

# Chapter 5. Achieving Reliability

In this chapter, you will work with a second coder to check intercoder agreement and use the results to revise your coding scheme. After you get a second coding on your data set, you will calculate the agreement between coders, using formulas for both simple and corrected agreement. You will then inspect the disagreements between coders and refine your analytic procedures to reduce them. This process should be repeated until an adequate level of agreement has been reached.

## ■ Introduction to Reliability

Reliability refers to the degree of consistency with which coding segments are assigned to the same categories. To say a coding scheme is reliable is to say that it can be used consistently, over and over again, to produce the same results, from day to day or coder to coder. That is, your coding scheme yields results that are replicable. When you achieve a reliable coding scheme, you assure yourself that the meaning of the coding scheme is clear to you and to those to whom you will be reporting your research.

The key tool in achieving a reliable coding scheme is intercoder agreement. Intercoder agreement is a measure of the extent to which coders assign the same codes to the same set of data. When two coders agree perfectly on their assignment of coders, they have an intercoder agreement of 100% or 1.0. When they totally disagree, they have an intercoder agreement of 0% or 0.0.

## ■ Perspectives on Intercoder Agreement

Perhaps no issue is more contentious in the coding of verbal data than reliability. Neuendorf (2016) argues that quantitative content analysis conducted in the positivist tradition should produce a “scientific” analysis that is reliable. That is, the phenomenon is expected to be stable and the analytic procedures so explicit that any reasonably qualified person would get the same results. In this view, the measuring instrument is the coding scheme, and that instrument is expected to work consistently.

In the qualitative tradition, positions on reliability are more varied. Those who favor a more subjective interpretation see the researcher herself as the measuring instrument, one honed by immersion in the context of the language production, producing interpretations that cannot be replicated by those outside that context. From this perspective, measuring reliability makes little sense as there is neither the need nor the possibility of achieving intercoder agreement.

Many qualitative researchers take a less radical approach to subjectivity and see interpretation as embedded in social life and thus able to be shared with others. Coding, then, need not be totally individualistic, but may be taken on by a team who work together to establish agreement among coders. Taking a grounded theory perspective, for example, Charmaz (2014) describes how team coding can contribute to a developing analysis:

In team research, several individuals may code data separately and then compare and combine their codes to evaluate their fit and usefulness. Might one team member come up with different codes than other members? Yes, our perspectives, social locations, and personal and professional experiences affect how we code. Thus, team researchers can scrutinize how differences among team members may generate new insights, rather than dismiss a colleague’s codes that differ.

Saldaña (2016) too refers to the role that checking intercoder agreement can play in a team coding environment, describing it as a kind of “crowd-sourcing reality check” (p. 36). For many qualitative researchers, then, intercoder agreement is valued as a tool for developing an analysis rather than a means for validating it.

## The Role of Context in Limiting Intercoder Agreement

From our perspective, the extent to which two people looking at the same verbal phenomena will make the same interpretation depends upon the extent to which the context of interpretation overlaps with the context of production.

The conditions for overlap are not fixed, however. As the diagram in Figure 5.1 suggests, some phenomena are more transparent than others. With relatively transparent phenomenon, the limits of interpretation extend far from the context of production, nearly reaching to the boundaries of the context in which your analysis is to be interpreted. With opaque phenomenon, on the other hand, the limits of interpretation are wrapped tightly around the context of production and few if any outside of that context can expect to be able to interpret what's going on.

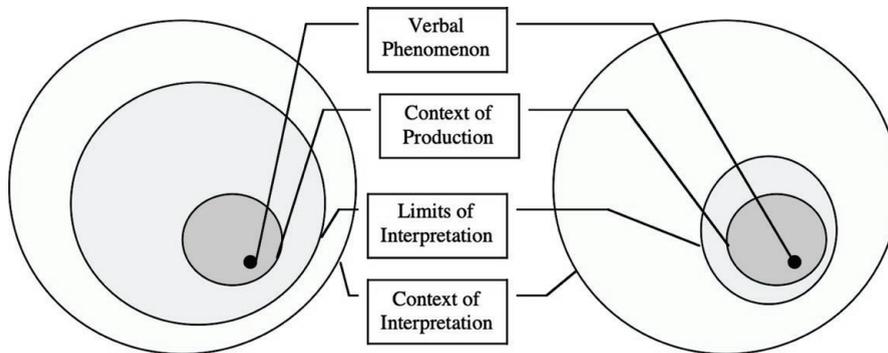


Figure 5.1: Transparent and opaque phenomenon.

For example, if we are coding for the phenomenon of author mentions—how often and when an interviewee mentions the names of the authors she is reading (Geisler, 1994)—we might expect the phenomenon to be relatively transparent. That is, in the context in which such an analysis might be expected to be interpreted, what is and is not an author mention would not seem to be problematic:

**P:** at what point did you stop on ... Friday I guess it was ... yeah  
Friday

V: can't even remember ... [] seemed like years ago ... I should write down like ...

P: well it's not important that I get precisely ... what you were doing ...

V: I don't remember what I was doing ...

P: but just tell me about what you were doing ...

V: I think I was reading this ... um ... I was reading over Gerald Dworkin... [flipping pages]

In this example, even if we are not familiar with the context, most of us can recognize that the phrase “Gerald Dworkin” refers to the author of a text the interviewee was reading.

With other verbal phenomena, interpretation is far more difficult for those outside of the context of production. If, for example, we were coding indexicals for what they refer to, most of us would be hard pressed to interpret the *this*'s, *here*'s, and *now*'s in the following conversational turn:

J: But I think th-at it would be good for us to really imagine what this could be because there are a number of issues th-at come up down here: to the new goal: where am I? Oh, I'm here. I just went off the screen. I think I had: this is deriving from the last: you know, was the revised goal of the DCR the same as the first one. And this is the one here: network support. See, I'm not sure. What about the dominance? What about the dominance in the computer right now.

In this example, interpretation is complicated by the usual opacity of topical information (“it would be good for us to really imagine what this could be”) and the temporal positioning of the conversation (“What about the dominance in the computer right now?”). Interpretation is further complicated by the fact that J is looking at a computer screen (“where am I? Oh, I'm here. I just went off the screen.”) and looking at a file containing text from which he may or may not be reading (“this is deriving from the last: you know, was the revised goal of the DCR the same as the first one. And this is the one here: network support.”).

When phenomena are relatively transparent, when the context of

production overlaps with the context of interpretation, we can expect achieving reliability to be relatively straightforward. That is, from one time to the next, from one coder to the next, judgments will remain relatively constant with respect to the phenomenon of interest (is this an author mention?). When phenomena are more opaque, when the context of production overlaps little with the context of interpretation, we can expect reliability to be harder, if not impossible, to achieve. In some cases, only the participant herself may be in a position to make a judgment (is this speaker deliberately lying?). Verbal phenomena, then, may range along a continuum of interpretation, with some phenomena being relatively opaque and some being relatively transparent, and a great many lying somewhere in between.

## ■ Uses of Intercoder agreement

In this chapter, we will describe methods aimed at achieving as reliable an analysis of verbal data as possible. Our position here is that reliability is important not because we expect verbal phenomenon to be wholly interpretable outside of its context of production (we do not) but because reliability is our key tool for insuring that we have been clear in the definition of our analytic constructs and that we have been explicit in our analytic procedures. As you will see, working with a second coder is an excellent way to understand the extent to which specific phenomenon are context bound and one of best ways to develop methods for communicating an interpretation of that phenomenon outside of the context of production.

Fundamentally, we believe that analysis is a rhetorical act of persuasion: We must persuade our intended readers that the pattern of phenomenon is as we claim. If a phenomenon is wholly opaque outside of its context of production, this rhetorical effort is hopeless. We can never expect to communicate an interpretation of what is going on to those who were not there. Happily, intended readers are usually more resilient in their powers of interpretation than that. Working to achieve reliability will help you to develop the means to help you and your readers to understand what you mean.

## ■ Measures of Intercoder Agreement

### ■ Simple Agreement

Intercoder agreement is a measure of the extent to which coders assign the same codes to the same set of data. In the first half of this chapter, we review the various measure associated with measuring intercoder agreement, some of which are complex. In the second half of the chapter, we introduce the less complex procedures you can use for your actual calculations.

The simplest measure of intercoder agreement is simple agreement, which is defined as the percentage of decisions that are agreements. If two coders agree entirely on how to code a data set, they will have an intercoder agreement of 100%. Simple agreement is calculated as

$$\text{\# of agreements} / \text{\# of coding decisions}$$

An example showing the calculation of simple agreement is shown in Figure 5.2. Column A contains the first coding; Column B the second coding; and Column C the agreement where 1 is used for agreement and 0 for disagreement. At the bottom of the column, we find the total number of agreements (14), the total number of coding decisions (16) and simple agreement (14/16 or .88).

Coder 1	Coder 2	
business	business	1
user	user	1
business	business	1
system	team	0
system	system	1
system	team	0
team	team	1
system	system	1
business	business	1
user	user	1
system	system	1
user	user	1
system	system	1
		14
		16
		0.88

Figure 5.2: Reliability data comparing the coding done by two coders.

Simple agreement is the most intuitive way to communicate the rate of intercoder agreement on a set of data. Your coding team will readily understand what it means when you say, “we agreed 88% of the time,” and this easy-to-calculate measure is also useful for tracking improvements in intercoder reliability as a coding scheme is developed: “We agreed 88% of the time today, compared to just 73% of the time last week.”

## ■ Corrected Agreement

Although simple agreement is an intuitive measure of reliability, using it alone doesn’t take into account the fact that two coders could have agreed simply by chance.

To understand the impact of chance on levels of agreement, imagine that I give you a coding scheme that has only two categories. The chances that we would pick the same category in any given coding decision are rather high, one out of two. If, on the other hand, we are using a coding scheme that has 10 categories, the chance of accidental agreement is a lot lower, 1 out of 10. Thus achieving an agreement level of 90% with a two-category scheme is a lot easier than achieving that same level of agreement with a 10-category scheme. Methods of calculating corrected agreement are a way of taking that fact into consideration.

The traditional measure for correcting for agreement by chance is known as Cohen’s kappa ( $\kappa$ ), named for Jacob Cohen who proposed it in 1960. Cohen’s kappa works by subtracting from the percentage for simple reliability a correction for chance agreement. So, for example, we would expect with Cohen’s kappa that the .88 simple reliability calculated for the data in Figure 5.2 to be corrected downward.

More recently, some researchers are calling for the use of a different statistic, Krippendorff’s alpha ( $\alpha$ ), championed by Klaus Krippendorff for use in content analysis (1970, 2016). Krippendorff’s alpha corrects for the raters’ bias as we’ll discuss later. Krippendorff’s alpha works by dividing the observed disagreement among coders by the disagreements one would expect if the coding was simply by chance.

Both of these measures of intercoder agreement can be calculated using a variety of on-line calculators, which we describe later in this chapter. But

because we believe you should understand the underlying choices behind the statistics you choose to use and report, we spend some time in the rest of this section describing how Cohen's kappa and Krippendorff's alpha work and what the differences are between them. If you simply want to calculate your reliabilities, you can skip to the next major section.

### ■ Understanding Cohen's Kappa ( $\kappa$ )

Correcting agreement using Cohen's kappa begins with a table of agreements & disagreements like that shown in Figure 5.3. Down the side, we list the categories assigned by the first coder in lowercase. Across the top, we list the categories assigned by the second coder in uppercase. In the table itself, we list the number of times each combination occurred. For example, the table in Figure 5.3 shows that the number of times that the first coder assigned *Business* while the second coder assigned *user* was 0. The last column shows the row totals; the last row shows the column totals. Together, these two rows are often called the marginals. The lower right-hand corner contains the grand total, shown in blue (16). Values on the diagonal, shown in yellow, represent the number of times the two coders agreed.

	Business	User	System	Team	
business	3	0	0	0	3
user	0	6	0	0	6
system	0	0	4	0	4
team	0	0	2	1	3
	3	6	6	1	16

Figure 5.3: Table of agreements & disagreements for Cohen's kappa.

If the two coders had been in perfect agreement, all of the values would be on the diagonal, and the rest of the values in the table would be 0. Here, agreement was not perfect because of those 2 coding decisions where the first coder recorded *System* while the second coder recorded *team*.

Using this table, simple reliability can be calculated as the sum of the diagonals:

$$3+6+4+1=14$$

divided by the grand total of 16, or the value of .88 we calculated earlier.

Corrected agreement using Cohen's kappa is calculated using the expected level of agreement for each coding category if the decisions were made simply by chance. The expected level of agreement on a category involves calculating what is known as the joint probability of that category.

To calculate joint probability of agreement for a specific cell, we take the probability that the first coder chose a particular value—what's called its simple probability—and multiple it by the simple probability the second coder chose that same value. In our example, what is the joint probability of an agreement on *business* with *Business* just by chance? The first coder chose *business* 3 times out of 16 decisions so its simple probability is:

$$P(\text{business}) = 3 \text{ in } 16 \text{ or } .19$$

The second coder chose *Business* 3 times out of 16 decisions as well, so its simple probability is also

$$P(\text{Business}) = 3 \text{ in } 16 \text{ or } .19$$

The joint probability of *business* with *Business* is the two simple probabilities multiplied together:

$$P(\text{business with Business}) = P(\text{business}) * P(\text{Business})$$

or

$$P(\text{business with Business}) = .19 * .19 = .035$$

To use the joint probability to calculate the expected frequency of a category, you multiply it by the total number of decisions made:

$$\text{business with Business expected} = P(\text{business with Business}) * \text{Grand Total}$$

or

$$\text{business with Business expected} = .035 * 16 = .56$$

The expected frequency for the other agreement combinations (*user* with *User*, *system* with *System*, *team* with *Team*) are calculated in the same way and then all of them are added up to give a total value for the expected level of agreement by chance, known as q.

P(business with Business)	0.56
P(user with User)	2.25
P(system with System)	1.5
P(team with Team)	.19
q (total)	4.5

Using q, we can then calculate Cohen's kappa as

$$\text{Kappa} = (d - q) / (N - q)$$

where

d = # of actual agreements

q = sum of agreement by chance

N = number of decisions

For the data in Figure 5.3, then,

$$\text{kappa} = (14 - 4.5) / (16 - 4.5)$$

or

$$\text{kappa} = 9.5 / 11.5 = .83$$

If we were to report the reliability for this coding scheme then, we could report, "Agreement between coders was .88 or .83 corrected using Cohen's kappa."

### ■ Understanding Krippendorff's Alpha ( $\alpha$ )

Correcting agreement using Krippendorff's alpha begins with the recording of coincidences as shown in Figure 5.4. In this table, each pair of decisions from the reliability data in Figure 5.2 yields two coincidences, once for Coder 1 with Coder 2 and once for Coder 2 with Coder 1. Thus, the number of coincidences for two coders will always be twice the number of decisions. For example, the three agreements on *Business* shown in Figure 5.4 yield the six coincidences shown in light orange.

	<b>Coding</b>		<b>Coincidences</b>	
	<b>Coder 1</b>	<b>Coder 2</b>	<b>Coder 1 with Coder 2</b>	<b>Coder 2 with Coder 1</b>
1	business	Business	business w/ Business	Business w/ business
2	user	User	user w/ User	User w/ user
3	business	Business	business w/ Business	Business w/ business
4	system	Team	system w/ Team	Team w/ system
5	system	System	system w/ System	System w/ system
6	system	Team	system w/ Team	Team w/ system
7	team	Team	team w/ Team	Team w/ team
8	system	System	system w/ System	System w/ system
9	business	Business	business w/ Business	Business w/ business
10	user	User	user w/ User	User w/ user
11	system	System	system w/ System	System w/ system
12	user	User	user w/ User	User w/ user
13	user	User	user w/ User	User w/ user
14	user	User	user w/ User	User w/ user
15	user	User	user w/ User	User w/ user
16	system	System	system w/ System	System w/ system

*Figure 5.4: Coincidence data for Krippendorff's alpha for intercoder agreement data given in Figure 5.2.*

All of the coincidences are counted and entered into a coincidence matrix like that shown in Figure 5.5. Here we see that the six coincidences concerning the Business code show up in the Business/business cell, again shown in light orange. The rest of the coincidental agreements are shown in yellow and the total number of coincidences in blue. Notice how the Coincidence Table used for Krippendorff's has exactly twice the number of coincidences as there were decisions in the table of agreements and disagreements used with Cohen's kappa. This is because each decision, shown in the first two columns in Figure 5.4, yields two coincidences, as shown in the last two columns.

	Business	User	System	Team	
business	6				6
user		12			12
system			8	2	10
team			2	2	4
	6	12	10	4	32

Figure 5.5: Coincidence matrix for Krippendorff's alpha.

The formula for hand calculating Krippendorff's alpha from the coincidence table is complex (Krippendorff, 2013b), but we walk through it for those interested:

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

We begin with the numerator (top) of this formula. On its left-hand side, it multiplies together:

- the number of expected coincidences that are free to vary ( $n-1$  or  $32-1$ ) and
- the sum of the total number of actual coincidences ( $\sum_c o_{cc}$  or  $6+12+8+2$ ).

For our reliability data this equals ( $31*(6+12+8+2)$ ) or 868.

Next, on its right-hand side, the formula subtracts the sum of the expected coincidences for Coder 1 using the formula  $\sum_c n_c(n_c-1)$ , which is calculated like this:

Coder 1 Marginals	Formula	Calculated Value
6	$6*(6-1)$	30
12	$12*(12-1)$	132
10	$10*(10-1)$	90
4	$4*(4-1)$	12
	sum	264

Subtracting this sum (264) from our first number (868) yields a value of 604 ( $868-264$ ) for the numerator in the formula for Krippendorff's alpha.

On its left-hand side, the denominator (bottom) of this formula multiplies together:

- the total number of coincidences (n or 32) and
- the number of expected coincidences that are free to vary<sup>6</sup> (n-1 or 32-1).

For our reliability data this yields 992.

On its right-hand side, the formula then subtracts a measure of the expected coincidences for Coder 2 ( $\sum_c n_c(n_c-1)$ ) which is exactly the same as for Coder 1 above. Subtracting this sum (264) from our first number (992) yields a value of 728 (992-264) for the denominator in the formula for Krippendorff's alpha.

The final value for Krippendorff's alpha then is the numerator calculated earlier (604) divided by this denominator (728) which yields .83.

## ■ Choosing a Measure of Corrected Agreement

As you may have noticed from the values just calculated, the values for Cohen's kappa and Krippendorff's alpha are often not far apart. For the data in Figure 5.2, Cohen's kappa yields a value of .83, the same as Krippendorff's alpha. As we shall see, when bias enters into your intercoder agreement data—bias toward favoring one code over another, making it so that there is a higher probability of agreement on that code—the two measures can become quite different. In this situation, Krippendorff's alpha will give you a more accurate estimate of the reliability of your coding scheme.

As Krippendorff's 2004 analysis shows, Cohen's kappa becomes an inaccurate measure of intercoder agreement when there is bias in the distribution of disagreements. In the agreements and disagreements for our sample data as shown in Figure 5.3, there is very little bias in the disagreements be-

---

6 The term “free to vary” refers to the fact that if the sum of a given set of values is known (264 in our example), and the total number of expected coincidences is known (32 in our example), then the first 31 of these expected coincidences can take on all possible values (in other words “are free to vary”), but the last value, the 32<sup>nd</sup> value, is not free to vary because it along with the other 31 coincidences, has to add up to the known sum (268). So it is not free to vary.

cause there were very few disagreements: The first coder disagreed with the second coder just two times. Another way to tell that there is little bias in our data is that the marginals for the two coders are very close: 3, 6, 6, 1 for the first coder and 3, 6, 4, 3 for the second coder. To be fair, our sample data does suggest a slight bias towards using team on the second coder's part, but the fact that Cohen's kappa and Krippendorff's alpha are almost equal suggests that the bias is very small.

Such is not always the case. In particular, the data Krippendorff (2004) used, shown in Figure 5.6, well illustrates the impact of bias on calculations of Cohen's kappa. In both tables, there are 46 agreements and 54 disagreements. In both tables, the agreements are distributed the same way:  $Aa = 12$ ;  $Bb = 14$ , and  $Cc = 20$ . The disagreements, however, are not distributed the same way. In the table on the top, the 54 disagreements are distributed absolutely without bias: 9 in each cell off the diagonal of agreements. In the contingency table on the bottom, however, the 54 disagreements show substantial bias. All of the disagreements are now distributed in the upper right-hand corner, shown in pink. And the marginals confirm the bias: 48, 32, and 20 are quite different from 12, 32, and 56.

	A	B	C	
a	12	9	9	30
b	9	14	9	32
c	9	9	20	38
	30	32	38	100

	A	B	C	
a	12	18	18	48
b	0	14	18	32
c	0	0	20	20
	12	32	56	100

Figure 5.6: Contingency tables where disagreements are without bias (at the top) and with maximum bias (at the bottom). Data taken from Krippendorff (2004).

No one would argue that the amount of intercoder agreement in the table on the bottom is greater than that for the table on the top. But, because of the way it is calculated, Cohen's kappa for the biased data on the top is actually higher than for the unbiased data on the bottom: .26 versus .19. This is a situation where using Cohen's kappa can be misleading.

To illustrate the magnitude of possible distortion, look at Figure 5.7, which illustrates the way that Cohen's kappa increases as the bias among disagreements increases. When the bias is 0, that is when the disagreements are all equal to 9 as shown in the top table of Figure 5.6, the kappa is .19 as we just mentioned. If 1/9 of the values below the diagonal migrate above the diagonal—i.e., all values below the diagonal become 8 and all values above the diagonal become 10—the kappa becomes a little higher. When all of the values below the diagonal migrate, leaving all 0s below the diagonal and 18s above the diagonal, we have a situation of strong bias and the kappa is .26. With Krippendorff's alpha, by contrast, strong bias does not have distorting effects. It yields an intercoder agreement measure of .19 for both the biased and unbiased data, and thus is a more accurate estimate of reliability.

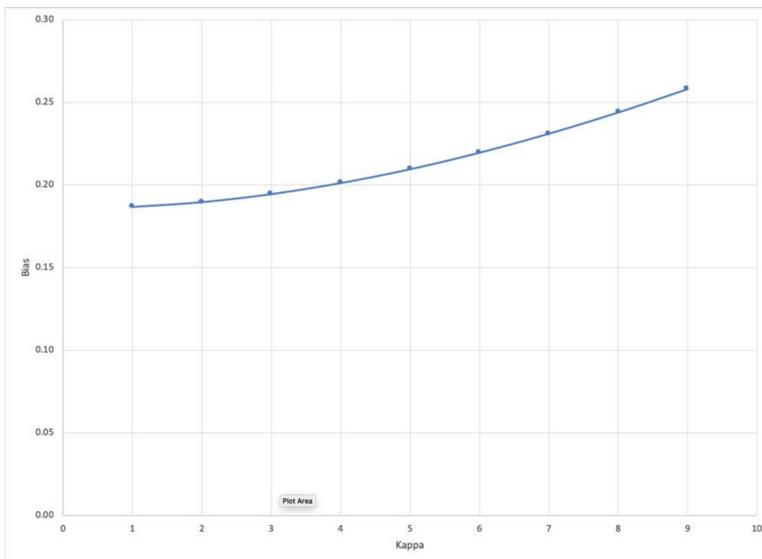


Figure 5.7: The increase of Cohen's kappa with increase in bias.

Bias is more common in coded data than you might expect. If you have a Miscellaneous or None category, one of your coders may default to it in cases where she is not sure. Or when two codes are hard to differentiate, one coder may prefer the first code while the second coder prefers the second code. Cohen's kappa is the more common measure of intercoder agreement, but the important take away for researchers seeking a measure of corrected intercoder agreement is not to rely on Cohen's kappa alone. Check the marginals for your data to see if one of your coders shows bias toward some codes over others. If so, calculate both Cohen's kappa and Krippendorff's alpha and report Krippendorff's alpha as the best measure of reliability when there is a discrepancy between the two.

---

## Exercise 5.1 Test Your Understanding

You can download this exercise at <https://wac.colostate.edu/books/practice/codingstreams/>.

The data shown in Figure 5.8 below and in the “unbiased” sheet of the Excel worksheet at the link above shows a low level of agreement, but no bias. That is, the marginals in the table of agreements and disagreements are identical. Simple agreement equals just .50, and if we used one of the online calculators described later in this chapter we would find that Cohen's kappa is .34 and Krippendorff's alpha is also .34.

	Coding		Agreement	Coincidences	
	Coder 1	Coder 2		Coder 1 with Coder 2	Coder 2 with Coder 1
1	business	Business	1	business w/ Business	Business w/ business
2	team	User	0	team w/User	User w/ team
3	business	Business	1	business w/ Business	Business w/ business
4	system	Team	0	system w/ Team	Team w/ system
5	system	System	1	system w/ System	System w/ system
6	system	Team	0	system w/ Team	Team w/ system
7	team	Team	1	team w/ Team	Team w/ team
8	system	System	1	system w/ System	System w/ system

9	business	Business	1	business w/ Business	Business w/ business
10	user	System	0	user w/ System	System w/ user
11	system	System	1	system w/ System	System w/ system
12	user	System	0	user w/ System	System w/ user
13	team	User	0	team w/User	User w/ team
14	user	Team	0	user w/ Team	Team w/ user
15	team	User	0	team w/User	User w/ team
16	system	System	1	system w/ System	System w/ system

Figure 5.8: Worksheet for Exercise 5.1.

In the worksheet labeled “exercise,” change the data to increase the bias of Coder 2 toward the Business code. Keep the level of agreement the same. [Hint: The easiest way to do this is to change Coder 2’s codes for every line where there is 0 agreement; these cells are marked in orange in the worksheet.] The table of agreements below the data in the worksheet will automatically be updated.

What do you predict will happen if we recalculate the reliability measures? Will simple reliability go up or down or remain unchanged? Will kappa go up or down or remain unchanged?

**For Discussion:** Looking at the marginals of your table of agreements and disagreements. Under what conditions might you expect this kind of bias to arise?

## ■ Standards for Intercoder Agreement

All measures of reliability vary between 0 and 1.0, and, as you work with a second coder to develop a reliable coding scheme, you should see your measure of intercoder agreement move closer to 1.0. But you may well be wondering how far you need to go to reach an acceptable level of agreement. As Neuen-dorf (2016) has pointed out, there are no uniform standards for intercoder agreement. Rules of thumb have been proposed for good agreement using Cohen’s kappa of .75 (Banerjee et al. 1999), .80 (Popping, 1988) and .81 (Landis & Kocj, 1977). Krippendorff (2013) has set .80 as the standard for reliability for Krippendorff’s alpha. We ourselves have used .80 as our goal for an accept-

able level of corrected agreement, though with particularly difficult coding schemes a .75 may be acceptable as the best that can be achieved.

## ■ Selecting Data for Second Coding

We agree with Neuendorf's (2016) endorsement of Lombard et al.'s (2002) three-part recommendation for achieving reliability:

- use at least two coders,
- calculate a measure of intercoder agreement for each coding scheme you use, and
- report the size of the sample you used to establish that agreement as well as your rationale for selecting it.

Keep in mind that the process of establishing intercoder agreement may require several cycles of second coding. As you will see, once you have coded a set of data and calculated your agreement with a second coder, you will inevitably refine your coding scheme and then try it out again. Each cycle will require a separate subset of the data; if you were to use the same subset over and over again with the same coders, they would gradually memorize the “correct” coding rather than follow the revised coding scheme. In order to avoid this effect, you will need to use a fresh sample of data for each cycle of second coding.

Generally speaking, researchers use at least 10% of the data for the final second coding to establish intercoder agreement; for smaller sets, the sample may get closer to 25%. In cycles of second coding leading up to this final cycle, you may use less than 10%, but make sure it is a well-chosen sample (more on this below). In any case, it is not unusual to go through two to four cycles of second coding in developing a coding scheme, so you need to start with a sample that is large enough to support the repeated subsampling for code development as well as the 10% you need for the final reliability check.

You need to be careful about how you choose a sample for second coding. Your goal is to develop a coding scheme that is sensitive to the range of variation in your data. To do that, you need to get this range of variation into the sample of data you use for second coding. If the design of your analysis in-

cludes major differences across data sets, you should include some data across each of these differences. If, for example, you are studying writing across the curriculum in the sciences and in the humanities, you would want to develop and test your coding scheme with some data from the sciences and some from the humanities. If you are looking for gender differences in contributions to online forums, you might want to select data from the range of forums you have looked at as well as selections of those forums in which women were active, men were active, and perhaps when both are active.

The selection of coders is equally important. Even if you plan to code the full data set yourself, you will need to work with at least two other people in second coding. For the purposes of developing your coding scheme, you will first want to choose someone who is willing to work with you over several cycles of second coding, perhaps extending over several weeks. A member of your own research team is ideal since the process of discussing coding decisions can enrich your team's analysis, a benefit of particular interest to those taking a qualitative perspective. If you are working on your own, you may want to partner for second coding with another researcher who also needs a second coder. This kind of reciprocal arrangement can enrich both projects.

Once you have reached an acceptable level of intercoder agreement with your first second coder, you will want to do a final coding with a different coder using the final data sample of 10% and the final coding scheme. This final second coder should not be someone from your team who has participated in the coding scheme development, but rather someone who comes to the final coding with fresh eyes. The measures of intercoder agreement that you calculate for this final second coding will be the ones that you report for your study.

### ■ Memo 5.1: Sampling for Second Coding

Compile a sample of your data to use to check the reliability of your coding. Make sure to pull together data that shows the full range of variation in your design. Include enough data to support several rounds of preliminary coding as well as 10% of the data for final coding.

Document the rationale behind your choices.

---

## ■ Managing the Second Coding

Coding is intense work. In early coding sessions, plan to give a second coder no more data than can be coded in a sitting of one to one and one-half hours. For the final coding, get as close as possible to 10% of the data. Coders are usually trying to do their best, so it makes sense to give them the most comfortable and least distracting circumstances possible.

## ■ Preparing the Data for Second Coding

To prepare data for a second coding, organize the data in such a way to give the second coder no hints about how the segments have been coded previously (see Excel Procedure 5.1 and MAXQDA Procedure 5.1)

## ■ Handling the Session

Begin the coding session with about 15 minutes familiarizing the coder with the coding scheme, the data, and the coding task. Concerning the data, make sure they understand how the data have been segmented and where they are to record their coding decisions. Concerning the coding scheme, make sure they know what the categories mean and how to apply them. One of the best ways to do this is to prepare a very small data set, formatted in the same way as the data to be coded, and ask them to try to apply the scheme. As they do so, you can then work through any questions they have about procedure.

Avoid using this training period to give the coder information about coding that is not included in the coding scheme itself. Of course, you never let them know what kinds of patterns you expect to see in the data (i.e., that you expect turns by one speaker to have more X than by another speaker). But even further, be sure not to communicate to them any additional information about how to decide how to apply the codes. After all, you are trying to find out how well your coding scheme can communicate—both to coders and to your eventual readers—the nature and variation in the data.



## Excel Procedure 5.1: Preparing Excel Data for Second Coding

<https://goo.gl/GDW4CW>

Prepare your data for second coding in Excel as follows:

1. Hide the results of your first coding by selecting the column and using the **Column/Hide** command under the **Format** menu as shown in Figure 5.9.

The coder ought to be able to see the column headers on the data sheet at all times. This can be accomplished by splitting the window as follows:

2. Use the **Split** command under the **Window** menu to split the window in two. Then drag the tiny bar at the top of the scroll bar to arrange the top pane so that the column header is the only thing visible as shown in Figure 5.10.
3. Choose the **Freeze Panes** command under the same **Window** menu.

Your second coder will also need to be able to consult the full coding scheme at all times. To insure this:

4. Print out the coding scheme.
5. Make sure it is formatted for ease of use, with names of codes in bold, cases indented and bulleted (See Figure 4.1 in Chapter 4 for an example).

Ideally a coding scheme is 1 page long; it should never exceed 2 pages.

Segment	Speaker	Clause	First Coder
13	Facilitator 1	Exactly..	
14	Jess	Math is merging us together.	other
15	Kara	There you go.	other
16	Facilitator 3	That's right. The bridge discipline.	
17	Facilitator 2	She's the bridge discipline.	
18	Facilitator 1	Actually what we better do is mechanical math materials chemistry.	
19	Kara	I do Materials...	other

Figure 5.9: Hiding columns to prepare for a second coder in Excel.

Segment	Speaker	Clause	cond Coding
13	Facilitator 1	Exactly..	
14	Jess	Math is merging us together.	
15	Kara	There you go.	
16	Facilitator 3	That's right. The bridge discipline.	
17	Facilitator 2	She's the bridge discipline.	
18	Facilitator 1	Actually what we better do is mechanical math materials chemistry.	
19	Kara	I do Materials...	
20	Kara	I do mechanical.	
21	Facilitator 1	She does a little of everything	
22	kara	Some Biology.	
23	Facilitator 1	OK, we're gonna start with the first question which is dealing with some of the norms, the agreed upon norms here at RPI. What do successful faculty in your department generally do to recruit students to work with? So I will leave it at the table.	

Figure 5.10: Splitting the window using Split command in Excel.



## MAXQDA Procedure 5.1: Preparing MAXQDA Data for Second Coding

<https://goo.gl/GDW4CW>

Prepare your data for second coding in MAXQDA as follows:

1. Duplicate the project with the first coding using the **Duplicate Project** command under **Project** in the menu bar.
2. Export your coding system from the duplicate project using the **Export Code System** command under **Codes** in the menu bar. Use the MAXQDA format.
3. Delete all the codes in the duplicate project by right clicking on each code in the **Code System** window and then selecting **Delete**. This will remove your first coding.
4. Import the code system back into the duplicate project using the **Import Code System** command under **Codes** in the menu bar.

The duplicate project will now be ready for second coding.

5. In the **Code System** window, in turn, click on each memo containing the code definition attached to a code.

As shown in Figure 5.11, the definitions will then open in a tabbed memo window with one tab for each code definition. Your second coder can click through them to consult the full coding scheme. Give some thought to the order in which they are placed, with preferred codes coming before less preferred codes.

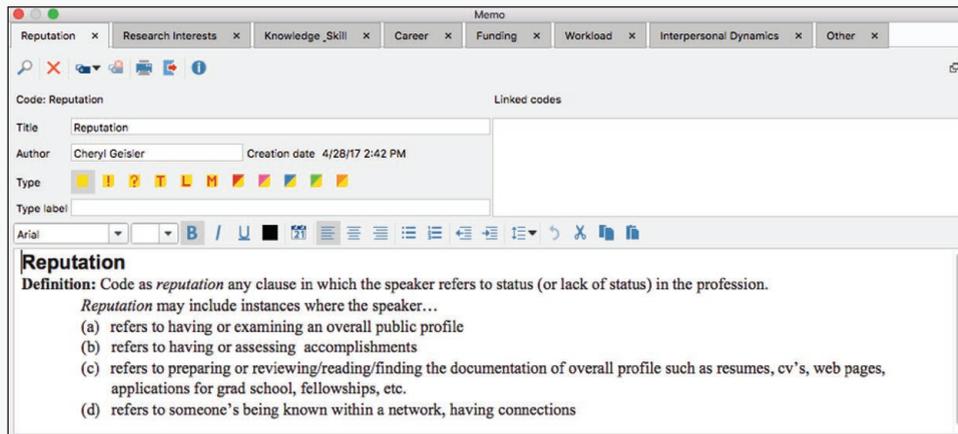


Figure 5.11: Making the full coding scheme available in MAXQDA used a tabbed code window.

## ■ Memo 5.2: Second Coding

Identify an appropriate second coder for your data. Prepare the data and coding scheme and schedule a time for a training session. Make sure to give your second coder only as much data as can be managed in a reasonable session of one to one and a half hours.

Document your round of coding. Make sure to link the coding results to the correct version of the coding scheme that you used.

---

## ■ Calculating Item-by-Item Agreement

Once a second coder has completed coding the sample data set, you need to put the two codings side by side and then calculate the level of agreement between the two codings (see Excel Procedures 5.2 and 5.3 and MAXQDA Procedures 5.2 and 5.3). As discussed earlier, intercoder agreement should be calculated both as simple agreement, which is a straightforward measure of agreement, and as corrected agreement using either Cohen's kappa or Krippendorff's alpha or both.



### Excel Procedure 5.2: Putting the Two Codings Side by Side

<https://goo.gl/GDW4CW>

When you get the worksheet with the second coding in Excel back from your second coder:

1. Select the columns on either side of the hidden column.
2. Choose the **Column Unhide** command under the **Format** menu.

This will place the two codings side by side in Excel.



## Excel Procedure 5.3: Checking Item-by-Item Agreement

<https://goo.gl/GDW4CW>

Once your data is side-by-side in Excel:

- Next to the first line of codes and in the third column, type =IF
- Click on the first code in the first column and then type =
- Click on the first code in the second column and then type ,1,0)
- Hit enter.
- Select the cell containing the new formula and drag it down next to each pair of codes.

In this formula, we tell Excel to check whether the code assigned by the first coder is the same as the code assigned by the second coder. If this is true, we tell Excel to record a 1 for 1 agreement. If there is no match, we tell Excel to record a 0 for no agreement. If you have no coding for some segments (as we do for the segments that have darkened cells in Figure 5.12), make sure to delete the formula from the Agreement column. Otherwise, those rows will count as agreements and artificially raise your agreement numbers.

	C	D	E	F	G
	Speaker	Clause	First Coder	Second Coder	Agreement
1	Facilitator 1	So we'll begin. Do you have the questions handy?			
2	Facilitator 2	Maybe we should just go around the room and make sure everyone knows every one else. I'm Debbie Kaminski from Mechanical			
3	Annie	Antoinette Maniatty from Mechanical Engineering	other	Other	1
4	Lakshmi	Linda Schadler from Material Science and Engineering	other	Other	1
5	Kara	Kristin Bennett from Math	other	Other	1
6	Jess	I'm Julie Stenckman in Chemistry	other	Other	1
7	Savannah	Susan Sanderson from Management	other	Other	1
8	Polly	I'm Pat Search from Language, Literature and Communication	other	Other	1
9	Facilitator 3	Cheryl Geisler Language Literature and Communication. You can see that we're kind of a really wide ranging group.			
10	Unknown	If you count all the engineers among us... [laughter]	other	Other	1
11	Unknown	that's because we don't have men in the room. ..	other	Other	1
12	unknown	they would do it some other way	other	Other	1
13	Facilitator 1	Exactly..			
14	Jess	Math is merging us together.	other	Other	1
15	Kara	There you go.	other	Other	1
16					

Figure 5.12: Checking item-by-item agreement in Excel.



## MAXQDA Procedure 5.2: Putting the Two Codings Side by Side

<https://goo.gl/GDW4CW>

1. Merge the two codings into the same project using the **Merge Projects** command under **Projects** in the menu bar.

The two codings appear as different documents shown in Figure 5.13.

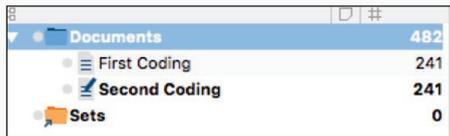


Figure 5.13: The two codings in parallel documents after merging projects in MAXQDA.

2. Run an intercoder analysis using the **Intercoder Agreement** command under **Analysis** in the menu bar. Choose **Segment Agreement** as your type of agreement and set the level to **100%** as shown in Figure 5.14.
3. Click on the **Excel** symbol in the upper right of the window containing the intercoder agreement results as shown in Figure 5.15.

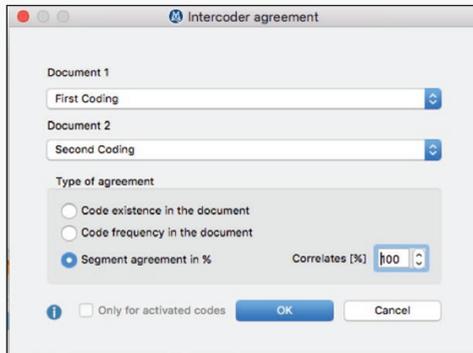


Figure 5.14: Running intercoder agreement analysis by segments at 100%.

Document	Code	Document 1	Document 2	Agree	Begin	End
First Coding	Research Intere...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	3	3
Second Coding	Research Intere...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	3	3
First Coding	Other	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	5	5
Second Coding	Other	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	5	5
First Coding	Research Intere...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	6	6
Second Coding	Research Intere...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	6	6
First Coding	Reputation	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	7	7
Second Coding	Reputation	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	7	7
First Coding	Reputation	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	9	9
Second Coding	Reputation	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	9	9
First Coding	Research Intere...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	10	10
Second Coding	Research Intere...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	10	10
First Coding	Other	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	11	11
Second Coding	Other	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	11	11
First Coding	Career	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	12	12
Second Coding	Career	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	12	12
First Coding	Workload	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	13	13
Second Coding	Knowledge & S...	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	13	13
First Coding	Career	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	14	14
Second Coding	Reputation	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	14	14
First Coding	Interpersonal D...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	15	15
Second Coding	Interpersonal D...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	15	15
First Coding	Interpersonal D...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	16	16

Figure 5.15: Intercoder agreement results.

This will open the data in Excel. To manipulate the data to put the two codings side by side:

4. Insert a second coding column next to the first one as shown in Figure 5.16. Change its format from **Text** to **General** using the **Number** tab of the **Cells** command under **Format** on the menu bar.

Continued ...



## MAXQDA Procedure 5.2: Putting the Two Codings Side by Side (continued)

<https://goo.gl/GDW4CW>

	A	B	C	D	E	F	G	H
1	Document name	First Coder	Second Coder	Document 1	Document 2	Agree	Begin	End
2	First Coding	Facilitator		1	1	1	1	1
3	Second Coding	Facilitator		1	1	1	1	1
4	First Coding	Research Interests		1	1	1	2	2
5	Second Coding	Research Interests		1	1	1	2	2
6	First Coding	Facilitator		1	1	1	3	3
7	Second Coding	Facilitator		1	1	1	3	3
8	First Coding	Other		1	1	1	4	4
9	Second Coding	Other		1	1	1	4	4
10	First Coding	Research Interests		1	1	1	5	5
11	Second Coding	Research Interests		1	1	1	5	5
12	First Coding	Reputation		1	1	1	6	6
13	Second Coding	Reputation		1	1	1	6	6
14	First Coding	Facilitator		1	1	1	7	7
15	Second Coding	Facilitator		1	1	1	7	7

Figure 5.16: Inserting a column for the second coder next to first coder column after opening MAXQDA data in Excel.

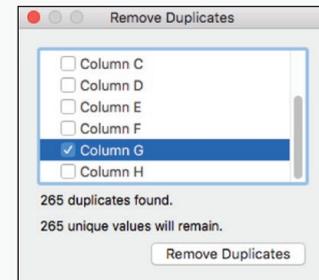


Figure 5.17: The Remove Duplicates dialog box in Excel.

5. Select these two cells and drag down to the end of the the data in the second column.
6. To fix these values in the second column, select the second column, copy it, and then, without moving your insertion point, use the **Paste Special Values** command.

Now the second column will contain fixed values for second coder codes rather than formulas.

7. Select all the data and use the **Remove Duplicates** command under the **Data** tab.
8. In the dialogue box, shown in Figure 5.17, choose the column with Begin as its header (Column G in our example). Click **Remove Duplicates**.
9. Filter the First Coder column for any dummy codes you may have used, and delete them in both the first and second coder columns.

In our data, this meant removing the Facilitator codes. The results should look like those shown in Figure 5.18.

	A	B	C	D	E	F
1	Document name	First Coder	Second Coder	Document 1	Document 2	Agree
2	First Coding			1	1	1
3	First Coding	Research Interests	Research Interests	1	1	1
4	First Coding			1	1	1
5	First Coding	Other	Other	1	1	1
6	First Coding	Research Interests	Research Interests	1	1	1
7	First Coding	Reputation	Reputation	1	1	1
8	First Coding			1	1	1
9	First Coding	Reputation	Reputation	1	1	1
10	First Coding	Research Interests	Research Interests	1	1	1
11	First Coding	Other	Other	1	1	1
12	First Coding	Career	Career	1	1	1
13	First Coding	Workload	Knowledge & Skill	1	0	0

Figure 5.18: MAXQDA intercoder agreement data manipulated in Excel to place first and second codings side by side in columns B and C.



## MAXQDA Procedure 5.3: Checking Item-by-Item Agreement for MAXQDA Data

<https://goo.gl/GDW4CW>

There is no easy way to check intercoder agreement for segmented verbal data in MAXQDA. You can calculate the Brennan-Prediger kappa (Brennan & Prediger, 1981) in MAXQDA by clicking on the kappa symbol in the upper left of the intercoder agreement results window in Figure 5.15. We do not recommend this method, but you can read the rationale of the developers of MAXQDA at <https://www.maxqda.com/help-max12/intercoder-agreement/the-agreement-testing-concept-in-maxqda> to decide for yourself.

Otherwise, once you have put your coding data side by side in Excel:

1. Type  
=IF
2. Click on the first code in the first column and then type  
=
3. Click on the first code in the second column and then type  
,1,0)
4. Hit enter.
5. Select the cell containing the new formula and drag it down next to each set of codes.

## ■ Calculating Simple Agreement Only

If you only want to calculate the rate of simple agreement between two coders, you can easily do this. Keep in mind, however, that using simple agreement only does not adjust for the degree of agreement that might occur just by chance. Especially if you have just a few categories, you will not want to rely just on a measure of simple agreement. Nevertheless, you may want to use the procedures shown in Excel Procedures 5.4 and MAXQDA Procedures 5.4 to quickly calculate simple agreement just to see how you're doing as you develop your coding scheme.



### Excel Procedure 5.4: Calculating Simple Agreement in Excel

<https://goo.gl/GDW4CW>

1. To calculate the sum of agreements, in the first cell below the data in your agreement column, type  
=SUM(
2. Then click and drag the cells above it (the 1s and 0s).
3. Type )
4. Then hit enter.
5. In the second cell down calculate the number of decisions by typing  
=COUNT(
6. Then click and drag the cells you want to count.
7. Type )
8. Then hit enter.
9. In the third cell down, calculate simple agreement by typing  
=
10. Click on the cell holding the sum.
11. Type /
12. Then click on the cell holding the count and hit enter.

## ■ Calculating Cohen's kappa only

You can rely on Cohen's kappa for an estimate of the reliability of your intercoder agreement if your marginals are pretty evenly distributed, showing little bias in the off-diagonal disagreements. Begin by making and formatting a table of agreements & disagreements (see Excel Procedures 5.5 and 5.6 and MAXQDA procedures 5.5 and 5.6) and then using that as input to GraphPad, an online calculator (see Excel Procedure 5.7 and MAXQDA Procedure 5.7).

The results of GraphPad's calculations are shown in Figure 5.24. The simple agreement is reported in the first line under the table as 65.58%. When this is corrected for agreement by chance using Cohen's kappa, the reliability is 56.2%. Note that GraphPad has a more generous understanding of what good intercoder reliability than the standards we described earlier.



### MAXQDA Procedure 5.4: Calculating Simple Agreement for MAXQDA Data

<https://goo.gl/GDW4CW>

1. To calculate the sum of agreements, in the first cell below the data in your agreement column, type  
=SUM(
2. Then click and drag the cells above it (the 1s and 0s).
3. Type )
4. Then hit enter.
5. In the second cell down calculate the number of decisions by typing  
=COUNT(
6. Then click and drag the cells you want to count.
7. Type )
8. Then hit enter.
9. In the third cell down, calculate simple agreement by typing  
=
10. Click on the cell holding the sum.
11. Type /
12. Then then click on the cell holding the count and hit enter.



## Excel Procedure 5.5: Making a Table of Agreements & Disagreements for Excel Data

<https://goo.gl/GDW4CW>

1. Make sure that there is nothing in the three columns of inter-coder agreement data except the coding decisions and the agreement calculations.
2. Select the three columns of intercoder agreement data (First Coder, Second Coder, Agreement).
3. Use the **Summarize with Pivot Table** command under **Data** on the menu bar to create a pivot table.
4. Use the defaults for the data range and where to place the pivot table (a new worksheet) and click **OK**.

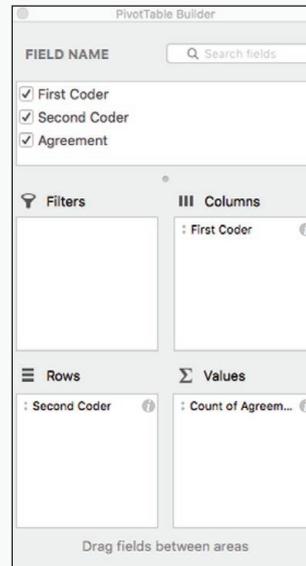


Figure 5.19: Selecting the parameters for the pivot table.

As shown in Figure 5.19:

5. Drag First Coder to the **Columns** field.
6. Drag Second Coder to the **Rows** field.
7. Drag Agreement to the **Values** field.
8. Finally, in the **Values** field, change Sum to Count by clicking on the **i** symbol to the right of **Agreement** in the **Values** field and as shown in Figure 5.20.

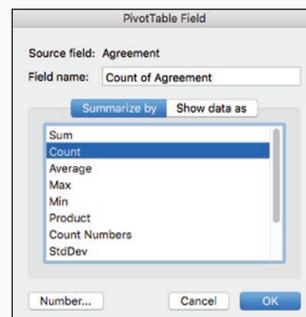


Figure 5.20: Changing Sum to Count in the PivotTable Field for Values.



## Excel Procedure 5.6: Formatting a Table of Agreements & Disagreements for Excel Data

<https://goo.gl/GDW4CW>

1. Select just the body of the pivot table beginning with Row Labels and Ending with the Grand Total as shown in the Figure 5.21.

Row Labels	career	funding	interpersonal	Knowledge	other	reputation	Research	Grand Total
Career	17	0	13	0	10	11	2	53
Funding	0	11	0	0	1	0	0	12
Interpersonal	1	0	1	0	3	0	0	5
Knowledge	0	0	1	0	0	0	0	1
Other	1	0	1	0	46	0	0	48
Reputation	2	0	3	0	0	16	0	21
Research	1	0	2	0	1	0	10	14
<b>Grand Total</b>	<b>22</b>	<b>11</b>	<b>21</b>	<b>0</b>	<b>61</b>	<b>27</b>	<b>12</b>	<b>154</b>

Figure 5.21: A table of agreements & disagreements created from the data using a pivot table.

2. Copy the selected portion and then paste it below the original pivot table.
3. Compare the codes listed across with those listed down. If one or more codes are missing, add a column or row for each missing code.

In Figure 5.19, we added a column for Knowledge, which was missing from the pivot table because Coder 1 did not use it.

4. Delete any columns or rows labeled blank or with dummy codes.
5. Add gridlines with the **Borders** menu on the **Home** ribbon.
6. Fill every other row with color.
7. Fill values on the diagonal with yellow to highlight the agreements.



## MAXQDA Procedure 5.5: Making a Table of Agreements & Disagreements for MAXQDA Data

<https://goo.gl/GDW4CW>

Once your MAXQDA data is in Excel with Item-by-Item agreement calculated, you can make a table of agreements & disagreements:

1. Select the three columns of intercoder agreement data (First Coder, Second Coder, Agreement).
2. Use the **Summarize with Pivot Table** command under **Data** on the menu bar to create a pivot table.
3. Use the defaults for the data range and where to place the pivot table (a new worksheet) and click **OK**.

As shown in Figure 5.19:

4. Drag First Coder to the **Columns** field.
5. Drag Second Coder to the **Rows** field.
6. Drag Agreement to the **Values** field.
7. Finally, in the **Values** field, change Sum to Count by clicking on the **i** symbol to the right of **Agreement** in the **Values** field and as shown in Figure 5.19.



## MAXQDA Procedure 5.6: Formatting a Table of Agreements & Disagreements for MAXQDA Data

<https://goo.gl/GDW4CW>

1. Make a copy of the pivot table beginning with row labels and ending with the grand total below the original pivot table.
2. Compare the codes listed across with those listed down. If one or more codes are missing, add a column or row for each of the missing codes.

In Figure 5.21, we added a column for Knowledge, which was missing from the pivot table because Coder 1 did not use it.

3. Delete any columns or rows labeled blank or with dummy codes (such as Facilitator).
4. Add gridlines with the **Borders** icon.
5. Fill every other row with color.
6. Fill values on the diagonal with yellow to highlight the agreements.

## Exercise 5.2 Try It Out

You can download this exercise at <https://wac.colostate.edu/books/practice/codingstreams/>).

Create a table of agreements & disagreements for the data found in the linked worksheet using a pivot table. Format it to highlight the agreements.

**For Discussion:** Are you satisfied with this level of agreement?



### Excel Procedure 5.7: Using GraphPad's Online Calculator for Cohen's Kappa for Excel Data

<https://goo.gl/GDW4CW>

Make sure you have Table of Agreements & Disagreements as input.

1. Access the input screen for GraphPad's Online Calculator for Cohen's Kappa, shown in Figure 5.22, at <https://graphpad.com/quickcalcs/kappa2/>
2. Select the number of categories in your pivot table.
3. Type the data into the browser window as shown in Figure 5.23.
4. Click **Calculate Now**.

**GraphPad Software** Scientific Software Data Analysis Resource Center

## QuickCalcs

1. Select category 2. Choose calculator 3. Enter data 4. View results

### Quantify agreement with kappa

This calculator assesses how well two observers, or two methods, classify subjects into groups. The degree of agreement is quantified by kappa.

1. How many categories?

Into how many categories does each observer classify the subjects? For example, choose 3 if each subject is categorized into 'mild', 'moderate' and 'severe'.

8 categories **Caution: Changing number of categories will erase your data.**

2. Enter data

Each cell in the table is defined by its row and column. The rows designate how each subject was classified by the first observer (or method). The columns designate how the other observer (or method) classified the subjects. For example, enter into the second row of the first column the number of subjects that the first observer classified into category 'B' and the second observer classified into category 'A'. Enter into the second column of the second row the number of subjects classified by both observers into category 'B'. Subjects categorized identically by both observers are always tallied in the diagonal of the table.

	A	B	C	D	E	F	G	H
A	<input type="text"/>							
B	<input type="text"/>							
C	<input type="text"/>							
D	<input type="text"/>							
E	<input type="text"/>							
F	<input type="text"/>							
G	<input type="text"/>							
H	<input type="text"/>							

3. View the results

Figure 5.22: GraphPad's online calculator for Cohen's kappa.



## MAXQDA Procedure 5.7: Using GraphPad's Online Calculator for Cohen's Kappa for MAXQDA Data

<https://goo.gl/GDW4CW>

Make sure you have Table of Agreements & Disagreements as input.

1. Access the input screen for GraphPad's Online Calculator for Cohen's Kappa, shown in Figure 5.22, at <https://graphpad.com/quickcalcs/kappa2/>.
2. Select the number of categories in your pivot table.
3. Type the data into the browser window as shown in Figure 5.23.
4. Click **Calculate Now**.

A	B	C	D	E	F	
0	13	0	10	11	2	
11	0	0	1	0	0	
0	1	0	3	0	0	
0	1	0	0	0	0	
0	1	0	46	0	0	
0	3	0	0	16	0	
0	2	0	1	0	10	>

Figure 5.23: Entering data from a pivot table into GraphPad's calculator for Cohen's kappa.

	A	B	C	D	E	F	G	Total
A	17	0	13	0	10	11	2	53
B	0	11	0	0	1	0	0	12
C	1	0	1	0	3	0	0	5
D	0	0	1	0	0	0	0	1
E	1	0	1	0	46	0	0	48
F	2	0	3	0	0	16	0	21
G	1	0	2	0	1	0	10	14
Total	22	11	21	0	61	27	12	154

Number of observed agreements: 101 ( 65.58% of the observations)

Number of agreements expected by chance: 32.9 ( 21.36% of the observations)

Kappa= 0.562

SE of kappa = 0.046

95% confidence interval: From 0.473 to 0.652

The strength of agreement is considered to be 'moderate'.

*Figure 5.24: GraphPad results for Cohen's kappa.*

## Calculating Both Krippendorff's alpha and Cohen's kappa

As we mentioned in the first half of this chapter, using Cohen's kappa alone can run the risk of over-stating your level of agreement if your marginals show bias. To get both Cohen's kappa and Krippendorff's alpha, you can use the ReCal2 online calculator as described below. To use this calculator, you must first put the agreement data in numeric form and then save it in the CSV format. See Excel Procedures 5.8 and 5.9, Procedure 5.1, and MAXQDA Procedures 5.8 and 5.9.



## Excel Procedure 5.8: Converting Codes to Numeric Values for Excel Data

<https://goo.gl/GDW4CW>

To convert verbal codes into numeric codes:

1. Make a copy of your worksheet showing the intercoder agreement data.
2. In the duplicate worksheet, insert 2 new columns next to your original two coding columns.
3. Copy the contents of the original columns into the 2 new columns.
4. Temporarily make a list of your codes and assign each one a number, beginning with 1.
5. Select the two newly copied columns.
6. Select the **Replace** command under the **Find** option of the **Edit** menu.
7. Enter your first non-numeric code and the numeric value you want to replace it with.
8. Click **Replace All**.
9. Continue in this manner until you have replaced all of your non-numeric codes.
10. After you have replaced all of your verbal codes with numeric codes, delete any uncoded data as well as all columns to the left of the numeric codes.

The result should be a file with just two columns of numeric codes.



## Excel Procedure 5.9: Saving to Alternative File Formats for Excel Data.

<https://goo.gl/GDW4CW>

The online calculator for Cohen's kappa and Krippendorff's alpha only accepts files in CSV format.

1. Use the **Save As** command under **File** on the menu bar.
2. Select **Comma Separated Values (.CSV)** from the drop down menu under **File Format**.
3. Click **Save**.



## Procedure 5.1: Using the ReCal2 Calculator for Cohen's Kappa & Krippendorff's Alpha

<https://goo.gl/GDW4CW>

1. Go to the ReCal2 online calculator at <http://dfreelon.org/utis/recalfront/recal2/>
2. Click **Choose File** and select the CSV file containing your numerically coded data.
3. Click **Calculate Reliability**.

The results of ReCal2's calculations are shown in Figure 5.25. Here we see the same results for simple reliability (65.6%) and Cohen's kappa (56.2%) as well as similar results (55.5%) for Krippendorff's alpha.

ReCal 0.1 Alpha for 2 Coders results for file "csv formatted data.csv"							
File size:		806 bytes					
N columns:		2					
N variables:		1					
N coders per variable:		2					
	Percent Agreement	Scott's Pi	Cohen's Kappa	Krippendorff's Alpha (nominal)	N Agreements	N Disagreements	N Cases
)	65.6%	0.554	0.562	0.555	101	53	154

Figure 5.25: Results from ReCal2.



## MAXQDA Procedure 5.8: Converting Codes to Numeric Values for MAXQDA Data

<https://goo.gl/GDW4CW>

To convert verbal codes into numeric codes for MAXQDA data in Excel:

1. Make a copy of your worksheet showing the intercoder agreement data.
2. In the duplicate worksheet, create 2 new columns next to your original two coding columns.
3. Make a list of your codes and assign each one a number, beginning with 1.
4. Select the two newly copied columns.
5. Select the **Replace** command under the **Find** option of the **Edit** menu.
6. Enter your first non-numeric code and the numeric value you want to replace it with.
7. Click **Replace All**.
8. Continue in this manner until you have replaced all of your non-numeric codes. After you have replaced all of your verbal codes with numeric codes, delete any uncoded data as well as all columns to the left of the numeric codes.



## MAXQDA Procedure 5.9: Saving to Alternative File Formats for MAXQDA Data

<https://goo.gl/GDW4CW>

The online calculator for Cohen's kappa and Krippendorff's alpha only accepts files in CSV format.

1. Use the **Save As** command under **File** on the menu bar.
2. Select **Comma Separated Values (.CSV)** from the drop down menu under **File Format**.
3. Click **Save**.

### Exercise 5.3 Test Your Understanding

You can download this exercise at <https://wac.colostate.edu/books/practice/codingstreams/>.

A set of 20 data segments were coded using two different coding schemes. One had 10 categories (A-J) and the other had five categories (A-E). When the researcher went to check the reliability of each scheme using second coders, the simple agreement using in both cases was pretty poor—the second coders agreed with her only 50% of the time.

Look at the table of agreements and disagreements and the corrected reliability for these two schemes given in Figure 5.26 and in the worksheet at the link above. Are both schemes equivalent in terms of their reliability; or is one more reliable than the other?

	a	b	c	d	e	f	g	h	i	j		
A	1										1	
B		1				1			1		3	
C			1								1	
D				1							0	
E					3	1					4	
F						1	1				2	
G							1				1	
H		2						1	1		4	
I									1	1	2	
J											2	2
	1	3	0	0	4	2	2	3	3	2	20	

Simple Agreement: .50  
Kappa: 0.422

	a	b	c	d	e	
A	1					
B	1	2				
C		2	1			
D			1	1		
E					5	
	3	5	3	3	6	
						20

Simple Agreement: .50  
Kappa: 0.36

Figure 5.26: Sample table of agreements and disagreements for Exercise 5.3.

**For Discussion:** Be prepared to explain your answer to your classmates.

## ■ Memo 5.3: Intercoder Agreement

Calculate measures of intercoder agreement between your two coders. Calculate simple reliability in Excel and corrected reliability using an online calculator for either Cohen's kappa, Krippendorff's alpha, or both.

Document the results of your calculations, making sure to clearly identify the coders, the data sample, and the version of the coding scheme that you used.

## ■ Revising Your Analytic Procedures

Increasing the reliability of a coding scheme involves inspecting the disagreements between coders for each category, identifying probable causes, and then revising your analytic procedures to eliminate them.

### ■ Inspecting Your Disagreements

Begin by looking at your table of agreements & disagreements to identify combinations in which there are disagreements. They often cluster in just a few areas. For example, the table of agreements & disagreements shown in Figure 5.27 suggests that the second coder is using the code *Career* far more often than Coder 1.

	Career	Funding	Interpersonal	Knowle	Other	Reputation	Research	
Career	17	0	13	0	10	11	2	53
Funding	0	11	0	0	1	0	0	12
Interpersonal	1	0	1	0	3	0	0	5
Knowledge	0	0	1	0	0	0	0	1
Other	1	0	1	0	46	0	0	48
Reputation	2	0	3	0	0	16	0	21
Research	1	0	2	0	1	0	10	14
	22	11	21	0	61	27	12	154

Figure 5.27: Table of agreements & disagreements.

Next return to your coding sheet and use Autofilter to look at one combination at a time. Returning to our data sheet, as shown in Figure 5.28, we filter Column E to show all data that was coded with *Career* by the second coder. We could further filter Column D a code at a time to look at the choices made

by the first coder. Looking at the data and the coding scheme, we try to understand the nature and cause of the disagreements.

D	E	F	G	H
First Code	Second Coder	Agreement	FirstCoderNumeric	SecondCoderNumeric
interpersonal	Career			
career	Career			
interpersonal	Career			
interpersonal	Career			
career	Career			
career	Career			
career	Career			
other	Career			
other	Career			
reputation	Career			
other	Career			
Research	Career			
reputation	Career			
reputation	Career			
reputation	Career			
career	Career			
reputation	Career			
career	Career			

First Coder

**Sort**

By color:

**Filter**

By color:

Choose One

- (Select All)
- career
- interpersonal
- other
- reputation
- Research

Figure 5.28: Filtering on disagreements over the use of the code Career.

## Exercise 5.4 Try It Out

You can download this exercise at <https://wac.colostate.edu/books/practice/codingstreams/>.

Roger created a 4-category coding scheme and applied it to a 99-segment sample of data. When he checked the level of agreement with a second coder, he was happy to find that his simple reliability was high: 80%. But when he looked at the corrected agreement using Cohen's kappa, he was concerned. It was only .42. His table of agreements and disagreements looked like the one shown in Figure 5.29.

	N	P	R	O	SUM
N	5	4	0	0	9
P	3	71	0	0	74
R	2	2	1	0	5
O	1	8	0	2	11
SUM	11	85	1	2	99

*Figure 5.29; Table of agreements and disagreements for Exercise 5.4.*

He is considering three different strategies to improve this reliability:

- Revise the definition of category *N* to eliminate the second coder's confusion with category *P*.
- Revise the category *O* to eliminate the coder's confusion with categories *N*, and *P*.
- Eliminate the category *O* altogether, including it in *R*.

Modify his coding data (available at the link above) in one of these three ways; then recalculate his simple and corrected reliability.

**For Discussion:** Based on the results of you and your classmates, which of the three strategies do you recommend that Roger adopt? What generalization might you make about the best strategies for improving reliability using the Table of Agreements & Disagreements as a guide?

## ■ Revising the Coding Scheme

In the simplest cases, disagreements between coders arise from lack of clarity in the coding scheme. By adding cases and examples of those cases, we can often better indicate to a coder that certain kinds of verbal data should go in one category rather than another. In the coding scheme found in Chapter 4, Figure 4.1, for example, coders were initially inconsistent in how they categorized t-units which contained phrases such as “definition” and “justification.” After reflection it became clear that such words signal attention to the discourse functions of a text and therefore should be coded as *Rhetorical Process*. The following case with examples under *Rhetorical Process* clarified this decision and eliminated this kind of confusion:

**“general categories of claims that can be made by authors: “a definition,” “a justification,” “a reason,” “a question”;**

Occasionally, you will find that some verbal phenomena consistently confuse your coders and need to be addressed explicitly. In the example coding scheme on Worlds of Discourse, for example, t-units with “you” in them always confused coders. Sometimes they were coded as *Rhetorical Process*, and sometimes as *Narrated Cases*. Looking at these disagreements, I realized a need to explicitly address the use of “you” that should be included in *Narrated Cases*, which I did with this addition to the coding scheme under *Narrated Case*:

**“you” or “I” when cast in a role involving an action that is taken to exist independently of the concepts in the domain but that may potentially be characterized with respect to these concepts.**

Finally, coding schemes can be revised to add categories or refocus definitions of categories so that analytic constructs are better understood. It was this kind of move that prompted me to add the category of *Narrated Cases* to my original scheme which had only included *Rhetorical Process* and *Domain Content* on its start list.

## ■ Changing the Unit of Analysis

More complex revisions to analytic processes can be made by changing the

unit of analysis. As described in Chapter 3, if the unit of analysis is inappropriate to the phenomenon of interest, coders will have great difficulty using a coding scheme. If the unit is too large, more than one category may apply. If the unit is too small, coders may not recognize the phenomenon as it is broken across segments. To remedy these problems, return to the original data in Word, resegment, recode, and then compare the results of a second coding.

## ■ Adding Another Dimension

As described in Chapter 4, we often find ourselves placing too much into a single coding scheme, trying to ask a coder to look for things that are, in essence, quite different. It is as if we were to ask a coder to tell us, “is it a yellow chick or a brown goat?” and then finding they do not know what to do with brown chicks. We could revise our coding scheme to direct the coder to put any brown chick into the Yellow Chick category, but this might violate coders’ intuitions and lead to inconsistent coding. A better approach might be to realize that our scheme had conflated two different dimensions, color and animal type.

If you realize that you have conflated dimensions of a phenomenon into a single coding scheme, you will need to break your scheme into two different schemes and code with each one separately. We would, for example, ask our coders to first decide whether the animal was a chick or a goat and only later ask whether it was yellow or brown.

## ■ Moving to Nested Coding

Another option in dealing with what looks like distinct dimensions of a phenomenon is to move to a nested coding scheme as described in Chapter 4. Suppose, for example, that we do not care much about goats in the previous example, but only want to look at the chicks. In this case, rather than code all the data for color, we might code in two stages. In the first stage, we would ask our coders to decide if it was a chick or not. Only if the answer were yes, would we go on to ask whether the chick were yellow or brown.

## ■ Acknowledging the Limits of Interpretation

Finally, in inspecting disagreements, you may realize that your judgments are relying on knowledge so contextualized that you could not expect a second coder to duplicate your judgments. In this case, you have two alternatives.

One choice is to move to an enumerative coding scheme where you list all of the cases which you, with your deep knowledge of the context, judge to be in a given category. Such an enumerative scheme can go a long way in communicating to your readers the substance of an analytic construct.

A second option is to admit the limits of interpretation on the analytic construct, attempt to describe and illustrate it as best you can in your report, but abandon the attempt to get a high level of agreement with a second coder. This step should be viewed as a last resort, of course, because what we often start out thinking cannot be made explicit can be done with more thought. Nevertheless, it is important not to narrow one's vision of verbal data in a way that unduly favors the relatively transparent (and perhaps less important) over that which is relatively opaque (and perhaps more important).

Occasionally, you will find that you have not reached anywhere near satisfactory levels of reliability even after several rounds of second coding. In such cases, you will want to step back from the analysis and think through the analytic constructs with which you are working. They may be unclear. Or the data may simply not be describable in their terms. Quite frankly, you may be looking for the rabbit in the wrong hole. While no one likes to abandon an analysis, sometimes that is the best course. Often, you can return to it at a later time when a fresh perspective and further insight may give you better guidance.

### ■ Memo 5.4: Revisions for Reliability

Review your table of agreements and disagreements to identify the areas that are causing the greatest disagreement. Examine each combination in the data and then develop a strategy for improvement. Make appropriate revisions to the coding scheme, to the segmenting procedure, or to the dimensions of the analysis.

Document your revisions and plan your next step to achieving an appropriate level of reliability.

---

## ■ Finalizing Reliability

Once you have revised your analytic procedures, you should repeat the process of working with a second coder until an adequate level of reliability is reached. Generally, you would like to see simple agreements of .80 or better, corrected agreement of 70% or better. This can usually be reached after two or three rounds of second coding.

Once you have reached this level of agreement, you need to take a new and as yet uncoded section of data, and give it to a new second coder, someone who has not yet worked with you. The level of agreement you achieve with this fresh data and fresh coder, along with the stabilized coding scheme that produced it, is what you include in your report of the analysis.

Generally speaking, you need not have this final second coder code your entire data set, unless the set is small enough that the task can be completed in a reasonable time. Since verbal data sets tend to be quite large, a more selective approach to final second coding is required. As mentioned earlier, at least 10% of the data ought to receive a second coding, with each kind of data represented. Your overall goal is to verify the reliability of your coding scheme on the full breadth and depth of the data even when it cannot all be coded twice.

After data has received a final second coding, you will still find disagreements between coders. To prepare the data for further analyses as we describe in the next few chapters, you will need to reconcile those disagreements. Inspect each one carefully and decided which coding decision to adopt.

Make sure to retain records of each round of your coding attempts, which version of the scheme was used, what level of agreement was received, how each segment of data was coded by each coder. For this reason, it is probably best to create a separate coding workbook for each round of coding and to label it with the date of the second coding. Then, if ever necessary, you can go back and recover your steps.

## ■ Memo 5.5: Final Reliability

Once you have reached an acceptable level of reliability, document the final coding scheme, the data used to achieve this level of reliability, and its reliability statistics.

---

## ■ Selected Studies Reporting Reliability

- Campbell, K. S., & Naidoo, J. S. (2017). Rhetorical move structure in high-tech marketing white papers. *Journal of Business and Technical Communication*, 31(1), 94-118.
- De Groot, E., Nickerson, C., Korzilius, H., & Gerritsen, M. (2016). Picture this: Developing a model for the analysis of visual metadiscourse. *Journal of Business and Technical Communication*, 30(2), 165-201.
- Felton, M., Crowell, A., & Liu, T. (2015). Arguing to agree: Mitigating my-side bias through consensus-seeking dialogue. *Written Communication*, 32(3), 317-331.
- Graham, S. S., Kim, S.-Y., DeVasto, D. M., & Keith, W. (2015). Statistical genre analysis: Toward big data methodologies in technical communication. *Technical Communication Quarterly*, 24 (1), 70-104.
- Hyland, K. & Jiang, F. (K.). (2016). Change of attitude? A diachronic study of stance. *Written Communication*, 33(3), 251-274.
- Shin, W., Pang, A., & Kim, H. J. (2015). Building relationships through integrated online media: Global organizations' use of brand web sites, Facebook, and Twitter. *Journal of Business and Technical Communication*, 29(2), 184-220.

## ■ For Further Reading

- Banerjee, M., Capozzoli, M., McSweeney, L., & Sinha, D. (1999). Beyond kappa: A review of interrater agreement measures. *Canadian Journal of Statistics*, 27( 1), 3-23.
- Brennan, R., & Prediger, D. J. (1981). Coefficient kappa: Some uses, misuses, and alternatives. *Educational and Psychological Measurement*, 41(3), 687-99.
- Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1), 37-46. <https://doi.org/10.1177/001316446002000104>.

- Gaskell, G., & Bauer, M. W. (2000). Towards public accountability: Beyond sampling reliability, and validity. In M. W. Bauer & G. Gaskell (Eds.), *Qualitative Researching with Text, Image, and Sound* (pp. 336-350). Thousand Oaks, CA: Sage.
- Geisler, C. (1994). *Academic literacy and the nature of expertise: Reading, writing, and knowing in academic philosophy*. Hillsdale, NJ: Lawrence Erlbaum Associates.
- Goetz, J. P., & LeCompte, M. D. (1984). *Ethnography and qualitative design in educational research* (pp. 211-220). Orlando, FL: Academic Press.
- Krippendorff, K. (2004). Reliability in content analysis: Some common misconceptions and recommendations. *Human Communication Research*, 30(3), 411-433.
- Krippendorff, K. (2013a). *Computing Krippendorff's alpha-reliability*. Retrieved from [https://repository.upenn.edu/asc\\_papers/43/](https://repository.upenn.edu/asc_papers/43/)
- Krippendorff, K. (2013b). *Content Analysis: An Introduction to its methodology* (3<sup>rd</sup> ed.). Los Angeles: Sage.
- Landis, J. R., & Koch, G. G. (1977). The measurement of observer agreement for categorical data. *Biometrics*, 33(1), 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.
- Miles, M. B., & Huberman, A. M. (1994). *Qualitative data analysis: An expanded sourcebook* (pp. 277-280). Thousand Oaks, CA: Sage.
- Neuendorf, K. (2016). *The content analysis guidebook*. London: Sage Publications.
- Popping, R. (1988). On agreement indices for nominal data. In W. E. Saris & I. N. Gallhofer (Eds.), *Sociometric research: Volume 1, data collection and scaling* (pp. 90-105). New York: St. Martin's.
- Saldaña, J. (2016). *The coding manual for qualitative researchers*. London: Sage.