Item Response Theory (IRT) Models in Stata

Kristin MacDonald

Director of Statistical Services
StataCorp LP

2016 Italian Stata Users Group meeting
Rome

Outline

1. What is IRT?
2. Modeling binary responses
3. Extensions – ordinal responses, multiple groups, and more

What is IRT?

- IRT allows us to investigate unobservable traits such as mathematical ability, attitude toward a policy, or satisfaction with a product.
- IRT is useful in designing tests or questionnaires that allow us to measure these unobservable traits.
- Once tests or questionnaires are developed, IRT is useful in estimating an individual's level of the unobservable trait based on their responses.
- A common example is the use of a standardized test to measure a particular type of ability. These are often designed and scored using IRT.

Motivating example

- Suppose your company needs to hire a Stata "expert". What counts as Stata expertise?

- You cannot directly observe Stata expertise so you design a test with questions covering various aspects of Stata such as data management, graphics, statistical analyses, and programming.

- You start with an "easy" question:

  1. Spell your favorite statistical program.

     STATA
     R
     Stata

- You start with an "easy" question:

  1. Spell your favorite statistical program.

     STATA
     R
     Stata

- You are shocked that half the candidates selected STATA.
  - Is the question harder than you thought?
  - Or did you get a bunch of unqualified candidates?
  - Or maybe the question is useless?

- After the test is done, how do you select the best candidate?
  - Do you pick the one who got the most correct answers?
  - Or do you pick the one who got most of the hard questions right? If so, which questions were the most difficult?

- Item response theory is a method that lets you investigate unobserved Stata expertise from the observed answers to questions on your Stata expertise test.

- IRT is also a tool you can use to refine your Stata expertise test. Some items may be more useful than other items.

Some vocabulary:

- An individual question is called an *item*.
- A series of questions is called an *instrument* or a *test*.
- An answer to an item is called a *response*.
- The instrument measures an unobservable characteristic called a *latent trait*. In our example, the latent trait represents Stata expertise. In educational testing, the latent trait is usually called *ability*.
- The "theory" part of IRT formalizes the relationship between the latent trait, the items, and the responses.

- IRT is not limited to testing. It can be used to analyze the relationship between any latent trait and responses to test or survey questions.
- For example, a latent trait might be
  - A person's level of financial strain
    - This might be measured by responses to a series of items about whether the individual is deprived of specific goods and services because of a lack of finances.
  - Impact of a disease on the patient's life
    - This might be measured by responses to items such as whether the disease prevented the patient from participating in specific activities, a categorical rating of severity of pain, and a categorical rating of fatigue level.

- Food security
  - This might be measured by responses to questions such as whether an individual did not eat for an entire day, whether he cut the size of meals, and whether he worried about running out of food before getting more money.
- Customer satisfaction
  - This might be measured by responses to survey questions regarding how happy a customer is with the product purchased, how they were treated by the company's employees, whether the price was fair, and whether they would recommend the business or product to a friend.
- Family satisfaction with end of life care
  - This might be measured by responses to items about quality of patient care, emotional support, personalization of care, and coordination.

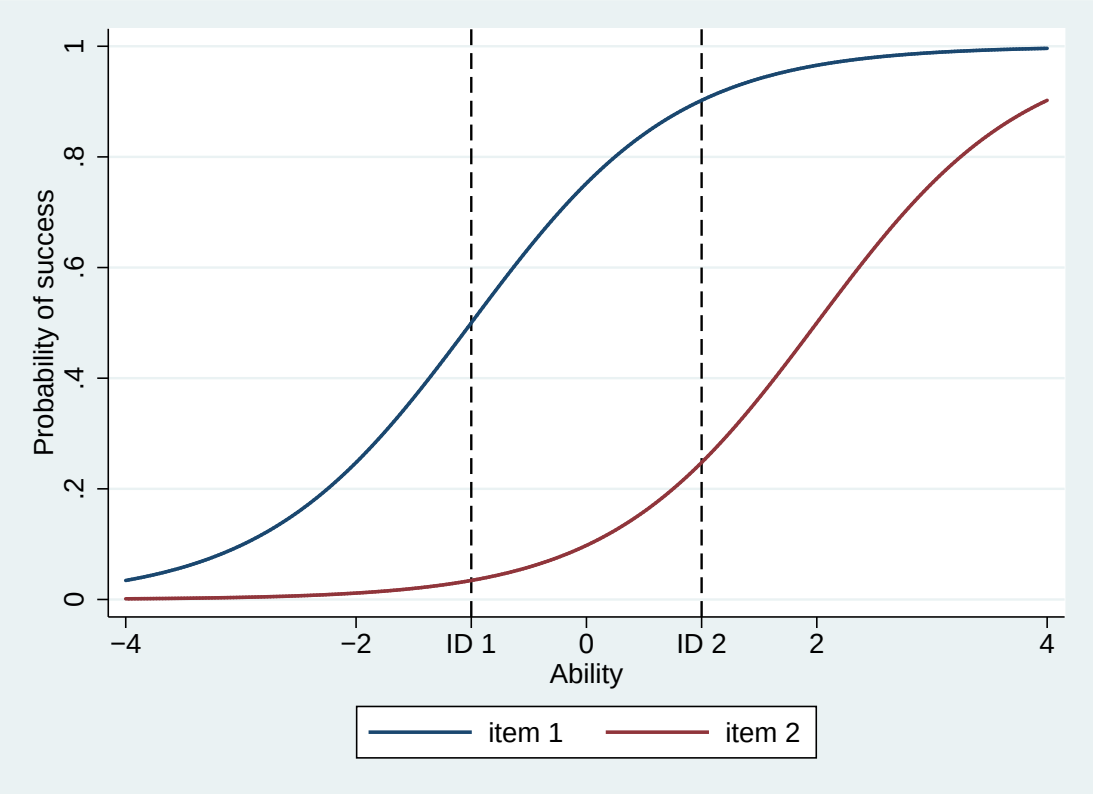
- From these examples, we can see that latent traits can be any unobservable characteristics and items may be binary, ordinal (Likert scale), or nominal.
- We focus first on models for binary items. We will think about this in terms of a test for an ability (Stata expertise).

- Individuals have different levels of ability.
- The probability of getting the correct answer increases with abilty.

- Items (test questions) have different levels of difficulty.
- The probability of getting the answer correct decreases with difficulty

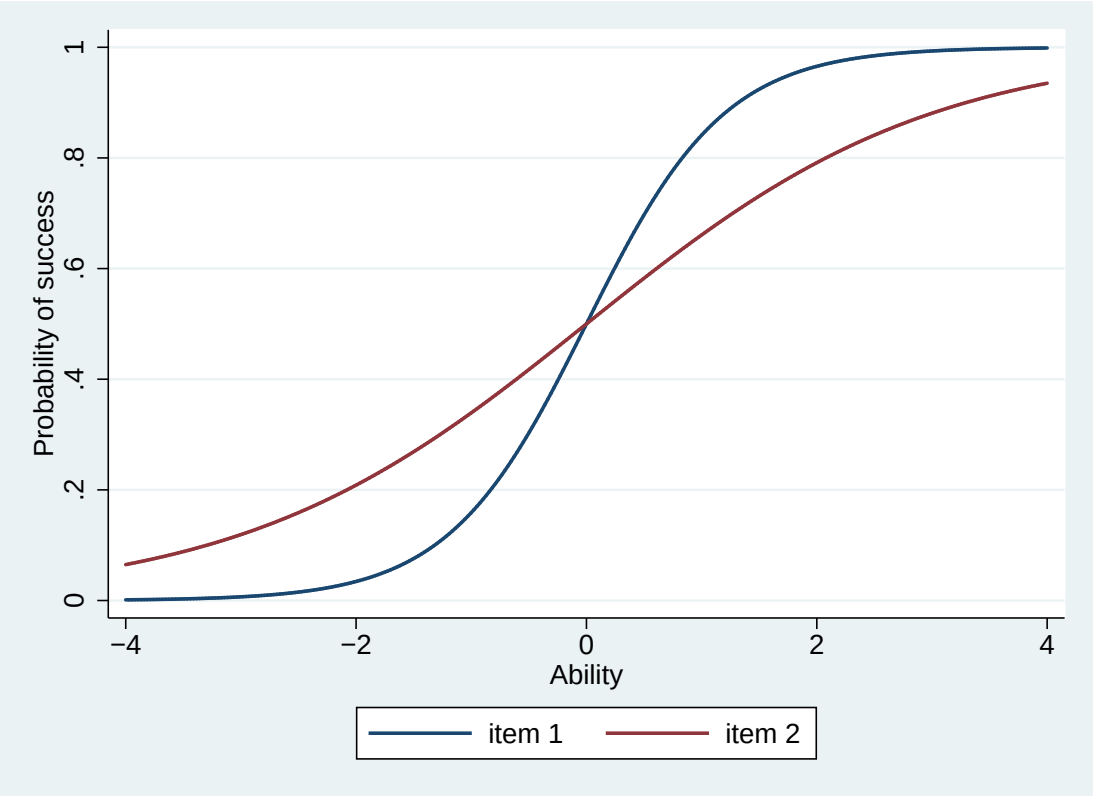
- In the simplest case of IRT, we model the probability of success on an item as a function of the respondent ability and the item difficulty.

It helps to think of this relationship graphically.



- These curves are called item characteristic curves (ICCs).
- An item's difficulty is the location (ability level) where the probability of success on the item is 0.5.
- Here, item 2 is more difficult.

We can also allow for items having different levels of discrimination.



- An item whose ICC has a steeper slope will distinguish better between low and high ability candidates.
- Here, item 1 is more discriminating.

A More Explicit Formulation

- Suppose that individual i has ability $\theta_i \sim N(0,1)$.
- Suppose that item j has difficulty b_j and discrimination a_j.
- We could then model the probability of a correct answer for person i on question j as

$$\Pr(Y_{ij} = 1|\theta_i) = F(a_j(\theta_i - b_j))$$

  where F is a cumulative distribution function
- We typically use the cumulative logistic distribution for binary outcomes.

$$\Pr(Y_{ij} = 1|\theta_i) = \frac{exp(a_j(\theta_i - b_j))}{1+exp(a_j(\theta_i - b_j))}$$

- We need two further assumptions:
  - The responses are driven by a single latent trait
  - Responses are independent, conditional on the latent trait

- Let's say our Stata expertise test has 11 questions.
- To fit an IRT model in Stata, our data must be arranged with an observation for each individual and a variable for each question.

```
. list q1-q11 in 1/5
```

	q1	q2	q3	q4	q5	q6	q7	q8	q9	q10	q11
1.	0	1	0	0	0	0	1	0	0	0	1
2.	1	0	0	0	1	0	1	0	0	1	1
3.	1	0	0	0	1	0	1	0	1	1	1
4.	0	0	0	0	0	0	1	0	0	0	0
5.	0	1	1	0	1	0	1	1	0	0	1

Only 11% of our applicants answered question q4 correctly while 70% answered question q11 correctly.

```
. summarize q1-q11
```

Variable	Obs	Mean	Std. Dev.	Min	Max
q1	300	.55	.498325	0	1
q2	300	.41	.4926551	0	1
q3	300	.18	.3848294	0	1
q4	300	.11	.3134125	0	1
q5	300	.6866667	.4646237	0	1
q6	300	.34	.4745003	0	1
q7	300	.7	.4590232	0	1
q8	300	.5366667	.4994869	0	1
q9	300	.2666667	.4429555	0	1
q10	300	.66	.4745003	0	1
q11	300	.7033333	.4575515	0	1

- The simplest IRT model for binary is called a one-parameter logistic (1PL) model. It allows our questions to vary in difficulty, but assumes that the discrimination is the same for all questions. Referring to the formula we saw previously,
  - F is the cumulative logistic distribution
  - $a_j = a$ for all j
  - only one parameter, b_j, is estimated separately for each item
- This is also known as the Rasch model.

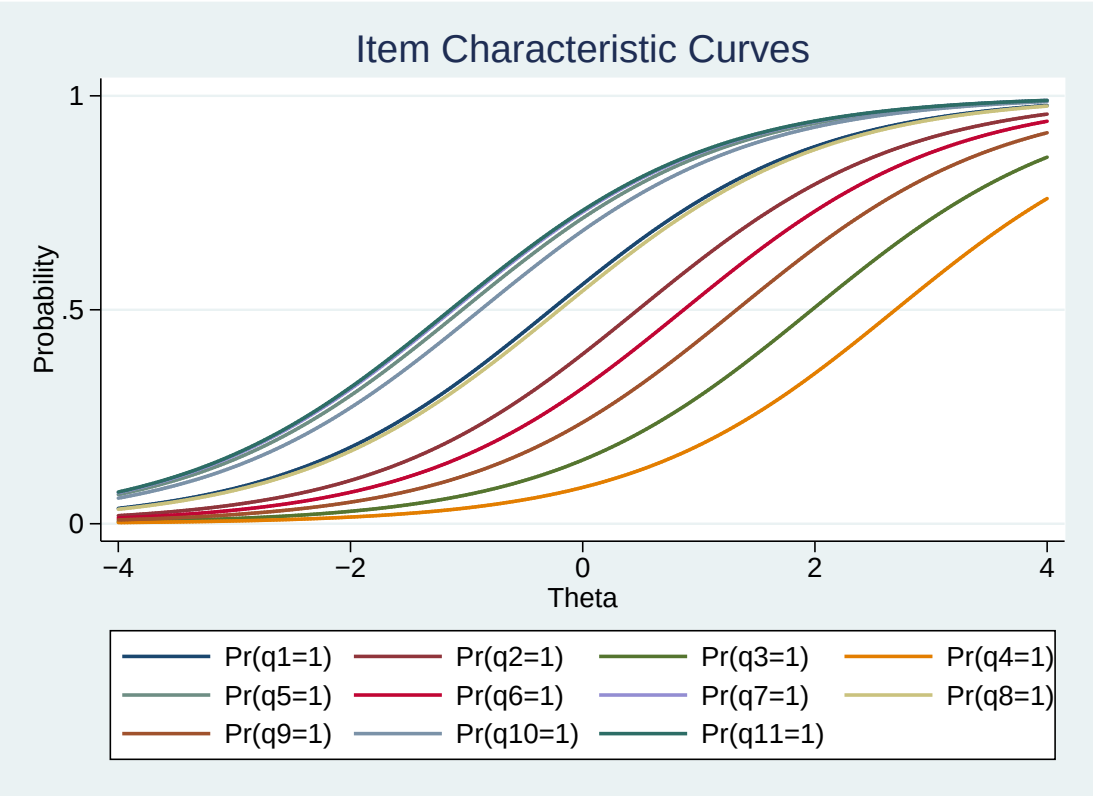
```
. irt 1pl q1-q11
  (output omitted)
One-parameter logistic model                    Number of obs     =        300
Log likelihood = -1897.8025
```

		Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
Discrim		.8825685	.0678221	13.01	0.000	.7496396	1.015497
q1	Diff	-.2717334	.1542369	-1.76	0.078	-.5740321	.0305653
q2	Diff	.4755239	.1580563	3.01	0.003	.1657392	.7853086
q3	Diff	1.97444	.2280679	8.66	0.000	1.527435	2.421445
q4	Diff	2.693887	.2854557	9.44	0.000	2.134404	3.25337
q5	Diff	-1.03764	.1745039	-5.95	0.000	-1.379662	-.695619

q6	Diff	.8702724	.1692517	5.14	0.000	.5385452	1.202
q7	Diff	-1.119027	.1779656	-6.29	0.000	-1.467833	-.7702209
q8	Diff	-.20071	.1535728	-1.31	0.191	-.5017072	.1002872
q9	Diff	1.325642	.1889948	7.01	0.000	.955219	1.696065
q10	Diff	-.8796848	.1684613	-5.22	0.000	-1.209863	-.5495067
q11	Diff	-1.139658	.1788797	-6.37	0.000	-1.490256	-.7890604

- We plot the model-implied ICCs for all questions by typing

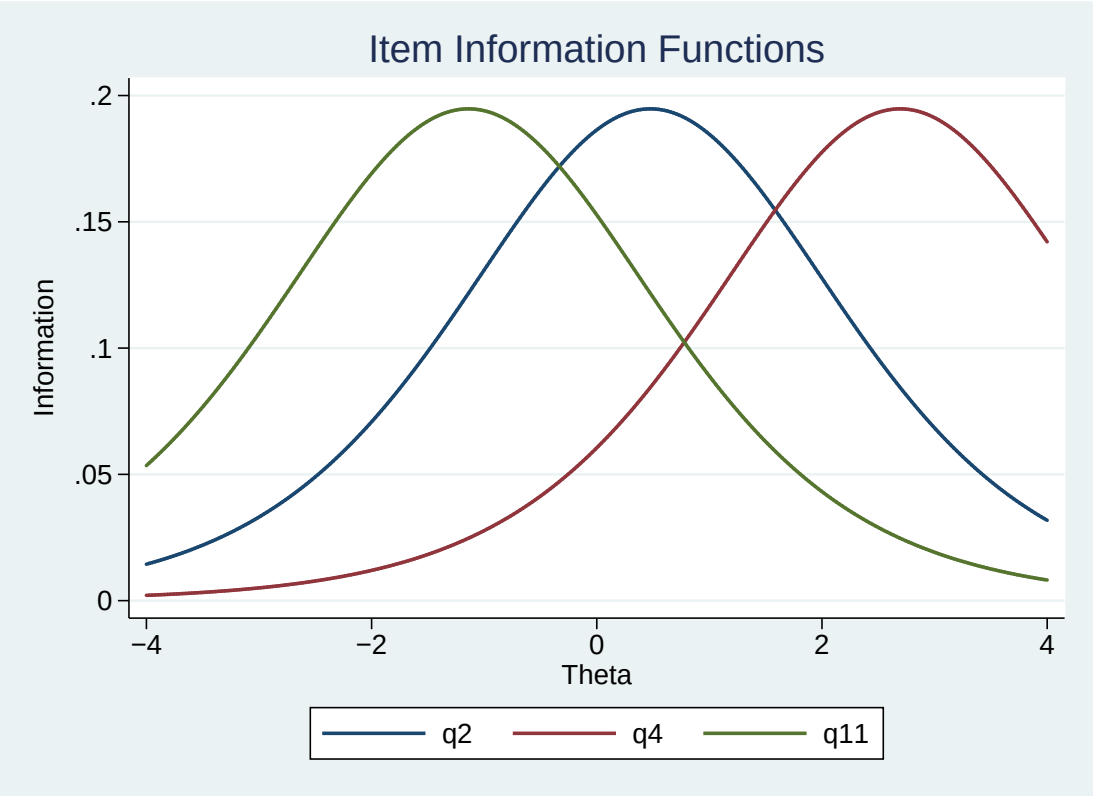
```
. irtgraph icc, legend(cols(4))
```



- Because the discrimination parameter is modeled to be the same for all questions, the curves are all shifted versions of the same curve.

- We plot the item information functions (IIFs) for questions q2, q4, and q11.

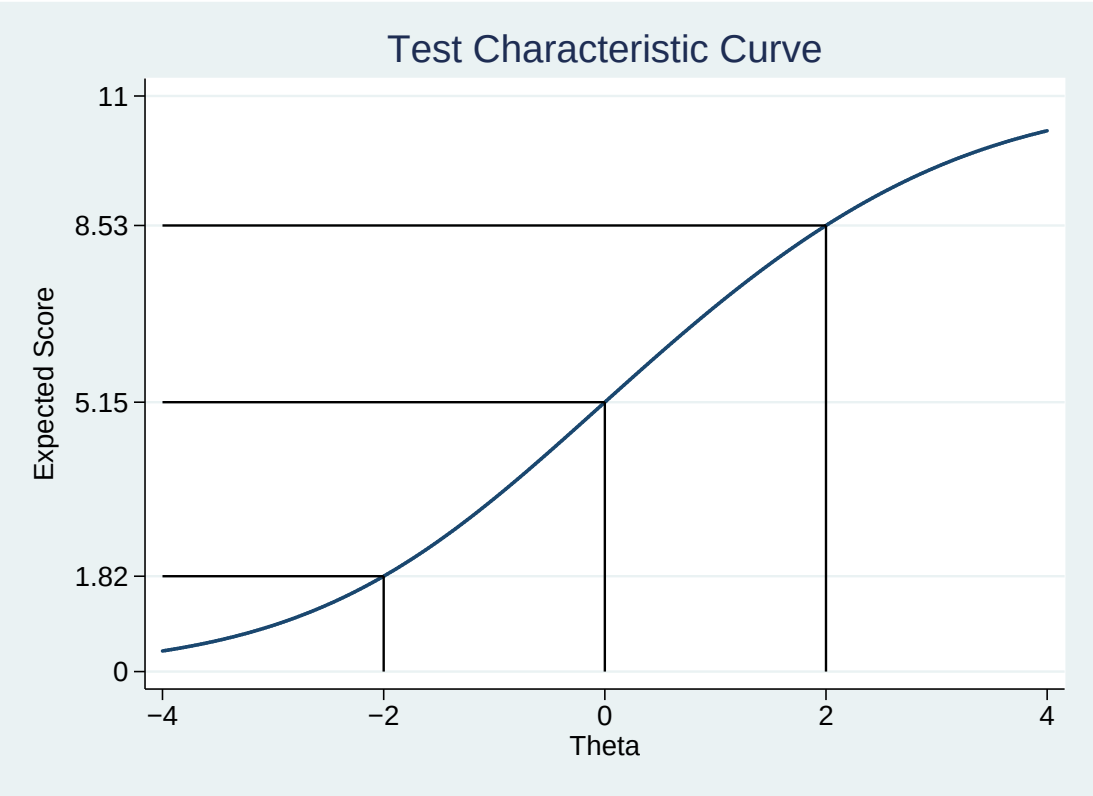
```
. irtgraph iif q2 q4 q11, legend(cols(3))
```



- The peak of this distribution is the location where the item provides the most information about Stata expertise. This location corresponds to the difficulty level of the question.
- Because we are fitting a one-parameter model and are not estimating separate discrimination parameters, the height of the curves are all the same.

- The test characteristic curve (TCC) plots the expected test score against Stata expertise.

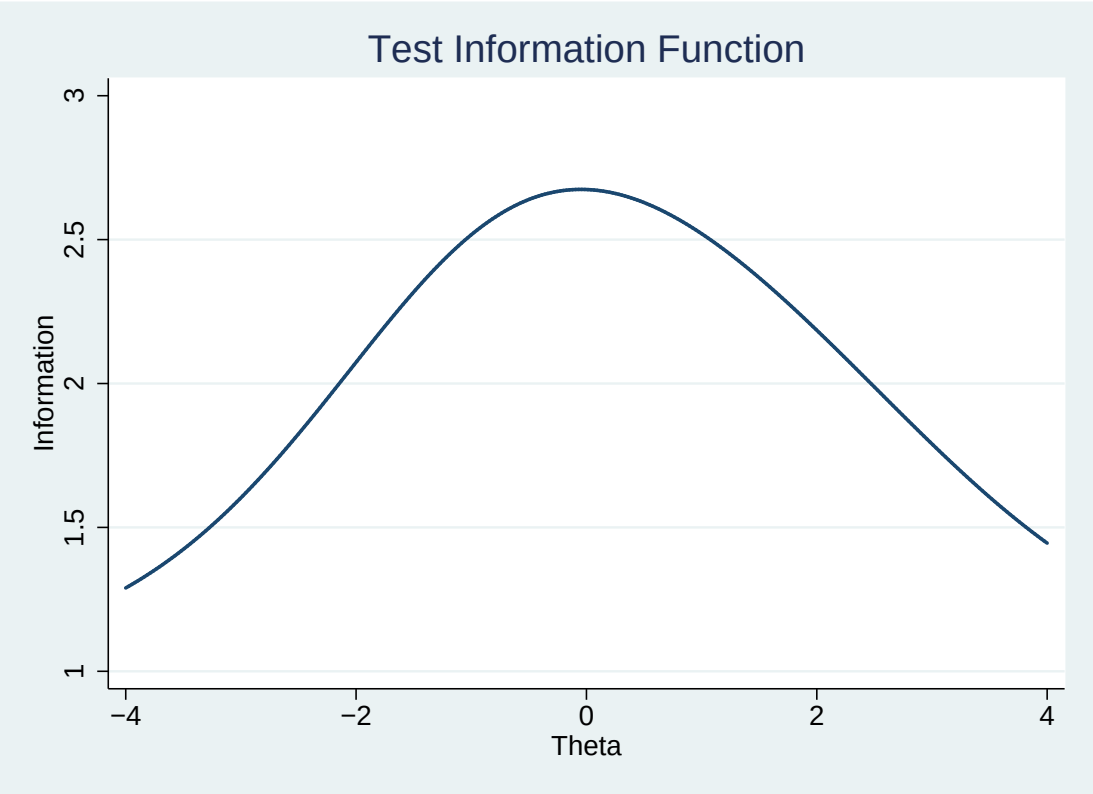
```
. irtgraph tcc, thetalines(-2 0 2)
```



- We expect someone with average expertise to answer 5 of the 11 questions correctly.

- The test information function (TIF) shows where the test gives the most reliable information about Stata expertise.

```
. irtgraph tif
```



- We can relax the constraint that all questions have the same discrimination and fit a two-parameter logistic (2PL) model.
- We now estimate an a_i for each question.
- We fit the 2PL model by typing

```
. irt 2pl q1-q11
```

Two-parameter logistic model Number of obs = 300
Log likelihood = -1879.4339

	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
q1						
Discrim	1.078989	.233719	4.62	0.000	.6209082	1.53707
Diff	-.2318368	.1372951	-1.69	0.091	-.5009302	.0372566
q2						
Discrim	1.558488	.3207313	4.86	0.000	.9298664	2.18711
Diff	.3337372	.114008	2.93	0.003	.1102856	.5571888
q3						
Discrim	1.355965	.3085007	4.40	0.000	.7513153	1.960616
Diff	1.473292	.2461484	5.99	0.000	.9908502	1.955734
q4						
Discrim	1.219007	.3198412	3.81	0.000	.5921298	1.845884
Diff	2.133656	.4100477	5.20	0.000	1.329977	2.937334
q5						
Discrim	.2914399	.1654996	1.76	0.078	-.0329334	.6158131
Diff	-2.745603	1.561356	-1.76	0.079	-5.805805	.3145983

q6						
	Discrim	1.143072	.2431083	4.70	0.000	.6665883	1.619555
	Diff	.7287562	.1681336	4.33	0.000	.3992204	1.058292
q7						
	Discrim	.6762115	.1945247	3.48	0.001	.2949502	1.057473
	Diff	-1.377988	.3882273	-3.55	0.000	-2.138899	-.6170761
q8						
	Discrim	.8085484	.194783	4.15	0.000	.4267808	1.190316
	Diff	-.2088736	.1682713	-1.24	0.214	-.5386793	.1209321
q9						
	Discrim	1.383048	.2927087	4.72	0.000	.8093494	1.956746
	Diff	.9836643	.1745053	5.64	0.000	.6416403	1.325688
q10						
	Discrim	.6322091	.1864631	3.39	0.001	.2667482	.99767
	Diff	-1.143299	.3542998	-3.23	0.001	-1.837714	-.4488842
q11						
	Discrim	.4013282	.1727228	2.32	0.020	.0627977	.7398586
	Diff	-2.230103	.9514962	-2.34	0.019	-4.095001	-.3652043

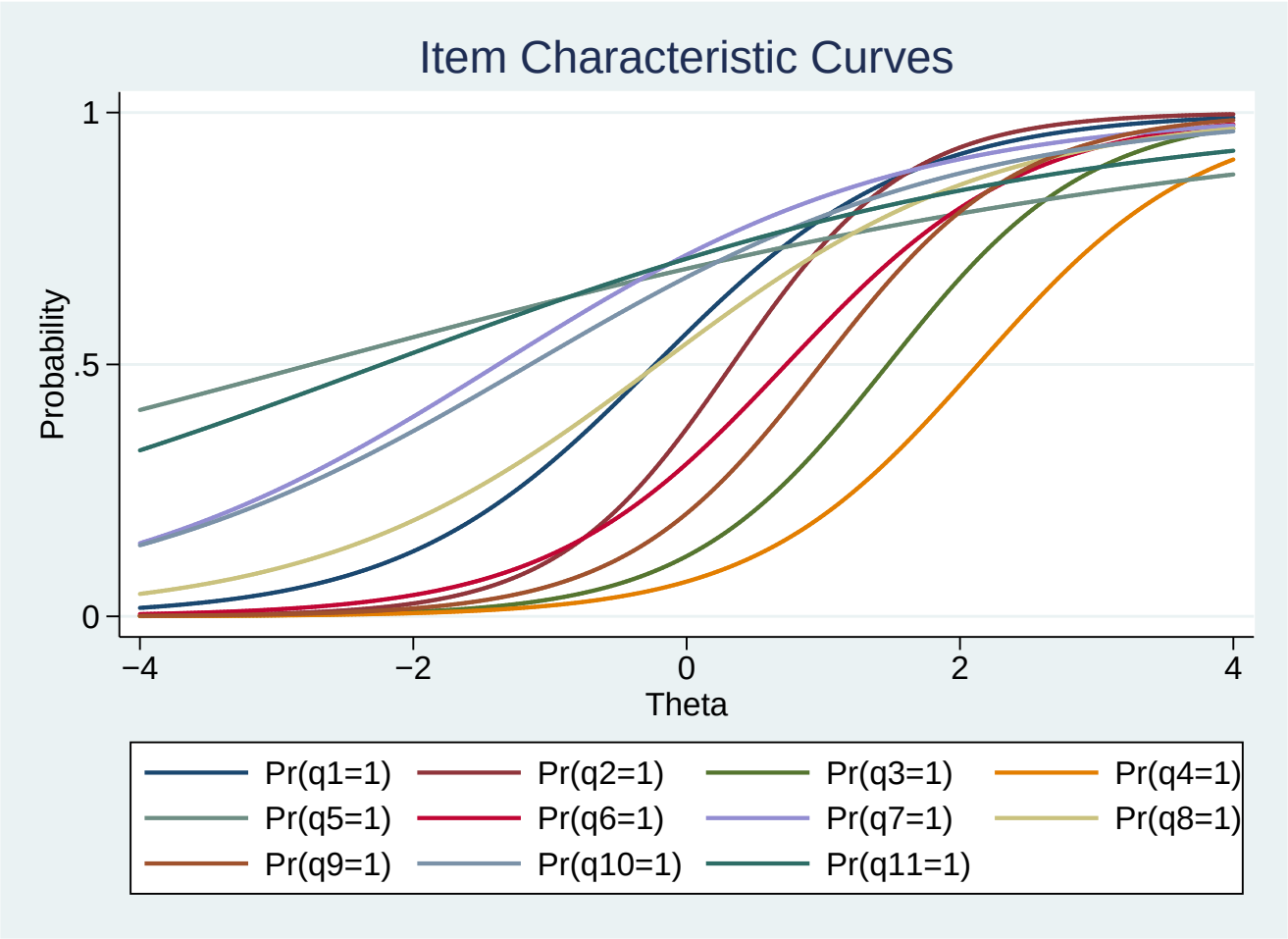
- We can sort the output by the estimated discrimination.

```
. estat report, byparm sort(a)
Two-parameter logistic model                    Number of obs     =        300
Log likelihood = -1879.4339
```

	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
Discrim						
q5	.2914399	.1654996	1.76	0.078	-.0329334	.6158131
q11	.4013282	.1727228	2.32	0.020	.0627977	.7398586
q10	.6322091	.1864631	3.39	0.001	.2667482	.99767
q7	.6762115	.1945247	3.48	0.001	.2949502	1.057473
q8	.8085484	.194783	4.15	0.000	.4267808	1.190316
q1	1.078989	.233719	4.62	0.000	.6209082	1.53707
q6	1.143072	.2431083	4.70	0.000	.6665883	1.619555
q4	1.219007	.3198412	3.81	0.000	.5921298	1.845884
q3	1.355965	.3085007	4.40	0.000	.7513153	1.960616
q9	1.383048	.2927087	4.72	0.000	.8093494	1.956746
q2	1.558488	.3207313	4.86	0.000	.9298664	2.18711
(*output omitted*)						

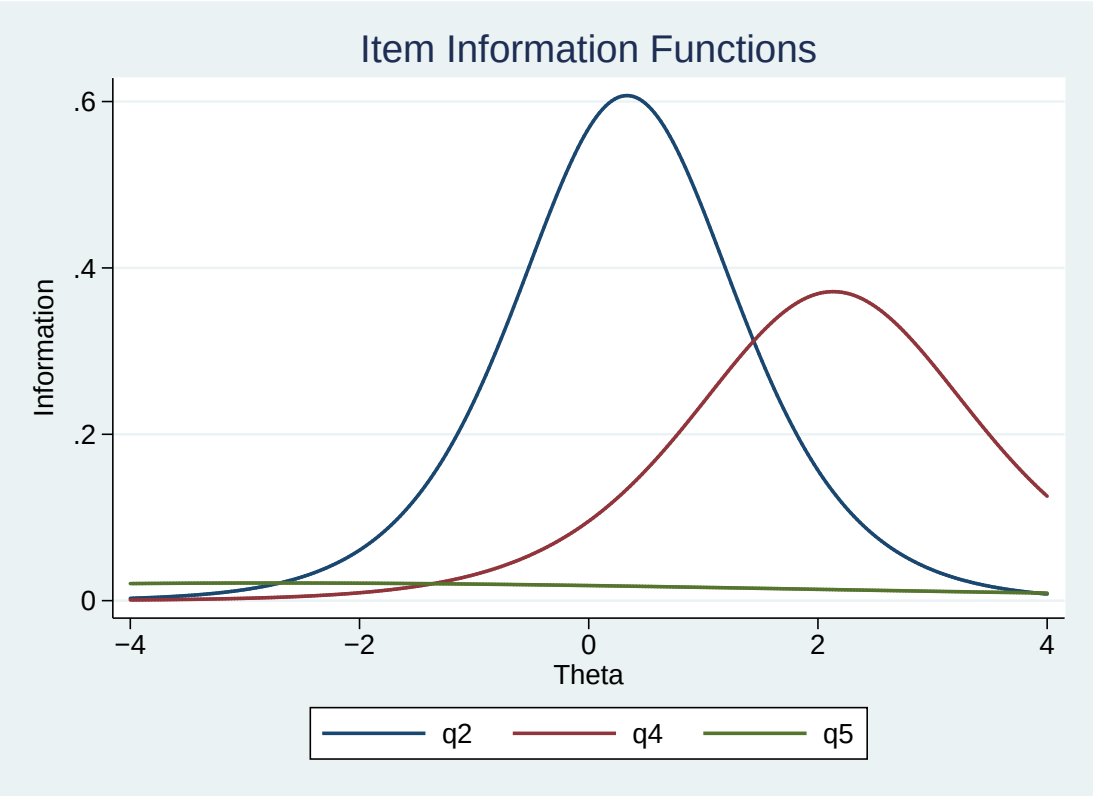
- The ICCs are no longer shifted versions of the same curve.

```
. irtgraph icc, legend(cols(4))
```



- More discriminating questions provide more information around their difficulty level.

```
. irtgraph iif q2 q4 q5, legend(cols(3))
```



- We can use a likelihood-ratio test to compare the two models:

```
. irt 1pl q1-q11
. estimates store onepl
. irt 2pl q1-q11
. estimates store twopl
. lrtest onepl twopl
```

```
. lrtest onepl twopl
Likelihood-ratio test                                 LR chi2(10) =     36.74
(Assumption: onepl nested in twopl)                   Prob > chi2 =    0.0001
```

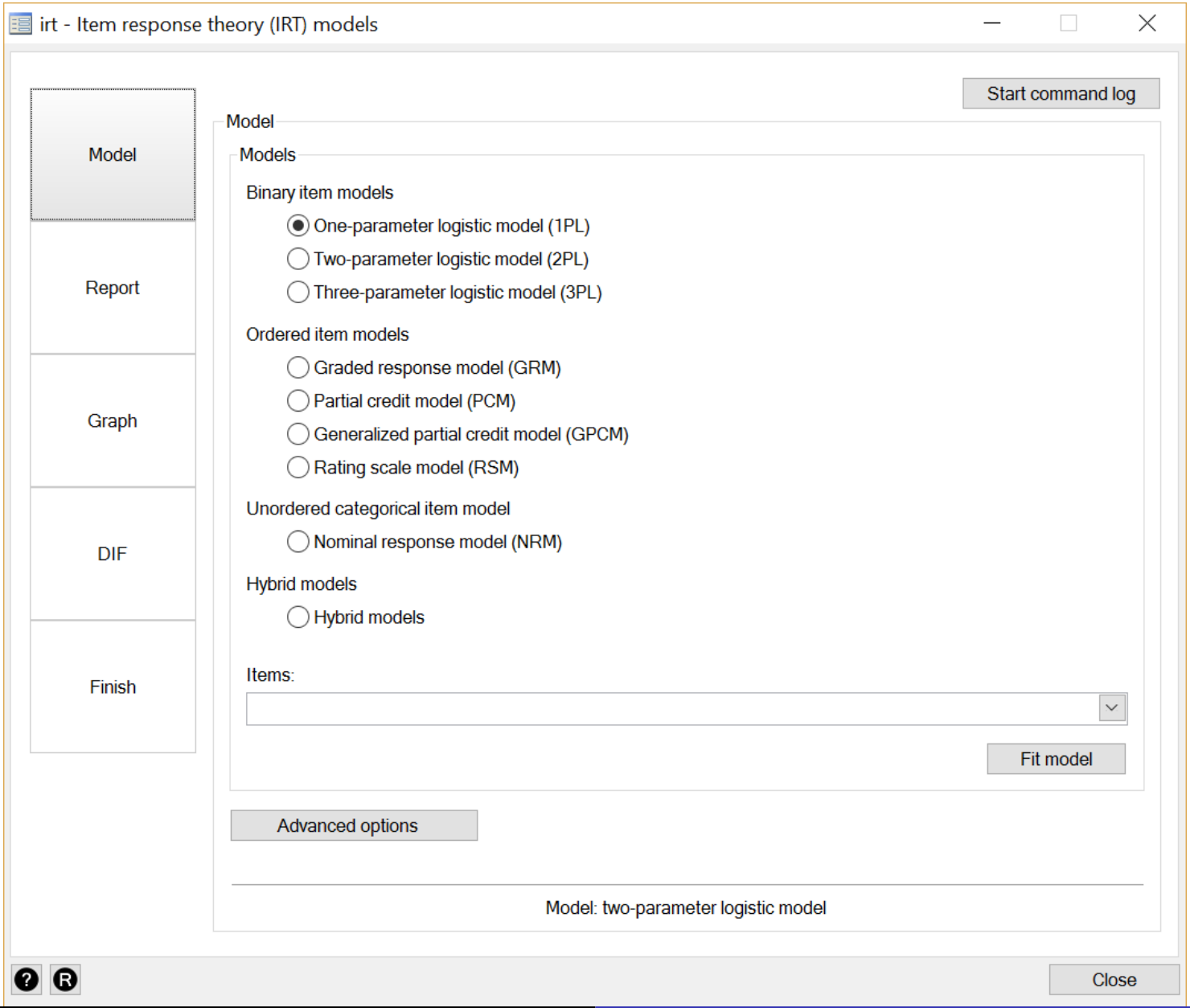
- We conclude the 2PL model is preferable in this case.

- We can now predict the Stata expertise level of our candidates based on the 2PL model.

```
. predict expertise, latent
(option ebmeans assumed)
(using 7 quadrature points)
. summarize expertise
    Variable |        Obs        Mean    Std. Dev.       Min        Max
-------------+---------------------------------------------------------
   expertise |        300    .0000389    .8013869  -1.673304   2.126724
. sort expertise
. list id expertise in -5/L

     +-------------------+
     |  id   expert~e    |
     |-------------------|
296. |  25   1.823069    |
297. | 293   1.886948    |
298. | 130   1.972455    |
299. | 285   2.126724    |
300. |  35   2.126724    |
     +-------------------+
```

- Extensions
  - Ordinal, categorical, and mixed responses
  - More extensions through `gsem`


irt - Item response theory (IRT) models
Start command log
Model
Report
Graph
DIF
Finish
Model
Models
Binary item models
One-parameter logistic model (1PL)
Two-parameter logistic model (2PL)
Three-parameter logistic model (3PL)
Ordered item models
Graded response model (GRM)
Partial credit model (PCM)
Generalized partial credit model (GPCM)
Rating scale model (RSM)
Unordered categorical item model
Nominal response model (NRM)
Hybrid models
Hybrid models
Items:
Fit model
Advanced options
Model: two-parameter logistic model
Close

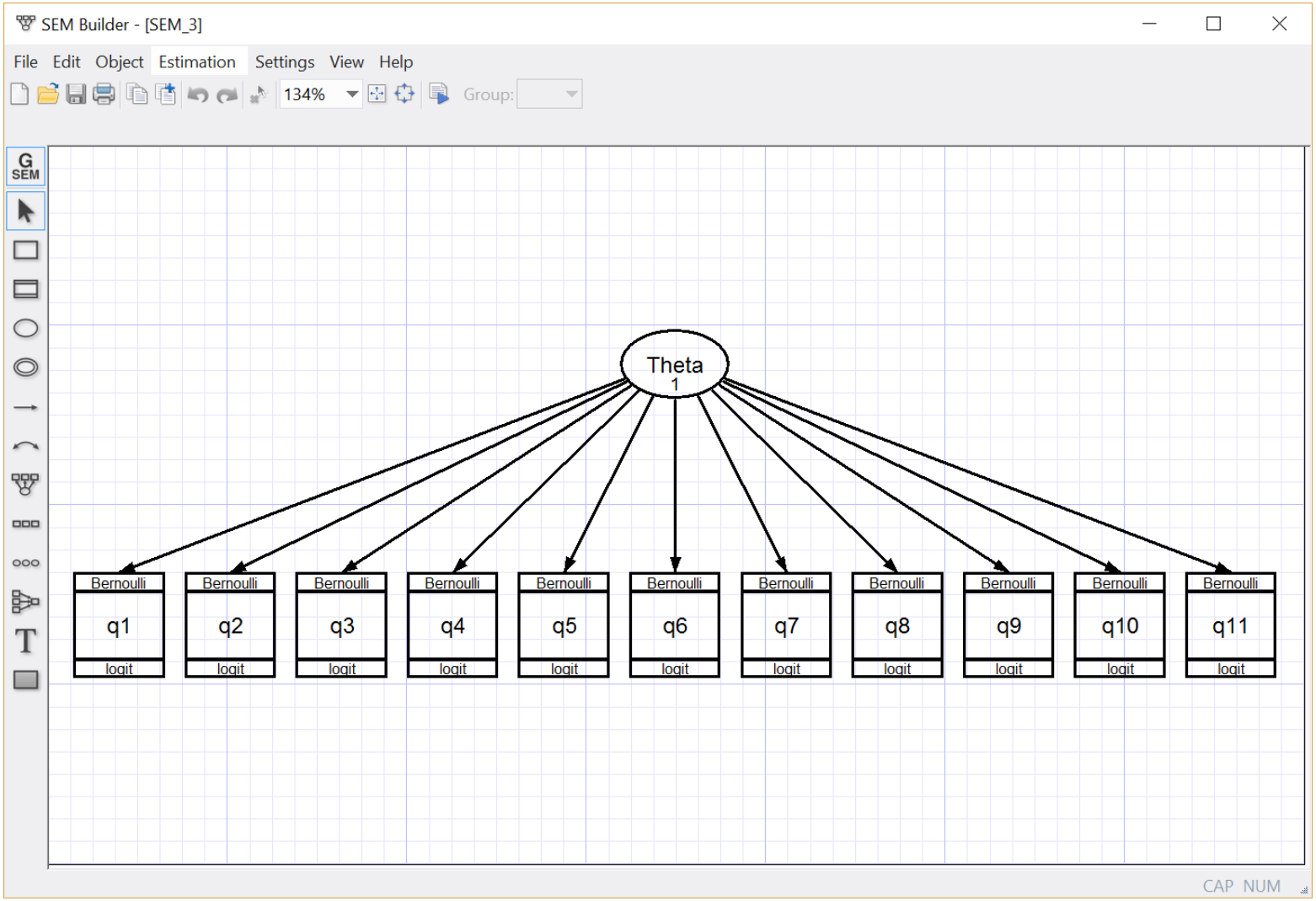
- The IRT commands in Stata are implemented using the command for fitting generalized structural equation models, `gsem`.
- We can see the `gsem` command that is being used if we type `display "‘e(cmdline2)’"` after fitting our model.

```
. irt 2pl q1-q11
  (output omitted)
. display "`e(cmdline2)´"
gsem (Theta -> q1 q2 q3 q4 q5 q6 q7 q8 q9 q10 q11, logit)   , variance(Theta@1)
>  latent(Theta) constraints( )
```

- We can actually simplify this a little and fit the 2PL model by typing

```
. gsem (Theta -> q1 q2 q3 q4 q5 q6 q7 q8 q9 q10 q11, logit),
variance(Theta@1)
```

- We might even draw the path diagram using the Builder.



- In either case, we get

```
. gsem (Theta -> q1 q2 q3 q4 q5 q6 q7 q8 q9 q10 q11, logit), variance(Theta@1)
  (output omitted)
```

		Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
q1 <-							
	Theta	1.078989	.233719	4.62	0.000	.6209082	1.53707
	_cons	.2501493	.1447082	1.73	0.084	-.0334736	.5337723
q2 <-							
	Theta	1.558488	.3207313	4.86	0.000	.9298664	2.18711
	_cons	-.5201255	.1752414	-2.97	0.003	-.8635924	-.1766586
q3 <-							
	Theta	1.355965	.3085007	4.40	0.000	.7513153	1.960616
	_cons	-1.997734	.2607476	-7.66	0.000	-2.508789	-1.486678
q4 <-							
	Theta	1.219007	.3198412	3.81	0.000	.5921298	1.845884
	_cons	-2.600941	.3152549	-8.25	0.000	-3.21883	-1.983053
(output omitted)							
var(Theta)		1	(constrained)				

- The output looks a bit different from what we saw with `irt 2pl`.
- This is because the IRT model in `gsem` is parameterized using the slope-intercept formulation with $\alpha_j + \theta_i\beta_j$ instead of the difficulty-discrimination parameterization with $a_j(\theta_i - b_j)$.
- A simple transformation converts one paramterization to the other. For the 2PL model, the discrimination is simply the slope, and the difficulty is the negative of the intercept divided by the slope.

- Using `gsem`, we can extend the IRT models available through Stata's `irt` commands in a variety of ways:
  - Fit models using other cumulative distributions
  - Fit multilevel models
  - Fit multiple-group models
  - Include an IRT model as part of a larger structural equation model

- While most of these extensions are straightforward with `gsem`, multiple-group analysis takes a little data management.
- We will first make copies of our questions for each group.
- Suppose that we gave this test to two different sets of candidates. One group (`group`= 1) took the test one month after Stata 14 was released and the other group (`group`= 2) took the test a year later.
- For simplicity, we will work with only 5 of our questions.

- We generate the new variables we need by typing

```
forvalues i =  1/5 {
        generate q‘i’_g1 = q‘i’ if group==1
        generate q‘i’_g2 = q‘i’ if group==2
}
```

- Now for each of our original questions, we have two variables, one for each group.

```
. list q1 group q1_g1 q1_g2 in 1/5
```

	q1	group	q1_g1	q1_g2
1.	0	2	.	0
2.	1	1	1	.
3.	1	2	.	1
4.	0	2	.	0
5.	0	1	0	.

- We are concerned that the model may behave differently across groups, in particular for question q1, because it involved working with Unicode characters and required the use of features introduced in Stata 14.
- We begin by fitting a model with the difficulty and discrimination parameters constrained to be equal across groups. Then we refit the model, allowing for the difficulty for q1 to vary across groups. We can then test for differences across groups.

```
gsem
   (Theta1                -> q1_g1)
   (Theta1                -> q2_g1)
   (Theta1                -> q3_g1)
   (Theta1                -> q4_g1)
   (Theta1                -> q5_g1)
```

```
gsem
  (Theta1     _cons     -> q1_g1)
  (Theta1     _cons     -> q2_g1)
  (Theta1     _cons     -> q3_g1)
  (Theta1     _cons     -> q4_g1)
  (Theta1     _cons     -> q5_g1)
```

```
gsem
  (Theta1    _cons    -> q1_g1)
  (Theta1    _cons    -> q2_g1)
  (Theta1    _cons    -> q3_g1)
  (Theta1    _cons    -> q4_g1)
  (Theta1    _cons    -> q5_g1)
  (Theta2    _cons    -> q1_g2)
  (Theta2    _cons    -> q2_g2)
  (Theta2    _cons    -> q3_g2)
  (Theta2    _cons    -> q4_g2)
  (Theta2    _cons    -> q5_g2),
```

```
gsem
  (Theta1@s1 _cons@i1 -> q1_g1)
  (Theta1    _cons    -> q2_g1)
  (Theta1    _cons    -> q3_g1)
  (Theta1    _cons    -> q4_g1)
  (Theta1    _cons    -> q5_g1)
  (Theta2@s1 _cons@i1 -> q1_g2)
  (Theta2    _cons    -> q2_g2)
  (Theta2    _cons    -> q3_g2)
  (Theta2    _cons    -> q4_g2)
  (Theta2    _cons    -> q5_g2),
```

```
gsem
  (Theta1@s1 _cons@i1 -> q1_g1)
  (Theta1@s2 _cons@i2 -> q2_g1)
  (Theta1@s3 _cons@i3 -> q3_g1)
  (Theta1@s4 _cons@i4 -> q4_g1)
  (Theta1@s5 _cons@i5 -> q5_g1)
  (Theta2@s1 _cons@i1 -> q1_g2)
  (Theta2@s2 _cons@i2 -> q2_g2)
  (Theta2@s3 _cons@i3 -> q3_g2)
  (Theta2@s4 _cons@i4 -> q4_g2)
  (Theta2@s5 _cons@i5 -> q5_g2),
```

```
gsem
  (Theta1@s1 _cons@i1 -> q1_g1)
  (Theta1@s2 _cons@i2 -> q2_g1)
  (Theta1@s3 _cons@i3 -> q3_g1)
  (Theta1@s4 _cons@i4 -> q4_g1)
  (Theta1@s5 _cons@i5 -> q5_g1)
  (Theta2@s1 _cons@i1 -> q1_g2)
  (Theta2@s2 _cons@i2 -> q2_g2)
  (Theta2@s3 _cons@i3 -> q3_g2)
  (Theta2@s4 _cons@i4 -> q4_g2)
  (Theta2@s5 _cons@i5 -> q5_g2),
     logit variance(Theta1@1)
     covariance(Theta1*Theta2@0)
     mean(Theta2)

estimates store constr
```

```
gsem
   (Theta1@s1 _cons    -> q1_g1)
   (Theta1@s2 _cons@i2 -> q2_g1)
   (Theta1@s3 _cons@i3 -> q3_g1)
   (Theta1@s4 _cons@i4 -> q4_g1)
   (Theta1@s5 _cons@i5 -> q5_g1)
   (Theta2@s1 _cons    -> q1_g2)
   (Theta2@s2 _cons@i2 -> q2_g2)
   (Theta2@s3 _cons@i3 -> q3_g2)
   (Theta2@s4 _cons@i4 -> q4_g2)
   (Theta2@s5 _cons@i5 -> q5_g2),
      logit variance(Theta1@1)
      covariance(Theta1*Theta2@0)
      mean(Theta2)

estimates store unconstr
```

- One way to test for a difference in the difficulty across groups is to use a likelihood-ratio test comparing these two models.

```
. lrtest constr unconstr
Likelihood-ratio test                                 LR chi2(1)  =      5.30
(Assumption: constr nested in unconstr)               Prob > chi2 =    0.0213
```

- We conclude that the difficulty parameters for q1 differ across groups.

- Stata's `irt` commands make it easy to fit the most common IRT models.
- `gsem` allows for many extensions.
- Give IRT in Stata a try!