Assessing Inflation Variance Estimates Due to Coding Error
using a Coder Reliability Study
Patrick Sturgis

I want to thank Skip and Jon for inviting me here. I spent a summer here in Ann Arbor. It is very nice to be here in the winter. It is not so nice to see the dollar being so unfavorable towards the pound though!

To situate myself within this group, I am very much a standard survey person and that is what my presentation really reflects. I will not be presenting any radically new ideas. In fact, the things that I am going to be talking about are rather ancient. These are ideas that go back to the '50's and '60's – classic survey methodology. I owe a debt of gratitude to Graham Kalton and Richard Collins for what they did on this in the '70's. So, the idea is to think about how we can quantify all these problems that we have been hearing about – coders and subjectivity and different codes being applied – how can we categorise and quantify those errors? How can we assess the damage?

In particular, how Skip framed this issue this morning, looking at this margin of error. That is one of the things that this presentation is really about – trying to quantify the impact of coder unreliability and variability of survey estimates. I am very much a PowerPoint person I'm afraid; I almost need one to speak to my wife in the evening! I will start by talking a little bit about what the motivation is here. I Then I am going to classify the different kinds of error that we get from coders applying their own cognitive outlook on the verbatims that are recorded and how we can classify those different areas. And then the different implications of these different kinds of errors – individually and taken together – the error in the survey instruments. We do that in particular by doing these coder reliability studies, which are actually much more talked about than done. You rarely see these reported in the survey methodological literature despite the findings showing how important they are. So I will show how we can do these kinds of coder reliability studies and what information we can get from them using an example from some work I did for the UK Office for National Statistics on the Time Use survey. It is not a politically-focused survey but there are lots of open questions to be coded to a very complex frame so the issues are very much the same. At the end I will draw some conclusions and come up with some thoughts about what Jon and Skip might think about doing.

This is also a very salient conference for me because I have a survey that is going out in the field in January. This survey is funded by the Wellcome Trust and it is about public knowledge, attitude and engagement in biomedical science and research. In this survey the funders have been very keen to have lots and lots of these open type responses. Science studies as Jon Miller will, I'm sure attest, contains lots of people who are not convinced of the 'positivist' survey method, so they are very keen on not 'framing the issue' by using pre-coded questions. We have been having discussions about how things are going to go. I have been saying – how are you actually going to do this? What is it going to look like when we get it back? They say 'we will get back to you on that'. They haven't gotten back to me yet. I am getting worried – what we are hearing this morning is making me even more worried!

This is a classic kind of open questions – please tell me in your own words, what comes to mind when you think about the term "medical research?" Now why would we want to ask a question like this? There are a lot of different reasons. Of course, the most obvious one is that the researcher can't be bothered to sit down to think about all the possible response alternatives. So, it is can be a sort of a get-out for a lazy researcher. Which is why, when we give students questionnaire design assignments, often a lot of the questions come back as open ones because they haven't sat down to think through what the possible range of answers might be.

But I guess the less cynical view is that we are going to get an increased range of responses. We can't, as we've heard this morning, necessarily think in advance what all the different viewpoints about medical research are actually going to be. We might have some 'out there' views, we might have some sensible views that we didn't think about because we are a particular kind of person. Another reason – and these are not mutually exclusive of course – is where we have a very large range of potential responses. So it wouldn't be feasible for the interviewer to present all 356 occupational codes to the respondent. It is just not feasible to use a fixed response type question. Another reason, again this is not mutually exclusive to the first two, but we can – in surveys we can often be criticized if we are finding this out about the population – we are finding out what we framed to be the problem in the first place. So we can get around this by saying that we have thrown in a few open-ended questions – we didn't tell people what to think, this is what they actually told us, un-prompted.

On the downside of course, if you are an analyst of surveys you still need to analyze these things responses using SPSS or SAS or whatever. You therefore need to turn them into fixed categories for analysis purposes. And, how often do analysts actually use the data that we get from open-coded responses in a non-quantitative manner? Often times they end up coding it down to something that is manageable and quantitative anyway, so the benefit of this approach is not always clear. In addition, this process, as we are well aware is error-prone and, of course, extremely expensive. So, those are the general pros and cons of open survey questions.

Now focusing a little bit more on this coder error and what it is. I think some of the talks that we had this morning focused more on the notion of bias when thinking about coder error. Jim and Randy were thinking about applying the 'wrong' code – this person has got it correct so we underestimate knowledge or we overestimate mobility in the labor market. Coders are applying the wrong code, so we are going to get a bias. But actually this is only one part of the story. For lots of things we are trying to estimate using these kinds of open questions – it is quite difficult to conceptualize a bias in the context of an attitude for example. What is the 'correct' code to apply to a respondent who says "I think about Doctor Who" – what is the attitude that is really in that person's mind? And who is the person who has the appropriate insight to that – is not an easy question to answer. We can do various things. We can get someone who has a Ph.D. and say they have a better view than someone who has a masters and so on down the educational rankings. We can compare to panels of experts. We can even go back to verify things with the respondent – this is what we coded for you – is this okay? It is rarely done but one could do this. The general problem is that bias is a hard thing to conceptualize and very difficult to estimate as well.

Now, I want to think a little bit about why we get differences in codes and, already today, we have seen a number of these things come up. Coders not being well motivated or not well trained or not well supervised. Maybe the environment in which they are working is not suitable. Maybe if they are coding at home there are kids running around the house. There are frame problems that we have seen – frames with gaps in them and ambiguous codes or too many codes, non-redundant and vague and so on. Often times there are technical problems – if you have an online coding facility you might not be able to see things properly on your monitor. These are things we are thinking about how to improve coding – these are the obvious places to look to reduce the kinds of errors we tend to see.

So what are the different components of coder error? In a coder reliability study we are generally assuming that we have a random sample of the population of possible coders and each coder has a random sample of verbatims. These are a couple of foundational assumptions. Given that, if coders apply different codes to the same coded response then we can think of these as being errors. The kind of error is going to be either an 'uncorrelated' (or 'simple') coder error, or a 'correlated' coder error. That is an important distinction. What does it mean? An uncorrelated or a simple coder error is that – coders apply different codes to the same text but this tendency is not systematically associated with particular coders. So it is random across coders – whether someone picks one different code or another different code is not related to particular coders. We can think of this as a measure of the reliability, which I will refer to as R, of a particular code. Now statically there are different ways we can measure this reliability coefficient. A standard way is to use P, the proportion of agreement. If we have, say, 5 coders, then P is the proportion of all pair-wise comparisons that apply the same code to the same response. This is essentially what Randy was showing earlier, looking at the proportion of pair-wise agreements across coders. There are other things that we can use to measure this if we don't have many codes that we might get chance agreement – so you might use something like Kappa, which makes a correction for chance agreement. When we have large numbers of codes, like we do in this example, chance agreement is rare and Kappa and P are essentially equivalent.

Now, this kind of reliability it is not directly estimable from survey data. There is nothing in there that enables us to say that this code has a reliability of .7 in the data. So we have to do a special study. And, we have to do it separate from the main data collection exercise to get an estimate of P. On the other hand, conventional variance estimators are unbiased in the presence of this kind of uncorrelated coder error. That is to say, it is incorporated in the standard error estimate. Variance will be larger if we have an unreliable code. The precision of the estimate will be reduced by this factor of 1-P. If we have a reliability of a code which is .7 then our variance is 30% larger than it would be if we had a perfectly reliable code. We can apply that same logic to costs and effective sample size. Our costs are 30% higher than they would be if we had a perfectly reliable code.

Turning now to correlated coder error - coders applying different codes to the same answer and the tendency to select a particular code is systematically higher to a particular coder or coders. You can imagine how this might happen – someone is not well trained, they are busy, they aren't motivated, they are looking through this hugely large code list and they keep coming to the same favorite one – you keep going until you find one that will do. You pick it and that becomes your favorite code. This is a type of error which is a within person or within coder error. If you want to think conceptually about this – if you know much about surveys – it is essentially equivalent to an interviewer design effects. Interviewers doing the same thing, they don't do it deliberately but they push people towards certain answers or they have a certain way of speaking to respondents. Now, we can measure this correlated coder error using an intra-class correlation coefficient which we will refer to as rho. Rho has not been investigated terribly often but the examples we heard about this morning, generally we have one coder doing this job, so we can't estimate this unless we have more than one coder. Actually you need a reasonably large number of coders if we want to make some kind of random effect assumption about this intra-class correlation then we need significantly more than two.

Where this correlated coder has been investigated in the past it has proven to be quite small. We get values close to zero, rarely above 3. So correlated coder error is not a substantial part of the total variance. The problem is, as we will see in the next slide, that the effects of correlated coder error on the variance is not just a function of rho, it is function of the average size of the coders' workload (as with interviewer design effects) and the reliability of the code.

Here is the formula for the inflation of the variance as a function of coder error taken it from the Biemer & Trewin piece 1997. Here we have Pc=rho; m=average workload; Pi=code reliability. If we just make up a few values for this – if we want to figure out what is the inflation in the variance of a particular estimate from a survey. We have a value of rho of 2% here. If a coder codes 1000 question and the code has a reliability of .75 which we said is not bad this morning – the variance would be inflated by almost 15 times. We can take the square root of that to find the standard error, that would be more intuitive, but that is still a big number and quite scary.

So, to estimate this kind of coder-related variance inflation we need an estimate of rho, the reliability of the code and the average workload of coders. We can get an estimate of rho direct from the sample data as long as we make an assumption that each coder is assigned a random sample of all the questionnaires. We can get an estimate of the reliability of the codes by doing a special study I am going to talk about in a minute. We know what the average coder workload is on the study.

I am going to focus my remaining 10 minutes on this Time Use Study where we did one of these coder reliability studies. The UK2000 Time Use Survey is a standard design where we interviewed all those aged 8 and above in the household. I guess that isn't too standard! Respondents filled in these diaries in their own words. They would write in what they were doing for every 10 minutes of the day. We code their activities into 3 levels getting increasingly more detailed as we go down the frame. If someone wrote in – took the dog for a walk – we code that as household and family care, within that we would say that is specifically gardening and pet care, and then walking the dog. That is what we are actually talking about here. This shows the diary where people write in what they are doing every 10 minutes. The coder then has to code each of these 10 minute slots to that frame.

In this study we had 5 coders. They each coded the same 40 diaries. Our unit of analysis is 10 minute time slots – what the respondent is doing every 10 minutes of the day. There are 144 of these in any given day. So across coders, days and diaries we have 28,800 observations that we are coded by these 5 coders.

Let's go straight to looking at the reliabilities of the code frame. Overall we find at the 3-digit level the proportion of agreement was 89%. You can see that for each coder the reliabilities are about the same. Coder 5 is slightly less in agreement than the other coders. Disagreement increases as you go down the levels of the coding frame – you might disagree about 'walking the dog', but when you go up the frame it is still household and family care. The unreliability reduces as you go up the coding frame. But even at the most detailed level, we have a very high proportion of agreement.

Just to put this in some context – if you have 5 coders and 4 of them agree and one of them disagrees that is still 60% proportion of agreement – 4 in agreement and one in disagreement. So, these numbers here, the averages across the whole code frame are very high. But this actually masks a lot of variability across the individual codes. Personal care and employment – very reliable; that is because that is sleeping so people drew a great big line through sleep and you don't disagree. Down here we have social life – there is a lot of code unreliability there.

Now, here we have the intra-class correlations for each of these codes at the 1-digit level. This is the 95% confidence interval. Only 2 of these are not significant, so we truncated them to zero so they don't go to negative. Employment and voluntary work were the only codes for which we didn't find significant correlated coder error. These are the reliabilities of the codes. The main thing to look at is this final column; this is the increase in the standard error in these estimates as a function of coder error. You can see that some of them are quite low but some of them are really big – hobbies & games – they should be 3 ½ times larger than we would estimate them using standard variance estimators.

As has also been pointed out already today – the respondents are writing things in here themselves with these diaries. But actually, a lot of the time with CAPI surveys it is even worse because we have error creeping in between what the respondent says and what the interviewer actually records. I haven't even looked at that in this study. That should make us even more worried.

Here is an illustration. I just got this data back a couple of weeks ago. We followed up people who had given us middle alternatives – we asked them to say in their own words why they said they neither agreed nor disagreed with the statement. This is what I got back from the data agency [points to slide of very scant, error-strewn list of responses]. The point is that this isn't what people said. There is going to be huge amount of disagreement across interviewers in terms of winding things down. This is another Pandora's Box.

So what do we conclude from all this? We know that conventional variance estimators (heroically) underestimate variance of coded verbatims. One of the reasons for this is that we don't use enough coders. As with interviewers, we wouldn't think it was a sensible sample design to have a few interviewers go into the field with very large assignments. So, one thing we can think about is that we should have more coders. That would reduce the variance inflation factor by – with this we doubled it – we would reduce inflation factor by one third in this Time Use Survey example.

Another thing that we have to think about is quality monitoring – we could make the codes more reliable by doing some of these studies, identifying coders who are out there and have lower proportions of agreement. We need to think about that. This is not a catch all solution. There are costs associated with hiring and training and getting decent coders. You have to think carefully about that. We can't just keep increasing this number because it has its own limitations. I've mentioned this point about transcription error.

We need to have much more continuous monitoring, identifying weaknesses in the frame as we go along. Problems with particular codes we can identify with this type of study and problems with particular coders. We can do better. We should implement these kinds of approaches as standard. It is not that big a job. It is not that expensive when you see the kinds of errors and you can't find these kinds of errors unless you do this kind of study.

Something we have heard a lot about this morning is having greater transparency about how coding is done. I haven't thought too much about – we've only looked at one kind of estimate of proportion – but if different analysts wanted to use this data we should try to help them to get this unreliability into whatever estimates they are making. A starting point would be to report on reliabilities and correlated errors.

Another thing is that if we are thinking about designing surveys – Jon & Skip have got to – who wants to be the person that after 50-60 years says 'I'm going to be the one to change it'. I wouldn't want to be in that position. But you have to think very hard about the cost and benefits of asking questions in this way. I think there are some potential benefits but I think sometimes we do it automatically and we aren't always making optimal decisions.

The last thing I would advocate are some experimental approaches. I am very keen on split ballot type experiments. I think it would be very nice to make some of these comparisons empirically about what are the analytical gains of asking questions in this open way and coding it and all the expense and errors and so on relative to having a question where we think we can do some pilot research and we get a good list of pre-coded alternatives with an 'other please specify' in there. Half the sample gets one and half the sample gets the other and we find something totally different? I think there is a lot we can do to evaluate what we gain from doing this kind of question in the first place. That is it.