Challenges and Innovations in Statistics Education
Edited by Péter Kovács. University of Szeged, January 2018
ISBN 978-963-306-575-4
Proceedings of Challenges and Innovations in Statistics Education Multiplier Conference of ProCivicStat

PERSONALIZED EXAMS IN PROBABILITY AND STATISTICS

Gallusz Abaligeti and Dániel Kehl
80 Rákóczi str, Pécs, 7622, Hungary
kehld@ktk.pte.hu

In this paper we introduce an example of using free open-source software solutions to create and grade personalized exams in probability and statistics. The statistical computing software R and the exams package is used together with Moodle, leading open-source learning management system. At the moment we use spreadsheets to store the text of the exercises and choose randomly amongst those. As a result our students face different stories (text), different numbers and potentially different questions but very similar structures. Results of the midterms on average are very similar to the previous paper and pencil results and students like the immediate feedback.

INTRODUCTON

At the University of Pécs, Faculty of Business and Economics BA students have to take two semesters of introductory statistics. Covered materials include basic probability calculations, descriptive statistics, inferential statistics, measures of association, correlation and regression analysis. From year to year the number of enrolled students is higher and higher, going over 400 recently. The computer lab can sit approximately 50-60 students both for practicing, midterms and finals. The two midterm tests focus on the ability of computing different statistical measures, applying methods and hence can be automated. The final focuses more on explanation of results and more comprehensive tasks so it is a mixture: paper and pencil answers based on computer aided calculations, but emphasis is on explaining and interpreting results. The high number of students and low number of seats at the computer lab result in a great number of groups in case of a midterm which means a lot of different versions of the same exercises. In line with the effort of our University to create a student-friendly learning environment we also offer midterm retakes, meaning even more exercises are needed. One year ago we decided to personalize the midterms so all students have their own tasks, leaving less chance of cheating or copying each other's solutions.

Preparing and especially grading midterms took a significant amount of working hours before using R (R core team, 2016) and Moodle (Moodle HQ, 2017). Learning the features of the exams package (Zeileis et al., 2014) and developing our solution clearly was a great initial effort but after having a decent number of reusable midterm problems creating, assigning and grading midterms happens almost automatically. Most of the students want to see their papers to understand which exercise was correct and which was incorrect. In case of over 400 students this procedure also took a lot of time. Using the automatic grading system of Moodle students see their solutions and the correction immediately after submitting their answers. They are given some time to go through these after the midterm and they can flag those questions where they see some problems with the correction of their answers (there are always some typos etc.). From the teachers' side it is a lot easier and quicker to go through those flagged questions in Moodle which saves a lot of time. According to our experiences students like the immediate results of their midterms.

In the next section we introduce a detailed example of creating an exercise and creating midterms using the above mentioned open-source software. We do not want to go into details regarding the exams package, there is a documentation (Gruen – Zeileis, 2009) and great tutorials (http://www.r-exams.org/tutorials)

A DETAILED EXAMPLE OF CREATING MIDTERMS

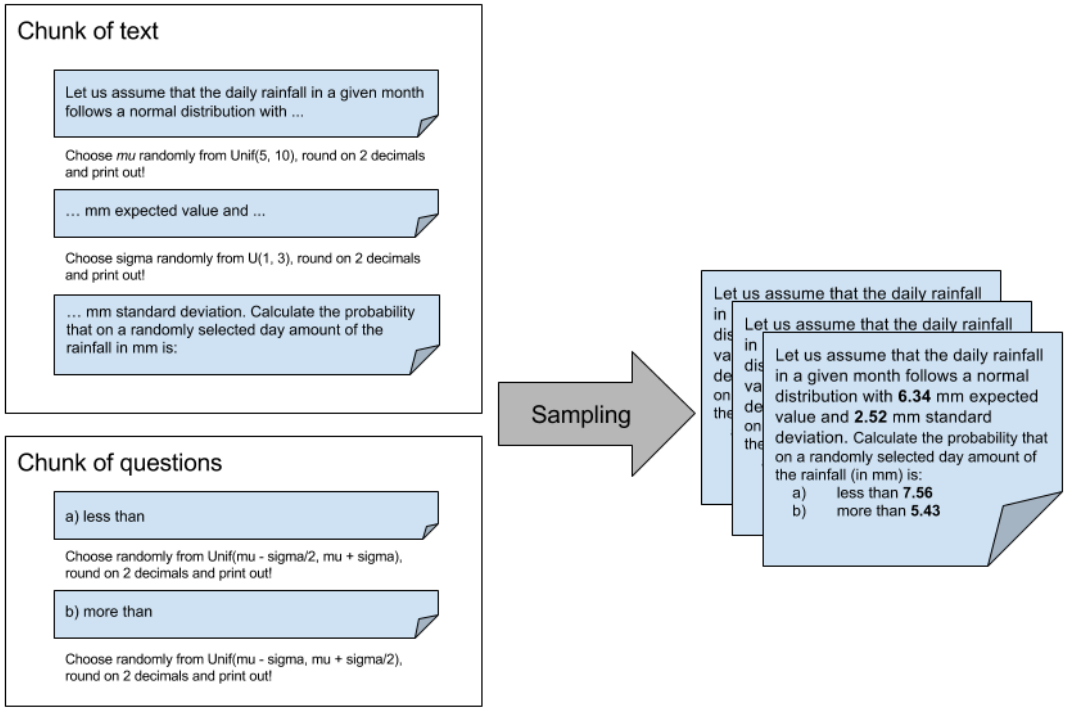
Next we introduce the exams package and how we use it to create midterm questions through an example of tasks related to the normal distribution (introduction to probability is a part of the two semesters). The exams package is working with Rnw files (sweave files) containing a mix of text and R code, allowing the user to execute and embed the results of R computations and graphics within a document. Using built-in functions one can create a variety of outputs, including pdf files and most importantly from our point of view Moodle xml files. As we did not want to use the same text in case of all our students we were looking for a possible way to change the "story"

behind the same exercise. At the moment we store the stories in csv files. As different stories need different random numbers to generate, we also store those together with the stories. To make the procedure more clear we show an example of creating as we call it a meta-exercise (related to the normal distribution). After that we outline a general scheme and workflow.

Creating a meta-exercise

As the normal distribution plays an important role both in probability theory and inferential statistics we chose this topic to introduce how we store the stories and their parameters and create exercises based on these. We want to create exercises where the students have to calculate probabilities of intervals and/or do the inverse cdf calculations based on different stories. For this purpose we have to identify those chunks of text that have to be replaced for each story. One also has to take into account that the solutions should be reasonable, that is we want to avoid result that are too small or too close to one. Also questions themselves should make sense, we want to avoid questions like what is the probability that the IQ of a randomly chosen student is between 101 and 101.5. Avoiding these type of problems needs cautious planning of generating random numbers.

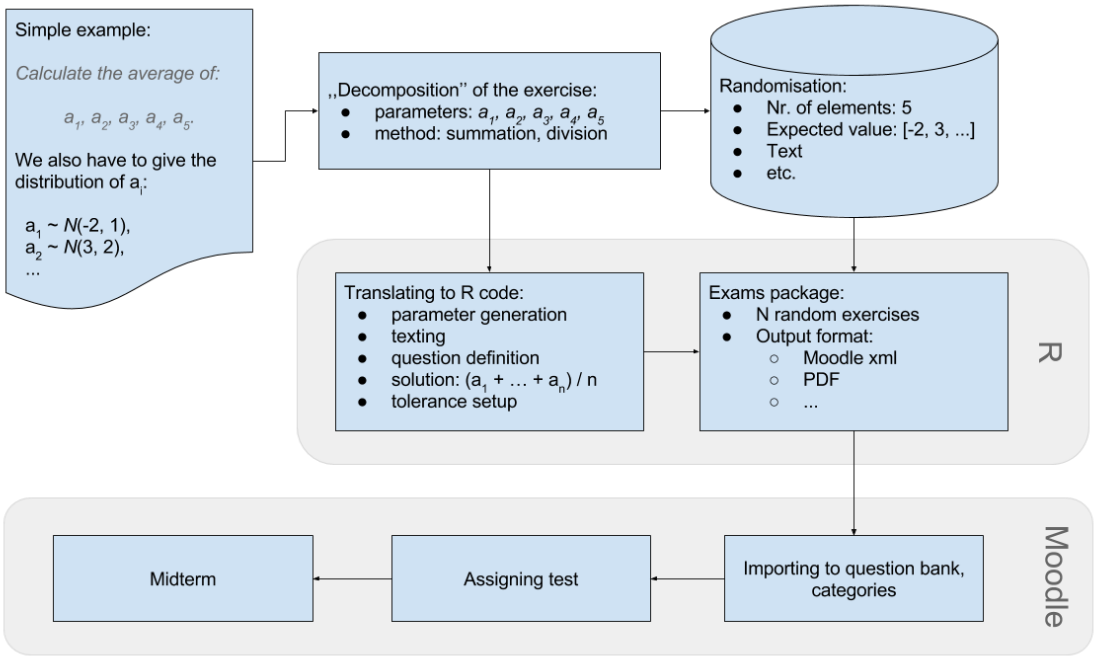
In case of the normal distribution meta-exercise we use 4 text chunks, one is containing the actual story of the exercise (e.g. Let us assume that the daily rainfall in a given month follows a normal distribution with *mumumu* mm expected value and *sigma* mm standard deviation. Calculate the probability, that on a randomly selected day the rainfall is…). Chunks 2-4 contain text in connection with questions for the inverse cdf (e.g. What is the amount of rainfall where the probability is *ppp* that on a given day there is more rain?). In the previous examples the italic values denote parameters that are random, hence all generated problems are going to be different. These parameters depend on the story. Expected value of daily rainfall might vary somewhere let's say between 5-10 mms, but other stories require different expected values. To generate different problems for students we use parameters *mu_from*, *mu_to* and *mu_by* for example for the expected value (e,g. *mu_from* would be 5, *mu_to* 10 and *mu_by* 0.5) In case of employee wages, parameters (expected value and standard deviation) can be in the range of hundreds of thousands (in forint). Another parameter is hence the number of significant digits that is what rounding should we use to make the example nice and meaningful. The last parameters for this meta-exercise is a probability, related to the inverse question, also using the "from, to, by" notation.



Creating a midterm

After creating .Rnw files or meta-exercises for the different topics that we want to include in a midterm we generate as many different tasks as needed and then set up the midterm in Moodle. Setting up the midterm consists of putting all the needed Rnw files in a vector and calling the *exams2moodle* function in R. There are many other supported formats (first of all pdf where there is an option to use custom LaTeX files as templates for the desired outlook of the result) in the package as well. The function creates xml files ready to read them into the Moodle question bank of the given course. Assigning each of the problems to students happens in Moodle. As the solutions (and possibly explanation of the solution) are embedded in the xml file, Moodle is able to grade the exercises automatically and also give a (textual) feedback.

As a summary of creating examples in R and the midterms in Moodle see the following general process:



CONCLUSION

As a result our students have to answer personalized exercises on the midterms which leaves a lot less opportunity for working together during the exam. Results after introducing this system are very similar to previous years. Although the population of student is clearly not the same it makes sense to compare the students as they are similar. Students like the immediate feedback of the system so they don't have to wait days until they get their results. In theory this system gives an opportunity to create unlimited number of practice problems but we did not implement such a system yet. The preparation, setting up the package and clearly creating general enough questions was a big effort but on the other hand now lecturers are able to save a lot of time and develop teaching material etc. instead of grading. We recommend our colleagues who are teaching similar courses to experiment with the exams package and automated creation of exams at least as part of the assessment.

REFERENCES

Gruen, B. & Zeileis A. (2009). Automatic Generation of Exams in R. *Journal of Statistical Software*, *29*(10), 1-14.

Moodle HQ (2017). *The Moodle project.* *https://moodle.org*.

R Core Team (2016). *R: A language and environment for statistical computing*. Vienna: R Foundation for Statistical Computing. *https://www.R-project.org/*.

R Tutorials. http://www.r-exams.org/tutorials/

Zeileis, A., Umlauf N. & Leisch, F. (2014). Flexible Generation of E-Learning Exams in R: Moodle Quizzes, OLAT Assessments, and Beyond. *Journal of Statistical Software, 58*(1), 1-36.