

Measurement in the Social Sciences (TT 2007)  
Appendix 4.1: Latent Variables for Binary Data -  
Stata

Dave Armstrong  
University of Oxford  
Department of Politics and International Relations

e: david.armstrong@politics.ox.ac.uk  
w: <http://users.ox.ac.uk/~polf0104>

May 17, 2007

## 1 Traditional Factor Analysis

I thought that before we get into the cumulative models, I would show you exactly what I did to estimate the factor models on the tetrachoric correlation matrix, just in case you ever want to do something like that. First off, it's worth taking a look at your data. You'll want to assess the symmetry of the cross-tabulations of your observed variables. I wrote a little Stata routine called `taball` that will give you all of the pairwise cross-tabulations for a list of variables. For this part, we'll use the `bindat.dta`. First, let's look at the data:

```
. taball V*
```

V1	V2		Total
	0	1	
0	305	200	505
1	189	306	495
Total	494	506	1,000

... some results omitted to save space

V5	V6		Total
	0	1	
0	312	195	507
1	192	301	493
Total	504	496	1,000

As you can see that all of the tables look pretty symmetric, so this is one indication that we might be on the right track using traditional factor analysis. If you wanted to do factor analysis on the  $\phi$  matrix, all you have to do is treat your variables as though they're continuous and do the factor analysis. Let's try that:

	V1	V2	V3	V4	V5	V6
V1	1.0000					
V2	0.2221	1.0000				
V3	0.3682	0.2699	1.0000			
V4	0.0199	0.0461	0.0481	1.0000		
V5	0.0039	0.0662	0.0522	0.2719	1.0000	
V6	0.0219	0.0441	0.0181	0.1859	0.2259	1.0000

The correlation matrix doesn't look all that great. There are some relatively high correlations, but there are also lots of low ones.

```
. factor V*, ipf fac(1) (obs=1000)
```

```
Factor analysis/correlation      Number of obs =      1000
Method: iterated principal factors  Retained factors =      1
Rotation: (unrotated)             Number of params =      6
```

Factor	Eigenvalue	Difference	Proportion	Cumulative
Factor1	0.91520	0.45854	1.0000	1.0000
Factor2	0.45666	0.45587	0.4990	1.4990
Factor3	0.00079	0.02794	0.0009	1.4999
Factor4	-0.02715	0.14122	-0.0297	1.4702
Factor5	-0.16836	0.09360	-0.1840	1.2862
Factor6	-0.26196	.	-0.2862	1.0000

```
LR test: independent vs. saturated:  chi2(15) = 395.10 Prob>chi2 = 0.0000
```

```
Factor loadings (pattern matrix) and unique variances
```

Variable	Factor1	Uniqueness
V1	0.5257	0.7236
V2	0.4203	0.8233
V3	0.6435	0.5859
V4	0.1319	0.9826
V5	0.1380	0.9810
V6	0.1078	0.9884

Notice, this is not a particularly good fitting model. The uniquenesses are very high for some variables and the first factor explains a variance of less than one.

You might want to try estimating the factor model with the tetrachoric correlation matrix. You can do this first by creating the matrix and then giving it as an argument to the `factormat` command in Stata.

```
. tetrachoric V*
```

```
Tetrachoric correlations (N=1000)
```

Variable	V1	V2	V3	V4	V5	V6
V1	1					
V2	.3407	1				
V3	.5421	.4093	1			
V4	.03126	.07239	.07556	1		
V5	.006065	.1038	.08191	.4121	1	
V6	.03443	.06921	.02845	.2872	.3462	1

```
. matrix tR=r(corr)
```

```
. factormat tR, n(1000) fac(1) ipf (obs=1000)
```

```
Factor analysis/correlation          Number of obs =      1000
Method: iterated principal factors    Retained factors =      1
Rotation: (unrotated)                 Number of params =      6
```

Factor	Eigenvalue	Difference	Proportion	Cumulative
Factor1	1.36418	0.66723	1.0000	1.0000
Factor2	0.69695	0.69591	0.5109	1.5109
Factor3	0.00104	0.04861	0.0008	1.5117
Factor4	-0.04757	0.21029	-0.0349	1.4768
Factor5	-0.25786	0.13474	-0.1890	1.2878
Factor6	-0.39260	.	-0.2878	1.0000

```
LR test: independent vs. saturated:  chi2(15) = 916.95 Prob>chi2 = 0.0000
```

```
Factor loadings (pattern matrix) and unique variances
```

Variable	Factor1	Uniqueness
V1	0.6350	0.5968
V2	0.5295	0.7196
V3	0.7715	0.4048
V4	0.1753	0.9693
V5	0.1836	0.9663
V6	0.1447	0.9791

This model performs at least a little better. There is one clear factor with an eigenvalue of over one. The uniquenesses are still quite high for some of the variables, but they are a bit lower in this model than the one before. If you want to use this model, you could go ahead and interpret this just like a factor analysis on continuous data. You could also use the `predict` command in Stata to generate an estimate of the underlying dimension.

## 2 Mokken Model

The Mokken scaling model can be done in Stata using the `msp` add-on command. MSP is also the name of a boutique software package that estimates all sorts of scaling

models. For this part of class, we're going to use the `demirt.dta` dataset that again has a number of measures of democracy:

Variable	Descriptioni
j	DPI Independent Judiciary
regimtyp	ACLP dichotomous measure of democracy
parreg	Polity IV regulation of participation
xconst	Polity IV executive constraints highest level=1
xrreg	Polity IV regulation of executive recruitment
parcomp	Polity IV competitiveness of political participation
legsel	Banks CNTS Data Archive measure of legislative selection

First off, we want to look at the cross-tabs to see which model is most appropriate. We can use the `taball` command for this. I have omitted the output from here in the interest of space, but you can see that most of these are markedly asymmetric. This would suggest that we maybe should not use the traditional factor model on either the  $\phi$  matrix of the  $\tilde{r}$  matrix.

As a brief aside, let's take a look at the correlation matrices for these data:

```
. cor j-xconstd
```

	j	parreg	regimtyp	legsel	parcomp	xrregd	xconstd
j	1.0000						
parreg	0.5747	1.0000					
regimtyp	0.2441	0.1809	1.0000				
legsel	0.2591	0.1918	0.5081	1.0000			
parcomp	0.5946	0.6648	0.2558	0.2788	1.0000		
xrregd	0.4474	0.5239	0.3317	0.1120	0.6013	1.0000	
xconstd	0.5759	0.6932	0.2198	0.2335	0.6263	0.5789	1.0000

```
. tetrachoric j-xconstd
```

Tetrachoric correlations (N=3294)

Variable	j	parreg	regimtyp	legsel	parcomp	xrregd	xconstd
j	1						
parreg	.8494	1					
regimtyp	.9706	1	1				
legsel	.9733	1	.8404	1			
parcomp	.8071	1	.8761	.9215	1		
xrregd	.6798	.9638	.9595	.2762	.8256	1	
xconstd	.798	.9228	.9414	.9467	.853	.8797	1

Both matrices look alright, but especially the tetrachoric correlation matrix looks very nice indeed. We might be tempted just to do factor analysis on this matrix as the correlations are so high. Well, this would probably be the wrong thing to do. Let's look, for instance at the cross-tabulation of regime type and regulation of participation:

```
. tab regimtyp parreg
```

regimtyp	parreg		Total
	0	1	
0	346	0	346
1	2,230	718	2,948
Total	2,576	718	3,294

If we think back to the formula for the tetrachoric correlation approximation:

$$\alpha = \frac{a \times d}{b \times c}$$

$$\tilde{r} = \frac{\alpha^{\frac{\pi}{4}} - 1}{\alpha^{\frac{\pi}{4}} + 1}$$

you can see  $\alpha$  isn't even defined here because you're dividing  $a \times d$  by 0. This is why we have to look at the cross-tabulations because if we don't we might be given a false sense of our data by the tetrachoric correlation matrix.

Now, let's take a look at the Mokken model:

```
Scale: 1
-----
Significance level: 0.002381
The two first items selected in the scale 1 are parreg and regimtyp (Hjk=1.0000)
Significance level: 0.001923
The item parcomp is selected in the scale 1      Hj=0.9861      H=0.9877
Significance level: 0.001667
The item xrregd is selected in the scale 1      Hj=0.8518      H=0.9000
Significance level: 0.001515
The item xconstd is selected in the scale 1    Hj=0.8353      H=0.8688
Significance level: 0.001429
The item legsel is selected in the scale 1     Hj=0.6699      H=0.8306
Significance level: 0.001389
The item j is selected in the scale 1          Hj=0.6958      H=0.7801
Significance level: 0.001389
There is no more items remaining.
```

Item	Obs	Easyess P(Xj=1)	Observed Guttman errors	Expected Guttman errors	Loevinger H coeff	z-stat.	p-value
j	3294	0.3406	800	2630.12	0.69583	65.2164	0.00000
legsel	3294	0.8834	279	972.94	0.71324	34.5958	0.00000
xconstd	3294	0.2996	523	2478.36	0.78897	71.5913	0.00000
xrregd	3294	0.4927	589	2440.06	0.75861	63.2864	0.00000
parcomp	3294	0.3868	558	2662.24	0.79040	73.4039	0.00000
parreg	3294	0.2180	182	1939.95	0.90618	70.1459	0.00000
regimtyp	3294	0.8950	155	906.91	0.82909	39.0875	0.00000
Scale	3294		1543	7015.29	0.78005	113.9663	0.00000

The first column is obviously the variable name and the second is the number of observations. These could be different across variables. The third column is the discrimination parameter, or actually  $\delta_j$  is 1 minus the easiness. What this tells you is how much of the underlying trait (democracy here) do you have to have before we would expect you to have a 1 on the observed variable. Regime type is quite “easy” meaning that you don’t need much democracy before this variable classified you as a democracy. The “hardest” item is regulation of participation. So you need a lot of democracy before we would expect you to have a 1 on this variable. The fourth and fifth columns are the observed and expected errors, respectively. The sixth column is the  $H$ -statistic that we talked about in the notes. These are all above .5, so we should be pretty confident that they all belong in the scale. The scaling procedure in Stat by default kicks out any variable that has an item-specific  $H$  less than 0.3. You can change this behavior with the `minvalue()` argument to the command. The seventh and eighth columns just give inferential statistics for the  $H$ -statistic.

Now if we want to get estimates of the underlying dimension, we could just add up the observed variables:

```
egen scale=rowtotal(j-xconstd)
```

### 3 Rasch Model

Now, let's take a look at the Rasch model. For this, we'll use the data called `data.dta`, which was supplied by the author of the `raschtest` command.

```

Estimation method: Conditional maximum likelihood (CML)
Number of items: 9
Number of groups: 10 (8 of them are used to compute the statistics of test)
Number of individuals: 472 (11 individuals removed for missing values)
Number of individuals with null or perfect score: 87
Conditional log-likelihood: -1202.5759
Log-likelihood: -1677.6510

```

Items	Difficulty			df	p-value	Standardized		
	parameters	Std. Err.	R1c			Outfit	Infit	U
c1	0.04911	0.15671	27.160	7	0.0003	4.679	4.193	5.347
c2	2.66079	0.19295	5.055	7	0.6532	-0.825	-0.476	-1.091
c3	2.55202	0.18943	9.258	7	0.2346	-0.563	-1.375	-1.232
c4	1.25788	0.16309	3.002	7	0.8848	-0.827	-0.647	-0.727
c5	0.50164	0.15721	15.525	7	0.0298	2.759	1.746	3.032
c6	1.34035	0.16410	4.504	7	0.7202	1.373	1.055	1.488
c7	1.04442	0.16083	18.397	7	0.0103	-3.536	-3.180	-3.111
c8	2.11059	0.17757	11.800	7	0.1073	-1.385	-1.413	-1.257
c9*	0.00000	.	3.567	7	0.8281	-0.865	-0.637	-0.672
R1c test				R1c=	98.708	56	0.0004	
Andersen LR test				Z=	89.995	56	0.0027	

\*: The difficulty parameter of this item had been fixed to 0  
 You have groups of scores with less than 30 individuals. The tests can be invalid.

Group	Score	Ability		Freq.	Expected	
		parameters	Std. Err.		Score	ll
0	0	-2.089	2.300	82	0.43	
1	1	-0.799	0.568	74	1.30	-120.6067
2	2	-0.074	0.316	79	2.19	-222.1793
3	3	0.500	0.231	63	3.10	-219.4348
4	4	1.018	0.201	48	4.03	-178.6362
5	5	1.523	0.203	49	4.96	-202.6461
6	6	2.048	0.235	34	5.89	-121.2928
7	7	2.634	0.322	24	6.81	-69.3590
8	8	3.372	0.577	14	7.71	-23.4237
9	9	4.676	2.336	5	8.58	

To score this variable, you could either just add up these variables as that's what Stata will do if you ask it to generate a score. However, you could also the 0-7 values here with the ability parameters. You can either do it "by hand" or type this in the do-file editor and run it:

```
matrix theta=e(theta)
forvalues i=1/10{
    replace rasch=theta[1,'i'] if rasch == 'i'-1
}
```

The Rasch model in Stata also offers a number of tests of model fit.<sup>1</sup> In general, we want the statistics below to have values that are not statistically significant. If they are, that means the model is not a particularly good fit for that variable.  $R_{1c}$  has a p-value associated with it so you can judge the statistical significance of this term. **OUTFIT** and **INFIT** are statistics based on the residuals - the difference between each observed variable and the predicted probability of observing a 1 on that variable. These two indices follow roughly a standard normal distribution, so absolute values bigger than 2 might suggest the model does not fit particularly well - meaning there are significant outliers or inliers - unusual responses on that variable. The  $U$ -statistic is yet another test that we hope has no statistically significant value.

As you can see from the above table, some of the item-specific statistics are statistically significant. You can start by removing the worst offender, which in this case is probably `c1`. Then rerun the model without that variable and keep doing that until there are no statistically significant item-specific statistics.

---

<sup>1</sup>This discussion is an abbreviated version of the one offered in Hardouin, Jean-Benoit. 2007. "Rasch Analysis: Estimation with Tests" *The Stata Journal*. 7(1): 1-23. You can get this article at <http://anaqol.org/biblio/raschtest.pdf>