UNIVERSITY OF AMSTERDAM

Calculating inter-coder reliability in media content analysis using Krippendorff's Alpha

Knut De Swert (University of Amsterdam)

– First version (01/02/2012).

Please send your comments, suggestions, additions and corrections to k.deswert@uva.nl . - Thanks !

For researchers doing content analysis, inter-coder reliability is crucial. However, there does not seem to be a general standard on how to do and report inter-coder reliability tests. In academic publications, a variety of measures is presented. The reasons for this lack of uniformity is not so much that there are technical disagreements between researchers on which measure would be best, but rather the lack of sufficient information about inter-coder reliability testing, how this test is calculated and how the results from this test should be interpreted.

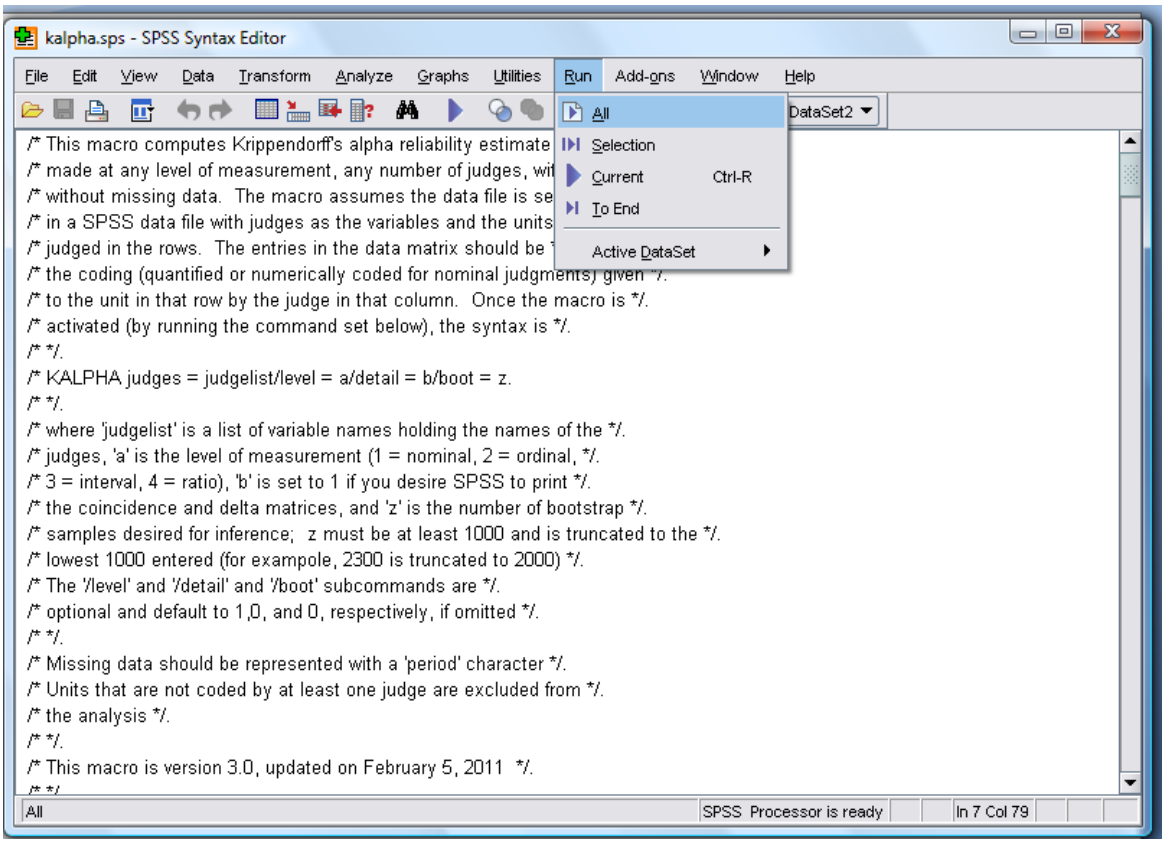
Hayes & Krippendorff (2007 - www.af**hayes**.com/public/**kalpha**.pdf) have clarified convincingly why Krippendorff's alpha (Further: KALPHA) should be the basic measure to apply for most researchers. Sample size, multiple (more than 2) coders or missing data are not problematic for calculating KALPHA, and all measurement levels can be tested. Thus, this manual starts off from that point, acknowledging KALPHA as the appropriate reliability test (for more discussion about value and choice of tests, see also Krippendorff 2010).

For actually running KALPHA tests, content analysis researchers owe a lot to the work of Hayes (2005) He developed a macro to make KALPHA calculation possible in SPSS, thus making KALPHA calculations easily accessible for the larger scientific community. Of course, more sophisticated software packages for calculation inter-coder reliability exist (e.g. PRAM, which calculates a variety of reliability tests), but this manual will work with KALPHA and in SPSS only.

This manual is intended to be highly accessible, both by students and researchers. The idea is that it is understandable without using and understanding formulas. In that way, it is meant to be complementary to existing (excellent, and more sophisticated but generally less concrete) overviews of how to calculate inter-coder reliability using KALPHA like e.g. Krippendorff (2011 - http://repository.upenn.edu/asc_papers/43)

KALPHA in SPSS

You will need the macro by Hayes (go to http://www.afhayes.com/spss-sas-and-mplus-macros-and-code.html and look for KALPHA.sps). You will need to download the macro, do not try to cut and paste it. Once you have downloaded and saved it to your computer as KALPHA.sps, you can open SPSS and then you can open this macro ("open syntax") and run it. Once that is done, you can use the command to calculate KALPHA (see infra). You will need to run the macro again whenever you start a new session of SPSS.



Before testing reliability using KALPHA in SPSS: What do you need?

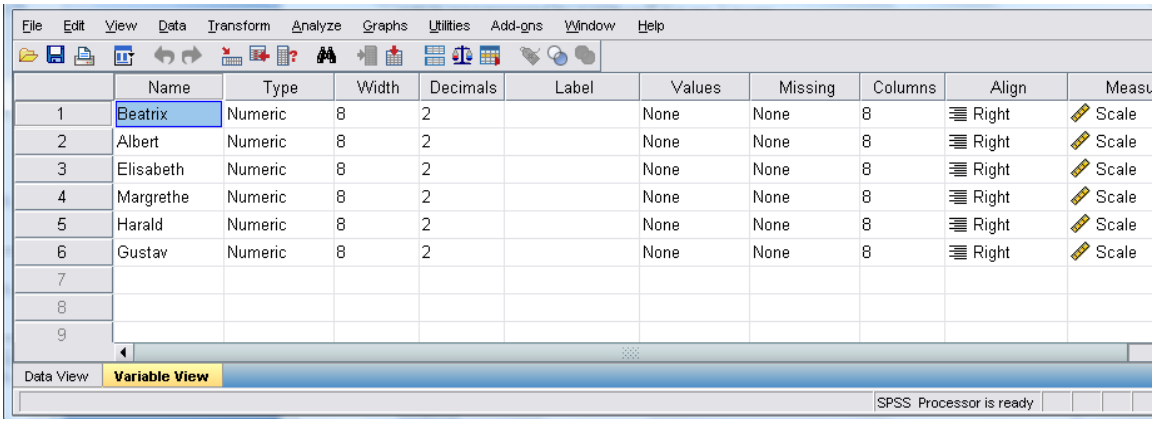
Naturally, to test inter-coder reliability, you need a subsample of your data to be coded (independently) by all the coders involved in the coding of the complete dataset. Ideally, the whole subsample is done by all coders, but KALPHA can deal with missing values pretty well. Considering the size of the subsample, ten percent of the complete dataset is an often found guideline. However, for small datasets a higher percentage might be more comfortable (see infra), and depending on the complexity and rareness of certain variables or crucial categories, one might also consider to increase the number of items included in the subsample for inter-coder reliability testing.

As said, KALPHA can be calculated for any number of coders, so it is not very efficient to engage in paired double coding. KALPHA is calculated per variable. It provides information on the reliability of variables, not of coders (even if structural patterns of different codings by certain coders may become apparent from merely preparing the data for the SPSS file you are going to use to test the reliability). For each variable you want to test for inter-coder reliability, you will need to make a separate SPSS file to calculate KALPHA.

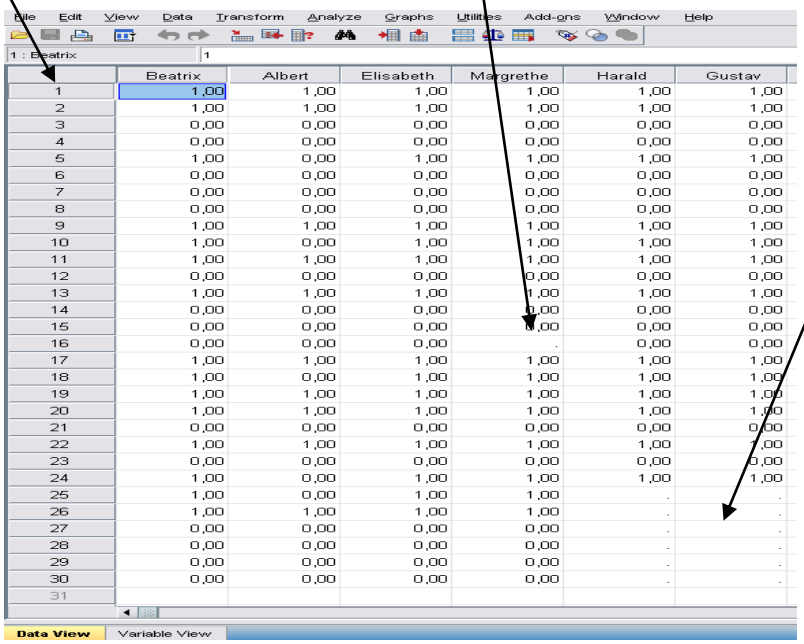
Preparing for KALPHA: how to prepare the SPSS file for a variable you want to check?

For the macro of Hayes to work, you will need to prepare the data in a way that involves the transformation of your dataset. The coders are to be placed as variables, and the different units coded are the cases. In the following example, we use fictive data on a variable that should be easily trainable and could be expected to end up with a high inter-coder reliability score. The items coded are tabloid news paper articles, and the variable tested here (V1) is dichotomous: "*Is the text of the article accompanied by a picture?*" 0 is no, 1 is yes.

There are six coders in this example. Insert them as variables in the Variable View screen in SPSS.



The subsample for the reliability test consists out of 30 tabloid newspaper articles. For example, on line 1, we find the coded value for variable V1 of the first article by all six coders (they all indicated that there was a picture with the article), etc. Most articles in this example were coded by all coders (as is preferable), except for the last articles, which were (for whatever reason) not coded by Harald and Gustav. Another missing is to be found for article 16 by coder Margrethe, which could be due to forgetfulness, input error or uncertainty of the coder. As we said earlier, missings like this (which are realistic) are unproblematic for KALPHA calculation, so these articles can stay in the subsample.



Running the KALPHA command

The basic command you need to give to SPSS (in a syntax) is quite simple. Thanks to the macro of Hayes you ran earlier, SPSS will recognize this command.

KALPHA judges = *judgelist*/level = *lev*/detail = *det*/boot *= z.*

Of course, you will need to customize this command according to your specific test.

== > Where it says "*judgelist*", you need to line up the coders (in this case the six names).

== >Where it says "*lev*", you need to add information on the measurement level of the variable you are testing.

Measurement level of tested variable	***Value to be added (level =)***
NOMINAL	"1"
ORDINAL	"2"
INTERVAL	"3"
RATIO	"4"
DICHOTOMOUS/BINARY	"1" (treat as nominal)

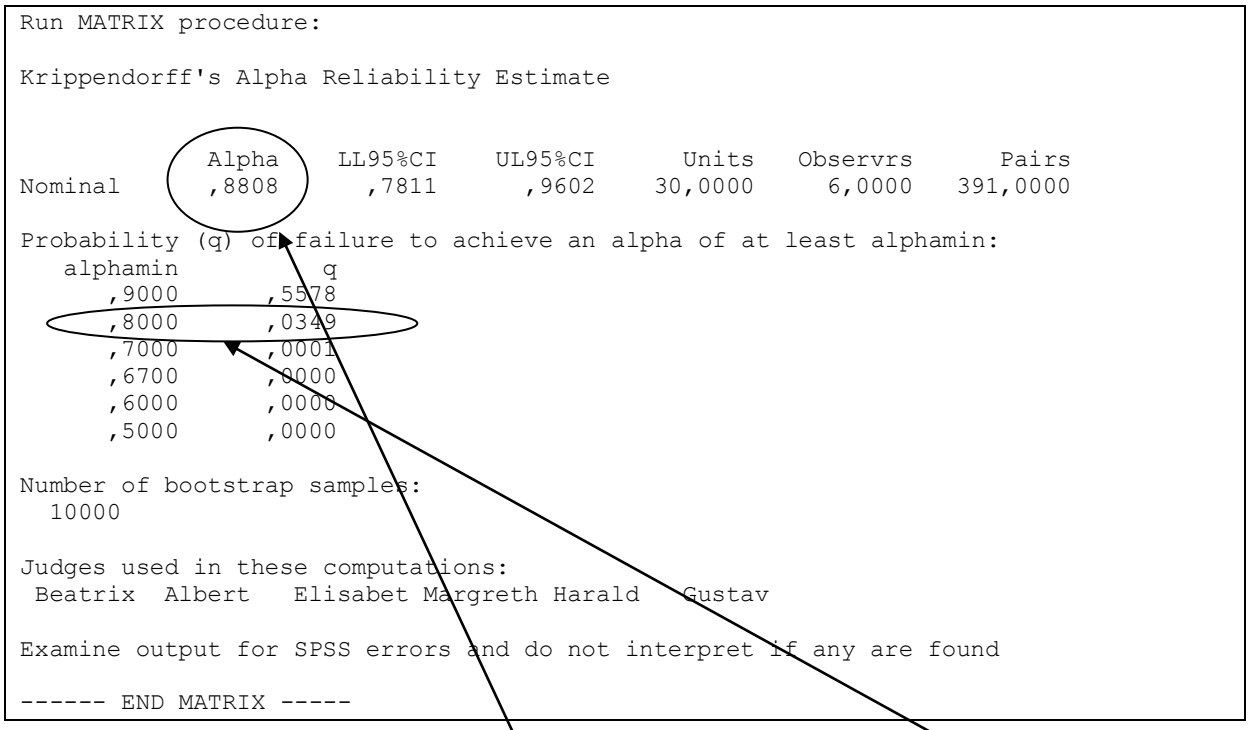
== > Where it says "*detail*", you can add "1" to get more detailed output. If you are only interested in the KALPHA value itself, you could fill out "0" or leave the detail part out of the command (default value is 0).

- ⇨ Where it says "z", you could determine how much bootstrapping you want SPSS to perform. This could also be dropped if considered unnecessary. Bootstrapping can be useful to allow statistical inferences about the population even when you only have a relatively small sample. It basically tells you how probable it is that KALPHA would be above or below certain border values (.90, .80, .70 etc.) if the whole population would have been tested for reliability. In the following examples, only the first one will show output of a KALPHA calculation with bootstrapping as an illustration.

So, if we want to know the KALPHA for this variable about presence of pictures in news paper articles, this is the command you would need:

KALPHA judges = Beatrix Albert Elisabeth Margrethe Harald Gustav/level = 1/detail = 0/boot = 10000.

The six coders are all included; the measurement level is set to 1 (nominal for the dichotomous variable). This is the SPSS output (note that we put detail=0 in the command to suppress additional information that would lead us too far at this moment).



```
Run MATRIX procedure:

Krippendorff's Alpha Reliability Estimate


            Alpha     LL95%CI     UL95%CI       Units    Observrs       Pairs
Nominal     ,8808       ,7811       ,9602     30,0000      6,0000    391,0000

Probability (q) of failure to achieve an alpha of at least alphamin:
   alphamin          q
      ,9000      ,5578
      ,8000      ,0349
      ,7000      ,0001
      ,6700      ,0000
      ,6000      ,0000
      ,5000      ,0000

Number of bootstrap samples:
  10000

Judges used in these computations:
 Beatrix  Albert   Elisabet Margreth Harald   Gustav

Examine output for SPSS errors and do not interpret if any are found

------ END MATRIX -----
```

This test gives us a good result, Kalpha = .88, which is quite high. Additionally, the bootstrapping procedure indicates that there is only 3.49 percent chance that the KALPHA would be below .80 if the whole population would be tested.

Kalpha=.80 is often brought forward as the norm for a good reliability test, with a minimum of .67 or even .60 (when it is that low, you might give some specific information why this is low and why you still choose to accept this variable in your analysis). However, a .80 or higher Kalpha should not always be considered satisfactory. When the variable is (in the eyes of the researcher) extremely easy to code (like in this example the presence of a picture, or gender etc.), one might raise the standards a little bit. In the case of this example, .88 seems great, but a closer look at the data entered in SPSS reveals that Kalpha would probably be much higher if one particular coder did not perform so badly (i.e. Albert – see data on the previous page).

In essence, Kalpha is a measurement of the reliability <u>of the variable</u> (i.e. actually the combination of the clarity of the variable description and the categories, the information and background in the codebook and the instructions given during training). If Kalpha is low, one might look for an explanation in one of these elements. However, if you calculate Kalpha on codings by a group of coders of which one or more coders are performing poorly (for any reason: stress, inattentiveness, personal issues, laziness, insufficient selection before starting to code, …), these bad coders could give you a misleading Kalpha. Then it is not the variable or training that is bad, but a coder. Discharging or retraining these coders might be an option if you are still in the pilot study phase. You could calculate Kalpha pairwise to find out these differences, but an easier way to get an indication is just to ask SPSS to calculate correlations[1] between the coders (who are variables anyway in your file).

[1] In this case, the variables are dichotomous, so Phi (in SPSS: Descriptive statistics – crosstabs – statistics) should be used to correlate the coders with each other. However, generally (and also in this case), very similar values emerge when you use one of the correlation coefficients in the "Correlate" menu in SPSS (e.g. Pearson),

If one particular coder is bad, this coder will stand out with generally lower correlation coefficients with the other coders (like in this case Albert). Dropping Albert from the reliability test, raises the Kalpha to 1.0 (maximum reliability).

KALPHA judges = Beatrix Elisabeth Margrethe Harald Gustav/level = 1/detail = 0/boot = 10000.

which can provide you an easily interpretable correlation matrix. Since we are just interested in an indication of which coder stands out, there is no problem doing it like that.

Testing inter-coder reliability: when does KALPHA return low and what does it mean?

Without going into formulas, some knowledge about KALPHA calculation can be really helpful. The element that makes KALPHA such a suitable test for inter-coder reliability, is the fact that it takes into account the **expected disagreement**, not only the **observed disagreement**.

$$\text{KALPHA} = 1 - \frac{\text{Observed disagreement}}{\text{Expected disagreement}}$$

The observed disagreement indicates the percentage of mismatches between coders in values attributed to the same units. This is not an appropriate measurement for inter-coder reliability, but you do find it back in plenty of published studies using content analysis. In the following example, coders Albert and Harald have coded the same five units. In 2 out of 6 cases, they disagree ➔ 33 percent disagreement and thus 66 percent coder agreement.

Unit	***Albert***	***Harald***
1	YES	NO
2	NO	YES
3	YES	YES
4	YES	YES
5	NO	NO
6	NO	NO

That is obviously not very impressive. Calculating KALPHA for this example leads to a KALPHA of .38, so this variable is clearly not reliable. But it still means that the coding was better than if it would have been done by chance. If you would flip coins six times to decide to code YES or NO for each case, you would in most cases end up with three rightly coded cases and three wrongly coded cases (since for this dichotomous variable there is a 50/50 chance each time that you have it right just by coincidence). If you would really try this out and enter these data in the SPSS file as an extra coder, KALPHA will tend to be close to .00. Thus, if you check reliability for a certain variable, and the test returns **a KALPHA of .00**, you might just as well have attributed random values. It is really bad news for your coding. It means your variable is really not well understood by your coders. But that is not the worst case scenario. **KALPHA can also return negative**, meaning that the coders are doing worse than coin flipping, indicating that at least some structural error exists. Typically, this is due to structural misunderstandings between coders. Asking the coders to explain how they code the variable, with the help of some concrete cases, will reveal these misunderstandings quickly.

We saw that dividing by the expected disagreement corrects the simple measure of coder (dis)agreement by the chance that coders code a unit rightly just by chance. This is calculated based on the amount of different values coded in the subsample for reliability testing (not the amount of categories in the codebook, but only those used in the data file of the reliability test !). If there are more different values, the chance to code rightly by random picking is smaller. To account for this expected disagreement, KALPHA takes into account the prevalence of the categories coded for the variable that is tested. So the rarity of categories will have an impact on KALPHA.

Looking at the following case, it becomes clear that **especially with binary variables for which one of the values (1 or 0) is very rare, KALPHA returns low even with very few mistakes of the coders**.

Dichotomous variables are very common in content analysis. For example, coders can be presented a battery of words or visuals they have cross off when present in a certain media outlet, e.g. the presence of background music in a television news item. Most news broadcasters only use background music very rarely, but it is of course crucial to know when they do. In the following example, 1 (meaning: background music present) is the rare value. Of the 30 news items coded by all coders, only Margrethe and Elisabeth have one mistake (first item). This corresponds with over 98 percent coder agreement! KALPHA, however, is only .59

	Beatrix	Albert	Elisabeth	Margrethe	Harald	Gustav
1	1,00	1,00	0,00	0,00	1,00	1,00
2	0,00	0,00	0,00	0,00	0,00	0,00
3	0,00	0,00	0,00	0,00	0,00	0,00
4	0,00	0,00	0,00	0,00	0,00	0,00
5	0,00	0,00	0,00	0,00	0,00	0,00
6	0,00	0,00	0,00	0,00	0,00	0,00
7	0,00	0,00	0,00	0,00	0,00	0,00
8	0,00	0,00	0,00	0,00	0,00	0,00
9	0,00	0,00	0,00	0,00	0,00	0,00
10	0,00	0,00	0,00	0,00	0,00	0,00
11	0,00	0,00	0,00	0,00	0,00	0,00
12	0,00	0,00	0,00	0,00	0,00	0,00
13	0,00	0,00	0,00	0,00	0,00	0,00
14	0,00	0,00	0,00	0,00	0,00	0,00
15	0,00	0,00	0,00	0,00	0,00	0,00
16	0,00	0,00	0,00	0,00	0,00	0,00
17	0,00	0,00	0,00	0,00	0,00	0,00
18	0,00	0,00	0,00	0,00	0,00	0,00
19	0,00	0,00	0,00	0,00	0,00	0,00
20	0,00	0,00	0,00	0,00	0,00	0,00
21	0,00	0,00	0,00	0,00	0,00	0,00
22	0,00	0,00	0,00	0,00	0,00	0,00
23	0,00	0,00	0,00	0,00	0,00	0,00
24	0,00	0,00	0,00	0,00	0,00	0,00
25	0,00	0,00	0,00	0,00	0,00	0,00
26	0,00	0,00	0,00	0,00	0,00	0,00
27	0,00	0,00	0,00	0,00	0,00	0,00
28	0,00	0,00	0,00	0,00	0,00	0,00
29	0,00	0,00	0,00	0,00	0,00	0,00
30	0,00	0,00	0,00	0,00	0,00	0,00
31						

```
Krippendorff's Alpha Reliability Estimate

             Alpha     LL95%CI     UL95%CI       Units    Observrs       Pairs
Nominal      ,5932      -,1442      1,0000     30,0000      6,0000   450,0000
```

The same amount of mistakes and thus level of agreement of 98% returns a much better KALPHA if the “1” category is not so rare as in the example. If you would change all the zeros in the last five cases into ones, KALPHA returns .94, which is highly reliable. The only difference between these two cases, is that in the second case, “1” is less rare.

This is in no way a plea to downgrade on the requirements for KALPHA. If you want to work with such a variable like background music, which is very rare and thus has a lot of zeros, you really do want to have the reassurance that IF the situation occurs that “1” should be coded, the coders also do so. So variables with rare categories like that, should get extra attention during coder training.

Examples of KALPHA calculation with ordinal, interval and ratio measurement levels

ORDINAL:

Consider the next example. The variable we are testing here is an ordinal variable (How large is the newspaper article?) with 4 categories: 1= Very large, 2= Large, 3= Medium, 4= Small. For the sake of the example, our coder team has coded this variable surprisingly well, and there are only eight occasions where a coder has coded something different than the other coders.

	Beatrix	Albert	Elisabeth	Margrethe	Harald	Gustav
1	1,00	1,00	1,00	1,00	1,00	1,00
2	1,00	1,00	1,00	1,00	1,00	1,00
3	2,00	2,00	2,00	2,00	2,00	2,00
4	3,00	3,00	3,00	3,00	3,00	3,00
5	3,00	3,00	3,00	3,00	3,00	3,00
6	1,00	1,00	1,00	1,00	1,00	1,00
7	3,00	3,00	3,00	3,00	3,00	3,00
8	3,00	3,00	3,00	3,00	3,00	3,00
9	3,00	3,00	3,00	3,00	3,00	3,00
10	1,00	1,00	1,00	1,00	1,00	4,00
11	1,00	1,00	1,00	4,00	1,00	1,00
12	2,00	2,00	2,00	2,00	2,00	2,00
13	2,00	2,00	2,00	2,00	2,00	2,00
14	2,00	2,00	2,00	2,00	2,00	2,00
15	3,00	3,00	3,00	3,00	3,00	3,00
16	4,00	4,00	4,00	4,00	4,00	1,00
17	3,00	3,00	3,00	3,00	3,00	3,00
18	3,00	3,00	3,00	3,00	3,00	3,00
19	3,00	3,00	3,00	3,00	3,00	3,00
20	1,00	1,00	4,00	1,00	1,00	1,00
21	2,00	2,00	2,00	2,00	2,00	2,00
22	2,00	2,00	2,00	2,00	2,00	2,00
23	1,00	4,00	1,00	1,00	1,00	1,00
24	3,00	3,00	3,00	3,00	3,00	3,00
25	4,00	4,00	1,00	4,00	4,00	4,00
26	4,00	1,00	4,00	4,00	4,00	4,00
27	4,00	4,00	4,00	1,00	4,00	4,00
28	2,00	2,00	2,00	2,00	2,00	2,00
29	2,00	2,00	2,00	2,00	2,00	2,00
30	2,00	2,00	2,00	2,00	2,00	2,00

Calculating KALPHA as we did earlier with lev=1 as if this variable was a nominal variable, returns KALPHA= .88, which sounds great. However, we know that this variable is not just nominal, there is also a rank order (from larger to smaller). The appropriate KALPHA command would then have lev=2.

KALPHA judges = Beatrix Albert Elisabeth Margrethe Harald Gustav/level = 2/detail = 0.

When this command is run, KALPHA= .61. This variable is not coded so well. When we look closer at the mistakes (marked in the data with circles), we can see why: the mistakes here are not just one step higher or lower, but always 4 (small) where other coders coded 1 (very large) or vice versa. These mistakes indicate that there is more going on than just some difficulties to differentiate between “large” and “very large”, and KALPHA picks up this problem.

If we, just for fun, take the same data and pick eight other ‘mistakes’, but then only between category 3 (medium) and 4 (small), the KALPHA (level: ordinal) returns .97 ! Thus, changing the level in the command into the right measurement level can also improve the KALPHA compared to treating the variable as nominal. Luckily, this latter case is more likely to occur !

	Beatrix	Albert	Elisabeth	Margrethe	Harald	Gustav
1	1,00	1,00	1,00	1,00	1,00	1,00
2	1,00	1,00	1,00	1,00	1,00	1,00
3	2,00	2,00	2,00	2,00	2,00	2,00
4	3,00	3,00	3,00	3,00	3,00	3,00
5	3,00	3,00	3,00	3,00	3,00	3,00
6	1,00	1,00	1,00	1,00	1,00	1,00
7	3,00	3,00	3,00	3,00	3,00	3,00
8	3,00	3,00	3,00	3,00	3,00	4,00
9	3,00	3,00	3,00	4,00	3,00	3,00
10	1,00	1,00	1,00	1,00	1,00	1,00
11	1,00	1,00	1,00	1,00	1,00	1,00
12	2,00	2,00	2,00	2,00	2,00	2,00
13	2,00	2,00	2,00	2,00	2,00	2,00
14	2,00	2,00	2,00	2,00	2,00	2,00
15	3,00	3,00	3,00	3,00	3,00	3,00
16	4,00	4,00	4,00	4,00	4,00	3,00
17	3,00	3,00	3,00	3,00	3,00	3,00
18	3,00	4,00	3,00	3,00	3,00	3,00
19	3,00	3,00	4,00	3,00	3,00	3,00
20	1,00	1,00	1,00	1,00	1,00	1,00
21	2,00	2,00	2,00	2,00	2,00	2,00
22	2,00	2,00	2,00	2,00	2,00	2,00
23	1,00	1,00	1,00	1,00	1,00	1,00
24	3,00	3,00	3,00	3,00	3,00	3,00
25	4,00	4,00	3,00	4,00	4,00	4,00
26	4,00	3,00	4,00	4,00	4,00	4,00
27	4,00	4,00	4,00	3,00	4,00	4,00
28	2,00	2,00	2,00	2,00	2,00	2,00
29	2,00	2,00	2,00	2,00	2,00	2,00
30	2,00	2,00	2,00	2,00	2,00	2,00

INTERVAL & RATIO

For interval and ratio variables, it is often more difficult for coders to come to the exact same results, especially when the coding includes non-computer-assisted measurements or counts. In the following example, the ratio variable represents the amount of casualties (dead people) the coders (visually) count in the war report news videos they are coding. Instructions include real-time visualization of the video (1 time) and manual counting of casualties visible in the video, according to some preset criteria (e.g. blood on clothes, position of the body etc.).

With a variable like this, and only short visualization, it would not be a surprise that coders would sometimes miss a person that should be counted or that they make judgment errors. However, the difference between counting 19 or 20 casualties in a report is not so large. As long as this variable is treated as a ratio variable, these kind of small disagreements should not be considered too problematic. In the example, it is clear that there are quite some differences between coders, usually small differences, becoming larger when the number of casualties increases (e.g. in the following example case 5 ranges from 12 casualties counted by Beatrix till 25 counted by Harald).

	Beatrix	Albert	Elisabeth	Margrethe	Harald	Gustav
1	0,00	.	0,00	0,00	0,00	.
2	0,00	.	2,00	0,00	0,00	.
3	4,00	4,00	3,00	4,00	4,00	3,00
4	6,00	6,00	6,00	6,00	7,00	8,00
5	12,00	15,00	20,00	18,00	25,00	15,00
6	0,00	0,00	0,00	0,00	0,00	0,00
7	0,00	0,00	0,00	0,00	0,00	0,00
8	3,00	3,00	3,00	3,00	3,00	3,00
9	1,00	1,00	1,00	1,00	1,00	1,00
10	1,00	2,00	.	2,00	5,00	.
11	5,00	4,00	.	5,00	5,00	.
12	7,00	4,00	7,00	7,00	7,00	7,00
13	11,00	8,00	11,00	11,00	9,00	13,00
14	2,00	2,00	1,00	2,00	2,00	2,00
15	3,00	3,00	3,00	4,00	3,00	3,00
16	3,00	4,00	6,00	4,00	5,00	4,00
17	1,00	1,00	1,00	1,00	1,00	1,00
18	1,00	1,00	1,00	2,00	0,00	1,00
19	0,00	0,00	0,00	0,00	0,00	0,00
20	0,00	0,00	0,00	0,00	0,00	0,00
21	0,00	0,00	0,00	0,00	0,00	0,00
22	0,00	0,00	0,00	0,00	0,00	0,00
23	1,00	1,00	1,00	1,00	0,00	0,00
24	0,00	0,00	0,00	0,00	0,00	0,00
25	6,00	4,00	6,00	5,00	6,00	5,00
26	8,00	8,00	8,00	8,00	8,00	7,00
27	2,00	12,00	2,00	11,00	8,00	1,00
28	0,00	2,00	0,00	0,00	0,00	1,00
29	0,00	0,00	0,00	0,00	0,00	0,00
30	19,00	17,00	18,00	19,00	10,00	18,00

Calculating KALPHA when treated as a nominal variable, this variable would not be reliable (KALPHA= .56). However, when treated (rightly) as a ratio variable, KALPHA increases till .85 . So, the reliability of this variable seems OK !

KALPHA judges = Beatrix Albert Elisabeth Margrethe Harald Gustav/level = 4/detail = 0.

```
Run MATRIX procedure:

Krippendorff's Alpha Reliability Estimate


           Alpha       Units     Obsrvrs       Pairs
Ratio      ,8513    30,0000      6,0000    414,0000

Judges used in these computations:
 Beatrix  Albert   Elisabet Margreth Harald   Gustav

Examine output for SPSS errors and do not interpret if any are found

------ END MATRIX -----
```

It is useful to point out that as soon as you would start treating this variable as a non-ratio variable in the analysis, e.g. by transforming it to a dichotomous variable distinguishing 0= news videos with 5 or less casualties and 1= news videos with more than five casualties), the nominal check is appropriate ! From the ratio reliability test, you do not know whether your cut-off point is reliable or not (which is quite essential if you make this dichotomy). That is why in this case, you should check reliability on the transformed dataset:

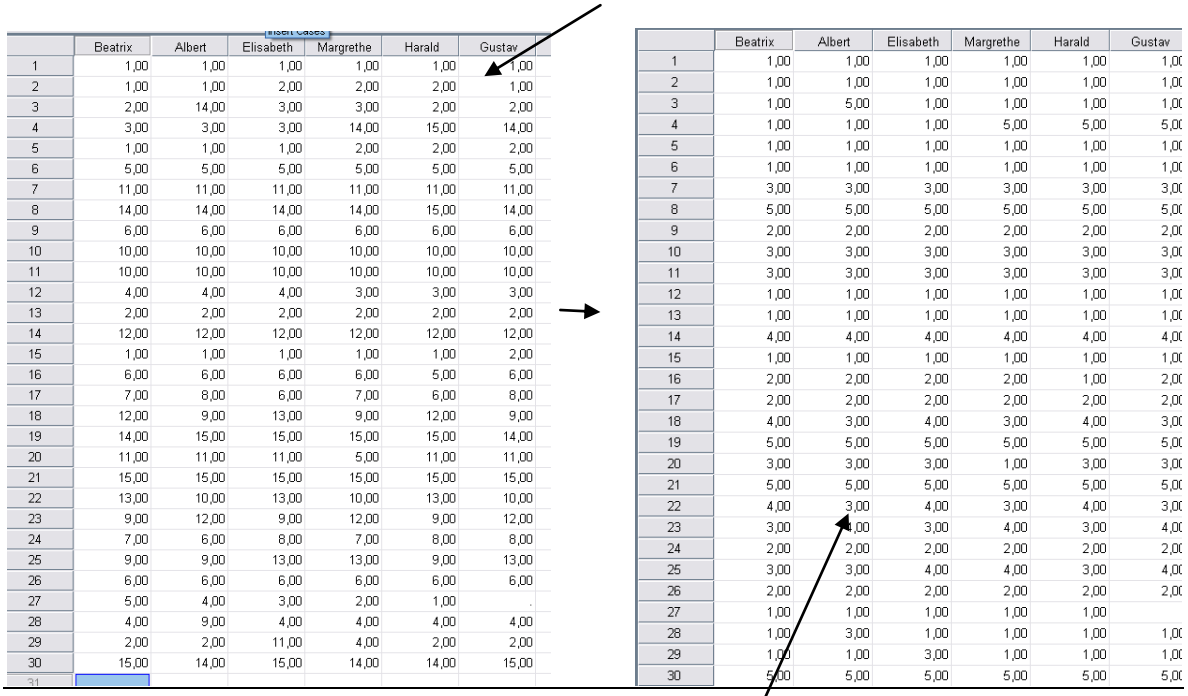
	Beatrix	Albert	Elisabeth	Margrethe	Harald	Gustav
1	0,00	1,00	0,00	0,00	0,00	1,00
2	0,00	1,00	0,00	0,00	0,00	1,00
3	0,00	0,00	0,00	0,00	0,00	0,00
4	0,00	0,00	0,00	0,00	1,00	1,00
5	1,00	1,00	1,00	1,00	1,00	1,00
6	0,00	0,00	0,00	0,00	0,00	0,00
7	0,00	0,00	0,00	0,00	0,00	0,00
8	0,00	0,00	0,00	0,00	0,00	0,00
9	0,00	0,00	0,00	0,00	0,00	0,00
10	0,00	0,00	1,00	0,00	0,00	1,00
11	0,00	0,00	1,00	0,00	0,00	1,00
12	1,00	0,00	1,00	1,00	1,00	1,00
13	1,00	1,00	1,00	1,00	1,00	1,00
14	0,00	0,00	0,00	0,00	0,00	0,00
15	0,00	0,00	0,00	0,00	0,00	0,00
16	0,00	0,00	0,00	0,00	0,00	0,00
17	0,00	0,00	0,00	0,00	0,00	0,00
18	0,00	0,00	0,00	0,00	0,00	0,00
19	0,00	0,00	0,00	0,00	0,00	0,00
20	0,00	0,00	0,00	0,00	0,00	0,00
21	0,00	0,00	0,00	0,00	0,00	0,00
22	0,00	0,00	0,00	0,00	0,00	0,00
23	0,00	0,00	0,00	0,00	0,00	0,00
24	0,00	0,00	0,00	0,00	0,00	0,00
25	0,00	0,00	0,00	0,00	0,00	0,00
26	1,00	1,00	1,00	1,00	1,00	1,00
27	0,00	1,00	0,00	1,00	1,00	0,00
28	0,00	0,00	0,00	0,00	0,00	0,00
29	0,00	0,00	0,00	0,00	0,00	0,00
30	1,00	1,00	1,00	1,00	1,00	1,00
31						

KALPHA judges = Beatrix Albert Elisabeth Margrethe Harald Gustav/level = 1/detail = 0.

The result of this test is: KALPHA=.67, which is a clearly more questionable reliability.

EXTRA: COLLAPSING CATEGORIES

It happens frequently that researchers conduct content analysis with variables containing considerably more (detailed) categories than strictly necessary for the aimed analysis. Often, this is found back in research reports stating that some of these detailed categories have been collapsed into larger and broader categories. A typical example of this is thematic coding: usually, there is a large codebook with many detailed topic categories, but the actual analysis in the end only takes into account broader categories, based on an aggregation of smaller categories. In the following example of a thematic coding, the original codebook listed 15 different topic codes for the coders.



	Beatrix	Albert	Elisabeth	Margrethe	Harald	Gustav
1	1,00	1,00	1,00	1,00	1,00	1,00
2	1,00	1,00	2,00	2,00	2,00	1,00
3	2,00	14,00	3,00	3,00	2,00	2,00
4	3,00	3,00	3,00	14,00	15,00	14,00
5	1,00	1,00	1,00	2,00	2,00	2,00
6	5,00	5,00	5,00	5,00	5,00	5,00
7	11,00	11,00	11,00	11,00	11,00	11,00
8	14,00	14,00	14,00	14,00	15,00	14,00
9	6,00	6,00	6,00	6,00	6,00	6,00
10	10,00	10,00	10,00	10,00	10,00	10,00
11	10,00	10,00	10,00	10,00	10,00	10,00
12	4,00	4,00	4,00	3,00	3,00	3,00
13	2,00	2,00	2,00	2,00	2,00	2,00
14	12,00	12,00	12,00	12,00	12,00	12,00
15	1,00	1,00	1,00	1,00	1,00	2,00
16	6,00	6,00	6,00	6,00	5,00	6,00
17	7,00	8,00	6,00	7,00	6,00	8,00
18	12,00	9,00	13,00	9,00	12,00	9,00
19	14,00	15,00	15,00	15,00	15,00	14,00
20	11,00	11,00	11,00	5,00	11,00	11,00
21	15,00	15,00	15,00	15,00	15,00	15,00
22	13,00	10,00	13,00	10,00	13,00	10,00
23	9,00	12,00	9,00	12,00	9,00	12,00
24	7,00	6,00	8,00	7,00	8,00	8,00
25	9,00	9,00	13,00	13,00	9,00	13,00
26	6,00	6,00	6,00	6,00	6,00	6,00
27	5,00	4,00	3,00	2,00	1,00	.
28	4,00	9,00	4,00	4,00	4,00	4,00
29	2,00	2,00	11,00	4,00	2,00	2,00
30	15,00	14,00	15,00	14,00	14,00	15,00

	Beatrix	Albert	Elisabeth	Margrethe	Harald	Gustav
1	1,00	1,00	1,00	1,00	1,00	1,00
2	1,00	1,00	1,00	1,00	1,00	1,00
3	1,00	5,00	1,00	1,00	1,00	1,00
4	1,00	1,00	1,00	5,00	5,00	5,00
5	1,00	1,00	1,00	1,00	1,00	1,00
6	1,00	1,00	1,00	1,00	1,00	1,00
7	3,00	3,00	3,00	3,00	3,00	3,00
8	5,00	5,00	5,00	5,00	5,00	5,00
9	2,00	2,00	2,00	2,00	2,00	2,00
10	3,00	3,00	3,00	3,00	3,00	3,00
11	3,00	3,00	3,00	3,00	3,00	3,00
12	1,00	1,00	1,00	1,00	1,00	1,00
13	1,00	1,00	1,00	1,00	1,00	1,00
14	4,00	4,00	4,00	4,00	4,00	4,00
15	1,00	1,00	1,00	1,00	1,00	1,00
16	2,00	2,00	2,00	2,00	1,00	2,00
17	2,00	2,00	2,00	2,00	2,00	2,00
18	4,00	3,00	4,00	3,00	4,00	3,00
19	5,00	5,00	5,00	5,00	5,00	5,00
20	3,00	3,00	3,00	1,00	3,00	3,00
21	5,00	5,00	5,00	5,00	5,00	5,00
22	4,00	3,00	4,00	3,00	4,00	3,00
23	3,00	4,00	3,00	4,00	3,00	4,00
24	2,00	2,00	2,00	2,00	2,00	2,00
25	3,00	3,00	4,00	4,00	3,00	4,00
26	2,00	2,00	2,00	2,00	2,00	2,00
27	1,00	1,00	1,00	1,00	1,00	
28	1,00	3,00	1,00	1,00	1,00	1,00
29	1,00	1,00	3,00	1,00	1,00	1,00
30	5,00	5,00	5,00	5,00	5,00	5,00

KALPHA for this variable (nominal, 15 categories) =.58, which is not sufficiently reliable. Sometimes a researcher did not plan to use the detailed categories (but then it would have been best to have less topic categories), and that is of course a good reason to collapse categories in broader categories, but this could also be done to try to "save" the data after the variable was found unreliable. In this example, the 15 topic categories were transformed to 5:

1-5 internal politics

6-8 social and economic topics

9-11 Human interest (e.g. 9 was sports, 10 was fashion and 11 was celebrity news)

12-13 Accidents and disasters

14-15 International Relations

When calculating KALPHA for this new situation (still nominal of course), it returns a quite satisfactory reliability (KALPHA= .79). Basically, this means that coders had quite some disagreements between the categories within a larger category, but not so much between issue coders belonging to different broader categories. E.g. coding 10 (fashion) instead of 11 (celebrity news), but not coding 12 (natural disasters) instead of celebrity news. If collapsed even more into for example Hard News (1, 2, 5) and Soft News (3,4) KALPHA increases to .92.

Note, however, that collapsing can also lead to a decrease of KALPHA. This is especially the case when a not so well-coded category is isolated from the rest of the topics. In this example, let's consider the situation where we collapse the topic codes into a dichotomy. 1= Accidents and disasters (detailed topics 12 and 13); 0 = All the other topics. So that is kind of an analysis of sensationalism. KALPHA, however, is only .56, and so we must not work with this specific distinction. The reason is not only that the specific topic codes 12 and 13 are coded poorly (= often mixed up with the other codes), but also that this broad category of sensationalism is pretty rare. As we saw earlier, that can easily lead to poor reliability if some disagreements between occur.

So do not just collapse categories and expect KALPHA to improve !

Summary:

- For calculating inter-coder reliability, use Krippendorff's Alpha
- For calculating Krippendorff's Alpha, use SPSS and Hayes' Macro. Always run the macro again when you restart SPSS.
- Make a separate file for each variable. Amount of coders or missing values do not matter.
- Make sure the coders are inserted as variables, and coded values as cases
- Make sure you are working with variables, not a constant value (at least two different categories must be coded to be able to calculate KALPHA)
- Make sure you include the right measurement level for your variable in the KALPHA command line (1=nominal, 2=ordinal, 3=interval, 4= ratio). This is important, because it can make huge difference.
- Watch out for rare categories
- You can collapse categories, but it does not automatically improve KALPHA (however, it usually does).
- Calculate KALPHA for the variables you are actually using in your analysis, at the measurement level and in the form you are using them. Report these KALPHA values in your research report (not a mean of all KALPHA's of all variables you tested).
- Always give some background for KALPHA's lower than .80.
- Sometimes, a KALPHA of .80 should not satisfy you, more specifically for variables that are pretty straightforward and easy to code.
- Low reliability = need to change something about description of the variable, content of the variable, categories, explanation in the codebook, training, coding procedure…

References

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

Krippendorff, K. (2004). *Content Analysis: An Introduction to Its Methodology. Second Edition*. Thousand Oaks, CA: Sage.

Krippendorff, K. (2011). "Computing Krippendorff's Alpha-Reliability."

Riffe, D., Lacy, S., Fico, F. (2005) *Analyzing Media Messages. Using Quantitative Content Analysis in Research, second edition.* New Jersey: Mahwah, pp138-155. 242p.