INDIVIDUALISED PROJECT ASSESSMENTS FOR STATISTICS COURSES - THE BEST OF BOTH WORLDS?

Robert Grant and Ahmed Younis
Faculty of Health, Social Care and Education
St George's, University of London, UK
robert.grant@sgul.kingston.ac.uk

Examinations in statistics have been criticized for failing to assess analytical thinking and practical problem-solving skills. Project-based assessment is a widely used alternative, but detection of plagiarism is a concern as students should arrive at the same results. Definition of plagiarism is also difficult; sharing ideas on methods is a positive learning experience, while sharing results and computer output is not. Creating a different dataset for each student can resolve these problems but requires automation to be feasible. We describe the experiences gained from programming a general algorithm for this in R and SPSS and piloting in two years of postgraduate healthcare research methods students. There is potential to introduce unfairness if the requirements of analysis, such as post-hoc testing, are not identical in all datasets. Our algorithm creates multiple datasets that are constrained to differ enough to be identifiable, while also sharing exactly the same analytical requirements.

INTRODUCTION

The assessment of statistics courses in higher education typically requires of the lecturer a choice between unseen examinations and project-based assessment. Examinations in statistics have been criticised for testing recall rather than deep understanding or critical thinking (Aliaga et al., 2012; Garfield, 1994). Only some analytical methods can be asked of a student in a short time scale without a computer, for example finding quartiles or conducting a chi-squared test. Even if examination takes place in a computer room, it is difficult to assess practical skills such as data management, using software or communicating results in the timescale.

On the other hand, project-based assessment, where the student is given data (or asked to collect it) to be analysed and reported in writing after a period of some days or weeks, allows more scope for independent critical thinking, project planning and use of software, but places a greater burden on markers and introduces problems around plagiarism. The definition and detection of plagiarism in projects is not clear-cut, because with the same data and the same tools at their disposal, one expects students to deliver nearly identical reports. Even the definition is difficult, unless any discussion or co-operation among students is prohibited, because it is to be expected that students will share tips on relevant books or software. Indeed, co-operating in learning software and analysis methods is likely to be beneficial for their learning (Aliaga et al., 2012; Gelman & Nolan, 2002). Most statistics and research methods courses that the authors are aware of do not go so far as explicitly to prohibit any co-operation in project-based assessment.

Giving each student their own data file avoids uncertainty around plagiarism but at the cost of increasing the burden of setting questions and making model answers. It is generally acknowledged that this approach is only feasible if it can be automated (Bidgood, Hunt, & Jolliffe, 2010). It is also important to achieve fairness under an automated system; if the different data sets lead to one student having a significant test result and another non-significant, there will be an unfair difference in the extent of further analysis (for example, post hoc tests) or interpretation (for example, significant interaction terms) required of them.

GENERIC METHOD AND ALGORITHM

We propose a general method to create multiple data files, test their suitability, and generate model answers. This can be implemented in any programmable statistical software such as SPSS, R, Stata or SAS. The resulting files can be supplied to students along with an assignment for a research methods or statistics course, requiring them to analyse and interpret the data and write a short report. Particular attention should be paid to features in the data that will detect copying of results rather than methods.

In K. Makar, B. de Sousa, & R. Gould (Eds.), *Sustainability in statistics education. Proceedings of the Ninth International Conference on Teaching Statistics (ICOTS9, July, 2014), Flagstaff, Arizona, USA.* Voorburg, The Netherlands: International Statistical Institute. iase-web.org

The data files we create need to have the following features:

- Differences from every other file, which are minimal but are detectable in statistics and graphs
- No qualitative differences in interpretation, i.e. a change from significant to non-significant test results or from a positive to negative direction of effect or correlation
- Differences between data files should not be predictable

These multiple datasets can be created in various ways: individual data in an original data file can be perturbed by adding random numbers, small data files can be sampled from a large one, data in each variable in an original data file can be shuffled or resampled within quantiles, or new data can be imputed from a conditional distribution, achieved by regressing the variable of interest on all others. Of these options, the first two are simple to achieve and hard to detect; there is no advantage to the third and fourth but we mention them here as technical alternatives which may yet prove useful in some settings. Our proposed approach can be described as the following detailed algorithm:

1. Open the original data file; choose a set of variables to be changed (which could be all of them).
2. Start a loop to run *n* times and produce a new data file at each iteration. The value of *n* should be greater than the number of students; some of the resulting *n* data files will be rejected later.
3. If perturbation is desired, add a random number (continuous or discrete as required) to each variable to be changed, and round the result if necessary to achieve the same level of precision as the original data. Categorical data can be derived by defining a latent continuous variable with thresholds chosen to achieve the correct frequencies, and perturbing this latent variable instead.
4. If sampling of a smaller data file is desired, randomly select the sample and delete the other data.
5. Run the statistical procedures that you want the students to do; save the output and store any results affecting the variables which were changed in memory.
6. Check p-values to make sure they have the required (non-)significance, and effect sizes and correlations to make sure they are positive or negative as required. Reject the new data if these criteria are not achieved and return to the start of the loop at (2) above.
7. Otherwise, store the results and return to the start of the loop at (2) above.
8. Revisit the results stored in memory for each new file; compare the descriptive and inferential statistics and store a *n*x*n* matrix which holds in element [*i*,*j*] the value 1 if *i* and *j* are compatible (their statistics differ to the required extent) and value 0 otherwise. If they are not compatible, then the differences between *i* and *j* may not be visible to markers, and one or both of those files must be rejected. The question of which to reject is not necessarily simple. The *n*x*n* matrix can be viewed as an adjacency matrix for a graph, and our task is to find a clique with at least as many vertices as there are students, for which algorithms exist (Csardi, 2013; Konc & Janezic, 2007; Konc, 2012).
9. Export the output for each retained data set in a format such as PDF that is accessible to markers.

In the authors' experience, only a small proportion of generated files will be retained, so it is sensible to create many more in the loop than are ultimately required, perhaps 20-100 times the number of students. The proportion retained will be affected by the requirements in step (6) and the compatibility criteria in step (8). A sensible starting point is to avoid changing any variable with statistics that are close to the threshold of acceptance for step (6), for example p-values close to 0.05. Likewise, the compatibility criteria should start with as few requirements as possible, perhaps that at least one of the statistics is different from all other retained data files at one decimal place, and can be increased from there as necessary. The process detailed above, although superficially complex, will run very quickly on modern computers, while having to re-run the algorithm to create a further batch of files and check they are different to those already made will require great

care and attention. In this, we are guided by the sage advice of R software developer Uwe Ligges: "RAM [computer memory] is cheap, and thinking hurts" (Ligges, Begum, & Burns, 2007).

If the required analyses are computer-intensive, such as multilevel models or Markov chain Monte Carlo, they will slow down the process. One efficient approach might be to make minimal changes to variables involved in the complex analyses, and to evaluate them out not on all n data sets but only on the retained ones as a final check along the lines of step (6).

The analyses should include obvious wrong choices that students might make (for example, independent t-tests instead of paired t-tests). This will allow markers to deduct some points for making the wrong choice but give others for then going on to report and interpret the results adequately. However, there is no need for these wrong answers to form part of the checks in steps (8) and (10) of the algorithm. Data cleaning may also introduce an element of choice for the student, for example whether to delete, ignore or correct outliers. In such circumstances, it is often the justification which the marker is interested in and not the actual choice of approach, so alternative sets of results could be produced for these choices.

PILOT EVALUATION

We piloted this approach in a postgraduate statistics and research methods module in 2011/12 and 2012/13. Students had backgrounds in healthcare or sports science. Students were taught to use SPSS software. There were nine students in 2011/12 and ten in 2012/13. The approach was the same as detailed in the algorithm above except that perturbation was done manually in 2011/12 and programmed in R in 2012/13. The R code perturbed three continuous variables, compared 63 statistics, produced 200 data sets and retained 14 of them, of which the first ten were used. Two of the variables were serum nutrient concentrations and therefore positively skewed and constrained to positive values; these were log-transformed prior to perturbation to retain their characteristic distributions.

This module had previously been assessed by a project where all students had the same data file. The assessment was for individual students, not small groups. Students were briefed on the assignment in writing and at one of their lectures. The need to use their own data file was emphasised and these files were supplied through the university's virtual learning environment Blackboard. (Blackboard Inc, 2012) They were encouraged to collaborate on using software and choosing analytical methods but warned not to share the results such as e-mailing each other copies of graphs.

To mark the reports, we checked statistics, both descriptive and inferential, against the model answers and we required students to draw scatterplots involving the perturbed variables, which allowed us to check visually that the data matched the model answer for that student.

All assignments in both years were marked from the printed model answers in a single pass by two markers, and no moderation was needed to agree overall marks for each student. Model answers were easy to use, although in 2011/12 we had not anticipated the possibility of students making different decisions about data cleaning at the very outset of their analysis, so in 2012/13 we created four alternative sets of answers for each student under different cleaning choices. All students had evidently used their own datasets, demonstrated by both statistics and graphs.

After the 2012/13 assignments were handed in, students were sent a short feedback form, asking them to choose “strongly agree”, “agree”, “neutral”, “disagree” or “strongly disagree” about whether they felt able to identify their own data file, convert it from Excel to SPSS; whether they understood that each student had their own data file, that each was slightly different, that each required the same effort, that they should not copy results or graphs, that they could collaborate on how to use the software and present the results; whether they felt safe in being able to collaborate on software without fear of plagiarism; and whether they felt comfortable with how this assignment was designed.

Six responses (out of ten students) were received. Each question had all six “strongly agree”, except for questions 4 and 9, where there were five “strongly agree” and one “agree”. Two students added written comments, one was commenting on specific content in the assignment but the other said that they “felt that assignment provided good, clear and unambiguous instructions and accompanying information that made completing the assignment straightforward”.

DISCUSSION

A project-based assessment for research methods and statistics invites a certain amount of co-operation among students. Boundaries for this need to be clear and we suggest that it is worth agreeing among faculty what a course's policy will be on scenarios that do not constitute outright plagiarism. Some of these may be positive experiences for the students in teamwork but it is important not to allow successful students to help others to such an extent as to limit their learning, nor to allow individuals to be disadvantaged by being isolated from the group.

The authors teach introductory courses with SPSS, using drop-down menus and dialog boxes rather than programming. In our experience, although the brightest students express an interest in programming, they do not have time during their course to learn and use it. One scenario we have not encountered but would find particularly troubling would be if a student who was already familiar with statistical software wrote a program to carry out the analyses and shared this. This would be very hard to trace, but would clearly constitute plagiarism, and faculty should consider warning students about this at the outset of any course using our method. As with any project-based assessment conducted under anything less rigorous than examination conditions, students who copy the analytical steps or interpretation of others will likely go undetected. An additional invigilated assessment may be necessary to identify students struggling with choosing methods and interpreting findings.

REFERENCES

Aliaga, M., Cobb, G., Cuff, C., Garfield, J. B., Gould, R., Lock, R., Moore, T., et al. (2012). *GAISE College Report*. American Statistical Association.

Bidgood, P., Hunt, N., & Jolliffe, F. (2010). *Assessment methods in statistical education: An international perspective*. New York: Wiley-Blackwell.

Blackboard Inc. (2012). *Blackboard homepage*. *www.blackboard.com*

Csardi, G. (2013). CRAN - Package igraph. *cran.r-project.org/web/packages/igraph/index.html*

Garfield, J. B. (1994). Beyond testing and grading: Using assessment to improve student learning. *Journal of Statistics Education*, *2*(1).

Gelman, A., & Nolan, D. (2002). *Teaching statistics: A bag of tricks*. Oxford: Oxford University Press.

Konc, J. (2012). *Maximum clique algorithm*. *www.sicmm.org/~konc/maxclique/*

Konc, J., & Janezic, D. (2007). An improved branch and bound algorithm for the maximum clique problem. *MATCH Communications in Mathematical and in Computer Chemistry*, *58*, 569-590.

Ligges, U., Begum, F., & Burns, P. (2007). About memory size. R Help mailing list. *r.789695.n4.nabble.com/About-Memory-size-td828355.html*