which relaxes the assumption of the independence of irrelevant alternatives. Finally, we present the rank-ordered logistic regression model, in which the outcome is the ranking of a set of alternatives.

- **Chapter 8** on *count outcomes* presents the Poisson and negative binomial regression models. We show how to test the Poisson model's assumption of equidispersion and how to incorporate differences in exposure time into the models. The next two models, the zero-truncated Poisson and negative binomial models, deal with the common problem of having no zeros in your data. We combine these models with the logit model to construct the hurdle model for counts. We conclude by considering two zero-inflated models that are designed for data with zero counts.
- **Chapter 9** covers more topics that extend material presented earlier. We discuss the use and interpretation of categorical independent variables, interactions, and nonlinear terms. We also provide tips on how to use Stata more efficiently and effectively.

4 Models for binary outcomes

Regression models for binary outcomes are the foundation from which more complex models for ordinal, nominal, and count models can be derived. Ordinal and nominal regression models are equivalent to the simultaneous estimation of a series of binary outcomes. Although the link is less direct in count models, the Poisson distribution can be derived as the outcome of many binary trials. More importantly for our purposes, the zero-inflated count models that we discuss in chapter 8 merge a binary logit or probit with a standard Poisson or negative binomial model. Consequently, the principles of fitting, testing, and interpreting binary models provide tools that can be readily adapted to models in later chapters. Thus although each chapter is largely self-contained, this chapter provides somewhat more detailed explanations than later chapters. As a result, even if your interests are in models for ordinal, nominal, or count outcomes, you should benefit from reading this chapter.

Binary dependent variables have two values, typically coded as 0 for a negative outcome (i.e., the event did not occur) and 1 as a positive outcome (i.e., the event did occur). Binary outcomes are ubiquitous, and examples come easily to mind. Did a person vote? Is a manufacturing firm unionized? Is someone a feminist or nonfeminist? Did a startup company go bankrupt? Five years after a person was diagnosed with cancer, is he or she still alive? Was a purchased item returned to the store or kept?

Regression models for binary outcomes allow a researcher to explore how each explanatory variable affects the probability of the event occurring. We focus on the two most often used models, the binary logit and binary probit models, referred to jointly as the *binary regression model* (BRM). Because the model is nonlinear, the magnitude of the change in the outcome probability that is associated with a given change in one of the independent variables depends on the levels of all the independent variables. The challenge of interpretation is to find a summary of the way in which changes in the independent variables are associated with changes in the outcome that best reflect the key substantive processes without overwhelming yourself or your readers with distracting detail.

The chapter begins by reviewing the mathematical structure of binary models. We then examine statistical testing and fit, and finally, methods of interpretation. These discussions are intended as a review for those who are familiar with the models. For a complete discussion, see Long (1997). You can obtain sample do-files and data files that reproduce the examples in this chapter by downloading the `spost9_do` and `spost9_ado` packages (see chapter 1 for details).

4.1 The statistical model

There are three ways to derive the BRM, with each method leading to the same mathematical model. First, an unobserved or latent variable can be hypothesized along with a measurement model relating the latent variable to the observed, binary outcome. Second, the model can be constructed as a probability model. Third, the model can be generated as a random utility or discrete-choice model. This last approach is not considered in our review; see Long (1997, 155–156) for an introduction or Pudney (1989) for a detailed discussion.

4.1.1 A latent-variable model

Assume a *latent* or unobserved variable y^* ranging from $-\infty$ to ∞ that is related to the observed independent variables by the structural equation

$$y_i^* = \mathbf{x}_i\boldsymbol{\beta} + \varepsilon_i$$

where i indicates the observation and ε is a random error. For one independent variable, we can simplify the notation to

$$y_i^* = \alpha + \beta x_i + \varepsilon_i$$

These equations are identical to those for the linear regression model except that the dependent variable is unobserved.

The link between the observed binary y and the latent y^* is made with a simple measurement equation:

$$y_i = \begin{cases} 1 & \text{if } y_i^* > 0 \\ 0 & \text{if } y_i^* \leq 0 \end{cases}$$

Cases with positive values of y^* are observed as $y = 1$, whereas cases with negative or zero values of y^* are observed as $y = 0$.

Imagine a survey item that asks respondents if they agree or disagree with the proposition that "a working mother can establish just as warm and secure a relationship with her children as a mother who does not work". Obviously, respondents vary greatly in their opinions on this issue. Some people adamantly agree with the proposition, some adamantly disagree, and still others have only weak opinions one way or the other. We can imagine an underlying continuum of possible responses to this item, with every respondent having some value on this continuum (i.e., some value of y^*). Those respondents whose value of y^* is positive answer "agree" to the survey question ($y = 1$), and those whose value of y^* is 0 or negative answer "disagree" ($y = 0$). A shift in a respondent's opinion might move them from agreeing strongly with the position to agreeing weakly with the position, which would not change the response we observe. Or, the respondent might move from weakly agreeing to weakly disagreeing, in which case we would observe a change from $y = 1$ to $y = 0$.

Consider a second example, which we use throughout this chapter. Let $y = 1$ if a woman is in the paid labor force and $y = 0$ if she is not. The independent variables

include variables such as number of children, education, and expected wages. Not all women in the labor force ($y = 1$) are there with the same certainty. One woman might be close to leaving the labor force, whereas another woman could be firm in her decision to work. In both cases, we observe $y = 1$. The idea of a latent y^* is that an underlying *propensity to work* generates the observed state. Again although we cannot directly observe the propensity, at some point a change in y^* results in a change in what we observe, namely, whether the woman is in the labor force.

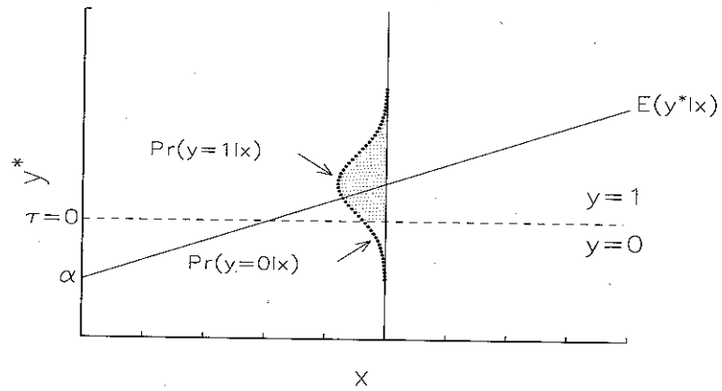


Figure 4.1: Relationship between latent variable y^* and $\Pr(y = 1)$ for the BRM.

The latent-variable model for binary outcomes is shown in figure 4.1 for one independent variable. For a given value of x, we see that

$$\Pr(y = 1 \mid x) = \Pr(y^* > 0 \mid x)$$

Substituting the structural model and rearranging terms,

$$\Pr(y = 1 \mid x) = \Pr(\varepsilon > -[\alpha + \beta x] \mid x) \tag{4.1}$$

This equation shows that the probability depends on the distribution of the error, ε.

Two distributions of ε are commonly assumed, both with an assumed mean of 0. First, ε is assumed to be distributed normally with $\mathrm{Var}(\varepsilon) = 1$. This leads to the binary probit model, in which (4.1) becomes

$$\Pr(y = 1 \mid x) = \int_{-\infty}^{\alpha+\beta x} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{t^2}{2}\right) dt$$

Alternatively, ε is assumed to be distributed logistically with $\mathrm{Var}(\varepsilon) = \pi^2/3$, leading to the binary logit model with the simpler equation

$$\Pr(y = 1 \mid x) = \frac{\exp(\alpha + \beta x)}{1 + \exp(\alpha + \beta x)} \tag{4.2}$$

The peculiar value assumed for $\text{Var}(\varepsilon)$ in the logit model illustrates a basic point about the identification of models with latent outcomes. In the LRM, $\text{Var}(\varepsilon)$ can be estimated because y is observed. For the BRM, the value of $\text{Var}(\varepsilon)$ must be assumed because the dependent variable is unobserved. The model is unidentified unless an assumption is made about the variance of the errors. For probit, we assume $\text{Var}(\varepsilon) = 1$ because this leads to a simple form of the model. If a different value were assumed, this would simply change the values of the structural coefficients uniformly. In the logit model, the variance is set to $\pi^2/3$ because this leads to the simple form in (4.2). Although the value assumed for $\text{Var}(\varepsilon)$ is arbitrary, the value chosen does *not* affect the computed value of the probability (see Long 1997, 49–50 for a simple proof). In effect, changing the assumed variance affects the spread of the distribution but not the proportion of the distribution above or below the threshold.

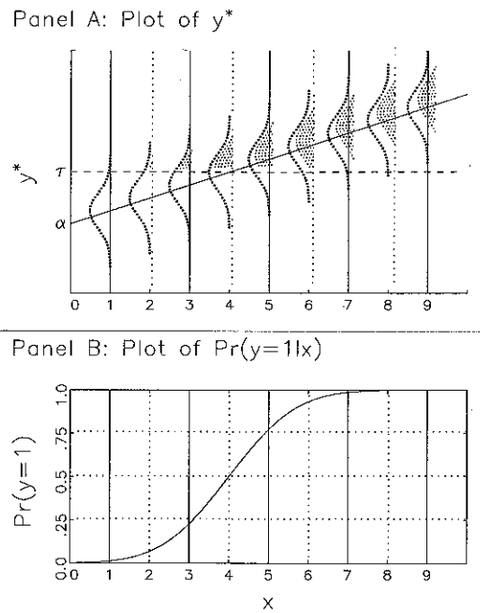


Figure 4.2: Relationship between the linear model $y^* = \alpha + \beta x + \varepsilon$ and the nonlinear probability model $\Pr(y = 1 \mid x) = F(\alpha + \beta x)$.

For both models, the probability of the event occurring is the cumulative density function (cdf) of ε evaluated at given values of the independent variables:

$$\Pr(y = 1 \mid \mathbf{x}) = F(\mathbf{x}\boldsymbol{\beta})$$

where F is the normal cdf Φ for the probit model and the logistic cdf Λ for the logit model. The relationship between the linear latent-variable model and the resulting nonlinear probability model is shown in figure 4.2 for a model with one independent variable. Panel A shows the error distribution for nine values of x, which we have labeled 1, 2, ..., 9. The area where $y^* > 0$ corresponds to $\Pr(y = 1 \mid x)$ and has been shaded. Panel B plots $\Pr(y = 1 \mid x)$ corresponding to the shaded regions in panel A. As we move from 1 to 2, only a portion of the thin tail crosses the threshold in panel A, resulting in a small change in $\Pr(y = 1 \mid x)$ in panel B. As we move from 2 to 3 to 4, thicker regions of the error distribution slide over the threshold, and the increase in $\Pr(y = 1 \mid x)$ becomes larger. The resulting curve is the well-known S-curve associated with the BRM.

4.1.2 A nonlinear probability model

Can all binary dependent variables be conceptualized as observed manifestations of some underlying latent propensity? Although philosophically interesting, perhaps, the question is of little practical importance, as the BRM can also be derived without appealing to a latent variable. This is done by specifying a nonlinear model relating the x's to the probability of an event. Following Theil (1970), the logit model can be derived by constructing a model in which the predicted $\Pr(y = 1 \mid \mathbf{x})$ is forced to be within the range 0 to 1. For example, in the linear probability model,

$$\Pr(y = 1 \mid \mathbf{x}) = \mathbf{x}\boldsymbol{\beta} + \varepsilon$$

the predicted probabilities can be greater than 1 and less than 0. To constrain the predictions to the range 0 to 1, we first transform the probability into the *odds*,

$$\Omega(\mathbf{x}) = \frac{\Pr(y = 1 \mid \mathbf{x})}{\Pr(y = 0 \mid \mathbf{x})} = \frac{\Pr(y = 1 \mid \mathbf{x})}{1 - \Pr(y = 1 \mid \mathbf{x})}$$

which indicate how often something happens ($y = 1$) relative to how often it does not happen ($y = 0$), and range from 0 when $\Pr(y = 1 \mid \mathbf{x}) = 0$ to ∞ when $\Pr(y = 1 \mid \mathbf{x}) = 1$. The log of the odds, or *logit*, ranges from $-\infty$ to ∞. This range suggests a model that is *linear in the logit*:

$$\ln \Omega(\mathbf{x}) = \mathbf{x}\boldsymbol{\beta}$$

This equation can be shown to be equivalent to the logit model from (4.2). Interpretation of this form of the logit model often focuses on factor changes in the odds, which are discussed below.

Other binary regression models are created by choosing functions of $\mathbf{x}\boldsymbol{\beta}$ that range from 0 to 1. Cumulative distribution functions have this property and readily provide several examples. For example, the cdf for the standard normal distribution results in the probit model.

4.2 Estimation using logit and probit

Logit and probit can be fitted with the following commands and their basic options:

`logit` *depvar* [*indepvars*] [*if*] [*in*] [*weight*] [, `noconstant` `level(#)` `or` `robust` `cluster(`*varname*`)` `nolog`]

`probit` *depvar* [*indepvars*] [*if*] [*in*] [*weight*] [, `noconstant` `level(#)` `robust` `cluster(`*varname*`)` `nolog`]

We have never had a problem with either of these models converging, even with small samples and data with wide variation in scaling.

Variable lists

depvar is the dependent variable. *indepvars* is a list of independent variables. If *indepvars* is not included, Stata fits a model with only an intercept.

Warning For binary models, Stata defines observations in which *depvar* = 0 as negative outcomes and observations in which *depvar* equals *any* other nonmissing value (including negative values) as positive outcomes. To avoid possible confusion, we urge you to explicitly create a 0/1 variable for use as *depvar*.

Specifying the estimation sample

if and in qualifiers can be used to restrict the estimation sample. For example, if you wanted to fit a logit model for only women who went to college (as indicated by the variable `wc`), you could specify `logit lfp k5 k618 age hc lwg if wc==1`.

Listwise deletion Stata excludes cases in which there are missing values for any of the variables in the model. Accordingly, if two models are fitted using the same dataset but have different sets of independent variables, it is possible to have different samples. We recommend that you use `mark` and `markout` (discussed in chapter 3) to explicitly remove cases with missing data.

Weights

Both `logit` and `probit` can be used with `fweights`, `pweights`, and `iweights`. In chapter 3, we provide a brief discussion of the different types of weights and how weighting variables are specified.

Options

`noconstant` specifies that the model should not have a constant term. This would rarely be used for these models.

`level(#)` specifies the level of the confidence interval. By default, Stata provides 95% confidence intervals for estimated coefficients. You can also change the default level, say to a 90% interval, with the command `set level 90`.

`or` (`logit` only) reports the "odds ratios" defined as $\exp(\widehat{\beta})$. Standard errors and confidence intervals are similarly transformed. Alternatively, our `listcoef` command can be used.

`robust` indicates that robust variance estimates are to be used. When `cluster()` is specified, robust standard errors are used automatically. We provide a brief general discussion of these options in chapter 3.

`cluster(`*varname*`)` specifies that the observations are independent across the groups specified by unique values of *varname* but not necessarily within the groups.

`nolog` suppresses the iteration history.

Example

Our example is from Mroz's (1987) study of the labor force participation of women, using data from the 1976 Panel Study of Income Dynamics.[1] The sample consists of 753 white, married women between the ages of 30 and 60 years. The dependent variable `lfp` equals 1 if a woman is employed and otherwise equals 0. Because we have assigned variable labels, a complete description of the data can be obtained using `describe` and `summarize`:

```
. use http://www.stata-press.com/data/lf2/binlfp2, clear
(Data from 1976 PSID-T Mroz)
. describe lfp k5 k618 age wc hc lwg inc
```

variable name	storage type	display format	value label	variable label
lfp	byte	%9.0g	lfplbl	Paid Labor Force: 1=yes 0=no
k5	byte	%9.0g		# kids < 6
k618	byte	%9.0g		# kids 6-18
age	byte	%9.0g		Wife's age in years
wc	byte	%9.0g	collbl	Wife College: 1=yes 0=no
hc	byte	%9.0g	collbl	Husband College: 1=yes 0=no
lwg	float	%9.0g		Log of wife's estimated wages
inc	float	%9.0g		Family income excluding wife's

1. These data were generously made available by Thomas Mroz.

```
. summarize lfp k5 k618 age wc hc lwg inc
```

Variable	Obs	Mean	Std. Dev.	Min	Max
lfp	753	.5683931	.4956295	0	1
k5	753	.2377158	.523959	0	3
k618	753	1.353254	1.319874	0	8
age	753	42.53785	8.072574	30	60
wc	753	.2815405	.4500494	0	1
hc	753	.3917663	.4884694	0	1
lwg	753	1.097115	.5875564	-2.054124	3.218876
inc	753	20.12897	11.6348	-.0290001	96

Using these data, we fitted the model

$$\Pr(\texttt{lfp}=1) = F(\beta_0 + \beta_{\texttt{k5}}\texttt{k5} + \beta_{\texttt{k618}}\texttt{k618} + \beta_{\texttt{age}}\texttt{age} + \beta_{\texttt{wc}}\texttt{wc} + \beta_{\texttt{hc}}\texttt{hc} + \beta_{\texttt{lwg}}\texttt{lwg} + \beta_{\texttt{inc}}\texttt{inc})$$

with both the `logit` and `probit` commands, and then we created a table of results with `estimates table`:

```
. logit lfp k5 k618 age wc hc lwg inc, nolog
Logistic regression                             Number of obs   =        753
                                                LR chi2(7)      =     124.48
                                                Prob > chi2     =     0.0000
Log likelihood = -452.63296                     Pseudo R2       =     0.1209

------------------------------------------------------------------------------
         lfp |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
          k5 |  -1.462913   .1970006    -7.43   0.000    -1.849027   -1.076799
        k618 |  -.0645707   .0680008    -0.95   0.342    -.1978499    .0687085
         age |  -.0628706   .0127831    -4.92   0.000    -.0879249   -.0378162
          wc |   .8072738   .2299799     3.51   0.000     .3565215    1.258026
          hc |   .1117336   .2060397     0.54   0.588    -.2920969     .515564
         lwg |   .6046931   .1508176     4.01   0.000     .3090961    .9002901
         inc |  -.0344464   .0082084    -4.20   0.000    -.0505346   -.0183583
       _cons |    3.18214   .6443751     4.94   0.000     1.919188    4.445092
------------------------------------------------------------------------------

. estimates store logit
```



```
. probit lfp k5 k618 age wc hc lwg inc, nolog
Probit regression                               Number of obs   =        753
                                                LR chi2(7)      =     124.36
                                                Prob > chi2     =     0.0000
Log likelihood = -452.69496                     Pseudo R2       =     0.1208

------------------------------------------------------------------------------
         lfp |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
          k5 |  -.8747112   .1135583    -7.70   0.000    -1.097281   -.6521411
        k618 |  -.0385945   .0404893    -0.95   0.340    -.117952    .0407631
         age |  -.0378235   .0076093    -4.97   0.000    -.0527375   -.0229095
          wc |   .4883144   .1354873     3.60   0.000     .2227642    .7538645
          hc |   .0571704   .1240052     0.46   0.645    -.1858754    .3002161
         lwg |   .3656287   .0877792     4.17   0.000     .1935847    .5376727
         inc |  -.020525    .0047769    -4.30   0.000    -.0298875   -.0111626
       _cons |   1.918422   .3806536     5.04   0.000     1.172355    2.66449
------------------------------------------------------------------------------

. estimates store probit
```

Although the iteration log was suppressed by the `nolog` option, the value of the log likelihood at convergence is listed as `Log likelihood`. The information in the header and table of coefficients is in the same form as discussed in chapter 3.

We can use `estimates table` to create a table that combines the results:

```
. estimates table logit probit, b(%9.3f) t label varwidth(30)
```

Variable	logit	probit
# kids < 6	-1.463	-0.875
	-7.43	-7.70
# kids 6-18	-0.065	-0.039
	-0.95	-0.95
Wife's age in years	-0.063	-0.038
	-4.92	-4.97
Wife College: 1=yes 0=no	0.807	0.488
	3.51	3.60
Husband College: 1=yes 0=no	0.112	0.057
	0.54	0.46
Log of wife's estimated wages	0.605	0.366
	4.01	4.17
Family income excluding wife's	-0.034	-0.021
	-4.20	-4.30
Constant	3.182	1.918
	4.94	5.04

legend: b/t

The estimated coefficients differ from logit to probit by a factor of about 1.7. For example, the ratio of the logit to probit coefficient for `k5` is 1.67 and for `inc` is 1.68. This illustrates how the magnitudes of the coefficients are affected by the assumed $\text{Var}(\varepsilon)$. The exception to the ratio of 1.7 is the coefficient for `hc`. This estimate has a great deal of sampling variability (i.e., a large standard error), and in such cases, the 1.7 rule often does not hold. Values of the z-tests are quite similar because they are not

affected by the assumed Var(ε). The z-test statistics are not exactly the same because the two models assume different distributions of the errors.

4.2.1 Observations predicted perfectly

ML estimation is not possible when the dependent variable does not vary within one of the categories of an independent variable. Say that you are fitting a logit model predicting whether a person voted in the last election, `vote`, and that one of the independent variables is whether the person is enrolled in college, `college`. If you had a small number of college students in your sample, it is possible that none of them voted in the last election. That is, `vote==0` every time `college==1`. The model cannot be fitted because the coefficient for `college` is effectively negative infinity. Stata's solution is to drop the variable `college` along with all observations where `college==1`. For example,

```
. logit vote college phd, nolog
Note: college!=0 predicts failure perfectly
      college dropped and 4 obs not used

Logistic regression                             Number of obs   =        299
  (output omitted)
```

4.3 Hypothesis testing with test and lrtest

Hypothesis tests of regression coefficients can be conducted with the z-statistics in the estimation output, with `test` for Wald tests of simple and complex hypotheses, and with `lrtest` for the corresponding likelihood-ratio tests. We consider the use of each of these to test hypotheses involving only one coefficient, and then we show you how both `test` and `lrtest` can be used to test hypotheses involving multiple coefficients.

4.3.1 Testing individual coefficients

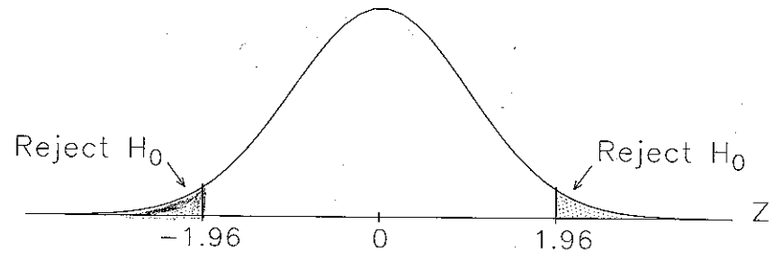
If the assumptions of the model hold, the ML estimators (e.g., the estimates produced by `logit` or `probit`) are distributed asymptotically normally:

$$\widehat{\beta}_k \stackrel{a}{\sim} \mathcal{N}\left(\beta_k, \sigma^2_{\widehat{\beta}_k}\right)$$

The hypothesis H_0: $\beta_k = \beta^*$ can be tested with the z-statistic:

$$z = \frac{\widehat{\beta}_k - \beta^*}{\widehat{\sigma}_{\widehat{\beta}_k}}$$

z is included in the output from `logit` and `probit`. Under the assumptions justifying ML, if H_0 is true, then z is distributed approximately normally with a mean of zero and a variance of one for large samples. This is shown in the following figure, where the shading shows the rejection region for a two-tailed test at the .05 level:



For example, consider the results for variable `k5` from the `logit` output generated in section 4.2:

lfp	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
k5	-1.462913	.1970006	-7.43	0.000	-1.849027	-1.076799

(output omitted)

We conclude that having young children has a significant effect on the probability of working ($z = -7.43$, $p < 0.01$ for a two-tailed test).

One- and two-tailed tests

The probability levels in the output for estimation commands are for two-tailed tests. That is, the result corresponds to the area of the curve that is either greater than $|z|$ or less than $-|z|$. When past research or theory suggests the sign of the coefficient, a one-tailed test can be used, and H_0 is rejected only when z is in the *expected* tail. For example, assume that my theory proposes that having children can have only a negative effect on labor force participation. For `k618`, $z = -0.95$ and `P > | z |` is .342. This is the proportion of the sampling distribution for z that is less than -0.95 or greater than 0.95. Because we want a one-tailed test, and the coefficient is in the expected direction, we want only the proportion of the distribution that is less than -0.95, which is .342/2 = .171. We conclude that having older children does not significantly affect a woman's probability of working ($z = -0.95$, $p = .17$ for a one-tailed test).

You should divide `P > | z |` by 2 only when the estimated coefficient is in the expected direction. Suppose I am testing a theory that having a husband who went to college has a negative effect on labor force participation, but the estimated coefficient is *positive* with $z = 0.542$ and `P > | z |` is .588. The one-tailed significance level would be the percentage of the distribution less than .542 (not the percentage less than $-.542$), which is equal to $1 - (.588/2) = .706$, not $.588/2 = .294$. We conclude that having a husband who attends college does not significantly affect a woman's probability of working ($z = 0.542$, $p = .71$ for a one-tailed test).

Testing single coefficients using test

The z-test included in the output of estimation commands is a Wald test, which can also be computed using `test`. For example, to test H_0: $\beta_{k5} = 0$,

```
. test k5
 ( 1)  k5 = 0
           chi2(  1) =   55.14
         Prob > chi2 =    0.0000
```

We can conclude that the effect of having young children on the probability of entering the labor force is significant at the .01 level ($X^2 = 55.14$, $df = 1$, $p < .01$).

The value of a chi-squared test with 1 degree of freedom is identical to the square of the corresponding z-test. For example, using Stata's `display` as a calculator

```
. display sqrt(55.14)
7.4256313
```

This corresponds to -7.43 from the `logit` output. Some packages, such as SAS, present chi-squared tests rather than the corresponding z-test.

Testing single coefficients using lrtest

An LR test is computed by comparing the log likelihood from a full model with that of a restricted model. To test a single coefficient, we begin by fitting the full model and storing the results:

```
. logit lfp k5 k618 age wc hc lwg inc, nolog
Logistic regression                             Number of obs   =        753
                                                LR chi2(7)      =     124.48
                                                Prob > chi2     =     0.0000
Log likelihood = -452.63296                     Pseudo R2       =     0.1209
  (output omitted)
. estimates store fmodel
```

Then we fit the model without k5 and run `lrtest`:

```
. logit lfp k618 age wc hc lwg inc, nolog
Logistic regression                             Number of obs   =        753
                                                LR chi2(6)      =      58.00
                                                Prob > chi2     =     0.0000
Log likelihood = -485.87503                     Pseudo R2       =     0.0563
  (output omitted)
. estimates store nmodel
. lrtest fmodel nmodel
Likelihood-ratio test                           LR chi2(1)  =      66.48
(Assumption: nmodel nested in fmodel)           Prob > chi2 =     0.0000
```

The resulting LR test can be interpreted as indicating that the effect of having young children is significant at the .01 level ($LRX^2 = 66.48$, $df = 1$, $p < .01$).

4.3.2 Testing multiple coefficients

Often you may wish to test complex hypotheses that involve more than one coefficient. For example, we have two variables that reflect education in the family, `hc` and `wc`. The conclusion that education has (or does not have) a significant effect on labor force participation cannot be based on a pair of tests of single coefficients. But a joint hypothesis can be tested using either `test` or `lrtest`.

Testing multiple coefficients using test

To test that the effect of the wife attending college and of the husband attending college on labor force participation are both equal to 0, H_0: $\beta_{wc} = \beta_{hc} = 0$, we fit the full model and then

```
. test hc wc
 ( 1)  hc = 0
 ( 2)  wc = 0
           chi2(  2) =   17.66
         Prob > chi2 =    0.0001
```

We conclude that the hypothesis that the effects of the husband's and the wife's education are simultaneously equal to zero can be rejected at the .01 level ($X^2 = 17.66$, $df = 2$, $p < .01$).

This form of the `test` command can be readily extended to hypotheses regarding more than two independent variables by listing more variables; for example, `test wc hc k5`.

`test` can also be used to test the equality of coefficients. For example, to test that the effect of the wife attending college on labor force participation is equal to the effect of the husband attending college, H_0: $\beta_{wc} = \beta_{hc}$:

```
. test hc=wc
 ( 1) - wc + hc = 0
           chi2(  1) =    3.54
         Prob > chi2 =    0.0600
```

Here `test` has translated $\beta_{wc} = \beta_{hc}$ into the equivalent expression $-\beta_{wc} + \beta_{hc} = 0$. We conclude that the null hypothesis that the effects of husband's and wife's education are equal is marginally significant at the .05 level ($X^2 = 3.54$, $df = 1$, $p = .06$). This result suggests that we have weak evidence that the effects are not equal.

Testing multiple coefficients using lrtest

To compute an LR test of multiple coefficients, we first fit the full model and then save the results using the command `estimates store`. Then to test the hypothesis that the effect of the wife attending college and of the husband attending college on labor force participation are both equal to zero, H_0: $\beta_{\text{wc}} = \beta_{\text{hc}} = 0$, we fit the model that excludes these two variables and then run `lrtest`:

```
. logit lfp k5 k618 age wc hc lwg inc, nolog
  (output omitted)
. estimates store fmodel
. logit lfp k5 k618 age lwg inc, nolog
  (output omitted)
. estimates store nmodel
. lrtest fmodel nmodel
Likelihood-ratio test                                 LR chi2(2)  =     18.50
(Assumption: nmodel nested in fmodel)                 Prob > chi2 =    0.0001
```

We conclude that the hypothesis that the effects of the husband's and the wife's education are simultaneously equal to zero can be rejected at the .01 level ($LRX^2 = 18.50$, $df = 2$, $p < .01$).

This logic can be extended to exclude other variables. Say that we wish to test the null hypothesis that all the effects of the independent variables are simultaneously equal to zero. We do not need to fit the full model again because the results are still saved from our use of `estimates store fmodel` above. We fit the model with no independent variables and run `lrtest`:

```
. logit lfp, nolog
  (output omitted)
. estimates store intercept_only
. lrtest fmodel intercept_only
Likelihood-ratio test                                 LR chi2(7)  =     124.48
(Assumption: intercept_only  nested in fmodel)        Prob > chi2 =    0.0000
```

We can reject the hypothesis that all coefficients except the intercept are zero at the .01 level ($LRX^2 = 124.48$, $df = 7$, $p < .01$). This test is identical to the test in the header of the `logit` output:
`LR chi2(7) = 124.48`.

4.3.3 Comparing LR and Wald tests

Although the LR and Wald tests are *asymptotically* equivalent, their values differ in finite samples. For example,

		LR test		Wald test	
Hypothesis	*df*	G^2	p	W	p
$\beta_{\text{k5}} = 0$	1	66.48	<.01	55.14	<.01
$\beta_{\text{wc}} = \beta_{\text{hc}} = 0$	2	18.50	<.01	17.66	<.01
All slopes = 0	7	124.48	<.01	95.0	< .01

Statistical theory is unclear on whether the LR or Wald test is to be preferred in models for categorical outcomes, although many statisticians, ourselves included, prefer the LR test. The choice of which test to use is often determined by convenience, personal preference, and convention within an area of research.

4.4 Residuals and influence using predict

Examining residuals and outliers is an important way to assess the fit of a regression model. *Residuals* are the difference between a model's predicted and observed outcome for each observation in the sample. Cases that fit poorly (i.e., have large residuals) are known as *outliers*. When an observation has a large effect on the estimated parameters, it is said to be *influential*.

Not all outliers are influential, as figure 4.3 shows. In the top panel, we show a scatterplot of some simulated data, and we have drawn the line that results from the linear regression of y on x. The residual of any observation is its vertical distance from the regression line. The observation highlighted by the box has a very large residual and so is an outlier. Even so, it is not very influential on the slope of the regression line. In the bottom panel, the only observation whose value has changed is the highlighted one. Now the magnitude of the residual for this observation is much smaller, but it is very influential; its presence is entirely responsible for the slope of the new regression line being positive instead of negative.

(*Continued on next page*)

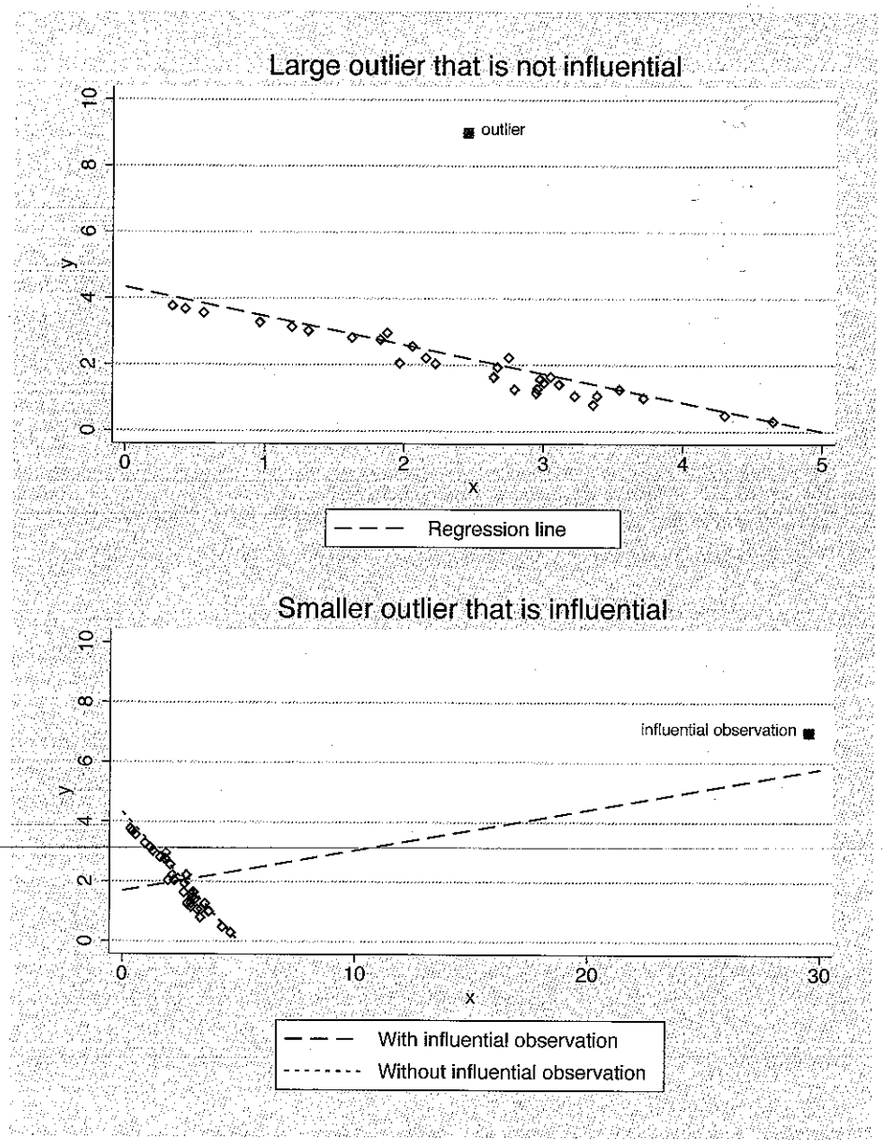


Figure 4.3: The distinction between an outlier and an influential observation.

Building on the analysis of residuals and influence in the linear regression model (see Fox 1991 and Weisberg 1980, chapter 5 for details), Pregibon (1981) extended these ideas to the BRM.

4.4.1 Residuals

If we define the predicted probability for a given set of independent variables as

$$\pi_i = \Pr(y_i = 1 \mid \mathbf{x}_i)$$

then the deviations $y_i - \pi_i$ are heteroskedastic, with

$$\text{Var}(y_i - \pi_i \mid \mathbf{x}_i) = \pi_i(1 - \pi_i)$$

This implies that the variance in a binary outcome is greatest when $\pi_i = .5$ and least as π_i approaches 0 or 1. For example, $.5(1 - .5) = .25$ and $.01(1 - .01) = .0099$. In other words, there is heteroskedasticity that depends on the probability of a positive outcome. This suggests the *Pearson residual*, which divides the residual $y - \widehat{\pi}$ by its standard deviation:

$$r_i = \frac{y_i - \widehat{\pi}_i}{\sqrt{\widehat{\pi}_i(1 - \widehat{\pi}_i)}}$$

Large values of r suggest a failure of the model to fit a given observation. Pregibon (1981) showed that the variance of r is not 1, as $\text{Var}(y_i - \widehat{\pi}_i) \neq \widehat{\pi}_i(1 - \widehat{\pi}_i)$, and proposed the *standardized Pearson residual*

$$r_i^{\text{Std}} = \frac{r_i}{\sqrt{1 - h_{ii}}}$$

where

$$h_{ii} = \widehat{\pi}_i(1 - \widehat{\pi}_i)\,\mathbf{x}_i\,\widehat{\text{Var}}\left(\widehat{\boldsymbol{\beta}}\right)\mathbf{x}_i' \tag{4.3}$$

Although r^{Std} is preferred over r because of its constant variance, we find that the two residuals are often similar in practice. But, because r^{Std} is simple to compute in Stata, we recommend that you use this measure.

Example

An *index plot* is a useful way to examine residuals by simply plotting them against the observation number. The standardized residuals can be computed by specifying the `rs` option with `predict`. For example,

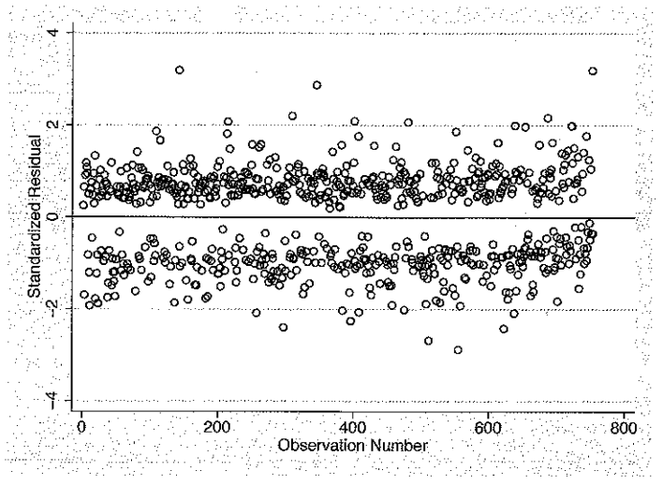
```
. logit lfp k5 k618 age wc hc lwg inc, nolog
(output omitted)
. predict rstd, rs
. label var rstd "Standardized Residual"
. sort inc, stable
. generate index = _n
. label var index "Observation Number"
```

Here we first fit the logit model. Second, we use the `rs` option for `predict` to specify that we want standardized residuals, which are placed in a new variable that we have named `rstd`. Third, we sort the cases by `income`, so that observations are ordered from

lowest to highest incomes. This results in a plot of residuals in which cases are ordered from low income to high income. The next line creates a new variable `index`, whose value for each observation is that observation's number (i.e., row) in the dataset. Note that `_n` on the right side of `generate` inserts the observation number. All that remains is to plot the residuals against the index using the commands[2]

```
. graph twoway scatter rstd index, xlabel(0(200)800) ylabel(-4(2)4) ///
> xtitle("Observation Number") yline(0) msymbol(Oh)
```

which produces the following index plot of standardized Pearson residuals:



There is no hard-and-fast rule for what counts as a "large" residual. Indeed, in their detailed discussion of residuals and outliers in the binary regression model, Hosmer and Lemeshow (2000, 176) sagely caution that it is impossible to provide any absolute standard: "In practice, an assessment of 'large' is, of necessity, a judgment call based on experience and the particular set of data being analyzed".

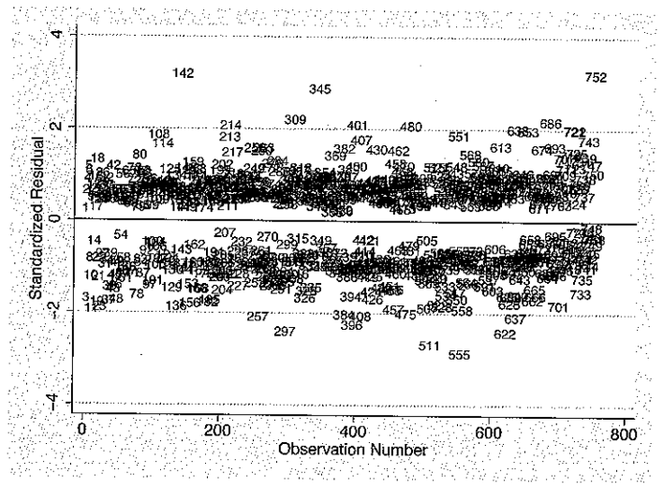
One way to search for problematic residuals is to sort the residuals by the value of a variable that you think may be a problem for the model. Here we sorted the data by `income` before plotting. If this variable had been primarily responsible for the lack of fit of some observations, the plot would show a disproportionate number of cases with large residuals among either the low-income or the high-income observations in our model. However, this does not appear to be the case for these data.

Still, in our plot, several residuals stand out as being large relative to the others. In such cases, it is important to identify the specific observations with large residuals for further inspection. We can do this by instructing `graph` to use the observation number to label each point in our plot. Recall that we just created a new variable called `index`

2. The /// is just a way of executing long lines in do-files. You should not type these characters if you are working from the Command window.

whose value is equal to the observation number for each observation. We want the values of this index variable to be the marker symbols. We do this by labeling the marker with the index value and then placing the label over an invisible marker. In the command below, `msymbol(none)` makes the marker symbol invisible, `mlabel(index)` specifies that the variable `index` contains the labels, and `mlabposition(0)` causes the label to be positioned where the marker would have appeared. For example,

```
. graph twoway scatter rstd index, xlabel(0(200)800) ylabel(-4(2)4) ///
> xtitle("Observation Number") yline(0)                              ///
> msymbol(none) mlabel(index) mlabposition(0)
```



Although labeling points with observations leads to chaos where there are many points, it effectively highlights and identifies the isolated cases. You can then easily list these cases. For example, observation 142 stands out and should be examined:

```
. list in 142, noobs
```

lfp	k5	k618	age	wc	hc	lwg	inc	rstd	index
inLF	1	2	36	NoCol	NoCol	-2.054124	11.2	3.191524	142

(Continued on next page)

We can also use `list` to list all observations with large residuals:

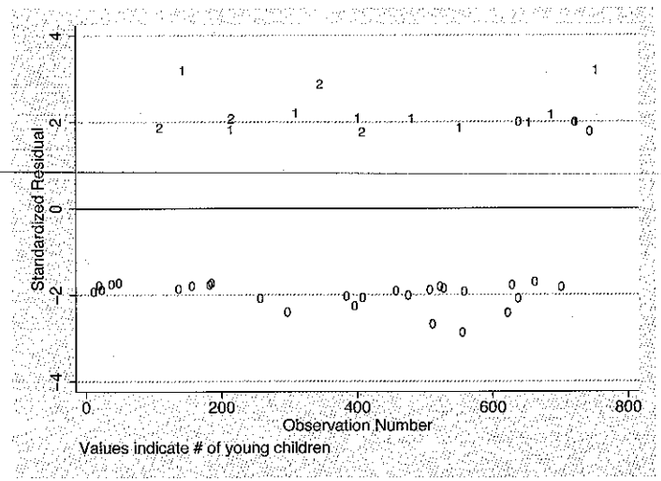
```
. list rstd index if rstd>2.5 | rstd<-2.5
```

	rstd	index
142.	3.191524	142
345.	2.873378	345
511.	-2.677243	511
555.	-2.871972	555
752.	3.192648	752

We can then check the listed cases to see if there are problems.

An initial review of the cases with large positive or negative residuals suggests that these could be related to the number of children a scientist has. To explore this further, we modify the last graph to plot only large residuals and to label these residuals with the number of young children a scientist has. To do this,

```
. graph twoway scatter rstd index if (rstd>1.7) | (rstd<-1.7), ///
> msymbol(none) mlabel(k5) mlabposition(0)                     ///
> caption("Values indicate # of young children")               ///
> xlabel(0(200)800) xtitle("Observation Number")               ///
> ylabel(-4(2)4) yline(0)
```



Here all the cases with negative residuals correspond to individuals without young children, whereas most cases with positive residuals have children. This suggests that we should further consider the way in which the number of children has been modeled.

Regardless of which method is used, further analyses of the highlighted cases might reveal either incorrectly coded data or some inadequacy in the specification of the model.

Cases with large positive or negative residuals should *not* simply be discarded from the analysis but rather should be examined to determine why they fit so poorly.

4.4.2 Influential cases

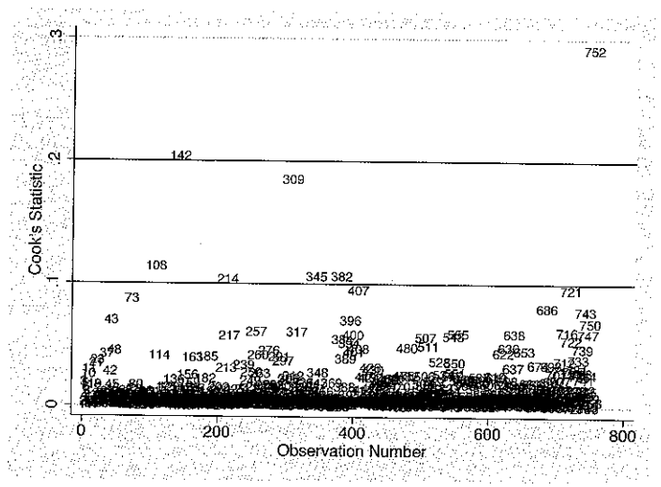
As shown in figure 4.3, large residuals do not necessarily have a strong influence on the estimated parameters, and observations with relatively small residuals can have a large influence. *Influential* points are also sometimes called *high-leverage* points. These can be determined by examining the change in the estimated $\widehat{\boldsymbol{\beta}}$ that occurs when the ith observation is deleted. Although estimating a new logit for each case is usually impractical (although as the speed of computers increases, this may soon no longer be so), Pregibon (1981) derived an approximation that requires fitting the model only once. This measure summarizes the effect of removing the ith observation on the entire vector $\widehat{\boldsymbol{\beta}}$, which is the counterpart to Cook's distance for the linear regression model. The measure is defined as

$$P_i = \frac{r_i^2 h_{ii}}{(1-h_{ii})^2}$$

where h_{ii} was defined in (4.3). In Stata, which refers to Cook's distance as `dbeta`, we can compute and plot Cook's distance as follows:

```
. predict cook, dbeta
. label var cook "Cook's Statistic"
. graph twoway scatter cook index, xlabel(0(200)800) ylabel(0(.1).3) ///
> xtitle("Observation Number") yline(.1 .2)                           ///
> msymbol(none) mlabel(index) mlabposition(0)
```

These commands produce the following plot, which shows that cases 142, 309, and 752 merit further examination:



Methods for plotting residuals and outliers can be extended in many ways, including plots of different diagnostics against one another. Details of these plots are found in Cook and Weisberg (1999), Hosmer and Lemeshow (2000), and Landwehr, Pregibon, and Shoemaker (1984).

4.4.3 Least likely observations

A common motivation for examining residuals in the linear regression model is to uncover the largest residuals and to check if there is some reason why the model fits these observations so poorly. Observations with large residuals are those for which the observed values of the dependent variable are most "surprising" given the regression coefficients and the values of the independent variables. Maximum likelihood estimates maximize the probability of observing the outcomes that were actually observed. In this context, we can think of the most surprising outcomes as those that have the smallest predicted probabilities of observing that outcome. These cases may warrant closer inspection precisely because observations with large residuals do in the more familiar linear regression model. Our command `leastlikely` (Freese 2002) will list the least likely observations. For example, for a binary model, `leastlikely` will list both the observations with the smallest $\widehat{\Pr}(y=0)$ among cases in which $y=0$ and the smallest $\widehat{\Pr}(y=1)$ for cases in which $y=1$. In addition to `logit` and `probit`, `leastlikely` can be used after most binary models in which the option `p` for `predict` generates the predicted probabilities of a positive outcome (e.g., `cloglog`, `scobit`, `hetprob`) and after many models for ordinal or nominal outcomes in which the option `outcome(#)` for `predict` generates the predicted probability of outcome # (e.g., `ologit`, `oprobit`, `mlogit`, `mprobit`, `slogit`). `leastlikely` is not appropriate for models in which the probabilities produced by `predict` are probabilities within groups or panels, such as `clogit`, `nlogit`, or `asmprobit`.

Syntax

The syntax for `leastlikely` is as follows:

`leastlikely` [*varlist*] [*if*] [*in*] [, `n(`#`)` `generate(`*varname*`)` [`no`]`display` `nolabel` `noobs`]

where *varlist* contains any variables whose values are to be listed in addition to the observation numbers and probabilities.

Options

`n(`#`)` specifies the number of observations to be listed for each level of the outcome variable. The default is `n(5)`. For multiple observations with identical predicted probabilities, all observations will be listed.

`generate(`*varname*`)` specifies that the probabilities of observing the outcome that was observed should be stored in *varname*.

Options controlling the list of values

`leastlikely` can also include any of the options available after `list`. These include the following:

[`no`]`display` forces the format into display or tabular (`nodisplay`) format. If you do not specify one of these two options, Stata chooses the one it decides will be most readable.

`nolabel` causes numeric values rather than labels to be displayed.

`noobs` suppresses printing of the observation numbers.

For example, we can use `leastlikely` to identify the least likely observations for our model of labor force participation and to list the values of the variables `k5`, `k618`, and `wc` for these observations.

```
. use http://www.stata-press.com/data/lf2/binlfp2
(Data from 1976 PSID-T Mroz)
. logit lfp k5 k618 age wc hc lwg inc
  (output omitted)
. leastlikely k5 k618 wc
Outcome: 0 (NotInLF)
```

	Prob	k5	k618	wc
60.	.1231792	0	1	College
172.	.1490344	0	2	College
221.	.1470691	0	2	College
235.	.1666356	0	4	College
252.	.1088271	0	0	College

```
Outcome: 1 (inLF)
```

	Prob	k5	k618	wc
338.	.1760865	1	2	College
534.	.0910262	1	2	NoCol
568.	.178205	1	5	NoCol
635.	.0916614	1	3	College
662.	.1092709	2	0	NoCol

Among women not in the labor force (`lfp` is 0), we find that the lowest predicted probability of not being in the labor force occurs for those who attended college and have young children. For women in the labor force (`lfp` is 1) with the lowest probabilities of being in the labor force, all individuals have young children and most have more than one older child. This suggests further consideration of how labor force participation is affected by having children in the family.

4.5 Measuring fit

As discussed in chapter 3, a scalar measure of fit can be useful in comparing competing models. Within a substantive area, measures of fit provide a *rough* index of whether a model is adequate. For example, if prior models of labor force participation routinely have values of .4 for a given measure of fit, you would expect that new analyses with a different sample or with revised measures of the variables would result in a similar value for that measure of fit. Remember: there is *no convincing evidence that selecting a model that maximizes the value of a given measure of fit results in a model that is optimal in any sense other than the model's having a larger value of that measure.* Details on these measures are presented in chapter 3.

4.5.1 Scalar measures of fit using fitstat

To illustrate the use of scalar measures of fit, consider two models. M_1 contains our original specification of independent variables `k5`, `k618`, `age`, `wc`, `hc`, `lwg`, and `inc`. M_2 drops the variables `k618`, `hc`, and `lwg` and adds `agesq`, which is the square of `age`. These models are fitted, and measures of fit are computed:

```
. quietly logit lfp k5 k618 age wc hc lwg inc, nolog
. estimates store model1
. quietly fitstat, save
. gen agesq = age*age
. quietly logit lfp k5 age agesq wc inc, nolog
. estimates store model2
```

We used `quietly` to suppress the output from `logit` and now use `estimates table` to combine the results from the two logits:

```
. estimates table model1 model2, b(%9.3f) t
```

Variable	model1	model2
k5	-1.463	-1.380
	-7.43	-7.06
k618	-0.065	
	-0.95	
age	-0.063	0.057
	-4.92	0.50
wc	0.807	1.094
	3.51	5.50
hc	0.112	
	0.54	
lwg	0.605	
	4.01	
inc	-0.034	-0.032
	-4.20	-4.18
agesq		-0.001
		-1.00
_cons	3.182	0.979
	4.94	0.40

legend: b/t

The output from `fitstat` for M_1 was suppressed, but the results were saved to be listed by a second call to `fitstat` using the `diff` option:

```
. fitstat, diff
Measures of Fit for logit of lfp
```

	Current	Saved	Difference
Model:	logit	logit	
N:	753	753	0
Log-Lik Intercept Only	-514.873	-514.873	0.000
Log-Lik Full Model	-461.653	-452.633	-9.020
D	923.306(747)	905.266(745)	18.040(2)
LR	106.441(5)	124.480(7)	18.040(2)
Prob > LR	0.000	0.000	0.000
McFadden's R2	0.103	0.121	-0.018
McFadden's Adj R2	0.092	0.105	-0.014
ML (Cox-Snell) R2	0.132	0.152	-0.021
Cragg-Uhler(Nagelkerke) R2	0.177	0.204	-0.028
McKelvey & Zavoina's R2	0.182	0.217	-0.035
Efron's R2	0.135	0.155	-0.020
Variance of y*	4.023	4.203	-0.180
Variance of error	3.290	3.290	0.000
Count R2	0.677	0.693	-0.016
Adj Count R2	0.252	0.289	-0.037
AIC	1.242	1.223	0.019
AIC*n	935.306	921.266	14.040
BIC	-4024.871	-4029.663	4.791
BIC'	-73.321	-78.112	4.791
BIC used by Stata	963.050	958.258	4.791
AIC used by Stata	935.306	921.266	14.040

```
Difference of    4.791 in BIC' provides positive support for saved model.
Note: p-value for difference in LR is only valid if models are nested.
```

These results illustrate the limitations inherent in scalar measures of fit. M_2 deleted two variables that were not significant and one that was from M_1. It added a new variable that was not significant in the new model. Because the models are not nested, they cannot be compared using a difference of chi-squares test.[3] What do the fit statistics show? First, the values of the pseudo-R^2s are slightly larger for M_1. If you take the pseudo-R^2s as evidence for the best model, which we do not, there is some evidence preferring M_1. Second, the BIC statistic is smaller for M_1, which provides support for that model. Following Raftery's (1996) guidelines, one would say that there is positive (neither weak nor strong) support for M_1.

4.5.2 Hosmer–Lemeshow statistic

Earlier we showed how to use `predict` to compute the predicted probabilities for each observation in the sample. The idea of the Hosmer–Lemeshow (HL) test statistic is to compare these predicted probabilities with the observed data (Lemeshow and Hosmer 1982; Hosmer and Lemeshow 1980). To explain what this statistic is doing, we go through the steps that are used to compute HL.

3. `fitstat, diff` computes the difference between all measures, even if the models are not nested. As with the Stata command `lrtest`, it is up to the user to determine if it makes sense to interpret the computed difference.

1. Fit the model.
2. Compute the predicted probabilities $\widehat{\pi}_i$.
3. Sort the data from the smallest value of $\widehat{\pi}_i$ to the largest.
4. Divide the observations into G groups, where 10 groups are often used. Each group will have $n_g \approx \frac{N}{G}$ cases (if G does not divide equally into N, the group sizes will differ slightly). The first group will have the n_1 smallest values of $\widehat{\pi}_i$, and so on.
5. Within each group, compute the mean prediction $\overline{\pi}_g = \sum_{\text{Group g}} \widehat{\pi}_i/n_g$ and the mean number of observed ones, $\overline{y}_g = \sum_{\text{Group g}} y_i/n_g$.
6. HL is a Pearson χ^2 statistic with $G-2$ degrees of freedom:

$$\text{HL} = \sum_{g=1}^{G} \frac{\left(n_g\overline{y}_g - n_g\overline{\pi}_g\right)^2}{n_g\overline{\pi}_g\left(1-\overline{\pi}_g\right)}$$

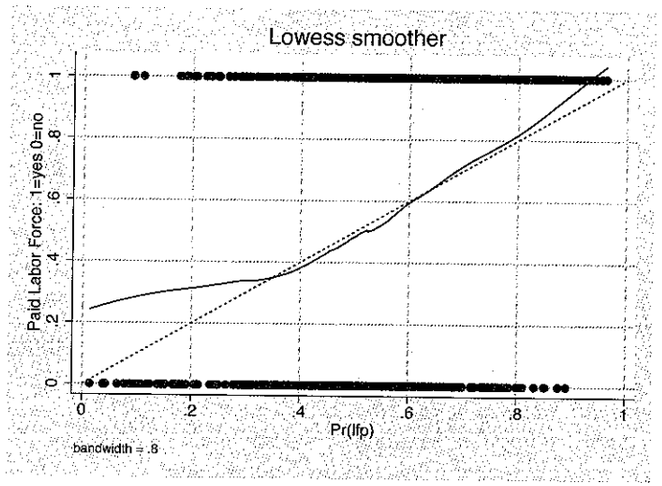
Hosmer and Lemeshow (2000) ran extensive simulations that showed that HL is approximately distributed as χ^2 if the model is correct. But, since the value of HL depends on the number of groups chosen, it is better to think of this statistic as a guide to assessing the fit of a model rather than a formal test. When using this statistic, keep in mind what Hosmer and Lemeshow (2000) wrote: "The advantage of a summary goodness-of-fit statistics like [HL] is that it provides a single, easily interpretable value that can be used to assess fit. The great disadvantage is that in the process of grouping we may miss an important deviation from fit due to the small number of individual data points. Hence, we advocate that, before finally accepting that a model fits, an analysis of the individual residuals and relevant diagnostic statistics be performed."

For our example of labor force participation, we computed the HL statistic using the command

```
. estat gof, group(10)
Logistic model for lfp, goodness-of-fit test
  (Table collapsed on quantiles of estimated probabilities)
      number of observations =       753
            number of groups =        10
    Hosmer-Lemeshow chi2(8) =     23.79
                Prob > chi2 =      0.0025
```

The HL statistic suggests that the model does not fit well. However, if you experiment with different values for `group()`, you will see that the p-value is sensitive to the number of groups used. Still, the results suggests that we should explore the fit of the model. One way to do this is to make a lowess graph comparing predicted probabilities to a moving average of the proportion of cases that are one. This can be done with the following commands:


```
. predict p1
(option p assumed; Pr(lfp))
. lowess lfp p1, ylabel(0(.2)1, grid) xlabel(0(.2)1, grid) ///
> addplot(function y = x, legend(off))
```



Roughly speaking, the solid line shows the fraction of observed cases that equal 1 at each level of the model's predicted probability of observing a 1. The closer the solid line to the diagonal, dashed line, the better the fit of the model. The graph suggests that the model fails in predicting the lower probabilities of being in the labor force, where the fractions of observed cases exceeds the predicted probabilities.

4.6 Interpretation using predicted values

Because the BRM is nonlinear, no approach to interpretation can fully describe the relationship between a variable and the outcome. We suggest that you try a variety of methods, with the goal of finding an elegant way to present the results that does justice to the complexities of the nonlinear model.

In general, the estimated parameters from the BRM do not provide directly useful information for understanding the relationship between the independent variables and the outcome. With the exception of the rarely used method of interpreting the latent variable (which we discuss in our treatment of ordinal models in chapter 5), substantively meaningful interpretations are based on predicted probabilities and functions of those probabilities (e.g., ratios, differences). As shown in figure 4.1, for a given set of values of the independent variables, the predicted probability in BRMs is defined as

$$\text{Logit: } \widehat{\Pr}(y=1 \mid \mathbf{x}) = \Lambda\left(\mathbf{x}\widehat{\boldsymbol{\beta}}\right) \qquad \text{Probit: } \widehat{\Pr}(y=1 \mid \mathbf{x}) = \Phi\left(\mathbf{x}\widehat{\boldsymbol{\beta}}\right)$$

where Λ is the cdf for the logistic distribution with variance $\pi^2/3$, and Φ is the cdf for the normal distribution with variance 1. For any set of values of the independent variables, the predicted probability can be computed. Several commands in Stata and our `pr*` commands make it simple to work with these predicted probabilities.

4.6.1 Predicted probabilities with predict

After running `logit` or `probit`,

`predict` *newvarname* [*if*] [*in*]

can be used to compute the predicted probability of a positive outcome for each observation, given the values on the independent variables for that observation. The predicted probabilities are stored in the new variable *newvarname*. The predictions are computed for all cases in memory that do not have missing values for the variables in the model, regardless of whether `if` and `in` had been used to restrict the estimation sample. For example, if you estimate `logit lfp k5 age if wc==1`, only 212 cases are used. But `predict` *newvarname* computes predictions for the entire dataset, 753 cases. If you want predictions only for the estimation sample, you can use the command `predict` *newvarname* `if e(sample)==1`.[4]

`predict` can be used to examine the range of predicted probabilities from your model. For example,

```
. predict prlogit
(option p assumed; Pr(lfp))
. summarize prlogit
```

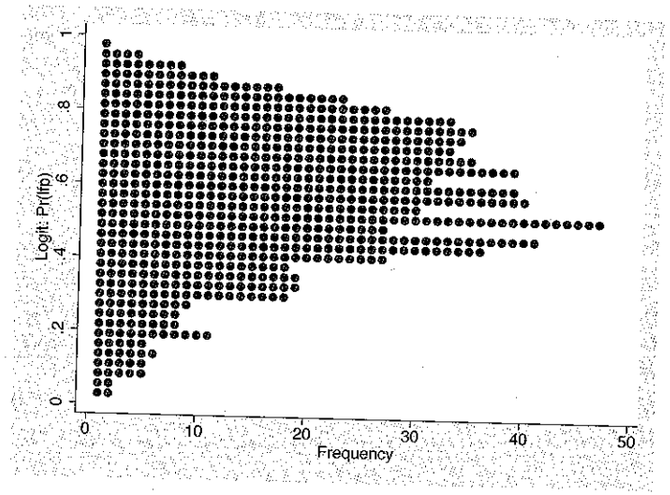
Variable	Obs	Mean	Std. Dev.	Min	Max
prlogit	753	.5683931	.1944213	.0139875	.9621198

The message `(option p assumed; Pr(lfp))` reflects that `predict` can compute many different quantities. Because we did not specify an option indicating which quantity to predict, option `p` for predicted probabilities was assumed, and the new variable `prlogit` was given the variable label `Pr(lfp)`. `summarize` computes summary statistics for the new variable and shows that the predicted probabilities in the sample range from .014 to .962, with a mean predicted probability of being in the labor force of .568.

4. Stata estimation commands create the variable `e(sample)`, indicating whether a case was used when fitting a model. Accordingly, the condition `if e(sample)==1` selects only cases used in the last estimation.

We can use `dotplot` to plot the predicted probabilities for our sample:

```
. label var prlogit "Logit: Pr(lfp)"
. dotplot prlogit, ylabel(0(.2)1)
```



The plot clearly shows that the predicted probabilities for individual observations span almost the entire range from 0 to 1 but that roughly two-thirds of the observations have predicted probabilities between .40 and .80.

`predict` can also be used to demonstrate that the predictions from logit and probit models are essentially identical. Even though the two models make different assumptions about $\text{Var}(\varepsilon)$, these differences are absorbed in the relative magnitudes of the estimated coefficients. To see this, we first fit the two models and compute their predicted probabilities:

```
. use http://www.stata-press.com/data/lf2/binlfp2, clear
(Data from 1976 PSID-T Mroz)
. logit lfp k5 k618 age wc hc lwg inc, nolog
  (output omitted)
. predict prlogit
(option p assumed; Pr(lfp))
. label var prlogit "Logit: Pr(lfp)"
. probit lfp k5 k618 age wc hc lwg inc, nolog
  (output omitted)
. predict prprobit
(option p assumed; Pr(lfp))
. label var prprobit "Probit: Pr(lfp)"
```

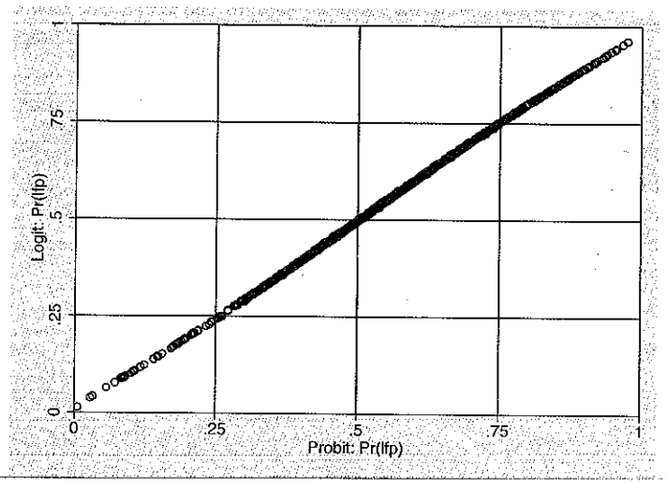
Next we check the correlation between the two sets of predicted values:

```
. pwcorr prlogit prprobit
```

	prlogit	prprobit
prlogit	1.0000	
prprobit	0.9998	1.0000

The extremely high correlation is confirmed by plotting them against one another:

```
. graph twoway scatter prlogit prprobit,  ///
> xlabel(0(.25)1) ylabel(0(.25)1)         ///
> xline(.25(.25)1) yline(.25(.25)1)       ///
> plotregion(margin(zero)) msymbol(Oh)
```



For predictions, there is little reason to prefer either logit or probit. If your substantive findings turn on whether you used logit or probit, *we would not place much confidence in either result*. In our own research, we tend to use logit, primarily because of the availability of interpretation in terms of odds and odds ratios (discussed below).

Overall, examining predicted probabilities for the cases in the sample provides an initial check of the model. To better understand and present the substantive findings, it is usually more effective to compute predictions at specific, substantively informative values. Our commands `prvalue`, `prtab`, and `prgen` are designed to make this simple.

4.6.2 Individual predicted probabilities with prvalue

A table of probabilities for ideal types of people (or countries, cows, or whatever you are studying) can quickly summarize the effects of key variables. In our example of labor force participation, we could compute predicted probabilities of labor force participation for women in these three types of families:

- young, low-income and low-education families with young children
- highly educated, middle-aged couples with no children at home
- an "average family" defined as having the mean on all variables.

This can be done with a series of calls to `prvalue` (see chapter 3 for a discussion of options for this command):[5]

```
. * young, low income, low education families with young children.
. prvalue, x(age=35 k5=2 wc=0 hc=0 inc=15) rest(mean)
logit: Predictions for lfp
Confidence intervals by delta method
                                95% Conf. Interval
  Pr(y=inLF|x):       0.1318   [ 0.0556,    0.2081]
  Pr(y=NotInLF|x):    0.8682   [ 0.7919,    0.9444]
          k5       k618        age         wc         hc        lwg
x=         2  1.3532537         35          0          0  1.0971148
         inc
x=        15
```

We have set the values of the independent variables to those that define our first type of family, with other variables held at their mean. The output shows the predicted probability of working, the confidence interval for that probability, and the specified values for the independent variables. At these values, we are 95% confident that the probability of being in the labor force is between .056 and .208. This process is repeated for the other ideal types.

```
. * highly educated families with no children at home.
. prvalue, x(age=50 k5=0 k618=0 wc=1 hc=1) rest(mean)
logit: Predictions for lfp
Confidence intervals by delta method
                                95% Conf. Interval
  Pr(y=inLF|x):       0.7166   [ 0.6333,    0.7999]
  Pr(y=NotInLF|x):    0.2834   [ 0.2001,    0.3667]
          k5       k618        age         wc         hc        lwg
x=         0          0         50          1          1  1.0971148
         inc
x=  20.128965
```

5. `mean` is the default setting for the `rest()` option, so `rest(mean)` does not need to be specified. We include it in many of our examples anyway, because its use emphasizes that the results are contingent on specified values for *all* of the independent variables.

```
. * an average person
. prvalue, rest(mean)
logit: Predictions for lfp
Confidence intervals by delta method

                              95% Conf. Interval
  Pr(y=inLF|x):       0.5778   [ 0.5392,    0.6164]
  Pr(y=NotInLF|x):    0.4222   [ 0.3836,    0.4608]

          k5       k618        age         wc         hc        lwg
x=   .2377158  1.3532537  42.537849   .2815405  .39176627  1.0971148

         inc
x=  20.128965
```

With predictions in hand, we can summarize the results and get a better general feel for the factors affecting a wife's labor force participation.

Ideal type	Probability of LFP (95% CI)
Young, low-income, and low-education families with young children	.13 (.06, .21)
Highly educated, middle-aged couples with no children at home	.72 (.63, .80)
An "average" family	.58 (.54, .62)

4.6.3 Tables of predicted probabilities with prtab

Sometimes the focus might be on two or three categorical independent variables. Predictions for all combinations of the categories of these variables could be presented in a table. For example,

No. of young children	Predicted probability: Did not attend college	Predicted probability: Attended college	Difference
0	.61	.78	.17
1	.26	.44	.18
2	.08	.16	.08
3	.02	.04	.02

This table shows the strong effect on labor force participation of having young children and how the effect differs according to the wife's education. One way to construct such a table is by a series of calls to `prvalue` (we use the `brief` option to limit output):


```
. prvalue, x(k5=0 wc=0) rest(mean) brief
logit: Predictions for lfp

                              95% Conf. Interval
  Pr(y=inLF|x):       0.6069   [ 0.5567,    0.6570]
  Pr(y=NotInLF|x):    0.3931   [ 0.3430,    0.4433]
. prvalue, x(k5=1 wc=0) rest(mean) brief
logit: Predictions for lfp

                              95% Conf. Interval
  Pr(y=inLF|x):       0.2633   [ 0.1932,    0.3335]
  Pr(y=NotInLF|x):    0.7367   [ 0.6665,    0.8068]
```

(*and so on*)

Even for a simple table, this approach is tedious and error prone. `prtab` automates the process by computing a table of predicted probabilities for all combinations of up to four categorical variables. For example,

```
. prtab k5 wc, rest(mean)
logit: Predicted probabilities of positive outcome for lfp

          |  Wife College:
          |    1=yes 0=no
# kids <  |
6         |   NoCol  College
----------+-------------------
        0 |  0.6069   0.7758
        1 |  0.2633   0.4449
        2 |  0.0764   0.1565
        3 |  0.0188   0.0412

          k5       k618        age         wc         hc        lwg
x=   .2377158  1.3532537  42.537849   .2815405  .39176627  1.0971148

         inc
x=  20.128965
```

The only disadvantage of using `prtab` is that it does not provide confidence intervals for the predictions.

4.6.4 Graphing predicted probabilities with prgen

When a variable of interest is continuous, you can either select values (e.g., quartiles) and construct a table or create a graph. For example, to examine the effects of `income` on labor force participation by `age`, we can use the estimated parameters to compute predicted probabilities as `income` changes for fixed values of `age`. This is shown in figure 4.4. The command `prgen` creates data that can be graphed in this way. The first step is to generate the predicted probabilities for those aged 30:

```
. prgen inc, from(0) to(100) generate(p30) x(age=30) rest(mean) n(11)
logit: Predicted values as inc varies from 0 to 100.
          k5       k618        age         wc         hc        lwg
x=   .2377158  1.3532537         30   .2815405  .39176627  1.0971148
         inc
x=  20.128965
. label var p30p1 "Age 30"
```

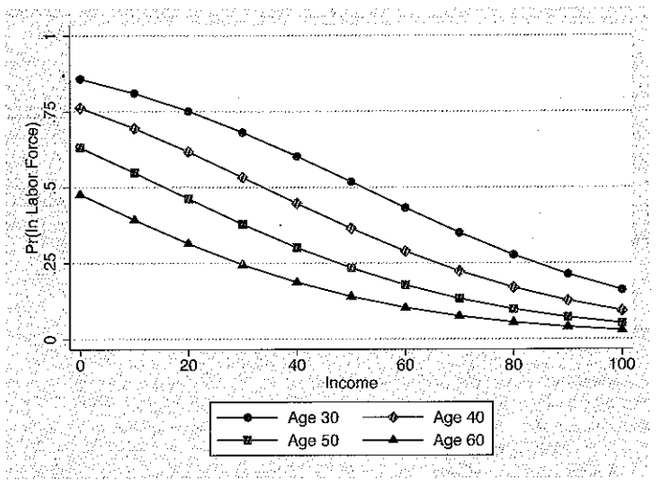


Figure 4.4: Graph of predicted probabilities created using prgen.

`inc` is the independent variable that we want to vary along the x-axis. The options that we use are

`from(0)` and `to(100)` specify the minimum and maximum values over which `inc` is to vary. The default is the variable's observed minimum and maximum values.

`generate(p30)` indicates the root name used in constructing new variables. `prgen` creates `p30x` that contains the values of `inc` that are used; `p30p1` with the values of the probability of a 1 and `p30p0` with values of the probability of a 0.

`x(age=30)` indicates that we want to hold the value of `age` at 30. By default, other variables will be held at their mean unless `rest()` is used to specify some other summary statistic.

`n(11)` indicates that 11 evenly spaced values of `inc` between 0 and 100 should be used. You should choose the value that corresponds to the number of symbols you want on your graph.

More calls of `prgen` are made holding `age` at different values:

```
. prgen inc, from(0) to(100) generate(p40) x(age=40) rest(mean) n(11)
(output omitted)
. label var p40p1 "Age 40"
. prgen inc, from(0) to(100) generate(p50) x(age=50) rest(mean) n(11)
(output omitted)
. label var p50p1 "Age 50"
. prgen inc, from(0) to(100) generate(p60) x(age=60) rest(mean) n(11)
(output omitted)
. label var p60p1 "Age 60"
```

Listing the values for the first 11 observations in the dataset for some of the new variables `prgen` has created may help you understand better what this command does:

```
. list p30p1 p40p1 p50p1 p60p1 p60x in 1/11
```

	p30p1	p40p1	p50p1	p60p1	p60x
1.	.8575829	.7625393	.6313345	.4773258	0
2.	.8101358	.6947005	.5482202	.3928797	10
3.	.7514627	.6172101	.462326	.3143872	20
4.	.6817801	.5332655	.3786113	.2452419	30
5.	.6028849	.4473941	.3015535	.187153	40
6.	.5182508	.36455	.2342664	.1402662	50
7.	.4325564	.289023	.1781635	.1036283	60
8.	.3507161	.2236366	.1331599	.0757174	70
9.	.2768067	.1695158	.0981662	.0548639	80
10.	.2133547	.1263607	.071609	.0395082	90
11.	.1612055	.0929622	.0518235	.0283215	100

The predicted probabilities of labor force participation for those averages on all other variables at ages 30, 40, 50, and 60 are in the first four columns. The clear negative effect of age is shown by the increasingly small probabilities as we move across these columns in any row. The last column indicates the value of income for a given row, starting at 0 and ending at 100. We can see that the probabilities decrease as income increases.

The following `graph` command generates the plot:

```
. graph twoway connected p30p1 p40p1 p50p1 p60p1 p60x,          ///
> ytitle("Pr(In Labor Force)") ylabel(0(.25)1) xtitle("Income")
```

Because we have not used `graph` much yet, it is worth discussing some points that we find useful (also see section 2.16).

1. Recall that `///` is a way of entering long lines in do-files.
2. `graph twoway` is the command for plotting a dependent variable on the y-axis against an independent variable along the x-axis. `graph twoway connected` specifies that the symbols used to mark the individual points be connected.

3. The variables to plot are p30p1 p40p1 p50p1 p60p1 p60x, where p60x, the last variable in the list, is the variable for the horizontal axis. All variables before the last variable are plotted on the vertical axis.
4. The options ytitle() and xtitle() specify the axis titles.
5. The ylabel() specifies which points on the y-axis to label.

4.6.5 Plotting confidence intervals

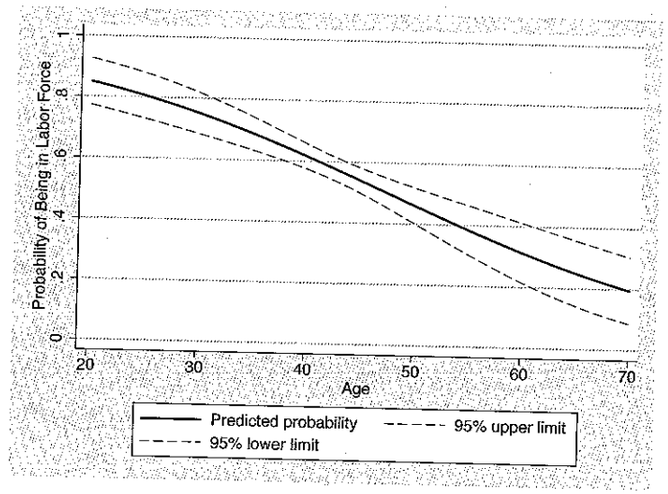
We can also use prgen to plot confidence intervals around our predictions by adding the ci option. Although you can use any of the options that control how confidence intervals are constructed (see page 123 for details), here we use the default options to keep things simple. We want to plot the probability of being in the labor force by age, adding the 95% confidence interval around the plot. The prgen command is

```
. prgen age, from(20) to(70) generate(prlfp) rest(mean) gap(2) ci
logit: Predicted values as age varies from 20 to 70.
            k5       k618        age         wc         hc        lwg
x=   .2377158  1.3532537  42.537849   .2815405  .39176627  1.0971148
           inc
x=  20.128965
. label var prlfpp1 "Predicted probability"
. label var prlfpp1ub "95% upper limit"
. label var prlfpp1lb "95% lower limit"
. label var prlfpx "Age"
```

We made several changes from the last time we used prgen. First, we added the ci option so that prgen would create variables with the estimates of the upper and lower bounds for the confidence interval of the predictions. These new variables are prlfpp1ub and prlfpp1lb. Second, rather than using the n() option to indicate the number of evenly spaced values of age to compute, we used the gap() option. With gap(), you simply indicate the spacing or gap between values. Here we want to compute values of age that increase by 2. Third, we have added variable labels for the variables created by prgen. These labels will be clearer in the graph than the default labels created by prgen. To plot the results, we use the following command:

```
. graph twoway                                              ///
>     (connected prlfpp1 prlfpx,                            ///
>         clcolor(black) clpat(solid) clwidth(medthick)     ///
>         msymbol(i) mcolor(none))                          ///
>     (connected prlfpp1ub prlfpx,                          ///
>         msymbol(i) mcolor(none)                           ///
>         clcolor(black) clpat(dash) clwidth(thin))         ///
>     (connected prlfpp1lb prlfpx,                          ///
>         msymbol(i) mcolor(none)                           ///
>         clcolor(black) clpat(dash) clwidth(thin)),        ///
>     ytitle("Probability of Being in Labor Force")         ///
>     yscale(range(0 .35))                                  ///
>     ylabel(, grid glwidth(medium) glpattern(solid))       ///
>     xscale(range(20 70)) xlabel(20(10)70)
```


This example of graph twoway shows how you can combine multiple plots into one graph. Each of the sections that begin with "(connected" and end with a ")" control the plotting of a different line. The resulting graph looks like this:

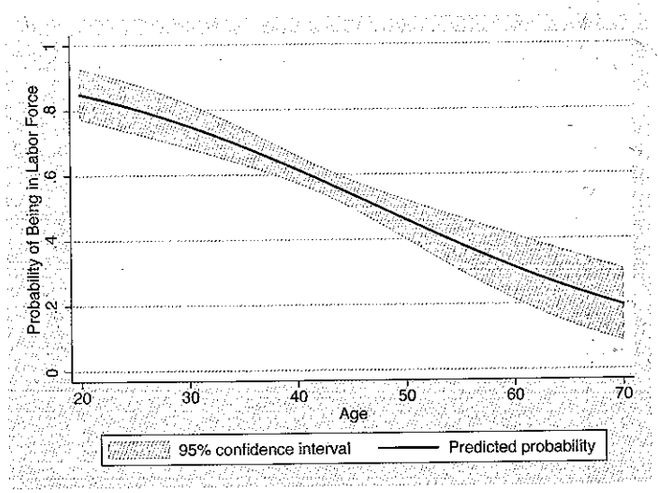


The graph shows that the confidence interval is smaller near the center of the data where age is 40 and increases as we move to younger or older ages.

Another way of showing the confidence intervals is to use shading. For this, we use the rarea type of plot. Specifying rarea(*yvar1 yvar2 xvar*, color(*color*)) will shade the area on the y-axis between the values of *yvar1* and *yvar2*, with the x-axis specified with *xvar*. We define the rarea graph before the connected graph since Stata draws overlaid graphs in the order specified, and we want the line indicating the predicted probabilities to appear on top of the shading. The improved command is

```
. graph twoway                                              ///
>     (rarea prlfpp1lb prlfpp1ub prlfpx, color(gs14))       ///
>     (connected prlfpp1 prlfpx,                            ///
>         clcolor(black) clpat(solid) clwidth(medthick)     ///
>         msymbol(i) mcolor(none)),                         ///
>     ytitle("Probability of Being in Labor Force")         ///
>     yscale(range(0 .35))                                  ///
>     ylabel(, grid glwidth(medium) glpattern(solid))       ///
>     xscale(range(20 70)) xlabel(20(10)70)                 ///
>     legend(label (1 "95% confidence interval"))
```

(*Continued on next page*)



4.6.6 Changes in predicted probabilities

Although graphs are useful for showing how predicted probabilities are related to an independent variable, for even our simple example it is not practical to plot all possible combinations of the independent variables. And sometimes the plots show that a relationship is linear, making a graph is superfluous. In such circumstances, a useful summary measure is the change in the outcome as one variable changes, holding all other variables constant.

Marginal change

In economics, the *marginal effect* or *change* is commonly used:

$$\text{marginal change} = \frac{\partial \Pr(y = 1 \mid \mathbf{x})}{\partial x_k}$$

The marginal change is shown by the tangent to the probability curve in figure 4.5. The value of the marginal effect depends on the level of all variables in the model. It is often computed with all variables held at their mean or by computing the marginal change for each observation in the sample and then averaging across all values.

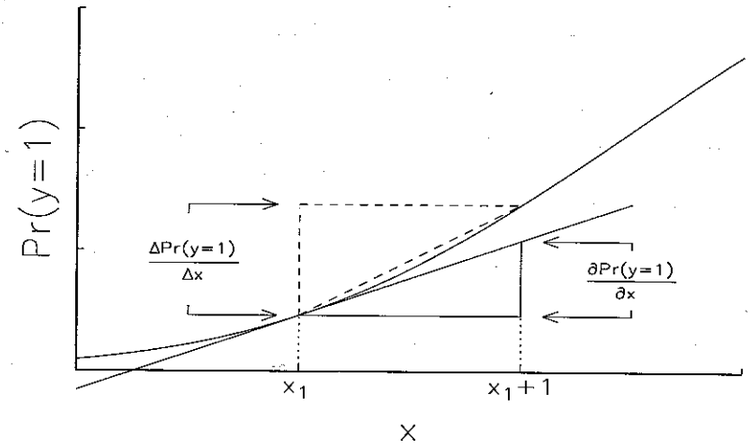


Figure 4.5: Marginal change compared with discrete change in the BRM.

Marginal change with prchange command

The command prchange computes the marginal at the values of the independent variables specified with x() or rest(). Running prchange with no options computes the marginal change (along with many other things discussed below) with all variables at their mean. Or, we can compute the marginal at specific values of the independent variables, such as when wc = 1 and age = 40. Here we request only the results for age:

```
. prchange age, x(wc=1 age=40) help
logit: Changes in Probabilities for lfp
       min->max      0->1      -+1/2     -+sd/2   MargEfct
age     -0.3940   -0.0017    -0.0121   -0.0971    -0.0121

         NotInLF      inLF
Pr(y|x)   0.2586    0.7414

              k5       k618        age         wc         hc        lwg        inc
   x=    .237716   1.35325         40          1   .391766    1.09711     20.129
sd(x)=   .523959   1.31987    8.07257   .450049   .488469    .587556    11.6348
 Pr(y|x): probability of observing each y for specified x values
Avg|Chg|: average of absolute value of the change across categories
Min->Max: change in predicted probability as x changes from its minimum to
          its maximum
    0->1: change in predicted probability as x changes from 0 to 1
   -+1/2: change in predicted probability as x changes from 1/2 unit below
          base value to 1/2 unit above
  -+sd/2: change in predicted probability as x changes from 1/2 standard
          dev below base to 1/2 standard dev above
MargEfct: the partial derivative of the predicted probability/rate with
          respect to a given independent variable
```

In plots that we do not show (but that we encourage you to create using `prgen` and `graph`), we found that the relationship between age and the probability of being in the labor force was essentially linear for those who attend college. Accordingly, we can take the marginal computed by `prchange`, multiply it by 10 to get the amount of change over 10 years, and report that for women who attend college, a 10-year increase in age decreases the probability of labor force participation by approximately .12, holding other variables at their mean.

When using the marginal, remember two points. First, the amount of change depends on the level of all variables. Second, as shown in figure 4.5, the marginal is the instantaneous rate of change. In general, it does not equal the actual change for a given finite change in the independent variable unless you are in a region of the probability curve that is approximately linear. Such linearity justifies the interpretation given above.

Marginal change with mfx command

The marginal change can also be computed using `mfx`, where the `at()` option is used to set values of the independent variables. Below we use `mfx` to estimate the marginal change for the same values that we used when calculating the marginal effect for `age` with `prchange` above:

```
. mfx, at(wc=1 age=40)
warning: no value assigned in at() for variables k5 k618 hc lwg inc;
   means used for k5 k618 hc lwg inc
Marginal effects after logit
      y  = Pr(lfp) (predict)
         =  .74140317
```

variable	dy/dx	Std. Err.	z	P>|z|	[95% C.I.]		X
k5	-.2804763	.04221	-6.64	0.000	-.363212	-.197741	.237716
k618	-.0123798	.01305	-0.95	0.343	-.037959	.013199	1.35325
age	-.0120538	.00245	-4.92	0.000	-.016855	-.007252	40
wc*	.1802113	.04742	3.80	0.000	.087269	.273154	1
hc*	.0212952	.03988	0.53	0.593	-.056866	.099456	.391766
lwg	.1159345	.03229	3.59	0.000	.052643	.179226	1.09711
inc	-.0066042	.00163	-4.05	0.000	-.009802	-.003406	20.129

```
(*) dy/dx is for discrete change of dummy variable from 0 to 1
```

`mfx` is particularly useful if you need estimates of the standard errors of the marginal effects; however, `mfx` computes the estimates using numerical methods, and for some models the command can take a long time.

Discrete change

Given the nonlinearity of the model, we prefer the *discrete change* in the predicted probabilities for a given change in an independent variable. To define discrete change, we need two quantities:

$\Pr(y = 1 \mid \mathbf{x}, x_k)$ is the probability of an event given $\mathbf{x}$, noting in particular the value of x_k.

$\Pr(y = 1 \mid \mathbf{x}, x_k + \delta)$ is the probability of the event with only x_k increased by some quantity δ.

Then the *discrete change* for a change of δ in x_k equals

$$\frac{\Delta \Pr(y = 1 \mid \mathbf{x})}{\Delta x_k} = \Pr(y = 1 \mid \mathbf{x}, x_k + \delta) - \Pr(y = 1 \mid \mathbf{x}, x_k)$$

which can be interpreted that for a change in variable x_k from x_k to $x_k + \delta$, the predicted probability of an event changes by $\{\Delta \Pr(y = 1 \mid \mathbf{x})\} / \Delta x_k$, holding all other variables constant.

As shown in figure 4.5, in general, the two measures of change are not equal. That is,

$$\frac{\partial \Pr(y = 1 \mid \mathbf{x})}{\partial x_k} \neq \frac{\Delta \Pr(y = 1 \mid \mathbf{x})}{\Delta x_k}$$

The measures differ because the marginal change is the instantaneous rate of change, whereas the discrete change is the amount of change in the probability for a given finite change in one independent variable. The two measures are similar, however, when the change occurs over a region of the probability curve that is roughly linear.

The value of the discrete change depends on

1. The start level of the variable that is being changed. For example, do you want to examine the effect of age beginning at 30? At 40? At 50?
2. The amount of change in that variable. Are you interested in the effect of a change of 1 year in age? Of 5 years? Of 10 years?
3. The level of all other variables in the model. Do you want to hold all variables at their mean? Or, do you want to examine the effect for women? Or, do you want to compute changes separately for men and women?

Accordingly, a decision must be made regarding each of these factors. See chapter 3 for more discussion.

For our example, let's look at the discrete change with all variables held at their mean, which is computed by default by `prchange`, where the `help` option is used to get detailed descriptions of what the measures mean:

```
. prchange, help
logit: Changes in Probabilities for lfp

      min->max      0->1      -+1/2     -+sd/2   MargEfct
k5    -0.6361   -0.3499   -0.3428   -0.1849   -0.3569
k618  -0.1278   -0.0156   -0.0158   -0.0208   -0.0158
age   -0.4372   -0.0030   -0.0153   -0.1232   -0.0153
wc     0.1881    0.1881    0.1945    0.0884    0.1969
hc     0.0272    0.0272    0.0273    0.0133    0.0273
lwg    0.6624    0.1499    0.1465    0.0865    0.1475
inc   -0.6415   -0.0068   -0.0084   -0.0975   -0.0084

         NotInLF      inLF
Pr(y|x)   0.4222    0.5778

            k5       k618      age       wc        hc       lwg       inc
   x=   .237716   1.35325   42.5378   .281541   .391766   1.09711   20.129
sd(x)=  .523959   1.31987   8.07257   .450049   .488469   .587556   11.6348

 Pr(y|x): probability of observing each y for specified x values
 Avg|Chg|: average of absolute value of the change across categories
 Min->Max: change in predicted probability as x changes from its minimum to
           its maximum
     0->1: change in predicted probability as x changes from 0 to 1
    -+1/2: change in predicted probability as x changes from 1/2 unit below
           base value to 1/2 unit above
   -+sd/2: change in predicted probability as x changes from 1/2 standard
           dev below base to 1/2 standard dev above
 MargEfct: the partial derivative of the predicted probability/rate with
           respect to a given independent variable
```

First, consider the results of changes from the minimum to the maximum. There is little to be learned by analyzing variables whose range of probabilities is small, such as hc, whereas age, k5, wc, lwg, and inc have *potentially* important effects. For these we can examine the value of the probabilities before and after the change by using the fromto option:

```
. prchange k5 age wc lwg inc, fromto
logit: Changes in Probabilities for lfp

      from:     to:     dif:     from:     to:     dif:     from:
     x=min     x=max  min->max     x=0     x=1     0->1     x-1/2
k5   0.6596    0.0235   -0.6361   0.6596   0.3097   -0.3499   0.7398
age  0.7506    0.3134   -0.4372   0.9520   0.9491   -0.0030   0.5854
wc   0.5216    0.7097    0.1881   0.5216   0.7097    0.1881   0.4775
lwg  0.1691    0.8316    0.6624   0.4135   0.5634    0.1499   0.5028
inc  0.7326    0.0911   -0.6415   0.7325   0.7256   -0.0068   0.5820

        to:      dif:     from:      to:      dif:
     x+1/2     -+1/2   x-1/2sd   x+1/2sd   -+sd/2   MargEfct
k5   0.3971   -0.3428    0.6675    0.4826   -0.1849   -0.3569
age  0.5701   -0.0153    0.6382    0.5150   -0.1232   -0.0153
wc   0.6720    0.1945    0.5330    0.6214    0.0884    0.1969
lwg  0.6493    0.1465    0.5340    0.6204    0.0865    0.1475
inc  0.5736   -0.0084    0.6258    0.5283   -0.0975   -0.0084

         NotInLF      inLF
Pr(y|x)   0.4222    0.5778

            k5       k618      age       wc        hc       lwg       inc
   x=   .237716   1.35325   42.5378   .281541   .391766   1.09711   20.129
sd(x)=  .523959   1.31987   8.07257   .450049   .488469   .587556   11.6348
```


We learn, for example, that varying age from its minimum of 30 to its maximum of 60 decreases the predicted probability by .44, from .75 to .31. Changing family income (inc) from its minimum to its maximum decreases the probability of a woman being in the labor force from .73 to .09. Interpreting other measures of change, the following interpretations can be made:

> *Using the unit change labeled* -+1/2: for a woman who is average on all characteristics, an additional young child decreases the probability of employment by .34.
>
> *Using the standard deviation change labeled* -+sd/2: a standard deviation change in age centered on the mean will decrease the probability of working by .12, holding other variables to their means.
>
> *Using a change from 0 to 1 labeled* 0->1: if a woman attends college, her probability of being in the labor force is .19 greater than a woman who does not attend college, holding other variables at their mean.

What if you need to calculate discrete change for changes in the independent values that are not the default for prchange (e.g., a change of 10 years in age rather than 1 year)? This can be done in two ways:

Confidence intervals for discrete change using the prvalue command

prchange does not provide confidence intervals. Although this feature might be added in the future, for now you need to compute these intervals using a series of calls to prvalue. Let's start with a simple example and then show how the process can be automated. We can use prchange to compute the discrete change when wc changes from 0 to 1, holding other variables at their mean:

```
. prchange wc, brief
    min->max      0->1      -+1/2     -+sd/2   MargEfct
wc    0.1881    0.1881    0.1945    0.0884    0.1969
```

Using prvalue, we quietly compute the predictions when wc is 0, save the results, and then compute predictions when wc is 1 using the diff option to compute discrete changes:

(*Continued on next page*)

```
. qui prvalue, x(wc=0) rest(mean) save
. prvalue, x(wc=1) diff
logit: Change in Predictions for lfp
Confidence intervals by delta method

                    Current      Saved     Change   95% CI for Change
  Pr(y=inLF|x):      0.7097     0.5216     0.1881  [ 0.0900,   0.2861]
  Pr(y=NotInLF|x):   0.2903     0.4784    -0.1881  [-0.2861,  -0.0900]

                k5        k618         age          wc          hc         lwg
Current=  .2377158   1.3532537   42.537849           1   .39176627   1.0971148
 Saved=   .2377158   1.3532537   42.537849           0   .39176627   1.0971148
  Diff=          0           0           0           1           0           0

                inc
Current=  20.128965
 Saved=   20.128965
  Diff=          0
```

The formal interpretation of these results requires that we imagine drawing repeated samples from the population and repeating all calculations for estimating the bounds of the confidence intervals for each sample (see page 88). About 95% of the computed confidence intervals would contain the true change in the predicted probability. In this sense, we are 95% confident that the true increase in the probability of a woman being in the labor force associated with her having been to college is between .09 and .29. More informally, we might say that we are 95% confident that the increase in the predicted probability of a woman being in the labor force associated with her having been to college is between .09 and .29. This confidence interval is the same as that computed by `mfx`, which uses numerical methods:

```
. mfx
Marginal effects after logit
      y  = Pr(lfp) (predict)
         =  .57779421
```

variable	dy/dx	Std. Err.	z	P>|z|	[95% C.I.]		X
k5	-.3568748	.04821	-7.40	0.000	-.451366	-.262383	.237716
k618	-.0157519	.01659	-0.95	0.342	-.048266	.016763	1.35325
age	-.0153371	.00311	-4.93	0.000	-.021434	-.00924	42.5378
wc*	.1880592	.05003	3.76	0.000	.09001	.286109	.281541
hc*	.0271985	.05004	0.54	0.587	-.070882	.125279	.391766
lwg	.1475137	.03674	4.01	0.000	.075496	.219532	1.09711
inc	-.0084031	.002	-4.19	0.000	-.012332	-.004474	20.129

(*) dy/dx is for discrete change of dummy variable from 0 to 1

Although `mfx` can compute confidence intervals for discrete changes for binary variables, it will not compute them for continuous variables and is slower than `prvalue` since `mfx` uses numerical methods to compute the confidence interval.

We can use a `foreach` loop (see [P] **foreach**) to automate the process of computing a series of discrete changes with confidence intervals. In the following example, we are computing a change from 0 to 1 for each of the variables `k5`, `k618`, and `wc`:


```
foreach v in k5 k618 wc {
      di _n "** Change from 0 to 1 in `v'"
      qui prvalue, x(`v'=0) rest(mean) save
          prvalue, x(`v'=1) diff brief
}
```

Line 1 causes the block of code between the braces `{` and `}` to be repeated three times. During the first pass, the local macro `v` is equal to `k5`, during the second pass it is equal to `k618`, and during the third pass it equals `wc`. (The ` and ' around `v` tells Stata to insert the value of the local macro v.) For each pass, line 2 labels the output; `` `v' `` inserts the name of the variable assigned in line 1, and line 3 `quietly` runs `prvalue`, assigning the variable represented by `` `v' `` to equal 0 and saving the result. Similarly, for each pass line 4 assigns the variable `` `v' `` to equal 1 and computes the difference between the new and saved results. Here is the output:

```
** Change from 0 to 1 in k5
logit: Change in Predictions for lfp

                    Current      Saved     Change   95% CI for Change
  Pr(y=inLF|x):      0.3097     0.6596    -0.3499  [-0.4334,  -0.2663]
  Pr(y=NotInLF|x):   0.6903     0.3404     0.3499  [ 0.2663,   0.4334]
** Change from 0 to 1 in k618
logit: Change in Predictions for lfp

                    Current      Saved     Change   95% CI for Change
  Pr(y=inLF|x):      0.5833     0.5990    -0.0156  [-0.0475,   0.0163]
  Pr(y=NotInLF|x):   0.4167     0.4010     0.0156  [-0.0163,   0.0475]
** Change from 0 to 1 in wc
logit: Change in Predictions for lfp

                    Current      Saved     Change   95% CI for Change
  Pr(y=inLF|x):      0.7097     0.5216     0.1881  [ 0.0900,   0.2861]
  Pr(y=NotInLF|x):   0.2903     0.4784    -0.1881  [-0.2861,  -0.0900]
```

The process is a bit more complicated when we want to compute the discrete, one-standard-deviation change centered around the mean. Again we use a `foreach` loop to repeat a block of code three times, once for `age`, again for `lwg`, and finally for `inc`. In the program below, line 2 runs the command `summarize` for the variable indicated by `` `v' ``. In line 3, we create a start value equal to the mean minus one-half of a standard deviation. Line 4 computes the end value equal to the mean plus one-half of a standard deviation. The starting and ending values are then passed to `prvalue` in lines 6 and 7.

```
. foreach v in age lwg inc {
 2.     qui summarize `v'
 3.     local start = r(mean) - (.5*r(sd))
 4.     local end = r(mean) + (.5*r(sd))
 5.     di _n "** Change from `start' to `end' in `v'"
 6.     qui prvalue, x(`v'=`start') rest(mean) save
 7.         prvalue, x(`v'=`end') dif brief
 8. }
```

```
** Change from 38.50156159842585 to 46.57413561272952 in age
logit: Change in Predictions for lfp
                    Current      Saved     Change   95% CI for Change
  Pr(y=inLF|x):      0.5150     0.6382    -0.1232  [-0.1717,  -0.0747]
  Pr(y=NotInLF|x):   0.4850     0.3618     0.1232  [ 0.0747,   0.1717]
** Change from .8033366225286708 to 1.390893047643295 in lwg
logit: Change in Predictions for lfp
                    Current      Saved     Change   95% CI for Change
  Pr(y=inLF|x):      0.6204     0.5340     0.0865  [ 0.0445,   0.1285]
  Pr(y=NotInLF|x):   0.3796     0.4660    -0.0865  [-0.1285,  -0.0445]
** Change from 14.31156614524011 to 25.94636467863254 in inc
logit: Change in Predictions for lfp
                    Current      Saved     Change   95% CI for Change
  Pr(y=inLF|x):      0.5283     0.6258    -0.0975  [-0.1428,  -0.0522]
  Pr(y=NotInLF|x):   0.4717     0.3742     0.0975  [ 0.0522,   0.1428]
```

Tip: Using estat summarize to get summary statistics for the estimation sample. Above we used `summarize` to compute values of the mean and standard deviation. If you want to view summary statistics restricted to the sample used to fit a model (that is, reflecting any `if` and `in` conditions you specified, as well as listwise deletion for observations with missing data), you can use `summarize` *varlist* `if e(sample)==1`. Or, more simply, `estat summarize` produces summary statistics restricted to the estimation sample for all variables in the model.

Nonstandard discrete changes with prvalue command

The command `prvalue` can be used to calculate the change in the probability for a discrete change of any magnitude in an independent variable. Say that we want to calculate the effect of a 10-year increase in age for a 30-year-old woman who is average on all other characteristics:

```
. prvalue, x(age=30) save brief
logit: Predictions for lfp
                              95% Conf. Interval
  Pr(y=inLF|x):      0.7506  [ 0.6830,   0.8183]
  Pr(y=NotInLF|x):   0.2494  [ 0.1817,   0.3170]
. prvalue, x(age=40) diff brief
logit: Change in Predictions for lfp
                    Current      Saved     Change   95% CI for Change
  Pr(y=inLF|x):      0.6162     0.7506    -0.1345  [-0.1784,  -0.0906]
  Pr(y=NotInLF|x):   0.3838     0.2494     0.1345  [ 0.0906,   0.1784]
```

The `save` option preserves the results from the first call of `prvalue`. The second call adds the `diff` option to compute the differences between the two sets of predictions. We find that an increase in `age` from 30 to 40 years decreases a woman's probability of being in the labor force by .13.

Nonstandard discrete changes with prchange

We can also use `prchange` with the `delta()` and `uncentered` options. `delta(#)` specifies that the discrete change is to be computed for a change of # units instead of a one-unit change. `uncentered` specifies that the change should be computed starting at the base value (i.e., values set by the `x()` and `rest()` options), rather than being centered on the base. Here we want an uncentered change of 10 units, starting at age=30:

```
. prchange age, x(age=30) uncentered delta(10) rest(mean) brief
        min->max       0->1      +delta        +sd  MargEfct
age    -0.4372    -0.0030    -0.1345    -0.1062    -0.0118
```

The result under the heading `+delta` is the same as what we just calculated using `prvalue`.

4.7 Interpretation using odds ratios with listcoef

Effects for the logit model, but *not* probit, can be interpreted in terms of changes in the odds. Recall that for binary outcomes, we typically consider the odds of observing a positive outcome versus a negative one:

$$\Omega = \frac{\Pr(y=1)}{\Pr(y=0)} = \frac{\Pr(y=1)}{1-\Pr(y=1)}$$

Recall also that the log of the odds is called the *logit* and that the logit model is *linear in the logit*, meaning that the log odds are a linear combination of the x's and βs. For example, consider a logit model with three independent variables:

$$\ln\left\{\frac{\Pr(y=1 \mid \mathbf{x})}{1-\Pr(y=1 \mid \mathbf{x})}\right\} = \ln\Omega(\mathbf{x}) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3$$

We can interpret the coefficients as indicating that for a unit change in x_k, we expect the logit to change by β_k, holding all other variables constant.

This interpretation does *not* depend on the level of the other variables in the model. The problem is that a change of β_k in the log odds has little substantive meaning for most people (including us). By taking the exponential of both sides of this equation, we can also create a model that is multiplicative instead of linear but in which the outcome is the more intuitive measure, the odds:

$$\Omega(\mathbf{x}, x_2) = e^{\beta_0} e^{\beta_1 x_1} e^{\beta_2 x_2} e^{\beta_3 x_3}$$

where we take particular note of the value of x_2. If we let x_2 change by 1,

$$\begin{aligned}\Omega(\mathbf{x}, x_2+1) &= e^{\beta_0} e^{\beta_1 x_1} e^{\beta_2 (x_2+1)} e^{\beta_3 x_3} \\ &= e^{\beta_0} e^{\beta_0} e^{\beta_1 x_1} e^{\beta_2 x_2} e^{\beta_2} e^{\beta_3 x_3}\end{aligned}$$

which leads to the *odds ratio:*

$$\frac{\Omega(\mathbf{x}, x_2+1)}{\Omega(\mathbf{x}, x_2)} = \frac{e^{\beta_0}e^{\beta_1 x_1}e^{\beta_2 x_2}e^{\beta_2}e^{\beta_3 x_3}}{e^{\beta_0}e^{\beta_1 x_1}e^{\beta_2 x_2}e^{\beta_3 x_3}} = e^{\beta_2}$$

Accordingly, we can interpret the exponential of the coefficient as follows:

> For a unit change in x_k, the *odds* are expected to change by a factor of $\exp(\beta_k)$, holding all other variables constant.

For $\exp(\beta_k) > 1$, you could say that the odds are "$\exp(\beta_k)$ times larger"; for $\exp(\beta_k) < 1$, you could say that the odds are "$\exp(\beta_k)$ times smaller". We can evaluate the effect of a standard deviation change in x_k instead of a unit change:

> For a standard deviation change in x_k, the odds are expected to change by a factor of $\exp(\beta_k \times s_k)$, holding all other variables constant.

The odds ratios for both a unit and a standard deviation change of the independent variables can be obtained with `listcoef`:

```
. listcoef, help
logit (N=753): Factor Change in Odds
  Odds of: inLF vs NotInLF
```

lfp	b	z	P>|z|	e^b	e^bStdX	SDofX
k5	-1.46291	-7.426	0.000	0.2316	0.4646	0.5240
k618	-0.06457	-0.950	0.342	0.9375	0.9183	1.3199
age	-0.06287	-4.918	0.000	0.9391	0.6020	8.0726
wc	0.80727	3.510	0.000	2.2418	1.4381	0.4500
hc	0.11173	0.542	0.588	1.1182	1.0561	0.4885
lwg	0.60469	4.009	0.000	1.8307	1.4266	0.5876
inc	-0.03445	-4.196	0.000	0.9661	0.6698	11.6348

```
      b = raw coefficient
      z = z-score for test of b=0
  P>|z| = p-value for z-test
    e^b = exp(b) = factor change in odds for unit increase in X
e^bStdX = exp(b*SD of X) = change in odds for SD increase in X
  SDofX = standard deviation of X
```

Some examples of interpretations are as follows:

> For each additional young child, the odds of being employed decrease by a factor of .23, holding all other variables constant.
>
> For a standard deviation increase in the log of the wife's expected wages, the odds of being employed are 1.43 times greater, holding all other variables constant.
>
> Being 10 years older decreases the odds by a factor of .53 ($= e^{[-.063]\times 10}$), holding all other variables constant.

Other ways of computing odds ratios Odds ratios can also be computed with the `or` option for `logit`. This approach does not, however, report the odds ratios for a standard deviation change in the independent variables.

Multiplicative coefficients

When interpreting the odds ratios, remember that they are multiplicative. This means that positive effects are greater than one and negative effects are between zero and one. *Magnitudes of positive and negative effects should be compared by taking the inverse of the negative effect (or vice versa).* For example, a positive factor change of 2 has the same magnitude as a negative factor change of $.5 = 1/2$. Thus a coefficient of $.1 = 1/10$ indicates a stronger effect than a coefficient of 2. Another consequence of the multiplicative scale is that to determine the effect on the odds of the event not occurring, you simply take the inverse of the effect on the odds of the event occurring. `listcoef` will automatically calculate this for you if you specify the `reverse` option:

```
. listcoef, reverse
logit (N=753): Factor Change in Odds
  Odds of: NotInLF vs inLF
```

lfp	b	z	P>|z|	e^b	e^bStdX	SDofX
k5	-1.46291	-7.426	0.000	4.3185	2.1522	0.5240
k618	-0.06457	-0.950	0.342	1.0667	1.0890	1.3199
age	-0.06287	-4.918	0.000	1.0649	1.6612	8.0726
wc	0.80727	3.510	0.000	0.4461	0.6954	0.4500
hc	0.11173	0.542	0.588	0.8943	0.9469	0.4885
lwg	0.60469	4.009	0.000	0.5462	0.7010	0.5876
inc	-0.03445	-4.196	0.000	1.0350	1.4930	11.6348

The header indicates that these are now the factor changes in the odds of `NotInLF` versus `inLF`, whereas before we computed the factor change in the odds of `inLF` versus `NotInLF`. We can interpret the result for `k5` as follows:

> For each additional child, the odds of not being employed are increased by a factor of 4.3 (=1/.23), holding other variables constant.

Effect of the base probability

The interpretation of the odds ratio assumes that the other variables have been held constant, but it does not require that they be held at any specific values. Although the odds ratio seems to resolve the problem of nonlinearity, remember: *a constant factor change in the odds does not correspond to a constant change or constant factor change in the probability.* For example, if the odds are 1/100, the corresponding probability

is .01.[6] If the odds double to 2/100, the probability increases only by approximately .01. Depending on your substantive purposes, this small change may be trivial or quite important (such as when you identify a risk factor that makes it twice as likely that a subject will contract a fatal disease). Meanwhile, if the odds are 1/1 and double to 2/1, the probability increases by .167. Accordingly, the meaning of a given factor change in the odds depends on the predicted probability, which in turn depends on the levels of all variables in the model.

Percent change in the odds

Instead of a multiplicative or factor change in the outcome, some people prefer the percent change,

$$100\{\exp(\beta_k \times \delta) - 1\}$$

which is listed by `listcoef` with the `percent` option.

```
. listcoef, percent
logit (N=753): Percentage Change in Odds
  Odds of: inLF vs NotInLF
```

lfp	b	z	P>|z|	%	%StdX	SDofX
k5	-1.46291	-7.426	0.000	-76.8	-53.5	0.5240
k618	-0.06457	-0.950	0.342	-6.3	-8.2	1.3199
age	-0.06287	-4.918	0.000	-6.1	-39.8	8.0726
wc	0.80727	3.510	0.000	124.2	43.8	0.4500
hc	0.11173	0.542	0.588	11.8	5.6	0.4885
lwg	0.60469	4.009	0.000	83.1	42.7	0.5876
inc	-0.03445	-4.196	0.000	-3.4	-33.0	11.6348

With this option, the interpretations would be the following:

> For each additional young child, the odds of being employed decrease by 77%, holding all other variables constant.
>
> A standard deviation increase in the log of the wife's expected wages increases the odds of being employed by 43%, holding all other variables constant.

Percentage and factor change provide the same information; which you use for the binary model is a matter of preference. Although we both tend to prefer percentage change, methods for the graphical interpretation of the multinomial logit model (chapter 6) work only with factor change coefficients.

6. The formula for computing probabilities from odds is $p = \Omega/1 + \Omega$.

Additional note If you report the odds ratios instead of the untransformed coefficients, the 95% confidence interval of the odds ratio is typically reported instead of the standard error. The reason is that the odds ratio is a nonlinear transformation of the logit coefficient, so the confidence interval is asymmetric. For example, if the logit coefficient is .75 with a standard error of .25, the 95% interval around the logit coefficient is approximately [.26, 1.24], but the confidence interval around the odds ratio exp(.75)=2.12 is [exp(.26)=1.30, exp(1.24)=3.46]. Using the `or` option with the `logit` command reports odds ratios and includes confidence intervals.

4.8 Other commands for binary outcomes

Logit and probit models are the most commonly used models for binary outcomes and are the only ones that we consider in this book, but other models exist that can be fitted in Stata. Among them, `cloglog` assumes a complementary log-log distribution for the errors instead of a logistic or normal distribution. `scobit` fits a logit model that relaxes the assumption that the marginal change in the probability is greatest when $\Pr(y = 1) = .5$. `hetprob` allows the assumed variance of the errors in the probit model to vary as a function of the independent variables. `ivprobit` fits a probit model where one or more of the regressors are endogenously determined. `biprobit` simultaneously fits two binary probits and can be used when errors are correlated with each other as in the estimation of seemingly unrelated regression models for continuous dependent variables.