# Title

stata.com

**irt 3pl —** Three-parameter logistic model



## Description

`irt 3pl` fits three-parameter logistic (3PL) models to binary items. In the 3PL model, items vary in their difficulty and discrimination and the possibility of guessing is allowed.

## Quick start

3PL model for binary items `b1` to `b10`

```
irt 3pl b1-b10
```

Group estimates by parameter type and sort items by difficulty

```
estat report, byparm sort(b)
```

Plot ICCs for all items

```
irtgraph icc
```

## Menu

Statistics > IRT (item response theory)

# Syntax

```
irt 3pl varlist [if] [in] [weight] [, options]
```

| *options* | Description |
|---|---|
| **Model** | |
| `listwise` | drop observations with any missing items |
| `sepguessing` | estimate a separate pseudoguessing parameter for each item |
| **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 |
| `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`, `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`, `dnumerical`, and `coeflegend` do not appear in the dialog box.
See **[U] 20 Estimation and postestimation commands** for more capabilities of estimation commands.

# Options

Model

`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 is a seldom used option; see the technical note below.

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 nonadaptive 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)
```

`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

The following discussion is about how to use `irt` to fit (3PL) models to binary items. If you are new to the IRT features in Stata, we encourage you to read [IRT] **irt** first.

In the 3PL model, item responses are typically of the form yes or no, correct or incorrect, agree or disagree, etc. Items are assumed to vary in discrimination and difficulty, and the model accommodates the possibility of guessing on a test. The probability of person $j$ providing a positive answer to item $i$ is given by

$$\Pr(y_{ij}=1|\theta_j) = c_i + (1-c_i)\frac{\exp\{a_i(\theta_j-b_i)\}}{1+\exp\{a_i(\theta_j-b_i)\}} \qquad \theta_j \sim N(0,1) \tag{1}$$

where $a_i$ represents the discrimination of item $i$, $b_i$ represents the difficulty of item $i$, $c_i$ represents the pseudoguessing parameter, and $\theta_j$ is the latent trait of person $j$. By default, the $c_i$ are constrained to be the same across all items; see the technical note below.

Although (1) is not in logistic form, the model is commonly referred to as a three-parameter logistic model.

The 3PL model was proposed by Birnbaum (1968). An earlier three-parameter model with a probit link was developed by Finney (1952).

## ❑ Technical note

By default, `irt 3pl` constrains the pseudoguessing parameter to be the same across all items. You can use the advanced option `sepguessing` to request a separate pseudoguessing parameter for each item. We do not recommend this option because this version of the 3PL model is plagued with identification problems; see, for example, Samejima (1973), Holland (1990), Yen, Burket, and Sykes (1991), Maris (2002), and San Martín, Rolin, and Castro (2013).

The `sepguessing` option can be useful in the context of hybrid IRT models, where separate pseudoguessing parameters can be estimated for a subset of items; see example 2 in [IRT] **irt hybrid**.

❑

## ▷ Example 1: Fitting a 3PL model

To illustrate the 3PL 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. Here we list the first five observations.

```
. use http://www.stata-press.com/data/r14/masc1
(Data from De Boeck & Wilson (2004))
. list in 1/5
```

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

Looking across the rows, we see that the first student correctly answered items `q1`, `q2`, `q3`, and `q8`, the second student correctly answered items `q3`, `q8`, and `q9`, and so on.

We fit a 3PL model to binary items q1–q9 as follows:

```
. irt 3pl q1-q9
Fitting fixed-effects model:
Iteration 0:   log likelihood = -5322.8824
Iteration 1:   log likelihood = -4317.9868
Iteration 2:   log likelihood = -4273.6659
Iteration 3:   log likelihood = -4269.7862
Iteration 4:   log likelihood = -4269.7825
Iteration 5:   log likelihood = -4269.7825
Fitting full model:
Iteration 0:   log likelihood = -4226.5553  (not concave)
Iteration 1:   log likelihood = -4127.0541
Iteration 2:   log likelihood = -4122.9138
Iteration 3:   log likelihood = -4116.4384
Iteration 4:   log likelihood = -4116.3432
Iteration 5:   log likelihood = -4116.3404
Iteration 6:   log likelihood = -4116.3404
Three-parameter logistic model                  Number of obs     =        800
Log likelihood = -4116.3404
```

| | Coef. | Std. Err. | z | P>\|z\| | [95% Conf. | Interval] |
|---|---|---|---|---|---|---|
| q1 | | | | | | |
| Discrim | 1.911892 | .3633522 | 5.26 | 0.000 | 1.199735 | 2.624049 |
| Diff | -.3040615 | .0970829 | -3.13 | 0.002 | -.4943405 | -.1137825 |
| q2 | | | | | | |
| Discrim | .7508883 | .1414087 | 5.31 | 0.000 | .4737323 | 1.028044 |
| Diff | .1506359 | .1667865 | 0.90 | 0.366 | -.1762597 | .4775314 |
| q3 | | | | | | |
| Discrim | .9674961 | .1682051 | 5.75 | 0.000 | .6378203 | 1.297172 |
| Diff | -1.508913 | .2358777 | -6.40 | 0.000 | -1.971225 | -1.046602 |
| q4 | | | | | | |
| Discrim | .9846873 | .1860973 | 5.29 | 0.000 | .6199432 | 1.349431 |
| Diff | .5726213 | .149159 | 3.84 | 0.000 | .280275 | .8649676 |
| q5 | | | | | | |
| Discrim | 1.439627 | .4426275 | 3.25 | 0.001 | .5720926 | 2.307161 |
| Diff | 1.605677 | .2144336 | 7.49 | 0.000 | 1.185395 | 2.025959 |
| q6 | | | | | | |
| Discrim | 1.369117 | .3249624 | 4.21 | 0.000 | .7322022 | 2.006031 |
| Diff | .7818606 | .1236339 | 6.32 | 0.000 | .5395425 | 1.024179 |
| q7 | | | | | | |
| Discrim | .4823125 | .1727574 | 2.79 | 0.005 | .1437144 | .8209107 |
| Diff | 3.010921 | .8924984 | 3.37 | 0.001 | 1.261657 | 4.760186 |
| q8 | | | | | | |
| Discrim | 1.436068 | .2482749 | 5.78 | 0.000 | .9494584 | 1.922678 |
| Diff | -1.594748 | .1918751 | -8.31 | 0.000 | -1.970817 | -1.21868 |
| q9 | | | | | | |
| Discrim | .6772548 | .1314524 | 5.15 | 0.000 | .4196127 | .9348968 |
| Diff | -1.213935 | .2661821 | -4.56 | 0.000 | -1.735642 | -.6922275 |
| Guess | .0904467 | .0359679 | 2.51 | 0.012 | .0199508 | .1609425 |

In the 3PL model, each test item has its own parameter estimates for discrimination and difficulty. The estimated common pseudoguessing parameter is reported at the end of the table.

In the following, we use `estat report` to replay the table of estimated IRT parameters and control how the output is reported. We include the `byparm` option, which arranges the output by parameter rather than by item, and the `sort(b)` option, which displays the items in an ascending order of difficulty. This makes it easy to see that item `q8` is least difficult and item `q7` is most difficult.

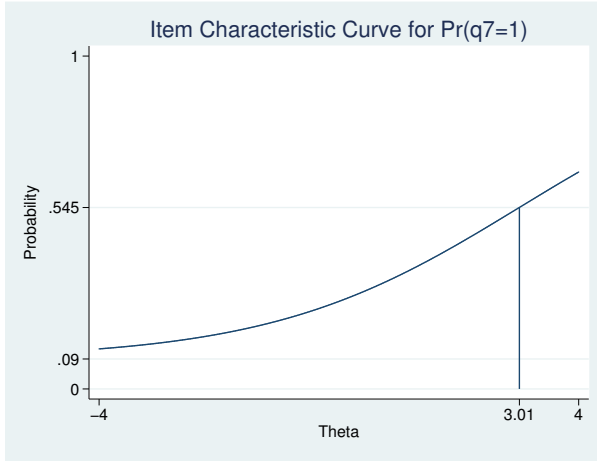
```
. estat report, byparm sort(b)
Three-parameter logistic model                  Number of obs     =        800
Log likelihood = -4116.3404
```

| | Coef. | Std. Err. | z | P>\|z\| | [95% Conf. | Interval] |
|---|---|---|---|---|---|---|
| Discrim | | | | | | |
| q8 | 1.436068 | .2482749 | 5.78 | 0.000 | .9494584 | 1.922678 |
| q3 | .9674961 | .1682051 | 5.75 | 0.000 | .6378203 | 1.297172 |
| q9 | .6772548 | .1314524 | 5.15 | 0.000 | .4196127 | .9348968 |
| q1 | 1.911892 | .3633522 | 5.26 | 0.000 | 1.199735 | 2.624049 |
| q2 | .7508883 | .1414087 | 5.31 | 0.000 | .4737323 | 1.028044 |
| q4 | .9846873 | .1860973 | 5.29 | 0.000 | .6199432 | 1.349431 |
| q6 | 1.369117 | .3249624 | 4.21 | 0.000 | .7322022 | 2.006031 |
| q5 | 1.439627 | .4426275 | 3.25 | 0.001 | .5720926 | 2.307161 |
| q7 | .4823125 | .1727574 | 2.79 | 0.005 | .1437144 | .8209107 |
| Diff | | | | | | |
| q8 | -1.594748 | .1918751 | -8.31 | 0.000 | -1.970817 | -1.21868 |
| q3 | -1.508913 | .2358777 | -6.40 | 0.000 | -1.971225 | -1.046602 |
| q9 | -1.213935 | .2661821 | -4.56 | 0.000 | -1.735642 | -.6922275 |
| q1 | -.3040615 | .0970829 | -3.13 | 0.002 | -.4943405 | -.1137825 |
| q2 | .1506359 | .1667865 | 0.90 | 0.366 | -.1762597 | .4775314 |
| q4 | .5726213 | .149159 | 3.84 | 0.000 | .280275 | .8649676 |
| q6 | .7818606 | .1236339 | 6.32 | 0.000 | .5395425 | 1.024179 |
| q5 | 1.605677 | .2144336 | 7.49 | 0.000 | 1.185395 | 2.025959 |
| q7 | 3.010921 | .8924984 | 3.37 | 0.001 | 1.261657 | 4.760186 |
| Guess | .0904467 | .0359679 | 2.51 | 0.012 | .0199508 | .1609425 |

The estimate of the pseudoguessing parameter is 0.09, which suggests a modest degree of guessing on the test. The pseudoguessing parameter represents the smallest probability of a correct response. Thus, according to this model, even the least able student has, at minimum, a 9% chance of responding correctly on any given item.

After `irt 3pl`, we can use `irtgraph icc` to plot the ICCs using the estimated 3PL parameters; see [IRT] **irtgraph icc**. To focus on the most difficult item, as reported by `estat report`, we restrict the plot to item `q7`. We use option `blocation` to add a vertical line at the estimated difficulty and option `ylabel()` to change the default labeling of the $y$ axis to include the lower asymptote and the midpoint probability, where $\theta$ equals the estimated difficulty for `q7`.

```
. irtgraph icc q7, blocation ylabel(0 0.09 0.545 1)
```



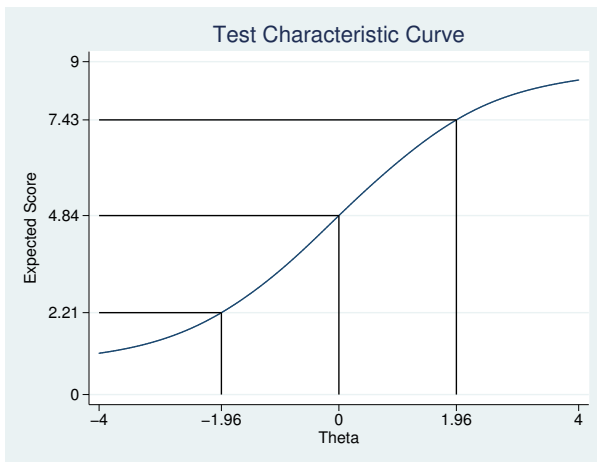

Notice that the estimate of the pseudoguessing parameter is now a lower asymptote for the plotted ICC. Also, because of the pseudoguessing parameter, the midpoint probability, where $\theta$ equals the estimated difficulty for `q7`, is

$$\widehat{c} + (1 - \widehat{c}) \times \frac{1}{2} = 0.09 + 0.91 \times \frac{1}{2} = 0.545$$

instead of 0.5, as in the case of 1PL and 2PL models.

The TCC plots the expected score as a function of $\theta$, using the estimated 3PL parameters. We use `irtgraph tcc` to plot the TCC. For 9 binary items, it is clear that the total score ranges from 0 to 9; however, because of the pseudoguessing parameter, the minimum expected score is $\widehat{c} \times 9 = 0.09 \times 9 = 0.81$. The `thetalines()` option plots the expected scores at the specified values for $\theta$.

```
. 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. For example, we can expect above-average individuals to score 4.84 or above. Actually, no one is expected to score 4.84 on a 9-item test, so a more realistic statement is that we expect above-average individuals to score above 4.

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 2.21 and 7.43. A more realistic statement is that we expect about 95% of randomly selected people to score from 2 to 7.

◁

## Stored results

`irt 3pl` 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 model groups |
| `e(k_items1)` | number of items in first IRT model group |
| `e(sepguess1)` | `1` if model contains a separate pseudoguessing parameter |
| `e(ll)` | log likelihood |
| `e(N_clust)` | number of clusters |
| `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)` | `3pl` |
| `e(items1)` | names of items in first IRT model group |
| `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(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(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 |
| Functions | |
| e(sample) | marks estimation sample |

## Methods and formulas

Let $Y_{ij}$ represent the (yet to be observed) outcome for item $i$ from person $j$, and let $y_{ij}$ be the observed value of $Y_{ij}$. Without loss of generality, we will use the terms "correct" and "incorrect" in reference to the outcomes of $Y_{ij}$. Furthermore, we will refer to $y_{ij} = 1$ as correct and $y_{ij} = 0$ as incorrect.

Using the IRT parameterization, we see that the probability of person $j$ with latent trait level $\theta_j$ (the latent trait) providing a correct response to item $i$ is given by

$$\Pr(Y_{ij} = 1|a_i, b_i, c_i, \theta_j) = c_i + (1 - c_i)\frac{\exp\{a_i(\theta_j - b_i)\}}{1 + \exp\{a_i(\theta_j - b_i)\}}$$

where $a_i$ represents the discrimination of item $i$, $b_i$ represents the difficulty of item $i$, and $c_i$ represents the pseudoguessing parameter. `irt 3pl` fits the model using the slope-intercept form, so the probability for providing a correct answer is parameterized as

$$\Pr(Y_{ij} = 1|\alpha_i, \beta_i, \gamma_i, \theta_j) = \frac{\exp(\gamma_i)}{1 + \exp(\gamma_i)} + \frac{1}{1 + \exp(\gamma_i)}\frac{\exp(\alpha_i\theta_j + \beta_i)}{1 + \exp(\alpha_i\theta_j + \beta_i)}$$

The transformation between these two parameterizations is

$$a_i = \alpha_i \qquad b_i = -\frac{\beta_i}{\alpha_i} \qquad c_i = \frac{\exp(\gamma_i)}{1 + \exp(\gamma_i)}$$

By default, the $\gamma_i$ (and thus the $c_i$) are constrained to be the same across all items.

Let $p_{ij} \equiv \Pr(Y_{ij} = 1|\alpha_i, \beta_i, \gamma_i, \theta_j)$ and $q_{ij} = 1 - p_{ij}$. Conditional on $\theta_j$, the item responses are assumed to be independent, so the conditional density for person $j$ is given by

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

where $\mathbf{y}_j = (y_{1j}, \ldots, y_{Ij})$, $\boldsymbol{B} = (\alpha_1, \ldots, \alpha_I, \beta_1, \ldots, \beta_I, \gamma_1, \ldots, \gamma_I)$, 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.

Gauss–Hermite quadrature and adaptive quadrature are documented in *Methods and formulas* of [IRT] **irt hybrid**.

## References


Birnbaum, A. 1968. Some latent trait models and their use in inferring an examinee's ability. In *Statistical Theories of Mental Test Scores*, ed. F. M. Lord and M. R. Novick, 395–479. Reading, MA: Addison–Wesley.

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

Finney, D. J. 1952. *Probit Analysis: A Statistical Treatment of the Sigmoid Response Curve*. 2nd ed. New York: Cambridge University Press.

Holland, P. W. 1990. The Dutch identity: A new tool for the study of item response models. *Psychometrika* 55: 5–18.

Maris, G. 2002. Concerning the identification of the 3PL model. Technical Report 2002-3, CITO National Institute for Educational Measurement, Arnhem, The Netherlands.

Samejima, F. 1973. A comment on Birnbaum's three-parameter logistic model in the latent trait theory. *Psychometrika* 38: 221–233.

San Martín, E., J.-M. Rolin, and L. M. Castro. 2013. Identification of the 1PL model with guessing parameter: Parametric and semi-parametric results. *Psychometrika* 78: 341–379.

Yen, W. M., G. R. Burket, and R. C. Sykes. 1991. Nonunique solutions to the likelihood equation for the three-parameter logistic model. *Psychometrika* 56: 39–54.


## Also see

[IRT] **irt 3pl postestimation** — Postestimation tools for irt 3pl

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

[IRT] **irt 1pl** — One-parameter logistic model

[IRT] **irt 2pl** — Two-parameter logistic model

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

[U] **20 Estimation and postestimation commands**