

Worked Examples for Nominal Intercoder Reliability

by Deen G. Freelon (deen@dfreelon.org)

October 30, 2009

<http://www.dfreelon.com/utis/recalfront/>

This document is an excerpt from a paper currently under review for publication. It is provided to assist those interested in calculating nominal intercoder reliability by hand, and contains original worked examples for the following coefficients: percent agreement (2 and 3 coders), Scott's pi (2 coders), Cohen's kappa (2 and 3 coders), Krippendorff's alpha (2 and 3 coders), and Fleiss' kappa (3 coders). **This document is a work in progress; please notify the author of any errors or suggestions for clarification.**

RECAL2 WORKED EXAMPLES

1. Percent agreement.

Consider an example ReCal2 data file consisting of 2 columns and 10 rows of numerical judgments (Table 1). ReCal2 would consider this file to contain one variable with ten units of analysis coded by two judges, who will also be referred to as "coders" throughout this document. This variable contains three possible coding categories (represented by the recurring numbers 0, 1, and 2). The formula for percent agreement between two coders is simply the number of times they agreed divided by the total number of units of analysis, and can be easily calculated for the example data. The two judges disagree on only one unit (in the seventh row); they thus agreed 90% of the time.

Table 1: Raw Example Data Formatted for ReCal2

0	0
0	0
1	1
0	0
2	2
1	1
1	0
2	2
0	0
1	1

2. Scott's pi and Cohen's kappa.

To calculate Scott's pi and Cohen's kappa, it will be necessary to make use of cross-tabulation tables that display the frequencies with which each coder agreed upon each coding category.

Table 2 contains these frequencies for the example data.

Table 2: Frequency Matrix for ReCal2 Example Data

		Coder 1 (= col. 1)			
		<i>Category 0</i>	<i>Category 1</i>	<i>Category 2</i>	<i>Total</i>
Coder 2 (= col. 2)	<i>Category 0</i>	4	1	0	5
	<i>Category 1</i>	0	3	0	3
	<i>Category 2</i>	0	0	2	2
	<i>Total</i>	4	4	2	10

Scott's pi and Cohen's kappa share the same general formula, although one of its components is calculated differently for each coefficient. The formula is

$$\frac{P_o - P_e}{1 - P_e}$$

where P_o is observed agreement and P_e is expected agreement (Scott, 1955; Cohen, 1960).

Observed agreement here is identical to percent agreement except that it is represented as a decimal. The formulae for expected agreement, however, are different for Scott's pi and Cohen's kappa. The expected agreement formula for Scott's pi is $\sum p_i^2$, where p_i is the proportion of the sample coded as belonging to the i th category (Scott, 1955). To calculate this quantity, we must begin by adding together each coder's respective totals, or marginals, within each coding category. The marginal sums for the example data can be seen in the fourth column of Table 4.

Next, each marginal sum is divided by the total number of decisions for the variable, which is the

number of cases multiplied by the number of coders. For the example variable, the total number of decisions is 20 (ten cases multiplied by two coders). The resulting quotients are known as the "joint marginal proportions" for each coding category. Each joint marginal proportion is squared, and these squares are added together to produce the expected agreement value for Scott's pi. In the current example, expected agreement is $0.2025 + 0.1225 + 0.04 = 0.365$

Table 3: Constituent Quantities for Scott's pi and Cohen's kappa for ReCal2 Example Data

Coding category	Marginal for coder 1	Marginal for coder 2	Sum of marginals	Product of marginals	Joint marginal proportion	JMP squared
0	4	5	9	20	0.45	0.2025
1	4	3	7	12	0.35	0.1225
2	2	2	4	4	0.2	0.04

Scott's pi can now be calculated fairly simply by hand. Plugging the numbers into the formula presented above, we get:

$$\pi = \frac{.9 - .365}{1 - .365} = .843$$

The expected agreement formula for Cohen's kappa is

$$P_e = \frac{1}{n^2} \sum pm_i$$

where n is the number of cases and $\sum pm_i$ is the sum of the marginal products (Neuendorf, 2002).

To calculate this quantity, we begin by multiplying each pair of marginals within each coding category (fifth column of Table 3) and then summing the products. This sum in our example is $20 + 12 + 4 = 36$, which is then multiplied by the reciprocal of the square of the number of cases, $1/n^2$. Our number of cases being ten, the latter quantity is equal to 0.01, or $1/100$. Expected agreement is therefore $0.01(36) = 0.36$. We can now complete the Cohen's kappa formula for the example,

$$\kappa = \frac{.9 - .36}{1 - .36} = .844$$

3. Krippendorff's alpha (2 coders).

Krippendorff's alpha functions somewhat differently from Scott's pi and Cohen's kappa, both of which are premised upon frequency tables that produce category marginals for each coder.

Instead, it requires coincidences within units, matrices of which produce marginals indicating the total number of times each category is employed in the data set¹. These values are identical to the sums of marginals used in the calculations for Scott's pi (Table 3, fourth column), and can be seen in the coincidence matrix for the current example (Table 4). Note that coincidences are counted twice in the matrix (Krippendorff, 2007): for example, the four 0-0 agreements entail a value of 8 in the 0-0 cell. Disagreements (represented by the off-diagonal cells) are also counted twice, but in different cells; thus the sole disagreement here is counted once in the 0-1 cell and again in the 1-0 cell.

Table 4: Krippendorff's alpha Coincidence Matrix for ReCal2 Example Data

	<i>Category 0</i>	<i>Category 1</i>	<i>Category 2</i>	<i>total</i>
<i>Category 0</i>	8	1	0	9
<i>Category 1</i>	1	6	0	7
<i>Category 2</i>	0	0	4	4
<i>total</i>	9	7	4	20

Table 4 contains all the information needed to calculate Krippendorff's alpha. Its general formula for multiple coders working with nominal data is as follows (Krippendorff, 2007):

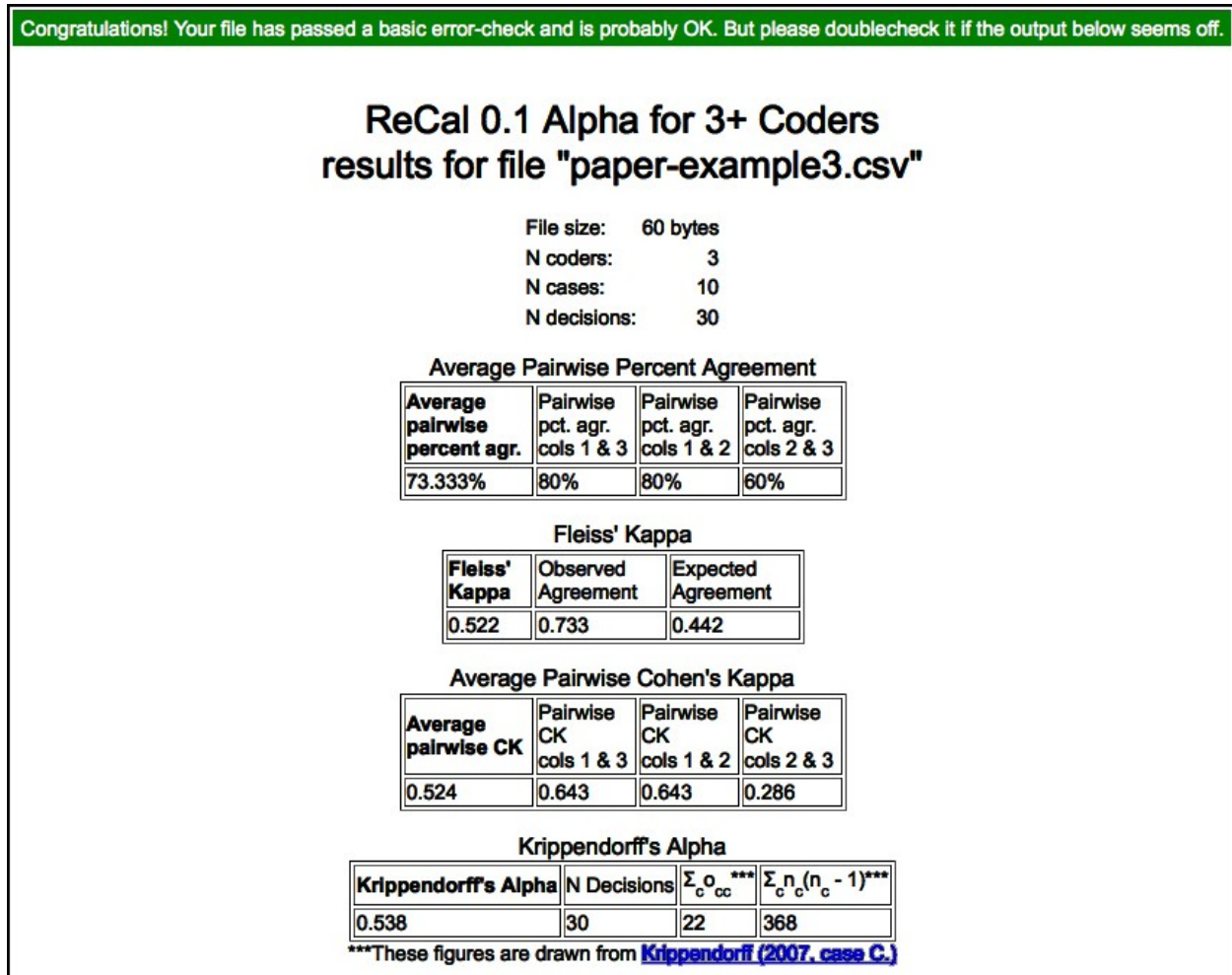
$$\frac{(n-1) \sum o_{cc} - \sum n_c (n_c - 1)}{n(n-1) - \sum n_c (n_c - 1)}$$

where n is the total number of coding decisions, o_{cc} is each agreement coincidence (diagonal cells in the coincidence matrix), and n_c is each coincidence marginal. Inserting the matrix values into the formula produces the following equation:

$$\alpha = \frac{(20-1)(8+6+4) - (9[9-1] + 7[7-1] + 4[4-1])}{20(20-1) - (9[9-1] + 7[7-1] + 4[4-1])} = .85$$

Figure 1 displays the ReCal2 output page for the data set represented in Table 1. Readers are encouraged to copy this data into their own CSV files and run them through ReCal to independently verify these results.

Figure 1: Output for ReCal2 example data



RECAL3 WORKED EXAMPLES

1. Average pairwise percent agreement.

Table 5 contains an example dataset for ReCal3 with 3 columns and 10 rows. ReCal3 would interpret a CSV file containing this information as 10 units of data for a single variable evaluated by 3 coders. The first statistics it computes are pairwise percent agreements (i.e. percent agreement between each possible pair of coders) and average pairwise percent agreement. In this

example, pairwise agreement is 80% for the first and second columns, 80% for the second and third columns, and 60% for the first and third columns. The definition of average pairwise percent agreement is the average of these three quantities, 73.3%.

Table 5: Raw Example Data Formatted for ReCal3

0	1	0
1	1	1
1	1	1
2	2	1
1	1	1
1	1	0
1	1	1
1	0	1
0	0	0
2	2	2

2. Fleiss' kappa.

Fleiss' kappa, actually an extension of Scott's pi for more than 2 coders, is defined by the following formula (Fleiss, 1971):

$$\frac{\bar{P} - \bar{P}_e}{1 - \bar{P}_e}$$

To determine \bar{P}_e , we must first calculate the quantity p_j , which is the proportion of judgments that a case belonged to the j th category; and to determine \bar{P} we need to know the quantity P_i ,

the proportion of agreement among all judges for the i th case. The formula for p_j is

$$p_j = \frac{1}{Nn} \sum n_{ij}$$

where N is the number of cases (ten in the example), n is the number of judges (three in the example), and n_{ij} is the number of judges who assigned the i th case to the j th category (hence $\sum n_{ij}$ is equal to the total number of times the category in question was used throughout the entire sample). This quantity is computed for each category the judges used (see Table 6), and the sum of their squares is \bar{P}_e , the expected agreement. \bar{P}_e in this example is $0.23^2 + 0.6^2 + 0.17^2 = 0.442$.

Table 6: Constituent Quantities of \bar{P}_e for ReCal3 Example Data

Coding category	$\sum n_{ij}$	p_j
0	7	0.23
1	18	0.6
2	5	0.17

P_i is calculated for each case (thus there will be ten values thereof in this example) using the following formula:

$$P_i = \frac{1}{n(n-1)} \sum n_{ij}^2 - n_{ij}$$

Therefore P_i for the first case of the data above would be

$$\frac{1}{3(3-1)} \left([2^2 - 2] + [1^2 - 1] \right)$$

which simplifies to 0.33, the proportion of possible pairwise agreements that were actually agreed upon. For the second case P_i would be 1 as all judges agreed that it belonged to the same category. \bar{P} is simply the average of all ten P_i s, which here is .733 (\bar{P} will always be equivalent to average pairwise percent agreement divided by 100). We now know all of the quantities necessary to complete the Fleiss' kappa formula. The equation with the example data inserted is

$$\kappa = \frac{.733 - .442}{1 - .442} = .522$$

3. Average pairwise Cohen's kappa.

ReCal3's Cohen's kappa calculations are identical to those of ReCal2 with one exception: after computing all possible pairwise kappas, it uses them to produce an average pairwise kappa value. The logic of this process is fairly straightforward, therefore no further elaboration will be presented here.

4. Krippendorff's alpha (3 coders).

Krippendorff's alpha for three coders functions slightly differently than with two coders. A coincidence matrix is constructed just as before, but it must account for all possible coincidences

within each unit, the number of which is given by the formula $m_u(m_u-1)$ where m_u is the number of coding decisions in a given unit. When only two coders are involved, m_u always equals two, which simplifies the process of constructing the coincidence matrix. With more than two coders, the method for completing the table becomes more complex. Table 7 is the coincidence matrix for the current dataset.

Table 7: Krippendorff's alpha Coincidence Matrix for ReCal3 Example Data

	<i>Category 0</i>	<i>Category 1</i>	<i>Category 2</i>	<i>total</i>
<i>Category 0</i>	4	3	0	7
<i>Category 1</i>	3	14	1	18
<i>Category 2</i>	0	1	4	5
<i>total</i>	7	18	5	30

To determine how much the first row of Table 5 contributes to the coincidence matrix, we must first calculate the number of coincidences it contains. With m_u being three, this row (along with all the others) contains $3(3 - 1) = 6$ coincidences. The single 1-1 agreement counts as two 1-1 pairs, and the two disagreements between 0 and 1 count as two 0-1 pairs and two 1-0 pairs (for a total of six). We now need to divide each of these quantities by $m_u - 1$ to determine how much it will contribute to the relevant cells. In all three cases, this value is $2/(3 - 1) = 1$; therefore the first row adds one to matrix cell 1-1, one to cell 0-1, and one to cell 1-0. Rows in which all coders agree are somewhat easier to interpret; such rows contribute m_u to the diagonal matrix cell representing the agreed-upon category. Since all three coders agreed that the second case belonged to category 1, that row contributes three to matrix cell 1-1. Applying these rules to each subsequent row of Table 5, we can complete the coincidence matrix such that its marginal totals

equal the number of times each category was used by all three coders.

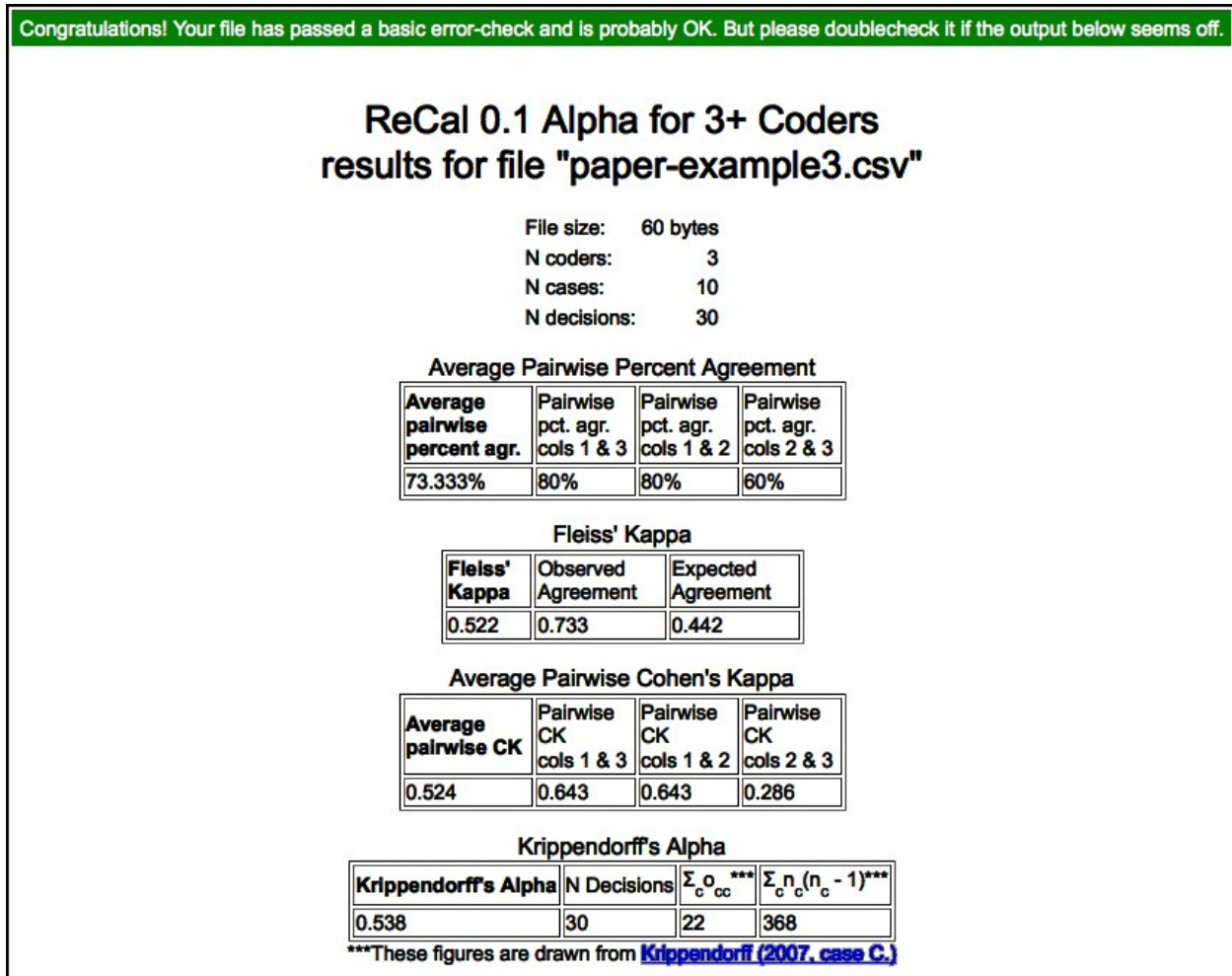
We can now calculate Krippendorff's alpha using the same formula as before:

$$\alpha = \frac{(30-1)(4+14+4) - (7[7-1] + 18[18-1] + 5[5-1])}{30(30-1) - (7[7-1] + 18[18-1] + 5[5-1])} = .538$$

Due to the complexity of the calculations for Krippendorff's alpha for more than two coders, readers may also want to consult the alternate worked example for four coders given by Krippendorff (2007).

Figure 2 displays the ReCal3 output page for this example data set.

Figure 2: Output for ReCal3 example data



NOTES

1 Technically this is only the case when all coders judge all units (i.e. there is no missing data), which ReCal requires.

REFERENCES

- Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and Psychological Measurement, 20*(1), 37–46.
- Fleiss, J. L. (1971). Measuring nominal scale agreement among many raters. *Psychological Bulletin, 76*(5), 378–382.
- Krippendorff, K. (2007). Computing Krippendorff's alpha-reliability. Annenberg School for Communication Departmental Paper 43. Retrieved May 13, 2008 from http://repository.upenn.edu/asc_papers/43/
- Neuendorf, K. A. (2002). *The content analysis guidebook*. Thousand Oaks, CA: Sage.
- Scott, W. A. (1955). Reliability of content analysis: The case of nominal scale coding. *Public Opinion Quarterly, 19*(3), 321–325.