power logistic general — Power analysis for logistic regression: General case[+]

[+]This command is part of StataNow.

Description

`power logistic` computes sample size, power, or effect size for a test of one coefficient in logistic regression. This entry describes how to use `power logistic` to plan a study that will be modeled using logistic regression with one covariate of interest, X, and up to 20 nuisance covariates $\mathbf{Z} = (Z_1, Z_2, \ldots)$. Covariates X and $\mathbf{Z}$ can be continuous, discrete, or a combination of both. For information about how to use `power logistic` in the special cases of one or two binary covariates, see [PSS-2] **power logistic onebin** and [PSS-2] **power logistic twobin**, respectively.

By default, `power logistic` computes sample size for a given power and effect size, where the effect size may be specified as a coefficient or an odds ratio. Alternatively, it can compute power given sample size and effect size, or it can compute the effect size given power and sample size.

Quick start

Sample size for logistic regression with one binary covariate X, given a coefficient of 0.4055 for X under the alternative hypothesis H_a, population prevalence of X of 0.22, and an intercept of -2; using the default power of 0.8 and significance level $\alpha = 0.05$

```
power logistic, x(distribution(bernoulli 0.22) coefficient(0.4055))     ///
    intercept(-2)
```

Same as above, but use argument *oratio*$_X$ to specify the odds ratio for X of $\exp(0.4055) = 1.5$ instead of the coefficient

```
power logistic 1.5, x(distribution(bernoulli 0.22)) intercept(-2)
```

Same as above, but specify the success probability of Y conditional on the means of X and Z instead of argument *oratio*$_X$

```
power logistic, x(distribution(bernoulli 0.22)) intercept(-2)           ///
    pycondxmzm(0.128892)
```

Same as above, but specify different values for the prevalence of X and display the effect of X as a coefficient

```
power logistic, x(distribution(bernoulli (0.2 0.22 0.24 0.26)))          ///
    intercept(-2) pycondxmzm(0.128892) effect(coefficient)
```

Sample size for logistic regression with covariate of interest $X \sim$ normal(20, 5) and nuisance covariates $\mathbf{Z} = (Z_1, Z_2)$, where $Z_1 \sim$ exponential(12) and $Z_2 \sim$ binomial(3, 0.4), and given, under the alternative hypothesis H_a, an odds ratio of 1.8 for a 5-unit change in X, and odds ratios of 1.1 and 1.5 for 1-unit changes in Z_1 and Z_2; also, specify a correlation of 0.28 between X and $\mathbf{Z}$ and $\Pr\{Y = 1 | X = E(X), \mathbf{Z} = E(\mathbf{Z})\} = 0.56$

```
power logistic, x(distribution(normal 20 5) oratio(1.8, unit(5)))        ///
    z1(distribution(exponential 12) oratio(1.1))                         ///
    z2(distribution(binomial 3 0.4) oratio(1.5)) pycondxmzm(0.56) corrxz(0.28)
```

Same as above, but specify different standard deviations for X, and discretize Z_1 into 25 bins

```
power logistic, x(distribution(normal 20 (4 5 6)) oratio(1.8, unit(5)))  ///
   z1(distribution(exponential 12, nbins(25)) oratio(1.1))               ///
   z2(distribution(binomial 3 0.4) oratio(1.5)) pycondxmzm(0.56) corrxz(0.28)
```

Power given a sample size of 200 and previous values of other parameters

```
power logistic, x(distribution(normal 20 (4 5 6)) oratio(1.8, unit(5)))  ///
   z1(distribution(exponential 12, nbins(25)) oratio(1.1))               ///
   z2(distribution(binomial 3 0.4) oratio(1.5)) pycondxmzm(0.56)         ///
   corrxz(0.28) n(200)
```

Effect size given a sample of size 200, power of 90%, $\Pr\{Y = 1|X = 0, \mathbf{Z} = E(\mathbf{Z})\} = 0.108$, and previous values of other parameters

```
power logistic, x(distribution(normal 20 (4 5 6)))                       ///
   z1(distribution(exponential 12, nbins(25)) oratio(1.1))               ///
   z2(distribution(binomial 3 0.4) oratio(1.5)) pycondx0zm(0.108)        ///
   corrxz(0.28) n(200) power(0.9)
```

Menu

Statistics > Power, precision, and sample size

Syntax

Compute sample size

```
power logistic oratio_X, x(xzspec) [z1(xzspec) [z2(xzspec) [...]]
                                   power(numlist) generalopts]
```

Compute power

```
power logistic oratio_X, x(xzspec) n(numlist) [z1(xzspec) [z2(xzspec) [...]]
                                              generalopts]
```

Compute effect size

```
power logistic, x(xzspec) n(numlist) power(numlist) [z1(xzspec) [z2(xzspec) [...]]
                                                    generalopts]
```

oratio$_X$ is the odds ratio for covariate of interest X under the alternative hypothesis H_a. Argument *oratio*$_X$ may be specified either as one number or as a list of values in parentheses (see **[U] 11.1.8 numlist**); argument *oratio*$_X$ does not appear in the dialog box.

generalopts	Description
Main	
* alpha(*numlist*)	significance level; default is alpha(0.05)
* power(*numlist*)	power; default is power(0.8)
* beta(*numlist*)	probability of type II error; default is beta(0.2)
* n(*numlist*)	sample size; required to compute power or effect size
nfractional	allow fractional sample size
† x(*xzspec*)	distribution and effect of covariate of interest X
effect(oratio | coefficient)	specify the type of effect to display; default is effect(oratio)
z[#](*xzspec*)	distribution and effect of nuisance covariate $Z_{\#}$
* corrxz(*numlist*)	coefficient of (multiple) correlation between covariate X and all covariates $\mathbf{Z}$; default is corrxz(0)
* pycondxmzm(*numlist*)	success probability of Y given mean values of X and $\mathbf{Z}$; $\Pr\{Y = 1 \mid X = E(X),\ \mathbf{Z} = E(\mathbf{Z})\}$
* pycondx0zm(*numlist*)	success probability of Y given $X = 0$ and mean values of covariates $\mathbf{Z}$; $\Pr\{Y = 1 \mid X = 0,\ \mathbf{Z} = E(\mathbf{Z})\}$
* intercept(*numlist*)	intercept for logistic regression
direction(upper | lower)	direction of the effect for effect-size determination; default is direction(upper), which means that the postulated odds ratio for X is greater than 1 (thus, the coefficient is positive)
parallel	treat number lists in starred options or in command arguments as parallel when multiple values per option or argument are specified (do not enumerate all possible combinations of values)
Discretization	
* minbins(*numlist*)	minimum product of bins for all covariates
* nbins(*numlist*)	number of bins to use for discretizing each binned covariate
Table	
[no]table[(*tablespec*)]	suppress table or display results as a table; see [PSS-2] **power, table**
saving(*filename* [, replace])	save the table data to *filename*; use replace to overwrite existing *filename*
Graph	
graph[(*graphopts*)]	graph results; see [PSS-2] **power, graph**
Iteration	
init(#)	initial odds ratio for effect-size calculation
iterate(#)	maximum number of iterations; default is iterate(500)
tolerance(#)	parameter tolerance; default is tolerance(1e-12)
ftolerance(#)	function tolerance; default is ftolerance(1e-12)
[no]log	suppress or display iteration log
[no]dots	suppress or display iterations as dots
coefx(*numlist*)	coefficient for X in logistic regression; specify instead of odds ratio $oratio_X$
notitle	suppress the title

†`x()` is required.

*Specifying a list of values in at least two starred options, or in argument *oratio*$_X$ and at least one starred option, results in computations for all possible combinations of the values; see **[U] 11.1.8 numlist**. Also see `parallel`.

`collect` is allowed; see **[U] 11.1.10 Prefix commands**.

`notitle` and `coefx()` do not appear in the dialog box.

xzspec	Description
†`distribution(`*distspec* [`, *nbins(`*numlist*`)`]`)`	covariate distribution and the number of bins for discretization
*`oratio(`*numlist*[`, unit(#|sd)`]`)`	odds ratio for the covariate and the unit change; default is `unit(1)`
*`coefficient(`*numlist*`)`	coefficient for the covariate

†`distribution()` is required.

*Specifying a list of values in at least two starred options, suboptions, or arguments results in computations for all possible combinations of the values; also see the `parallel` option.

distspec	Distribution
`bernoulli` p^*	Bernoulli with success probability p; synonym for `binomial(1` p`)`
`beta` a^* b^*	beta with shape parameters a and b
`binomial` n p^*	binomial with n trials and success probability p
`exponential` b^*	exponential with scale parameter b
`laplace` m^* b^*	Laplace with mean m and scale parameter b
`logistic` m^* s^*	logistic with mean m and scale parameter s
`lognormal` μ^* σ^*	lognormal with mean μ and standard deviation σ
`normal` μ^* σ^*	normal with mean μ and standard deviation σ
`ordinal` $(v_1^*\ \ p_1)$ $(v_2^*\ \ p_2)$ [$(v_3^*\ \ p_3)$ [`...`]]	ordinal with values v_1^*, v_2^*, etc., and respective probabilities p_1, p_2, etc.
`poisson` m^*	Poisson with mean m
`uniform` a^* b^*	uniform on the interval $[a, b]$

*Starred parameters may be specified either as one number or as a list of values in parentheses (see **[U] 11.1.8 numlist**). Specifying a list of values in at least two starred parameters, options, suboptions, or arguments results in computations for all possible combinations of the values; also see the `parallel` option.

where *tablespec* is

column[:*label*] [*column*[:*label*] [...]] [, *tableopts*]

column is one of the columns defined below, and *label* is a column label (may contain quotes and compound quotes).

column	Description	Symbol
`alpha`	significance level	α
`power`	power	$1-\beta$
`beta`	type-II-error probability	β
`N`	number of subjects	N
`delta`	effect size	δ
`oratiox`	odds ratio for X	OR_X
`unitx`	unit change in X for odds ratio	u_X
`coefx`	coefficient for X	β_X
`nbinsx`	number of bins for discretized X	B_X
`oratioz#`	odds ratio for $Z_\#$	$\text{OR}_{Z\#}$
`unitz#`	unit change in $Z_\#$ for odds ratio	$u_{Z\#}$
`coefz#`	coefficient for $Z_\#$	$\zeta_\#$
`nbinsz#`	number of bins for discretized $Z_\#$	$B_{Z\#}$
`corrxz`	multiple correlation between X and $\mathbf{Z}$	R
`pycondxmzm`	success probability of Y given mean values of X and $\mathbf{Z}$, $\Pr\{Y=1\|X=E(X),\ \mathbf{Z}=E(\mathbf{Z})\}$	$p_{Y\|X=E(X),\,Zs=E(Zs)}$
`pycondx0zm`	success probability of Y given X is 0 and mean value of $\mathbf{Z}$, $\Pr\{Y=1\|X=0,\ \mathbf{Z}=E(\mathbf{Z})\}$	$p_{Y\|X=0,\,Zs=E(Zs)}$
`intercept`	intercept	ζ_0
`nbins`	number of bins to use for discretizing each covariate	B_{all}
`minbins`	minimum product of bins for all covariates	B_{min}
`totalbins`	actual product of bins for all covariates	B_{tot}
`a[x|z#]`	parameter a from distribution of X or $Z_\#$ (if specified)	a_X or $a_{Z\#}$
`b[x|z#]`	parameter b from distribution of X or $Z_\#$ (if specified)	b_X or $b_{Z\#}$
`m[x|z#]`	mean m from distribution of X or $Z_\#$ (if specified)	m_X or $m_{Z\#}$
`n[x|z#]`	number of trials n from binomial distribution of X or $Z_\#$ (if specified)	n_X or $n_{Z\#}$
`p[x|z#]`	success probability from distribution of X or $Z_\#$ (if specified)	p_X or $p_{Z\#}$
`s[x|z#]`	scale s from logistic distribution of X or $Z_\#$ (if specified)	s_X or $s_{Z\#}$
`mu[x|z#]`	mean μ from distribution of X or $Z_\#$ (if specified)	μ_X or $\mu_{Z\#}$
`sigma[x|z#]`	standard deviation σ from distribution of X or $Z_\#$ (if specified)	σ_X or $\sigma_{Z\#}$
`v#[x|z#]`	level # ordinal value, $v_\#$, from ordinal distribution of X or $Z_\#$ (if specified)	$v\#_X$ or $v\#_{Z\#}$
`p#[x|z#]`	probability $p_\#$ that X or $Z_\#$equals $v_\#$ from ordinal distribution (if specified)	$p\#_X$ or $p\#_{Z\#}$
`nl[x|z#]`	number of levels from ordinal distribution of X or $Z_\#$ (if specified)	nl_X or $\text{nl}_{Z\#}$
`target`	target parameter; odds ratio or coefficient for X	
`_all`	display all supported columns	

Options

Main

`alpha()`, `power()`, `beta()`, `n()`, `nfractional`; see [PSS-2] **power**. The `nfractional` option is allowed only for sample-size determination.

`x(`*xzspec*`)` is a required option that specifies information about the covariate of interest, X. This includes the name of the X distribution and any parameters needed to specify that distribution, as well as the number of bins to use when discretizing X. When calculating power or sample size, *xzspec* specifies the effect of X as a coefficient or an odds ratio.

xzspec consists of the following suboptions: `distribution(`*distspec* [`, nbins(`*numlist*`)`]`)`, `oratio(`*numlist* [`, unit(#|sd)`]`)`, and `coefficient(`*numlist*`)`.

`distribution(`*distspec* [`, nbins(`*numlist*`)`]`)` is a required suboption, where *distspec* specifies the distribution of the covariate and `nbins()` specifies how it is to be discretized.

distspec consists of the distribution name and parameters. Starred parameters may be specified as a number or numlist in parentheses. *distspec* is one of the following:

`bernoulli` p^* specifies a Bernoulli distribution with parameter p, where $0 < p < 1$. The Bernoulli distribution describes a binary trial with outcomes 0 (failure) or 1 (success). Parameter p is the probability of success, and a Bernoulli random variable has mean p. Bernoulli covariates always use two bins during discretization, one for each possible outcome.

`beta` a^* b^* specifies a beta distribution with shape parameters a and b, where $a > 0$ and $b > 0$. A random variable following a beta distribution is defined only over the interval $[0, 1]$, and its mean is $a/(a + b)$.

`binomial` n p^* specifies a binomial distribution with parameters n and p, where n is a positive integer and $0 < p < 1$. A binomial random variable models the number of successes in n Bernoulli trials, each with success probability p, and its mean is $n \times p$. Binomial covariates always use $n + 1$ bins during discretization, one for each possible outcome.

`exponential` b^* specifies an exponential distribution with scale parameter b, where $b > 0$. A random variable following an exponential distribution can take only positive values, and its mean is b.

`laplace` m^* b^* specifies a Laplace distribution with mean m and scale parameter b, where $b > 0$. The Laplace distribution is symmetric around its mean and is defined for all real numbers.

`logistic` m^* s^* specifies a logistic distribution with mean m and scale parameter s, where $s > 0$. The logistic distribution is symmetric around its mean and is defined for all real numbers.

`lognormal` μ^* σ^* specifies a lognormal distribution with parameters μ and σ, where $\sigma > 0$. If random variable Q is lognormal with parameters μ and σ, its mean is $\exp(\mu + \sigma^2/2)$, and the natural logarithm of Q follows a normal distribution with mean μ and standard deviation σ.

`normal` μ^* σ^* specifies a normal distribution with mean μ and standard deviation σ, where $\sigma > 0$. The normal distribution is symmetric around its mean and is defined for all real numbers.

`ordinal` $(v_1^* \; p_1)$ $(v_2^* \; p_2)$ [$(v_3^* \; p_3)$ [...]] specifies an ordinal distribution with parameters **v** and **p**, which are equal-length vectors. Parameter **v** is an ordered vector of values $(v_1, v_2, \ldots, v_J)$, where $v_1 < v_2 < \cdots < v_J$ and $J \leq 20$. Parameter **p** is a vector of probabilities corresponding to those values $(p_1, p_2, \ldots, p_J)$, where each probability is between 0 and 1 and $p_1 + p_2 + \cdots + p_J = 1$. To specify an ordinal covariate, enclose corresponding pairs of v_j and p_j values in parentheses, such as `x(ordinal (1 0.3) (2 0.5) (3 0.2))`. During discretization, ordinal covariates always use as many bins as they have values: one for each possible outcome. The mean of an ordinal random variable is $v_1 \times p_1 + v_2 \times p_2 + \cdots + v_J \times p_J$.

`poisson` m^* specifies a Poisson distribution with parameter m, where $m > 0$. The Poisson distribution is often used to model count data. A Poisson random variable can take only nonnegative integer values, and its mean is m.

`uniform` a^* b^* specifies a continuous uniform distribution over the interval $[a, b]$, where $a < b$. The mean of a uniformly distributed random variable is $(a + b)/2$.

`nbins(`*numlist*`)` specifies the number of bins to use when discretizing the covariate; the number of bins must be an integer between 2 and 100,000,000. For this covariate, the `nbins()` suboption overrides any value set by the `nbins()` global option. The `nbins()` suboption is not allowed with the `bernoulli`, `binomial`, or `ordinal` distribution.

`oratio(`*numlist*[`, unit(#|sd)`]`)` specifies the odds ratio for the covariate in the logistic regression. The odds ratio must be positive, and only one of the `oratio()` or `coefficient()` suboption may be specified for each covariate. When you specify the covariate of interest X, the `oratio()` suboption may not take the value 1, and `oratio()` is not a valid suboption of `x()` when calculating effect size or when argument *oratio*$_X$ is specified.

`unit(#|sd)` specifies the unit change for the odds ratio. Specifying `unit(sd)` indicates that the odds ratio is for a 1-standard-deviation increase in the covariate. The relationship between the odds ratio and the coefficient is `oratio` = exp(`coefficient`×`unit`). The default is `unit(1)`.

`coefficient(`*numlist*`)` specifies the coefficient for the covariate in the logistic regression. Only one of the `coefficient()` or `oratio()` suboption may be specified for each covariate. When you specify the covariate of interest X, the `coefficient()` suboption may not take the value 0, and `coefficient()` is not a valid suboption of `x()` when calculating effect size or when argument *oratio*$_X$ is specified.

`effect(oratio|coefficient)` specifies how to report the effect size in the output. By default, the effect is output as the odds ratio for X unless the coefficient for X is specified, in which case it defaults to the coefficient. The `effect()` option is used to override the default.

`z[#](`*xzspec*`)` specifies information about nuisance covariate $Z_{\#}$. This includes the distribution and its parameters for $Z_{\#}$, the coefficient or odds ratio for $Z_{\#}$, and the number of bins to use when discretizing $Z_{\#}$. Up to 20 Z covariates may be specified, and Z covariates are assumed to be uncorrelated with each other (correlation with X is allowed; see the `corrxz()` option).

`corrxz(`*numlist*`)` specifies the correlation between X and **Z**, labeled R, where $-1 < R < 1$. If there is just one Z covariate, this is Pearson's correlation coefficient. If there are multiple Z covariates, `corrxz()` specifies the coefficient of multiple correlation, a generalization of Pearson's correlation coefficient. The default is `corrxz(0)`, which indicates no correlation between X and **Z**.

`pycondxmzm(`*numlist*`)` specifies the conditional success probability of Y given mean values of X and all $\mathbf{Z}$ covariates, $\Pr\{Y = 1|X = E(X), \mathbf{Z} = E(\mathbf{Z})\}$, where $0 < \Pr\{Y = 1|X = E(X), \mathbf{Z} = E(\mathbf{Z})\} < 1$. This option is not allowed with effect-size determination.

`pycondx0zm(`*numlist*`)` specifies the conditional success probability of Y given $X = 0$ and mean values of all $\mathbf{Z}$ covariates, $\Pr\{Y = 1|X = 0, \mathbf{Z} = E(\mathbf{Z})\}$, where $0 < \Pr\{Y = 1|X = 0, \mathbf{Z} = E(\mathbf{Z})\} < 1$. `pycondx0zm()` may not be combined with the `intercept()` option.

`intercept(`*numlist*`)` specifies the intercept for the logistic regression, ζ_0. The `intercept()` option may not be combined with the `pycondx0zm()` option.

`direction()`, `parallel`; see [PSS-2] **power**.

Discretization

`minbins(`*numlist*`)` specifies the minimum product of the bins for all covariates B_{min}, where B_{min} is an integer between 2 and 100,000,000. Covariates with Bernoulli, binomial, and ordinal distributions always use one bin for each value they can take, and the `nbins()` suboption of `x()` and `z#()` sets the number of bins for one covariate at a time. The value of `minbins()` is used when determining how many bins to allocate to the remaining covariates; they are discretized such that the product of the number of bins of each covariate exceeds `minbins()`. $B_{\text{min}} \leq B_X \times B_{Z_1} \times B_{Z_2} \ldots$, where B_X is the number of bins for X, B_{Z_1} is the number of bins for Z_1, and so on. The default is `minbins(10000)` for power and sample-size calculations and `minbins(1000)` for effect-size calculations. The `minbins()` option may not be combined with the `nbins()` global option (but it can be combined with the `nbins()` suboption of `x()` or `z#()`).

`nbins(`*numlist*`)` specifies the number of bins to use when discretizing each covariate; the number of bins must be an integer between 2 and 100,000,000. The `nbins()` option can be overridden on a per-covariate basis by specifying the `nbins()` suboption of `x()` or `z#()`. Covariates with Bernoulli, binomial, and ordinal distributions always use one bin for each value they can take, so they do not respect `nbins()`. The `nbins()` option may not be combined with the `minbins()` option (but the `nbins()` suboption of `x()` or `z#()` can be combined with `minbins()`). Note that the product of the number of bins for all covariates, B_{tot}, may not exceed 100,000,000. ($B_{\text{tot}} = B_X \times B_{Z_1} \times B_{Z_2} \cdots \leq$ 100,000,000, where B_X is the number of bins for X, B_{Z_1} is the number of bins for Z_1, and so on). Thus, the maximum of `nbins(100000000)` may be specified only if there is a single covariate of interest without any nuisance covariates.

Table

`table`, `table()`, `notable`; see [PSS-2] **power, table**.

`saving()`; see [PSS-2] **power**.

Graph

`graph`, `graph()`; see [PSS-2] **power, graph**. Also see the *column* table for a list of symbols used by the graphs.

Iteration

`init(#)` specifies the initial value of the odds ratio for X during effect-size determination. The default is `init(1.5)` with `direction(upper)` and `init(0.67)` with `direction(lower)`.

`iterate()`, `tolerance()`, `ftolerance()`, `log`, `nolog`, `dots`, `nodots`; see [PSS-2] **power**.

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

`coefx(`*numlist*`)` specifies the coefficient for covariate X in the logistic regression, β_X, where $\beta_X \neq 0$. The `coefx()` option may be specified instead of argument *oratio*$_X$. This option is not allowed with effect-size determination.

`notitle`; see [PSS-2] **power**.

Remarks and examples

Remarks are presented under the following headings:

This entry describes the `power logistic` command and the methodology for power and sample-size analysis for logistic regression with one covariate of interest, X, and up to 20 nuisance covariates $\mathbf{Z} = (Z_1, Z_2, \ldots)$. Covariates X and $\mathbf{Z}$ can be continuous, discrete, or a combination of both. See [PSS-2] **Intro (power)** for a general introduction to power and sample-size analysis, and see [PSS-2] **power** for a general introduction to using the `power` command for hypothesis tests.

Introduction

Logistic regression is a commonly used statistical method for analyzing binary outcome variables. Researchers often need to determine the appropriate sample size to ensure sufficient power for detecting the association between a covariate of interest (X) and a binary outcome variable (Y) while controlling for the effect of nuisance covariates ($\mathbf{Z}$).

For example, consider a study that examines factors influencing whether migratory birds return to the same nesting site from one year to the next. Binary outcome Y is an indicator of whether the nesting site was reused, where the observed $y_i = 1$ if bird i returns to the nesting site it used last year and $y_i = 0$ if bird i does not. We will use logistic regression to test whether birds of one sex are more likely to return to the same nesting site than birds of the other sex. We define binary covariate of interest X as an indicator for female birds, where the observed $x_i = 1$ if bird i is female and $x_i = 0$ if bird i is male. Previous observations suggest that birds that mate with the same partner as last year are more likely to return to the same nesting site, as are heavier birds. We are not interested in studying the effect of a bird's mate or weight, but it would be foolhardy to ignore these effects. We include mate in our logistic regression as nuisance covariate Z_1, where the observed $z_{1i} = 1$ if bird i has the same mate as last year and $z_{1i} = 0$ if not. And we include weight as nuisance covariate Z_2, where the observed z_{2i} is the weight of bird i. The logistic regression can be written as

$$\Pr(y_i = 1 | x_i, \mathbf{z}_i) = H(\beta_X x_i + \zeta_0 + \zeta_1 z_{1i} + \zeta_2 z_{2i}) \qquad i = 1, 2, \ldots, n$$

where β_X is the coefficient quantifying the effect of covariate X, a bird's sex, on nesting-site reuse; ζ_0 is the logistic intercept; ζ_1 is the effect of covariate Z_1, partnering with the same mate; ζ_2 is the effect of covariate Z_2, weight; and n is the sample size. Function $H(\eta) = \{1 + \exp(-\eta)\}^{-1}$ is the logistic distribution function.

The effect of covariate X can also be expressed in terms of an odds ratio,

$$\text{OR}_X = \exp(\beta_X \text{u}_X) = \Pr(Y=1|X=1)\Pr(Y=0|X=0)/\{\Pr(Y=0|X=1)\Pr(Y=1|X=0)\}$$

where u_X is the unit change in X for the odds ratio and $\text{u}_X > 0$. In this example, u_X must equal 1 because X is a Bernoulli random variable. The null hypothesis is $H_0 : \beta_X = 0$, which can also be expressed as $H_0 : \text{OR}_X = 1$. The alternative hypothesis is $H_a : \beta_X \neq 0$ or, equivalently, $H_a : \text{OR}_X \neq 1$.

Logistic regression is commonly used to analyze binary outcomes in observational studies, such as the study of nesting-site reuse. But it can also be used to analyze data from a randomized controlled trial, where trial participants are randomly assigned to a treatment. For instance, a public health study might investigate whether attending a support group helps smokers quit smoking. In this example, participation in the support group is binary covariate X: Some participants will be randomly assigned to attend support group meetings for, say, three months, whereas other participants will be randomized to the control group, which does not attend support group meetings. At the end of three months, smoking status is recorded; this is binary outcome Y. Based on previous research, we anticipate males and females will have different success rates at quitting smoking, and we expect younger smokers to be more successful at quitting than older smokers. We are not interested in studying the effect of sex or age on smoking cessation, but it would be foolhardy to ignore these effects, so we include sex in our logistic regression as nuisance covariate Z_1, and we include age as nuisance covariate Z_2.

The `power logistic` command provides power and sample-size analysis for the test of $\beta_X = 0$ in logistic regression. The formula for power, sample-size, and effect-size calculations is based on the likelihood-ratio test of $\beta_X = 0$. However, Bush (2015) demonstrates that sample-size requirements for Wald and score tests are generally equivalent to the sample-size requirement for the likelihood-ratio test. Thus, these calculations may be used to plan studies that will be analyzed using the `logit` command, which conducts a Wald test of $\beta_X = 0$, or the `logistic` command, which conducts an equivalent Wald test of $\text{OR}_X = 1$. If you prefer a likelihood-ratio test, you can use the `lrtest` command.

Using power logistic with arbitrary covariates

`power logistic` computes sample size, power, or effect size for a test of one coefficient in logistic regression. This entry describes how to use `power logistic` when the logistic regression can have multiple discrete and continuous covariates, only one of which is of interest (X), while the others are nuisance covariates (**Z**). All computations are performed for a two-sided hypothesis test where, by default, the significance level is set to 0.05. You may change the significance level by specifying the `alpha()` option. You must specify the distribution of covariate X in the target population with the `x()` option. Nuisance covariates can be specified in `z1()`, `z2()`, and so on.

Power and sample-size calculations require that you specify information about three types of parameters from the logistic regression: the X coefficient (β_X), the Z coefficients ($\zeta_1, \zeta_2, \ldots$), and the intercept (ζ_0). For each nuisance covariate $Z_\#$ in the logistic regression, you must specify information about $\zeta_\#$ using either `z#(oratio())` or `z#(coefficient())`.

The information about the X coefficient (β_X) can be specified directly in the `x(coefficient())` option or indirectly as an argument *oratio*$_X$, in the `x(oratio())` option, or in the `pycondxmzm()` option. The information about the intercept (ζ_0) can be specified directly in the `intercept()` option or indirectly in the `pycondx0zm()` or `pycondxmzm()` option. See figure 1 for a graphical depiction of various ways to provide information about β_X and ζ_0.

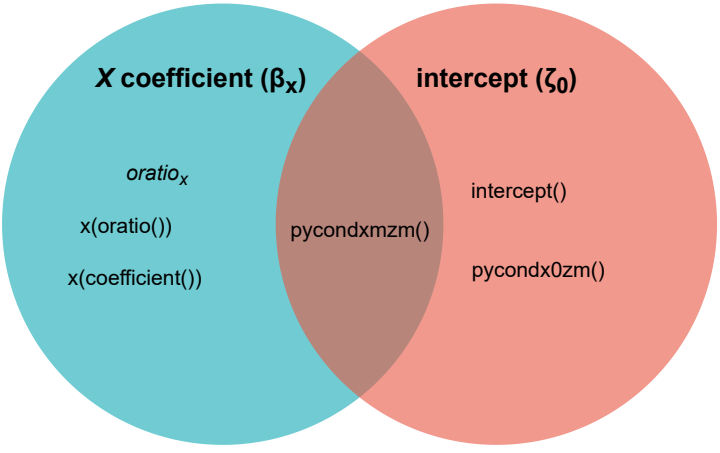

Figure 1. Specifying information about β_X and ζ_0

Valid ways of specifying the X coefficient and intercept include, for example, *oratio*$_X$ and `intercept()` or `pycondxmzm()` and `pycondx0zm()`. However, the combination of `intercept()` and `pycondx0zm()` is invalid because no information is provided about the X coefficient. When you compute sample size, the power of the test may be specified using the `power()` option, which has a default of 0.8. When you compute power, the sample size must be specified using the `n()` option.

When `power logistic` is used to calculate effect size β_X, the command specification cannot include *oratio*$_X$ or options that provide information about the X coefficient, such as the `pycondxmzm()` or `x(oratio())` option. In this case, the procedure for specifying the Z coefficient or Z coefficients is unchanged, but information about the intercept must be specified using `intercept()` or `pycondx0zm()`.

To calculate effect size, you must specify sample size using the `n()` option, power using the `power()` option, and, optionally, the direction of the effect using the `direction()` option. The default is `direction(upper)`, which means that coefficient β_X is assumed to be positive. This is equivalent to assuming that the odds ratio OR$_X$ is greater than 1. You can change the direction to `lower`, which means that $\beta_X < 0$ or, equivalently, OR$_X < 1$.

The `effect()` option can be used to specify the type of effect to be reported in the output. Valid choices are `effect(oratio)` and `effect(coefficient)`. If the coefficient for X is specified, the default output parameterizes the effect size as a coefficient; otherwise, the default is an odds ratio. The `effect()` option is used to override the default.

By default, the computed sample size is rounded up. You can specify the `nfractional` option to see the corresponding fractional sample size; see *Fractional sample sizes* in [PSS-4] **Unbalanced designs** for an example. The `nfractional` option is allowed only for sample-size determination.

Some of the computations of `power logistic` require iteration, specifically, the computations used in effect-size determination. The default initial value for OR$_X$ is 1.5 with `direction(upper)` and 0.67 with `direction(lower)`. This may be changed by specifying the `init()` option. See [PSS-2] **power** for descriptions of other options that control the iteration procedure.

In the following sections, we describe the use of `power logistic` to compute sample size, power, and effect size.

Computing sample size

To compute sample size, you must specify the distribution of the covariate of interest X using the `distribution()` suboption of `x()`; the distributions and coefficients or odds ratios of any nuisance covariates using `z1()`, `z2()`, and so on; the information necessary to determine parameters β_X and ζ_0, as described in *Using power logistic with arbitrary covariates*; and, optionally, the power of the test using the `power()` option or the type-II-error probability using the `beta()` option. A default power of 0.8 is assumed if the power of the test is not specified. The level of the test is specified using the `alpha()` option, with a default of `alpha(0.05)`.

Example 1: Sample size with a standard normal covariate of interest

Consider an example from the seminal work on sample-size calculation for logistic regression by Whittemore (1981, 31) that describes the design of a study testing "the null hypothesis that risk of coronary heart disease (CHD) among white males aged 39–59 is unaffected by serum cholesterol levels". In example 1 of [PSS-2] **power logistic onebin** and example 1 of [PSS-2] **power logistic twobin**, we approached this scenario using a binary indicator variable for elevated cholesterol, but here we model serum cholesterol as a normally distributed random variable, just as Whittemore did.

Normally distributed covariate of interest X is serum cholesterol, which has been standardized to have mean 0 and standard deviation 1. We want to control for the effects of triglyceride, nuisance covariate Z_1, which is a known risk factor for CHD. Log-triglyceride levels are standardized and modeled as a normal random variable with mean 0 and standard deviation 1. Following Whittemore (1981), we anticipate the odds ratio for a one-unit change in Z_1 to be 1.25, and the correlation between X and Z_1 is estimated to be 0.4. Based on data from Hulley et al. (1980), we assume a probability of 0.07 that an individual in the target population will develop CHD during an 18-month study period if he has average values for serum cholesterol and triglycerides; this will be entered as `pycondxmzm(0.07)`.

Like Whittemore, we will calculate the sample size required to detect an odds ratio of 1.65 at the $\alpha = 0.05$ level, but we will use the default power of 80%.

```
. power logistic 1.65, x(distribution(normal 0 1))
> z1(distribution(normal 0 1) oratio(1.25))
> corrxz(0.4) pycondxmzm(0.07)

Estimated sample size for logistic regression odds-ratio test
Likelihood-ratio test
H0: OR_X = 1  versus  Ha: OR_X != 1

Study parameters:

        alpha =    0.0500
        power =    0.8000
        delta =    1.6500  (odds ratio)
       corrxz =    0.4000
   pycondxmzm =    0.0700

Covariate of interest X: Normal(mux, sigmax), bins = 100
      oratiox =    1.6500
          mux =    0.0000
       sigmax =    1.0000

Nuisance covariate Z1: Normal(muz1, sigmaz1), bins = 100
     oratioz1 =    1.2500
         muz1 =    0.0000
      sigmaz1 =    1.0000

Estimated sample size:

            N =       521
```

The output of power logistic begins by displaying information about the test to be conducted and its null and alternative hypotheses. The study parameters we specified are listed next, followed by information about covariates X and Z_1. For both X and Z_1, we see the odds ratios we specified, along with means and standard deviations. We also see that these normal distributions were discretized into 100 bins each. We did not specify the number of bins to use, so power logistic used the default of minbins(10000) for sample-size calculations. The minbins() option specifies the minimum value for the product of bins for all covariates rather than directly specifying the number of bins per covariate (which can be done with the nbins() option or the nbins() suboption of the x(distribution()) and z#(distribution()) options). With 100 bins per covariate, we have a product of $100 \times 100 = 10{,}000$, satisfying the default minbins(10000).

Finally, we find that a sample of 521 subjects is required to detect an odds ratio of 1.65 with 80% power using a 5% level test. But how sensitive is this sample-size estimate to our discretization process? We repeat the previous calculation, but this time, we specify different values for the minimum product of bins for all covariates. For demonstration purposes, we move the specification of OR_X from argument *oratio*$_X$ to suboption x(oratio()).

```
. power logistic, x(distribution(normal 0 1) oratio(1.65))
> z1(distribution(normal 0 1) oratio(1.25))
> corrxz(0.4) pycondxmzm(0.07)
> minbins(100 1000 10000 100000 1000000)

Estimated sample size for logistic regression odds-ratio test
Likelihood-ratio test
H0: OR_X = 1  versus  Ha: OR_X != 1

Covariate of interest X: Normal(mux, sigmax)
  Nuisance covariate Z1: Normal(muz1, sigmaz1)
```

alpha	power	N	delta	oratiox	mux	sigmax	nbinsx	oratioz1
.05	.8	600	1.65	1.65	0	1	10	1.25
.05	.8	539	1.65	1.65	0	1	32	1.25
.05	.8	521	1.65	1.65	0	1	100	1.25
.05	.8	514	1.65	1.65	0	1	317	1.25
.05	.8	512	1.65	1.65	0	1	1000	1.25

muz1	sigmaz1	nbinsz1	corrxz	pycondxmzm	minbins	totalbins
0	1	10	.4	.07	100	100
0	1	32	.4	.07	1000	1024
0	1	100	.4	.07	10000	10000
0	1	317	.4	.07	1.0e+05	1.0e+05
0	1	1000	.4	.07	1.0e+06	1.0e+06

When we specify multiple values for one or more parameters or arguments, power logistic presents the results as a table. Above the table is information about the test to be conducted, with null and alternative hypotheses followed by the distributions of covariates X and Z_1. The table has columns for each of the logistic regression parameters we specified, as well as estimated sample size N and additional details about discretization (nbinsx, nbinsz1, and totalbins).

By increasing the minimum product of bins from 100 to 1,000,000 by successive orders of magnitude, we caused the number of bins for each covariate to increase from 10 all the way to 1,000. Using more bins to discretize the covariates increased the precision of our sample-size calculation, but it came at the expense of increased computational time, with diminishing returns above `minbins()` values of 10,000.

◁

▷ Example 2: Sample size with continuous covariates

Instead of standardizing serum cholesterol and log triglycerides as in example 1, we now input the means and standard deviations in the `x(distribution())` and `z1(distribution())` suboptions. To indicate that the odds ratios now refer to a 1-standard-deviation increase in the covariates, we include suboption `unit(sd)` when specifying the odds ratios of these covariates. We leave the rest of the command specification unchanged.

```
. power logistic, x(distribution(normal 212 38) oratio(1.65, unit(sd)))
> z1(distribution(normal 4.9 0.3) oratio(1.25, unit(sd)))
> corrxz(0.4) pycondxmzm(0.07)

Estimated sample size for logistic regression odds-ratio test
Likelihood-ratio test
H0: OR_X = 1  versus  Ha: OR_X != 1

Study parameters:

        alpha =    0.0500
        power =    0.8000
        delta =    1.0133  (odds ratio)
       corrxz =    0.4000
   pycondxmzm =    0.0700

Covariate of interest X: Normal(mux, sigmax), bins = 100
      oratiox =    1.6500
        unitx =   38.0000
          mux =  212.0000
       sigmax =   38.0000

Nuisance covariate Z1: Normal(muz1, sigmaz1), bins = 100
     oratioz1 =    1.2500
       unitz1 =    0.3000
         muz1 =    4.9000
      sigmaz1 =    0.3000

Estimated sample size:

            N =       521
```

The output of this command is similar to that of example 1, but now information about `unitx` and `unitz1` is included in the descriptions of those covariates. The estimated sample size, however, is unchanged.

◁

▷ Example 3: Sample size with continuous and discrete covariates

We now include an additional nuisance covariate, smoking status, as Z_2. The probability that an individual in the target population is a smoker is 0.38, and the odds ratio for Z_2 is 3. Smoking status is uncorrelated with the covariate of interest, so the coefficient of multiple correlation remains 0.4. We leave the rest of the command specification unchanged except for the addition of the `effect(coefficient)` option to display the effect of X as a coefficient instead of an odds ratio.

```
. power logistic, x(distribution(normal 212 38) oratio(1.65, unit(sd)))
> z1(distribution(normal 4.9 0.3) oratio(1.25, unit(sd)))
> z2(distribution(bernoulli 0.38) oratio(3))
> corrxz(0.4) pycondxmzm(0.07) effect(coefficient)

Estimated sample size for logistic regression coefficient test
Likelihood-ratio test
H0: beta_X = 0  versus  Ha: beta_X != 0

Study parameters:

        alpha =    0.0500
        power =    0.8000
        delta =    0.0132  (coefficient)
       corrxz =    0.4000
   pycondxmzm =    0.0700

Covariate of interest X: Normal(mux, sigmax), bins = 71
        coefx =    0.0132
          mux =  212.0000
       sigmax =   38.0000

Nuisance covariate Z1: Normal(muz1, sigmaz1), bins = 71
     oratioz1 =    1.2500
       unitz1 =    0.3000
         muz1 =    4.9000
      sigmaz1 =    0.3000

Nuisance covariate Z2: Bernoulli(pz2), bins = 2
     oratioz2 =    3.0000
          pz2 =    0.3800

Estimated sample size:

            N =       494
```

Under `Covariate of interest X:`, we now see `coefx` instead of `oratiox` and `unitx`. But the major change is the addition of covariate Z_2, which reduces the estimated sample size to 494. Bernoulli random covariate Z_2 can take only 2 values, which is why it is discretized into 2 bins. discretized into 71 bins each, yielding a product of $71 \times 71 \times 2 = 10{,}082$ bins.

◁

Computing power

To compute power, you must specify the distribution of covariate of interest X using the `x(distribution())` option; the distributions and coefficients or odds ratios of any nuisance covariates using `z1()`, `z2()`, and so on; the information necessary to determine parameters β_X and ζ_0, as described in *Using power logistic with arbitrary covariates* ; and the sample size using the `n()` option. The level of the test is specified using the `alpha()` option, with a default of `alpha(0.05)`.

Example 4: Power of a logistic regression odds-ratio test

Continuing with example 3, we anticipate a sample of 600 subjects and would like to compute the power corresponding to this sample size. We specify the same study parameters we used in example 3, but now we use the `n()` option to specify a sample size of 600.

```
. power logistic, x(distribution(normal 212 38) oratio(1.65, unit(sd)))
> z1(distribution(normal 4.9 0.3) oratio(1.25, unit(sd)))
> z2(distribution(bernoulli 0.38) oratio(3))
> corrxz(0.4) pycondxmzm(0.07) n(600)

Estimated power for logistic regression odds-ratio test
Likelihood-ratio test
H0: OR_X = 1  versus  Ha: OR_X != 1

Study parameters:

        alpha =    0.0500
            N =       600
        delta =    1.0133  (odds ratio)
       corrxz =    0.4000
   pycondxmzm =    0.0700

Covariate of interest X: Normal(mux, sigmax), bins = 71
      oratiox =    1.6500
        unitx =   38.0000
          mux =  212.0000
       sigmax =   38.0000

Nuisance covariate Z1: Normal(muz1, sigmaz1), bins = 71
     oratioz1 =    1.2500
       unitz1 =    0.3000
         muz1 =    4.9000
      sigmaz1 =    0.3000

Nuisance covariate Z2: Bernoulli(pz2), bins = 2
     oratioz2 =    3.0000
          pz2 =    0.3800

Estimated power:

        power =    0.8707
```

If the study recruits 600 participants, the power to detect an odds ratio of 1.65 climbs to 87.07%.

▷ Example 5: Multiple values of study parameters

To investigate the effect of sample size on power, we can specify a list of sample sizes in the `n()` option:

```
. power logistic, x(distribution(normal 212 38) oratio(1.65, unit(sd)))
> z1(distribution(normal 4.9 0.3) oratio(1.25, unit(sd)))
> z2(distribution(bernoulli 0.38) oratio(3))
> corrxz(0.4) pycondxmzm(0.07) n(400 500 600 700)

Estimated power for logistic regression odds-ratio test
Likelihood-ratio test
H0: OR_X = 1  versus  Ha: OR_X != 1

Covariate of interest X: Normal(mux, sigmax)
Nuisance covariates:
   Z1: Normal(muz1, sigmaz1)
   Z2: Bernoulli(pz2)
```

alpha	power	N	delta	oratiox	unitx	mux	sigmax	oratioz1
.05	.7132	400	1.013	1.65	38	212	38	1.25
.05	.8052	500	1.013	1.65	38	212	38	1.25
.05	.8707	600	1.013	1.65	38	212	38	1.25
.05	.9158	700	1.013	1.65	38	212	38	1.25

unitz1	muz1	sigmaz1	oratioz2	pz2	corrxz	pycondxmzm
.3	4.9	.3	3	.38	.4	.07
.3	4.9	.3	3	.38	.4	.07
.3	4.9	.3	3	.38	.4	.07
.3	4.9	.3	3	.38	.4	.07

As expected, when the sample size increases, the power increases toward 1.

For multiple values of parameters, the results are automatically displayed in a table, as we see above. For more examples of tables, see [PSS-2] **power, table**. If you wish to produce a power plot, see [PSS-2] **power, graph**.

◁

Computing effect size

By default, effect size δ for a logistic regression odds-ratio test is defined as the odds ratio for X: $\delta = \text{OR}_X = \exp(\beta_X/\text{u}_X)$. Sometimes, we want to know the smallest effect that can be detected with a level α test at a prespecified power and sample size.

To compute the effect size, you must specify the distribution of the covariate of interest X using the `x(distribution())` option; the distributions and coefficients or odds ratios of any nuisance covariates using `z1()`, `z2()`, and so on; parameter ζ_0 using the `pycondx0zm()` or `intercept()` option; the sample size using the `n()` option; and the power of the test using the `power()` option or the type II error probability using the `beta()` option. In addition, you must pick the level of the test and the direction of the effect. The level of the test is specified using the `alpha()` option, with a default of `alpha(0.05)`. The direction of the effect is specified using the `direction()` option; the default is `direction(upper)`, which means that $\text{OR}_X > 1$ or, equivalently, $\beta_X > 0$. Specifying `direction(lower)` means that $\text{OR}_X < 1$ and $\beta_X < 0$. The estimated minimum detectable effect size is reported as an odds ratio by default. To display it as a coefficient, specify `effect(coefficient)`.

▷ Example 6: Minimum detectable odds ratio

We continue with example 4, where we learned that a size 0.05 test with 600 subjects would have 87.07% power to detect an odds ratio of 1.65. How much larger would the odds ratio need to be to detect it with 90% power? We use `power logistic` to find out.

Instead of specifying the untransformed mean and standard deviation of serum cholesterol, we return to using standardized values as we did in example 1. The reason for this is twofold: First, there is no closed-form solution for the effect-size calculation, so it requires a nonlinear solver, which performs much more efficiently on the standardized values. Second, it allows us to use a trick to specify `pycondx0zm()` (which is allowed for effect-size calculations) instead of `pycondxmzm()` (which is not). The mean of X is 0 after standardizing, so $\Pr\{Y = 1|X = E(X), \mathbf{Z} = E(\mathbf{Z})\} = \Pr\{Y = 1|X = 0, \mathbf{Z} = E(\mathbf{Z})\}$, which means that we can specify `pycondx0zm(0.07)`.

```
. power logistic, x(distribution(normal 0 1))
> z1(distribution(normal 4.9 0.3) oratio(1.25, unit(sd)))
> z2(distribution(bernoulli 0.38) oratio(3))
> corrxz(0.4) pycondx0zm(.07) n(600) power(0.9)

Performing iteration ...

Estimated odds ratio for logistic regression odds-ratio test
Likelihood-ratio test
H0: OR_X = 1  versus  Ha: OR_X != 1

Study parameters:

        alpha =    0.0500
        power =    0.9000
            N =       600
       corrxz =    0.4000
   pycondx0zm =    0.0700

Covariate of interest X: Normal(mux, sigmax), bins = 23
          mux =    0.0000
       sigmax =    1.0000

Nuisance covariate Z1: Normal(muz1, sigmaz1), bins = 23
     oratioz1 =    1.2500
       unitz1 =    0.3000
         muz1 =    4.9000
      sigmaz1 =    0.3000

Nuisance covariate Z2: Bernoulli(pz2), bins = 2
     oratioz2 =    3.0000
          pz2 =    0.3800

Estimated effect size and odds ratio:

        delta =    1.7077  (odds ratio)
      oratiox =    1.7077
```

We see that a slightly larger odds ratio of 1.7077 can be detected with 90% power. The two normal covariates are discretized into only 23 bins each because the default value of `minbins()` for effect-size calculations is 1,000, and $23 \times 23 \times 2 = 1{,}058$.

In the above, we assumed the effect to be in the upper direction. By symmetry, there exists an effect size in the lower direction that can also be detected with 90% power. We specify `direction(lower)` to find it, and we add the `effect(coefficient)` option to display it as a coefficient instead of an odds ratio.

```
. power logistic, x(distribution(normal 0 1))
> z1(distribution(normal 4.9 0.3) oratio(1.25, unit(sd)))
> z2(distribution(bernoulli 0.38) oratio(3))
> corrxz(0.4) pycondx0zm(.07) n(600) power(0.9)
> direction(lower) effect(coefficient)

Performing iteration ...

Estimated coefficient for logistic regression coefficient test
Likelihood-ratio test
H0: beta_X = 0  versus  Ha: beta_X != 0

Study parameters:

        alpha =    0.0500
        power =    0.9000
            N =       600
       corrxz =    0.4000
   pycondx0zm =    0.0700

Covariate of interest X: Normal(mux, sigmax), bins = 23
          mux =    0.0000
       sigmax =    1.0000

Nuisance covariate Z1: Normal(muz1, sigmaz1), bins = 23
     oratioz1 =    1.2500
       unitz1 =    0.3000
         muz1 =    4.9000
      sigmaz1 =    0.3000

Nuisance covariate Z2: Bernoulli(pz2), bins = 2
     oratioz2 =    3.0000
          pz2 =    0.3800

Estimated effect size and coefficient:

        delta =   -0.5351  (coefficient)
        coefx =   -0.5351
```

By specifying `direction(lower)`, we anticipate coefficient $\beta_X < 0$, which is what we see. Had we omitted the `effect(coefficient)` option, the odds ratio $\text{OR}_X = \exp(\beta_X/1)$ would have been displayed as $\exp(-0.5351) = 0.5856$.

◁

Performing hypothesis tests with logistic regression

In this section, we briefly demonstrate the use of the `logit` command for testing logistic regression coefficients; see [R] **logit** for details. Alternatively, we could use the `logistic` command to perform logistic regression because `logistic` performs the same calculations as `logit` but reports odds ratios instead of coefficients; see [R] **logistic** for details, and see example 7 of [PSS-2] **power logistic onebin** for a demonstration of how `logistic` can be used to analyze the results of a pilot study.

▷ Example 7: Analyzing a pilot study

The `nlsw88` dataset contains employment data from the 1988 extract of the National Longitudinal Study of Young Women. We will treat this dataset as if it came from a pilot study investigating the relationship between union membership (`union`) and years of job experience (`ttl_exp`) and use it to plan a

follow-up study. Our target population is American young women, some of whom are married (`married`) and some of whom are college graduates (`collgrad`). These are both factors known to influence union membership, so we include them as nuisance covariates in our logistic regression.

```
. use https://www.stata-press.com/data/r19/nlsw88
(NLSW, 1988 extract)
. logit union ttl_exp married collgrad, nolog
Logistic regression                                     Number of obs =  1,878
                                                        LR chi2(3)    =  25.42
                                                        Prob > chi2   = 0.0000
Log likelihood = -1033.9131                             Pseudo R2     = 0.0121
```

union	Coefficient	Std. err.	z	P>|z|	[95% conf.	interval]
ttl_exp	.0213728	.0120162	1.78	0.075	-.0021785	.0449241
married	-.2328041	.1117316	-2.08	0.037	-.4517941	-.0138141
collgrad	.4777123	.1192228	4.01	0.000	.24404	.7113846
_cons	-1.380919	.1854974	-7.44	0.000	-1.744487	-1.017351

Years of job experience is our covariate of interest X, so the output tells us that β_X, the coefficient for `ttl_exp`, is 0.02. This suggests that women are more likely to be union members as they accumulate more job experience, but the evidence is not strong enough to reject $H_0 : \beta_X = 0$ at the 0.05 level. We want to design a follow-up study that has 80% power to detect a coefficient of 0.02 with a 0.05-level test, and we will use the parameter estimates from our pilot study to do so.

In addition to an estimate of β_X, the output of the `logit` command provides estimates of the coefficients for `married` ($\zeta_1 = -0.23$), `collgrad` ($\zeta_2 = 0.48$), and the logistic intercept `_cons` ($\zeta_0 = -1.38$). To use `power logistic`, we need additional information about the distributions of our covariates.

To visually examine the distribution of X covariate `ttl_exp`, we use the `histogram` command with the `normal` option to draw a histogram of `ttl_exp` values with a normal density for reference; see [R] **histogram** for details. We include the expression `if e(sample)` to restrict the computation so that it includes only the 1,878 participants whose data were used to fit the logistic regression; see **[U] 20.7 Specifying the estimation subsample** for details about `e(sample)`.

```
. histogram ttl_exp if e(sample), normal
(bin=32, start=.11538462, width=.89903845)
```

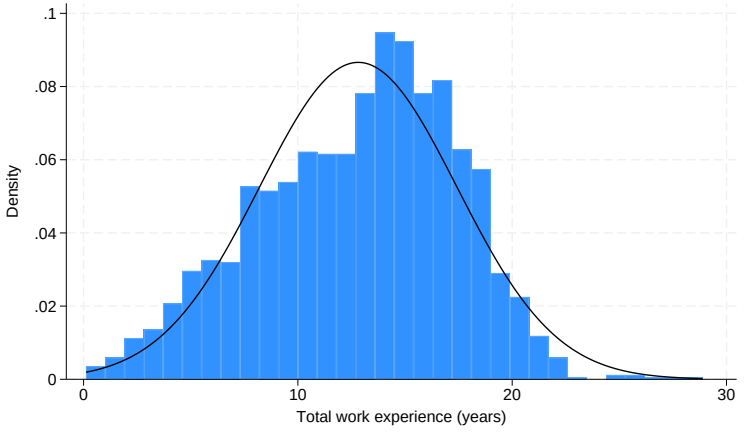

Figure 2. Histogram of total work experience

Total work experience is not quite normally distributed, but the fit is close enough to use a normal distribution to conduct a sensitivity analysis for sample-size calculations. Next we use the `summarize` command to calculate the means and standard deviations of our covariates; see [R] **summarize** for details.

```
. summarize ttl_exp married collgrad if e(sample)
```

Variable	Obs	Mean	Std. dev.	Min	Max
ttl_exp	1,878	12.81837	4.606392	.1153846	28.88461
married	1,878	.6506922	.4768783	0	1
collgrad	1,878	.2470714	.4314235	0	1

The mean of `ttl_exp` is 12.8, and the standard deviation is 4.6, but to account for uncertainty, we will perform a sensitivity analysis by specifying a numlist of values from 4 to 5 for the standard deviation of X. Binary covariates `married` and `collgrad` follow Bernoulli distributions where parameter p is equal to the mean, so we use 0.65 and 0.25 as their respective values of p.

The easiest way to calculate the multiple correlation coefficient between X and the **Z** covariates is to perform a linear regression of X on Z and take the square root of the coefficient of determination, R^2. We use the `regress` command to perform linear regression of `ttl_exp` on `married` and `collgrad` to calculate the coefficient of multiple correlation; see [R] **regress** for details.

```
. regress ttl_exp married collgrad if e(sample), notable
```

Source	SS	df	MS		
				Number of obs	= 1,878
				F(2, 1875)	= 14.73
Model	615.91641	2	307.958205	Prob > F	= 0.0000
Residual	39211.8662	1,875	20.9129953	R-squared	= 0.0155
				Adj R-squared	= 0.0144
Total	39827.7826	1,877	21.2188506	Root MSE	= 4.5731

```
. display "Multiple correlation coefficient: " sqrt(e(r2))
Multiple correlation coefficient: .12435631
```

Putting this all together, we calculate the required sample size for a 5% test with 80% power to detect a coefficient β_X of 0.02.

```
. power logistic, x(distribution(normal 12.8 (4(0.2)5)) coefficient(0.02))
> z1(distribution(bernoulli 0.65) coefficient(-0.23))
> z2(distribution(bernoulli 0.25) coefficient(0.48))
> corrxz(0.124) intercept(-1.38)

Estimated sample size for logistic regression coefficient test
Likelihood-ratio test
H0: beta_X = 0  versus  Ha: beta_X != 0

Covariate of interest X: Normal(mux, sigmax)
Nuisance covariates:
   Z1: Bernoulli(pz1)
   Z2: Bernoulli(pz2)
```

alpha	power	N	delta	coefx	mux	sigmax	coefz1	pz1
.05	.8	6,866	.02	.02	12.8	4	-.23	.65
.05	.8	6,228	.02	.02	12.8	4.2	-.23	.65
.05	.8	5,675	.02	.02	12.8	4.4	-.23	.65
.05	.8	5,192	.02	.02	12.8	4.6	-.23	.65
.05	.8	4,769	.02	.02	12.8	4.8	-.23	.65
.05	.8	4,395	.02	.02	12.8	5	-.23	.65

coefz2	pz2	corrxz	intercept
.48	.25	.124	-1.38
.48	.25	.124	-1.38
.48	.25	.124	-1.38
.48	.25	.124	-1.38
.48	.25	.124	-1.38
.48	.25	.124	-1.38

Depending on the standard deviation of X, the test will require between 4,395 and 6,866 participants to have 80% power. We have the budget to recruit 6,866 participants, so we perform a power analysis to see what the power of the test would be with a sample size of 6,866 over a range of standard deviations.

To display the result visually, we use the `graph` option.

```
. power logistic, x(distribution(normal 12.8 (4(0.2)5)) coefficient(0.02))
> z1(distribution(bernoulli 0.65) coefficient(-0.23))
> z2(distribution(bernoulli 0.25) coefficient(0.48))
> corrxz(0.124) intercept(-1.38) n(6866) graph
Covariate of interest X: Normal(mux, sigmax)
Nuisance covariates:
   Z1: Bernoulli(pz1)
   Z2: Bernoulli(pz2)
```

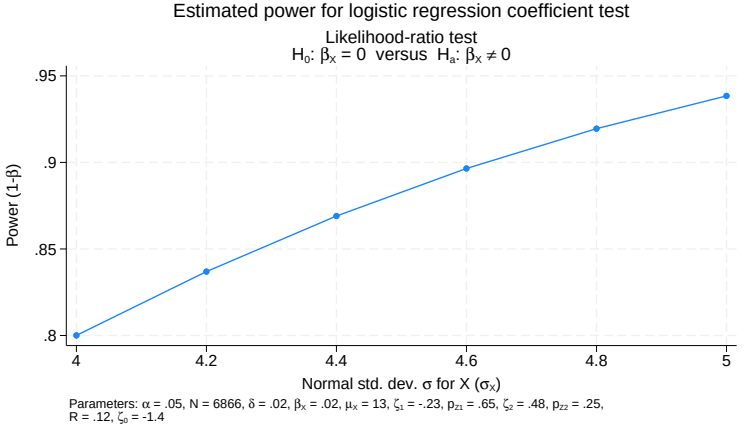

Figure 3. Power curve for a sample of 6,866

With 6,866 participants, the power of our test ranges from 80% when `ttl_exp` has a standard deviation of 4 to over 90% when the standard deviation is greater than 4.6.

Instead of specifying the intercept, we could calculate $\Pr\{Y = 1|X = E(X), \mathbf{Z} = E(\mathbf{Z})\} =$ `invlogit(`$\overline{X}\beta_X + \zeta_0 + \overline{Z}_1\zeta_1 + \overline{Z}_2\zeta_2$`) =` `invlogit(`$12.8 \times 0.02 - 1.38 + 0.65 \times -0.23 + 0.25 \times 0.48$`)` to specify `pycondxmzm(0.23985)`. Alternatively, we could calculate $\Pr\{Y = 1|X = 0, \mathbf{Z} = E(\mathbf{Z})\}$ = `invlogit(`$\zeta_0 + \overline{Z}_1\zeta_1 + \overline{Z}_2\zeta_2$`) =` `invlogit(`$-1.38 + 0.65 \times -0.23 + 0.25 \times 0.48$`)` to specify `pycondx0zm(0.19631)`. Either alternative parameterization will yield the same result; doing so is left as an exercise for the reader.

One detail that bears mentioning is that these power and sample-size calculations are for likelihood-ratio tests, but the `logit` and `logistic` commands report Wald tests of coefficients. Fortunately, an extensive simulation study by Bush (2015) demonstrates that sample-size requirements for Wald and likelihood-ratio tests of logistic regression coefficients are nearly identical. If you prefer a likelihood-ratio test, you can use the `lrtest` command; see [R] **lrtest** for details.

◁

Stored results

`power logistic` in the general case stores the following in `r()`:

Scalars

`r(alpha)`	significance level
`r(power)`	power
`r(beta)`	probability of a type II error
`r(delta)`	effect size
`r(N)`	sample size
`r(nfractional)`	`1` if `nfractional` is specified, `0` otherwise
`r(pycondxmzm)`	success probability of Y given mean values of X and $\mathbf{Z}$
`r(pycondx0zm)`	success probability of Y given $X = 0$ and mean values of covariates $\mathbf{Z}$
`r(intercept)`	intercept from logistic regression
`r(corrxz)`	correlation between X and $\mathbf{Z}$
`r(coefx)`	coefficient for X
`r(oratiox)`	odds ratio for X
`r(unitx)`	unit change in X for odds ratio
`r(nbinsx)`	number of bins for discretized X
`r(coefz#)`	coefficient for $Z_\#$ (if $Z_\#$ is specified)
`r(oratioz#)`	odds ratio for $Z_\#$
`r(unitz#)`	unit change in $Z_\#$ for odds ratio
`r(nbinsz#)`	number of bins for discretized $Z_\#$
`r(ax)`	parameter *a* of the distribution of X
`r(az#)`	parameter *a* of the distribution of $Z_\#$
`r(bx)`	parameter *b* of the distribution of X
`r(bz#)`	parameter *b* of the distribution of $Z_\#$
`r(mx)`	parameter *m* of the distribution of X
`r(mz#)`	parameter *m* of the distribution of $Z_\#$
`r(nx)`	parameter *n* of the distribution of X
`r(nz#)`	parameter *n* of the distribution of $Z_\#$
`r(px)`	parameter *p* of the distribution of X
`r(pz#)`	parameter *p* of the distribution of $Z_\#$
`r(sx)`	parameter *s* of the distribution of X
`r(sz#)`	parameter *s* of the distribution of $Z_\#$
`r(mux)`	parameter μ of the distribution of X
`r(muz#)`	parameter μ of the distribution of $Z_\#$
`r(sigmax)`	parameter σ of the distribution of X
`r(sigmaz#)`	parameter σ of the distribution of $Z_\#$
`r(v#x)`	parameter $v_\#$ of ordinal distribution of X
`r(v#z#)`	parameter $v_\#$ of ordinal distribution of $Z_\#$
`r(p#x)`	parameter $p_\#$ of ordinal distribution of X
`r(p#z#)`	parameter $p_\#$ of ordinal distribution of $Z_\#$
`r(nlx)`	number of levels of ordinal distribution of X
`r(nlz#)`	number of levels of ordinal distribution of $Z_\#$
`r(nbins)`	requested number of bins per covariate (if specified)
`r(minbins)`	minimum requested product of all bins
`r(totalbins)`	actual product of all bins
`r(separator)`	number of lines between separator lines in the table
`r(divider)`	`1` if `divider` is requested in the table, `0` otherwise

r(init)	initial value for odds ratio (if specified)
r(maxiter)	maximum number of iterations (for effect-size calculation)
r(iter)	number of iterations performed (for effect-size calculation)
r(tolerance)	requested parameter tolerance (for effect-size calculation)
r(deltax)	final parameter tolerance achieved (for effect-size calculation)
r(ftolerance)	requested distance of the objective function from zero (for effect-size calculation)
r(function)	final distance of the objective function from zero (for effect-size calculation)
r(converged)	1 if iteration algorithm converged, 0 otherwise (for effect-size calculation)
Macros	
r(type)	test
r(method)	logistic
r(direction)	upper or lower (for effect-size calculation)
r(columns)	displayed table columns
r(labels)	table column labels
r(widths)	table column widths
r(formats)	table column formats
r(distx)	distribution of X
r(distz#)	distribution of $Z_{\#}$ (if $Z_{\#}$ is specified)
Matrices	
r(pss_table)	table of results
r(ordinalx)	values and probabilities for ordinal covariate X (if specified)
r(ordinalz#)	values and probabilities for ordinal covariate $Z_{\#}$ (if specified)

Methods and formulas

Methods and formulas are presented under the following headings:

Coefficient tests in logistic regression

Shieh (2000a) used simulation to compare the performance of two sample-size formulas for coefficient tests in logistic regression: the method of Whittemore (1981) and that of Self, Mauritsen, and Ohara (1992). Shieh generalized the superior of those two methods, that of Self, Mauritsen, and Ohara, in Shieh (2000b), which provides the formulas implemented in `power logistic` for power, sample-size, and effect-size calculations for likelihood-ratio tests in logistic regression.

In practice, it is more common to use the Wald test of logistic regression coefficients than the likelihood-ratio test, so there has been some concern about whether these calculations are appropriate for use with a Wald test (Demidenko 2007). Demidenko notes that the Wald and likelihood-ratio tests have asymptotically equivalent type I errors and that they are "*locally* equivalent, so that the power functions are close when the alternative approaches the null" (Demidenko 2007, 3385). Nevertheless, Demidenko raises the point that the two tests are not globally equivalent, so there is no theoretical guarantee that the power functions will be similar under the alternative hypothesis. Thankfully, an extensive simulation study by Bush (2015) found little difference between the power curves of the Wald, likelihood-ratio, and score tests over a range of scenarios. Additionally, Bush compared the performance of seven sample-size formulas for logistic regression and determined that the method of Shieh (2000b) was consistently accurate, regardless of the test that was used.

Logistic regression

The logistic regression can be written as

$$\Pr(y_i = 1|x_i, \mathbf{z}_i) = H(x_i\beta_X + \zeta_0 + \mathbf{z}_i\boldsymbol{\zeta}_Z) \qquad i = 1, 2, \ldots, n$$

where x_i is the observed value of covariate of interest X for subject i, β_X is the X coefficient, ζ_0 is the logistic intercept, $\mathbf{z}_i = (z_{1i}, z_{2i}, \ldots, z_{Ki})$ is the row vector of observed values of K nuisance covariates Z_1 through Z_K for subject i, $\boldsymbol{\zeta}_Z = (\zeta_1, \zeta_2, \ldots, \zeta_K)'$ is the column vector of K coefficients for the nuisance covariates, and n is the sample size. Function $H(\eta) = \{1+\exp(-\eta)\}^{-1}$ is the logistic distribution function.

The effect of covariate X can also be expressed in terms of an odds ratio, $\text{OR}_X = \exp(\beta_X \text{u}_X)$, where $\text{u}_X > 0$ is the unit change in X for the odds ratio. The null hypothesis is $H_0 : \beta_X = 0$, which can also be expressed as $H_0 : \text{OR}_X = 1$. The alternative hypothesis is $H_a : \beta_X \neq 0$ or, equivalently, $H_a : \text{OR}_X \neq 1$.

Parameters $\zeta_1, \zeta_2, \ldots, \zeta_K$ must be specified as coefficients or odds ratios in the respective `z#()` options. Intercept ζ_0 may be specified directly in the `intercept()` option or the information necessary to calculate ζ_0 may be specified using either the `pycondxmzm()` or `pycondx0zm()` option. Coefficient β_X may be specified as a coefficient or odds ratio, or the information necessary to calculate β_X may be specified using the `pycondxmzm()` option. When one or both of `pycondxmzm()` and `pycondx0zm()` are specified, we solve for the unknown parameters (β_X, ζ_0, or both) using the equations

$$\begin{aligned} \Pr\{Y = 1|X = E(X), \mathbf{Z} = E(\mathbf{Z})\} &= H\{E(X)\beta_X + \zeta_0 + E(\mathbf{Z})\boldsymbol{\zeta}_Z\} \\ \Pr\{Y = 1|X = 0, \mathbf{Z} = E(\mathbf{Z})\} &= H\{\zeta_0 + E(\mathbf{Z})\boldsymbol{\zeta}_Z\} \end{aligned}$$

where expected values $E(X)$ and $E(\mathbf{Z})$ are determined based on the specified distributions.

Power, sample-size, and effect-size calculations

Shieh (2000b) builds on the work of Self, Mauritsen, and Ohara (1992) and Self and Mauritsen (1988) to estimate the distribution of the likelihood-ratio statistic: $2\{l(\hat{\beta}_X, \hat{\mathbf{c}}) - l(0, \hat{\mathbf{c}}_0)\}$. Here $l(\cdot)$ is the log-likelihood function for the logistic regression, and $\mathbf{c} = (\zeta_0, \boldsymbol{\zeta}_Z{}')$ is a vector of nuisance parameters. $\hat{\beta}_X$ and $\hat{\mathbf{c}}$ are the maximum likelihood estimates of β_X and $\mathbf{c}$ under the alternative hypothesis, and $\hat{\mathbf{c}}_0$ is the maximum likelihood estimate of $\mathbf{c}$ under the null hypothesis. If the null hypothesis is not true, $\hat{\mathbf{c}}_0$ is not a consistent estimate of $\mathbf{c}$ but instead converges to $\mathbf{c}_0^* = (\zeta_0^*, \boldsymbol{\zeta}_Z{}')$, where $\zeta_0^* = \zeta_0 + E(X)\beta_X$, as described in Self and Mauritsen (1988, eq. 2.2). For a full decomposition of the likelihood-ratio statistic, see Shieh (2000b, 1193).

To calculate sample size, we begin by specifying the desired size of the test (also known as the type I error, α) and the desired power (which equals $1 - \beta$, where β is the desired type II error of the test). We equate the $(1 - \alpha)100$th percentile of a central χ^2 distribution with 1 degree of freedom to the $\beta 100$th percentile of a noncentral χ^2 distribution with noncentrality parameter $\lambda = n\Delta^*$. Here $\Delta^* = 2E(W^*)$, and we define W^* as

$$W^* = H(\eta)(\eta - \eta^*) - \log\{1 + \exp(\eta)\} + \log\{1 + \exp(\eta^*)\}$$

where $\eta = X\beta_X + (1, \mathbf{Z})\mathbf{c}'$ and $\eta^* = (1, \mathbf{Z})\mathbf{c}_0^{*\prime}$. Sample size is calculated as $n = \lambda/\{\Delta^*(1 - R^2)\}$, where R is the coefficient of (multiple) correlation between X and $\mathbf{Z}$. R is specified using the `corrxz()` option, with a default of `corrxz(0)`; the adjustment for correlated covariates is based on Whittemore (1981). Power is computed similarly by starting with a known value of n and solving for the power required to yield the desired value of λ.

There is no closed-form expression that can be used to calculate effect size δ, either coefficient β_X or odds ratio OR_X. Effect size is estimated iteratively, and the default starting value for OR_X is 1.5 with the default `direction(upper)` or 0.67 with `direction(lower)`; see [M-5] `solvenl()` for details. You can use the `init()` option to specify a starting odds ratio for the nonlinear solver. You can control the iteration process with the `iterate()`, `tolerance()`, `ftolerance()`, `log`, `nolog`, `dots`, and `nodots` options.

Discretization

To approximate the expected value of W^* when calculating Δ^*, we discretize the covariates into bins. Bernoulli, binomial, and ordinal random variables have a fixed number of possible outcomes, so they always use one bin per outcome. For covariates with other distributions, the number of bins is determined as follows. If the `nbins()` option is specified, then the specified number of bins is used. If the `nbins()` suboption is specified in the `distribution()` option, this number overrides the number specified with the `nbins()` option for the specified covariate. All other covariates for which the number of bins was not specified are assigned an equal number of bins, such that the product of bins is greater than or equal to `minbins()`. The default value for `minbins()` is 10,000 for power and sample-size calculations and 1,000 for effect-size calculations. The formula for the total number of bins is $\text{B}_{\text{tot}} = \text{B}_X \prod_{k=1}^{K} \text{B}_{Z_k}$, where $\text{B}_{\text{tot}} \leq 100{,}000{,}000$ is the product of the bins for all covariates, B_X is the number of bins for discretized X, and B_{Z_k} is the number of bins for discretized Z_k.

For Bernoulli, binomial, and ordinal random variables, the probability of each outcome is defined by the distribution. For all other distributions, bins are assigned such that their midpoints yield quantiles of equal probability. To calculate the expectation, we use all B_{tot} possible combinations of binned variables, with each combination weighted by its probability: $E(W^*) \approx \sum_{c=1}^{\text{B}_{\text{tot}}} W_c^* \pi_c$, where W_c^* is W^* calculated at combination c and π_c is the probability of observing combination c, calculated under the assumption of independence between covariates.

References

Bush, S. 2015. Sample size determination for logistic regression: A simulation study. *Communications in Statistics—Simulation and Computation* 44: 360–373. https://doi.org/10.1080/03610918.2013.777458.

Demidenko, E. 2007. Sample size determination for logistic regression revisited. *Statistics in Medicine* 26: 3385–3397. https://doi.org/10.1002/sim.2771.

Hulley, S. B., R. H. Rosenman, R. D. Bawol, and R. J. Brand. 1980. Epidemiology as a guide to clinical decisions—the association between triglyceride and coronary heart disease. *New England Journal of Medicine* 302: 1383–1389. https://doi.org/10.1056/nejm198006193022503.

Self, S. G., and R. H. Mauritsen. 1988. Power/sample size calculations for generalized linear models. *Biometrika* 44: 79–86. https://doi.org/10.2307/2531897.

Self, S. G., R. H. Mauritsen, and J. Ohara. 1992. Power calculations for likelihood ratio tests in generalized linear models. *Biometrika* 48: 31–39. https://doi.org/10.2307/2532736.

Shieh, G. 2000a. A comparison of two approaches for power and sample size calculations in logistic regression models. *Communications in Statistics—Simulation and Computation* 29: 763–791. https://doi.org/10.1080/03610910008813639.

———. 2000b. On power and sample size calculations for likelihood ratio tests in generalized linear models. *Biometrics* 56: 1192–1196. https://doi.org/10.1111/j.0006-341x.2000.01192.x.

Whittemore, A. S. 1981. Sample size for logistic regression with small response probability. *Journal of the American Statistical Association* 76: 27–32. https://doi.org/10.2307/2287036.

Also see

[PSS-2] **power logistic** — Power analysis for logistic regression+

[PSS-2] **power logistic onebin** — Power analysis for logistic regression with one binary covariate+

[PSS-2] **power logistic twobin** — Power analysis for logistic regression with two binary covariates+

[PSS-2] **power** — Power and sample-size analysis for hypothesis tests

[PSS-2] **power, graph** — Graph results from the power command

[PSS-2] **power, table** — Produce table of results from the power command

[PSS-5] **Glossary**

[R] **logistic** — Logistic regression, reporting odds ratios

[R] **logit** — Logistic regression, reporting coefficients

[R] **lrtest** — Likelihood-ratio test after estimation