

Package: gradeR (via r-universe)

October 18, 2026

Title Helps Grade Assignment Submissions in Common R Formats

Version 2.0.1

Description After being given the location of your students' submissions and a test file, the function runs each file that is an R script, R Markdown file, or Quarto document, and evaluates the results from all the given tests. Results are neatly returned in a data frame that has a row for each student, and a column for each test.

Depends R (>= 3.4)

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.3

Imports testthat, methods, callr, jsonlite

Suggests knitr, rmarkdown

VignetteBuilder knitr

Config/pak/sysreqs cmake make libuv1-dev

Repository <https://tbrown122387.r-universe.dev>

Date/Publication 2026-08-19 20:56:25 UTC

RemoteUrl <https://github.com/tbrown122387/grader>

RemoteRef HEAD

RemoteSha 0176e51359b836dc334fab81433d45d20b912dd5

Contents

calcGrades	2
calcGradesForGradescope	3
findBadEncodingFiles	4
findGlobalPaths	5
getTestScriptReport	5

Index	7
--------------	----------

calcGrades	<i>The grading function.</i>
------------	------------------------------

Description

This function grades a bunch of R script assignments

Usage

```
calcGrades(  
  submission_dir,  
  your_test_file,  
  suppress_warnings = TRUE,  
  verbose = FALSE  
)
```

Arguments

`submission_dir` where the assignments are located

`your_test_file` the path to your testthat test file (e.g. `grade_hw1.R`)

`suppress_warnings`
warning handlers prevent code from being run after they catch something. Suppress this behavior by setting this argument to `TRUE`.

`verbose` set to true if you want to print the name of the file as it's being ran

Value

a data.frame of all the scores for each student

Examples

```
# change paths to *your* paths  
submissions <- "extdata/example/assignment1_submissions/"  
my_test_file <- system.file("extdata/example", "grade_hw1.R", package = "gradeR")  
results <- calcGrades(submissions, my_test_file)
```

`calcGradesForGradescope`*The grading function for Gradescope.*

Description

This function grades one R script assignment submission and writes results out to a properly-formatted json file for Gradescope. Supports R scripts (.r, .R) as well as R Markdown (.Rmd) and Quarto (.qmd) documents.

Usage

```
calcGradesForGradescope(  
  submission_file,  
  test_file,  
  which_results = "gradescope",  
  suppress_warnings = TRUE  
)
```

Arguments

<code>submission_file</code>	the path to the assignment submission file (e.g. "hw1.r", "hw1.Rmd", or "hw1.qmd"). For Rmd/Qmd files, R code will be automatically extracted.
<code>test_file</code>	the path to the .r file with test_that tests (e.g. "hw1_tests.R")
<code>which_results</code>	Choose either "testing" or "gradescope". If equal to "gradescope", the json file is written to /autograder/results/results.json. Otherwise, results.json is written to your current working directory.
<code>suppress_warnings</code>	If FALSE, warnings are fatal; if set to TRUE, warnings will not prematurely terminate running of student submission scripts.

Details

Each `test_that()` block in `test_file` becomes one row in the Gradescope results table. Two things about a `test_that()` block are controlled by strings embedded in its **name** (the first argument to `test_that()`):

- **Visibility**, via (visible), (hidden), (after_due_date), or (after_published) – see [getTestVisibility](#). If none is present, the test defaults to "after_due_date".

Point values, on the other hand, are *not* read from the `test_that()` name. Instead, each individual `expect_*()` call inside the block is its own "criterion" worth 1 point by default. To customize a criterion's point value (and its message), pass `label = "your message [Npts]"` to that `expect_*()` call – see [extractPointValue](#). A test's max_score on Gradescope is the sum of its criteria's point values, so a block with three `expect_*()` calls worth [1pts], [2pts], and unlabeled (1 pt) is worth 4 points total. For example:

```
test_that("p1a (visible)", {
  expect_true(is.numeric(p1a), label = "p1a is numeric [1pts]")
  expect_equal(p1a, 42, label = "p1a has the correct value [3pts]")
})
```

Before uploading a test file to Gradescope, use [getTestScriptReport](#) on it to see the total point value across all tests – that total is what you should enter as the assignment’s max score on Gradescope.

Value

Invisibly returns NULL. The function’s primary purpose is the side effect of writing a JSON results file.

See Also

[getTestScriptReport](#) to preview a test file’s tests and point totals before uploading it to Gradescope.

Examples

```
## Not run:
# For local testing
calcGradesForGradescope("student_hw1.r", "hw1_tests.R", which_results = "testing")

# For Gradescope autograder (inside Gradescope environment)
# calcGradesForGradescope("hw1.r", "hw1_tests.R", which_results = "gradescope")

# Works with R Markdown files too
# calcGradesForGradescope("student_hw1.Rmd", "hw1_tests.R", which_results = "testing")

## End(Not run)
```

findBadEncodingFiles *This function finds unreadable files.*

Description

A function that finds student submissions with poorly encoded characters

Usage

```
findBadEncodingFiles(submission_dir)
```

Arguments

submission_dir where the assignments are located

Examples

```
# change paths to *your* paths
submissions <- "extdata/assignment1_submissions/"
findBadEncodingFiles(submissions) # perhaps ask these students to resubmit
```

findGlobalPaths	<i>This function finds files with global file paths.</i>
-----------------	--

Description

A function that finds student submissions that refer to machine-specific file paths

Usage

```
findGlobalPaths(submission_dir)
```

Arguments

submission_dir where the assignments are located

Examples

```
# change paths to *your* paths
submissions <- "extdata/assignment1_submissions/"
findGlobalPaths(submissions) # perhaps ask these students to resubmit
```

getTestScriptReport	<i>The function for analyzing and summarizing R test scripts.</i>
---------------------	---

Description

This function scans a given test script and summarizes the number of tests and test criteria in the script, as well as point values. Points are read the same way [calcGradesForGradescope](#) reads them: from label = "... [Npts]" arguments on individual expect_*() calls, defaulting to 1 point per expect_*() call when no [Npts]/(Npts) tag is present. Run this on a test file *before* uploading it to Gradescope so you know what total max score to enter for the assignment (the "Total points" line of the printed report).

Usage

```
getTestScriptReport(script_path)
```

Arguments

script_path the name of the .r file containing tests tests (e.g. "hw1_tests.R")

Value

Invisibly returns a list with components summary (total tests, criteria, and points), criteria_per_test, expect_function_counts, and test_details. Called primarily for its printed report, produced via [getPrettyReport](#).

Examples

```
my_test_file <- system.file("extdata/example", "grade_hw1.R", package = "gradeR")
getTestScriptReport(my_test_file)
```

Index

- * **Gradescope**
 - calcGradesForGradescope, [3](#)
- * **calcGradesForGradescope**
 - calcGradesForGradescope, [3](#)
- * **calcGrades**
 - calcGrades, [2](#)
 - findBadEncodingFiles, [4](#)
 - findGlobalPaths, [5](#)
- * **findBadEncodingFiles**
 - findBadEncodingFiles, [4](#)
- * **findGlobalPaths**
 - findGlobalPaths, [5](#)
- * **getPrettyReport**
 - getTestScriptReport, [5](#)
- * **getTestScriptReport**
 - getTestScriptReport, [5](#)

calcGrades, [2](#)
calcGradesForGradescope, [3](#), [5](#)

extractPointValue, [3](#)

findBadEncodingFiles, [4](#)
findGlobalPaths, [5](#)

getPrettyReport, [6](#)
getTestScriptReport, [4](#), [5](#)
getTestVisibility, [3](#)