Measurement in the Social Sciences (TT 2009) Appendix 1.1: Reliability Extended Examples in Stata

Dave Armstrong
University of Oxford
Department of Politics and International Relations

April 30, 2009

1 Good Uni-dimensional Scale

Table 1: Variables used in the analysis (nes04.dta)

spendserv	Spending and Services -7-point scale self-placement
defspend	Defense spending - 7-point scale self-placement
insurance	Govt/private medical insurance scale: self-placement
jobsliv	Job and Good Standard of Living -scale self-placement
gablacks	Government assistance to blacks-7 point scale self-placement
envjobs	Environment vs. jobs tradeoff scale - self-placement
fedgun	Should fed govt make more difficult to buy gun - self-placement
womrole	Women's role - 7-point self-placement

The first step in the process is to take a look at the correlation matrix of these variables. We can get this in Stata by issuing the following command:

```
. pwcorr spendserv defspend insurance jobsliv gablacks envjobs fedgun womrole

             | spends~v defspend insura~e  jobsliv gablacks  envjobs   fedgun  womrole
-------------+------------------------------------------------------------------------
   spendserv |   1.0000
    defspend |   0.1639   1.0000
   insurance |   0.4220   0.2575   1.0000
     jobsliv |   0.4052   0.2454   0.4969   1.0000
    gablacks |   0.2944   0.2494   0.2870   0.5097   1.0000
     envjobs |   0.2093   0.2029   0.2250   0.2059   0.1831   1.0000
      fedgun |   0.1992   0.1851   0.1710   0.1895   0.2057   0.0995   1.0000
     womrole |   0.1344   0.1120   0.1606   0.1249   0.1331   0.2028   0.1477   1.0000
```

You can see that the correlations generally range from .15 to .5 in absolute value. The only variable that looks like it may not belong is women's role as it has a number of relatively small correlations. This is something to keep in mind as we continue, but it is not evidence to exclude this variable now. Another thing to notice is that the spending and services seems to be negatively correlated with everything else, so we will want to look at the question wording and then if the question is worded in the opposite direction, we will turn reverse this item in the scale.

Now, we can perform the reliability anaylsis:

```
. alpha spendserv defspend insurance jobsliv gablacks envjobs fedgun womrole,
    item std casewise

Test scale = mean(standardized items)

                                                            average
                              item-test     item-rest     inter-item
Item         |  Obs  Sign   correlation   correlation     correlation     alpha
-------------+-----------------------------------------------------------------
spendserv    |  749    -       0.6615        0.5157          0.2463      0.6959
defspend     |  749    +       0.5513        0.3785          0.2713      0.7227
insurance    |  749    +       0.6857        0.5469          0.2409      0.6896
jobsliv      |  749    +       0.7059        0.5735          0.2363      0.6841
gablacks     |  749    +       0.6403        0.4886          0.2511      0.7013
envjobs      |  749    +       0.5267        0.3490          0.2769      0.7283
fedgun       |  749    +       0.5129        0.3326          0.2800      0.7314
womrole      |  749    +       0.4739        0.2869          0.2889      0.7398
-------------+-----------------------------------------------------------------
Test scale   |                                               0.2615      0.7391
-------------------------------------------------------------------------------
```

Notice that α is relatively high (>.7). Here, we are asked to make a judgement call about whether to remove an item to get a very little bit higher reliability. Alpha (in Stata) doesn't come with a significance test or confidence interval, but you could create one with a technique called bootstrapping.[1] You can do this in Stata by issuing the following command:

```
Bootstrap results                               Number of obs      =       792
                                                Replications       =      1000

      command:  alpha spendserv insurance jobsliv gablacks envjobs fedgun
                womrole, item std casewise
            a:  r(alpha)

------------------------------------------------------------------------------
             |   Observed   Bootstrap                         Normal-based
             |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
           a |   .7108918   .0164999    43.08   0.000     .6785526    .7432309
------------------------------------------------------------------------------
```

This gives a 95% confidence interval for α of [0.68, 0.74]. So, we can use this information to evaluate the results from the `alpha` command. Since we could not get a *significantly* better α by dropping a variable, I suggest that we leave all of them in for right now. Also, before we carry on, let's generate standardized versions of all of these variables (mean=0 and sd=1) so when we do things outside the alpha command we're getting comparable results.

Now, it is probably worth trying to address the monotone homogeneity assumption. For this, like I said before you can plot each variable in turn against its rest score. This is pretty easy, if a bit tedious, in Stata. The graph that I'm talking about is called a "Lowess" graph, which stands for <u>Lo</u>cally <u>We</u>ighted <u>S</u>catterplot <u>S</u>moother. Essentially, this technique fits a smooth line to a scatterplot. There is no assumption that the line has to take any particular form, so non-monotonicity is a possibility. If we see this in the graphs, that is evidence that the included item does not belong in the scale. The syntax would look something like:

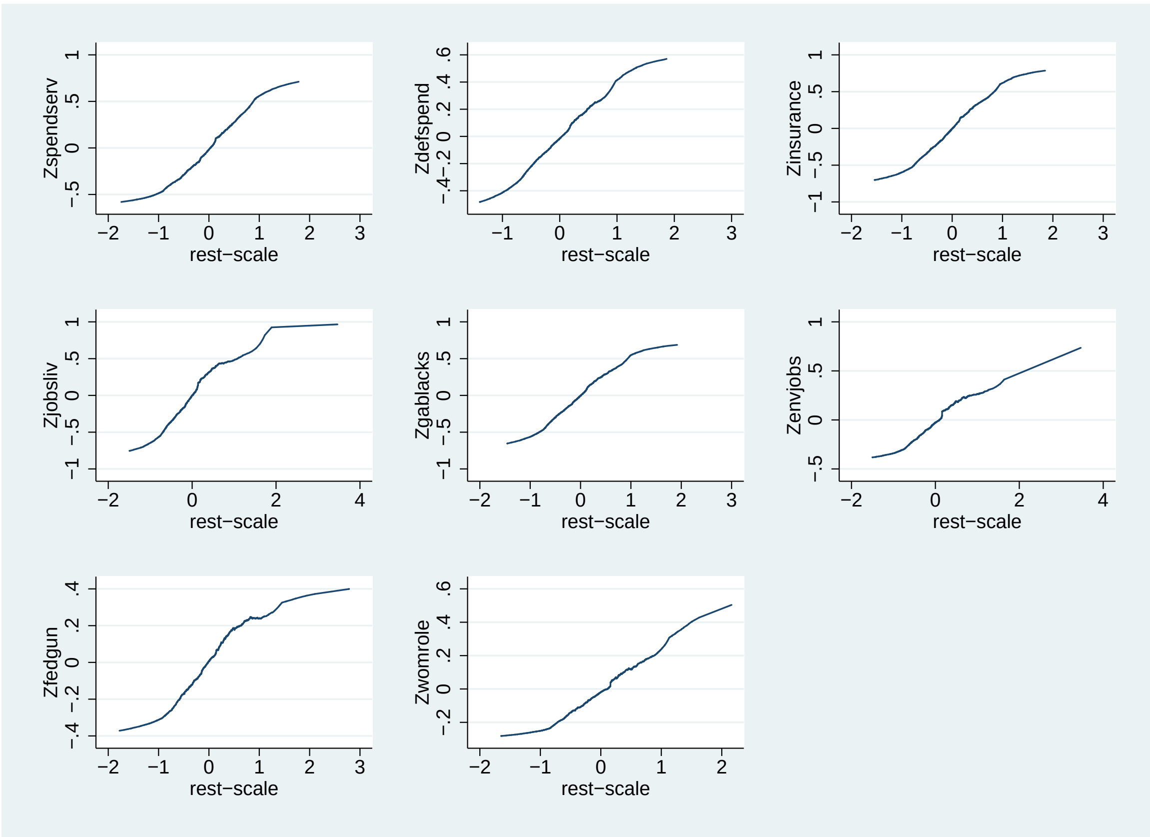
[1] Bootstrapping draws a sample *with replacement* of length n (where n is the number of observations in the dataset) and computes a sample statistic for each of what is usually between 1000 and 2500 iterations. So, Stata draws a sample of 749 with replacement from the valid cases, then calculates α, then draws a new sample with replacement and does this again. The result is an empirical confidence interval for the sample statistic.

```
egen rest1 = rowmean(Zdefspend Zinsurance Zjobsliv Zgablacks Zenvjobs Zfedgun Zwomrole)
twoway (lowess Zspendserv rest1)
egen rest2 = rowmean(Zspendserv Zinsurance Zjobsliv Zgablacks Zenvjobs Zfedgun Zwomrole)
twoway (lowess Zdefspend rest2)
...
egen rest8 = rowmean(Zspendserv Zdefspend Zinsurance Zjobsliv Zgablacks Zenvjobs Zfedgun)
twoway (lowess Zwomrole rest8)
```

I wrote a program that automates this for you and puts all the graphs on one page called "restplot" which takes only a variable list. The syntax and output are below:

```
restplot Zspendserv Zdefspend Zinsurance Zjobsliv Zgablacks Zenvjobs Zfedgun Zwomrole
```

Figure 1: Output from `restplot`



What you can see here is that for the most part, the plots are all monotonic with respect to the rest-scale. I would probably leave these variables in, however you could reasonably justify doing otherwise.

If we're confident that we have the "right" model, that is that there is evidence in support of the assumptions of the model, then we are left with the task of interpretation.

One thing that is simultaneously good and bad about scaling solutions is that they do not interpret themselves. Once we develop a scale, it is up to the researcher to assign it a name and interpretation. This scale should work as a reasonable measure of political ideology in the US in 2004. Given that the ANES asks a question about liberal-conservative self-placement on a 7-point scale, this new measure should correlate relatively highly with the liberal-conservative variable. If we take liberal-conservative placement as the "true" dimension (which it is not), we should see higher correlations between the liberal-conservative placement and the scale than any of the individual variables.

Table 2: Correlations of the scale and individual items with liberal-conservative self-placement

Variable	Liberal-Conservative
Scale	0.640
Spending and services	−0.443
Defense spending	0.382
Private vs. govt health insurance	0.455
Jobs vs. standard of living	0.451
Govt aid to blacks	0.384
Environment vs. jobs	0.340
Easier to buy a gun	0.334
Woman's role	0.327

We will never know the true dimension, but it is interesting that the scale correlates higher with liberal-conservative self-placement than any of the other variables that might be used to measure the same concept. This is strictly a pedagogical exercise. It is nice that the scale correlates with ideological self-placement, but we should not suggest that it represents the true dimension.

1.1 Generating the Variable

Once you have the scale you want, you'll also want to generate the scale variable that is an estimate of the underlying dimension. To do this, you can use the `gen()` option of the `alpha` command. Now, Stata does *not* casewise delete by default here, it generates an estimate for every observation that has a response on at least 1 variable and then divides the sum of the observations on all non-missing variables for observation i by the number of non-missing items for observation i. If this makes sense, you can use the gen command without a problem.

```
alpha spendserv defspend insurance jobsliv gablacks envjobs fedgun
   womrole, item std gen(scale)
```

However, you might only want estimates for observations where there are more than a certain number of non-missing items. You can see how many non-missing items there are with the following command:

```
egen nonmiss = rownonmiss(spendserv defspend insurance jobsliv gablacks
   envjobs fedgun womrole)
```

Then, you can see how many observations had each possible number of non-missing items by issuing the following command:

```
. tab nonmiss

    nonmiss |      Freq.     Percent        Cum.
------------+-----------------------------------
          0 |          1        0.08        0.08
          1 |          5        0.41        0.50
          2 |         11        0.91        1.40
          3 |          8        0.66        2.06
          4 |         33        2.72        4.79
          5 |         58        4.79        9.57
          6 |        107        8.83       18.40
          7 |        240       19.80       38.20
          8 |        749       61.80      100.00
------------+-----------------------------------
      Total |      1,212      100.00
```

You can see, only about 60% of the observations had no missing values. We might want to use only those observations that had less than or equal to half of the items with valid responses. We could replace the rest of the cases as missing on the scale variable.

```
replace scale = . if nonmiss < 4
```

2 Bad Uni-dimensional Model

Now, let's look at an example where the summated rating model is not appropriate. Here, the data are simulated, but they show the point well, so it's worth taking a look at them. In this dataset, there are 6 variables `V1-V6`. First, let's look at the correlation matrix:

```
 cor V*
(obs=1000)

             |       V1       V2       V3       V4       V5       V6
-------------+------------------------------------------------------
          V1 |   1.0000
          V2 |   0.7660   1.0000
          V3 |   0.7808   0.7540   1.0000
          V4 |  -0.4939  -0.4969  -0.4843   1.0000
          V5 |  -0.4883  -0.4923  -0.4816   0.8879   1.0000
          V6 |  -0.4935  -0.5046  -0.4783   0.8923   0.9011   1.0000
```

You can see that this correlation matrix is "block diagonal", that is there are high correlations among variables `V1-V3` and `V4-V6`, but negative correlations between variables 1-3 and variables 4-6. This would suggest that you would need a 2-dimensional model (especially when the off-diagonal block is populated with negative correlations) to account for these, but we'll try our uni-dimensional model and see what happens.

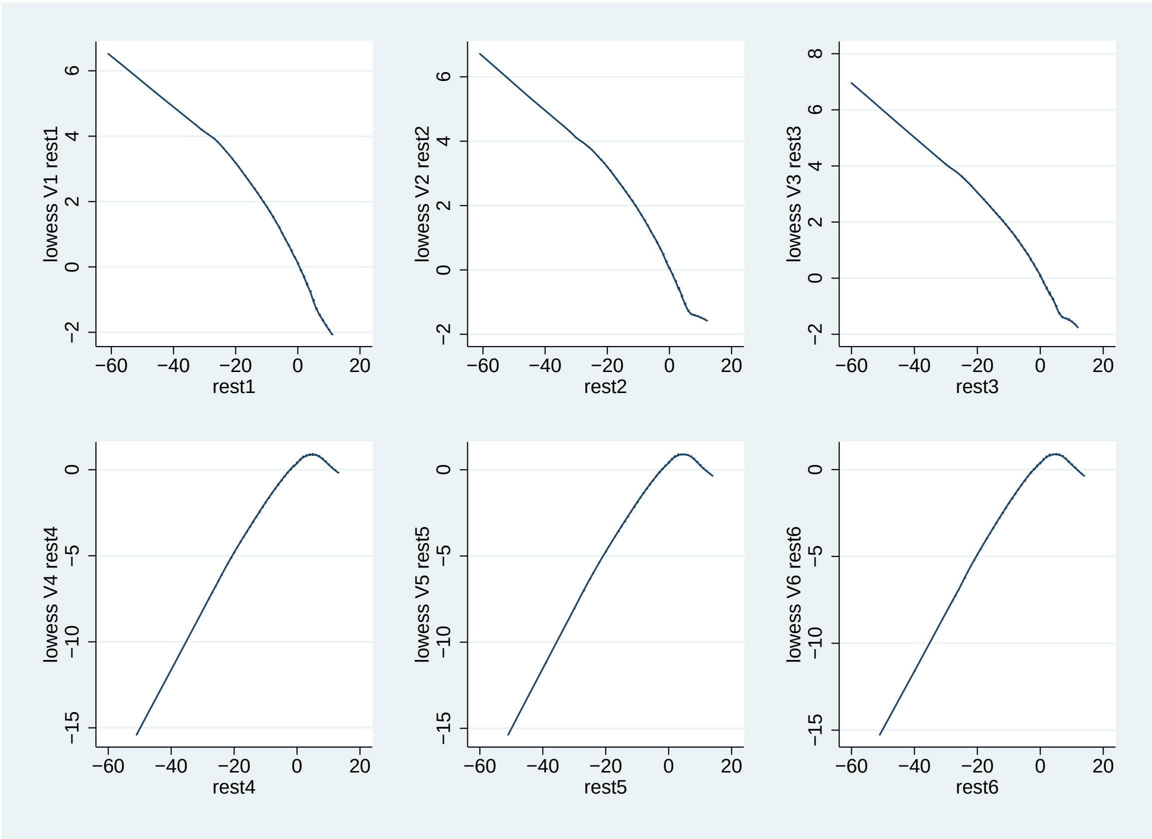
```
. alpha V*, std item

Test scale = mean(standardized items)

                                                            average
                                 item-test     item-rest     inter-item
Item         |  Obs  Sign   correlation   correlation     correlation     alpha
-------------+-----------------------------------------------------------------
V1           | 1000    -        0.8079        0.7175          0.6373      0.8978
V2           | 1000    -        0.8061        0.7151          0.6382      0.8982
V3           | 1000    -        0.7991        0.7054          0.6417      0.8995
V4           | 1000    +        0.8547        0.7831          0.6141      0.8883
V5           | 1000    +        0.8538        0.7819          0.6145      0.8885
V6           | 1000    +        0.8575        0.7872          0.6126      0.8877
-------------+-----------------------------------------------------------------
Test scale   |                                                0.6264      0.9096
-------------------------------------------------------------------------------
```

From the output, this looks like a pretty fantastic model. The α value is quite high and we couldn't get a higher value by removing any of the variables. Some of them are reversed, but that could be alright depending on the substantive meaning of each variable. Since we're out of the world of substance here, we'll assume that it's OK. However, when we look at the restplots, we can see some problems:

Figure 2: Restplots from bad model



The last three variables have some non-monotonicity toward the end of the graphs. These all look similar because they are quite similar because the data have been simulated for the purposes of demonstration, but you can see that even though the model looks good, there are some serious problems with using the summated rating model on these data.

Hopefully, the point here is clear - **ALWAYS** investigate the data before applying the model. Without doing so, you are leaving yourself open to making a pretty big statistical faux-pas.