Title

> **irt grm —** Graded response model

Description	Quick start	Menu	Syntax
Options	Remarks and examples	Stored results	Methods and formulas
References	Also see		

Description

irt grm fits graded response models (GRMs) to ordinal items. In the GRM, items vary in their difficulty and discrimination. This model is an extension of the 2PL model to ordered categorical items.

Quick start

GRM for ordinal items o1 to o5

    irt grm o1-o5

Report items sorted by discrimination

    estat report, sort(a)

Plot CCCs for o1

    irtgraph icc o1

Menu

Statistics > IRT (item response theory)

Syntax

> irt grm *varlist* [*if*] [*in*] [*weight*] [, *options*]

options	Description
group(*varname*)	fit model for different groups
Model	
cns(*spec*)	apply specified parameter constraints
listwise	drop observations with any missing items
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

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

bootstrap, by, collect, jackknife, statsby, and svy are allowed; see [U] 11.1.10 Prefix commands.

Weights are not allowed with the bootstrap prefix; see [R] bootstrap.

vce() and weights are not allowed with the svy prefix; see [SVY] svy.

fweights, iweights, and pweights are allowed; see [U] 11.1.6 weight.

startvalues(), noestimate, estmetric, dnumerical, and coeflegend do not appear in the dialog box.

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

Options

group(*varname*) specifies that the model be fit separately for the different values of *varname*; see [IRT] **irt, group( )** for details.

―――― 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.

―――― 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
   Video example

Overview

The following discussion is about how to use irt to fit GRMs to ordinal items. If you are new to
the IRT features in Stata, we encourage you to read [IRT] **irt** first.

In the GRM, item responses are categorical and ordered, for example, "poor", "good", and "excellent"
or "strongly disagree", "disagree", "agree", and "strongly agree". If there are only two outcomes,
the GRM is equivalent to the 2PL model; see [IRT] **irt 2pl**. If the item responses are not ordered, see
[IRT] **irt nrm**.

The GRM allows the ordered categories to vary between items; however, to keep the following
discussion from being overly complicated, we will assume the outcome levels for all items are given
by $k = 0, 1, \ldots, K$.

In the GRM, each item is modeled with its own discrimination parameter and cutpoints that identify boundaries between the ordered outcomes. The probability of observing outcome k or higher for item i and person j is given by

$$\Pr(Y_{ij} \geq k | \theta_j) = \frac{\exp\{a_i(\theta_j - b_{ik})\}}{1 + \exp\{a_i(\theta_j - b_{ik})\}} \qquad \theta_j \sim N(0,1)$$

where a_i represents the discrimination of item i, b_{ik} is the kth cutpoint for item i, and θ_j is the latent trait of person j. The cutpoint b_{ik} can be considered the difficulty of responding with category k or higher for item i.

The GRM is defined in terms of cumulative probabilities, but we can calculate the probability of observing outcome k as

$$\Pr(Y_{ij} = k | \theta_j) = \Pr(Y_{ij} \geq k | \theta_j) - \Pr(Y_{ij} \geq k+1 | \theta_j)$$

where we take $\Pr(Y_{ij} \geq 0) = 1$ and $\Pr(Y_{ij} > K) = 0$. Because of the additional calculation step required to obtain the probability of observing a particular outcome, the GRM is an indirect IRT model, also known as a difference model; see Thissen and Steinberg (1986).

The GRM was proposed by Samejima (1969). In the multilevel literature, the GRM is known as the cumulative logit model; see [ME] **meologit**. When no latent variable is present, the model for a single item is known as the proportional odds model; see [R] **ologit**.

▷ Example 1: Fitting a GRM

To illustrate the GRM, we use the data from Zheng and Rabe-Hesketh (2007). `charity.dta` contains five survey questions, `ta1` through `ta5`, measuring faith and trust in charity organizations. Responses are strongly agree (0), agree (1), disagree (2), and strongly disagree (3). Higher scores indicate higher levels of distrust. Here we list the first five observations.

```
. use https://www.stata-press.com/data/r18/charity
(Data from Zheng & Rabe-Hesketh (2007))

. list in 1/5, nolabel
```

	ta1	ta2	ta3	ta4	ta5
1.	.	2	1	1	.
2.	0	0	0	0	0
3.	1	1	2	0	2
4.	1	2	2	0	1
5.	.	1	1	1	1

Looking across the first row, we see that the first respondent did not provide an answer to items `ta1` and `ta5`, answered 2 on item `ta2`, and answered 1 on items `ta3` and `ta4`. All `irt` commands exclude missing items for a given observation from the likelihood calculation but keep the nonmissing items for that observation. If you wish to remove the entire observation from the model, add the `listwise` option at estimation time.

We fit a GRM as follows:

```
. irt grm ta1-ta5

Fitting fixed-effects model:

Iteration 0:  Log likelihood = -5559.6414
Iteration 1:  Log likelihood = -5473.9434
Iteration 2:  Log likelihood = -5467.4082
Iteration 3:  Log likelihood = -5467.3926
Iteration 4:  Log likelihood = -5467.3926

Fitting full model:

Iteration 0:  Log likelihood = -5271.0634
Iteration 1:  Log likelihood = -5162.5917
Iteration 2:  Log likelihood = -5159.2947
Iteration 3:  Log likelihood = -5159.2791
Iteration 4:  Log likelihood = -5159.2791
```

Graded response model   Number of obs = 945
Log likelihood = -5159.2791

	Coefficient	Std. err.	z	P>|z|	[95% conf. interval]	
ta1						
Discrim	.907542	.0955772	9.50	0.000	.7202142	1.09487
Diff						
>=1	-1.540098	.1639425			-1.861419	-1.218776
>=2	1.296135	.1427535			1.016343	1.575927
=3	3.305059	.3248468			2.668371	3.941747
ta2						
Discrim	.9434675	.0967483	9.75	0.000	.7538444	1.133091
Diff						
>=1	-1.661331	.167878			-1.990366	-1.332296
>=2	.0068314	.082222			-.1543208	.1679836
=3	2.531091	.2412513			2.058247	3.003935
ta3						
Discrim	1.734201	.1554383	11.16	0.000	1.429548	2.038855
Diff						
>=1	-1.080079	.0835119			-1.243759	-.9163983
>=2	1.016567	.0796635			.8604297	1.172705
=3	2.232606	.1497814			1.93904	2.526172
ta4						
Discrim	1.93344	.1857629	10.41	0.000	1.569351	2.297528
Diff						
>=1	-.3445057	.0578468			-.4578833	-.2311282
>=2	1.466254	.0983823			1.273428	1.65908
=3	2.418954	.162392			2.100672	2.737237
ta5						
Discrim	1.42753	.1263962	11.29	0.000	1.179798	1.675262
Diff						
>=1	-.8552358	.0833158			-1.018532	-.6919399
>=2	.6805315	.07469			.5341418	.8269211
=3	2.074243	.1538858			1.772632	2.375853

Because the GRM is basically an ordered logistic model, each item's difficulty parameters are naturally estimated in an increasing order. The difficulties represent a point at which a person with trait level $\theta_j = b_{ik}$ has a 50% chance of responding in category k or higher. We make cumulative comparisons because the model is defined in terms of cumulative probabilities.

For example, looking at the estimated parameters of item `ta1`, we see that a person with $\theta = -1.54$ has a 50% chance of answering 0 versus greater than or equal to 1, a person with $\theta = 1.30$ has a 50% chance of answering 0 or 1 versus greater than or equal to 2, and a person with $\theta = 3.31$ has a 50% chance of answering 0, 1, or 2 versus 3.

To illustrate this, we plot the BCCs as a function of θ for `ta1` using the estimated GRM parameters. The `blocation` option adds a vertical line at the estimated difficulties; see [IRT] **irtgraph icc**.

```
. irtgraph icc ta1, blocation
```
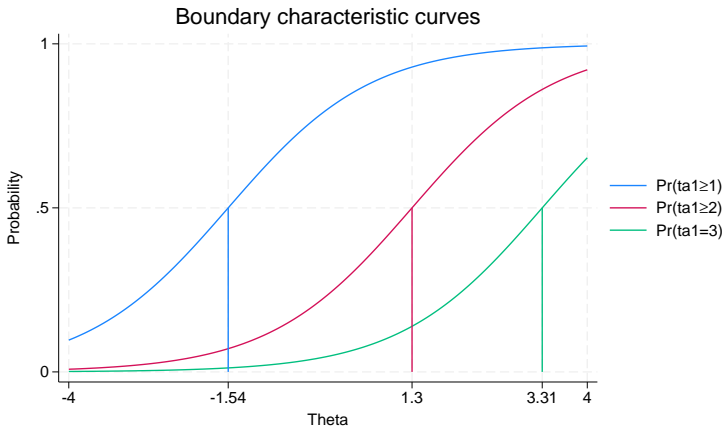

We see that the estimated difficulty parameters correspond to the point on the latent trait scale at which $\Pr(Y \geq k|\theta) = 0.5$. You can think of these curves as item characteristic curves where each curve dichotomizes the ordered responses into successive $\Pr(Y \geq k)$ and $\Pr(Y < k)$ categories. The estimated discrimination parameter for `ta1` is 0.91; thus, the curves have relatively flat slopes.

We can also plot category probabilities, $\Pr(Y = k)$, as a function of θ, which in fact is the default behavior of `irtgraph icc`. For categorical responses, such plots are called category characteristic curves (CCCs). Here we plot the CCCs for item `ta1`.

```
. irtgraph icc ta1, xlabel(-4 -1.35 1.55 2.9 4)
```
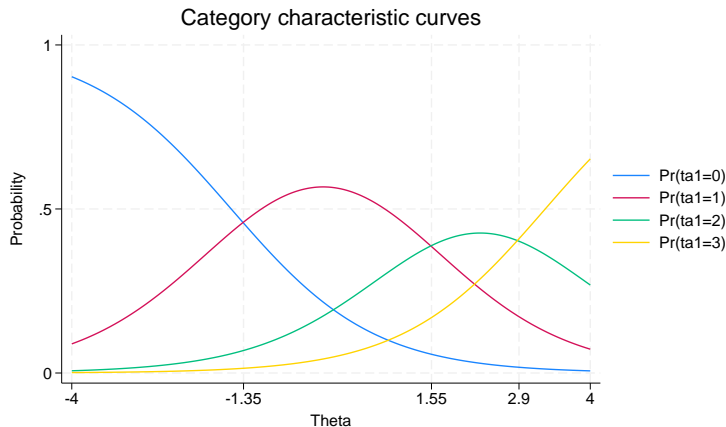

The graph shows that respondents with the latent trait level below approximately -1.35 are most likely to respond in the first category, respondents with the latent trait level between approximately -1.35 and 1.55 are most likely to respond in the second category, and so on.

We use irtgraph tcc to plot the TCC using the estimated GRM parameters; see [IRT] **irtgraph tcc**. Because we have 5 items, each coded 0 to 3, the total score ranges from 0 to 15. The thetalines() option plots the expected scores at the specified values for θ.

```
. irtgraph tcc, thetalines(-1.96 0 1.96)
```
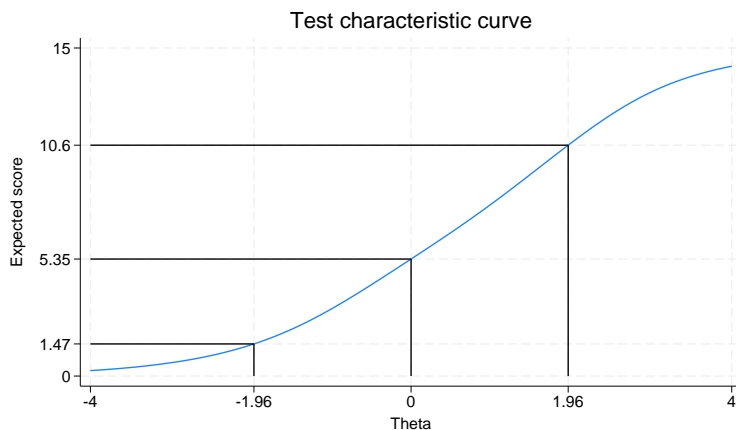

This plot tells us what kind of scores we can expect from individuals with different levels of the latent trait (trust in charities).

For example, we can expect above-average individuals to score 5.35 or above. Actually, no one is expected to score exactly 5.35 on this survey, so a more realistic statement is that we expect above-average individuals to score above 5 out of a possible score of 15.

Using the 95% critical values from the standard normal distribution (-1.96 and 1.96), this plot also tells us that we can expect 95% of randomly selected people to score between 1.47 and 10.6.

Again, a more realistic statement is that we expect about 95% of randomly selected people to score from 2 to 10, which can be interpreted that most people either trust or slightly distrust charities.

◁

Video example

[Item response theory using Stata: Graded response models (GRMs)](#)

Stored results

`irt grm` stores the following in `e()`:

Scalars
e(N)	number of observations
e(k)	number of parameters
e(k_eq)	number of equations in e(b)
e(k_dv)	number of dependent variables
e(k_rc)	number of covariances
e(k_rs)	number of variances
e(irt_k_eq)	number of IRT equations
e(k_items1)	number of items in first IRT equation
e(k_cat#)	number of categories for the #th item, ordinal
e(ll)	log likelihood
e(N_clust)	number of clusters
e(N_groups)	number of groups
e(n_quad)	number of integration points
e(rank)	rank of e(V)
e(ic)	number of iterations
e(rc)	return code
e(converged)	1 if target model converged, 0 otherwise

Macros
e(cmd)	gsem
e(cmd2)	irt
e(cmdline)	command as typed
e(model1)	grm
e(items1)	names of items in first IRT equation
e(depvar)	names of all item variables
e(wtype)	weight type
e(wexp)	weight expression
e(title)	title in estimation output
e(clustvar)	name of cluster variable
e(groupvar)	name of group variable
e(family#)	family for the #th *item*
e(link#)	link for the #th *item*
e(intmethod)	integration method
e(vce)	*vcetype* specified in vce()
e(vcetype)	title used to label Std. err.
e(opt)	type of optimization
e(which)	max or min; whether optimizer is to perform maximization or minimization
e(method)	estimation method: ml
e(ml_method)	type of ml method
e(user)	name of likelihood-evaluator program
e(technique)	maximization technique
e(datasignature)	the checksum
e(datasignaturevars)	variables used in calculation of checksum
e(properties)	b V
e(estat_cmd)	program used to implement estat
e(predict)	program used to implement predict

e(covariates)	list of covariates
e(footnote)	program used to implement the footnote display

Matrices
e(_N)	sample size for each item
e(b)	coefficient vector, slope-intercept parameterization
e(b_pclass)	parameter class
e(cat#)	categories for the #th item, ordinal
e(Cns)	constraints matrix
e(ilog)	iteration log (up to 20 iterations)
e(gradient)	gradient vector
e(V)	variance–covariance matrix of the estimators
e(V_modelbased)	model-based variance
e(groupvalue)	vector of group values in e(groupvar)
e(nobs)	vector with number of observations per group

Functions
e(sample)	marks estimation sample

In addition to the above, the following is stored in r():

Matrices
r(table)	matrix containing the coefficients with their standard errors, test statistics, p-values, and confidence intervals

Note that results stored in r() are updated when the command is replayed and will be replaced when any r-class command is run after the estimation command.

Methods and formulas

Let Y_{ij} represent the (yet to be observed) outcome for item i from person j. Without loss of generality, we will assume all items take on the ordered categories, $k = 0, 1, \ldots, K$.

Using the IRT parameterization, we see that the probability of person j with latent trait level θ_j (the latent trait) providing response k or above for item i is given by

$$\Pr(Y_{ij} \geq k | a_i, \mathbf{b}_i, \theta_j) = \frac{\exp\{a_i(\theta_j - b_{ik})\}}{1 + \exp\{a_i(\theta_j - b_{ik})\}}$$

where a_i represents the discrimination of item i, $\mathbf{b}_i = (b_{i1}, \ldots, b_{iK})$ represent the difficulties that distinguish the ordered categories of item i, and it is understood that $\Pr(Y_{ij} \geq 0 | a_i, \mathbf{b}_i, \theta_j) = 1$ and $\Pr(Y_{ij} > K | a_i, \mathbf{b}_i, \theta_j) = 0$. The probability of observing outcome k is then

$$\Pr(Y_{ij} = k | a_i, \mathbf{b}_i, \theta_j) = \Pr(Y_{ij} \geq k | a_i, \mathbf{b}_i, \theta_j) - \Pr(Y_{ij} \geq k+1 | a_i, \mathbf{b}_i, \theta_j)$$

irt grm fits the model using the slope-intercept form, so the probability for providing response k or above is parameterized as

$$\Pr(Y_{ij} \geq k | \alpha_i, \boldsymbol{\beta}_i, \theta_j) = \frac{\exp(\alpha_i \theta_j - \beta_{ik})}{1 + \exp(\alpha_i \theta_j - \beta_{ik})}$$

The transformation between these two parameterizations is

$$a_i = \alpha_i \qquad b_{ik} = \frac{\beta_{ik}}{\alpha_i}$$

Let y_{ij} be the observed response for Y_{ij} and $p_{ij} = \Pr(Y_{ij} = y_{ij} | \alpha_i, \boldsymbol{\beta}_i, \theta_j)$. Conditional on θ_j, the item responses are assumed to be independent, so the conditional density for person j is given by

$$f(\mathbf{y}_j | \boldsymbol{B}, \theta_j) = \prod_{i=1}^{I} p_{ij}$$

where $\mathbf{y}_j = (y_{1j}, \ldots, y_{Ij})$, $\boldsymbol{B} = (\alpha_1, \ldots, \alpha_I, \boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_I)$, and I is the number of items.

Missing items are skipped over in the above product by default. When the `listwise` option is specified, persons with any missing items are dropped from the estimation sample.

The likelihood for person j is computed by integrating out the latent variable from the joint density

$$L_j(\boldsymbol{B}) = \int_{-\infty}^{\infty} f(\mathbf{y}_j | \boldsymbol{B}, \theta_j) \, \phi(\theta_j) \, d\theta_j$$

where $\phi(\cdot)$ is the density function for the standard normal distribution. The log likelihood for the estimation sample is simply the sum of the log likelihoods from the N persons in the estimation sample.

$$\log L(\boldsymbol{B}) = \sum_{j=1}^{N} \log L_j(\boldsymbol{B})$$

The integral in the formula for $L_j(\boldsymbol{B})$ is generally not tractable, so we must use numerical methods.

Models for multiple groups, Gauss–Hermite quadrature, and adaptive quadrature are documented in *Methods and formulas* of [IRT] **irt hybrid**.

References

Samejima, F. 1969. Estimation of latent ability using a response pattern of graded scores. *Psychometrika Monograph Supplement*, no. 17.

Thissen, D., and L. Steinberg. 1986. A taxonomy of item response models. *Psychometrika* 51: 567–577. https://doi.org/10.1007/BF02295596.

Zheng, X., and S. Rabe-Hesketh. 2007. Estimating parameters of dichotomous and ordinal item response models with gllamm. *Stata Journal* 7: 313–333.

Also see

For suggested citations, see the FAQ on citing Stata documentation.