VII. Rules of Thumb, Shortcuts, Loose Ends, and Other Off-Topic Topics: *Significant Relationships in the Life of a Psychometrician*

Unless you can prove your approximation is as good as my exact solution, I am not interested in your approximation. R. Daryl Bock[1]

Unless you can show me your exact solution is better than my approximation, I am not interested in your exact solution. Benjamin D. Wright[2]

Rule of Thumb Estimates for Rasch Standard Errors

The asymptotic standard error for Marginal Maximum Likelihood estimates of the Rasch difficulty δ or ability β parameters is:

1. $$se_x = \frac{1}{\sqrt{\sum_k p_{xk}(1-p_{xk})}}.$$

The summation is over the *N* people in the sample for the standard error of calibration for an item or over the *L* items on the test for the standard error of measurement of a person. A plot of this function is bowl-shaped with a rather wide, rather flat bottom. This should be used whenever reporting results and measures. For purposes of planning, there's a quick and dirty way.

Rule of Thumb

The basic formula for the standard error (Wright and Stone, 1979) adequate for comparing test lengths or determining sample size for purposes of design and development is:

2. $$se_x = \frac{2.5}{\sqrt{k}},$$ $k = N$, the number of people, for the standard error of d_x, or

$k = L$, the number of test items, for the standard error of b_x.

Examples

A reasonably typical value for the standard error of measurement for, say, a 60-item test is:

3. $$se_b = \frac{2.5}{\sqrt{60}} = 0.32 \text{ logits.}$$

If you have a six-item subtest and the developers or marketing staff want to report the scores, they should first be told that the typical standard error for that score is:

4. $$se_b = \frac{2.5}{\sqrt{6}} = 1.02 \text{ logits.}$$

Basis of the Rule

[1] I first applied to the University of Chicago because Prof. Bock was there.

[2] There was a reason I ended up working with Prof. Wright.

If every item is of equal difficulty and exactly on target for a person, then $p = 0.5$ and equation 1 for b becomes:

5. $$se_b = \frac{1}{\sqrt{(0.5)(0.5)L}} = \frac{2}{\sqrt{L}}.$$

Because not all items are of equal difficulty and not all tests are perfectly targeted, expression 5 is an unreachable lower limit.

If the probability for every item is 0.9 rather than 0.5, the numerator will be $1/\sqrt{[(0.9)(0.1)]} = 3.33$ rather than 2. The value of 2.5 in expression 2 is a middling value that reflects the situation that most items are fairly well, but far from perfectly, targeted for most people.

One of the advances Rasch brings over traditional true score theory is that the standard error isn't a one-size-all answer but a function of the number correct score; this rule of thumb doesn't seen to fit with that philosophy. It is intended for planning and design purposes, not for final reporting. The rule of thumb works better for longer tests than for shorter.

Analogous arguments can be used to make an informed guess at the standard error for estimated difficulty by replacing L with N throughout.

Rule of Thumb Estimate for Test Reliability

The traditional notion of test reliability might be defined as the ratio of the *true score* variance σ_t^2 to the total variance σ_T^2. The total variance is the true score variance plus the error variance σ_e^2.

6. $$r = \frac{\sigma_t^2}{\sigma_T^2} = \frac{\sigma_t^2}{\sigma_t^2 + \sigma_e^2}$$

This whole idea has to do with how consistently the instrument orders some population of examinees rather than how precisely any one examinee's score is estimated. It may still be relevant in the norm referenced world if you want to go there. It has nothing to do with Rasch measurement. But sometimes people ask.

Rule of Thumb

The basic formula for estimating the reliability of a proposed test design that you can do in your head or on cocktail napkins is:

7. $$r = \frac{L}{L+6},$$ where L is the number of test items.

This assumes items of typical quality. The expression can, of course, be turned around to give the test length needed for a given reliability.

8. $$L = 6\frac{r}{1-r}$$

Example

If we have the time and money to administer a 60-item test, we can expect a reliability of $60 / 66 = 0.91$. For the hypothetical six-item subtest, the reliability could be 6 / 12 = 0.5.

If the contract requires a test reliability of *0.85*, as suggested by many powers that be, the test length required is *6(0.85 / 0.15) = 34*. I'd throw in a few more just to be safe.

Basis of the Rule

Experience with the logit metric native to the Rasch measurement model has suggested that the typical within grade variance is about one logit, over a variety of grades, content areas, and assessments. Using this an estimate of the unknown true score variance, expression 2 provides a rule of thumb for the error variance. Substituting these values into expression 5 gives:

$$9. \qquad r = \frac{1.0}{1.0 + (2.5/\sqrt{L})^2} = \frac{1.0}{1.0 + 6.25/L}.$$

Using $6.25 = 2.5^2$ instead 6.0 makes expression 2 into 2.45 rather than 2.5 and would imply a reliability of 0.9091 rather than 0.9057 for a test length of 60 items. This is beyond the accuracy of the assumption that the unknown variance equals one and beyond the credibility of the reasoning that led to the value of 2.5 in the first place. The rule of thumb works better for longer tests than for shorter and it may well change when we write better items. We'd learn to live with it.

Rule of Thumb Estimate of Standard Errors of Measurement from Raw-to-Logit Conversion Table

No respectable statistic would be seen in public without its standard error but not everyone involved in educational measurement appreciates this decree. It is not uncommon to see tables like the following intended to be used to look up scale scores given a number correct score on a fixed form.

Unbeknownst to the system analysts who created the table, the standard errors of measurement are embedded.[3]

Rule of Thumb

A rule of thumb for retrieving the standard error of measurement for the *logit ability* b_r from a raw-to-scale score conversion table is:

$$10. \qquad SEM(b_r) = \sqrt{\frac{b_{r+1} - b_{r-1}}{2}}.$$

The calculation needs to be done in logits but while we're here, the standard error of measurement $SEM(G_r)$ in the *GRit* metric is $SEM(b_r)$ multiplied by the scaling factor used to convert logit scores to *GRits*; in our example, that's 91.

Example

The standard error of measurement associated with a score of 10 for the table below is:

[3] The *GRit* scores, in this example, are a linear transformation of the logit score equal to *91(logit) + 600*. I have included them here, not because I need them, but because this type of table is generally created to look up scale scores for reporting. I will continue to think in logits as long as I can.

$$\text{SEM}_{10} = \sqrt{\{[0.2695 - (-0.2695)]/2\}} = 0.52.$$

The short cut of the preceding section would give 2.5/√20 =0.56 for a 20-item test but the center of a test (e.g., around 10 out of 20) will generally be lower.

Table 1: Raw-to-Scale Conversion Table 20-Item Fixed Form; No Standard Errors

Number Correct	Percent Correct	Logit Ability	GRit Score
0	0%	-4.8180	161.6
1	5%	-3.5363	278.2
2	10%	-2.7283	351.7
3	15%	-2.2032	399.5
4	20%	-1.7909	437.0
5	25%	-1.4379	469.2
6	30%	-1.1201	498.1
7	35%	-0.8245	525.0
8	40%	-0.5428	550.6
9	45%	-0.2695	575.5
10	50%	0.0000	600.0
11	55%	0.2695	624.5
12	60%	0.5428	649.4
13	65%	0.8245	675.0
14	70%	1.1201	701.9
15	75%	1.4379	730.8
16	80%	1.7909	763.0
17	85%	2.2032	800.5
18	90%	2.7283	848.3
19	95%	3.5363	921.8
20	100%	4.8180	1038.4

Basis of the Rule

The Newton method iteration for estimating the logit ability for a raw score of *r* is:

11. $$b_r^{k+1} = b_r^k + \frac{r - \sum_i p_{ri}}{\sum_i p_{ri}(1 - p_{ri})},$$ where *k* is the previous iteration, and

Σp_{ri} is the expected score based on b_r^k.

If we use a raw score of $(x-1)$ as our starting point for computing the logit ability for a score of *x*, then $\Sigma p_{(x-1)i}$ will be equal to *x-1* because $b_{(x-1)}$ is defined as the value satisfying the equation $(x-1) - \Sigma p_{(x-1)i} = 0$. In this case, equation *11* becomes:

12. $$b_x = b_{x-1} + \frac{x - (x-1)}{\sum_i p_{(x-1)i}(1 - p_{(x-1)i})} = b_x + \frac{1}{\sum_i p_{(x-1)i}(1 - p_{(x-1)i})},$$ and

13. $$b_x - b_{x-1} = \frac{1}{\sum_i p_{(x-1)i}(1 - p_{(x-1)i})}.$$

The square root of expression *13* was defined in expression *1* as the standard error of estimation.

The rule of thumb in expression *9* averaged the distances to the two scores adjacent to *r*, $[(b_{r+1} – b_r) + (b_r – b_{r-1})] / 2$, to make the process a little more symmetric. Once again the approximation is weakest in the extremes. Depending on the convergence criterion, Newton's Method would typically require two or three more iterations at the extreme scores even for nicely constructed tests like this.

Longer Example

We repeat half the table used above with the standard errors that should have been included included and our rule of thumb approximations for comparison. These tables tend to be more or less symmetric around L/2.

Table 2: One Half of a Raw-to-Scale Conversion Table
20-Item Fixed Form with Standard Errors

Number Correct	Logit Ability	Logit Standard Error	Rule of Thumb SEM	Difference	GRit Score	GRit Std Error	GRit Rule of Thumb	Difference
0	-4.8180	1.8554			161.6	169.8		
1	-3.5363	1.0548	1.0222	-0.0326	278.2	96.0	93.0	-3.0
2	-2.7283	0.7856	0.8164	0.0308	351.7	71.5	74.3	2.8
3	-2.2032	0.6749	0.6846	0.0097	399.5	61.4	62.3	0.6
4	-1.7909	0.6142	0.6186	0.0044	437.0	55.9	56.3	0.4
5	-1.4379	0.5767	0.5791	0.0024	469.2	52.5	52.7	0.2
6	-1.1201	0.5523	0.5538	0.0015	498.1	50.3	50.4	0.1
7	-0.8245	0.5362	0.5373	0.0011	525.0	48.8	48.9	0.1
8	-0.5428	0.5260	0.5268	0.0008	550.6	47.9	47.9	0.1
9	-0.2695	0.5203	0.5210	0.0007	575.5	47.3	47.4	0.1
10	0.0000	0.5185	0.5191	0.0006	600.0	47.2	47.2	0.0

We can also use this relationship to save a little time when computing MMLE estimates of ability. Typically, we use $ln[r / (L – r)]$ as the starting value in the iterative process for each raw score *r*, which assumes the item difficulties are centered at zero. We can instead begin in the center for $r = L/2$ and $b_r = 0$; once we have that estimate and its standard error, we can move up one and down one score by adding (or subtracting) the standard error squared to the last estimate and use that as a starting value. This might save one round of iteration, but it's cuter.

Everything You Need To Know

With the possible exceptions of programmers' convenience, there is no reason to store four versions of the observation, x_{vi}, p_{vi}, y_{vi}, and z_{vi}. All we really need are the y_{vi}; everything else can be reconstituted when and if we need them.

The last line of the table goes a step further and deduces the difficulty of the item from the residual when we know the estimated ability for the person. This is most useful when examinees can take unique item sets. The relationship is turned around in the penultimate line to provide the ability given the difficulty. This exposes another vein to control the model that, to my knowledge, has not been either mined or bled.

Table 3: Relationships of y_{vi} *with other statistics*

When …	Incorrect	Correct
then $y_{vi} = x_{vi} - p_{vi}$	**< 0**	**> 0**
Observed x_{vi}	0	1
Expected p_{vi}	$-y_{vi}$	$1-y_{vi}$
Standardized z_{vi}^2	$-y_{vi} / (1+y_{vi})$	$y_{vi} / (1-y_{vi})$
Variance s^2	$-y_{vi}(1+y_{vi})$	$y_{vi}(1-y_{vi})$
Ability b_v	$d_i + ln[(-y_{vi})/(1+y_{vi})]$	$d_i + ln[(1-y_{vi})/y_{vi}]$
Difficulty d_i	$b_v - ln[(-y_{vi})/(1+y_{vi})]$	$b_v - ln[(1-y_{vi})/y_{vi}]$

This entire topic may be residual paranoia from the era when storage was expensive. However, the *N* by *L* arrays can be large and it may be convenient, perhaps even prudent, not to have too many versions of the same information lying around.

Alpha-Integer Scoring

While we are in the business of minimizing storage space, there is one version of the observation that y_{vi} does not capture. That is the actual response, typically *A, B, C,* or *D* for a multiple choice item. One common practice is to code the response with alphas (i.e., *A, B, C,* or *D*) when correct and integers (i.e., *1, 2, 3,* or *4*) when incorrect. Table 4 shows the internal representation for this coding, which suggests the rationale.

Table 4: ASCII Codes for Alpha-Integer Coding.

	Correct			Incorrect		
Response	Value	Hex	Binary	Value	Hex	Binary
A	A	41	0100 0001	1	31	0011 0001
B	B	42	0100 0010	2	32	0011 0010
C	C	43	0100 0011	3	33	0011 0011
D	D	44	0100 0100	4	34	0011 0100

The raw response is captured in the low order byte of the binary or hex code, which, whether correct or incorrect, are identical and can be manipulated as short integers. The scored response is captured in the second bit of the high order byte, which is *1* if correct and *0* if incorrect. If you are clever enough with bits, bytes, and masks, the zero/one scores can be deduced without resulting to IF statements. There are even trickier, more compact formats that might be devised, but modern, state-of-the-art system analysts don't seem much concerned with saving storage space when stacked against the high from writing scrutable code. They may however be addicted to speed, or at least caffeine.

First Principle

I have now reported almost everything I know about the basic Rasch model; anything left out is a simple variation on the first principle and follows directly from something Georg Rasch had said by 1960. The first principle is that for any person, regardless of individualities or idiosyncrasies, the probability of success on any item, regardless of particularities or peculiarities, is controlled by the simple distance between the person's ability and the item's difficulty, $\beta_v - \delta_i$. That principle makes everything, from calibration to control to application, too simple to be scholarly.

This is the ultimate Rasch shortcut. But revealing the aspect and defining the construct will still take real effort and premeditation.