The Stata Journal (2018)
18, Number 4, pp. 871–901

Implementing a general framework for assessing interrater agreement in Stata

Daniel Klein
International Centre for Higher Education Research Kassel
Kassel, Germany
klein@incher.uni-kassel.de

Abstract. Despite its well-known weaknesses, researchers continuously choose the kappa coefficient (Cohen, 1960, *Educational and Psychological Measurement* 20: 37–46; Fleiss, 1971, *Psychological Bulletin* 76: 378–382) to quantify agreement among raters. Part of kappa's persistent popularity seems to arise from a lack of available alternative agreement coefficients in statistical software packages such as Stata. In this article, I review Gwet's (2014, *Handbook of Inter-Rater Reliability*) recently developed framework of interrater agreement coefficients. This framework extends several agreement coefficients to handle any number of raters, any number of rating categories, any level of measurement, and missing values. I introduce the `kappaetc` command, which implements this framework in Stata.

Keywords: st0544, kappaetc, kappaetci, Cohen, Fleiss, Gwet, interrater agreement, kappa, Krippendorff, reliability

1 Introduction

The kappa statistic (Cohen 1960; Fleiss 1971) is one of the most popular coefficients to quantify agreement among raters. Researchers have also criticized kappa on various grounds and proposed alternative agreement coefficients (ACs) (Byrt, Bishop, and Carlin 1993; Feinstein and Cicchetti 1990; Gwet 2008a; Warrens 2012). Recently, Gwet (2014) developed a statistical framework that embraces several known ACs. Within this framework, he extends all of these ACs to the case of multiple raters, multiple rating categories, any level of measurement, and a varying number of ratings per subject. Gwet (2014) also introduces a new approach to statistical inference and a probabilistic benchmarking method for ACs. To date, none of these developments are readily available in major statistical software packages such as Stata. In the remainder of this article, I briefly summarize Gwet's (2014) framework of interrater ACs. I then introduce the `kappaetc` command, which implements this framework in Stata.

2 A general framework of ACs

Let us loosely define interrater agreement as the propensity for two or more raters (coders, judges, observers, etc.) to independently classify a given subject (object, unit of analysis, etc.) into the same predefined category (Gwet 2014, 14). In this section, I discuss the concept of chance-corrected ACs to measure this propensity. All formulas are as presented in Gwet (2014) except for slightly simplified notation.

2.1 A basic measure of agreement

Consider the simple case where $r = 2$ raters classify n subjects into one of q predefined categories. We record the resulting data in a $q \times q$ contingency table such as table 1.

Table 1. Distribution of n subjects by rating category and rater

	Rater B				
Rater A	1	2	...	q	Total
1	n_{11}	n_{12}	...	n_{1q}	$n_{1.}$
2	n_{21}	n_{22}	...	n_{2q}	$n_{2.}$
⋮	⋮	⋮	⋮	⋮	⋮
q	n_{q1}	n_{q2}	...	n_{qq}	$n_{q.}$
Total	$n_{.1}$	$n_{.2}$	...	$n_{.q}$	n

The observed proportion of agreement between the two raters, denoted p_o, is given by

$$p_o = \sum_{k=1}^{q} \frac{n_{kk}}{n}$$

where the numerator, n_{kk}, is the number of subjects that both raters classified into category k and the denominator, n, is the total number of subjects.

Although it satisfies our definition of interrater agreement, the observed proportion of agreement might not accurately reflect what researchers are interested in: the reliability of the classification process. The classification process is considered to be reliable when raters classify a given subject into the same category with a high probability because of particular characteristics of that subject. However, raters might be uncertain about the characteristics of some subjects and might therefore choose to classify those subjects into a category randomly. Given the limited number of predefined categories, the raters might still choose the same category by pure chance. This so-called chance agreement is not related to the characteristics of the subjects and thus cannot be regarded as evidence of reliability. Because the observed proportion of agreement does not differentiate between agreement due to the characteristics of the subjects and chance agreement, it might overestimate reliability.

2.2 Chance-corrected agreement: Cohen's kappa

Several researchers have proposed coefficients for measuring agreement beyond that expected by chance (Brennan and Prediger 1981; Cohen 1960; Fleiss 1971; Gwet 2008a; Krippendorff 2013; Scott 1955). These coefficients differ mainly in their definition of chance agreement, denoted p_e. Cohen (1960), for example, adopts the concept of sta-

tistical independence and defines chance agreement as the sum of the products of the marginal classification probabilities of both raters as follows:

$$p_e = \sum_{k=1}^{q} \frac{n_{k\cdot}}{n} \times \frac{n_{\cdot k}}{n}$$

Cohen (1960) proposes removing chance agreement from the observed proportion of agreement to obtain his chance-corrected AC kappa (denoted by the Greek letter κ). Gwet (2014) gives the general form for chance-corrected ACs, including kappa, as

$$\kappa_{\cdot} = \frac{p_o - p_e}{1 - p_e} \tag{1}$$

where $\kappa_{\cdot}$ is used as a generic symbol for various chance-corrected ACs. We discuss alternative definitions of chance agreement and the resulting chance-corrected ACs in section 2.6.

2.3 Partial agreement and weighted kappa

Suppose that we can order the q rating categories in table 1 so that adjacent categories imply a somehow less serious disagreement than nonadjacent categories. To reflect such different degrees of agreement, Cohen (1968) proposed a weighted variant of kappa: Define a set of weights, $w_{kl} \in [0, 1]$, so that 1 indicates perfect agreement and 0 implies complete disagreement; any weight in between reflects the extent of partial agreement.

Two sets of weights that are commonly applied are the linear and quadratic weights. To fix ideas, suppose there are $q = 4$ rating categories: 1, 2, 3, and 4. The corresponding linear and quadratic weights are illustrated in table 2 and table 3.

Table 2. Linear weights

	Rating			
Rating	1	2	3	4
1	1.00			
2	0.67	1.00		
3	0.33	0.67	1.00	
4	0.00	0.33	0.67	1.00

Table 3. Quadratic weights

	Rating			
Rating	1	2	3	4
1	1.00			
2	0.89	1.00		
3	0.56	0.89	1.00	
4	0.00	0.56	0.89	1.00

Both linear and quadratic weights are special cases of so-called power weights (Warrens 2014). Formally, we obtain the off-diagonal weights as

$$w_{kl} = 1 - \frac{|x_k - x_l|^a}{|x_{\max} - x_{\min}|^a} \; \forall \, k \neq l \tag{2}$$

where x_k and x_l refer to the kth and lth sorted ratings and $x_{\max}$ and $x_{\min}$ are the maximum and minimum of all ratings.[1] All diagonal weights are set to 1, indicating perfect agreement. Inserting different values for a into (2) gives linear weights ($a = 1$) and quadratic weights ($a = 2$), respectively. It is obvious that setting $a = 0$ results in the $q \times q$ identity weighting matrix that yields the unweighted AC.

The weighted observed agreement is then obtained as

$$p_o = \sum_{k=1}^{q} \sum_{l=1}^{q} w_{kl} \times \frac{n_{kl}}{n}$$

and Cohen's (1968) weighted chance agreement is given by

$$p_e = \sum_{k=1}^{q} \sum_{l=1}^{q} w_{kl} \times \frac{n_{k.}}{n} \times \frac{n_{.l}}{n}$$

Inserting both the weighted observed agreement and the weighted chance agreement into (1) yields the weighted kappa coefficient.

Gwet (2014, 91–97) suggests additional sets of weights to account for partial agreement that is implied by the data's level of measurement. With a suitable set of weights, any AC applies to any level of measurement, not just to the nominal scale.[2] Section 4.3 provides the formulas for the respective ordinal, radical, ratio, circular, and bipolar weights.

2.4 Dealing with missing values

In real-world data, the presence of missing values is a common problem. The usual approach for dealing with missing values is listwise deletion, where all subjects that have not been rated by both raters are excluded from all calculations. While we can naturally observe agreement between the raters only if both have classified a given subject, Gwet (2014, 36) suggests using all subjects, including those classified only by one rater, to estimate chance agreement. This increases the accuracy of the estimates for the marginal probabilities.[3]

To include all subjects in the calculation of chance agreement, we add a category (X) for missing values to the contingency table. This is illustrated in table 4.

1. Note the difference in notation between rating categories, which are always the integer sequence $1, 2, \ldots, k, l, \ldots, q$, and actual ratings x_k and x_l, which may take on any numeric value. Weights are calculated based on the latter.
2. Krippendorff (2011, 2013) suggests essentially the same approach, calling the weights "metric difference functions".
3. Krippendorff (2011, 2013) uses a similar approach for three or more raters. Zapf et al. (2016) provide empirical evidence that, compared with listwise deletion, the approach produces less biased results when ratings are missing completely at random. To the best of my knowledge, there are no studies addressing the validity of the approach for other mechanisms leading to missing values, such as missing at random or missing not at random.

Table 4. Distribution of n subjects by rating category and rater; missing ratings

	Rater B					
Rater A	1	2	...	q	X	Total
1	n_{11}	n_{12}	...	n_{1q}	n_{1X}	$n_{1.}$
2	n_{21}	n_{22}	...	n_{2q}	n_{2X}	$n_{2.}$
⋮	⋮	⋮	⋮	⋮	⋮	⋮
q	n_{q1}	n_{q2}	...	n_{qq}	n_{qX}	$n_{q.}$
X	n_{X1}	n_{X2}	...	n_{Xq}	0	$n_{X.}$
Total	$n_{.1}$	$n_{.2}$	...	$n_{.q}$	$n_{.X}$	n

In cell (X, q) of table 4, n_{Xq} is the number of subjects that rater A did not classify and that rater B classified into category q. Conversely, n_{qX} is the number of subjects not rated by rater B but classified into category q by rater A. Subjects that are not classified by either rater do not convey any information and are excluded so that n_{XX} is always 0.

We now obtain the observed proportion of agreement as

$$p_o = \sum_{k=1}^{q} \sum_{l=1}^{q} w_{kl} \times \frac{n_{kl}}{n - (n_{X.} + n_{.X})} \tag{3}$$

where the denominator in (3) is the number of subjects rated by both raters. Note that weights for partial agreement, w_{kl}, which were discussed in section 2.3, are easily included. Cohen's (1968) weighted chance agreement is calculated as

$$p_e = \sum_{k=1}^{q} \sum_{l=1}^{q} w_{kl} \times \frac{n_{k.}}{n - n_{X.}} \times \frac{n_{.l}}{n - n_{.X}} \tag{4}$$

respectively, and the kappa coefficient is still obtained by inserting (3) and (4) into (1).

2.5 Agreement among three or more raters

The concept of agreement between two raters that we have considered thus far is comparatively intuitive; defining agreement among three or more raters is not as straightforward. One possible way to define agreement among three or more raters is to consider all $r\,(r-1)\,/2$ possible pairs of the r raters (Gwet 2014, 48–52). Averaging the pairwise agreement given by (3) over all pairs of raters yields a measure for the observed proportion of agreement among three or more raters.

For three or more raters, it is no longer convenient to record the distribution of subjects by rating category and raters. Instead, we record the distribution of raters by subjects and rating category in an $n \times q$ table such as table 5.

Table 5. Distribution of r raters by subject and rating category

	Category					
Subject	1	...	k	...	q	Total
1	r_{11}	...	r_{1k}	...	r_{1q}	r_1
⋮	⋮		⋮		⋮	⋮
i	r_{i1}	...	r_{ik}	...	r_{iq}	r_i
⋮	⋮		⋮		⋮	⋮
n	r_{n1}	...	r_{nk}	...	r_{nq}	r_n
Average	$\overline{r}_{1.}$	...	$\overline{r}_{k.}$	...	$\overline{r}_{q.}$	$\overline{r}$

In the center cell of table 5, r_{ik} is the number of raters who classified subject i into category k. The row total, r_i, represents the number of raters who classified subject i. When there are no missing ratings, all row totals, r_1, r_2, ..., r_n, are identical and equal to the average number of raters per subject, denoted $\overline{r}$. We calculate the average weighted pairwise agreement as

$$p_o = \frac{1}{n'} \sum_{i=1}^{n'} \sum_{k=1}^{q} \frac{r_{ik} \left(\sum_{l=1}^{q} w_{kl} r_{il} - 1 \right)}{r_i \left(r_i - 1 \right)} \tag{5}$$

where n' is the number of subjects that are classified by two or more raters.

Conger (1980) applies the idea of averaging over chance agreement, defined in (4), to generalize Cohen's kappa to three or more raters. The formulas are somewhat complex; see Gwet (2014, 86) for more details.[4] The more popular generalization of Cohen's kappa to three or more raters, proposed by Fleiss (1971), is discussed in the next section.

2.6 Various definitions of chance agreement

Cohen's (1960, 1968) kappa is probably the most frequently used coefficient to assess agreement between two raters; for three or more raters, researchers usually apply Fleiss's (1971) generalization of Scott's (1955) pi. Fleiss called his coefficient kappa, giving the wrong impression that it is a generalized version of the coefficient proposed by Cohen (1960, 1968) before. However, Fleiss's kappa is based on a different concept of chance agreement and reduces to Scott's pi in the case of two raters. While the observed

4. Cohen's kappa cannot be estimated based solely on a table of rating frequencies such as table 5. For estimating kappa, the unique raters must be identified, which is not possible in tables of rating frequencies.

proportion of agreement for Fleiss's kappa is still given by (5), Gwet (2014, 87) gives the respective weighted chance agreement as

$$p_e = \sum_{k=1}^{q} \sum_{l=1}^{q} w_{kl} \pi_k \pi_l \tag{6}$$

with

$$\pi_k = \frac{1}{n} \sum_{i=1}^{n} \frac{r_{ik}}{r_i} \tag{7}$$

Brennan and Prediger (1981), following Bennett, Alpert, and Goldstein (1954), suggest another chance-corrected AC. Their concept of chance agreement is one of the simplest and adjusts merely for the number of possible rating categories, denoted q. Brennan and Prediger's weighted chance agreement is given by

$$p_e = \frac{1}{q^2} \sum_{k=1}^{q} \sum_{l=1}^{q} w_{kl}$$

and boils down to $p_e = 1/q$ in the unweighted case. The resulting coefficient is equivalent to the prevalence-adjusted and bias-adjusted kappa, also known as the PABAK (Byrt, Bishop, and Carlin 1993).

More recently, Gwet (2008a, 2014) proposes his AC that incorporates both the number of rating categories and the frequency with which they are used by the raters. The weighted chance agreement is given by

$$p_e = \frac{1}{q\,(q-1)} \sum_{k=1}^{q} \sum_{l=1}^{q} w_{kl} \sum_{k=1}^{q} \pi_k \,(1 - \pi_k)$$

where the term π_k is defined in (7). A thorough theoretical discussion and formal derivation of his coefficient is given in Gwet (2014, chap. 4).

The last coefficient that we will briefly discuss is Krippendorff's (1970, 2011, 2013) alpha. Although the original author's notation might imply otherwise, Gwet (2014) shows that this coefficient can be obtained using the concepts defined so far. In fact, Krippendorff's alpha is similar to Fleiss's kappa with four slight modifications. The weighted chance agreement is basically calculated using (6). However, in (7) n is replaced with n', and r_i in the denominator is replaced with $\overline{r}$. Further, the formula for observed agreement, given in (5), is modified by replacing the first occurrence of r_i in the denominator with $\overline{r}$. Notice that all three changes so far are relevant only in the presence of missing ratings.[5] Finally, the observed proportion of agreement is adjusted for small samples to obtain

5. This is because when all subjects are classified by all raters, $n = n'$ and $r_i = \overline{r}$. What is implied here is that all subjects classified by only one rater are excluded from the analysis before Krippendorff's alpha is computed.

$$p'_o = \left(1 - \frac{1}{n'\overline{r}}\right) p_o + \left(\frac{1}{n'\overline{r}}\right)$$

which is then inserted into (1) in place of p_o to calculate Krippendorff's alpha.

2.7 Statistical inference

Gwet (2014) distinguishes broadly between a model-based and a design-based approach to statistical inference for ACs. The former is more common in the literature and relies on a hypothetical distribution of ratings under H_0 to obtain an approximate standard error for testing the coefficient against 0 (compare Fleiss, Cohen, and Everitt [1969] and Fleiss [1971]). Although Gwet (2014, 21) explicitly states that this approach is valid, it has problems. The obtained standard errors might be invalid for testing coefficients against nonzero values or for estimating confidence intervals (compare Reichenheim [2004]). Further, the distribution of some coefficients (for example, Krippendorff's alpha) remains unknown, and analytical standard errors are not available (Hayes and Krippendorff 2007). Considering the goal of statistical inference, which is generalizing results to a larger population, this larger population is not clearly defined within the model-based approach.

Gwet (2014, chap. 5), therefore, proposes a design-based approach to statistical inference that relies on the principles of finite populations. He argues that statistical inference for ACs should account for two sources of variance: Subjects might be sampled from a subject universe, and raters might be drawn from a larger rater population. Previous literature on ACs has especially neglected the sampling of raters.

Let S_n denote a specific sample of subjects, and let S_r denote a specific sample of raters.[6] The variance due to the sampling process of subjects, given a specific set of raters, is

$$V\left(\kappa_{\cdot}\middle|\, S_r\right) = \frac{1 - f_n}{n\,(n-1)} \sum_{i=1}^{n} \left(\kappa_{\cdot i} - \kappa_{\cdot}\right)^2 \tag{8}$$

where f_n is the sampling fraction of subjects. Further, $\kappa_{\cdot i}$ is defined as

$$\kappa_{\cdot i} = \frac{n}{n'} \times \frac{p_{o_i} - p_e}{1 - p_e} - 2\left(1 - \kappa_{\cdot}\right) \frac{p_{e_i} - p_e}{1 - p_e}$$

where n' is the number of subjects that are classified by two or more raters and p_{o_i} and p_{e_i} are the subject-level observed and expected proportions of agreement. The formulas for the subject-level observed and expected proportions of agreement are similar to those discussed in sections 2.5 and 2.6. For example, the subject-level observed proportion of agreement is basically given by

$$p_{o_i} = \sum_{k=1}^{q} \frac{r_{ik}\left(\sum_{l=1}^{q} w_{kl} r_{il} - 1\right)}{r_i\left(r_i - 1\right)}$$

6. Gwet (2014) discusses only the case of simple random sampling with equal selection probabilities.

where all terms are the same as those in (5).[7] See Gwet (2014, 147–149) for a detailed description of the different subject-level proportions of chance agreement.

Conditional on the sample of subjects, Gwet (2014, 153) suggests the following jackknife approach to estimate the variance due to the sampling of raters:

$$V\left(\kappa_{\cdot}\middle|\, S_n\right) = \frac{\left(1-f_r\right)\left(r-1\right)}{r}\sum_{g=1}^{r}\left(\kappa_{\cdot(g)} - \overline{\kappa_{\cdot}}\right)^2 \tag{9}$$

In (9), f_r is the sampling fraction of raters, $\kappa_{\cdot(g)}$ is the AC when rater g is omitted, and $\overline{\kappa_{\cdot}}$ is the mean of all $\kappa_{\cdot(g)}$ ACs.[8]

Depending on the target population of interest, the variance estimator in (8) may be used to generalize results to the subject universe conditional on the specific raters. Conversely, results may be projected to the population of raters given the specific sample of subjects using the jackknife estimator in (9). To generalize results to both the subject universe and the population of raters, Gwet (2014, 155–158) proposes the unconditional variance estimator given by

$$V\left(\kappa_{\cdot}\right) = V\left(\kappa_{\cdot}\middle|\, S_r\right) + V\left(\kappa_{\cdot}\middle|\, S_n\right) \tag{10}$$

Gwet (2008a,b) proves that standard errors derived from (8), (9), and (10) are valid for testing against 0 or any other value of interest and for confidence interval construction. Confidence intervals are based on the t distribution with $n-1$ degrees of freedom for standard errors obtained from (8), and the standard normal distribution is used as an approximation otherwise.

2.8 A probabilistic benchmarking method

Once the extent of agreement among raters is quantified using one of the chance-corrected ACs, researchers are usually keen to interpret and communicate the results. For this purpose, they often rely on some benchmarking scale. Table 6 shows one of the most popular benchmarking scales suggested by Landis and Koch (1977). Similar scales have been proposed by Fleiss, Levin, and Paik (2003) and Altman (1991).

7. The subject-level observed proportion of agreement is set to $p_{o_i} = 0$ for all subjects that are rated by only one of the raters.
8. Note that three or more raters are required for the jackknife estimator. There is currently no alternative approach for estimating the variance due to the sampling of (two) raters.

Table 6. Benchmark scale by Landis and Koch (1977)

Coefficient			Interpretation
	below	0.00	Poor
0.00	to	0.20	Slight
0.21	to	0.40	Fair
0.41	to	0.60	Moderate
0.61	to	0.80	Substantial
0.81	to	1.00	Almost Perfect

Gwet (2014, chap. 6) points out a fundamental flaw in the current practice of interpreting the extent of interrater agreement: Recall that ACs are point estimates; as such, they have a probability distribution and an error margin associated with them. The probability distribution and the error margin of an AC depend on the number of subjects, the number of raters, and the number of rating categories in the study. Current practice is to compare the estimated coefficient with predetermined benchmark thresholds without recognizing the variance due to subjects, raters, and rating categories. In other words, current practice ignores any uncertainty associated with the estimated AC.

Gwet criticizes current practice for being "deterministic" when benchmarking should be probabilistic in nature. As an alternative, he proposes a statistical approach that consists of the following three steps:

1. Compute the probability for a coefficient to fall into each of the benchmark intervals.
2. Compute the cumulative probability, starting from the highest benchmark level.
3. Select the first benchmark interval for which the cumulative probability exceeds a given threshold (95% by convention).

To elaborate on these three steps, let us assume that we have estimated an AC, $\kappa.$, and its variance, $V(\kappa.)$. Given these two quantities, we start with step 1 and compute the probability for our estimated coefficient to fall into each of the benchmark intervals of the Landis and Koch (1977) benchmark scale, shown in table 6. For example, the probability that $\kappa.$ falls into the highest benchmark interval, from 0.81 to 1.00, is given by

$$P(0.81 < \kappa. < 1.00) = \Phi\left(\frac{\kappa. - 0.81}{\sqrt{V(\kappa.)}}\right) - \Phi\left(\frac{\kappa. - 1.00}{\sqrt{V(\kappa.)}}\right)$$

where Φ denotes the cumulative standard normal distribution. Gwet (2014) calls this probability an interval membership probability (IMP). The IMPs for each remaining interval of the benchmarking scale are computed accordingly.

Given all IMPs from step 1, we turn to step 2. Starting from the highest benchmark interval, we compute the cumulative membership probabilities for each benchmark interval. The cumulative membership probability for a given benchmark interval is simply the (running) sum of the IMPs down to the respective benchmark interval.

Given the cumulative probabilities from step 2, we can finally turn to step 3 and select the benchmark interval that is associated with the smallest cumulative membership probability that exceeds 95%. We can then claim with 95% confidence that the extent of agreement corresponds to the selected benchmark interval.

3 Stata commands to assess interrater agreement

Before I introduce the new `kappaetc` command that implements the concepts discussed so far, I will give a short overview of existing Stata commands that address the issue of interrater agreement.[9]

3.1 Official Stata commands

Official Stata capabilities for assessing interrater agreement are limited to Cohen's (1960, 1968) and Fleiss's (1971) variants of kappa. For two raters, the `kap` command (see [R] **kappa**) estimates the former. It requires that each observation represent one subject and that variables record the ratings. Linear and quadratic weights for partial agreement are available, and users may define their own set of custom weights with the `kapwgt` command.[10] Subjects that are rated by only one of the raters are excluded from the analysis.

For three or more raters, `kap` estimates Fleiss's kappa for each rating category against all remaining categories; the overall kappa is estimated as well. Weights for partial disagreement cannot be used for three or more raters, but subjects are included in the analysis when at least one rater has classified them. The implemented method for dealing with missing ratings differs from the one discussed in section 2.4 and leads to slightly different results.

The `kappa` command requires data in the form of rating frequencies as shown in table 5. It always calculates Fleiss's unweighted kappa coefficient. Note that for two raters, `kappa` may be used to obtain Scott's (1955) pi.

9. Note that I do not discuss the intraclass correlation coefficient as a measure of interrater reliability here. This approach may be used with interval level data, and it is implemented in the `icc` command (see [R] **icc**).
10. Note that Stata's implementation of weights differs from that discussed in section 2.3. `kap` bases the weights on the integer sequence $1, 2, \ldots, q$ of rating categories, not the actual ratings.

Both official Stata commands provide a test of kappa against zero, using a model-based standard error, but neither estimates confidence intervals. Stata also lacks commands for estimating any other ACs that we have discussed. Several community-contributed additions address some of these shortcomings.[11]

3.2 Community-contributed commands

For two raters and two rating categories, `kapci` (Reichenheim 2004) calculates analytic (that is, model-based) standard errors and corresponding confidence intervals for Cohen's weighted kappa. For more than two raters, it calculates Fleiss's unweighted kappa. For three or more raters or when there are more than two rating categories, the command uses Stata's `bootstrap` (see [R] **bootstrap**) capabilities to obtain standard errors and confidence intervals. The two related commands `kappci` and `kappaci` (Harrison 2004) are restricted to binary ratings and do not support weighted disagreement in the first place. Because all three commands are based on `kap` and `kappa`, they treat missing values in the same way that Stata's native commands do.

Lazaro et al. (2013) are the first to implement Conger's (1980) kappa for three or more raters in Stata. Their `kappa2` command supports all weights used with `kap` and applies them to any number of raters.[12] It also includes all subjects rated by one or more raters, although it uses yet another approach than the one discussed in section 2.4 or the one implemented in `kap`.[13] Standard errors and confidence intervals are optionally obtained using Stata's `jackknife` (see [R] **jackknife**) procedure.

Krippendorff's (1970, 2011, 2013) alpha was first implemented in the `krippalpha` command (Staudt and Krewel 2013). The command handles any number of raters, any number of rating categories, and missing values. Partial agreement may be weighted with ordinal, quadratic, or ratio weights by specifying the respective data metric. Standard errors and confidence intervals are not currently calculated, but they may be obtained using Stata's `jackknife` or `bootstrap` prefix. The `kalpha` command (Klein 2014) adds circular and bipolar weights for partial agreement and implements a bootstrap algorithm, described by Hayes and Krippendorff (2007), to obtain confidence intervals. However, Gwet (2015) has severely criticized this algorithm and Zapf et al. (2016) provide empirical evidence for its poor performance. Therefore, the bootstrap algorithm implemented in `kalpha` should not be used to obtain confidence intervals. Finally, the `kanom` command (Mitnik 2016; Mitnik and Cumberworth 2016) is restricted to two raters and nominal ratings but provides a standard error and confidence interval-based on the delta method. Its ability to cope with complex sample designs is unique.

11. My discussion of community-contributed commands is not an exhaustive list of available software in this area, but a subjective selection of a few commands that I found useful.
12. A unique feature of `kappa2` is its support for an alternative definition of observed agreement, so-called majority agreement. It is beyond the scope of this article to discuss the concept of major agreement; the relevant references are given in the help file for `kappa2`.
13. It is beyond the scope of this article to systematically compare the three approaches. Basic simulations (not reported here) suggest that all three approaches yield similar answers if one assumes the ratings to be missing completely at random.

Currently, no community-contributed command implements the remaining two ACs proposed by Brennan and Prediger (1981) and Gwet (2008a, 2014).

❑ **Technical note**

When Stata's `bootstrap` or `jackknife` prefix is used to obtain standard errors and confidence intervals for weighted ACs, the same set of weights should be used in each replicated sample. Recall that the weights are usually based on the rating categories and that the latter are technically obtained from the observed data. Even if all conceivable ratings are observed in the full data, there is no guarantee that the same is true for each replicated sample of observations. Therefore, I recommend specifying a fixed set of custom weights when resampling methods are used even if prerecorded weights are available.

❑

4 The kappaetc command

4.1 Description

The new `kappaetc` (read: "kappa, etc."[14]) command estimates all ACs discussed in section 2 along with their standard errors and confidence intervals. It handles missing ratings and supports a variety of prerecorded weights for disagreement and user-defined weights.[15]

14. The name `kappaetc` is borrowed from `entropyetc` (Cox 2016) with approval from Nicholas Cox.
15. The `kappaetc` command has more capabilities: it performs paired t tests of correlated ACs (compare Gwet [2016]), estimates the limits of agreement and produces Bland–Altman plots (Bland and Altman 1986), and estimates intraclass correlation coefficients (for repeatedly measured subjects). The latter approaches are appropriate with interval-level ratings that are not predetermined. Because we have confined ourselves to ACs for predetermined categories, a further discussion of these capabilities is beyond the scope of this article. Gwet (2014) discusses the respective concepts in length.

4.2 Syntax

Interrater agreement, variables record raw ratings:

```
kappaetc varname1 varname2 [varname3 ...] [if] [in] [weight]
    [, wgt(wgtid [, wgt_options]) se(se_type) frequency categories(numlist)
    listwise level(#) showweights
    benchmark[([benchmark_method][, benchmark_options])] showscale
    testvalue([relop]#) nociclipped noheader notable cformat(%fmt)
    pformat(%fmt) sformat(%fmt) nsubjects(#) nraters(#) largesample]
```

Immediate command, interrater agreement, two raters, contingency table:

```
kappaetci #11 #12 [...] \ #21 #22 [...] [\ ...]
    [, wgt(wgtid [, wgt_options]) se(se_type) categories(numlist) listwise
    level(#) showweights
    benchmark[([benchmark_method][, benchmark_options])] showscale
    testvalue([relop]#) nociclipped noheader notable cformat(%fmt)
    pformat(%fmt) sformat(%fmt) nsubjects(#) nraters(#) largesample tab]
```

`fweight`s and `iweight`s are allowed; see [U] **11.1.6 weight**.

`by` is allowed only with `kappaetc`; see [D] **by**.

4.3 Options

Main options

`wgt(`*wgtid* [`,` *wgt_options*]`)` specifies that *wgtid* be used to weight disagreements. The available weights and *wgt_options* are described below.

`identity` weights are the $q \times q$ identity matrix, where q is the number of categories used to rate subjects. Identity weights are the default and result in the unweighted analysis.

`ordinal` weights are defined as

$$w_{kl} = 1 - \binom{|k-l|+1}{2} \Big/ \binom{q}{2} \;\forall\, k \neq l$$

where k and l represent the ranked categories $1, 2, \ldots, q$ and q is the number of rating categories. The *wgt_option* `krippendorff` is allowed and specifies that ordinal weights suggested by Krippendorff (2011) be used instead. Note that standard errors are not available with Krippendorff's ordinal weights.

`linear`, `quadratic`, and `radical` weights are special cases of power weights, discussed in section 2.3 and obtained by inserting $a = 1$, $a = 2$, or $a = 0.5$ into (2). The *wgt_option* `noabsolute` is allowed and specifies that weights be based on the rating categories $1, 2, \ldots, q$ instead of the actual ratings. `w` and `w2` are synonyms for `linear` and `quadratic` weights with *wgt_option* `noabsolute`. These two weights are the same as those used by `kap` (see [R] **kappa**).

`ratio` weights are defined as

$$w_{kl} = 1 - \frac{\{(x_k - x_l) \,/\, (x_k + x_l)\}^2}{\{(x_{\max} - x_{\min}) \,/\, (x_{\max} + x_{\min})\}^2} \;\forall\, k \neq l$$

where x_k and x_l refer to the observed ratings. The *wgt_option* `noabsolute` is allowed and specifies that weights be based on the rating categories $k, l = 1, 2, \ldots, q$ instead of the actual ratings.

`circular` [`pi` | `180` | #] weights are defined as

$$w_{kl} = 1 - \frac{\sin\{\mathit{angle}\,(x_k - x_l) \,/\, (x_{\max} - x_{\min} + 1)\}^2}{\max(w_{kl})} \;\forall\, k \neq l$$

where *angle* is π radians if `pi` was specified or 180° if `180` was specified and where x_k and x_l refer to the observed ratings and $\max(w_{kl})$ is the maximum of all weights. When $\# \in [0, 1]$ is specified, circular weights proposed by Warrens and Pratiwi (2016) are used instead. The latter are defined as

$$w_{kl} = \begin{cases} \# & \forall\ (k - l = 1) \vee (k - l = q - 1) \\ 1 & \forall\, k = l \\ 0 & \text{otherwise} \end{cases}$$

When # is specified, the *wgt_option* `noabsolute` is required and specifies that weights be based on the rating categories $k, l = 1, 2, \ldots, q$.

`bipolar` weights are defined as

$$w_{kl} = 1 - \frac{(x_k - x_l)^2}{\max(w_{kl})\,(x_k + x_l - 2x_{\min})\,(2x_{\max} - x_k - x_l)}$$

`power` # weights are discussed in section 2.3 and defined in (2), where $a = \#$. The *wgt_option* `noabsolute` is allowed and specifies that weights be based on the rating categories $k, l = 1, 2, \ldots, q$ instead of the actual ratings.

kapwgt and *matname* are weights defined by the `kapwgt` command (see [R] **kappa**) or in a Stata matrix. The *wgt_options* `kapwgt` and `matrix` are allowed and must be used if *kapwgt* or *matname* has the same name as any of the prerecorded weights (or their abbreviations) discussed above.

`se(`*se_type*`)` specifies how standard errors are estimated. Standard errors may be estimated conditional upon the samples of raters and the sample of subjects, or unconditional, accounting for the two respective sampling errors. Their appropriateness depends on the research questions. Available *se_type*s were discussed in section 2.7 and are described below.

`conditional [raters]` are the default standard errors and are estimated conditionally upon the sample of raters as the square root of (8). These standard errors are appropriate when results are to be generalized to the subject universe, given the specific raters.

`conditional subjects` requests that standard errors be estimated conditionally upon the sample of subjects. The extent of agreement among all but one rater is obtained for each of the r raters in the sample. Technically, these standard errors are implemented using the jackknife approach in (9). These standard errors allow projection of results to the rater population, given the rated subjects.

`unconditional` standard errors are appropriate if the results are to be projected to the universe of subjects and the rater population. They are estimated according to (10).

`frequency` specifies that variables represent rating categories. The first variable records the frequency of the first rating category, the second variable records the frequency of the second rating category, and so on. Rating categories are assumed to be the integer sequence $1, 2, \ldots, q$ (but see the option `categories()`). Note that all possible ratings must be represented by one variable even if the frequency is 0 for all subjects. Cohen's (1960, 1968) and Conger's (1980) kappa cannot be calculated from recorded rating frequencies, and only the default standard errors, conditional on the rater sample, are available.

`categories(`*numlist*`)` specifies the predetermined rating categories. By default, the set of ratings is obtained from the data. There are two situations where this option should be used.

When variables contain ratings (the default), the full set of possible rating categories must be specified if not all of them are observed in the data. Failing to do so may lead to incorrect results. The order in which rating categories are specified does not matter; categories are sorted internally. Note that noninteger values are processed in double precision. To convert them to float precision, specify `categories(float(`*numlist*`))`.

With the frequency option, the ratings are assumed to be the integer sequence $1, 2, \ldots, q$ that corresponds to the specified variables. Likewise, with the immediate form of the command, the ratings are assumed to be the integer sequence $1, 2, \ldots, q$ of rows and columns entered. In both cases, the `categories()` option may be used to specify alternative rating categories, including noninteger, negative, and even missing values. Also in both cases, the order in which the ratings are specified matters and corresponds to the respective variables or sorted values underlying the table.

`listwise` specifies that subjects with missing ratings be excluded from the analysis. By default, all subjects that are rated by at least one (two, for Krippendorff's alpha) rater or raters are used to estimate expected agreement. Observed agreement is based only on those subjects that are rated by two or more raters. `casewise` is a synonym for `listwise`.

Reporting options

`level(`*#*`)` specifies the confidence level, as a percentage, for confidence and benchmark intervals. The default is `level(95)`.

`showweights` additionally displays the weighting matrix below the coefficient table. For the unweighted analysis, the identity matrix is shown.

`benchmark`[`(`[*benchmark_method*][`,` *benchmark_options*]`)`] benchmarks the estimated interrater ACs using the Landis and Koch (1977) scale and the method discussed in section 2.8. When specified, `kappaetc` displays the estimated coefficients and their standard errors. It reports the probability for each coefficient to fall into the selected benchmark interval along with the cumulative probability exceeding the predetermined threshold associated with this benchmark interval. The benchmark interval limits are shown as well.

The two available *benchmark_method*s are described below.

`probabilistic` is the default method and selects the benchmark interval associated with the smallest cumulative membership probability exceeding `c(level)`. The threshold is controlled by the `level()` option.

`deterministic` selects the benchmark interval associated with the estimated AC. This method is deterministic in that the chosen interval is determined solely by the point estimate, ignoring any uncertainty associated with its estimation.

With both *benchmark_method*s, the following *benchmark_option* is allowed.

`scale(`*spec*`)` specifies the benchmark scale. *spec* is usually one of `landis` (or `koch`), `fleiss`, or `altman`. The default is `scale(landis)` (or `scale(koch)`) and results in the Landis and Koch scale. `fleiss` requests a three-level scale suggested by Fleiss, Levin, and Paik (2003), and `altman` collapses the first two levels of the default scale into one category yielding the Altman (1991) scale. Alternatively, *spec* explicitly specifies the (upper-limit) benchmarks as a *numlist*. The Landis and Koch scale could be obtained as `scale(0(.2)1)`.

`showscale` additionally displays the benchmark scale for interpreting coefficients. This option is ignored when `benchmark` is not specified.

`testvalue(`[*relop*]*#*`)` tests whether the estimated ACs equal *#*. The default is `testvalue(0)`. *relop* is one of the relational operators `>`[`=`] or `<`[`=`] and performs one-sided tests.

`nociclipped` reports confidence intervals as estimated. The default is to restrict confidence limits to fall into the range of $-1 <\#< 1$.

`noheader` suppresses the report about the number of subjects, ratings per subject, and rating categories. Only the coefficient table is displayed.

`notable` suppresses the display of the coefficient table.

`cformat(%`*fmt*`)` specifies how to format coefficients, standard errors, and confidence limits. The maximum format width is 8.

`pformat(%`*fmt*`)` specifies how to format p-values. The maximum format width is 5.

`sformat(%`*fmt*`)` specifies how to format test statistics. The maximum format width is 6.

Advanced options

`nsubjects(#)` specifies the size of the subject universe to be used for the finite sample correction. The default is `nsubjects(.)`, leading to a sampling fraction of 0 that is assumed to be negligible. This option is seldom used.

`nraters(#)` specifies the size of the rater population to be used for the finite sample correction. The default is `nraters(.)`, leading to a sampling fraction of 0 that is assumed to be negligible. This option is relevant only for standard errors that are conditional on the sample of subjects or unconditional standard errors. It is seldom used, although the default might overestimate the variance for small rater populations.

`largesample` specifies that the calculation of p-values and intervals be based on the standard normal distribution rather than the t distribution. This is the default for unconditional standard errors. `largesample` is a reporting option and it is seldom used.

Immediate command

`tab` displays the two-way table of cell frequencies. The option is useful for data entry verification.

5 Examples of the kappaetc command

In this section, I demonstrate how to use the `kappaetc` command to estimate the interrater agreement. Starting with a simple example, I will also cover known paradoxes with the kappa coefficient and discuss how the two newly implemented coefficients provide more robust estimates. Throughout this section, we use our generic symbol, $\kappa.$, to refer to different ACs. To be more specific, we denote Cohen's and Conger's kappa κ_κ, Fleiss's kappa κ_π, Brennan and Prediger's coefficient κ_q, Gwet's AC κ_γ, and Krippendorff's alpha κ_α.

5.1 Kappa and other chance-corrected ACs

▷ Example 1

We consider two radiologists who classify 85 xeromammograms into one of four categories (see also example 1 in [R] **kappa**).

```
. webuse rate2
(Altman p. 403)
. tabulate rada radb
```

Radiologist A's assessment	Radiologist B's assessment				
	Normal	benign	suspect	cancer	Total
Normal	21	12	0	0	33
benign	4	17	1	0	22
suspect	3	9	15	2	29
cancer	0	0	0	1	1
Total	28	38	16	3	85

The two radiologists' ratings are recorded in the two variables `rada` and `radb`, and each of the 85 observations represents one xeromammogram.

We can estimate agreement between the two radiologists with the `kappaetc` command as follows:

```
. kappaetc rada radb
Interrater agreement                            Number of subjects =      85
                                                  Ratings per subject =       2
                                       Number of rating categories =       4
```

	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
Percent Agreement	0.6353	0.0525	12.10	0.000	0.5309	0.7397
Brennan and Prediger	0.5137	0.0700	7.34	0.000	0.3745	0.6530
Cohen/Conger's Kappa	0.4728	0.0731	6.46	0.000	0.3273	0.6182
Scott/Fleiss' Pi	0.4605	0.0781	5.89	0.000	0.3051	0.6159
Gwet's AC	0.5292	0.0679	7.80	0.000	0.3942	0.6642
Krippendorff's Alpha	0.4637	0.0781	5.93	0.000	0.3083	0.6191

The output header summarizes the study parameters: there are 85 subjects, each of which received 2 ratings from a set of 4 possible rating categories. The first row of the coefficient table, labeled `Percent Agreement`, contains the observed proportion of agreement. We find that the two radiologists agree on 63.5% of the subjects (that is, the xeromammograms) that they have classified. Cohen's kappa coefficient, reported in the third row, corrects observed agreement for chance and is estimated as $\kappa_\kappa = 0.473$. Both Fleiss's extension of Scott's pi ($\kappa_\pi = 0.461$) and Krippendorff's alpha ($\kappa_\alpha = 0.464$) yield results that are similar to kappa. Moreover, the pi coefficient and Krippendorff's alpha are almost identical to each other; this should not be surprising given their near-identical mathematical formulation discussed in section 2.6. Slightly higher values are obtained

for the Brennan and Prediger coefficient ($\kappa_q = 0.514$) and Gwet's AC ($\kappa_\gamma = 0.529$). The observed differences should not be surprising either; while the coefficients by Cohen, Fleiss, Scott, and Krippendorff all calculate chance agreement based on the marginal distributions, both Brennan and Prediger's coefficient and Gwet's coefficient account for the number of rating categories.

The standard errors, reported by `kappaetc`, are estimated conditional on the two radiologists according to (8). Like the chance-corrected ACs, all standard errors are highly similar. According to the corresponding p-values, all coefficients are statistically significant at any level. Altogether, all chance-corrected ACs yield quite similar answers as will usually be the case (compare Feng [2013]). We will discuss some known exceptions in the next examples.

◁

5.2 Problems with the kappa coefficient

▷ **Example 2**

Example 1 showed only minor differences between the various chance-corrected ACs. Although this will often be the case, the differences between chance-corrected ACs can become quite large in certain situations. Consider the rating frequencies of two raters who classify 125 subjects into one of two categories (compare Gwet [2008a]). The fictional data are shown in table 7.

Table 7. Distribution of 125 subjects by rating category and rater

	Rater B		
Rater A	1	2	Total
1	118	5	123
2	2	0	2
Total	120	5	125

To estimate interrater agreement, we use the immediate form of `kappaetc` (see [U] **19 Immediate commands**). We specify the `tab` option to verify that we have entered the data correctly:

```
. kappaetci 118 5 \ 2 0, tab
```

row	col 1	2	Total
1	118 94.40	5 4.00	123 98.40
2	2 1.60	0 0.00	2 1.60
Total	120 96.00	5 4.00	125 100.00

```
Interrater agreement                            Number of subjects =     125
                                                Ratings per subject =      2
                                      Number of rating categories =        2
```

	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
Percent Agreement	0.9440	0.0206	45.72	0.000	0.9031	0.9849
Brennan and Prediger	0.8880	0.0413	21.50	0.000	0.8063	0.9697
Cohen/Conger´s Kappa	-0.0234	0.0123	-1.90	0.060	-0.0478	0.0010
Scott/Fleiss´ Pi	-0.0288	0.0109	-2.64	0.009	-0.0504	-0.0072
Gwet´s AC	0.9408	0.0231	40.80	0.000	0.8951	0.9864
Krippendorff´s Alpha	-0.0247	0.0109	-2.26	0.026	-0.0463	-0.0031

The two raters agree on 94.4% of the subjects that they have classified. Despite this high-observed proportion of agreement, Cohen's kappa coefficient is an estimated negative value of $\kappa_\kappa = -0.023$, indicating that the agreement between the two raters is worse than would have been expected by chance. Clearly, this result is counter-intuitive. Feinstein and Cicchetti (1990) discuss this situation as the so-called high-agreement but low-kappa paradox. Note that both Fleiss and Scott's pi ($\kappa_\pi = -0.029$) and Krippendorff's alpha ($\kappa_\alpha = -0.025$) yield similar results; their respective 95% confidence intervals do not even include 0, meaning that the estimated negative values are statistically significant at the 5% level. Notably higher estimates are obtained for the Brennan and Prediger coefficient ($\kappa_q = 0.888$) and Gwet's AC ($\kappa_\gamma = 0.941$). The last two results are more in line with the observed proportion of agreement and arguably represent the data more accurately.

◁

▷ Example 3

A second paradox pointed out by Feinstein and Cicchetti (1990) is kappa's dependency on the marginal distribution. The authors compare two sets of results to illustrate the problem. In both cases, two raters classify 100 subjects into one of two categories. Table 8 illustrates the resulting distributions of subjects by raters and rating category.

Table 8. Two distributions of 100 subjects by rating category and rater

	Rater B		
Rater A	1	2	Total
1	45	15	60
2	25	15	40
Total	70	30	100

	Rater B		
Rater A	1	2	Total
1	25	35	60
2	5	35	40
Total	30	70	100

Before we evoke `kappaetc`, let us compare the two marginal distributions. In the left panel of table 8, the marginal counts are close, that is, balanced, while in the right panel, the raters agree less on the marginal counts. Now let us estimate chance-corrected agreement for the two contingency tables with `kappaetc`. To save space, we specify the `noheader` option.

```
. kappaetci 45 15 \ 25 15, noheader
```

	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
Percent Agreement	0.6000	0.0492	12.19	0.000	0.5023	0.6977
Brennan and Prediger	0.2000	0.0985	2.03	0.045	0.0046	0.3954
Cohen/Conger´s Kappa	0.1304	0.0992	1.32	0.191	-0.0663	0.3272
Scott/Fleiss´ Pi	0.1209	0.1017	1.19	0.238	-0.0810	0.3228
Gwet´s AC	0.2661	0.1039	2.56	0.012	0.0599	0.4723
Krippendorff´s Alpha	0.1253	0.1017	1.23	0.221	-0.0766	0.3272

```
. kappaetci 25 35 \ 5 35, noheader
```

	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
Percent Agreement	0.6000	0.0492	12.19	0.000	0.5023	0.6977
Brennan and Prediger	0.2000	0.0985	2.03	0.045	0.0046	0.3954
Cohen/Conger´s Kappa	0.2593	0.0775	3.34	0.001	0.1054	0.4131
Scott/Fleiss´ Pi	0.1919	0.0989	1.94	0.055	-0.0044	0.3882
Gwet´s AC	0.2079	0.0995	2.09	0.039	0.0105	0.4054
Krippendorff´s Alpha	0.1960	0.0989	1.98	0.050	-0.0003	0.3922

Notice that the observed proportion of agreement is the same for both tables: $p_o = 0.60$. Notice also that the kappa coefficient for the left panel of table 8 ($\kappa_\kappa = 0.130$) is only half the size of the kappa coefficient for the right panel ($\kappa_\kappa = 0.259$). Given the same observed proportion of agreement, it appears as if the kappa coefficient punishes raters for agreeing on the marginal distribution. The differences between the respective coefficients proposed by Scott and Fleiss, Krippendorff, and Gwet are slightly less sensitive to shifts in the marginal distribution;[16] the Brennan and Prediger coefficient is not affected at all.

◁

16. Gwet's coefficient actually yields a higher value when the marginal distributions are balanced. See Feng (2013) for a mathematical explanation.

▷ Example 4

A third, less well-known paradox of kappa has recently been pointed out by Warrens (2012). In section 2.3, we discussed quadratic weights to account for partial agreement among raters. Warrens (2012) identifies a severe problem in certain situations when the quadratic weighted kappa is applied. Consider the following data, where 2 raters classify 30 subjects into 1 of 3 predetermined categories.

Table 9. Distribution of 30 subjects by rating category and rater

	Rater B			
Rater A	1	2	3	Total
1	1	15	1	17
2	3	0	3	6
3	2	3	2	7
Total	6	18	6	30

From table 9, we can tell that the two raters agreed on the classification of 3 out of 30 subjects, leading to an (unweighted) observed proportion of agreement of 10%. The count of 0 in the center cell of table 9 indicates that the raters did not agree to classify any subjects into category 2. Now, assume that the three rating categories in table 9 are ordered so that adjacent categories imply a less serious disagreement than nonadjacent categories. Suppose further that we wish to reflect the partial agreement between the two raters that is implied by the ordered nature of the rating categories. Therefore, we decide to apply quadratic weights. We request quadratic weights with `kappaetc`'s `wgt()` option. To inspect the weighting matrix that `kappaetc` uses, we also specify the `showweights` option.

```
. kappaetci 1 15 1 \ 3 0 3 \ 2 3 2, wgt(quadratic) showweights
Interrater agreement                            Number of subjects =      30
(weighted analysis)                             Ratings per subject =      2
                                      Number of rating categories =       3
```

	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
Percent Agreement	0.7000	0.0455	15.39	0.000	0.6070	0.7930
Brennan and Prediger	0.1000	0.1365	0.73	0.470	-0.1791	0.3791
Cohen/Conger´s Kappa	-0.0000	0.1663	-0.00	1.000	-0.3402	0.3402
Scott/Fleiss´ Pi	-0.0485	0.1648	-0.29	0.770	-0.3855	0.2884
Gwet´s AC	0.1523	0.1437	1.06	0.298	-0.1416	0.4461
Krippendorff´s Alpha	-0.0311	0.1648	-0.19	0.852	-0.3680	0.3059

```
Weighting matrix (quadratic weights)
  1.0000  0.7500  0.0000
  0.7500  1.0000  0.7500
  0.0000  0.7500  1.0000
```

The output header confirms that we look at weighted analysis, and `kappaetc` has printed the quadratic weighting matrix below the coefficient table as requested. The quadratic weighted observed agreement, denoted $p_{o_{w2}}$, is $p_{o_{w2}} = 0.70$, and the weighted kappa coefficient is $\kappa_{\kappa_{w2}} = -0.00$.

Compare the above results with the situation illustrated in table 10.

Table 10. Distribution of 30 subjects by rating category and rater

	Rater B			
Rater A	1	2	3	Total
1	1	1	1	3
2	3	17	3	23
3	2	0	2	4
Total	6	18	6	30

Here the two raters agree on the classification of 20 out of 30 subjects, yielding an (unweighted) observed agreement of 66.67%. Focusing on the center cell of table 10, we see that 17 subjects were classified into category 2 by both raters. These numbers suggest a much higher agreement than table 9.

```
. kappaetci 1 1 1 \ 3 17 3 \ 2 0 2, wgt(quadratic) noheader
```

	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
Percent Agreement	0.8417	0.0556	15.15	0.000	0.7280	0.9553
Brennan and Prediger	0.5250	0.1667	3.15	0.004	0.1841	0.8659
Cohen/Conger´s Kappa	-0.0000	0.2596	-0.00	1.000	-0.5310	0.5310
Scott/Fleiss´ Pi	-0.0009	0.2611	-0.00	0.997	-0.5350	0.5332
Gwet´s AC	0.6939	0.1421	4.88	0.000	0.4032	0.9845
Krippendorff´s Alpha	0.0158	0.2611	0.06	0.952	-0.5183	0.5499

As expected, the weighted agreement increases to $p_{o_{w2}} = 0.84$. Yet the quadratic weighted kappa is exactly the same as before, $\kappa_{\kappa_{w2}} = -0.00$. This paradox turns out to be systematic; Warrens (2012) proves that, under certain conditions, the quadratic weighted kappa does not depend on the center cell or middle column and row of the contingency table. Note that neither Fleiss and Scott's pi nor Krippendorff's alpha accurately reflects the increase in observed agreement. In contrast, Brennan and Prediger's coefficient and Gwet's AC seem to capture the differences much better.

◁

5.3 Multiple raters, missing values, and benchmarking coefficients

▷ Example 5

For the next example, suppose that 5 raters classify 10 subjects into 1 of 3 rating categories (see also example 9 in [R] **kappa**).

```
. webuse rvary2, clear
. list, separator(0) noobs
```

subject	rater1	rater2	rater3	rater4	rater5
1	1	2	2	.	2
2	1	1	3	3	3
3	3	3	3	3	3
4	1	1	1	1	3
5	1	1	1	3	3
6	1	2	2	2	2
7	1	1	1	1	1
8	2	2	2	2	3
9	1	3	.	.	3
10	1	1	1	3	3

Note that raters 3 and 4 did not classify all subjects. We can still estimate the agreement among multiple raters in the presence of missing values with `kappaetc`.

```
. kappaetc rater1-rater5
Interrater agreement                            Number of subjects =      10
                                      Ratings per subject: min =       3
                                                           avg =     4.7
                                                           max =       5
                                    Number of rating categories =       3
```

	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
Percent Agreement	0.5833	0.0759	7.69	0.000	0.4117	0.7550
Brennan and Prediger	0.3750	0.1138	3.29	0.009	0.1175	0.6325
Cohen/Conger´s Kappa	0.3854	0.1047	3.68	0.005	0.1485	0.6224
Scott/Fleiss´ Kappa	0.3586	0.1207	2.97	0.016	0.0856	0.6316
Gwet´s AC	0.3829	0.1145	3.34	0.009	0.1238	0.6420
Krippendorff´s Alpha	0.3897	0.1226	3.18	0.011	0.1122	0.6671

The output header provides essentially the same information that we have seen before. Note that the number of subjects is 10, meaning that all observations are used in the analysis. The ratings per subject now vary between 3 and 5 with an average of 4.7. The observed proportion of agreement is estimated as 58.3%, while the different chance-corrected ACs range from 0.359 for Fleiss's kappa to 0.390 for Krippendorff's alpha. All coefficients are statistically significant at the 5% level.

Using the Landis and Koch (1977) scale, discussed in section 2.8, we would be tempted to conclude that the chance-corrected coefficients all indicate a `Fair` amount of agreement because they all fall into the interval $0.2 < \kappa. < 0.4$. However, the confidence

intervals associated with the respective point estimates tell a quite different story. The upper and lower limits span almost over the entire scale, ranging from `Substantial` agreement all the way down to `Slight` agreement. When we decide to take the point estimates as fixed and compare them with a given threshold, we are ignoring a large amount of uncertainty. The `kappaetc` command implements the probabilistic benchmarking approach, discussed in section 2.8, which accounts for this uncertainty. Let us determine the benchmark level for our estimated coefficients.

```
. kappaetc, benchmark showscale noheader
```

	Coef.	Std. Err.	P in.	P cum. >95%	Probabilistic [Benchmark	Interval]
Percent Agreement	0.5833	0.0759	0.57	0.980	0.4000	0.6000
Brennan and Prediger	0.3750	0.1138	0.07	0.995	0.0000	0.2000
Cohen/Conger´s Kappa	0.3854	0.1047	0.05	0.997	0.0000	0.2000
Scott/Fleiss´ Kappa	0.3586	0.1207	0.10	0.992	0.0000	0.2000
Gwet´s AC	0.3829	0.1145	0.07	0.995	0.0000	0.2000
Krippendorff´s Alpha	0.3897	0.1226	0.07	0.994	0.0000	0.2000

```
Benchmark scale
      <0.0000      Poor
0.0000-0.2000      Slight
0.2000-0.4000      Fair
0.4000-0.6000      Moderate
0.6000-0.8000      Substantial
0.8000-1.0000      Almost Perfect
```

Note that we did not specify a variable list in the above code. Although `kappaetc` is an r-class program (see [P] **return**), it mimics some of the features that are typical for estimation commands (see [U] **20 Estimation and postestimation commands**). We can retype `kappaetc` without arguments to redisplay its latest results, and we can add reporting options to modify the output. Here we specified the `benchmark` option to determine the probabilistic benchmark level for our coefficients. We also specified `showscale`, so the benchmarking scale will be printed below the coefficient table.

The first two columns of the coefficient table are just the same as before. The remaining columns correspond to steps 1 to 3 of the algorithm described in section 2.8. The third column, labeled `P in.`, corresponds to step 1 and holds the estimated IMPs, that is, the probability for the respective coefficient to fall into the selected interval. The column `P cum. >95%` represents step 2 and shows the cumulative IMP, starting from the highest benchmark level, that exceeds 95%. Finally, the last two columns show the selected benchmark intervals from step 3. Here we can claim with 99.7% certainty that Conger's kappa indicates (at least) a `Slight` extent of agreement.

Notice that for all chance-corrected coefficients, except Fleiss's kappa, the probability to fall into the selected interval is below 0.1. It would be interesting to know the confidence level associated with the more traditional, "deterministic" benchmarking approach. Let us find out.

```
. kappaetc, benchmark(deterministic) noheader
```

	Coef.	Std. Err.	P in.	P cum.	Deterministic [Benchmark	Interval]
Percent Agreement	0.5833	0.0759	0.57	0.980	0.4000	0.6000
Brennan and Prediger	0.3750	0.1138	0.51	0.921	0.2000	0.4000
Cohen/Conger´s Kappa	0.3854	0.1047	0.50	0.945	0.2000	0.4000
Scott/Fleiss´ Kappa	0.3586	0.1207	0.52	0.889	0.2000	0.4000
Gwet´s AC	0.3829	0.1145	0.49	0.927	0.2000	0.4000
Krippendorff´s Alpha	0.3897	0.1226	0.45	0.921	0.2000	0.4000

Above, we have requested the `deterministic` benchmarking method. We see that all coefficients are now placed into the benchmark interval that encloses the respective point estimate, ignoring any uncertainty associated with its estimation. Looking at the cumulative probabilities for the chance-corrected coefficients, we find that, except for Fleiss's kappa, we could claim with 90% certainty or more that the extent of agreement is `Fair`.

Some researchers (for example, Hayes and Krippendorff [2007]) have suggested computing the probability that an AC fails to reach some required minimum level. This approach seems most popular when Krippendorff's alpha is estimated. Typically, a minimum of $\kappa_\alpha > 0.8$ or at least $\kappa_\alpha > 0.67$ is recommended. We can modify the t test that `kappaetc` reports by default to reflect the corresponding null hypothesis $H_0 : \kappa_\cdot < 0.67$.

```
. kappaetc, testvalue(< 0.67) noheader
```

	Coef.	Std. Err.	t	P>t	[95% Conf.	Interval]
Percent Agreement	0.5833	0.0759	-1.14	0.859	0.4117	0.7550
Brennan and Prediger	0.3750	0.1138	-2.59	0.985	0.1175	0.6325
Cohen/Conger´s Kappa	0.3854	0.1047	-2.72	0.988	0.1485	0.6224
Scott/Fleiss´ Kappa	0.3586	0.1207	-2.58	0.985	0.0856	0.6316
Gwet´s AC	0.3829	0.1145	-2.51	0.983	0.1238	0.6420
Krippendorff´s Alpha	0.3897	0.1226	-2.29	0.976	0.1122	0.6671

```
t test Ho: Coef. <=    0.6700  Ha: Coef. >    0.6700
```

Below the coefficient table, `kappaetc` now displays the null hypothesis and the alternative. The reported t statistics are calculated as

$$t = \frac{\kappa_\cdot - \kappa_\cdot^{H_0}}{\sqrt{V\left(\kappa_\cdot \middle| S_r\right)}}$$

where $\kappa_\cdot^{H_0}$ is the coefficient under H_0 and $V\left(\kappa_\cdot \middle| S_r\right)$ is defined in (8). The associated p-values for the one-sided tests indicate that we cannot reject the null hypothesis at any conventional level.

◁

5.4 Unconditional standard errors

▷ **Example 6**

The standard errors that we have estimated throughout examples 1 to 5 were conditional upon the specific sample of raters, as is usually the case in the interrater agreement literature. Thus, we can project our results only to a larger subject universe. We may also wish to generalize our findings to the population of raters. Therefore, we estimate unconditional standard errors, defined in (10).

```
. kappaetc rater1-rater5, se(unconditional) noheader
```

	Unconditional					
	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
Percent Agreement	0.5833	0.1738	3.36	0.001	0.2427	0.9240
Brennan and Prediger	0.3750	0.2607	1.44	0.150	-0.1359	0.8859
Cohen/Conger´s Kappa	0.3854	0.2428	1.59	0.112	-0.0904	0.8613
Scott/Fleiss´ Kappa	0.3586	0.2717	1.32	0.187	-0.1740	0.8911
Gwet´s AC	0.3829	0.2576	1.49	0.137	-0.1219	0.8877
Krippendorff´s Alpha	0.3897	0.2381	1.64	0.102	-0.0769	0.8563

Note that the estimated standard errors are much larger than before, and none of the chance-corrected ACs is statistically significant at the conventional 5% level.

◁

6 Conclusion

In this article, I discussed Gwet's (2014) recently developed general framework for assessing interrater agreement and introduced the `kappaetc` command, which implements it in Stata. All discussed ACs apply to the case of multiple raters, multiple rating categories, any level of measurement, and in the presence of missing values. Standard errors are estimated based on concepts for finite population inference, and the uncertainty associated with the estimated coefficients is accounted for by a probabilistic benchmarking method.

Examples of paradoxes, observed with kappa, have shown that different ACs might yield different results for the same underlying data. Although a more systematic evaluation of these differences is beyond the scope of this article (see Feng [2013] and Wongpakaran et al. [2013] for examples), the `kappaetc` command is a useful tool for such investigations. Researchers with a substantial interest in interrater agreement should check the robustness of their findings with respect to the AC of their choice. `kappaetc` provides an easy way of doing just that and, additionally, makes this choice independent of study parameters such as the number of raters or the data's level of measurement.

7 Acknowledgments

I thank Kilem Gwet for continuous support and patiently clarifying my questions during the implementation of the software. I also thank an anonymous reviewer for valuable comments and suggestions that helped improve the manuscript.

8 References

Altman, D. G. 1991. *Practical Statistics for Medical Research*. London: Chapman & Hall.

Bennett, E. M., R. Alpert, and A. C. Goldstein. 1954. Communications through limited-response questioning. *Public Opinion Quarterly* 18: 303–308.

Bland, J. M., and D. G. Altman. 1986. Statistical methods for assessing agreement between two methods of clinical measurement. *Lancet* 327: 307–310.

Brennan, R. L., and D. J. Prediger. 1981. Coefficient kappa: Some uses, misuses, and alternatives. *Educational and Psychological Measurement* 41: 687–699.

Byrt, T., J. Bishop, and J. B. Carlin. 1993. Bias, prevalence and kappa. *Journal of Clinical Epidemiology* 46: 423–429.

Cohen, J. 1960. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement* 20: 37–46.

———. 1968. Weighted kappa: Nominal scale agreement provision for scaled disagreement or partial credit. *Psychological Bulletin* 70: 213–220.

Conger, A. J. 1980. Integration and generalization of kappas for multiple raters. *Psychological Bulletin* 88: 322–328.

Cox, N. J. 2016. entropyetc: Stata module for entropy and related measures for categories. Statistical Software Components S458272, Department of Economics, Boston College. https://ideas.repec.org/c/boc/bocode/s458272.html.

Feinstein, A. R., and D. V. Cicchetti. 1990. High agreement but low kappa: I. The problems of two paradoxes. *Journal of Clinical Epidemiology* 43: 543–549.

Feng, G. C. 2013. Factors affecting intercoder reliability: A Monte Carlo experiment. *Quality and Quantity* 47: 2959–2982.

Fleiss, J. L. 1971. Measuring nominal scale agreement among many raters. *Psychological Bulletin* 76: 378–382.

Fleiss, J. L., J. Cohen, and B. S. Everitt. 1969. Large sample standard errors for kappa and weighted kappa. *Psychological Bulletin* 72: 323–327.

Fleiss, J. L., B. Levin, and M. C. Paik. 2003. *Statistical Methods for Rates and Proportions*. 3rd ed. Hoboken, NJ: Wiley.

Gwet, K. L. 2008a. Computing inter-rater reliability and its variance in the presence of high agreement. *British Journal of Mathematical and Statistical Psycholgy* 61: 29–48.

———. 2008b. Variance estimation of nominal-scale inter-rater reliability with random selection of raters. *Psychometrika* 73: 407–430.

———. 2014. *Handbook of Inter-Rater Reliability: The Definitive Guide to Measuring the Extent of Agreement Among Raters*. 4th ed. Gaithersburg, MD: Advanced Analytics.

———. 2015. Standard error of Krippendorff's alpha coefficient. K. Gwet's Inter-Rater Reliability Blog. http://inter-rater-reliability.blogspot.de/2015/08/standard-error-of-krippendorffs-alpha.html.

———. 2016. Testing the difference of correlated agreement coefficients for statistical significance. *Educational and Psychological Measurement* 76: 609–637.

Harrison, D. 2004. kaputil: Stata module to generate confidence intervals and sample size calculations for the kappa-statistic. Statistical Software Components S446501, Department of Economics, Boston College. https://ideas.repec.org/c/boc/bocode/s446501.html.

Hayes, A. F., and K. Krippendorff. 2007. Answering the call for a standard reliability measure for coding data. *Communication Methods and Measures* 1: 77–89.

Klein, D. 2014. kalpha: Stata module to compute Krippendorff's alpha-reliability. Statistical Software Components S457862, Department of Economics, Boston College. https://ideas.repec.org/c/boc/bocode/s457862.html.

Krippendorff, K. 1970. Estimating the reliability, systematic error and random error of interval data. *Educational and Psychological Measurement* 30: 61–70.

———. 2011. Computing Krippendorff's alpha-reliability. https://repository.upenn.edu/asc_papers/43/.

———. 2013. *Content Analysis: An Introduction to Its Methodology*. 3rd ed. Thousand Oaks, CA: Sage.

Landis, J. R., and G. G. Koch. 1977. The measurement of observer agreement for categorical data. *Biometrics* 33: 159–174.

Lazaro, J., J. Zamora, V. Abraira, and A. Zlotnik. 2013. kappa2: Stata module to produce generalizations of weighted kappa for incomplete designs. Statistical Software Components S457739, Department of Economics, Boston College. https://ideas.repec.org/c/boc/bocode/s457739.html.

Mitnik, P. 2016. kanom: Stata module to estimate Krippendorff's alpha for nominal variables. Statistical Software Components S458277, Department of Economics, Boston College. https://ideas.repec.org/c/boc/bocode/s458277.html.

Mitnik, P., and E. Cumberworth. 2016. Measuring social class with changing occupational classifications: Reliability, competing measurement strategies, and the 1970–1980 U.S. classification divide. Working Paper, Stanford Center on Poverty and Inequality. https://web.stanford.edu/~pmitnik/Mitnik_Cumberworth_2016.pdf.

Reichenheim, M. E. 2004. Confidence intervals for the kappa statistic. *Stata Journal* 4: 421–428.

Scott, W. A. 1955. Reliability of content analysis: The case of nominal scale coding. *Public Opinion Quarterly* 19: 321–325.

Staudt, A., and M. Krewel. 2013. krippalpha: Stata module to compute Krippendorff's alpha intercoder reliability coefficient. Statistical Software Components S457750, Department of Economics, Boston College. https://ideas.repec.org/c/boc/bocode/s457750.html.

Warrens, M. J. 2012. Some paradoxical results for the quadratically weighted kappa. *Psychometrika* 77: 315–323.

———. 2014. Power weighted versions of Bennett, Alpert, and Goldstein's S. *Journal of Mathematics* 2014: 231909.

Warrens, M. J., and B. C. Pratiwi. 2016. Kappa coefficients for circular classifications. *Journal of Classification* 33: 507–522.

Wongpakaran, N., T. Wongpakaran, D. Wedding, and K. L. Gwet. 2013. A comparison of Cohen's kappa and Gwet's AC1 when calculating inter-rater reliability coefficients: A study conducted with personality disorder samples. *BMC Medical Research Methodology* 13: 61.

Zapf, A., S. Castell, L. Morawietz, and A. Karch. 2016. Measuring inter-rater reliability for nominal data—Which coefficients and confidence intervals are appropriate? *BMC Medical Research Methodology* 16: 93.

About the author

Daniel Klein is a research assistant at the International Centre for Higher Education Research Kassel. He is a sociologist with a special interest in social inequality, theoretical models of decision making, and quantitative methods of empirical social research.