KappaAcc: A program for assessing the adequacy of kappa

Roger Bakeman[1]

Abstract

Categorical cutpoints used to assess the adequacy of various statistics—like small, medium, and large for correlation coefficients of .10, .30, and .50 (Cohen, Cohen, J. (1988). Statistical power analysis for the behavioral sciences (2nd ed.). Lawrence Erlbaum Associates.)—are as useful as they are arbitrary, but not all statistics are suitable candidates for categorical cutpoints. One such is kappa, a statistic that gauges inter-observer agreement corrected for chance (Cohen *Educational and Psychological Measurement, 20*(1), 37–46, Cohen, Educational and Psychological Measurement 20:37–46, 1960). Depending on circumstances, a specific value of kappa may be judged adequate in one case but not in another. Thus, no one value of kappa can be regarded as universally acceptable and the question for investigators should be, are observers accurate enough, not is kappa big enough. A principled way to assess whether a specific value of kappa is adequate is to estimate observer accuracy—how accurate would simulated observers need to be to have generated a specific value of kappa obtained by actual observers, given specific circumstances. Estimating observer accuracy based on a kappa table the user provides is what KappaAcc, the program described here, does.

Keywords Statistics · Kappa · Kappa accuracy computer program

Categorical cutpoints used to assess the adequacy of various statistics—like small, medium, and large for correlation coefficients of .10, .30, and .50 (Cohen, 1988)—are as useful as they are arbitrary. Of course, most authors know, for example, that the difference between *p* values falling just below or just above the conventional cutpoint of .05 is inconsequential (Cohen, 1990; Rosnow & Rosenthal, 1989). Nonetheless, descriptions of results, especially when many variables are analyzed, are often improved with the use of defined and consistently used categorial terms.

Not all statistics are suitable candidates for categorical cutpoints, however. One such is kappa, a statistic that gauges inter-observer agreement corrected for chance (Cohen, 1960). Depending on circumstances I will detail shortly, a specific value of kappa may be judged adequate in one case but not in another. Thus, kappa is unlike other common statistics for which it is reasonable to say, for example, that a specified value represents a medium effect. Nonetheless, categorical terms for specific values of kappa have appeared in the literature (e.g., Fleiss, 1981, characterized kappas of .40–.60 as fair, .60–.75 as good, and over .75 as excellent). However, these definitions were not supported with a convincing rationale—unlike Cohen (1988) who offered detailed rationales for his suggested categorical terms—and did not take into account circumstances affecting the value of kappa, most importantly the order of the kappa table (i.e., the number of codes). In sum, no one value of kappa can be regarded as universally acceptable. The question for investigators should be, are observers accurate enough, not is kappa big enough.

Estimating how accurate observers would need to be to have generated a specific value of kappa, given its specific circumstances, provides a principled way to assess whether a specific value of kappa is adequate. When training and checking observers, our main concern should not be the magnitude of kappa but the level of observer accuracy we regard as acceptable. As always, the cutpoint selected is arbitrary. Gardner (1995) characterized 80% accuracy as discouragingly low "but possibly representative of the accuracy of classification for some social behaviors or expressions of affect" (p. 347). It seems reasonable to expect better, and—although 100% accuracy will likely elude us—85% or 90% accuracy may represent reasonable goals. KappaAcc, the computer program described here, computes estimated observer accuracy for a kappa table the user provides. It is based on equations developed by Gardner (1995) that take into account the circumstances of the particular kappa table.

✉ Roger Bakeman
bakeman@gsu.edu

[1] Department of Psychology, Georgia State University, Atlanta, GA 30303, USA

Kappa and weighted kappa

Researchers who ask observers to code or rate behavior often gauge inter-observer agreement with kappa (Cohen, 1960), primarily because kappa corrects for chance agreement, whereas a percentage agreement—sometimes seen in older literature—does not. To check agreement, researchers ask two observers to code (nominal scale) or rate (ordinal scale) the same sequence of events (or time intervals, or sessions), applying K codes or ratings to N events. Their N pairs of judgments are then tallied in a $K \times K$ kappa table (also called an agreement or confusion matrix). Rows represent one observer, columns the other observer, and rows and columns are labeled with the K mutually exclusive and exhaustive codes or ratings.

Cohen's kappa is an omnibus statistic, a single number that summarizes the agreement evidenced by the kappa table. The standard formula is:

$$\kappa = \frac{P_O - P_C}{1 - P_C}, \text{where} P_O = \sum_{i=1}^{K} p_{ii} \text{and} P_C = \sum_{i=1}^{K} p_{+i} p_{i+}$$

P_O represents observed agreement (the sum of the probabilities on the upper-left to lower-right diagonal), P_C represents chance agreement (the sum of the corresponding row and column probability products), and the formula emphasizes that kappa gauges observer agreement corrected for chance.

The formula for weighted kappa (Cohen, 1968) is more general:

$$\kappa_{wt} = 1 - \frac{\sum_{i=1}^{K} \sum_{j=1}^{K} w_{ij} x_{ij}}{\sum_{i=1}^{K} \sum_{j=1}^{K} w_{ij} e_{ij}}$$

Each observed value (x_{ij}) and each expected value (e_{ij}) is multiplied by the corresponding cell in an array of weights (w_{ij}). (Note: The formula for weighted kappa in Bakeman & Quera, 2011, p. 82, contains a typo; the "1 –" before the fraction was inadvertently omitted.) With standard weights—all cells on the diagonal set to 0, indicating agreement, and all off-diagonal cells set to 1, indicating that all disagreements are weighted equally—both the standard and the weighted kappa formulas yield identical results. If observers agreed for all events, the sum of the $w_{ij} x_{ij}$ products would be 0, the fraction after "1 –"would be 0, and so κ and κ_{wt} would equal 1, indicating perfect agreement.

When codes are nominal, it usually makes the most sense to weight all disagreements equally. But when ratings are ordinal, other arrays of weights could make sense. Figure 1 shows four possible weighting schemes, assuming $K = 5$: (a) the *standard* array weights agreements 0 and disagreements 1; (b) the *linear* array weights more extreme disagreements more highly (e.g., weighting a 1–3 disagreement 2 but a 1–5 disagreement 4); (c) the *w/1 standard* array regards disagreements within one scale point as agreements and weights them 0; and (d) the *w/1 linear* array likewise regards disagreements within one scale point as agreements but weights more extreme disagreements more highly. The default for the KappaAcc program is the standard array, but the user can select one of the other three weighting schemes or define a custom scheme if they wish.

Observer accuracy

Once paired observer judgments are tallied in a kappa table and kappa or weighted kappa has been computed, researchers understandably want to know whether the computed value is big enough—although, as noted earlier, I think the better question is whether observers are accurate enough, not whether kappa is big enough, that is does the value of kappa indicate adequate observer agreement? *Observer*
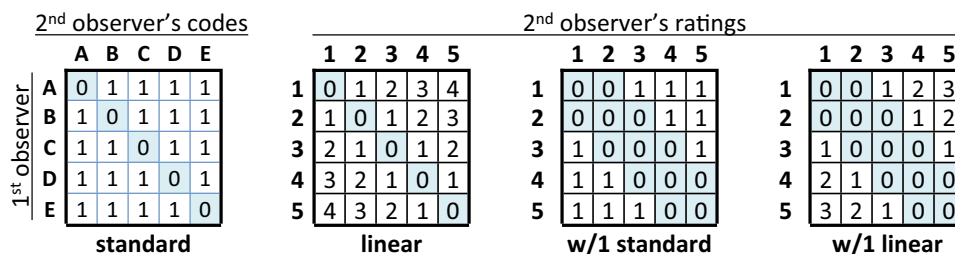

Fig. 1 Standard weight matrix for five codes and three possible weight matrices for 1 to 5 ratings. *Note*. In addition to the standard weight matrix for kappa, three other possibilities—potentially useful when rating instead of coding—are (a) the linear matrix, which weights more extreme disagreements more highly (e.g., weighting a 1–3 disagreement 2 but a 1–5 disagreement 4); (b) the w/1 standard matrix, which regards disagreements within one scale point as agreements and weights them 0, weighting other disagreements 1; and (c) the w/1 linear matrix, which likewise regards disagreements within one scale point as agreements but weights more extreme disagreements more highly. Agreements (weighted 0) are shaded

accuracy provides a reasoned way for determining whether a particular value of kappa is adequate. Unfortunately, its computation requires that we know the "true" state of affairs, whereas in the real world the true value of observer accuracy is unknowable. But in an ideal world of simulated observers, we can specify the true state of affairs, specifically: (a) the number of codes or ratings, (b) their simple probabilities, and (c) observer accuracy (the conditional probability that an observer will assign code A when the event is truly an A). KappaAcc assumes this ideal world.

Gardner (1995) has shown us how to model observer decision making in the ideal world. His equations let us determine the value of kappa that would result if two observers of specified accuracy were asked to assign *K* codes or ratings to events of specified probability (see also Bakeman et al., 1997; Bakeman & Quera, 2011). The inference is, if simulated observers of known accuracy achieve a value of kappa as big as the value achieved by observers in the real world, it is reasonable to assume that the actual observers are as accurate as the simulated ones. This is the fundamental assumption of the KappaAcc program. Using Gardner's model, it determines the percentage accuracy required of simulated observers to achieve the magnitude of the kappa observed by the actual observers. See the Appendix for details concerning Gardner's equations for computing estimated accuracy.

Circumstances affecting the value of kappa

Bakeman and Quera (2011) listed five circumstances that affect the value of kappa. First is observer accuracy, as just discussed. Second is the number of (mutually exclusive and exhaustive) codes or ratings. Third is the *prevalence* for individual codes or ratings (i.e., their simple probabilities); these could be equiprobable, moderately variable, or highly variable. Fourth is observer *bias* (i.e., the difference in prevalence between observers); its lack is evidenced when observers report similar probabilities for corresponding codes or ratings. And fifth is observer independence (as every researcher knows, when assessing observer agreement, observers must code or rate "blind," without knowledge of how the other observer did so). As detailed by Bakeman et al., (1997; also Bakeman & Quera, 2011), when *K* is less than five, especially when prevalence is highly variable, similar values for observer accuracy result in lower values of kappa. However, when *K* is greater than five, a larger number of codes or ratings and prevalence variability matter little.

Based on Gardner's (1995) model and corresponding equations, Bakeman and Quera (2011) provided tables showing values of kappa that would be achieved if observers were 80, 85, 90, and 95% accurate for various values of *K*, assuming that both observers' prevalence was equiprobable,

moderately variable, or highly variable. This was somewhat limiting and required interpolation. In contrast, KappaAcc uses information extracted from the kappa table the user provides to find a value of observer accuracy for simulated observers that would result in a value for kappa that is as large as the value obtained by the real observers. As noted earlier, our assumption is that the real observers must have been at least this accurate.

Here is how KappaAcc deals with the five circumstances affecting the value of kappa:

1. The accuracy for simulated observers is what KappaAcc produces. KappaAcc assumes that both observers are equally accurate. Gardner's model allows that accuracy for the observers be set separately, and even separately for the different codes. But doing so seems needlessly complicated and rationalizing different accuracies for different codes seems somewhere between challenging to impossible. The single percentage KappaAcc produces indicates that both simulated observers were at least that accurate for all codes or ratings.
2. KappaAcc extracts the value for *K*, the number of codes, from the kappa table the user provides.
3. KappaAcc sets prevalence as the means for the two observers' corresponding probabilities, based on the kappa table the user provides,. Thus, unlike the tables in Bakeman and Quera (2011), which reflect only four levels of variability, the simple probabilities used by KappaAcc reflect the observers' actual variability. Gardner's model requires a single set of probabilities—the "true" probabilities from the ideal world—which is why they are estimated with means for KappaAcc's computations.
4. For this reason, bias is not reflected in Gardner's model. Normally, we expect the row and column probabilities for a kappa table to be roughly similar. If not, we would have expected that discrepancies would have occasioned discussion with the observers and possibly retraining.
5. KappaAcc assumes that the two observers coded or rated the same session independently; this is simply standard practice.

The KappaAcc program

KappaAcc is an extension of the ComKappa program (Robinson & Bakeman, 1998). Most notably, it has the additional capability to compute estimated observer accuracy. Figure 2 shows the main KappaAcc window after entering tallies for 120 paired observer judgements for a scheme containing five codes.

Selecting the table icon (or *Run > Define a new table*) lets you define the number of codes or ratings and provide labels for them. You can then enter the values for the kappa
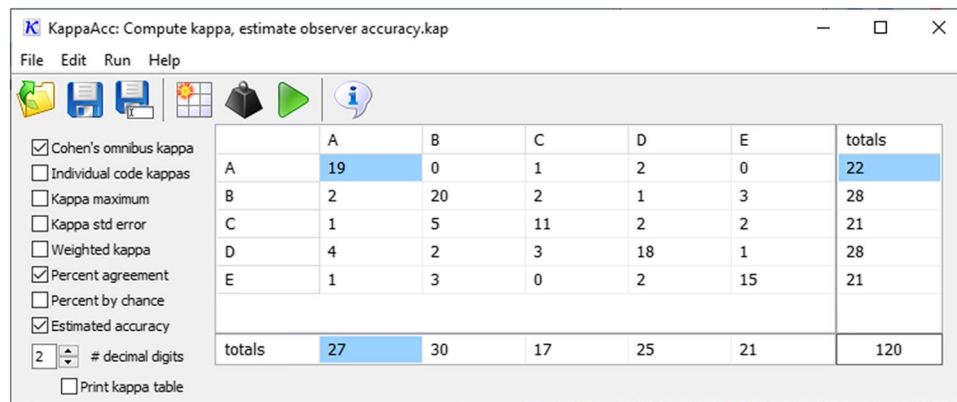
Fig. 2 KappaAcc main window showing tallies for a five-code kappa table

table directly in the window or copy-and-paste the values from a spread sheet. If you want other than the standard weights, select the weight icon (or *Run > Specify weights*) to select the weights you want. Finally check the statistics you want computed and select the compute icon (or *Run > Compute stats*). The possible statistics are:

1. Cohen's omnibus kappa: Kappa as described earlier using standard weights.
2. Individual code kappas: As noted earlier, kappa is an omnibus statistic; it summarizes agreement for a set of mutually exclusive and exhaustive codes. Computing a separate kappa for each code (forming a 2×2 table for each code and computing its kappa) can be informative because it identifies particularly problematic codes.
3. Kappa maximum: In theory, values of kappa can vary from -1 to $+1$, where 1 represents perfect agreement. Negative values are rare and indicate greater than chance disagreement, but kappa can equal 1 only when the tallies for the corresponding rows and columns are the same—that is, when the simple probabilities for each code are the same for both observers. If not, the value of kappa can be no higher than kappa maximum.
4. Kappa standard error: For completeness, KappaAcc computes kappa's standard error, although its usefulness is limited. Statistical significance for kappa is rarely reported; as Bakeman and Gottman (1997) wrote, even relatively low values of kappa can still be significantly different from zero, but not of sufficient magnitude to satisfy investigators.
5. Weighted kappa: If you specified other than standard weights, check this box so that the value of weighted kappa will be displayed.
6. Percent agreement: The agreement observed—P_O in the standard kappa formula.
7. Percent by chance. Agreement expected by chance—P_C in the standard kappa formula.

8. Estimated accuracy. If checked, weighted kappa is also displayed; it will have the same value as omnibus kappa if standard weights are used.

For the example data in Fig. 2, the standard or omnibus kappa was .61 (69% agreement, uncorrected for chance). Standard weights were used so the value of weighted kappa was the same. KappaAcc determined that simulated observers would need to be 82% accurate to achieve a kappa of .61. Often journal articles give just a value of kappa for each mutually exclusive and exhaustive scheme. I recommend that researchers who gauge observer agreement using kappa also give—and that editors and reviewers ask for—not just estimated observer accuracy, but also the number of sessions (if results from several sessions are pooled), the number of codes, and the number of tallies in the kappa table—information that provides necessary context. Moreover, for all the reasons noted here, a lone value of kappa is almost impossible to interpret, whereas observer accuracy admits to intuitive understanding.

For the example just given, results could be stated as follows: Using five codes, two observers coded one session and independently made 120 judgments. The value of kappa was .61 (69% agreement uncorrected for chance). To produce a kappa of this magnitude simulated observers would need to be 82% accurate, which was somewhat below our target of 85%.

Program details

KappaAcc is programmed in Pascal and compiled using Embarcadero® Delphi 10 Seattle. It will run on Windows computers or on Apple computers with a Windows simulator. It is contained in a single executable file, KappaAcc. exe. Once placed in a folder on your computer it can be invoked, like any other program, with a double click; you could also create a shortcut and place it on your desktop.

If your computer's security measures block running of "unknown" executable files, you may need help from your local IT people. This write-up as a PDF file and the KappaAcc.exe file are contained in a file (KappaAcc.zip) that can be downloaded at no charge from http://bakeman.gsucreate.org /kappa.

Appendix

Gardner ([1995](#)) modeled the decision making of the observers with two matrices of conditional probabilities, labeled *rho* (ρ) for the first observer and *sigma* (σ) for the second. Rows represent the observers' decision—the code or rating they assigned—and columns represent the true state of affairs. Thus cells on the diagonal (upper-left to lower-right) indicate the probability that an observer will code or rate an event accurately. If we assume that both observers are equally accurate, then the ρ and σ conditional probability matrices are the same.

For example, if we assume that observers are 82% accurate for each of five codes, then the diagonal cells of the ρ and σ matrices contain .82. And if we further assume that observers are equally inaccurate, then the off-diagonal cells of the ρ and σ matrices all contain .045: $(1 - .82)$ divided by $(K - 1)$. These example ρ and σ matrices are shown in Fig. [3](#) Panel A.

In addition to the ρ and σ matrices, the simple probabilities for each code or rating constitute a third array—a vector labeled pi (π). For example, if we estimate the true simple probabilities with the average simple probabilities for the example data given in Fig. [2](#), we would get the values for π shown in Fig. [3](#) Panel B.

As detailed in Bakeman et al. ([1997](#)), we can now compute the expected unconditional probabilities for the cells of the agreement matrix given fallible observers. This matrix, symbolized *u*, represents the decisions of simulated observers of the accuracy specified and is the basis for computing kappa. The formula is

$$u_{ij} = \sum_{k=1}^{K} \rho_{i|k} \sigma_{j|k} \pi_k$$

where u_{ij} represents a cell in the $K \times K$ agreement matrix. Each u_{ij} is the sum of K terms, where each term represents the probability that the first observer will code an event C_i and the second observer will code it C_j given a C_k event. Per basic probability theory, the probability of the joint event that constitutes each term is a product (AND), whereas the probability of any of these joint events occurring is a sum (OR). The terms in each series exhaust the possible ways the first observer might code an event C_i when the second observer codes it C_j. Applying Gardner's formula

A. Conditional probabilities (ρ and σ)

		True state of affairs				
		A	**B**	**C**	**D**	**E**
Observer coded	**A**	.820	.045	.045	.045	.045
	B	.045	.820	.045	.045	.045
	C	.045	.045	.820	.045	.045
	D	.045	.045	.045	.820	.045
	E	.045	.045	.045	.045	.820

B. Simple probabilities (π)

A	B	C	D	E
.204	.242	.158	.221	.175

C. Unconditional probabilities (*u*)

		2nd observer				
		A	**B**	**C**	**D**	**E**
1st observer	**A**	.1389	.0176	.0147	.0168	.0152
	B	.0176	.1640	.0160	.0182	.0166
	C	.0147	.0160	.1082	.0152	.0137
	D	.0168	.0182	.0152	.1501	.0158
	E	.0152	.0166	.0137	.0158	.1193

Fig. 3 Conditional, simple, and unconditional probabilities for simulated coders. *Note.* The conditional probabilities reflect accuracies (i.e., the probability that a simulated observer will code a given event correctly or incorrectly). KappaAcc assumes both observers are equally accurate, thus the single array given in the figure represents both the ρ matrix for the first observer and the σ matrix for the second observer. The simple probabilities (the π vector) are the estimated true probabilities for the five codes. The unconditional probabilities are the expected probabilities for the agreement matrix, based on the accuracy of simulated observers for five codes

for expected unconditional probabilities to the arrays for rho, sigma, and pi that assume 82% observer accuracy produces the values shown in Fig. [3](#) Panel C.

References

Bakeman, R., & Gottman, J. M. (1997). *Observing interaction: An introduction to sequential analysis* (2nd ed.). Cambridge University Press.

Bakeman, R., & Quera, V. (2011). *Sequential analysis and observational methods for the behavioral sciences*. Cambridge University Press.

Bakeman, R., Quera, V., McArthur, D., & Robinson, B. F. (1997). Detecting sequential patterns and determining their reliability with fallible observers. *Psychological Methods, 2*(4), 357–370. https://doi.org/10.1037/1082-989X.2.4.357

Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and Psychological Measurement, 20*(1), 37–46. https://doi.org/10.1177/001316446002000104

Cohen, J. (1968). Weighted kappa: Nominal scale agreement with provision for scaled disagreement or partial credit. *Psychological Bulletin, 70*(4), 213–220. https://doi.org/10.1037/h0026256

Cohen, J. (1988). *Statistical power analysis for the behavioral sciences* (2nd ed.). Lawrence Erlbaum Associates.

Cohen, J. (1990). Things I have learned (so far). *American Psychologist, 45*(12), 1304–1312. https://doi.org/10.1037/0003-066X.45.12.1304

Fleiss, J. L. (1981). *Statistical methods for rates and proportions*. Wiley.

Gardner, W. (1995). On the reliability of sequential data: Measurement, meaning, and correction. In J. M. Gottman (Ed.), *The analysis of change* (pp. 339–359). Lawrence Erlbaum Associates.

Robinson, B. F., & Bakeman, R. (1998). ComKappa: A Windows 95 program for calculating kappa and related statistics. *Behavior Research Methods, Instruments, and Computers, 30*, 731–732. https://doi.org/10.3758/BF03209495

Rosnow, R. L., & Rosenthal, R. (1989). Statistical procedures and the justification of knowledge in the psychological sciences. *American Psychologist, 44*(10), 1276–1284. https://doi.org/10.1037/0003-066X.44.10.1276

Open practices statement Because this paper does not report an experiment or experiments, no data or materials for experiments are available and no experiments were preregistered. However, the files containing the Pascal computer code are available on request for individual, non-commercial use; e-mail bakeman@gsu.edu.