Pedagogy and Technology: Increasing Accuracy and Efficiency in Grade Calculations

Nicholas R. Thompson

This paper was completed and submitted in partial fulfilment of the Master Teacher Program, a 2-year faculty professional development program conducted by the Center for Faculty Excellence, United States Military Academy, West Point, N Y, 2019.

Abstract.

This paper provides educators and evaluators with a viable grade calculator web application open to the public free of use. The calculator, called the *Multi-Dimensional Grading Calculator* (MDGC) computes the points earned, a letter grade for the assignment, and a grade percentage. The computations are weighted at *40%* for the content portion of a written assignment; *20%* for the organization and/or citations in an assignment, and *40%* for the writing and grammar of the assignment. The calculator returns accurate computations based upon the instructor's assessment of the different dimensions based upon an estimated letter grade. The MDGC can be accessed from any web browser.

Introduction.

Teaching would be easy, except for having to grade. Grading written work is especially taxing since the grader must learn what the writer is trying to convey. In well-written papers, grading is reasonably easy and straightforward because the material is as close to everyday published material. Such closeness brings a sense of relief to the teacher: all of the instruction is not in vain! Poorly written papers, on the other hand, require strict focus in the areas indicated in the class syllabus that are most pertinent to the individual class' content and writing style. This capstone provides a usable writing grading calculator that eliminates the need to build excel or google sheet formula calculators.

As an instructor at the United States Military Academy (USMA) from 2017-2019, the author discovered the importance of grading papers in a timely and efficient manner. How can professors, teachers, and instructors reduce the time spent calculating grades for written work? Heuristically speaking, grading can run the gamut from full credit for completing a written assignment to meticulous copy and content editing.

Somewhere in between the two extremes lies an optimum level of time spent grading, time spent writing feedback (Heen and Stone 2014), and time spent capturing the grade for assessment against student peers and administrative record keeping for the institution. This capstone project addresses the last item in the list: finding efficiency in calculating written assignments based upon a standardized rubric. Time, being a scarce resource, is better spent developing quality prompts and content for students, writing quality feedback, and developing quality lessons. In the writing process alone, teachers and instructors can lose hours and possibly days over a single the semester calculating grades for hundreds of papers. Hours may not seem like much, but startup costs often turn minutes into hours and scaling up; hours into days. Compounding the scarcity of time is the broken nature of grading throughout weekends and holidays to ensure feedback does not trickle into weeks.

A typical writing assignment includes instruction to guide the students in the general direction to research and answer the prompt, a well-developed prompt, time to write the assignment, a deadline and venue for submitting the assignment, grading, feedback, and recording the grade (Fink 2003). While the entire process is of great import, developing a pedagogical course for the entire process remains outside the scope of this work; indeed others have already finished comprehensive works on the matter (Fink 2003). For this capstone, the author will focus on content, organization and citation accuracy, and writing style and tone as the three general categories for grading papers. The author includes Appendix I as a reference for the reader to understand the qualitative nature of the grading rubric and how it interacts with the qualitative numbers assigned for a 100-point assignment.

The main argument of this capstone that justifies the development of a web app that can quickly and accurately calculate grades for multi-dimensional rubrics with varying assigned points. Additionally, teachers and instructors must evaluate composition across many different writing levels, throughout multiple assignments, and over multiple semesters. This capstone argues such variation is ripe for inadequate, untimely, and overly subjective feedback that is both inefficient for the teacher and unproductive for the student.

The remainder of the study proceeds with a brief literature review. The third section provides a detailed explanation of the HTML web application developed in this project. The final section briefly discusses some insights gained while developing this web application.

Literature Review.

Accurately capturing the relevant literature requires spanning two disciplines. Pedagogy is ultimately the driving force for web application development, which is this project's main contribution. The literature on pedagogy is vast and deep. Therefore, this capstone project will focus solely on those pedological elements necessary to support the subjective interpretation of writing and translating that subjective assessment into a repeatable process. The interpretation should also provide a substantive feedback mechanism for the adept student willing to reference both the teachers grading notes and the class rubric on writing.

Pedagogy: Why writing matters.

On average, third and fourth-year college students at the United States Military Academy fall at a high level of *stage-two* (Grow 1991). The average student takes a personal stake in the syllabus, but might only reference it periodically throughout the semester. Those who gravitate towards *stage-three* tend to see themselves as participants in what they learn, why, and particularly how they are learning the content (Grow 1991).

Writing demands retrieval (Brown, Roediger, and McDaniel 2014), critical thinking (Willingham 2007), self-motivation (Grow 1991; Brophy 2013), overcoming a fear of failure (Brown, Roediger, and McDaniel 2014), and helps students to "avoid the illusion of knowing" (Brown, Roediger, and McDaniel 2014, 102). Writing provides a venue for students to identify their learning styles, explore subjects in which they find interest, and become active in their learning (Brown, Roediger, and McDaniel 2014). Perhaps the most important effect, however, that writing has on an individual is to increase their ability in a dimension they would otherwise never explore, and with the help of an individual dedicated to helping them become better by training their brain in new and expanding ways (Brown, Roediger, and McDaniel 2014, 176-178).

Pedagogy: Why rubrics and grade matter.

Scoring rubrics provide a guide to assessing and understanding the processes of student's efforts (Brookhart 1999; Moskal 2000). When a wide array of evaluators must assess the same writing prompt—such as college entrance essays or evaluative tests like the Graduate Requisite Exam (GRE)—rubrics are essential (Moskal 2000). However, rubrics also have value when a single instructor must evaluate a wide array of knowledge and writing skills across classes and time.

This capstone utilizes a rubric loosely formulated on Leydens and Thompson (1997), which is provided in Appendix I. The use of rubrics is well supported by the literature. Rubrics help to evaluate group activities, extended projects, and oral presentations, and their applications span both primary and secondary education (Moskal 2000). The author has found especially important the use of numerical weights to sub-skills within the writing process. Moskal highlights the value of weighting sub-skills, "A student who receives a '70' out of '100', may not know how to improve his or her performance on the next assignment. Scoring rubrics respond to this concern by providing descriptions at each level as to what is expected." (Moskal 2000, 2) These sub-skill weights help the student to understand where they must improve to make their writing, paper organization, and content better on the next assignment.

In short, rubrics provide a base of understanding for the students and a mechanism of standardization for the teacher or instructor. Assessment in the classroom is paramount (Brookhart 1999) and provides the teacher or instructor with valuable feedback to understand what the students have learned. Feedback allows the teacher to modify the content and provides the student with a consistent information return on the work they have put into a writing assignment.

One of the prime challenges to efficient and accurate grading derives from a phenomenon articulated by Brookhart (1999): grade inflation results when teachers must combine weighted scores into an individual score. Grade inflation is particularly alluring when a cursory review of the graded assignment does not present any glaring problems. Finally, a rubric allows teachers to fall back on their assessments with a degree of confidence that a challenge by the student is backed up by a published document outlining the expectations for writing.

Grades serve an important purpose in the pedagogical process by providing teachers with feedback for self-assessment (Airasian and Gullickson 1994). According to Airasian and Gullickson, teacher self-assessment has seven characteristics: (1) the teacher has knowledge and control over the self-assessment process; (2) internal conditions, such as dissonance in the learning environment and personal curiosity, exist; (3) the teacher sets personal or problematically motivated self-assessment; (4) experience and novice teachers have different self-assessment activities when compared with one another; (5) teachers must assume responsibility if meaningful change is to occur; (6) the teacher will recall pertinent information and integrate it according to the needs of the students; and (7) the product of self-assessment is conscious (Airasian and Gullickson 1994, 196). There is also a practical application to grading. Teachers use grades for administrative purposes, provide students feedback, provide guidance to students, provide feedback to teachers for instructional planning, and to motivate students (Peter 1994).

Web Application Development.

As teachers it is not uncommon to have four of five writing assignments for student each semester. Some are smaller than others. For example, the first two writing assignments in the DS360, Special Operations / Low-intensity Conflict course are less than one page each. However, students are challenged with writing a high quality and succinct product. The writing must include appropriate citation format from the Chicago Author-Date style. Although the first two assignments are less than one single-spaced page each, students routinely make simple and fundamental mistakes that are clearly articulated in the writing prompt, writing guide, and writing rubric.

If all students were able to write well, then grading would be easy. However, there is a great degree of variation in their abilities. Consequently, providing timely and accurate feedback necessitates the use of a writing rubric to assign a percentage of points weighted by category. In the case of the DS360 course, weighting content and writing ability provide the teacher with a way of emphasizing components that aid the students in preparing a well written final document. On the one hand, the rubric provides a strong foundation; however, on the other hand the weighting of three categories splitting one-hundred percent of the points for a given assignment demands meticulous calculations for each student. Furthermore, people seem more adept at saying in their mind, “the content is worth an A, but the writing seems more like a B.” The intersection of thinking in letter grades but needing to assign a points value is the driving force

for building a wed application that makes calculating written assignment grades quick, efficient, and accurate—every time.

Technology: How web applications increase accuracy and save time.

Any teacher at the collegiate level has the skill to calculate weighted categories. For the more technologically adept spreadsheets provide a bountiful playground for formula building to automate the weighting of the categories. In both cases, human error can play a role. Sometimes spreadsheet cells become corrupted or weight change, necessitating the re-writing of the code.

Using a web application, such as the one developed for this capstone, meets the needs of accuracy by allowing the teacher to input the assigned points. In the current version of the web application the weights are restricted to forty percent for content, twenty percent for organization and citations, and forty percent for writing. Future versions will include the ability to change weights. After inputting the assignment points, teachers input their assessment of the content, organization and citations, and writing in subsequent drop-down boxes that range from *F* at the low end to *A++* at the high end. The author included *A+* and *A++* because sometimes students achieve a 99% and sometimes there is the exceptional work that deserves the highest mark and a 97%, 98%, or 99% are just not quite the same. Web applications also save time by reducing the amount of calculations per paper and the coding time spent on spreadsheets.

Developing the web application.

The author built the web application in R, version 3.5.1 (2018-07-02) -- "Feather Spray", Copyright (C) 2018 The R Foundation for Statistical Computing, Platform: x86_64-apple-darwin15.6.0 (64-bit). The functional code in the app.R runs on the *shinyapps.io* server, and the following hyperlink provides access for anyone to use in dependent of other's calculations.

Hyperlink to the Multi-Dimensional Grading Calculator:	https://rshinywebapps42.shinyapps.io/gradecalculatorwebapp/

The code for the application provides the user with three inputs that are depicted in Figure 1 as *A++, A++,* and *A++* for the content, organization, and writing, respectively. One would expect the *gradePct* column to be 1, which depicts 100 *PointsEarned* earned out of a possible 100 *TotalPoints.*

Figure 2 demonstrates the output from a seventy-five-point assignment where the student received a *B-* for content, an *A-* for organization, and a *C-* for writing. The output is intuitive; the general feeling of the paper was not one of high-quality, but for the sake of fairness—often personal irritation with the student—use of the rubric and calculator provides a fair and repeatable grade. In this case the student earned a *C+* letter grade totaling *58.5* points. Using the web application, challenges to the grade are standardized and the calculations of the teacher can be referred to the open source code in the web application.

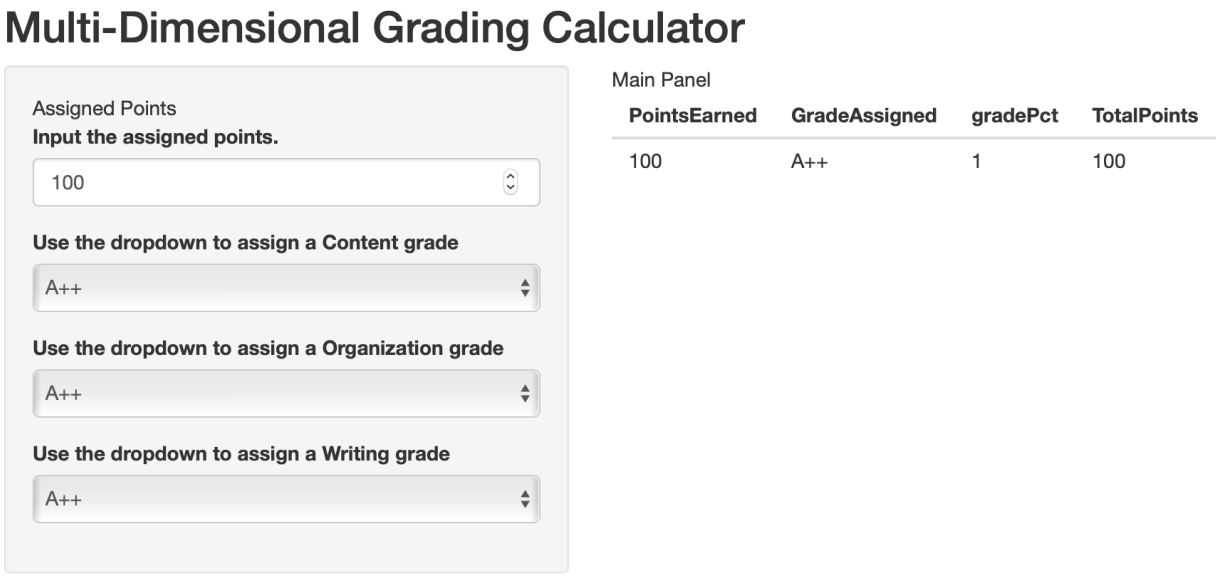
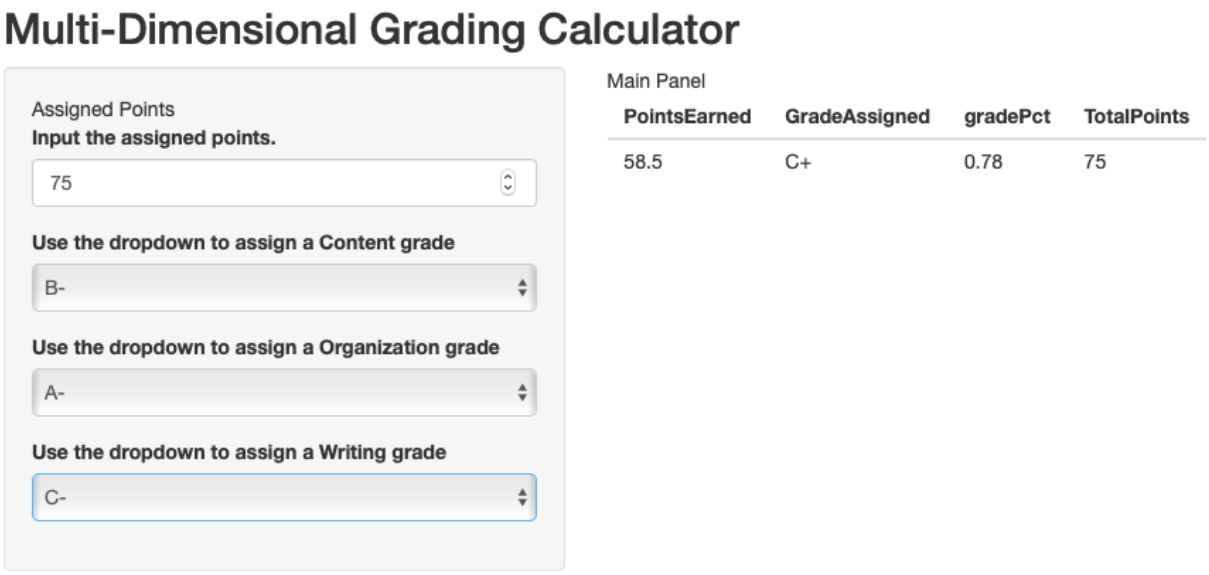


Figure 1: The web interface for the Multi-Dimensional Grading Calculator. Along the left sidebar, users can input their assignment's points value. Immediately below, evaluators assign the letter grade for the content, organization, and writing respectively. The right main panel provides the output of the calculations run in the background by the programmatic script.



The author of this capstone project wishes for full transparency, believing the code should be open for all to evaluate on their own. Published in Annex II, the code is also available under the file name: *20190415_thompson_GradeCalculatorapp_FINAL.R*

The Shiny app (Chang et al. 2018) is part of the broader R statistics (R Core Team 2018) computational platform and for this capstone the author used the RStudio (RStudio, Inc 2013) development IDR is developed as an open-source system. As part of the application, the author wrote a separate function called *calcGrade()* that is also included in the GitHub Repository: https://github.com/NickThompson42/Pedagogy

Conclusion.

This capstone provides teachers, instructors, and evaluators with a unique tool designed for their specific purposes. Future developmental needs and requirements should be forwarded to the author at nick.thompson3853@gmail.com. This project was two years in the making. From the first master trainer program class in the fall of 2017 to the spring of 2019 the author learned new and important lessons in the development of a quality class. The culmination of this online web application is both a personal best with a number of firsts. The *calcGrade()* function is the first function the author has ever written in R. The *Multi-dimensional Grading Calculator* © is the first shiny app the author has written. Finally, uploading this application for the world to use is a purely unique and unexpected turn of events in finishing this project.

Rubrics are an absolute necessity for teachers that wish to remain fair and accurate over the course of years. Additionally, without a rubric that is tied to a specific calculation, teachers are unable to make adjustments to their students learning needs. The author invites all teachers to try out the *Multi-dimensional Grading Calculator* © and implement rubrics into their teaching methodology.

Appendix I		Criteria					
	Levels	Content & Analysis		Organization & Citations		Writing	
Elements	Novice	F (26 points)	Thesis is essentially missing or indiscernible. Ideas are extremely simplistic, show signs of confusion, or demonstrate a fundamental misunderstanding of the assignment and course concepts.	F (13)	Introduction, thesis and conclusion not present or have no logical connection to the main points. Organization, if evident at all, is confusing and disjointed. Paragraph structure is weak and transitions are missing or illogical. Citations do not exist or are in incorrect style.	F (26)	Simple sentences used almost exclusively or overly complex sentences that inhibit all meaning. Pervasive sentence structure and mechanical errors. Possesses no awareness of a particular audience and tone is inappropriate or very inconsistent.
		D (27)		D (13.5)		D (27)	
	Developing	C- (29)	Thesis and ideas are noticeably simplistic and unfocused, with limited sense of purpose or control of the argument. Evidence provides insufficient, non- specific or irrelevant support for main ideas. Argument demonstrates limited knowledge of course concepts.	C- (14)	Introduction and conclusion provide poor guides for the reader regarding purpose and thesis. Organization is generally unsuccessful as paragraphs are simple, disconnected and/or formulaic. Lacks all transitions or planned sequences. Citations are in the correct style, but in-text citations are improper. References do not match in-text citations.	C- (29)	Writing is unclear and inefficient due to poor sentence construction and regular mechanical errors. Possesses little awareness of a particular audience and lacks an appropriate tone.
		C (30)		C (15)		C (30)	
		C+ (31)		C+ (15.5)		C+ (31)	
		B- (32)	Capable thesis, but borders on being weak, unclear or too broad. Main ideas are loosely related to one another and the thesis. They are mostly intelligible, but evidence is sometimes insufficient, unspecific, or indirectly supports the argument. Course concepts are present and capably integrated into the argument.	B- (16)	Introduction and conclusion somewhat guide the reader, but lack clarity in purpose. Limited attempts to organize around thesis, with mostly stand-alone paragraphs that lack weak or non-evident transitions. Citations are formatted properly in-text, but are improperly formatted in references or vice-versa.	B- (32)	While writing is satisfactory, it is occasionally unclear and inefficient, and possesses noticeable mechanical errors. Sentence construction lacks desired variety. Possesses unrefined tone and awareness of intended audience.
		B (34)		B (17)		B (34)	
		B+ (35)		B+ (17.5)		B+ (35)	
		A- (36)	Competent and well developed thesis, representing sound and adequate evaluation of the topic. Main ideas are adequately related to each other and thesis. Ideas are supported with sufficient evidence that is sound, valid and logically ordered. Course concepts are well integrated into the argument.	A- (18)	Introduction and conclusion provide clear guides for the reader. Competent organization, but with limited sophistication. Competent paragraph structure, with suitable transitions. Citations are correctly formatted in-text and references.	A- (36)	Mostly clear and efficient writing that is generally free of mechanical errors. Good use of strong, positive, active language with well-constructed sentences. Possesses suitable tone and awareness of intended audience.
		A (38)		A (19)		A (38)	
		A+ (39)	Clear and fully developed thesis. Supporting main ideas work as a unified whole, demonstrate critical use of course concepts, and are consistently supported with valid and specific evidence	A+ (19.5)	Introduction and conclusion provide excellent guides to the reader. Organization is thoughtful and supports the argument's logic. Paragraphs are consistently well developed and appropriately divided. Main ideas are linked with smooth and effective transitions. Citations are correctly formatted in-text and references.	A+ (39)	Exceptionally clear and efficient writing that is virtually free of mechanical errors. Excellent use of strong, positive, active language with powerfully constructed sentences. Possesses appropriate tone and awareness of intended audience.
		A++ (40)		A++ (20)		A++ (40)	

The following R script provides the backend calculations for the Shiny App for the calcGrade() function.

```
library(dplyr)
calcGrade <- function(ContentGrade_Alpha, OrganizationGrade_Alpha,
WritingGrade_Alpha, TotalPoints){
  # Points Assigned for Content (ifelse) ----------------------------------------------------
  PointsAssigned_Content <- ifelse(ContentGrade_Alpha == "F", 0.66,
                                   ifelse(ContentGrade_Alpha == "D", 0.67,
                                          ifelse(ContentGrade_Alpha == "C-", 0.70,
                                                 ifelse(ContentGrade_Alpha == "C", 0.75,
                                                        ifelse(ContentGrade_Alpha == "C+", 0.78,
                                                               ifelse(ContentGrade_Alpha == "B-", 0.80,
                                                                      ifelse(ContentGrade_Alpha == "B",
0.85,
                                                                             ifelse(ContentGrade_Alpha ==
"B+", 0.88,
                                                                                    ifelse(ContentGrade_Alpha
== "A-", 0.90,
ifelse(ContentGrade_Alpha == "A", 0.95,
ifelse(ContentGrade_Alpha == "A+", 0.97,
ifelse(ContentGrade_Alpha == "A++", 1.00, NA)))))))))))

  # Points Assigned for Organization (ifelse) ----------------------------------------------
  PointsAssigned_Organization <- ifelse(OrganizationGrade_Alpha == "F", 0.66,
                                        ifelse(OrganizationGrade_Alpha == "D", 0.67,
                                               ifelse(OrganizationGrade_Alpha == "C-", 0.70,
                                                      ifelse(OrganizationGrade_Alpha == "C", 0.75,
                                                             ifelse(OrganizationGrade_Alpha == "C+",
0.78,
```

```
                                                ifelse(OrganizationGrade_Alpha ==
"B-", 0.80,
                                                    ifelse(OrganizationGrade_Alpha
== "B", 0.85,
ifelse(OrganizationGrade_Alpha == "B+", 0.88,
ifelse(OrganizationGrade_Alpha == "A-", 0.90,
ifelse(OrganizationGrade_Alpha == "A", 0.95,
ifelse(OrganizationGrade_Alpha == "A+", 0.97,
ifelse(OrganizationGrade_Alpha == "A++", 1.00, NA)))))))))))

  # Points Assigned for Writing (ifelse) ------------------------------------------------
--------
  PointsAssigned_Writing <- ifelse(WritingGrade_Alpha == "F", 0.66,
                            ifelse(WritingGrade_Alpha == "D", 0.67,
                                ifelse(WritingGrade_Alpha == "C-", 0.70,
                                    ifelse(WritingGrade_Alpha == "C", 0.75,
                                        ifelse(WritingGrade_Alpha == "C+", 0.78,
                                            ifelse(WritingGrade_Alpha == "B-", 0.80,
                                                ifelse(WritingGrade_Alpha == "B",
0.85,
                                                    ifelse(WritingGrade_Alpha ==
"B+", 0.88,
                                                        ifelse(WritingGrade_Alpha
== "A-", 0.90,
ifelse(WritingGrade_Alpha == "A", 0.95,
ifelse(WritingGrade_Alpha == "A+", 0.97,
ifelse(WritingGrade_Alpha == "A++", 1.00, NA)))))))))))

  # Points calculation ------------------------------------------------------
```

```
  # [in the future, replace constant values with variable options in the app]

  ContentPoints = TotalPoints * PointsAssigned_Content * 0.4
  OrganizationPoints = TotalPoints * PointsAssigned_Organization * 0.2
  WritingPoints = TotalPoints * PointsAssigned_Writing * 0.4
  PointsMissed = TotalPoints - sum(ContentPoints, OrganizationPoints,
WritingPoints)
  PointsEarned = TotalPoints - PointsMissed
  gradePct = (TotalPoints - PointsMissed) / TotalPoints

  # Final Grade Assigned (ifelse) ----------------------------------------------------------

  GradeAssigned <- ifelse(gradePct <= 0.66,                    "F",
                    ifelse(gradePct < 0.66  & gradePct < 0.7,   "D",
                        ifelse(gradePct >=0.7   & gradePct <= 0.72, "C-",
                            ifelse(gradePct >= 0.73 & gradePct <= 0.76, "C",
                                ifelse(gradePct >= 0.77 & gradePct  <= 0.79,"C+",
                                    ifelse(gradePct >= 0.80 & gradePct <= 0.82, "B-
",
                                        ifelse(gradePct >= 0.83 & gradePct <=
0.86, "B",
                                            ifelse(gradePct >= 0.87 & gradePct <=
0.89, "B+",
                                                ifelse(gradePct >= 0.90 & gradePct
<= 0.92, "A-",
                                                    ifelse(gradePct >= 0.93 &
gradePct <= 0.96, "A",
                                                        ifelse(gradePct >= 0.97 &
gradePct < 1,     "A+",
                                                            ifelse(gradePct == 1,
"A++", NA)))))))))))
  # Table Creation -------------------------------------------------------
  Table01 = cbind(PointsEarned, GradeAssigned, gradePct, TotalPoints)
}
```

The following code provides the following functions for the application. The user interface, of ui, the server or server, and the shiny app or shinyApp.

```
library(shiny)
library(dplyr)
library(xtable)
library(httr)
source("calcGrade.R")
GradeScale_Alpha <- c("A++","A+","A","A-","B+","B","B-","C+","C","C-","D","F")



ui <- fluidPage(
  titlePanel("Multi-Dimensional Grading Calculator"), # titlePanel()
  sidebarLayout(
    sidebarPanel("Assigned Points",
                 numericInput(inputId = "tagPoints_assigned",
                              label = "Input the assigned points.",
                              value = 100), # numreicInput()
                 selectInput(inputId = "tagContent_grade",
                             label = "Use the dropdown to assign a Content grade",
                             choices = GradeScale_Alpha,
                             selectize = F), # selectInput(tagContent_grade)
                 selectInput(inputId = "tagOrganization_grade",
                             label = "Use the dropdown to assign a Organization grade",
                             choices = GradeScale_Alpha,
                             selectize = F), # selectInput(tagOrganization_grade)
                 selectInput(inputId = "tagWriting_grade",
                             label = "Use the dropdown to assign a Writing grade",
                             choices = GradeScale_Alpha,
                             selectize = F) # selectInput(tagWriting_grade)
    ), # sidebarPanel()
    mainPanel("Main Panel",
              tableOutput(outputId = "tagGradeTable_out")) # mainPanel()
```

```
  ) #sidebarLayout()
) # fluidPage()


server <- function(input, output){
  output$tagGradeTable_out <- renderTable({
    tagTable01 <-
calcGrade(input$tagContent_grade,input$tagOrganization_grade,input$tagWritin
g_grade,input$tagPoints_assigned)
  }) # renderTable(){}
} # function(input,output){}


shinyApp(ui = ui, server = server)
```

References

Airasian, Peter W., and Arlen Gullickson. 1994. “Examination of Teacher Self-Assessment.” *Journal of Personnel Evaluation in Education* 8 (2): 195–203. https://doi.org/10.1007/BF00972263.

Brookhart, Susan M. 1999. *The Art and Science of Classroom Assessment. The Missing Part of Pedagogy. ASHE-ERIC Higher Education Report, Volume 27, Number 1.* ERIC.

Brophy, Jere. 2013. *Motivating Students to Learn*. Routledge.

Brown, Peter C., Henry L. Roediger, and Mark A. McDaniel. 2014. *Make It Stick*. Harvard University Press.

Chang, Winston, Joe Cheng, J. J. Allaire, Yihui Xie, and Jonathan McPherson. 2018. *Shiny: Web Application Framework for R*. https://CRAN.R-project.org/package=shiny.

Fink, L. Dee. 2003. “A Self-Directed Guide to Designing Courses for Significant Learning.” *University of Oklahoma* 27 (11).

Grow, Gerald O. 1991. “Teaching Learners to Be Self-Directed.” *Adult Education Quarterly* 41 (3): 125–49.

Heen, Sheila, and Douglas Stone. 2014. “Find the Coaching in Criticism.” *Harv Bus Rev*, 108–11.

Leydens, J., and D. Thompson. 1997. *Writing Rubrics Design (EPICS) I*. Internal Communication, Design (EPICS) Program, Colorado School of Mines ….

Moskal, Barbara M. 2000. *Scoring Rubrics: How?* ERIC Clearinghouse on Assessment and Evaluation, University of Maryland.

Peter, Airasian W. 1994. “Classroom Assessment.” *New York: McGraw–Hi|| Inc*.

R Core Team. 2018. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org/.

RStudio, Inc. 2013. *Easy Web Applications in R.*

Willingham, Daniel T. 2007. “Critical Thinking: Why It Is so Hard to Teach?” *American Federation of Teachers Summer 2007, p. 8-19*.