USING SELECTED INDICES TO MONITOR CHEATING ON MULTIPLE-CHOICE EXAMS

Larry R. Nelson

Curtin University of Technology, Australia

บทคัดย่อ

บทความนี้อภิปรายวิธีตรวจสอบการโกงการสอบในแบบทดสอบแบบหลายตัวเลือก โดยใช้ดัชนี Harpp-Hogan (H-H) การศึกษาความคงเส้นคงวาของดัชนี H-H จากการทดสอบของศูนย์การสอบ 2 ศูนย์ ผลการศึกษามีข้อเสนอว่า ควรนำดัชนี H-H ไปใช้ด้วยความระมัดระวัง การเปรียบเทียบลักษณะของซอฟต์แวร์ที่ใช้ตรวจสอบการโกงการสอบ รวมทั้งให้ข้อแนะนำสำหรับผู้ที่ต้องการใช้ซอฟต์แวร์ตรวจสอบการโกงการสอบ

USING SELECTED INDICES TO MONITOR CHEATING ON MULTIPLE-CHOICE EXAMS

Larry R. Nelson

Curtin University of Technology, Australia

ABSTRACT

Methods for detecting cheating on multiple-choice tests are discussed, with particular focus on the Harpp-Hogan index. An investigation of the reliability of the H-H index was undertaken in two professional testing environments, with results suggesting the index can only be used with great caution. A comparison is made of the features available in selected software packages, and recommendations made for practitioners.

As the author of the Lertap item and test analysis package (Nelson, 2000[1]), I attempt to respond to modification requests from users as time allows. Early in 2005, the director of a large-scale testing program wrote to ask if Lertap might someday build in support for cheat checking, that is, for detecting the extent to which students in a given test venue may have engaged in answer copying or sharing. The director was familiar with the work of Wesolowsky (2000), and asked if I had seen it.

I had not. Detecting cheating on multiple-choice exams was not something I was familiar with. I obtained a copy of Wesolowsky's (2000) article, and began to adapt Lertap so that it would provide support for users wanting an index of cheating.

The Harpp-Hogan index

Some readers may already be aware of something which quickly became apparent to me: efforts to measure cheating have been going on for a very long time. Frary (1993) reviewed cheating indices dating back to the late 1920s, following their development up to the early 1990s.

Frary himself has worked with colleagues to develop cheating detection indices (Frary, Tideman, & Watts, 1977; Frary & Tideman, 1997), and these are very much still in use today–the *Integrity* system[2] is one software package which features Frary et al.'s detection indices, as well as others.

Wesolowsky (2000) recommended a modification of the Frary *et al.* indices which he suggested "*has a more intuitively appealing form, and exhibits more consistency with respect to certain other constraints*". I was attracted by these comments, and decided to begin Lertap modifications by basing them on Wesolowsky's work.

Wesolowsky's (2000) article also includes references to the work of Harpp and Hogan (1993), and to Harpp, Hogan, & Jennings (1996). The latter article includes an empirical assessment of a descriptive cheating index which Wesolowsky refers to as H-Hstat, and which I will refer to here as the "H-H" index.

To understand the H-H index, consider the responses of any given pair of students who have sat the same multiple-choice exam. It is of course to be expected that some of the responses given by the students will be the same; in fact, if they're top students, they might each return a perfect exam score, in which case all of their item responses will be identical.

[1] See www.lertap.curtin.edu.au

[2] See www.integrity.castlerockresearch.com

But let us consider the more common case: the students will not have perfect papers. They will get some items correct, some wrong, and, for some reason, they may omit a few items, leaving them unanswered.

The H-H index is based on two characteristics of the students' item responses: the number of exact errors in common, EEIC, and the number of different responses, D. The H-H index is expressed as a ratio of these two numbers: H-H = EEIC/D.

Two students are said to have an "exact error in common" when they both select the same distractor to an item, that is, when they choose exactly the same incorrect answer to an item.

Harpp, Hogan, & Jennings (1996) reported on their observation of the H-H index's behavior, tracking it over years of application, reporting they found it to be "*a powerful indicator of copying*". They wrote:

> *Analyses of well over 100 examinations during the past six years have shown that when this number is ~1.0 or higher, there is a powerful indication of cheating. In virtually all cases to date where the exam has ~30 or more questions, has a class average <80% and where the minimum number of EEIC is 6, this parameter has been nearly 100% accurate in finding highly suspicious pairs.*

One would do well to ask how Harpp et al. substantiated their findings. They did so by confirming that the suspicious pairs of students, ones whose H-H index was equal to or greater than one, also deviated "*significantly from the norm of probabilities*", and were found to be have been seated "*in immediate proximity to one another*".

Harpp et al. have not suggested a sampling distribution for their H-H index. When they imply that they have looked at the relationship between the H-H index and "the norm of probabilities", they refer to a method set out in Harpp and Hogan (1993). Briefly, Harpp and Hogan (1993) determined the probability of any pair of students returning a given number of identical responses under the assumption that the pair had not shared or copied item responses. Were this assumption true, we could say that the item responses of the two students are independent of each other – any matches in their item responses may be attributed to chance alone.

Harpp et al. developed a test statistic having a Gaussian distribution, and suggested that those student pairs with a test statistic at or greater than five standard deviations could be said to significantly deviate from the norm. (The probability of obtaining a z-score of 5.0 or more in a normal distribution is about 0.0000003.) In other words, Harpp and Hogan (1993) devised a procedure for testing the hypothesis that the item responses

given by a pair of students were statistically unrelated, or independent of each other; if they were, the value of their test statistic would be less than 5.0, and the hypothesis would be accepted. A test statistic of 5.0 or more would result in the hypothesis being rejected; in this case, we would have strong statistical evidence to suspect that the similarities found in the students' item responses were beyond what we'd expect by chance, pointing to the possibility of cheating.

I was encouraged by these findings, and considered that the first alteration for Lertap would involve getting it to compute the H-H index. In July 2005 a new version of Lertap, "5.5", was released, one which featured the computation of the H-H index. This version was also capable of producing a data file formatted for direct input to Wesolowsky's "*SCheck*" program – this was done so that Lertap users would have recourse to a rigorous statistical procedure for testing the null hypothesis of response independence, in a manner analogous but not identical to that used by Harpp and Hogan (1993).

The new version of Lertap has now been fairly extensively applied in two major testing centers, looking at results from tens of examinations involving several thousand students.

Has the H-H index lived up to its billings? I had high hopes for it as it is extremely easy to compute, requiring little processing time; I also felt it was a statistic reasonably amenable to interpretation. If I could confirm the results summarized in Harpp, Hogan, & Jennings (1996), Lertap would then have a handy cheat detector ready to use without requiring additional computations. I report findings below.

Confirming the Harpp-Hogan index

The directors of both testing centers had previously shared several data sets for use in testing former Lertap versions. The first substantial application I made of Lertap 5.5 was to give it one of larger data sets from test center "A", making sure that it involved an exam which met the basic criteria set out by Harpp, Hogan, & Jennings (1996): at least 30 items, and an average exam score below 80%.

The selected data set involved five hundred students from test venues in several localities, responding to a multiple-choice test having sixty items, with four options per item. The average test score was 44.45 (74%).

Lertap revealed there were just five student pairs with an H-H index at 1.00 or above. These are shown in Table 1[3].

[3] Table data as produced by Lertap 5.51.

Table 1: Center A Data Set 1

Data row	Correct	EEIC	D	H-H index
437 438	34 33	26	1	26.00
158 160	47 47	11	3	3.67
138 143	50 44	10	6	1.67
412 481	54 49	6	5	1.20
214 444	51 51	6	6	1.00

The “Correct” column in the table indicates the number of items a student answered correctly. For the first pair of students, respective exam scores were 34 (57% of 60, the total number of items) and 33 (55%). Of their 60 item responses, the first pair of students had 26 exact errors in common, and, over all sixty items, there was but a single difference in their responses.

Did these five H-H values correspond to student pairs whose test scores failed the response-independence hypothesis, as indexed by Wesolowsky’s *SCheck* program? No: using its default settings, *SCheck* suggested that only the two highest H-H values corresponded to student pairs whose hypothesis could be rejected.

Rejection of the null hypothesis is not a fail-proof process – there is always the possibility of a Type I error, always some chance that a statistically-significant test statistic will be a false positive. *SCheck* goes to great lengths to minimize false positives, as did Harpp and Hogan (1993) when they set their test z-score minimum at 5.0.

To check on the possibility of false positives, we can ask if the two student pairs sat the test at the same venue, and were seated next to each other. If they were not, if they were miles apart when sitting the test, then we’d rather reasonably assume they had no chance whatsoever to cheat, and we’d conclude that both the H-H index and the *SCheck* significance test had returned a false positive.

My colleague at test center A confirmed that the pair with the highest H-H value, 26.00, corresponded to two brothers who sat in adjacent seats at the same venue, but was unable to confirm the other pair due to the unavailability of relevant venue codes.

How do these results compare with those found by Harpp, Hogan, & Jennings (1996)? I have five H-H values at or above their H-H cut-off of 1.00, and have determined that just two of these five H-H values were corroborated by additional evidence, that is, by the rejection of the corresponding response-independence hypothesis, and by finding that one of the two pairs was indeed seated adjacently. We would think that Harpp, Hogan, & Jennings (1996) would have expected the corroborating evidence to support all five cases, given the findings reported in their article.

Despite this discrepancy, I judged the results to be not too bad. I took note of the fact that Harpp, Hogan, & Jennings (1996) were using test items with five options, compared to the four used in test center A. One might suspect that this could serve to increase the EEIC figure as fewer item options means more opportunity to find students selecting the same distractor.

Perhaps, I thought, one should raise the Harpp, Hogan, & Jennings H-H cut-off bar when the number of item options is less than five. Were it increased to 1.50, for example, there would be only one discrepancy between the results from test center A and those seen in Harpp, Hogan, & Jennings (1996).

There was a clear need to look at more data sets. I turned to review results from two selected data sets at test center "B".

The first of these involved an exam with more than three thousand students scattered over a large number of test venues, responding to a multiple-choice test of sixty items, four options per item. The average exam score was 40.80 (68%).

With this many students, there would be almost five million[4] unique student pairs, each pair with its own H-H value.

How many H-H values at or above 1.00 did Lertap find[5]? Forty (40). Of these, twenty-five were 1.00 exactly, followed by seven H-H values close to 1.10, then by the fourteen cases shown below in Table 2.

[4] N(N-1)/2

[5] It took almost 30 minutes to process this data set through Lertap 5.5, and about 22 minutes through *SCheck* (using a late-model laptop with a power-saving processor).

Table 2: Center B Data Set 1

Data row	Correct	EEIC	D	H-H index
587 588	45 43	15	2	7.50
2055 2256	51 52	7	3	2.33
2247 2250	50 50	8	4	2.00
1887 2219	51 53	6	4	1.50
599 2878	47 44	10	8	1.25
285 868	22 28	23	19	1.21
121 1795	53 50	6	5	1.20
1376 1955	49 54	6	5	1.20
1051 2493	21 18	26	22	1.18
29 1299	49 50	7	6	1.17
29 2016	49 50	7	6	1.17
2009 2785	53 47	7	6	1.17
562 1403	47 50	8	7	1.14
1355 1679	39 32	16	14	1.14

Regarding evidence to support a link between these H-H values and possible cheating, the *SCheck* program, operating with its default settings, suggested that the response-independence hypothesis could be rejected only for the student pair with H-H=7.50. Were these two students seated next to each other? Yes, they were. They both had the same family name.

I next asked the *SCheck* program to relax its control for false positives somewhat[6]; after this it rejected the response-independence hypothesis for a total of four student pairs: those shown in Table 2 with H-H values of 7.50, 1.21, and 1.14 (with EEIC=16), and one pair, not shown in the table, with an H-H index of 1.00 (EEIC of 17, D of 17). Of these four pairs, only one sat the test at the same venue, that with H-H=7.50, implying that relaxing *SCheck*'s Type I error rate control has produced three false positives.

Altogether, of the forty student pairs with H-H values at or above 1.00, only two pairs (2) sat the test at the same venue. In Table 2, these two pairs correspond to the first and third entries, to H-H values of 7.50 and 2.00. And of these two, only the first had its response-independence hypothesis rejected by *SCheck*, when running with its default error rate control.

In the case of this large data set, it is apparent that use of the Harpp, Hogan, & Jennings (1996) H-H cut-off of 1.00 was not supported. Nor was my thought that the cut-off should be raised to something like 1.50.

A second data set from test center B was then investigated. It involved an exam with some six hundred students sitting in numerous distinct test venues, responding to a multiple-choice test of seventy items, four options per item. The average score on the exam was 49.85 (71.2%).

Table 3: Center B Data Set 2

Data row	Correct	EEIC	D	H-H index
257 471	60 60	7	5	1.40
32 316	60 62	6	5	1.20
401 443	44 40	19	16	1.19
539 601	60 54	9	8	1.13
123 393	57 59	7	7	1.00

Lertap found five H-H values at or in excess of the Harpp, Hogan, & Jennings (1996) H-H cut-off of 1.00. These are shown above in Table 3.

[6] The Bonferroni cut-off was increased from its default value of 0.01 to 10.00, resulting in a cut-off Zb value of 4.46.

Of these five student pairs, only the pair with H-H=1.19 sat the exam at the same venue. *SCheck*, running in default mode, did not reject the response-independence hypothesis for this pair. However, when *SCheck*'s control for false positives was relaxed[7], the hypothesis was rejected for this pair, and for this pair only.

Downplaying the Harpp-Hogan index

Having had a thorough look at three data sets, *I cannot recommend that the H-H index be relied on to identify student pairs who might be suspected of cheating*. My findings, as reported above, do not support those found in Harpp, Hogan, & Jennings (1996).

Nonetheless I may continue to push data sets through Lertap, looking for patterns in the H-H values it produces. It is apparent that really extreme H-H values, such as the 26.00 and 7.50 figures discussed above, probably indicate cheating. It might also be noted that Harpp, Hogan, & Jennings (1996) suggest caution whenever the exam scores of the paired students are high. Their suggestion is to raise the caution flag whenever test scores are in the 90% and above range. If I brought this down to 80% and above, the lowest three of the five Table 1 pairs would drop out, leaving only those H-H values corresponding to significant *SCheck* outcomes; 24 of the 40 cases from the first test center B data set would go missing; and (remarkably) all but one of the five cases in Table 3 data set would disappear, leaving just the single H-H value with a significant *SCheck* outcome (H-H=1.19).

Increasing the EEIC minimum from the value of six recommended by Harpp, Hogan, & Jennings (1996) to ten would accomplish much the same result for the data sets used above.

These will be matters to explore further. In the meantime, one might ask why, given these findings, I would use Lertap's H-H values at all? Convenience. Had I confirmed the Harpp, Hogan, & Jennings (1996) findings, Lertap 5.5 users would have an index of possible cheating ready to hand, with no need to resort to an additional program. As it is, it seems advisable for Lertap users wanting to gauge the extent of possible cheating to cross-check Lertap 5.5's present H-H results with another program[8].

[7] The Bonferroni cut-off was increased from its default value of 0.01 to 10.00, resulting in a cut-off Zb value of 5.31.

[8] A revised Lertap, with enhanced cheat-checking, may be forthcoming early in 2006.

The *Integrity* program

Scrutiny!, *Integrity*, and *SCheck* are the names of three other programs designed to assist with the process of detecting cheating on multiple-choice exams. *Scrutiny!*, like Lertap, is a commercial package which may be obtained from Assessment Systems Corporation[9].

I have not looked at *Scrutiny!*. Cizek (2001) reported that *Scrutiny!* is "*easy to use*", and "*is compatible with many common input file formats*". But Cizek also related that *Scrutiny!* uses "*a method which, unfortunately, has not received strong recommendation in the professional literature*".

Integrity[10] is a comprehensive system which incorporates no less than five different methods of cheating detection, among them the "g2" procedure from the work of Frary, Tideman, and Watts (1977).

Using *Integrity* involves an off-line "batch" process somewhat reminiscent of mainframe computing: (1) the two data files required by the program are prepared on the user's computer; (2) the files are uploaded to the *Integrity* computer via the internet; and, (3) after a period of time, *Integrity*'s results are then downloaded, again using the internet. Users don't have to wait for their job to finish – once a job is submitted in step (2), a user may turn off his/her computer, and re-connect to the *Integrity* computer at a later time.

I took out a free *Integrity* trial account, and submitted the test center A data set discussed above to the program.

Whilst Cizek found *Scrutiny!* to be compatible with a variety of file formats, the same could not be said of the present version of *Integrity*. Data must be formatted as a "csv" text file[11], and item responses have to be coded as digits.

Almost all of my cognitive test data sets use letters as response options, usually {A,B,C,D}, or {A,B,C,D,E}. These had to be transformed to {1,2,3,4} or {1,2,3,4,5} for *Integrity*. Omitted responses have to be coded as a zero. The *Integrity* documentation implies that most users will have their exam results prepared by a scanner, and getting the scanner to output item responses as digits, and omitted responses as zeroes, is, it is suggested, something the user's "scanner operators" will set up.

My first trial *Integrity* run involved reformatting the test center A data set mentioned above so that it could be submitted to *Integrity*. Once I had the data file in the correct format, submitting it to *Integrity* via the

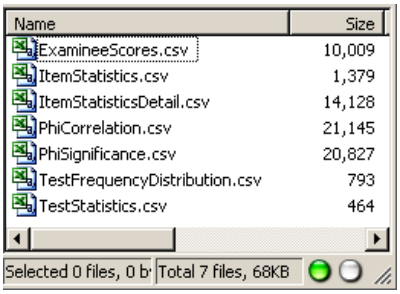
[9] See www.assess.com

[10] See www.integrity.castlerockresearch.com

[11] Comma-separated values, a common format easily used with Excel.

internet was straightforward. I accepted all of *Integrity*'s default options, left the job with *Integrity*, and returned to the website to pick up my results after a short period of time.

Results consisted of three files: a Zip folder and two PDF files. One of the PDF files, "IntegrityItemReport.pdf" was over a megabyte in size, while the other, "IntegrityOverallReport.pdf" was a mere 270 KB. The Zip folder contained the seven data files seen in the snapshot below.



The overall report contained the collusion detection results; a snapshot of the report's table of contents will provide an idea of the scope of these results:

Integrity's summary findings regarding collusion are well formatted, as may be seen in the screen snapshot below:

Collusion detection

Summary

The item responses for all examinees have been compared
2 pairs of examinees have been identified by the collusion detection analysis.

Detailed collusion detection report (all examinees)

	Examinee ID	Writing center	B-Index	PAIR1	PAIR2	MESA	g2
Pair 1	210515		High	High	High	Low	High
	210516		9.522	1404.000	2304.000	2.128E-009	8.556
							8.519
Pair 2	210123		N/A	N/A	Low	N/A	N/A
	210126				553.000		

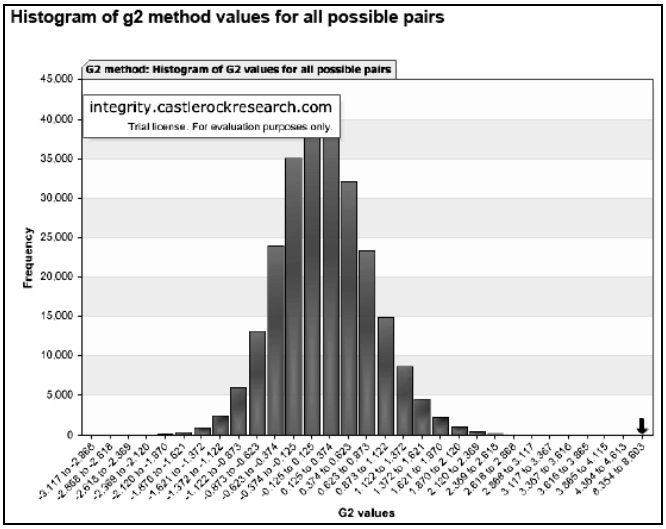
How do these results from *Integrity* correspond to what I found using Lertap's H-H index?

Integrity's "Pair 1", IDs 210515 and 210516, corresponds to the H-H value of 26.00 seen in Table 1. Each of *Integrity*'s five collusion detection methods has flagged this pair as possibly engaging in shenanigans.

Integrity's "Pair 2", IDs 210123 and 210126, corresponds to the second H-H value seen in Table 1, that is, H-H=3.67. Note that only one of the five *Integrity* methods has marked this pair as suspect.

There is some agreement here, but it's not something I would stress – the H-H value of 26.00 is such an extreme outcome that almost any detection method should flag it.

Another nice feature of *Integrity* is its graphics output. The histogram captured below was part of the overall report[12]:



The arrow to the right of the graph is pointing to the student pair having the H-H 26.00 value.

The Harpp, Hogan, and Jennings (1996) article has histograms of H-H values plotted in a similar format, and, as above, one is tempted to remark "*Look at that outlier! It's miles removed from the main body*". But some caution is required: the scale of the graph makes it impossible to show all bars – frequency is plotted from zero to 45,000, meaning that histogram bars corresponding to frequencies of 100 or less are not likely to be plotted, the graph's scale is too coarse for that. In turn, this means that the trip from the right-most bar out to the arrow is most certainly not really passing over empty land – there are g2 values not plotted. (Lertap

[12] The small box with the web address does not appear on the graph when users have purchased a license for *Integrity*.

fixes this problem by having a complete frequency table on display, making it possible to better assess just how far the extreme H-H values are from their closest neighbors.)

Integrity comes ready to breakout results by "writing center", and by subgroups, such as gender, a very useful capability.

There is more to *Integrity*. Its item analysis report includes a full page of data for *each* test item. Such pages have a graph showing response patterns, somewhat like Lertap's quintile plots with data tables, and selected classical item statistics, such as difficulty and discrimination. Whereas Lertap produces three reports with item statistics in tables, Integrity has summary paragraphs which verbally interpret various item results, such as distractor performance.

I found it interesting to note that the reports produced by *Integrity* assumed item responses to be letters, not digits. So, whereas one has to take care to use digits as item responses at the time of data entry, *Integrity*'s item analysis results use letters for item responses.

*Integrity'*s batch mode of operation is not as cumbersome as might be thought; its dependence on internet access could be a limitation to some. Perhaps a future version will support more input data formats.

Integrity is clearly a useful tool for those with an interest in collusion detection, although readers might want to take in the comments on *SCheck* below.

The *SCheck* program

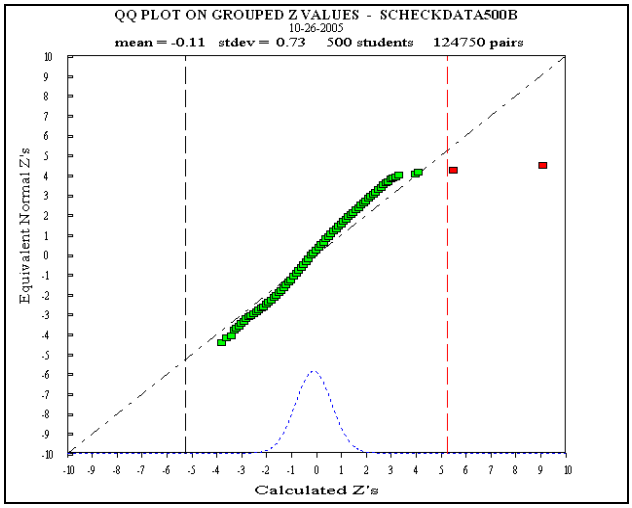
The g2 collusion index seen in *Integrity* stems from what Wesolowsky (2000) has referred to as the "seminal work" of Frary, Tideman, and Watts (1977). Wesolowsky's paper presented a modification to Frary et al.

In researching Wesolowsky's modification, Tideman and Kheirandish (2003) found it had "*noticeably better power than the probabilities suggested by Frary* et al.".

Better power means that Wesolowsky's method has a greater likelihood of rejecting the response-independence hypothesis if the hypothesis is in fact false – it is more capable of detecting possible cheating, less likely to make a Type II error.

Returning to the test center A data set discussed above, *SCheck* rejected the response-independence hypothesis for the first two cases seen in Table 1, whereas g2 rejected only the first of these two – this may be an example of *SCheck*'s superior statistical power.

Wesolowsky's method is manifested in the *SCheck*[13] program. One of its features involves a "Q-Q" plot of results – the plot below was produced by *SCheck* for the test center A data set:



A Q-Q plot, a quantile-quantile plot, may be used to check on the goodness of a model. The "Calculated Z's" along the graph's abscissa correspond to *SCheck*'s test statistic, *Zb*, a standardized deviate calculated from the responses of each pair of students (Wesolowsky, 2000). The "Equivalent Normal Z's" along the ordinate permit an assessment of the degree to which the distribution of the test statistic follows the normal (Gaussian) curve – if the calculated *Zb* values are normally distributed, the points in the plot will fall on a straight line.

The dotted line to the right of the graph is *SCheck*'s cut-off value[14]. Student pairs whose *Zb* values fall beyond the cut-off are suspect; the response-independence hypothesis is rejected for these pairs, pointing to possible cheating.

In the case of the test center A data set, there are two Q-Q "blips" to the right of the cut-off. It has to be noted that each blip may actually represent more than one student pair; the precision of the graph is such as to preclude plotting each and every actual *Zb* value, much as was the case for the histogram of g2 test values seen earlier. However, *SCheck*'s graph makes it possible to see how outlying the outliers are, and it's also possible to effectively gauge what's happening to the immediate left of the cut-off. In the graph above, we see another pair of blips which are on the right, and set off from the main body. When I relaxed *SCheck*'s Type I error rate

[13] See www.business.mcmaster.ca/msis/profs/wesolo/wesolo.htm

[14] The cut-off Zb value was 5.24

control for the test center A data set, the student pairs corresponding to these blips then came to fall to the right of the new cut-off *Zb* value[15].

One of *SCheck*'s standard reports, the ".OUT" file, is used to precisely identify student pairs whose test statistic has fallen beyond *SCheck*'s *Zb* cut-off. Samples from an ".OUT" file are shown below:

```
** pair = 156  158 **  Harpp-Hogan stat = 3.67
#####################################################################
Zb =  5.431  'equivalent' z from the BVP model
Significance of Zb on a pre-selected pair = 2.8E-8
Significance bound (Bonferroni)
on program selected pairs = 3.5E-3
#matches = 57 | 60   (mu,s)=(  40.347,   3.329)
prop. right for 156 = 0.783        prop. right for 158 = 0.783
```

```
** pair = 435  436 **  Harpp-Hogan stat = 26.00
#####################################################################
Zb =  9.084  'equivalent' z from the BVP model
Significance of Zb on a pre-selected pair = 5.2E-20
Significance bound (Bonferroni)
on program selected pairs = 6.5E-15
#matches = 59 | 60   (mu,s)=(  27.840,   3.707)
prop. right for 435 = 0.567        prop. right for 436 = 0.550
```

Little formatting sophistication is found in the output produced by the version of *SCheck* used for this paper, version "5c". The ".OUT" file is a simple text file, devoid of formatting, with no page breaks. Like the version of *Integrity* used above, input data files have to carefully follow one of the formats produced by certain scanners. Fortunately, item responses may be coded as digits or letters – if you've used test items with options of {A,B,C,D,E}, for example, there is no requirement to convert them to {1,2,3,4,5}.

[15] The Bonferroni cut-off was increased from its default value of 0.01 to 10.00, resulting in a cut-off Zb of 3.77

SCheck's output includes much more than the Q-Q plot and summary *Zb* results exemplified in the two samples immediately above. Indices of item difficulty and discrimination are created, as well as tables which provide information on model fit and performance, both for individual test items, and for individual students. Once again, however, *SCheck* displays its output in a format which is much less than elegant.

Whereas Lertap and *Integrity* are commercial products, *SCheck* is kept under careful wraps – those interested in using it will want to contact Professor Wesolowsky[16].

Like *Integrity*, *SCheck* has clear value for those interested in detecting cheating on multiple-choice exams, even though its output is not extravagantly formatted. Tideman and Kheirandish (2003) give *SCheck* an edge when it comes to methodology, and Lertap users will find an in-built interface which eases the process of preparing data for input to *SCheck*.

Summary

A new version of Lertap with some support for detecting cheating was released in July, 2005. It featured the use of the H-H index, a descriptive statistic based on dividing the number of exact item response errors made by a given pair of students by their total number of response differences over all test items.

Previous research on the application of this index was promising, suggesting it to be a "powerful" indicator of cheating. The results of the present research, however, have tended to refute the earlier findings, suggesting that the H-H index should be used, at best, with great caution.

Integrity and *SCheck* are other programs working in the area of cheating detection. Both have some limitations, but they have undoubted relevance and promise for users of multiple choice tests wanting to discern the possible presence of student collusion.

To circumvent the limitations of the H-H index highlighted by this research, present Lertap users should consider use of the *SCheck* interface built into version 5.5.

It is probable that all of the software systems mentioned in this paper will continue to improve. Another year might well see them with considerable enhancements, and, perhaps, fewer differences.

[16] See www.business.mcmaster.ca/msis/profs/wesolo/wesolo.htm

References

Cizek, G.J. (2000, April). *An overview of issues concerning cheating on large-scale tests*. Paper presented at the Annual Meeting of the National Conference on Measurement in Education, Seattle, Washington.

Frary, R.B. (1993). Statistical detection of multiple-choice answer copying: Review and commentary. *Applied Measurement in Education,* 6(2), 153-165.

Frary, R.B., & Tideman, T.N. (1997). Comparison of two indices of answer copying and development of a spliced index. *Educational and Psychological Measurement,* 57, 20-32.

Frary, R.B., Tideman, T.N., & Watts, T.M. (1977). Indices of cheating on multiple-choice tests. *Journal of Educational Statistics, 2*, 235-256.

Harpp, D.N., & Hogan, J.J. (1993). Crime in the classroom – detection and prevention of cheating on multiple-choice exams. *Journal of Chemical Education,* 70(4), 306-311.

Harpp, D.N., Hogan, J.J., & Jennings, J.S. (1996). Crime in the classroom – Part II, an update. *Journal of Chemical Education,* 73(4), 349-351.

Nelson, L.R. (2000). *Item analysis for tests and surveys using Lertap 5.* Perth, Western Australia: Faculty of Education, Language Studies, and Social Work, Curtin University of Technology.

Tideman, N., & Kheirandish, R. (2003). Structurally consistent probabilities of selecting answers. *Journal of Applied Statistics,* 30(7), 803-811.

Wesolowsky, G.O. (2000). Detecting excessive similarity in answers on multiple choice exams. *Journal of Applied Statistics,* 27(7), 909-921.