Title

irt, group() — IRT models for multiple groups

Description

Multiple-group IRT models are combined models across groups of the data. They allow some parameters to vary across groups and constrain others to be equal. The groups could be males and females, age categories, and the like.

The `irt` commands fit multiple-group IRT models when the `group(`*varname*`)` option is specified.

Quick start

1PL model for binary items `b1` to `b9` with difficulty and discrimination parameters equal across groups

```
irt 1pl b1-b9, group(grpvar)
```

As above, but with difficulty and discrimination parameters allowed to differ across groups

```
irt (0: 1pl b1-b9) (1: 1pl b1-b9), group(grpvar)
```

As above, but with discrimination parameter (`a`) equal across groups

```
irt (0: 1pl b1-b9, cns(a@k)) (1: 1pl b1-b9, cns(a@k)), group(grpvar)
```

1PL model with different items administered across groups

```
irt (0: 1pl b1-b7) (1: 1pl b3-b9), group(grpvar)
```

As above, but with difficulty and discrimination of `b3-b7` equal across groups

```
irt (0: 1pl b1 b2) (1pl b3-b7) (1: 1pl b8 b9), group(grpvar)
```

Display parameter estimates in compact form

```
estat greport
```

Plot ICCs for item `b5` for both groups

```
irtgraph icc b5
```

As above, but change the line pattern for group 1

```
irtgraph icc (0: b5) (1: b5, lpattern(dash))
```

Menu

Statistics > IRT (item response theory)

Syntax

Single-equation syntax

`irt` *model* *varlist* ..., `group(`*varname*`)` [*options*]

Multiple-equation syntax

`irt` `(`[#:] *model* $varlist_1$ [, *mopts*]`)` `(`[#:] *model* $varlist_2$ [, *mopts*]`)` ...,
`group(`*varname*`)` [*options*]

#: specifies the group for which *model* is to be fit.

model	Description
`1pl`	One-parameter logistic model
`2pl`	Two-parameter logistic model
`3pl`	Three-parameter logistic model
`grm`	Graded response model
`pcm`	Partial credit model
`gpcm`	Generalized partial credit model
`rsm`	Rating scale model
`nrm`	Nominal response model

options	Description
group(*varname*)	fit model for different groups
Model	
* cns(*spec*)	apply specified parameter constraints
listwise	drop observations with any missing items
* sepguessing	estimate a separate pseudoguessing parameter for each item
* gsepguessing	estimate separate pseudoguessing parameters for each group
SE/Robust	
vce(*vcetype*)	*vcetype* may be oim, robust, cluster *clustvar*, bootstrap, or jackknife
Reporting	
level(#)	set confidence level; default is level(95)
notable	suppress coefficient table
noheader	suppress output header
display_options	control columns and column formats
Integration	
intmethod(*intmethod*)	integration method
intpoints(#)	set the number of integration points; default is intpoints(7)
Maximization	
maximize_options	control the maximization process; seldom used
startvalues(*svmethod*)	method for obtaining starting values
noestimate	do not fit the model; show starting values instead
estmetric	show parameter estimates in the estimation metric
dnumerical	use numerical derivative techniques
coeflegend	display legend instead of statistics

* *mopts* are cns(), sepguessing, and gsepguessing.

intmethod	Description
mvaghermite	mean–variance adaptive Gauss–Hermite quadrature; the default
mcaghermite	mode-curvature adaptive Gauss–Hermite quadrature
ghermite	nonadaptive Gauss–Hermite quadrature

See **[U] 20 Estimation and postestimation commands** for more capabilities of estimation commands.

Options

group(*varname*) specifies that the model parameters be allowed to vary across values of *varname*. *varname* might be sex, and then parameters may vary for males and females, or *varname* might be something else and perhaps take on more than two values. Whatever *varname* is, group(*varname*) defaults to constraining all item coefficients to be equal across groups in each model specified without a group identifier. The mean and the variance of the latent trait are constrained to 0 and 1, respectively, for the group corresponding to the smallest value of *varname* (reference group) and estimated for the remaining groups (focal groups).

Model

cns(*spec*) constrains item parameters to a fixed value or constrains two or more parameters to be equal; see [IRT] **irt constraints** for details.

listwise handles missing values through listwise deletion, which means that the entire observation is omitted from the estimation sample if any of the items are missing for that observation. By default, all nonmissing items in an observation are included in the likelihood calculation; only missing items are excluded.

sepguessing specifies that a separate pseudoguessing parameter be estimated for each item. This option is allowed only with a 3pl model; see [IRT] **irt 3pl** for details.

gsepguessing specifies that separate pseudoguessing parameters be estimated for each group. This option is allowed only with a 3pl model.

SE/Robust

vce(*vcetype*) specifies the type of standard error reported, which includes types that are derived from asymptotic theory (oim), that are robust to some kinds of misspecification (robust), that allow for intragroup correlation (cluster *clustvar*), and that use bootstrap or jackknife methods (bootstrap, jackknife); see [R] ***vce_option***.

Reporting

level(*#*); see [R] **Estimation options**.

notable suppresses the estimation table, either at estimation or upon replay.

noheader suppresses the output header, either at estimation or upon replay.

display_options: noci, nopvalues, cformat(*%fmt*), pformat(*%fmt*), sformat(*%fmt*), and nolstretch; see [R] **Estimation options**.

Integration

intmethod(*intmethod*) specifies the integration method to be used for computing the log likelihood. mvaghermite performs mean and variance adaptive Gauss–Hermite quadrature; mcaghermite performs mode and curvature adaptive Gauss–Hermite quadrature; and ghermite performs non-adaptive Gauss–Hermite quadrature.

The default integration method is mvaghermite.

intpoints(*#*) sets the number of integration points for quadrature. The default is intpoints(7), which means that seven quadrature points are used to compute the log likelihood.

The more integration points, the more accurate the approximation to the log likelihood. However, computation time increases with the number of integration points.

Maximization

maximize_options: difficult, technique(*algorithm_spec*), iterate(*#*), [no]log, trace, gradient, showstep, hessian, showtolerance, tolerance(*#*), ltolerance(*#*), nrtolerance(*#*), nonrtolerance, and from(*init_specs*); see [R] **Maximize**. Those that require special mention for irt are listed below.

from() accepts a properly labeled vector of initial values or a list of coefficient names with values. A list of values is not allowed.

The following options are available with `irt` but are not shown in the dialog box:

`startvalues()` specifies how starting values are to be computed. Starting values specified in `from()` override the computed starting values.

`startvalues(zero)` specifies that all starting values be set to 0. This option is typically useful only when specified with the `from()` option.

`startvalues(constantonly)` builds on `startvalues(zero)` by fitting a constant-only model for each response to obtain estimates of intercept and cutpoint parameters.

`startvalues(fixedonly)` builds on `startvalues(constantonly)` by fitting a full fixed-effects model for each response variable to obtain estimates of coefficients along with intercept and cutpoint parameters. You can also add suboption `iterate(#)` to limit the number of iterations `irt` allows for fitting the fixed-effects model.

`startvalues(ivloadings)` builds on `startvalues(fixedonly)` by using instrumental-variable methods with the generalized residuals from the fixed-effects models to compute starting values for latent-variable loadings. This is the default behavior.

`noestimate` specifies that the model is not to be fit. Instead, starting values are to be shown (as modified by the above options if modifications were made), and they are to be shown using the `coeflegend` style of output. An important use of this option is before you have modified starting values at all; you can type the following:

```
. irt ..., ... noestimate
. matrix b = e(b)
. ... (modify elements of b) ...
. irt ..., ... from(b)
```

`estmetric` displays parameter estimates in the slope-intercept metric that is used for estimation.

`dnumerical` specifies that during optimization, the gradient vector and Hessian matrix be computed using numerical techniques instead of analytical formulas. By default, `irt` uses analytical formulas for computing the gradient and Hessian for all integration methods.

`coeflegend`; see [R] **Estimation options**.

Remarks and examples stata.com

Remarks are presented under the following headings:

Overview

The following discussion is about how to perform multiple-group analysis with `irt`. If you are new to the IRT features in Stata, we encourage you to read [IRT] **irt** first.

Multiple-group IRT analysis is usually performed when you believe one or more items function differently across groups; see [IRT] **DIF** for further details.

Baseline group model

▷ Example 1: Fitting a 2PL model for different groups of the data

To illustrate a multiple-group IRT model, we use an abridged version of the mathematics and science data from De Boeck and Wilson (2004). Student responses to test items are coded 1 for correct and 0 for incorrect. There are 761 male students and 739 female students.

```
. use https://www.stata-press.com/data/r17/masc2
(Data from De Boeck & Wilson (2004))

. tabulate female

     Female |      Freq.     Percent        Cum.
------------+-----------------------------------
       Male |        761       50.73       50.73
     Female |        739       49.27      100.00
------------+-----------------------------------
      Total |      1,500      100.00
```

Say we are interested in measuring mathematical ability using binary items q1–q5. We fit a two-group 2PL model as follows:

```
. irt 2pl q1-q5, group(female)
Fitting fixed-effects model:
Iteration 0:   log likelihood = -4594.5412
Iteration 1:   log likelihood = -4590.4516
Iteration 2:   log likelihood = -4590.4502
Iteration 3:   log likelihood = -4590.4502
Group: Male
Group: Female
Fitting full model:
Iteration 0:   log likelihood = -4503.5396  (not concave)
Iteration 1:   log likelihood = -4479.7967
Iteration 2:   log likelihood = -4476.3965
Iteration 3:   log likelihood = -4476.3448
Iteration 4:   log likelihood = -4476.3447
```

```
Two-parameter logistic model                         Number of obs = 1,500
Log likelihood = -4476.3447
Group: Male
```

	Coefficient	Std. err.	z	P>|z|	[95% conf.	interval]
q1						
Discrim	1.187923	.1804778	6.58	0.000	.8341933	1.541653
Diff	-.5507796	.0894536	-6.16	0.000	-.7261054	-.3754538
q2						
Discrim	.90663	.1318739	6.87	0.000	.6481618	1.165098
Diff	-.0450698	.0761722	-0.59	0.554	-.1943645	.104225
q3						
Discrim	.8828704	.1462984	6.03	0.000	.5961307	1.16961
Diff	-1.703158	.2385734	-7.14	0.000	-2.170753	-1.235563
q4						
Discrim	.8196789	.1221824	6.71	0.000	.5802057	1.059152
Diff	.3770973	.0993197	3.80	0.000	.1824342	.5717603
q5						
Discrim	1.439933	.2218141	6.49	0.000	1.005185	1.874681
Diff	1.197739	.1437481	8.33	0.000	.9159978	1.47948
mean(Theta)	0	(omitted)				
var(Theta)	1	(constrained)				

Group: Female

	Coefficient	Std. err.	z	P>|z|	[95% conf.	interval]
q1						
Discrim	1.187923	.1804778	6.58	0.000	.8341933	1.541653
Diff	-.5507796	.0894536	-6.16	0.000	-.7261054	-.3754538
q2						
Discrim	.90663	.1318739	6.87	0.000	.6481618	1.165098
Diff	-.0450698	.0761722	-0.59	0.554	-.1943645	.104225
q3						
Discrim	.8828704	.1462984	6.03	0.000	.5961307	1.16961
Diff	-1.703158	.2385734	-7.14	0.000	-2.170753	-1.235563
q4						
Discrim	.8196789	.1221824	6.71	0.000	.5802057	1.059152
Diff	.3770973	.0993197	3.80	0.000	.1824342	.5717603
q5						
Discrim	1.439933	.2218141	6.49	0.000	1.005185	1.874681
Diff	1.197739	.1437481	8.33	0.000	.9159978	1.47948
mean(Theta)	-.1348222	.0721434	-1.87	0.062	-.2762206	.0065763
var(Theta)	.6239155	.1239068			.4227474	.9208111

By default, the item parameters (the difficulty and discrimination) are constrained to be equal across groups. The mean and variance of the reference group (males) are constrained to 0 and 1, while the mean and variance of the focal group (females) are estimated.

We use `estat greport` to arrange the output in a more compact and readable format.

```
. estat greport
```

Parameter	Male	Female
q1		
Discrim	1.1879233	1.1879233
Diff	-.55077963	-.55077963
q2		
Discrim	.90662997	.90662997
Diff	-.04506976	-.04506976
q3		
Discrim	.88287037	.88287037
Diff	-1.7031579	-1.7031579
q4		
Discrim	.81967885	.81967885
Diff	.37709727	.37709727
q5		
Discrim	1.4399331	1.4399331
Diff	1.1977389	1.1977389
mean(Theta)	0	-.13482217
var(Theta)	1	.6239155

Now we can clearly see which parameters are constrained across groups and which are estimated freely.

We store our estimates for later use.

```
. estimates store nodif
```

◁

Differential item functioning

Differential item functioning occurs when respondents with the same ability have different probabilities of succeeding on a given item. This difference in probabilities may be caused by a shift in the discrimination parameter (a-DIF), the difficulty parameter (b-DIF), or both (ab-DIF). We begin with the most general case of ab-DIF.

▷ Example 2: Nonuniform DIF (ab-DIF)

Suppose we suspect item `q4` behaves differently across groups and want to test for a shift in the a and b parameters. We let item `q4` parameters vary across groups, and we keep the coefficients on the remaining items constrained to be the same across the groups. By specifying `(0: 2pl q4) (1: 2pl q4)`, we tell `irt` to estimate separate discrimination and difficulty parameters for item `q4` for each group.

```
. irt (0: 2pl q4) (1: 2pl q4) (2pl q1 q2 q3 q5), group(female)
Fitting fixed-effects model:
Iteration 0:   log likelihood = -4588.2619
Iteration 1:   log likelihood = -4584.1505
Iteration 2:   log likelihood =  -4584.149
Iteration 3:   log likelihood =  -4584.149
Group: Male
Group: Female
Fitting full model:
Iteration 0:   log likelihood = -4538.2523  (not concave)
Iteration 1:   log likelihood = -4484.1177
Iteration 2:   log likelihood = -4469.9992
Iteration 3:   log likelihood = -4469.5296
Iteration 4:   log likelihood = -4469.5261
Iteration 5:   log likelihood = -4469.5261
Hybrid IRT model                                  Number of obs = 1,500
Log likelihood = -4469.5261
Group: Male
```

	Coefficient	Std. err.	z	P>|z|	[95% conf.	interval]
2pl						
q4						
Discrim	.6144966	.128936	4.77	0.000	.3617867	.8672065
Diff	.2554018	.1365998	1.87	0.062	-.0123289	.5231326
2pl						
q1						
Discrim	1.306988	.1969137	6.64	0.000	.9210443	1.692932
Diff	-.4777569	.0820373	-5.82	0.000	-.6385471	-.3169668
q2						
Discrim	.9316811	.1344814	6.93	0.000	.6681023	1.19526
Diff	-.0097118	.0749322	-0.13	0.897	-.1565762	.1371525
q3						
Discrim	.930189	.1493857	6.23	0.000	.6373984	1.22298
Diff	-1.587128	.2173969	-7.30	0.000	-2.013218	-1.161038
q5						
Discrim	1.445573	.2228226	6.49	0.000	1.008848	1.882297
Diff	1.208255	.1453832	8.31	0.000	.9233095	1.493201
mean(Theta)	0	(omitted)				
var(Theta)	1	(constrained)				

Group: Female

	Coefficient	Std. err.	z	P>|z|	[95% conf. interval]	
2pl						
q4						
Discrim	1.354599	.3649997	3.71	0.000	.6392133	2.069986
Diff	.3935696	.1292876	3.04	0.002	.1401707	.6469686
2pl						
q1						
Discrim	1.306988	.1969137	6.64	0.000	.9210443	1.692932
Diff	-.4777569	.0820373	-5.82	0.000	-.6385471	-.3169668
q2						
Discrim	.9316811	.1344814	6.93	0.000	.6681023	1.19526
Diff	-.0097118	.0749322	-0.13	0.897	-.1565762	.1371525
q3						
Discrim	.930189	.1493857	6.23	0.000	.6373984	1.22298
Diff	-1.587128	.2173969	-7.30	0.000	-2.013218	-1.161038
q5						
Discrim	1.445573	.2228226	6.49	0.000	1.008848	1.882297
Diff	1.208255	.1453832	8.31	0.000	.9233095	1.493201
mean(Theta)	-.0628546	.0706378	-0.89	0.374	-.201302	.0755929
var(Theta)	.5030102	.1159462			.3201628	.7902832

We use `estat greport` to arrange the output in a more compact and readable format.

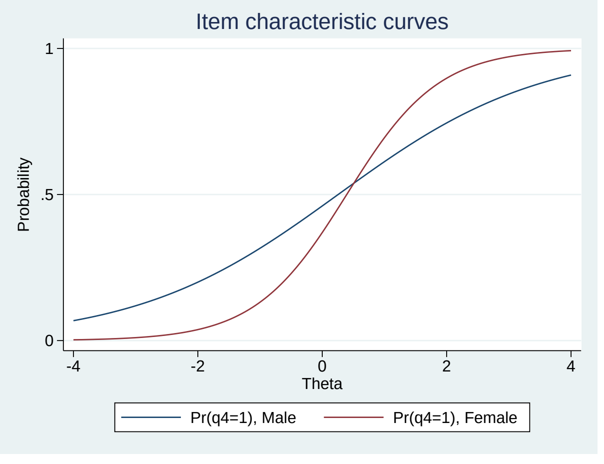
```
. estat greport
```

Parameter	Male	Female
q4		
Discrim	.6144966	1.3545995
Diff	.25540185	.39356964
q1		
Discrim	1.306988	1.306988
Diff	-.47775692	-.47775692
q2		
Discrim	.93168108	.93168108
Diff	-.00971184	-.00971184
q3		
Discrim	.93018903	.93018903
Diff	-1.5871277	-1.5871277
q5		
Discrim	1.4455726	1.4455726
Diff	1.2082554	1.2082554
mean(Theta)	0	-.06285455
var(Theta)	1	.50301021

Now it is easy to see that both the discrimination and difficulty parameters for item `q4` differ between groups.

We plot the ICCs for item `q4` for both groups using `irtgraph icc`; see [IRT] **irtgraph icc** for details.

```
. irtgraph icc q4
```



The graph suggests item `q4` exhibits a significant amount of DIF. We confirm it formally using a likelihood-ratio test; see [R] **lrtest** for details.

```
. lrtest nodif .
Likelihood-ratio test
Assumption: nodif nested within .

 LR chi2(2) =  13.64
Prob > chi2 = 0.0011
```

We reject the null hypothesis that the discrimination and difficulty parameters are the same across the groups and conclude that item q4 exhibits a-DIF, b-DIF, or ab-DIF.

◁

▷ Example 3: Uniform DIF (b-DIF)

Suppose we wish to test whether a model that allows the difficulty of q4 to differ across groups fits better than a model where the difficulty is constrained.

We impose the required equality constraint on the discrimination, the a parameter, for item q4 using the cns() option. The constraints are on the discrimination, the a parameter. The @k is a symbolic constraint that tells irt that parameters adorned with the same symbol should be constrained to be equal; see [IRT] **irt constraints** for details.

```
. irt (0: 2pl q4, cns(a@k)) (1: 2pl q4, cns(a@k))
>     (2pl q1 q2 q3 q5), group(female)
```

(output omitted)

```
. estat greport
```

Parameter	Male	Female
q4		
Discrim	.78202733	.78202733
Diff	.20904803	.64854407
q1		
Discrim	1.2249413	1.2249413
Diff	-.50829857	-.50829857
q2		
Discrim	.91473677	.91473677
Diff	-.01287912	-.01287912
q3		
Discrim	.89618684	.89618684
Diff	-1.651863	-1.651863
q5		
Discrim	1.4209752	1.4209752
Diff	1.2393731	1.2393731
mean(Theta)	0	-.07076921
var(Theta)	1	.62679028

We see that in this model the estimated discrimination parameter for item q4 is the same between groups and the estimated difficulty parameter differs.

We compare this model with the fully constrained model from example 1 and conclude that the current model is preferable.

```
. lrtest nodif .

Likelihood-ratio test
Assumption: nodif nested within .

 LR chi2(1) =   8.21
Prob > chi2 = 0.0042
```

◁

Reference

De Boeck, P., and M. Wilson, ed. 2004. *Explanatory Item Response Models: A Generalized Linear and Nonlinear Approach*. New York: Springer.

Also see

[IRT] **irt, group( ) postestimation** — Postestimation tools for group IRT

[IRT] **DIF** — Introduction to differential item functioning

[IRT] **irt** — Introduction to IRT models

[IRT] **irt constraints** — Specifying constraints

[SEM] **gsem** — Generalized structural equation model estimation command

[SVY] **svy estimation** — Estimation commands for survey data

[U] 20 Estimation and postestimation commands