval`, `usena`, `useinf`, `quiet`.

See Also

[grade](#) [grade.set](#) [grade.number](#)

Examples

```
grade.parse("[1, 2, 3]") # returns c(1,2,3)
grade.parse("[NA, 1, 2]") # returns NULL
grade.parse("[NA, 1, 2]", usena=TRUE) # returns c(NA, 1, 2)
grade.parse("[pi]") # returns 3.141...
grade.parse("[pi]", useeval=FALSE) # returns NULL

grade.parsechunk("1") # 1
grade.parsechunk(",1") # NULL
grade.parsechunk("[1]", quiet=FALSE) # NULL, with error message

grade.parseset("[1,2,3]") # c(1,2,3)

grade.isscalar(1) # TRUE
grade.isscalar(c(1,2)) # FALSE
grade.isscalar(NA) # FALSE
grade.isscalar(NA, usena=TRUE) # TRUE
grade.isscalar(Inf) # FALSE
grade.isscalar(Inf, useinf=TRUE) # TRUE
```

grade.set

*Grade Sets***Description**

Checks a the set (vector in R) studentans against correctans. grade.orderedset enforces order, grade.set does not.

Usage

```
grade.set(correctans, studentans, tolerance=0.01, useeval=TRUE,
          usena=FALSE, useinf=FALSE, quiet=TRUE)
grade.orderedset(correctans, studentans, tolerance=0.01, useeval=TRUE,
                  usena=FALSE, useinf=FALSE, quiet=TRUE)
```

Arguments

correctans	a vector of type numeric or a string
studentans	a vector of type numeric or a string
tolerance	a string or numeric representing the accepted component wise tolerance
useeval	TRUE or FALSE indicates whether or not to use 'eval' on strings
usena	TRUE or FALSE indicating whether or not NA is an accepted value
useinf	TRUE or FALSE indicating whether or not Inf and -Inf are accepted values
quiet	TRUE or FALSE. If TRUE there are more warning messages when checks fail. Can be helpful for debugging.

Value

TRUE if the sets match. FALSE otherwise.

Note

The [grade](#) main page contains a discussion of the common parameters correctans, studentans, useeval, usena, useinf, quiet.

See Also

[grade](#) [grade.number](#)

Examples

```
grade.set(c(1,2), "[1,2]") # TRUE
grade.orderedset(c(1,2), "[1,2]") # TRUE

grade.set(c(2,1), "[1,2]") # TRUE
grade.orderedset(c(2,1), "[1,2]") # FALSE
```

```

grade.set(c(1,2), "[1.1,2]", tolerance=".01") # FALSE

grade.set(c(1,2,3,4,5), "(5,4,3,2,1)") # TRUE
grade.set(c(1,2,3,4,5), "(5,4,3,2)") # FALSE

grade.orderedset("[NA, 1, 2]", c(NA, 1, 2)) #FALSE, usena=F
grade.orderedset("[NA, 1, 2]", c(NA, 1, 2), quiet=FALSE) # FALSE, but with warning
grade.orderedset("[NA, 1, 2]", c(NA, 1, 2), usena=TRUE) # TRUE

```

grade.truefalse	<i>Grade True False Answers</i>
-----------------	---------------------------------

Description

Checks studentans against correctans. For true/false answers only.

Usage

```

grade.truefalse(correctans, studentans, tolerance=0.01,
                useeval=TRUE, usena=FALSE, useinf=FALSE, quiet=TRUE)

```

Arguments

correctans	TRUE or FALSE or a string
studentans	TRUE or FALSE or a string
tolerance	a string or numeric representing the accepted component wise tolerance
useeval	TRUE or FALSE indicates whether or not to use 'eval' on strings
usena	TRUE or FALSE indicating whether or not NA is an accepted value
useinf	TRUE or FALSE indicating whether or not Inf and -Inf are accepted values
quiet	TRUE or FALSE. If TRUE there are more warning messages when checks fail. Can be helpful for debugging.

Value

TRUE if studentans==correctans AND both studentans and correctans are TRUE or FALSE. FALSE otherwise.

Note

The [grade](#) main page contains a discussion of the common parameters correctans, studentans, useeval, usena, useinf, quiet. grade.truefalse does not accept usena or useinf. Setting usena=TRUE or useinf=TRUE will result in a warning. tolerance is not used in grade.truefalse. These arguments are included for compatibility with the other function calls in grade.

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

```
grade.truefalse(TRUE, TRUE) # TRUE
grade.truefalse(TRUE, "TRUE") # TRUE
grade.truefalse("FALSE", "TRUE") # FALSE
## depending on your environment settings, this next example may work.
#grade.truefalse("F", F) # TRUE if your environment has not redefined 'F'
```

Index

* **misc**

- grade.discreteprobability, [3](#)
- grade.interval, [5](#)
- grade.negative, [6](#)
- grade.number, [7](#)
- grade.parse, [8](#)
- grade.set, [10](#)
- grade.truefalse, [11](#)

* **package**

- grade-package, [2](#)

as.numeric, [3](#), [9](#)

eval, [3](#), [9](#)

grade, [4-7](#), [9-12](#)

grade (grade-package), [2](#)

grade-package, [2](#)

grade.discreteprobability, [3](#)

grade.interval, [5](#)

grade.isscalar (grade.parse), [8](#)

grade.negative, [6](#), [7](#)

grade.number, [6](#), [7](#), [9](#), [10](#)

grade.orderedset (grade.set), [10](#)

grade.parse, [8](#)

grade.parsechunk (grade.parse), [8](#)

grade.parseset (grade.parse), [8](#)

grade.set, [4](#), [6](#), [7](#), [9](#), [10](#)

grade.truefalse, [11](#)