Picking the Best Intercoder Reliability Statistic for Your Digital Activism Content Analysis

TL;DR Version: Use percent agreement, Scott's pi, and Krippendorff's alpha for studies using two coders. Use percent agreement, Fleiss' kappa, and Krippendorff's alpha for studies using three or more coders.

* * * *

[UPDATED] Stories of digital activism, created by both citizen and professional journalists, are shared freely on the web. Content created by the activists themselves – webpages, video, Facebook groups – also often remain online and public long after their campaigns end. Because of the wide available of freely accessible content, content analysis has become a valuable method for studying global digital activism.

What is Intercoder Reliability?

According to Kimberly Neuendorf, author of the popular educational text The Content Analysis Guidebook, content analysis can be defined as "the systematic, objective quantitative analysis of message characteristics" (p.1). Often it involves trained analysts, called coders, analyzing text, video, or audio and describing the content according to a group of open-ended (write-in) and close-ended (multiple choice) variables. For the Global Digital Activism Data Set (GDADS) we are reviewing mostly textual sources (and the occasional video) that describe instances of digital activism around the world. Our central sources are accounts created by reliable third party sources, either citizen or professional journalists. These account are augmented by analysis of materials created by the activists themselves, such as website and Facebook pages.

Coders read sources assigned to them, and code for the variables in the coding scheme in this codebook, entering their answers on a Google form. The goal is that all coders code the same content with the same value. For example, if they read about a campaign which targeted the municipal police of Kuala Lumpur they would correctly code the target country as Malaysia and not Indonesia or Sri Lanka. When coders agree about how to code a piece of content, that is an indicator, though not a guarantee of *reliability*, the trustworthiness of the data. According to Klaus Krippendorf, the foremost living expert on intercoder reliability in content analysis, "agreement is what we measure; reliability is what we wish to infer from it" (2004a, p. 215).

Why Measure Intercoder Reliability?

The reason we measure reliability is to demonstrate the *trustworthiness* of data, but when we measure reliability we are actually measuring *reproducibility:* the likelihood that different coders who receive the same training and textual guidance will assign the same value to the same piece of content. According to Krippendorff, "(i)n content analysis, reproducibility is arguably the most important interpretation of reliability" (2004a, p. 215).

If you do not measure intercoder agreement, you do not know if your data is reliable, if the conclusion that result from its analysis are accurate or misleading. A sense of your data being right, which Krippendorff calls "face reliability" is meaningless because it is extremely subjective (2004b, p. 413). On a more practical level, if your data does not include accepted levels of agreement according to measures, no journal will publish your data. Your data will be reliable only to people who are ignorant, a pretty low standard for research.

How to Establish Intercoder Reliability

When you have high intercoder reliability it means that different coders perceive a piece of content in the same way and code it accordingly. So how do you establish this agreement? Because coders may come to the coding process with different experiential and intellectual backgrounds, training and a clear codebook are important. Though a codebook (the list of variables, their values, and their definitions) is an important part of research documentation, coders are unlikely to refer to it often. They are more likely to retain the verbal training that they receive at the beginning of the project and to look at the instructional prompts on the coding form, which they read as they enter codes. It may also be helpful to have the project manager sit in the same room as the coders for at least the beginning of the coding process so that the coders can ask questions about how to code ambiguous cases and clarify misunderstanding of the codebook or coding process.

It is also useful to calculate intercoder reliability statistics on an ongoing basis throughout the coding process, not only during the training phase. This allows coders to have constant feedback on the quality of their coding and for you, the researcher, to be immediately aware of any coding problems so you can fix them. A good way to do this ongoing computation of agreement is to give coders a percentage of cases (the GDADS uses 20%) which are coded by all coders. You the researcher code these cases as well. If you will act as a coder, include your codes in the computation of intercoder reliability (hereafter ICR). If you will not be coding, exclude them and code only so you have familiarity with the case. Then review the cases on Monday with the coders, identifying areas of disagreement and clarifying the right answer or clarifying the codebook as needed. For the GDADS we code 4 cases together each week and it takes about an hour for us to review them each week. Though ICR statistics are designed only to deal with close-ended numeric answers, you can also convert your own-ended textual variables into dummies to get a rough idea of agreement.

In order to calculate an intercoder reliability statistic all coders need to code the same case so that a direct comparison can be made. However, having all coders code the same case also means lost time since if one case is coded three times by three coders the result is still only 1 new coded case, not 3. For example, if you have 2 coders coding 10 cases each with 20% overlap, each coder will code 10 cases, but you will end up with only 18 cases coded. This is because 16 cases were coded once (80% of the assignment), but 2 cases were coded twice so that intercoder reliability statistics could be calculated (20% of the assignment). You can calculate the number of final coded cases with a given multiple-coding rate by using the following equation. This equation will allow you to know how many new coded cases will be coded get if assigne X% of cases to all coders in order to calculate ICR statistics.

> *Figure 1: Calculating Total Cases Coded*
>
> *(multiple-coding percentage) x (number of cases/coder) + (1 – multiple percentage) x (number of cases/coder) x (number of coders)*
> ↓ ↓
> *New cases resulting from the multiple-coding + New cases resulting from normal single coding*
>
> *For example, for a 20% double-coding rate, 10 cases/coder, and 2 coders the equation is:*
> *(.2) x (10) + (1-.2) x (10) x (2) =*
> *(2) + (16) = 18 new coded cases*
>
> *For a 50% triple-coding rate, 10 cases/coder, and 3 coders, the equation is :*
> *(.5) x (10) + (1 – .5) x (10) x (3) =*
> *(5) + (15) = 20 new coded cases*

Notice that in the second example, even with more coders, choosing to code a higher rate of cases more that once

means that less cases case coded in the end.

In the second version of the Global Digital Activism Data Set (GDADS), we are calculating this statistic for 20% of coded cases every week and then reviewing those cases with the coders the following week. This allows us to know on a week to week basis to what extent we agree with one another, to see on which variable agreement is increasing (the goal), where is it static over time (good or bad, depending on whether or not the agreement is high), and where is is falling (definitely not good and needs to be actively remedied). According to Lombard, Snyder-Duch, and Bracken (2002), I am aiming for agreement of over 80% for all close-ended variables. Even after I achieve this rate, I continue to calculate the statistic on a weekly basis to ensure that the value remains high.

There is no clear standard for the percentage of content units one should recode in order to calculate agreement rates. In some content analysis 64% or even 100% of observations are coded multiple times (Lombard, Snyder-Duch, and Bracken, 2002; Reichert et al., 1999). In these cases the observations were journal articles and magazine ads, respectively. Authors discussing content analysis of internet-based documents coded 29% of their 2,758 observations (messages on websites) (Van Selm and Jankowski 2004, p. 29). Chew and Eysenbach coded 10% of their 1,200 observations (tweets) more than once to obtain statistics for an article on pandemics in the age of Twitter (2010, p. 5). In another study of Twitter, authors had three coders code 0.9% of the 30,675 observations (links in tweets) (Agarwal at al., 2013, p. 33). In general we can say that the sample percentage for ICR statistics is lower for online content, and our decision that 20% of all coding assignments were set aside for multiple coding falls at the top of this range.

Measuring Intercoder Reliability

Once calculates reliability using ICR statistics, which measure the extent to which coders agree with one another. If agreement is high, that means that a number of coders would agree that a piece of content should be coded in a given way. If agreement is low, that means that one coder would code that same content in one way, while another would code it in another. (All calculations in this section are made using the free web-based software ReCal, created by Dr. Deen Freelon.)

There are a number of statistics that a researcher can use to measure agreement among coders in the context of a content analysis. They are:

1. Percent agreement
2. Scott's pi (π)
3. Cohen's kappa (κ)
4. Fleiss' kappa (K)
5. Krippendorf's alpha (α)

Percent Agreement

> *Bottom Line: Use for diagnostics during coding. Report for publication, but it cannot be the only agreement statistc.*

Percent agreement (also called simple agreement) is both the easiest statistic to compute and the easiest to interpret, which is why it remains so popular despite three important criticisms. In fact, you can probably do the calculation in your head. To calculate pairwise agreement, you calculate the agreement between a pair of coders. Given only two coders and one observation, your results can only be 100% (they agree) ore 0% (they disagree). If you are working with multiple coders and multiple cases, then you calculate the average pairwise agreement among all possible coder pairs across observations.

However, the measure becomes more incremental when one uses more coders or more cases. For three coders, two of whom agree, the reliability is 33.$\bar{3}$%. This calculation requires *average pairwise percent agreement*, in which the agreements of all possible pairs are calculated and averaged. For example, for 1 case of digital activism (called observations) three coders (April, Nabil, Maria) code the following values.

> *Figure 2: Average Pairwise Percent Agreement*
>
> *April Nabil Maria*
> *Observation 1: 1 0 0 = 33.$\bar{3}$% percent agreement*
>
> *The agreement pairs are as follows:*
>
> *April Nabil*
> *1 0 = 0% agreement*
> *April Maria*
> *1 0 = 0% agreement*
> *Nabil Maria*
> *0 0 = 100% agreement*
> *Average Pairwise Percent Agreement = 100 + 0 + 0 / 3 = 33.$\bar{3}$%*

The general rule of thumb for percent agreement is presented in Neuendorf: “Coefficients of .90 or greater are nearly always acceptable, .80 or greater is acceptable in most situations, and .70 may be appropriate in some exploratory studies for some indices” (Neuendorf 2002, p. 145). For social science studies in the communication field, the goal is often .80 or 80% pairwise agreement. In a separate article Lombard, Snyder-Duch, and Bracken suggest a higher threshold of .90 (90%) for percent agreement because of the weaknesses described below (2002, p. 596)

This is the statistic I calculate on a weekly basis for the GDADS for diagnostic purposes and I aim for 80% agreement or higher. You can have high percent agreement and low agreement by other statistical measures, but it is rare to have high percent agreement by other statistical measures and low percent agreement. It is a box you need to be able to check, though it does not demonstrate reliability on its own.

Why can you not rely exclusively on percent agreement? There are three drawbacks.

1) A Statistic in Isolation:

There is no comparative reference point to let you know whether your rate of agreement is higher or lower than chance. For example, let’s say that your coding results for a given variable are as follows across 4 observations, again with coders April, Nabil, and Maria.

> *Figure 3: Observed vs. Expected Agreement*
>
> *April Nabil Maria*
> *Observation 1 : 2 2 2*
> *Observation 2 : 2 2 2*
> *Observation 3 : 2 2 2*
> *Observation 4 : 3 2 2*

In this example, Nabil and Maria coded all 4 observations identically and April agreed with them in all but one decision (observed agreement, the only thing that percent agreement measures). Their observed agreement is

83.3̄% (100% + 100% + 100% + 33.3̄% / 4 = 83.3̄%). This is an actual example of coding of the GDADS variable TARGLEV, which measures the geographic scope of the campaign target, where 2 = national (ie, a national government) and 3 = international (ie, a international institution like the UN). Most of the cases target national governments, however, so the expected agreement is also high. According to one measure of agreement which considers both observed and expected agreement, expected agreement is calculated as 84.7% and this statistic (Fleiss's Kappa, discussed later) actually calculates agreement for this example as -0.091, despite the .83̄3 figure for percent agreement! It might be nice to ignore chance and just calculate observed agreement, but calculating agreement without context leaves out a lot of information.

2) Hidden Disagreement:

Because percent agreement is figured as an average across observations or across variables it can hide important disagreements. For example, though there is high average percent agreement in the example below, that 83.3̄% agreement figure hides the fact that there is 0% agreement on the values 3 and 4. "Averages over all categories of a variable… hide unreliable categories behind reliable ones", warns Krippendorff (2004a, p. 426). To remedy this he suggests that in some cases it is appropriate to conduct multiple tests within a single variable. "All distinctions that matter should be tested for their reliability," he writes (2004b, p.429).

> *Figure 4: Hidden Disagreement*
>
> *————————April Nabil*
> *Observation 1 : 2 2 = 100%*
> *Observation 2 : 1 1 = 100%*
> *Observation 3 : 2 2 = 100%*
> *Observation 4 : 3 2 = 0%*
> *Observation 5 : 2 2 = 100%*
> *Observation 6 : 2 2 = 100%*
> *Observation 7 : 2 2 = 100%*
> *Observation 8 : 2 2 = 100%*
> *Observation 9 : 3 4 = 0%*
> *Observation 10 : 2 2 = 100%*
> *Observation 11 : 1 1 = 100%*
> *Observation 12 : 1 1 = 100%*
> *Average Percent Agreement = 83.3̄%*

3. Vulnerability to Gaming

If you average percent agreement across variables you can also artificially raise your agreement rate by putting in variables with low variance and high agreement. For example, in our data set almost no cases include physical violence perpetrated by activists, so agreement is near 100%. We didn't add this variable to game the system, but it does raise our average agreement rate. According the Krippendorff, "[w]hen all coders use only one category, there is no variation and hence no evidence of reliability" (2004b, p. 425).

Gaming can occur regardless of agreement statistic, but it particularly easy with percent agreement because variables with zero disagreement are treated as perfectly reliable, rather than having undefined reliability, which is the case with more complex measures like Krippendorf's alpha and Fleiss' kappa. Notes Krippendorff, "[t]he lesson learned from this admittedly simplistic example is that reliability should always be tested for the distinctions that matter. The inclusion of irrelevant distinctions can overestimate or underestimate the reliability of a variable" (2004a p. 427).

Recommendation

Krippendorff strongly dislikes percent agreement as a measure of intercoder reliability. “The use of percent agreement should be actively discouraged,” he writes. (2004b, p. 425). However, I think Lombard et al. have a better position. They suggest that researchers not use “*only* percent agreement to calculate reliability” (emphasis added) (Lombard et al., p. 601). Percent agreement has meaning. It is easily calculated and interpreted. It is a great diagnostic tool. It simply has too many weaknesses to be the sole indicator or reliability. Though it is still possible to get published in a prestigious journal with only percent agreement reported (Bennett, Foot, and Xenos, 2011), to do so is risky. Which is why there are more options….

Scott’s Pi and Cohen’s Kappa

> *Bottom Line: Only for two coders. Scott’s pi recommended. Cohen’s kappa explicitly not recommended.*

This section groups Scott’s pi (π) or Cohen’s kappa (κ) because both coefficients are for use only with two coders. They both improve upon percent agreement by factoring in the extent to which a given value will be coded by chance. While percent agreement is calculated based on observed agreement, both Scott’s pi and Cohen’s kappa also include a calculation for expected agreement in their equations. The difference in the equations is how this expected agreement is calculated. In fact, the underlying function of each is the same (see below). Pr(a) stands for observed agreement and Pr(e) stands for expected agreement. (Their Wikipedia pages include worked examples: Scott’s pi page; Cohen’s kappa page.)

Because the GDADS has always used more than two coders, we knew that we would not use either of these statistics. Scott’s pi has been generalized to three or more coders by Joseph Fleiss, who created a statistics called (confusingly) Fleiss’ kappa. We use the statistic in calculating agreement for the GDADS, and it is discussed further below.

$$\kappa = \frac{\Pr(a) - \Pr(e)}{1 - \Pr(e)},$$

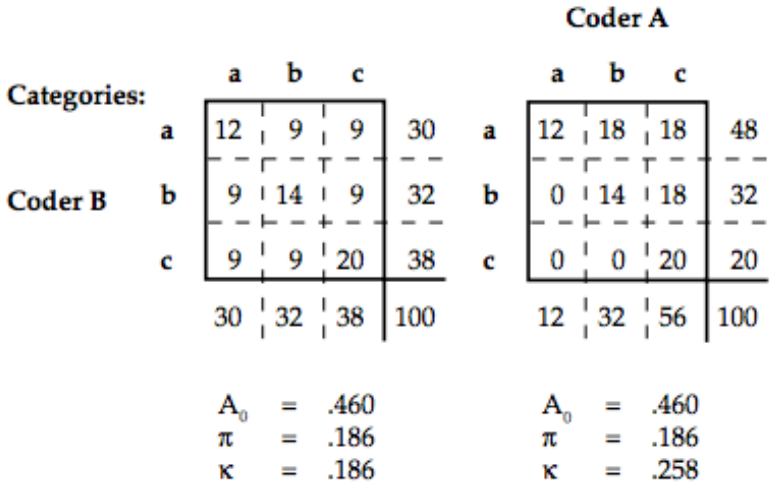
We could also technically calculate Cohen’s kappa for more than two coders in the same way we calculate percent agreement: by calculating pairwise averages. Though Cohen’s kappa can be computing in a pairwise manner, it is still contraindicated for content analysis. To put in bluntly, Krippendorff hates this statistic. Much of his 2004 article, “Reliability in Content Analysis: Some Common Misconceptions and Recommendations” is dedicated to pillaging this statistic, which he calls “just about worthless as a reliability index in content analysis” (2004b, p. 422). He writes:

> *Notwithstanding κ’s popularity… the mathematical structure of Cohen’s κ is simply incommensurate with the logic of the situation that content analysts are facing when the reliability of their data is in question. Kappa (κ) cannot be recommended…. (2004b, p. 419)*

The reason that Krippendorff dislikes Cohen’s kappa is that it could consistent disagreement as expected agreement, as the figure below, from Krippendorff’s 2004 article, shows.

> *Figure 5: The Weakness of Cohen’s Kappa*

The figure above shows two matrixes of coding results by two coders coding one variable with three possible values (categories): “a,” “b,” and “c.” Each matrix shows the results of coding of one variable across multiple observations. The diagonal shows where the coders agreed, where Coder A and B both chose “a” or “b” or “c.” The other cells represent observations where the coders selected different values. For example, in the matrix on the left, the upper righthand corner tells us that 9 times Coder A coded an “a” while Coder B coded a “c.”



Three agreement statistics are coded for each matrix: percent agreement (A), Scott’s pi (π), and Cohen’s kappa (κ). While the statistics are the same (.186) when the distribution of disagreements is evenly distributed (the matrix on the left), when the distribution of disagreements is not evenly distributed (the matrix on the right), Cohen’s kappa is higher than Scott’s pi. This is because Coder B is more likely to code a “c” regardless of case. That coder’s coding of “c” is more predictable. Likewise, Coder A is very unlikely to code a “c”, again, regardless of case.

But predictability is not the same as expected agreement. In fact, it is a form of bias because the coder is more likely to code a given value regardless of the content on the case. Krippendorff explains:

> *“When coders disagree on these frequencies—when they show unequal proclivities for the available categories, as is apparent in the margins of the table [on the left] —κ exceeds π. Kappa (κ) does not ignore the disagreements between the coders’ use of categories; it adds them to the measure as an agreement!” (2004b, p. 420)*

Counting bias as agreement clearly makes Krippendorff livid. “Its behavior clearly invalidates widely held beliefs about κ, which are uncritically reproduced in the literature.” (2004b, p. 421).

Fleiss’ Kappa

> *Bottom line: Recommended for studies with three or more coders. Use for an entire dataset, not for week-to-week diagnostics.*

Fleiss’ kappa is a generalization of Scott’s pi (the one Krippendorff likes) for more than two coders. Like Scott’s pi and Cohen’s kappa it compares observed agreement with expected agreement, a second figure that represents the likelihood of coding a value by chance. While the values in percent agreement run from 0% to 100% (0 to 1), the values of Fleiss’ kappa run from 1 to -1, where a negative value indicates that observed agreement was lower than the expected value. In the equation “P” is observed percent agreement and “Pe” is expected percent agreement. The equation for Fleiss’ kappa is below. A worked example is on its Wikipedia page, including how to calculate expected agreement (observed percent agreement is the same calculation as percent agreement.) Try coding the statistic a few times by hand to get a sense of it. It isn’t hard.

Fleiss' kappa is higher than or and similar to average pairwise percent agreement where there is high agreement and high variability (high diversity of coded values). By contrast, Fleiss' kappa can be much lower than average pairwise percent agreement when there is low agreement on even one value if there is also low variability among values (low diversity of coded values). Where there is low variance in your data (ie, almost all observations are coded the same), Fleiss' kappa is even more sensitive to disagreement, because it interprets low variance as high expected agreement.

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

For example, in Figure 3 percent agreement is 83.3̄%, but Fleiss' kappa is -0.091. Because variability is low, expected agreement is high (84.7%). Fleiss' kappa will be negative whenever observed agreement is lower than expected agreement because the numerator of the equation subtracts the expected agreement from observed agreement. This unfortunate quality of Fleiss' kappa makes it brutal when kappas are averaged. A single negative score can bring down an entire average significantly.

Of course, Fleiss' kappa can also be unfairly generous. For example, there are 249 country codes in our coding system. According to Fleiss' kappa, this high variance means that expected agreement is low (10.6%), and thus agreement is highly rewarded. In a test of 14 cases, coded by 3 coders, Fleiss' kappa was .95. In reality, however, accurately coding the country is which as instance of activism occurs is pretty easy as the country is almost always stated as a manifest value in the coding source, so the "real" expected agreement should be pretty high. But the Fleiss' kappa equation does not have a way of knowing how "easy" it is to code a given variable. It only knows about variance and agreement. Here Fleiss' kappa is misleadingly high because it inaccurately underestimates expected agreement.

Stranger yet, Fleiss' kappa penalizes 100% agreement. The denominator of the equation is 1 – expected agreement, so if expected agreement = 1 then the denominator = 0 and the equation is undefined. This means the Fleiss' kappa of a variable with perfect agreement cannot be included in a Fleiss' kappa average across a number of variables (for example, when calculating average Fleiss' kappa for an entire dataset).

Also, Fleiss' kappa should not be interpreted in the same way as percent agreement. While 80% (.80) is a good target for percent agreement which 90% (.90) is excellent, the Fleiss value can be a little lower. Fleiss gives the following guidance for interpreting his statistic (1981):

> *Figure 6: Interpreting Fleiss' Kappa*
>
> - *< 0.40 = Poor agreement*
> - *0.60 – 0.74 = Intermediate to good agreement*
> - *≥ .75 = Excellent agreement*

Because Fleiss' kappa penalizes variables with low variance, the solution is to calculate Fleiss' Kappa for a large enough sample of observations that the actual variance in the data set appears. Since one cannot know this true variance unless all observations are coded multiple times and subjected to agreement tests, the best strategy is to use Fleiss on as large a group of cases as possible and consider the coefficient a more accurate measure of reliability the larger the group of observations. For this reason it is not recommended for week-to-week coding diagnostics, which are bound to include a small number of observations.

Krippendorff's Alpha

> *Bottom line: Recommended for studies with three or more coders. Use for an entire dataset and week-to-week diagnostics.*

Of the five measures of agreement discussed here, Krippendorff's alpha (α) is the most reliable, but also the conceptually and computationally difficult. Unlike Scott', Cohen', and Fleiss' statistics, which measure observed and expected agreement, Krippendorff's equation measures observed and expected *dis*agreement. Krippendorff's alpha ranges between 1 and 0. He explains, "when observers agree perfectly, observed disagreement Do=0 and α=1, which indicates perfect reliability. When observers agree as if chance had produced the results, Do=De and α=0, which indicates the absence of reliability" (2011, p. 1). The basic form of his equation is below. The term "Do" is observed disagreement and "De" is expected disagreement based on an interpretation of chance.

$$\alpha = 1 - \frac{D_o}{D_e}$$

After that the equation gets more complex. Also unlike the other weighted agreement statistics (Scott, Cohen, Fleiss), the coincidence matrices one uses to calculate Krippendorff require one to count all values rather than all decisions, which can be confusing. The explanation on Wikipedia is not particularly clear and the mathematical symbols used may be confusing. A better explanation, with 4 worked examples, is provided by Krippendorff himself in this paper (2011). Below is an example of the most basic type of calculation for Krippendorff's alpha, for a binary variable with no missing data. The example is from the afformentioned paper by Krippendorff. Ten observations are coded by two coders:

> *Figure 7: Calculating Krippendorff's Alpha for a Binary Variable*
>
> *————————April Nabil*
> *Observation 1 : 1 0 = disagreement*
> *Observation 2 : 1 1*
> *Observation 3 : 1 0 = disagreement*
> *Observation 4 : 0 0*
> *Observation 5 : 0 0*
> *Observation 6 : 1 0 = disagreement*
> *Observation 7 : 0 0*
> *Observation 8 : 0 0*
> *Observation 9 : 0 1 = disagreement*
> *Observation 10 : 0 0*
>
> *Total disagreements (decision pairs) = 4*
> *Total coded values of 1 = 6*
> *Total coded values of 0 = 14*
> *Total coded values = 20*
>
> *General form of Krippendorff's alpha for a binary variable:*
>
> *Worked example of that equation for this example:*

$$_{\text{binary}}\alpha = 1 - \frac{D_o}{D_e} = 1 - (n-1)\frac{o_{01}}{n_0 \cdot n_1}$$

$$_{\text{binary}}\alpha = 1 - (20-1)\frac{4}{14 \cdot 6} = 0.095$$

Krippendorff's alpa has a number of benefits. It can be used for any number of coders (not just two). It can also be used for different kinds of variables (nominal, ordinal, interval, ratio, and more). As implied by the example above, for each type of variable the equation is different. Unlike Fleiss, it can be used for large or small sample sizes and has no minimum. Finally, one of the features Krippendorff is most proud of, it can be used for incomplete or missing data. Krippendorff's alpha uses a system of "bootstrapping" but which missing values are replaced with existing values samples form within the data set. Krippendorff suggests the following for interpreting his coefficient: "[I]t is customary to require α ≥ .800. Where tentative conclusions are still acceptable, α≥ .667 is the lowest conceivable limit (2004a, p. 241).

Acceptable Methods of Improving Agreement After the Fact

It is not so uncommon for one to produce data only to find that it is lower than expected or, despite diligent diagnosis of coding error throughout the coding process to simply be unable to raise a variable to an acceptable level of agreement. For this reason, many scholars seek to improve agreement after the fact (after data have been coded).

Some methods of improving agreement, such as creating indices based on multiple variable, Krippendorff discourages. It is fine toe create an index, but it cannot be used to hid poor agreement. "[W]hen reporting on an index composed of several variables…" he advises, "the reliability of each variable should be measured separately and the smallest reliability among them should be taken as the reliability of the whole system" (2004b, p. 429). He also notes that "[r]esolving disagreements by majority among three or more coders may make researchers feel better about their data, but does not affect the measured reliability" since the coders still disagreed (2004b, p. 430).

However, there are methods that Krippendorff endorses, specifically improving reliability for an entire data by "removing unreliable distinctions from the data," by "recoding or lumping categories," or by "dropping variables that do not meet the required level of reliability" (2004b, p. 430). In other words, no massaging data and no hiding unreliable variables or unreliable values.

Krippendorff's standards are tough, but it is because he is serious about empiricism, serious about producing reliable social science. Because digital activism is a new field, it may be possible to argue for lower standards of agreement. But we as researchers should aim to produce high-quality research, even in this new and challenging topic area, not skirt or pay lip-service to standards of agreement because they are difficult to achieve. Producing data with low agreement means producing data that is unreliable, potentially misleading, and even wrong. To do so is a disservice to the study of digital activism and to social science.

References

Agarwal, S. D., Bennett, W. L., Johnson, C. N., & Walker, S. (2013). *A model of crowd enabled organization: Theory and methods for understanding the role of twitter in the occupy protests*. Unpublished manuscript.

Bennett, W. L., Foot, K., & Xenos, M. (April 01, 2011). Narratives and Network Organization: A Comparison of Fair Trade Systems in Two Nations. *Journal of Communication, 61,* 2.)

Chew, C., & Eysenbach, G. (January 01, 2010). "Pandemics in the age of twitter: content analysis of tweets during the 2009 H1N1 outbreak." *Plos One, 5,* 11.)

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

Hayes, A. F., & Krippendorff, K. (2007). "Answering the call for a standard reliability measure for coding data." *Communication Methods and Measures,* Vol. 1, No. 1, 77-89.

Krippendorff, K. (2011). "Computing Krippendorff's alpha-reliability." Philadelphia: Annenberg School for Communication Departmental Papers.

Krippendorff, K. (2004a). *Content analysis: An introduction to its methodology*. Thousand Oaks, California: Sage.

Krippendorff, K. (2004b) "Reliability in content analysis: Some common misconceptions and recommendations." in *Human Communication Research*. Vol. 30, pp. 411-433.

Landis, J. R. and Koch, G. G. (1977) "The measurement of observer agreement for categorical data" in *Biometrics*. Vol. 33, pp. 159–174

Lombard, M., Snyder-Duch, J., & Bracken, C. C. (2002). "Content analysis in mass communication: Assessment and reporting of intercoder reliability." *Human Communication Research*, 28(4), 587–604.

Neuendorf, K. A. (2002). *The content analysis guidebook*. Thousand Oaks, California: Sage Publications.

Reichert, T., Lambiase, J., Morgan, S., Carstarphen, M., & Zavoina, S. (June 06, 1999). Cheesecake and Beefcake: No Matter How You Slice It, Sexual Explicitness in Advertising Continues to Increase. *Journalism and Mass Communication Quarterly,76,* 1, 7-20.

Van Selm, M., & Jankowski, N. (2004). “Content analysis of internet-based documents.” In M. van Selm & N. Jankowski (Eds.), *Researching new media: An advanced-level textbook*. Thousand Oaks, CA: Sage Publications.

Copyright © 2015. Madidus Theme by VoodooPixel & Different Themes