

[Description](#)
[Options](#)
[References](#)[Quick start](#)
[Remarks and examples](#)
[Also see](#)[Menu](#)
[Stored results](#)[Syntax](#)
[Methods and formulas](#)

Description

`scobit` fits a maximum-likelihood skewed logit model.

Quick start

Skewed logistic regression of binary variable `y` on `x1` and `x2`

```
scobit y x1 x2
```

With robust standard errors

```
scobit y x1 x2, vce(robust)
```

Same as above, and display coefficients and std. err. with two digits to the right of the decimal

```
scobit y x1 x2, vce(robust) cformat(%8.2f)
```

Same as above, and also display *p*-values with two digits to the right of the decimal

```
scobit y x1 x2, vce(robust) cformat(%8.2f) pformat(%5.2f)
```

Menu

Statistics > Binary outcomes > Skewed logistic regression

Syntax

```
scobit depvar [indepvars] [if] [in] [weight] [, options]
```

<i>options</i>	Description
Model	
<code>noconstant</code>	suppress constant term
<code>offset(<i>varname</i>)</code>	include <i>varname</i> in model with coefficient constrained to 1
<code>asis</code>	retain perfect predictor variables
<code>constraints(<i>constraints</i>)</code>	apply specified linear constraints
SE/Robust	
<code>vce(<i>vcetype</i>)</code>	<i>vcetype</i> may be <code>oim</code> , <code>robust</code> , <code>cluster <i>clustvar</i></code> , <code>opg</code> , <code>bootstrap</code> , or <code>jackknife</code>
Reporting	
<code>level(#)</code>	set confidence level; default is <code>level(95)</code>
<code>nocnsreport</code>	do not display constraints
<code>display_options</code>	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
Maximization	
<code>maximize_options</code>	control the maximization process
<code>collinear</code>	keep collinear variables
<code>coeflegend</code>	display legend instead of statistics

indepvars may contain factor variables; see [U] 11.4.3 Factor variables.

`bayesboot`, `bootstrap`, `by`, `collect`, `fp`, `jackknife`, `nestreg`, `rolling`, `statsby`, `stepwise`, 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`.

`collinear` and `coeflegend` do not appear in the dialog box.

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

Options

Model
<code>noconstant</code> , <code>offset(<i>varname</i>)</code> , <code>constraints(<i>constraints</i>)</code> ; see [R] Estimation options.
<code>asis</code> forces retention of perfect predictor variables and their associated perfectly predicted observations and may produce instabilities in maximization; see [R] <code>probit</code> .
SE/Robust
<code>vce(<i>vcetype</i>)</code> specifies the type of standard error reported, which includes types that are derived from asymptotic theory (<code>oim</code> , <code>opg</code>), that are robust to some kinds of misspecification (<code>robust</code>), that allow for intragroup correlation (<code>cluster <i>clustvar</i></code>), and that use bootstrap or jackknife methods (<code>bootstrap</code> , <code>jackknife</code>); see [R] <code>vce_option</code> .

Reporting

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

`nocnsreport`; see [R] [Estimation options](#).

display_options: `noci`, `nopvalues`, `noomitted`, `vsquish`, `noemptycells`, `baselevels`, `allbaselevels`, `nofvlabel`, `fvwrap(#)`, `fvwrapon(style)`, `cformat(%fmt)`, `pformat(%fmt)`, `sformat(%fmt)`, and `nolstretch`; see [R] [Estimation options](#).

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](#). These options are seldom used.

Setting the optimization type to `technique(bhhh)` resets the default *vcetype* to `vce(opg)`.

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

`collinear`, `coeflegend`; see [R] [Estimation options](#).

Remarks and examples

Remarks are presented under the following headings:

Skewed logistic model

Robust standard errors

Skewed logistic model

`scobit` fits maximum likelihood models with dichotomous dependent variables coded as 0/1 (or, more precisely, coded as 0 and not 0).

► Example 1

We have data on the make, weight, and mileage rating of 22 foreign and 52 domestic automobiles. We wish to fit a model explaining whether a car is foreign based on its mileage. Here is an overview of our data:

```
. use https://www.stata-press.com/data/r19/auto
(1978 automobile data)

. keep make mpg weight foreign
. describe
Contains data from https://www.stata-press.com/data/r19/auto.dta
Observations:      74      1978 automobile data
Variables:         4      13 Apr 2024 17:45
                        (_dta has notes)
```

Variable name	Storage type	Display format	Value label	Variable label
make	str18	%-18s		Make and model
mpg	int	%8.0g		Mileage (mpg)
weight	int	%8.0gc		Weight (lbs.)
foreign	byte	%8.0g	origin	Car origin

```
Sorted by: foreign
Note: Dataset has changed since last saved.
```

```
. inspect foreign
```

foreign: Car origin			Number of observations		
			Total	Integers	Nonintegers
#		Negative	-	-	-
#		Zero	52	52	-
#		Positive	22	22	-
#					
#	#	Total	74	74	-
#	#	Missing	-		
0		1	74		

(2 unique values)

foreign is labeled and all values are documented in the label.

The variable `foreign` takes on two unique values, 0 and 1. The value 0 denotes a domestic car, and 1 denotes a foreign car.

The model that we wish to fit is

$$\Pr(\text{foreign} = 1) = F(\beta_0 + \beta_1 \text{mpg})$$

where $F(z) = 1 - 1/\{1 + \exp(z)\}^\alpha$.

To fit this model, we type

```
. scobit foreign mpg
Fitting logistic model:
Iteration 0: Log likelihood = -45.03321
Iteration 1: Log likelihood = -39.380959
Iteration 2: Log likelihood = -39.288802
Iteration 3: Log likelihood = -39.28864
Iteration 4: Log likelihood = -39.28864
Fitting full model:
Iteration 0: Log likelihood = -39.28864
Iteration 1: Log likelihood = -39.286393
Iteration 2: Log likelihood = -39.284415
Iteration 3: Log likelihood = -39.284234
Iteration 4: Log likelihood = -39.284197
Iteration 5: Log likelihood = -39.284196
```

```
Skewed logistic regression          Number of obs   =       74
                                   Zero outcomes      =       52
Log likelihood = -39.2842          Nonzero outcomes =       22
```

foreign	Coefficient	Std. err.	z	P> z	[95% conf. interval]	
mpg	.1813879	.2407362	0.75	0.451	-.2904463	.6532222
_cons	-4.274883	1.399305	-3.06	0.002	-7.017471	-1.532295
/lnalpha	-.4450405	3.879885	-0.11	0.909	-8.049476	7.159395
alpha	.6407983	2.486224			.0003193	1286.133

```
LR test of alpha=1: chi2(1) = 0.01          Prob > chi2 = 0.9249
```

Note: Likelihood-ratio tests are recommended for inference with scobit models.

We find that cars yielding better gas mileage are less likely to be foreign. The likelihood-ratio test at the bottom of the output indicates that the model is not significantly different from a logit model. Therefore, we should use the more parsimonious model.



□ Technical note

Stata interprets a value of 0 as a negative outcome (failure) and treats all other values (except missing) as positive outcomes (successes). Thus if the dependent variable takes on the values 0 and 1, then 0 is interpreted as failure and 1 as success. If the dependent variable takes on the values 0, 1, and 2, then 0 is still interpreted as failure, but both 1 and 2 are treated as successes.

Formally, when we type `scobit y x`, Stata fits the model

$$\Pr(y_j \neq 0 \mid \mathbf{x}_j) = 1 - 1 / \left\{ 1 + \exp(\mathbf{x}_j \beta) \right\}^{\alpha}$$



Robust standard errors

If you specify the `vce(robust)` option, `scobit` reports robust standard errors as described in [U] 20.22 Obtaining robust variance estimates. For the model of `foreign` on `mpg`, the robust calculation increases the standard error of the coefficient on `mpg` by around 25%:

```
. scobit foreign mpg, vce(robust) nolog
Skewed logistic regression          Number of obs   =       74
                                   Zero outcomes      =       52
Log pseudolikelihood = -39.2842    Nonzero outcomes =       22
```

foreign	Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
mpg	.1813879	.3028487	0.60	0.549	-.4121847	.7749606
_cons	-4.274883	1.335521	-3.20	0.001	-6.892455	-1.657311
/lnalpha	-.4450405	4.71561	-0.09	0.925	-9.687466	8.797385
alpha	.6407983	3.021755			.0000621	6616.919

Without `vce(robust)`, the standard error for the coefficient on `mpg` was reported to be 0.241, with a resulting confidence interval of $[-0.29, 0.65]$.

Specifying the `vce(cluster clustvar)` option relaxes the independence assumption required by the skewed logit estimator to being just independence between clusters. To demonstrate this, we will switch to a different dataset.

► Example 2

We are studying the unionization of women in the United States and have a dataset with 26,200 observations on 4,434 women between 1970 and 1988. For our purposes, we will use the variables `age` (the women were 14–26 in 1968 and the data thus span the age range of 16–46), `grade` (years of schooling completed, ranging from 0 to 18), `not_smsa` (28% of the person-time was spent living outside an SMSA—standard metropolitan statistical area), `south` (41% of the person-time was in the South), and `year`. Each of these variables is included in the regression as a covariate along with the interaction between `south` and `year`. This interaction, along with the `south` and `year` variables, is specified in the `scobit` command using factor-variables notation, `south##c.year`. We also have variable `union`. Overall, 22% of the person-time is marked as time under union membership and 44% of these women have belonged to a union.

We fit the following model, ignoring that women are observed an average of 5.9 times each in these data:

```
. use https://www.stata-press.com/data/r19/union, clear
(NLS Women 14-24 in 1968)
. scobit union age grade not_smsa south##c.year, nrtol(1e-3)
(output omitted)
```

```
Skewed logistic regression      Number of obs   =    26,200
                                Zero outcomes      =    20,389
Log likelihood = -13540.61      Nonzero outcomes =     5,811
```

	union	Coefficient	Std. err.	z	P> z	[95% conf. interval]	
	age	.0185363	.0043615	4.25	0.000	.0099879	.0270848
	grade	.0452801	.0057124	7.93	0.000	.034084	.0564761
	not_smsa	-.1886826	.0317801	-5.94	0.000	-.2509705	-.1263947
	1.south	-1.422372	.3949301	-3.60	0.000	-2.196421	-.6483233
	year	-.0133016	.0049575	-2.68	0.007	-.0230181	-.0035851
	south#c.year						
	1	.0105663	.0049233	2.15	0.032	.0009167	.0202158
	_cons	-10.3557	68.9757	-0.15	0.881	-145.5456	124.8342
	/lnalpha	9.136018	68.97395	0.13	0.895	-126.0504	144.3225
	alpha	9283.719	640334.8			1.81e-55	4.77e+62

```
LR test of alpha=1: chi2(1) = 3.76      Prob > chi2 = 0.0524
```

```
Note: Likelihood-ratio tests are recommended for inference with scobit models.
```

The reported standard errors in this model are probably meaningless. Women are observed repeatedly, so the observations are not independent. Looking at the coefficients, we find a large southern effect against unionization and a different time trend for the south. The `vce(cluster clustvar)` option provides a way to fit this model and obtains correct standard errors:

```
. scobit union age grade not_smsa south##c.year, vce(cluster id) nrtol(1e-3)
(output omitted)
Skewed logistic regression                Number of obs    =    26,200
                                           Zero outcomes    =    20,389
Log pseudolikelihood = -13540.61          Nonzero outcomes =     5,811
                                           (Std. err. adjusted for 4,434 clusters in idcode)
```

union	Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
age	.0185363	.0084867	2.18	0.029	.0019027	.03517
grade	.0452801	.0125765	3.60	0.000	.0206306	.0699295
not_smsa	-.1886826	.0642037	-2.94	0.003	-.3145194	-.0628457
1.south	-1.422372	.5064933	-2.81	0.005	-2.415081	-.4296635
year	-.0133016	.0090622	-1.47	0.142	-.0310632	.0044599
south#c.year						
1	.0105663	.0063172	1.67	0.094	-.0018153	.0229478
_cons	-10.3557	.9416702	-11.00	0.000	-12.20134	-8.510064
/lnalpha	9.136018	.7432386	12.29	0.000	7.679297	10.59274
alpha	9283.719	6900.019			2163.098	39844.45

scobit, vce(cluster *clustvar*) is robust to assumptions about within-cluster correlation. That is, it inefficiently sums within cluster for the standard error calculation rather than attempting to exploit what might be assumed about the within-cluster correlation (as do the xtgee population-averaged models; see [XT] xtgee).

Technical note

The scobit model can be difficult to fit because of the functional form. Often, it requires many iterations, or the optimizer prints out warning and informative messages during the optimization. For example, without the nrtol(1e-3) option, the model using the union dataset will not converge. See [R] Maximize for details about the optimizer.

Technical note

The main reason for using scobit rather than logit is that the effects of the regressors on the probability of success are not constrained to be the largest when the probability is 0.5. Rather, the independent variables might show their largest impact when the probability of success is 0.3 or 0.6. This added flexibility results because the scobit function, unlike the logit function, can be skewed and is not constrained to be mirror symmetric about the 0.5 probability of success.

As Nagler (1994) pointed out, the point of maximum impact is constrained under the scobit model to fall within the interval $(0, 1 - e^{(-1)})$ or approximately $(0, 0.63)$. Achen (2002) notes that if we believe the maximum impact to be outside that range, we can instead fit the “power logit” model by simply reversing the 0s and 1s of our outcome variable and fitting a scobit model on failure, rather than success. We would need to reverse the signs of the coefficients if we wanted to interpret them in terms of impact on success,

or we could leave them as they are and interpret them in terms of impact on failure. The important thing to remember is that the scobit model, unlike the logit model, is not invariant to the choice of which result is assigned to success.



Stored results

scobit stores the following in `e()`:

Scalars

<code>e(N)</code>	number of observations
<code>e(k)</code>	number of parameters
<code>e(k_eq)</code>	number of equations in <code>e(b)</code>
<code>e(k_aux)</code>	number of auxiliary parameters
<code>e(k_dv)</code>	number of dependent variables
<code>e(ll)</code>	log likelihood
<code>e(ll_c)</code>	log likelihood, comparison model
<code>e(N_f)</code>	number of failures (zero outcomes)
<code>e(N_s)</code>	number of successes (nonzero outcomes)
<code>e(alpha)</code>	alpha
<code>e(N_clust)</code>	number of clusters
<code>e(chi2_c)</code>	χ^2 for comparison test
<code>e(rank)</code>	rank of <code>e(V)</code>
<code>e(ic)</code>	number of iterations
<code>e(rc)</code>	return code
<code>e(converged)</code>	1 if converged, 0 otherwise

Macros

<code>e(cmd)</code>	scobit
<code>e(cmdline)</code>	command as typed
<code>e(depvar)</code>	name of dependent variable
<code>e(wtype)</code>	weight type
<code>e(wexp)</code>	weight expression
<code>e(title)</code>	title in estimation output
<code>e(clustvar)</code>	name of cluster variable
<code>e(offset)</code>	linear offset variable
<code>e(chi2_ct)</code>	Wald or LR; type of model χ^2 test corresponding to <code>e(chi2_c)</code>
<code>e(vce)</code>	<i>vcetype</i> specified in <code>vce()</code>
<code>e(vctype)</code>	title used to label Std. err.
<code>e(opt)</code>	type of optimization
<code>e(which)</code>	max or min; whether optimizer is to perform maximization or minimization
<code>e(ml_method)</code>	type of ml method
<code>e(user)</code>	name of likelihood-evaluator program
<code>e(technique)</code>	maximization technique
<code>e(properties)</code>	b V
<code>e(predict)</code>	program used to implement predict
<code>e(footnote)</code>	program used to implement the footnote display
<code>e(asbalanced)</code>	factor variables <code>fvset</code> as asbalanced
<code>e(asobserved)</code>	factor variables <code>fvset</code> as asobserved

Matrices

<code>e(b)</code>	coefficient vector
<code>e(Cns)</code>	constraints matrix
<code>e(ilog)</code>	iteration log (up to 20 iterations)
<code>e(gradient)</code>	gradient vector
<code>e(V)</code>	variance–covariance matrix of the estimators
<code>e(V_modelbased)</code>	model-based variance

Functions

<code>e(sample)</code>	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

Skewed logit analysis is an alternative to logit that relaxes the assumption that individuals with initial probability of 0.5 are most sensitive to changes in independent variables.

The log-likelihood function for skewed logit is

$$\ln L = \sum_{j \in S} w_j \ln F(\mathbf{x}_j \mathbf{b}) + \sum_{j \notin S} w_j \ln \{1 - F(\mathbf{x}_j \mathbf{b})\}$$

where S is the set of all observations j such that $y_j \neq 0$, $F(z) = 1 / \{1 + \exp(z)\}^\alpha$, and w_j denotes the optional weights. $\ln L$ is maximized as described in [R] [Maximize](#).

This command supports the Huber/White/sandwich estimator of the variance and its clustered version using `vce(robust)` and `vce(cluster clustvar)`, respectively. See [P] [_robust](#), particularly [Maximum likelihood estimators](#) and [Methods and formulas](#).

`scobit` also supports estimation with survey data. For details on VCEs with survey data, see [SVY] [Variance estimation](#).

References

- Achen, C. H. 2002. Toward a new political methodology: Microfoundations and ART. *Annual Review of Political Science* 5: 423–450. <https://doi.org/10.1146/annurev.polisci.5.112801.080943>.
- Nagler, J. 1994. Scobit: An alternative estimator to logit and probit. *American Journal of Political Science* 38: 230–255. <https://doi.org/10.2307/2111343>.

Also see

- [R] [scobit postestimation](#) — Postestimation tools for `scobit`
- [R] [cloglog](#) — Complementary log–log regression
- [R] [glm](#) — Generalized linear models
- [R] [logistic](#) — Logistic regression, reporting odds ratios
- [SVY] [svy estimation](#) — Estimation commands for survey data
- [U] [20 Estimation and postestimation commands](#)

Stata, Stata Press, and Mata are registered trademarks of StataCorp LLC. Stata and Stata Press are registered trademarks with the World Intellectual Property Organization of the United Nations. StataNow and NetCourseNow are trademarks of StataCorp LLC. Other brand and product names are registered trademarks or trademarks of their respective companies. Copyright © 1985–2025 StataCorp LLC, College Station, TX, USA. All rights reserved.

For suggested citations, see the FAQ on [citing Stata documentation](#).

