Automated feedback on the structure of hypothesis tests

Sietske Tacoma[1], Bastiaan Heeren [1,2], Johan Jeuring[1,2] and Paul Drijvers[1]

[1]Utrecht University, Faculty of Science, Utrecht, The Netherlands; s.g.tacoma@uu.nl, bastiaan.heeren@ou.nl, j.t.jeuring@uu.nl, p.drijvers@uu.nl

[2]Open University of the Netherlands, Heerlen, The Netherlands

Understanding the structure of the hypothesis testing procedure is challenging for many first-year university students. In this paper we investigate how providing automated feedback in an Intelligent Tutoring System can help students in an introductory university statistics course. Students in an experimental group (N=154) received elaborate feedback on errors in the structure of hypothesis tests in six homework tasks, while students in a control group (N=145) received verification feedback only. Effects of feedback type on student behavior were measured by comparing the number of tasks tried, the number of tasks solved and the number of errors in the structure of hypothesis tests between groups. Results show that the elaborate feedback did stimulate students to solve more tasks and make fewer errors than verification feedback only. This suggests that the elaborate feedback contributes to students' understanding of the structure of the hypothesis testing procedure.

Keywords: Intelligent tutoring systems, domain reasoner, hypothesis testing, statistics education.

Introduction

Hypothesis testing is widely used in scientific research and is therefore covered in most introductory statistics courses in higher education (Carver et al., 2016). This topic is challenging for many students, due to the large number of complex concepts involved (Castro Sotos, Vanhoof, Van den Noortgate, & Onghena, 2007). Students struggle to understand the logic of hypothesis testing and the role and interdependence of the concepts in the testing procedure (Vallecillos, 1999). Appropriate feedback might support students in comprehending this structure and understanding the line of argumentation in hypothesis tests (Garfield et al., 2008). Since group sizes in introductory statistics courses are often large, automated feedback in an online system seems a promising direction.

Carrying out a hypothesis test requires several steps, which have to be performed in a particular order. To address the structure of hypothesis tests, feedback should address all aspects of a solution: not only the content of individual steps, but also the relations between steps. An Intelligent Tutoring System (ITS) can provide such sophisticated feedback on the level of steps and can provide detailed diagnostics of student errors (Nwana, 1990). Such feedback on the step level is generally more effective than feedback on the level of complete solutions (VanLehn, 2011).

Although ITSs vary considerably in design, they generally contain the following four components: an expert knowledge module, a student model module, a tutoring module, and a user interface module (Nwana, 1990). Of these four, the expert knowledge module is the most domain-dependent. It contains information about exercises that students can solve in the ITS and about domain knowledge required to solve these exercises (Heeren & Jeuring, 2014), and is therefore also referred to as *domain reasoner*. Two important paradigms for constructing domain reasoners are model-tracing, in which the ITS checks that a student follows the rules of a model solution, and constraint-based modeling, in which the ITS checks whether a student violates constraints. There exist ITSs that support hypothesis testing based on either of these approaches (Kodaganallur, Weitz, & Rosenthal, 2005). We combined the two paradigms in a single ITS supporting hypothesis tests, and in this paper we evaluate the feedback by this ITS. The research question guiding this evaluation is: does the feedback provided by the ITS contribute to student proficiency in producing well-structured hypothesis tests? Before describing the design of the ITS and the study design, we briefly discuss the two paradigms; for a more extensive overview, see the work of Kodaganallur and colleagues (2005).

Model-tracing

The model-tracing paradigm concentrates on the solution process. The domain reasoner contains a set of expert rules: rules that an expert would apply to construct solutions to exercises in the domain. It can also contain buggy rules: incorrect rules that arise from incorrect domain knowledge. Finally, the domain reasoner contains a model tracer that can identify which expert and buggy rules a student has applied to arrive at a (partial) solution. After this identification, several actions can take place. If a buggy rule is recognized, a diagnosis of the student's error can be given. Otherwise, a hint for a next step can be provided, or just feedback that the current solution path is correct.

Constraint-based modeling

Constraint-based modeling concentrates on partial solutions, rather than on the solution process. The underlying idea is that incorrect knowledge emerges as inconsistencies in students' partial solutions (Mitrovic, Mayo, Suraweera, & Martin, 2001). Domain knowledge is represented as a set of constraints, consisting of a relevance condition and a satisfaction condition. For instance, in the domain of hypothesis testing a relevance condition might be that the partial solution contains an alternative hypothesis and a rejection region, and the corresponding satisfaction condition that the position of the rejection region (left side, right side or two sides) matches with the sign of the alternative hypothesis. Errors in student solutions emerge as violated constraints, that is, as constraints for which the relevance condition is satisfied, but the satisfaction condition is not. If a student's solution does not violate any constraint, it is diagnosed as a correct (partial) solution.

Methods

Design of the domain reasoner

The technical design of the domain reasoner evaluated in this study is based on the Ideas framework (Heeren & Jeuring, 2014), which uses a model-tracing approach to calculate feedback. However, for the domain of hypothesis testing the constraint-based modeling approach was also considered useful, because of its capability to identify inconsistencies in solution structures. Addressing these inconsistencies might contribute to student understanding of the relations between steps, and hence of the structure of hypothesis tests. Therefore, the domain reasoner designed in this study combines elements of the two paradigms. It contains 36 expert rules, 16 buggy rules, and 49 constraints. Every time a student adds a step to a hypothesis testing procedure, for example defines an alternative hypothesis or calculates the value of the test statistic, the domain reasoner checks the student's solution so far. First, all constraints are checked, and if one is violated, the domain reasoner determines whether a buggy rule was applied. If so, a specific feedback message corresponding to this buggy rule is displayed to the student, and otherwise a general message for the violated constraint is reported. For example, a student solution that does not contain an alternative hypothesis, but does contain a rejection region, violates the constraint with relevance condition “the solution contains a rejection region” and satisfaction condition “the solution contains an alternative hypothesis”. The corresponding feedback message addresses the role of the hypotheses: “To which hypotheses does this rejection region correspond? First state hypotheses”.

If no constraints are violated, the domain reasoner tries to identify the rule the student applied to arrive at the current partial solution. A feedback message corresponding to the identified rule is shown to the student, for example: “Your rejection region is correct”. If no rule is identified, the student’s partial solution is marked correct (“This is a correct step”), since no constraints are violated. Besides checking partial solutions, the domain reasoner can also provide hints on what next step to take.

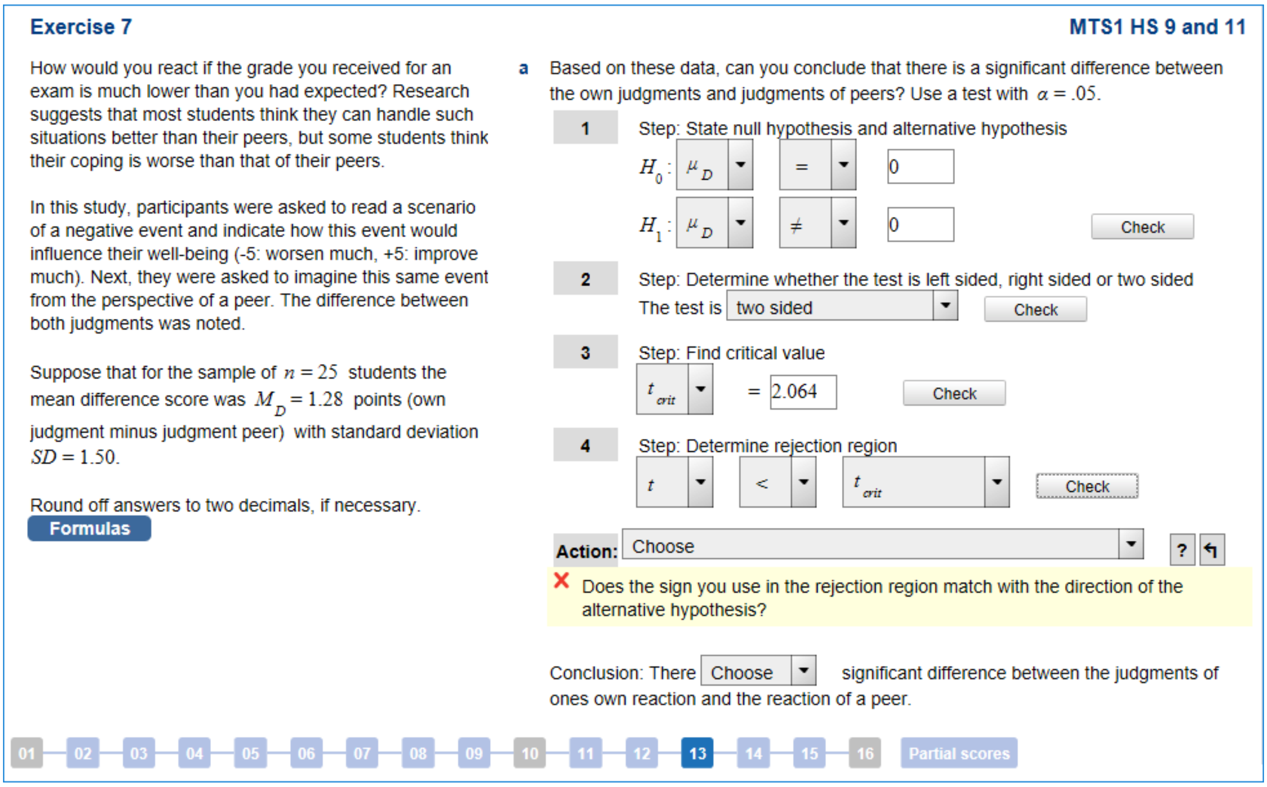
The design of expert rules, buggy rules and constraints was informed by discussions with three teachers of introductory statistics courses in various disciplines about the structure of hypothesis tests and common errors by students. Furthermore, textbooks were consulted. Based on this input, we decided to support two methods for structuring hypothesis tests: the conclusion about the hypotheses can be drawn based on comparison of the test statistic with a critical value, or based on comparison of a p-value with a significance level. In each method, a complete solution should contain four essential steps: (1) state hypotheses, (2) calculate a test statistic, (3) either find a critical value or find a p-value, and (4) draw a conclusion about the hypotheses. Although crucial for the logic of hypothesis testing, stating a significance level was not regarded as an essential step, since it was always specified in the task description. Besides these four essential steps, students could include several other steps in their hypothesis tests, such as a summary of sample statistics and an explicit specification of whether the test was left sided, right sided or two sided.

Participants and study design

The domain reasoner was used in a compulsory Methods and Statistics course for first-year psychology students at Utrecht University. In five weeks of this ten-week course, students received online homework sets consisting of 7 to 13 tasks that were selected by the teachers of the course. These homework sets were designed in the Freudenthal Institute’s Digital Mathematics Environment (DME; see (Drijvers, Boon, Doorman, Bokhove, & Tacoma, 2013), which supports various interaction types, such as formula input and multiple choice tasks. To enable intelligent feedback on hypothesis tests, a connection was set up between the DME and the domain reasoner.

The third, fourth and fifth homework set concerned hypothesis testing. Each of these three homework sets contained two tasks that specifically challenged the students’ proficiency in carrying out hypothesis tests, by asking the students to select steps from a drop-down menu and to complete these steps. An example is shown in Figure 1: after selecting a step from the drop-down menu called “Action”, it appears as next step in the step construction area. Next, the student can complete the step by filling in the answer boxes and use the check button to check the hypothesis test procedure so far. After finishing the construction of the hypothesis test, the student should state the overall conclusion in the final conclusion area, below the drop-down menu with steps.

To evaluate the effects of the domain reasoner feedback, students in the course were randomly divided into an experimental and a control group. This allowed for a comparison of student work between both groups, and differences in student behavior and learning effects could be ascribed to the only difference between both groups, the domain reasoner feedback (Cohen, Manion, & Morrison, 2011). From the 310 students in the experimental group 226 students worked on the hypothesis testing tasks, of which 154 gave consent for the use of their work in this study. From the 309 students in the control group 216 students worked on the tasks, of which 145 gave consent. The participants were on average 19.3 years old (*SD* = 1.7 years) and 77% were female. To reduce the Hawthorne effect, i.e. the effect that students might behave differently because they were part of an experiment (Cohen et al., 2011), students were not told about the different conditions and whether they were in the experimental or in the control condition.



Exercise 7

MTS1 HS 9 and 11

How would you react if the grade you received for an exam is much lower than you had expected? Research suggests that most students think they can handle such situations better than their peers, but some students think their coping is worse than that of their peers.

In this study, participants were asked to read a scenario of a negative event and indicate how this event would influence their well-being (-5: worsen much, +5: improve much). Next, they were asked to imagine this same event from the perspective of a peer. The difference between both judgments was noted.

Suppose that for the sample of $n = 25$ students the mean difference score was $M_D = 1.28$ points (own judgment minus judgment peer) with standard deviation $SD = 1.50$.

Round off answers to two decimals, if necessary.

a Based on these data, can you conclude that there is a significant difference between the own judgments and judgments of peers? Use a test with $\alpha = .05$.

1 Step: State null hypothesis and alternative hypothesis

H_0 : μ_D = 0

H_1 : μ_D ≠ 0

2 Step: Determine whether the test is left sided, right sided or two sided

The test is two sided

3 Step: Find critical value

t_{crit} = 2.064

4 Step: Determine rejection region

t < t_{crit}

Action: Choose

✗ Does the sign you use in the rejection region match with the direction of the alternative hypothesis?

Conclusion: There Choose significant difference between the judgments of ones own reaction and the reaction of a peer.

Figure 1: Digital Mathematics Environment displaying one of the hypothesis testing tasks (translated)

The homework sets were equal in both groups, except for the six tasks in which students constructed complete hypothesis tests by selecting and filling in steps. Students in the experimental group received feedback from the domain reasoner, whereas students in the control group only received verification feedback. Hence students in the experimental group received elaborate feedback on errors in the structure of their hypothesis tests, while students in the control group only received feedback on the correctness of their current step, irrespective of previous steps. As a consequence, for solving a task completely, students in the experimental group needed to include all four essential steps, since otherwise one or more constraints would be violated. For students in the control group, tasks were already marked as solved once they contained a correct conclusion about the null hypothesis. A final difference between the two groups was the hint functionality of the domain reasoner, but in this study we did not take the students' use of hints into account.

Data and data analysis

Data for this study consisted of the students' work on the six hypothesis testing tasks in the DME, that is, all their attempts at constructing steps and drawing final conclusions in these tasks. After exporting log files with the students' attempts from the DME, work from students who did not give consent was deleted and all other attempts were anonymized.

Two measures served as indicators for feedback effectiveness: skipping vs. trying behavior and solving vs. failing behavior (Narciss et al., 2014). Skipping vs. trying concerned the number of students who skipped constructing steps and only filled in a final conclusion versus the number of students who tried constructing steps. This can be regarded as a measure of feedback effectiveness, since students who filled in a final answer without constructing steps apparently did not perceive the feedback on the steps as helpful. Solving vs. failing concerned the number of students who produced complete hypothesis tests versus the number of students who failed to produce complete tests. Since feedback was more elaborate in the experimental group than in the control group, we expected the proportion of solving students to be higher in the experimental group.

To find the number of students who tried constructing steps, for each task the number of students who made at least one step using the drop-down menu was counted. The total number of trying students over all tasks was calculated as the sum of these counts. Next, the total number of all trying and skipping students was found by summing the number of students who used the final answer boxes over all tasks. For both groups the proportion of trying students was found by dividing the number of trying students by the number of trying and skipping students. For solving vs. failing, we considered a task solved if a student had correctly included all four essential steps in the solution. The number of solving students was calculated over all tasks, and divided by the total number of trying students, to find the proportion of solving students. Chi-squared tests were used to evaluate whether the two proportions (trying as proportion of all students and solving as proportion of trying students) were significantly larger in the experimental group than in the control group.

The second part of data analysis concerned errors in the structure of hypothesis tests. Structure errors could be missing information, such as stating a rejection region before stating hypotheses, or inconsistent information, such as a rejection region that did not match with the alternative hypothesis. For the first few tasks, we expected an equal number of structure errors in both groups, since students had not received feedback on their errors yet. In later tasks we expected fewer structure errors in the experimental group than in the control group, and we especially expected that fewer students would make the same structure error in multiple tasks in the experimental group. For each task, the proportion of students who made at least one structure error was calculated and the proportions were compared using Chi-squared tests. Next, for each structure error that the domain reasoner could diagnose, the number of students who made it in only one task and the number of students who made it in multiple tasks were counted and these counts were summed. Again, a Chi-squared test was used to compare the proportions between groups.

Results

Table 1 contains the proportion of students who tried using the stepwise structure in the experimental and the control group, and the proportion of students who completely solved the tasks, as proportion of all students who tried the task. As can be seen in the table, students in both groups tried to construct steps almost 80% of the times, and there is no significant difference between these proportions in both groups: $c^2(1, N = 1558) = 0.09, p = .759$. This implies that the domain reasoner feedback did not impact the students' decision to try or skip using the stepwise structure to solve tasks.

	Experimental group	Control group	c^2	*df*	*p*-value	Effect size
Total number of final solutions in six tasks	802	756				
Proportion of students who tried (vs. skipped)	0.77	0.78	0.09	1	.759	
Proportion of students who solved (vs. failed)	0.52	0.40	19.18	1	< .001	0.13

Table 1: Proportions of students who tried vs. skipped, and who solved vs. failed, in experimental and control group, over all six tasks

The proportion of students who included all essential steps in their solutions does differ significantly between the groups: in the experimental group, this proportion was 0.52 and in the control group it was 0.40 ($c^2(1, N = 1208) = 19.18, p < .001$). The effect is regarded small ($\varphi = 0.13$). This result suggests that the feedback from the domain reasoner did support students in producing more complete hypothesis tests. Although positive, this result is not very surprising, since the domain reasoner feedback explicitly notified students of missing steps. To find out how much the domain reasoner really contributed to the students' proficiency in producing well-structured hypothesis tests, we also considered the errors students made in the structure of their hypothesis tests.

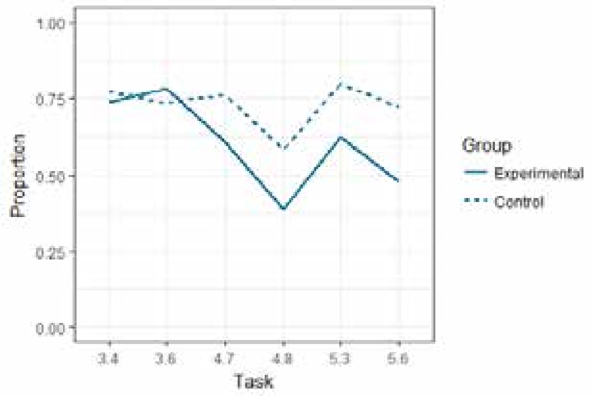


Figure 2: Proportion of students with errors in solution structure in both groups

Task	*N*	c^2	*df*	*p*-value	φ
3.4	297	0.345	1	.557	-
3.6	216	0.501	1	.479	-
4.7	207	5.086	1	.024	0.16
4.8	194	6.665	1	.010	0.19
5.3	158	4.936	1	.026	0.18
5.6	136	7.663	1	.006	0.24

Table 2: Errors in solution structure in both groups for each of the six tasks

Figure 2 shows the proportion of students with errors in their solution structure for both groups, compared to the number of students who attempted the tasks. The total number of students who attempted each task is given in Table 2. The graph in Figure 2 shows that for the first two tasks, 3.4 and 3.6, the proportion of students who made structure errors were quite similar in both groups, and above 0.70. This similarity between groups was to be expected, since students just started working with the feedback. For the latter tasks, however, the proportion of students who made structure errors decreased strongly in the experimental group, and remained higher in the control group. Table 2 displays the results of a series of Chi-squared tests which confirmed this finding: in the first two tasks, the number of students who made structure errors did not differ significantly between groups, whereas in the latter four tasks, in the experimental group significantly fewer students made errors in solution structure than in the control group. This suggests that the feedback contributed to the students' proficiency in producing hypothesis tests without making structure errors. Considering the numbers of students who made the various structure errors just once and the number of students who repeated the same structure error in multiple tasks corroborates this finding: a Chi-squared test on these numbers yielded $c^2(1, N = 1053) = 38.29$, $p < .001$, $\varphi = 0.19$, indicating that students in the experimental group repeated the same structure error in multiple tasks significantly less often than students in the control group. This confirms that the domain reasoner feedback was effective in reducing students' mistakes with respect to the structure of hypothesis tests.

Conclusion and discussion

In this paper we have evaluated the influence of feedback from an Intelligent Tutoring System that supports hypothesis testing on student behavior in six hypothesis testing tasks. Our aim was to evaluate whether this feedback contributes to student proficiency in producing well-structured hypothesis tests.

Students who received domain reasoner feedback did not use the stepwise structure for producing hypothesis tests in more tasks than students who received verification feedback alone, but they did produce more solutions that included all four essential steps in hypothesis testing. Hence, as a consequence of the domain reasoner feedback, students in the experimental group had more opportunities for valuable practice on these essential steps (Narciss et al., 2014). Students in the control group tended to use the steps more as "isolated checkers", rather than as building blocks for producing complete hypothesis tests. By only selecting the steps they wanted to check and omitting the other steps from their solutions, these students missed out on opportunities to practice with the line of argumentation that is important for understanding hypothesis testing (Garfield et al., 2008).

Not only did the domain reasoner feedback encourage students to produce more complete hypothesis tests, it also supported students in doing this more independently. In the first two tasks the number of errors in solution structure was similar between groups, and hence students in both groups relied equally on the feedback to help them correct their errors. In the final four tasks, however, students who received domain reasoner feedback made significantly fewer structure errors than students in the control group. This indicates that domain reasoner feedback did more effectively support students in resolving their misunderstandings than verification feedback alone, which is in line with earlier findings that elaborate feedback is more effective than verification feedback (van der Kleij, Feskens, & Eggen, 2015).

In this paper, we just examined direct effects of the domain reasoner feedback on student behavior; we have only considered tasks in which domain reasoner feedback was provided and no subsequent work. Future work will focus on longer term learning effects of the domain reasoner feedback; do we see the same effects for other hypothesis testing tasks in which students do not receive feedback on their solution structure? Results from the study discussed in this paper are promising; students demonstrate proficiency in structuring their hypothesis tests and hence seem to have gained understanding of the roles of the different steps in this complex procedure.

Acknowledgments

We thank teachers Jeltje Wassenberg-Severijnen and Corine Geurts and software developers Noeri Huisman, Martijn Fleuren, Peter Boon and Wim van Velthoven for the valuable collaboration.

References

Carver, R., Everson, M., Gabrosek, J., Horton, N., Lock, R., Mocko, M., . . . Wood, B. (2016). Guidelines for Assessment and Instruction in Statistics Education College Report 2016. *American Statistical Association.* http://www.amstat.org/education/gaise

Castro Sotos, A. E., Vanhoof, S., Van den Noortgate, W., & Onghena, P. (2007). Students' misconceptions of statistical inference: A review of the empirical evidence from research on statistics education. *Educational Research Review, 2*(2), 98–113. https://doi.org/10.1016/j.edurev.2007.04.001

Cohen, L., Manion, L., & Morrison, K. (2011). *Research methods in education* (7th ed.). New York: Routledge.

Drijvers, P., Boon, P., Doorman, M., Bokhove, C., & Tacoma, S. (2013). Digital design: RME principles for designing online tasks. In C. Margolinas (Ed.), *Proceedings of ICMI Study 22 Task Design in Mathematics Education* (pp. 55–62). Clermont-Ferrand, France: ICMI.

Garfield, J. B., Ben-Zvi, D., Chance, B., Medina, E., Roseth, C., & Zieffler, A. (2008). Learning to reason about statistical inference. *Developing students' statistical reasoning* (pp. 261–288) Springer, Dordrecht.

Heeren, B., & Jeuring, J. (2014). Feedback services for stepwise exercises. *Science of Computer Programming, 88*, 110–129. https://doi.org/10.1016/j.scico.2014.02.021

Kodaganallur, V., Weitz, R. R., & Rosenthal, D. (2005). A comparison of model-tracing and constraint-based intelligent tutoring paradigms. *International Journal of Artificial Intelligence in Education, 15*(2), 117–144.

Mitrovic, A., Mayo, M., Suraweera, P., & Martin, B. (2001). Constraint-based tutors: A success story. In L. Monostori, J. Vancza, M. Ali (Eds.), *Engineering of Intelligent Systems. IEA/AIE 2001* (pp. 931–940). Springer, Berlin, Heidelberg.

Narciss, S., Sosnovsky, S., Schnaubert, L., Andrès, E., Eichelmann, A., Goguadze, G., & Melis, E. (2014). Exploring feedback and student characteristics relevant for personalizing feedback strategies. *Computers & Education, 71*, 56–76. https://doi.org/10.1016/j.compedu.2013.09.011

Nwana, H. S. (1990). Intelligent tutoring systems: An overview. *Artificial Intelligence Review, 4*(4), 251–277.

Vallecillos, A. (1999). Some empirical evidence on learning difficulties about testing hypotheses. *Bulletin of the International Statistical Institute: Proceedings of the Fifty-Second Session of the International Statistical Institute, 58*, 201–204.

Van der Kleij, F., Feskens, R., & Eggen, T. (2015). Effects of feedback in a computer-based learning environment on students' learning outcomes. A meta-analysis. *Review of Educational Research, 85*(4), 475–511. https://doi.org/10.3102/0034654314564881

VanLehn, K. (2011). The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems. *Educational Psychologist, 46*(4), 197–221. https://doi.org/10.1080/00461520.2011.611369