

A web-based application for generating new versions of math-based assessments created using LaTeX

Christopher J. Casement

casementc@gmail.com
Department of Mathematics
Fairfield University
Fairfield, Connecticut, 06824
USA

Grant Innerst

grant.innerst@gmail.com
Department of Mathematics
Shippensburg University
Shippensburg, Pennsylvania 17257
USA

Melissa Innerst

melissainnerst@gmail.com
Department of Mathematics
Juniata College
Huntingdon, Pennsylvania 16652
USA

Abstract

The creation of multiple versions of assessments is common in mathematics-based courses at all levels and across disciplines. It can be helpful in facilitating the use of both mastery-based and standards-based assessment models, as well as in mitigating cheating on assessments. The process of making new versions, however, is oftentimes arduous and time-consuming, even with the assistance of software designed to simplify the process. In this article, we provide an overview of some of the widely-used software for both mathematics and mathematics-centric education – both free and proprietary – that assists educators when making assessments, followed by a discussion of various drawbacks to the software. We then describe a free, user-friendly, web-based application we developed to address these drawbacks. The application enables teachers of mathematics-based courses to easily and quickly generate numerous new versions of assessments typed using the popular typesetting system LaTeX. We also demonstrate the application's versatile features through the use of detailed examples.

1. Introduction

In mathematics-based courses, the ability to quickly create personalized assessments has both pedagogical and practical advantages. It helps facilitate the use of a mastery-based or standards-based grading assessment model and is useful in mitigating cheating during assessments.

One trademark of a mastery-based or standards-based grading classroom assessment model is that students are allowed multiple attempts to demonstrate mastery of a topic on assessments. Allowing multiple attempts has been shown to lower math anxiety, support a growth mindset, and also help students develop a deeper understanding of the course material [11]. The ability to create assessments that are different enough for authentic reassessment but a similar level of difficulty is key to facilitating this classroom model and is also a major hurdle in implementation. The act of

creating multiple versions of the same assessment is tedious and time-consuming, and educators often cannot devote enough time to make the assessments authentically different. This problem has been exacerbated during the COVID-19 pandemic as educators are asked to do more work to prepare their classes for hybrid instruction.

In addition to pedagogical advantages, personalized assessments are often utilized to mitigate plagiarism [4, 15, 16]. Plagiarism on assessments in both in-person and online environments has been widely studied [20]. While technological tools have made it easier to detect plagiarism, the same technologies have made it easier to cheat as well [7, 20]. Personalized assessments work to mitigate incidents of plagiarism by offering each student, or subsets of students, a unique problem set. This strategy limits their ability to copy work from others and, in cases in which problems are not pulled from established problem banks where answers exist, also limits a student's ability to copy work from the outside sources.

A variety of different software tools have been developed that offer the capability to create personalized assessments. These tools come in a wide range of price, flexibility, and ease-of-use. They also take many different forms, such as online educational systems, test generators, and specific packages within larger pieces of software. Additionally, they differ in how students interact with the assessments created by the tool.

There is extensive research regarding the effect on both student understanding of material and performance on assessments when online versus traditional assessment methods are used [10]. Online assessment offers many benefits such as providing immediate and personalized feedback and multiple attempts at similar problems when appropriate, both of which can encourage mastery of a topic [10, 13, 19]. However, a drawback to such assessments is that they might entice a student to utilize a trial-and-error submission style, where the student either repeats a problem type until a repeated question appears or guesses answers without cognitive thought until their guess is correct. Strategies such as these emphasize the final answer rather than the importance of showing work and building answers [5, 17].

In this work, we introduce an alternative to the existing tools: an application that is free, web-based, extremely flexible, and which leverages the powerful syntax of the LaTeX typesetting language to create multiple versions of assessments for offline use. The outline is as follows. In Section 2, we provide a background on existing tools and software that assist in the creation of personalized assessments. In Section 3, we describe our web-based application and its features in detail. In Section 4, we provide multiple examples of how users can utilize the application to generate multiple versions of assessments. We conclude with a discussion of the application's relevancy in Section 5.

2. Background and related work

Many tools that facilitate multiple versions of an assessment exist as propriety software from textbook publishers in the form of interactive, online educational systems, including MyLab Math by Pearson Education, WileyPLUS by John Wiley & Sons, Inc., and WebAssign by Cengage Learning. The extensive question banks included with these tools allow for the automatic generation and grading of homework assessments, quizzes, and exams. Additionally, instructors are not limited only to the use of premade questions. These tools give users the flexibility to write and include their own questions on assessments, including the capability to vary algorithmic values within the questions so that students experience different versions of the assessment. The advantages of these propriety tools include the high level of convenience and flexibility offered. The major disadvantage of these tools is their cost, which often falls upon the students.

There are free alternatives, the most popular of which is WeBWorK. WeBWorK is an open-source, online homework system for math-based courses supported by the Mathematical Association of America (MAA) and the National Science Foundation (NSF). Like its proprietary counterparts, WeBWorK has the capability to automatically generate and grade multiple versions of the same question type. Another main open-source option is MyOpenMath, which contains a large question bank and the ability for users to create their own questions and vary algorithmic values using custom question-builder software.

While all of the previously mentioned systems offer a wide range of capabilities, we have identified some disadvantages in terms of ease of use and the ability to interact with the material offline. A common disadvantage among the options previously mentioned, both open-source and proprietary, is the learning curve for instructors using the tool. This includes both learning how to access the tool and how to use it to create questions and assessments. Each tool has a separate interface for creating questions, and the interfaces vary greatly in their ease of use. Additionally, some of the programs are not compatible with the LaTeX typesetting system, which many educators use to create assessments. Lastly, while these systems do offer users the ability to download material for offline use, only one version of the assessment can be downloaded at a time.

Another set of tools that facilitates the creation of multiple versions of an assessment is test generators. As with the online educational systems, the capabilities and pricing of different exam generators vary widely. Textbook publishers often provide exam generation software that accompanies their books. Similar to the software mentioned above, these generators benefit from large question banks and the ability to write personalized questions, but suffer from specialized interfaces that take time to learn and the lack of ability to vary more than numbers in a question. In addition to software provided by book publishers, there are numerous other programs that do essentially the same tasks.

Non-web-based options come in the form of various specialized packages in the typesetting language LaTeX. One popular example is the `examdesign` package, which offers users a set of tools that aids in the creation of exams and question sets. Users can make use of LaTeX environments that create templates of common types of exam questions like true/false, multiple-choice, fill-in-the-blank, and many others. Among other features, users have the ability to generate multiple versions of the same exam by simply specifying the number of versions to be made. Multiple versions are attained by randomly permuting the order of questions within a section. Other packages, like the `xsim` package, offer similar functionality, including the ability to randomly permute questions. The packages offer the user flexibility in creating assessments, but in the context of generating multiple versions of an assessment, their abilities are limited to permuting questions or choices within questions. Another tool for scientific word processing is Scientific WorkPlace from MacKichan Software. Scientific WorkPlace is a GUI for editing LaTeX source files that offers the same ease-of-use as a word processor and does not require the user to learn LaTeX [12]. A notable feature of this software is the integration of the MuPad computer algebra system, which allows for the generation of randomized questions, including graphics, and the development of solutions to those questions. However, in 2021, MacKichan Software announced that it will no longer be selling licenses to its products and, due to the integration of a propriety computer algebra system, Scientific WorkPlace is currently unable to be offered as open-source software.

3. Description of the application

We now discuss the web-based application and its capabilities for generating new versions of LaTeX-based documents. The application was created using the `Shiny` package [6] in R [18] and is available

for free online at <https://educationapps.shinyapps.io/newVersionsGenerator>. In addition to `shiny`, the app is bolstered by a multitude of packages, most notably two focused on character string processing: `stringi` [9] and `stringr` [21]. We now discuss strengths of the app, including how they address the issues and concerns raised in Sections 1 and 2, followed by a detailed description of the app's capabilities.

3.1 Strengths of the application

The app possesses various features that help it stand out from other tools available for math-based courses. First, it contains flexible options that enable instructors to generate many unique versions of LaTeX-based documents containing examples and assessments, offering potentially countless practice problems and personalized assessments, while maintaining the LaTeX typesetting used in the original file. Second, the app works in any web browser, making it accessible on numerous types of devices, including laptops, tablets, and smartphones. Third, the app is available for free and does not require the user to create an account. Next, it employs a user-friendly, point-and-click interface where the user can simply type their desired specifications into text boxes instead of needing to learn and follow strict formatting requirements. Fifth, the app includes step-by-step instructions that navigate the user through the full process of generating new versions of assessments while keeping the time it takes to do so to a minimum. Having presented a few of the strengths of the app, we now discuss its capabilities in detail.

3.2 Capabilities of the application

The application was designed to enable the generation of new versions of math-based documents (e.g., class assessments) typed in LaTeX. The app contains two main implementations that allow the user to accomplish this goal: (1) the replacement of either specific or all numbers in the document, and (2) the creation of a question bank from which questions are randomly drawn. We now detail the main options available to the user for these implementations.

3.2.1 Making new versions of uploaded files

The first of the two main features offered by the app allows for the replacement of numbers in an existing LaTeX document. The user has the option of making replacements on a per-number or per-number-type (i.e., integer vs. decimal) basis, as we discuss further in Sections 3.2.1.1 and 3.2.1.2. In both cases, the user specifies sets of numbers to be considered as replacements, and the app randomly selects values from those sets to be used as the actual replacements for the original numbers selected by the user. The user also sets the number of new versions of their original document to make. As the number of values to be replaced and/or the number of values in the replacement sets grows, the number of potential unique versions of the document increases. The app, which reruns the random selection process each time a new version is generated, thus enables the generation of many unique versions of a math-based assessment. In addition, the app is able to do so quickly once the user has specified all of the desired replacement options. All new versions are downloaded as individual `.tex` files in a single zipped folder, preserving all original LaTeX code used aside from replacements made by the app. We now describe the process of replacing specific values in LaTeX documents.

3.2.1.1 Replacing specific values

If the user wants to replace numbers on a per-number basis, they should select the “Specific Values” option in the “Make Replacements” tab. When doing so, they are able to see every number in a specified question (when the question contains no parts) or every number in the specified part of a chosen question (when the question does contain parts).

After the user inputs a question (and part, when applicable), the interface presented to the user is split into two main parts. The right-hand part of the page displays the output of the uploaded LaTeX code. To enable the user to more easily differentiate among questions and parts when reading the output, including the question (and part) selected, the app uses bold and colored text as well as highlighting – changes made for display purposes only and not in the new versions generated by the app. In terms of highlighting, when the question contains no parts, the output for that entire question is highlighted in yellow. On the other hand, when the question does contain parts (and a part other than the main piece is selected), the main piece of the selected question is highlighted in gray, while the selected part of that question is highlighted in yellow. The left-hand portion of the page consists of a panel that contains inputs for the desired question and part, as well as a display of every number contained in the chosen question (and part, when applicable), in the exact order they appear in the original LaTeX document. The app is able to identify and read numbers typed as either English numerals (e.g., “1”) or English words (e.g., “one”) and maintains the type once replacements are made later (including lower case vs. upper case usage) when an English word is used. The panel also consists of inputs for the user to set when specifying what the potential set of replacements should consist of for the respective number. When the user wishes to replace a particular number, they click the “Replace” checkbox next to that number and then select one of three options: (1) sequence, (2) manual input, or (3) a number or set of numbers used previously. If the user elects for a sequence of values, they must set a minimum and maximum value for the sequence, as well as the increments to use between them. If the user elects to input a manual set of values, they simply type the values one at a time, separated by commas. If they elect to use a set of values used previously, they are able to select a desired set from a list of all (unique) sets already specified.

Figure 1 contains an example of a short LaTeX document with two questions, where the second part of the second question has been selected. The three numbers contained in that part are displayed, as are examples of inputs (with their defaults) available to the user. For a complete example of generating new versions of a document through the replacement of specific values, see Section 4.

3.2.1.2 Replacing all values

If the user wants to replace all numbers in a document based on the type of number (e.g., integer vs. decimal) and using identical replacement sets to sample from for each, they should select the “All Values” option in the “Make Replacements” tab. They then are able to make replacements for three types of numbers – integers, decimals, and common trigonometric angles – as well as common trigonometric functions. Each time a number or trigonometric function is found in the document, the app draws a replacement at random from the respective set of possible replacements, which we now describe.

When replacing integers or decimals, the user specifies a replacement set in the form of a sequence of numbers. To do so, they input a minimum and maximum value, as well as the increments to use between the minimum and maximum. When replacing a common trigonometric angle, the user first selects the measurement type of the angle – radians or degrees – and then forms a replacement set by selecting individual angles from a drop-down list that contains common values (e.g., $\pi/4$ and

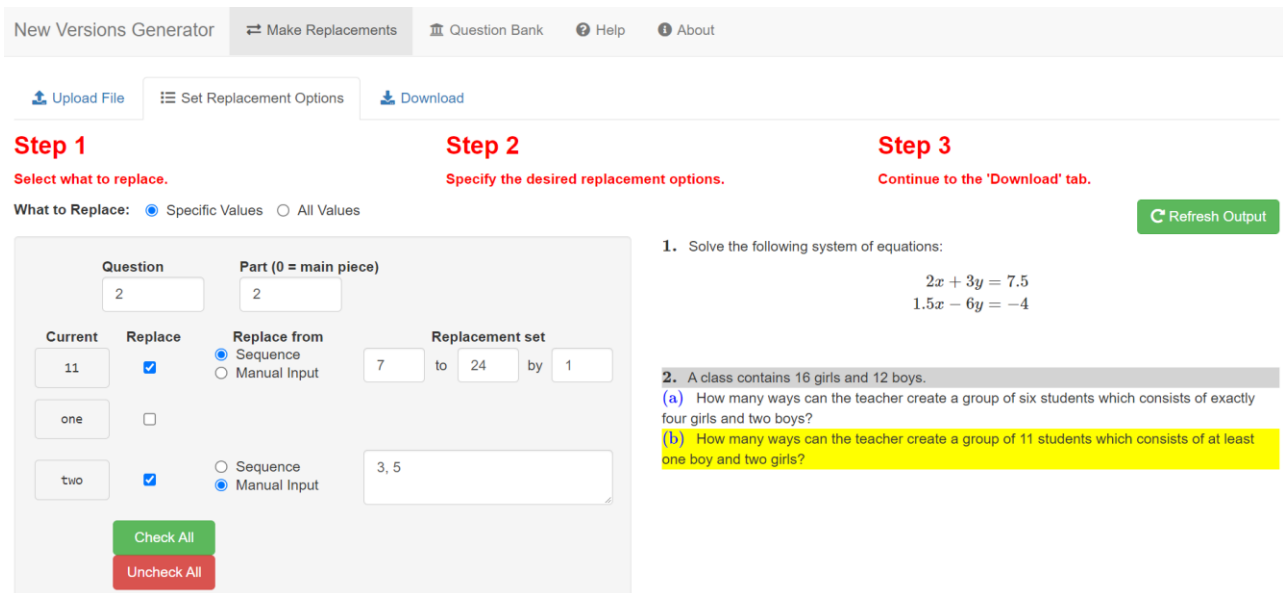


Figure 1: Example of app inputs, including replacement inputs by number for a specified question and part, when the user has chosen to replace specific values (left), as well as output of the original LaTeX document (right).

$3\pi/2$). When replacing a trigonometric function, the app allows for the replacement of any of six possible functions (sine, cosine, tangent, cosecant, secant, and cotangent), with a replacement set formed through the user’s selection of individual trigonometric functions from a drop-down list of the six common functions.

Figure 2 contains an example where all types of numbers considered by the app, as well as trigonometric functions, will be replaced when new versions of the uploaded document are generated. For each option, the replacement sets are formed based on the inputs specified by the user. For instance, every integer in the document will be replaced by one number in the set $\{1, 2, 3, \dots, 10\}$; every decimal will be replaced by a number in the set $\{0, 0.1, 0.2, \dots, 10\}$; every special trigonometric angle will be replaced by an angle from the set $\{\pi/6, \pi/3, 2\pi/3, 5\pi/4\}$; and every common trigonometric function will be replaced by one function from the set {sine, cosine, tangent, cosecant, secant, cotangent}. As a note, the app maintains the original formatting used when replacing trigonometric angles (e.g., $\frac{\pi}{6}$ vs. $\frac{\pi}{6}$ vs. $\pi/6$) and functions (e.g., \sin vs. \sin).

3.2.2 The “Question Bank” feature

The second of the primary features implemented in the app is a question bank. When inputting questions, the user has two options: (1) upload a LaTeX file that contains all of the questions, or (2) manually type (or copy and paste) and add questions one at a time in the app. To offer even more flexibility to the user when inputting questions, such as if the user initially uploads a file but then decides to add a question contained in another file, option (2) works in conjunction with option (1).

The “Question Bank” feature also contains important functionality when updating groups (e.g., adding and removing) and questions (e.g., copying, moving, and removing). It also provides the user with the option to either add new versions of existing questions or update them (also an option for individual parts) by replacing select numbers with numbers randomly selected from manual sets or sequences of numbers determined by the user, similar to how the replacements work

New Versions Generator Make Replacements Question Bank Help About

Upload File Set Replacement Options Download

Step 1
Select what to replace.
What to Replace: Specific Values All Values

Step 2
Specify the desired replacement options.

Step 3
Continue to the 'Download' tab.

Replace Integers
Specify the conditions for the randomly-generated integer replacements.
Min: 1 Max: 10 Increments: 1

Replace Decimals
Specify the conditions for the randomly-generated decimal replacements.
Min: 0 Max: 10 Increments: 0.1

Replace Trig Functions
Click the box below and select all functions you want considered to replace all trig functions in your document.
sin cos tan csc sec cot

Replace Trig Angles
Unit of Measure: Radians Degrees
Click the box below and select all angles you want considered to replace all trig angles (in radians) in your uploaded document.
pi/6 pi/3 2pi/3 5pi/4

Figure 2: Example of replacement options when the user has chosen to replace all numbers of particular types (e.g., integers, decimals, and common trigonometric angles) as well as trigonometric functions in their LaTeX document.

in the “Make Replacements” tab. Once the user has finished making the bank, the app allows them to select how many questions to randomly draw from each group, as well as how they want the questions ordered – by maintaining the original order or using a randomized one. The app, which reruns the random selection of questions within a group each time a new version is generated, thus enables the generation of many unique versions of an assessment. All new versions are downloaded as individual .tex files, in a single zipped folder. Section 4 contains a step-by-step example of the use of the “Question Bank” feature to generate new versions of a LaTeX document.

3.3 Requirements of uploaded LaTeX documents

The application generates new versions of documents typed in LaTeX so long as the user utilizes the commonly-used *enumerate* environment, along with the associated `\item` command, to identify and separate the individual questions contained in the document. To allow for differing preferences when utilizing *enumerate*, the app works if users include this environment in one of two ways: (1) one main *enumerate* environment containing all questions, which are separated using individual `\item` commands, or (2) one *enumerate* environment for either each question or each set of questions. Figures 3 and 4 include examples of code that contain these two *enumerate* options, while Figure 5 contains an example of code that is not compatible with the app due to the absence of *enumerate*. In all three examples, additional formatting commands (e.g., `\noindent`) are left out to improve clarity. When specifying parts of main questions, the user may use nested *enumerate* environments in the typical way, as can also be seen in Figures 3 and 4.

4. Examples of using the application to generate new versions of assessments

The two primary features offered by the application – making replacements of numbers (and trigonometric functions) and the question bank – were described in Section 3. We now demonstrate the app’s ability to generate new versions of LaTeX files through the use of short, sample assessments

created in LaTeX. We provide step-by-step instructions the user would follow from start to finish, first using the “Make Replacements” feature and then using the “Question Bank” feature.

```

\begin{document}

\begin{enumerate}
\item Find  $f'(x)$  when  $f(x) = 6x^4 + 2x^2 - 5$ .

\item Let  $g(x) = x^2 - 3x + 2$ .
\begin{enumerate}
\item Find  $g'(x)$ .
\item Find all critical points of  $g(x)$ .
\end{enumerate}
\end{enumerate}

\end{document}

```

Figure 3: Example of LaTeX code in which one overarching *enumerate* environment encompasses the code that contains all of the questions. Code like this is compatible with the app.

```

\begin{document}

Find  $f'(x)$  for the function below.
\begin{enumerate}
\item  $f(x) = 6x^4 + 2x^2 - 5$ 
\end{enumerate}

Let  $g(x) = x^2 - 3x + 2$ .
\begin{enumerate}\setcounter{enumi}{1}
\item Find  $g'(x)$ .
\item Find all critical points of  $g(x)$ .
\end{enumerate}

\end{document}

```

Figure 4: Example of LaTeX code in which a new *enumerate* environment is used for each main question. Such usage makes the code compatible with the app.

```

\begin{document}

1. Find  $f'(x)$  when  $f(x) = 6x^4 + 2x^2 - 5$ . \\
2. Let  $g(x) = x^2 - 3x + 2$ . \\
(a) Find  $g'(x)$ . \\
(b) Find all critical points of  $g(x)$ .

\end{document}

```

Figure 5: Example of LaTeX code that is not compatible with the app due to the absence of *enumerate* environment(s).

We note that additional instructions can be found in the “Help” tab in the app. Further, links to video tutorials for using the app can be found below:

- Using the “Make Replacements” feature to replace select values:
<https://www.youtube.com/watch?v=QJFjb3Iph0A>
- Using the “Make Replacements” feature to replace all values:
<https://www.youtube.com/watch?v=ORYHRazal0Q>
- Using the “Question Bank” feature:
<https://www.youtube.com/watch?v=nyvK6AJz2W4>

as well as in the app’s “Help” tab.

4.1 Using the “Make Replacements” feature to create new versions of an existing assessment

When making replacements, the user begins on the “Upload File” tab. There, they select the desired LaTeX file from their device and upload it to the app. Figure 6 shows an example of what the app displays after the upload process, which takes place almost instantly, is complete. The user then proceeds to Step 2, where they must choose the way in which they used *enumerate* environments in their file, with Figure 7 showing the pop-up window that contains the two options.

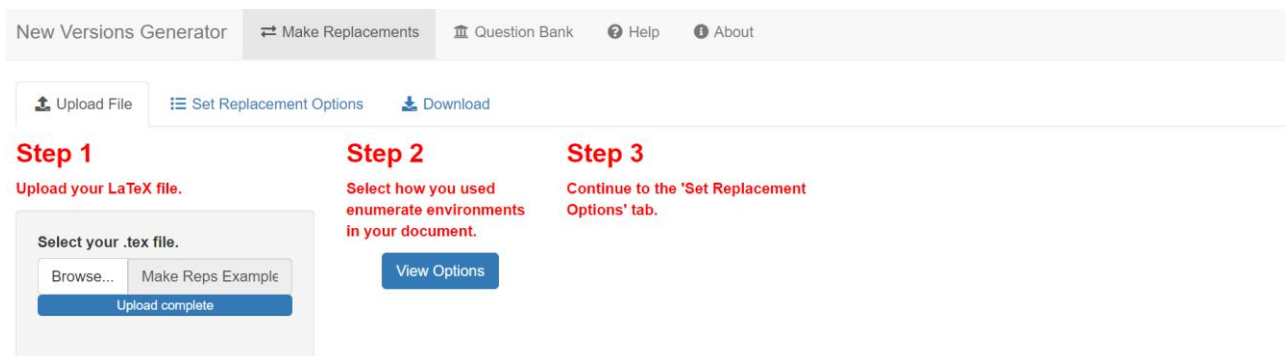


Figure 6: Output after uploading a LaTeX file.

After clicking the button that corresponds to the appropriate *enumerate* option, the user navigates to the “Set Replacement Options” tab, where they choose between replacing either specific numbers in their document or all values of certain types (e.g., decimals or integers), as described in Section 3. Suppose the user chooses to replace specific values. They then select the “Specific Values” option, at which point they commence the replacement specification process. Also suppose the user chooses to replace numbers in part 2 of question 2 only. They would first input “2” for both “Question” and “Part,” after which the app would update and display the inputs and output found in Figure 8, including replacement options for the three numbers found in the selected part: “11,” “one,” and “two.” As can be seen in Figure 8, the user has chosen to replace “11” with a randomly-selected integer from the set $\{7, 8, 9, \dots, 24\}$ and “two” with a number (as an English word to preserve the original number type) randomly selected from the set $\{\text{three}, \text{five}\}$. These specifications are saved automatically for the user, and new random selections are made later for each new version of the file. While this example demonstrates the replacement process involving only one part of one question, the app allows for the replacement of numbers from all questions and parts contained in the file.

Click the button that corresponds to the way you used enumerate environments in your document. Then close this window.

Enumerate Method	Example Code	Example Output
<p>Option 1: One* main enumerate environment for the entire document</p> <p>*You may still use nested enumerate environments within the main enumerate environment if you have questions with multiple parts.</p>	<pre>\begin{document} \begin{enumerate} \item Find \$5 + 3\$. \item Find \$7 + 2\$. \item Find \$5 - 3\$. \item Find \$7 - 2\$. \end{enumerate} \end{document}</pre>	<p>1. Find 5 + 3. 2. Find 7 + 2. 3. Find 5 - 3. 4. Find 7 - 2.</p> <p>Option 1</p>
<p>Option 2: Separate** enumerate environments for different questions</p> <p>**You may still use nested enumerate environments within these environments if you have questions with multiple parts.</p>	<pre>\begin{document} \noindent Find the following sums. \begin{enumerate} \item \$5 + 3\$ \item \$7 + 2\$ \end{enumerate} \noindent Find the following differences. \begin{enumerate}\setcounter{enumi}{2} \item \$5 - 3\$ \item \$7 - 2\$ \end{enumerate} \end{document}</pre>	<p>Find the following sums. 1. 5 + 3 2. 7 + 2</p> <p>Find the following differences. 3. 5 - 3 4. 7 - 2</p> <p>Option 2</p>

Figure 7: Pop-up window for selecting the type of *enumerate* method used.

New Versions Generator Make Replacements Question Bank Help About

Upload File Set Replacement Options Download

Step 1
Select what to replace.

What to Replace: Specific Values All Values

Step 2
Specify the desired replacement options.

Question	Part (0 = main piece)
2	2
Current	Replace
11	<input checked="" type="checkbox"/>
one	<input type="checkbox"/>
two	<input checked="" type="checkbox"/>
Replace from	Replacement set
<input checked="" type="radio"/> Sequence <input type="radio"/> Manual Input	7 to 24 by 1
<input type="radio"/> Sequence <input checked="" type="radio"/> Manual Input	3, 5

Check All Uncheck All

Step 3
Continue to the 'Download' tab.

Refresh Output

1. Solve the following system of equations:

$$2x + 3y = 7.5$$

$$1.5x - 6y = -4$$

2. A class contains 16 girls and 12 boys.
 (a) How many ways can the teacher create a group of six students which consists of exactly four girls and two boys?
 (b) How many ways can the teacher create a group of 11 students which consists of at least one boy and two girls?

Figure 8: Example of specifications made by the user when setting replacement options for a selected question and part.

Once they have finished setting all of the desired replacement specifications for their document, the user moves to the “Download” tab, where they select the number of new versions of the document to create. After clicking the “Make New Version(s)” button, the app generates the new versions, and a “Download All” button appears. The user may choose a main name for the new files and then download all of them in a single zipped folder. These inputs can be seen in Figure 9.

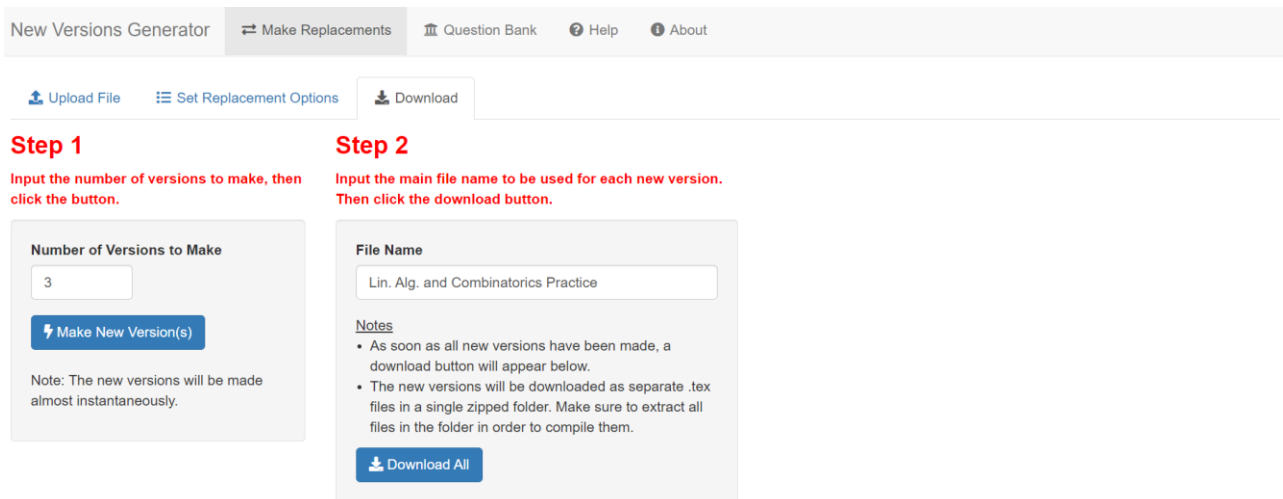


Figure 9: User inputs when generating and downloading new versions of a LaTeX document. The user has chosen to make three new versions of the document based on the specifications in Figure 8.

After downloading and extracting the new versions of the LaTeX document on their device, the user can open them in their typical LaTeX editor. Figures 10 and 11 show the original LaTeX code and one new version of that code, respectively. As expected, only two numbers from the original document changed – the “11” and “two” – and with numbers from the user’s specified replacement sets – “8” and “three” (along with number type preservation). These changes can be seen in the last “\item” line when comparing the code in Figures 10 and 11.

```

\documentclass[11pt]{article}
\usepackage{amsmath, amssymb, enumitem}

\begin{document}
\begin{center}
Linear Algebra and Combinatorics Practice
\end{center}

\begin{enumerate}
\item Solve the following system of equations:
\begin{align*}
2x+3y &= 7.5 \\
1.5x-6y &= -4
\end{align*}

\item A class contains 16 girls and 12 boys.
\begin{enumerate}
\item How many ways can the teacher create a group of six
students which consists of exactly four girls and two boys?
\item How many ways can the teacher create a group of 11
students which consists of at least one boy and two girls?
\end{enumerate}

\end{enumerate}
\end{document}

```

Figure 10: Example of a LaTeX assessment before using the app to replace any numbers.

```

\documentclass[11pt]{article}
\usepackage{amsmath, amssymb, enumitem}

\begin{document}
\begin{center}
Linear Algebra and Combinatorics Practice
\end{center}

\begin{enumerate}
\item Solve the following system of equations:
\begin{align*}
2x+3y &= 7.5 \\
1.5x-6y &= -4
\end{align*}
\item A class contains 16 girls and 12 boys.
\begin{enumerate}
\item How many ways can the teacher create a group of six
students which consists of exactly four girls and two boys?
\item How many ways can the teacher create a group of 17
students which consists of at least one boy and three girls?
\end{enumerate}
\end{enumerate}

\end{document}

```

Figure 11: Example of one new version (out of three generated by the app) of the LaTeX assessment found in Figure 10, after using the app to make replacements. Only two numbers, both in the final “\item,” were replaced, as requested by the user.

4.2 Using the “Question Bank” feature to create different versions of assessments

When using the “Question Bank” feature, the user begins on the “Input” tab. Suppose they choose to upload an existing LaTeX file containing questions – already sorted into premade groups using the appropriate syntax specified in the “Help” tab – that will be used in the bank. They then need to choose the *enumerate* method used in that file, from the options found in Figure 7. Once the user selects the appropriate option, the app places all of the uploaded questions into their respective groups, as can be seen in Figure 12. The user can then (optionally) move to the “Manage Questions” tab, where they have various options available to them regarding individual questions and groups, though this tab is more of use when the user does not sort the questions into groups in their original document.

At this point the user also has the option to update the numbers in existing questions/parts or add new questions/parts to the bank in the “Add Generated Questions” tab. Any new questions or parts are the same as existing ones except with numbers replaced based on the user’s specifications, similar to the way replacements are made using the “Make Replacements” feature. In the question bank, however, any changes or additions are displayed in the app (as opposed to only after downloading new versions) so that the user can view every question and part. Figure 13 shows the updated question bank after adding a question based on user specifications.

Once they are satisfied with the full bank of questions, the user moves to the “Select Questions” tab, where they input how many questions to draw at random from each of the groups in the bank, for each new version of the document created. They also specify the code they wish to comprise the beginning of each generated document. For instance, the user might choose to select only one question at random from each of the two groups, and to use the initial LaTeX code (i.e., all code before the questions begin) from the uploaded document, as can be seen in Figure 14.

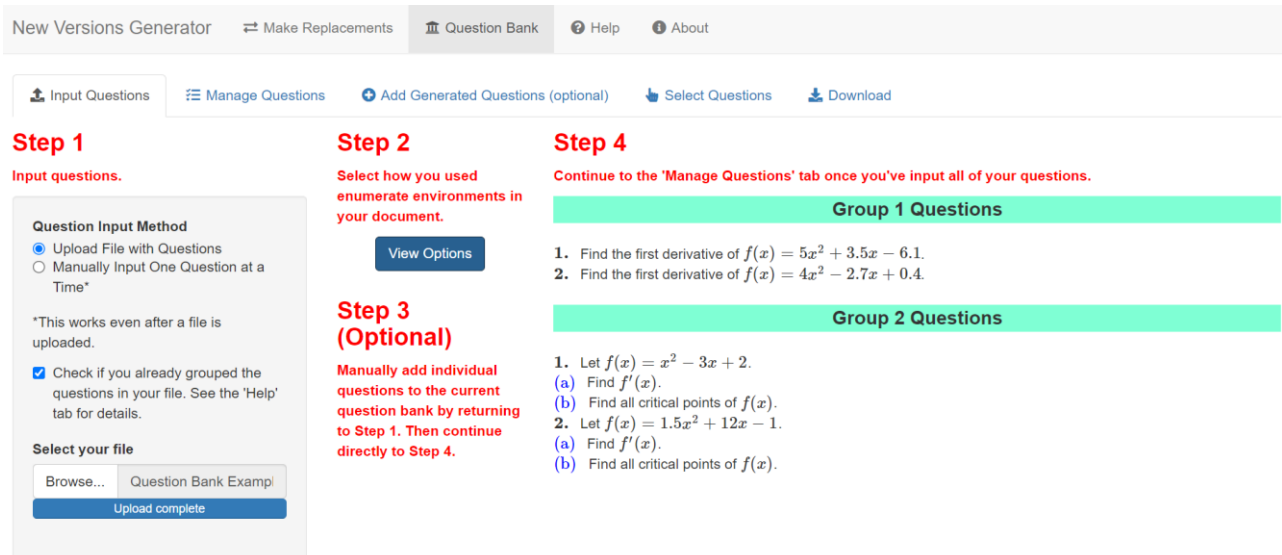


Figure 12: Example where the user has uploaded a file containing four questions in two premade groups. For Step 2, the pop-up window shown to the user is the same as that found in Figure 7.

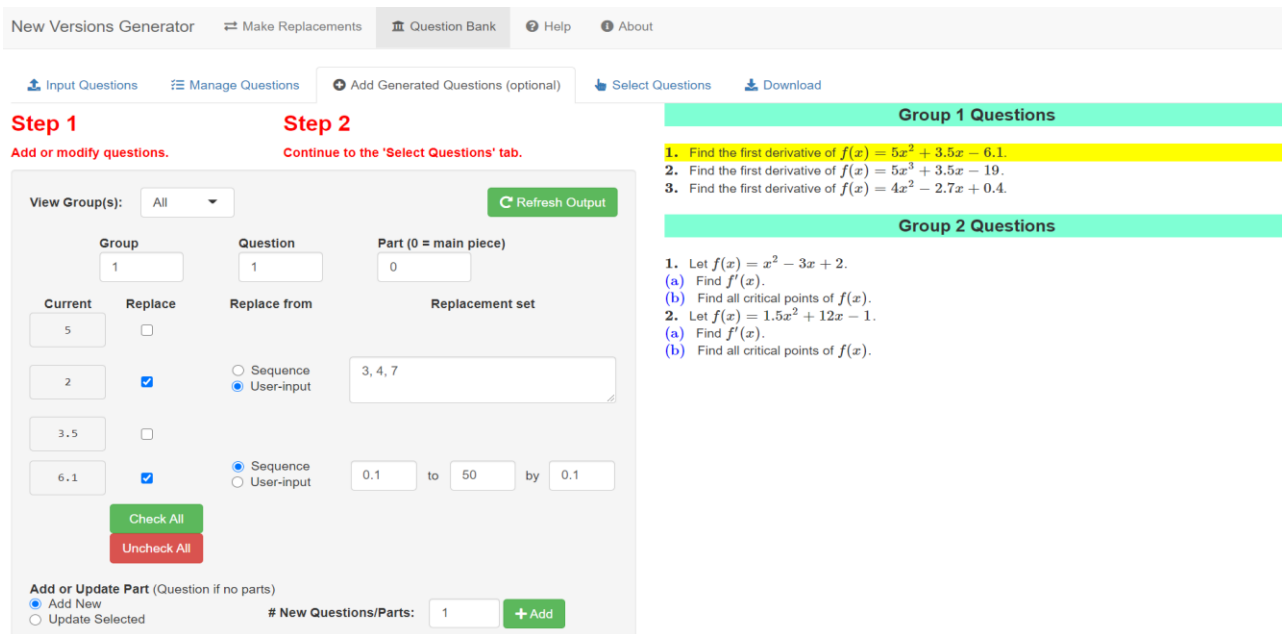
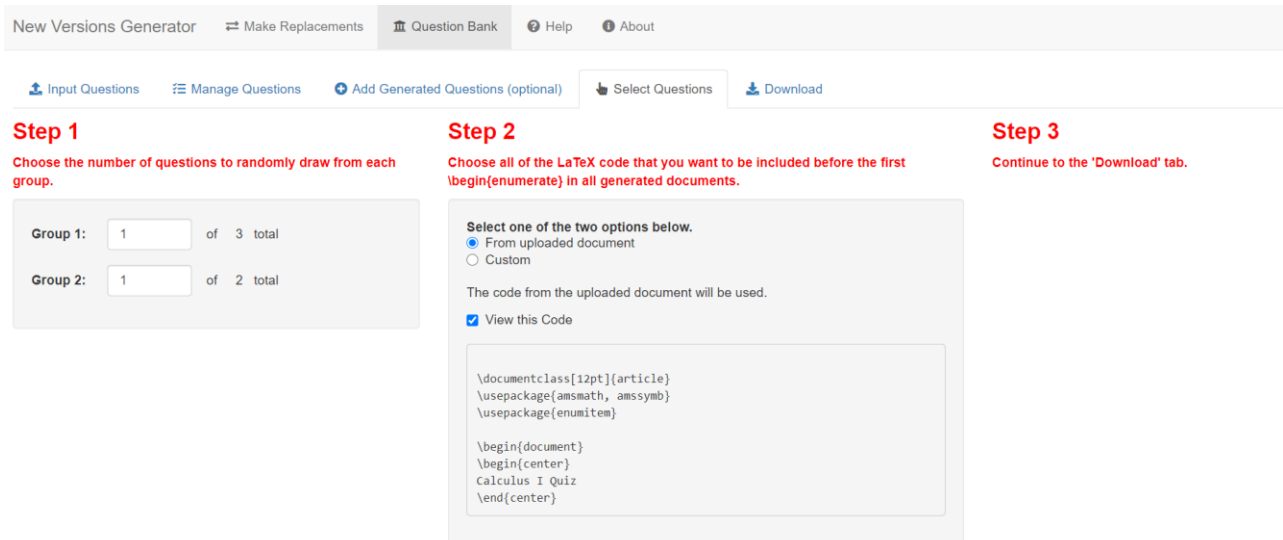


Figure 13: Output after the user has (optionally) added a new question (question 2) to group 1, with the new numbers for that question selected based on the user's specified replacement options.

The user then proceeds to the "Download" tab, where they select the number of versions of the document to create, in addition to their preference for the ordering of the questions (original vs. randomized). After clicking the "Make New Version(s)" button, the app generates the desired number of files, and a "Download All" button appears. The user may choose a main name for the new files and then download all of them in a single zipped folder. These inputs can be seen in Figure 15.



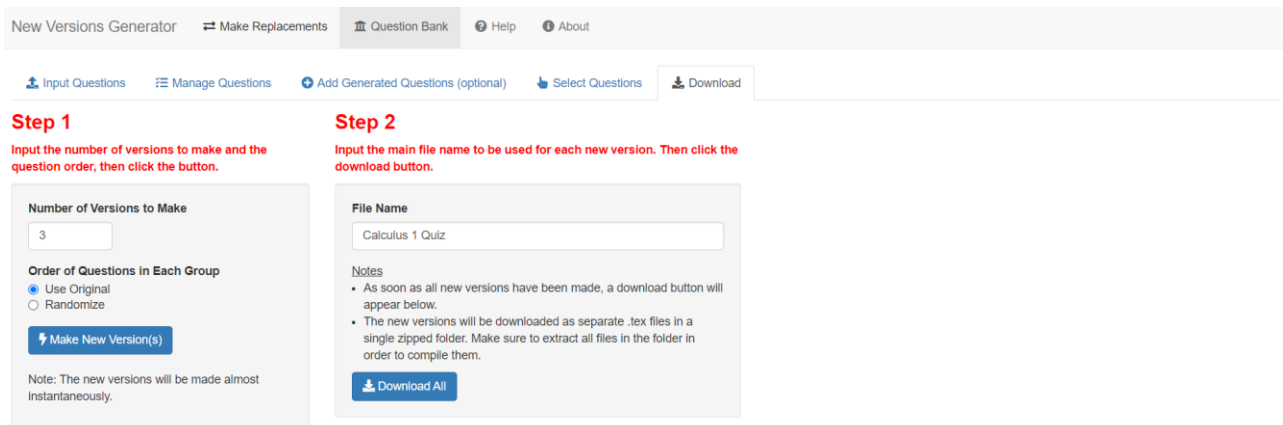
The screenshot shows the 'New Versions Generator' interface. At the top, there are navigation tabs: 'New Versions Generator', 'Make Replacements', 'Question Bank', 'Help', and 'About'. Below these are action buttons: 'Input Questions', 'Manage Questions', 'Add Generated Questions (optional)', 'Select Questions', and 'Download'. The interface is divided into three steps:

- Step 1:** 'Choose the number of questions to randomly draw from each group.' It contains two input fields: 'Group 1: 1 of 3 total' and 'Group 2: 1 of 2 total'.
- Step 2:** 'Choose all of the LaTeX code that you want to be included before the first `\begin{enumerate}` in all generated documents.' It offers two options: 'From uploaded document' (selected) and 'Custom'. Below, it states 'The code from the uploaded document will be used.' and has a checked checkbox for 'View this Code'. A code block shows LaTeX preamble and document start code:


```
\documentclass[12pt]{article}
\usepackage{amsmath, amssymb}
\usepackage{enumitem}

\begin{document}
\begin{center}
Calculus I Quiz
\end{center}
```
- Step 3:** 'Continue to the 'Download' tab.'

Figure 14: User inputs for the number of questions to draw at random from each group in the question bank, as well as what code should be included before the questions in all generated files.



The screenshot shows the 'New Versions Generator' interface. At the top, there are navigation tabs: 'New Versions Generator', 'Make Replacements', 'Question Bank', 'Help', and 'About'. Below these are action buttons: 'Input Questions', 'Manage Questions', 'Add Generated Questions (optional)', 'Select Questions', and 'Download'. The interface is divided into two steps:

- Step 1:** 'Input the number of versions to make and the question order, then click the button.' It contains a 'Number of Versions to Make' input field with the value '3', and 'Order of Questions in Each Group' radio buttons for 'Use Original' (selected) and 'Randomize'. A 'Make New Version(s)' button is present. A note states: 'Note: The new versions will be made almost instantaneously.'
- Step 2:** 'Input the main file name to be used for each new version. Then click the download button.' It contains a 'File Name' input field with the value 'Calculus 1 Quiz'. Below, there are 'Notes' and a 'Download All' button.

Figure 15: User inputs when generating and downloading LaTeX assessments using the “Question Bank” feature.

After downloading and extracting the new versions of the LaTeX document on their device, the user can open them in their typical LaTeX editor. Figures 16 and 17 show the original LaTeX code and one new version of the code, respectively. As expected, the new version contains only two questions, with one question drawn at random from each of the two groups created by the user – including the question added to Group 1 when using the app. The question ordering from the original file has also been preserved as directed.

5. Conclusion

In this article, we describe a free, web-based application that assists instructors of math-based courses when creating new versions of assessments typed using LaTeX. By possessing implementations that enable the replacement of either user-selected numbers individually or all numbers contained in a versatile functionalities. Our hope is that the app will be helpful for instructors of math-based courses at various levels and in wide-ranging disciplines when making practice problems and assessments.

```

\documentclass[12pt]{article}
\usepackage{amsmath, amssymb}
\usepackage{enumitem}

\begin{document}
\begin{center}
Calculus I Quiz
\end{center}

\begin{enumerate}
\item[]@group % Group 1

\item Find the first derivative of  $f(x) = 5x^2 + 3.5x - 6.1$ .

\item Find the first derivative of  $f(x) = 4x^2 - 2.7x + 0.4$ .

\item[]@group % Group 2

\item Let  $f(x) = x^2 - 3x + 2$ .
\begin{enumerate}
\item Find  $f'(x)$ .
\item Find all critical points of  $f(x)$ .
\end{enumerate}

\item Let  $f(x) = 1.5x^2 + 12x - 1$ .
\begin{enumerate}
\item Find  $f'(x)$ .
\item Find all critical points of  $f(x)$ .
\end{enumerate}

\end{enumerate}
\end{document}

```

Figure 16: Example of a LaTeX assessment before using the app’s “Question Bank” feature.

```

\documentclass[12pt]{article}
\usepackage{amsmath, amssymb}
\usepackage{enumitem}

\begin{document}
\begin{center}
Calculus I Quiz
\end{center}

\begin{enumerate}
\item Find the first derivative of  $f(x) = 4x^2 - 2.7x + 0.4$ .
\item Let  $f(x) = x^2 - 3x + 2$ .
\begin{enumerate}
\item Find  $f'(x)$ .
\item Find all critical points of  $f(x)$ .
\end{enumerate}
\end{enumerate}
\end{document}

```

Figure 17: Example of one new version (out of three generated by the app) of the LaTeX assessment found in Figure 16, after using the app’s “Question Bank” feature. file, as well as the ability to build a question bank in a multitude of ways, the app provides users with

Acknowledgements

We offer our sincere thanks to the editors of eJMT as well as the anonymous reviewers for their helpful comments that strengthened both this article and the application described within it.

References

- [1] Attali, D. (2020). *shinyjs: Easily improve the user experience of your Shiny apps in seconds*. R package version 2.0.0.
- [2] Attali, D., & Edwards, T. (2020). *shinyalert: Easily create pretty popup messages (modals) in 'Shiny'*. R package version 2.0.0.
- [3] Bailey, E. (2015). *shinyBS: Twitter bootstrap components for Shiny*. R package version 0.61.
- [4] Bernardi, R. A., Baca, A. V., Landers, K. S., & Witek, M. B. (2008). Methods of cheating and deterrents to classroom cheating: An international study. *Ethics & Behavior*, 18(4), 373-391.
- [5] Bonham, S., Beichner, R., & Deardorff, D. (2001). Online homework: Does it make a difference?. *The Physics Teacher*, 39(5), 293-296.
- [6] Chang, W., Cheng, J., Allaire, J.J., Sievert, C., Schloerke, B., Xie, Y., Allen, J., McPherson, J., Dipert, A., & Borges, B. (2021). *shiny: Web application framework for R*. R package version 1.6.0.
- [7] Chirumamilla, A., Sindre, G., & Nguyen-Duc, A. (2020). Cheating in e-exams and paper exams: the perceptions of engineering students and teachers in Norway. *Assessment & Evaluation in Higher Education*, 45(7), 940-957.
- [8] Fox, J., Venables, B., Damico, A., & Salverda, A.P. (2020). *english: Translate integers into English*. R package version 1.2-5.
- [9] Gagolewski, M., & others (2020). *stringi: Character string processing facilities*. R package version 1.5.3.
- [10] Halcrow, C., & Dunnigan, G. (2012). Online homework in calculus I: Friend or foe?. *PRIMUS*, 22(8), 664-682.
- [11] Harsy, A. & Hoofnagle, A. (2020). Comparing mastery-based testing with traditional testing in Calculus II. *International Journal for the Scholarship of Teaching and Learning*, 14(2).
- [12] Karlsson, A. (2007). Scientific Workplace 5.5 and LyX 1.4.2. *Journal of Statistical Software*, 17, 1-11.
- [13] Lunsford, M. L., & Pendergrass, M. (2016). Making online homework work. *PRIMUS*, 26(6), 531-544.
- [14] Magalhães, P., Ferreira, D., Cunha, J., & Rosário, P. (2020). Online vs traditional homework: A systematic review on the benefits to students' performance. *Computers & Education*, 152, 103869.
- [15] Manoharan, S. (2016). Personalized assessment as a means to mitigate plagiarism. *IEEE Transactions on Education*, 60(2), 112-119.
- [16] Manoharan, S. (2019). Cheat-resistant multiple-choice examinations using personalization. *Computers & Education*, 130, 139-151.
- [17] Mendicino, M., Razzaq, L., & Heffernan, N. T. (2009). A comparison of traditional homework to computer-supported homework. *Journal of Research on Technology in Education*, 41(3), 331-359.
- [18] R Core Team (2021). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria.
- [19] Rodriguez Jr, R. P., & Smith, L. M. (2016). Using an online homework management system in tax accounting: does it advance learning?. *International Journal of Learning Technology*, 11(1),

44-65.

- [20] Rogers, C. F. (2006). Faculty perceptions about e-cheating during online testing. *Journal of Computing Sciences in Colleges*, 22(2), 206-212.
- [21] Wickham, H. (2019). *stringr: Simple, consistent wrappers for common string operations*. R package version 1.4.0.
- [22] Wickham, H., & Hester, J. (2020). *readr: Read rectangular text data*. R package version 1.4.0.
- [23] Wickham, H., François, R., Henry, L., & Müller, K. (2021). *dplyr: A grammar of data manipulation*. R package version 1.0.5.