

Regression with a binary dependent variable

So far we have regularly used binary (dummy) variables as regressors and they caused no particular problems. But when the DV is binary, things are more difficult: **what does it mean to fit a line to a DV that can take on only two values, zero and one?**

One possible answer to this question is to interpret the regression function as a **predicted probability**. But the predicted probability interpretation also suggests that alternative **nonlinear regression models** can do a better job modeling these probabilities. These models are called **“probit” and “logit” regression**. **The method used to estimate the coefficients of the probit and logit regressions is the method of maximum likelihood estimation.**

Binary DV and the Linear Probability Model

The simplest model to use with binary DV is using an OLS, producing what we call “a linear probability model”.

Example: Using the dataset on California schools ([school.dta](#)), we want to explain the quality of a high school in California `hiqual`. Such variable has only two values (“quality is high”=1 or “quality is not high”=0). As IV we employ the average education `avg_ed` (ranging from 1 to 5) of the parents of the students in the high schools.

```
twoway (scatter hiqual avg_ed) (lfit hiqual avg_ed)
```

The scatterplot looks different than the other scatterplots you plotted in the first module because we now have a binary DV. Still, the graph seems to show a positive relationship between the IV and the DV. There are a lot of low quality schools when parents’ education is low, and a lot of high quality schools when parents’ education is high.

Let’s estimate an OLS regression then:

```
regress hiqual avg_ed
```

The regression gives us a coefficient of 0.43. What about expected values? For example, when parents’ education is equal to 3, then the predicted (or expected) value of `hiqual` is 0.43.

```
margins, at(avg_ed=3)
```

And what is the expected value of the binary variable `hiqual` when parents' education is fixed at its maximum value?

Coming back to the example above, what does it mean for the predicted value of the binary DV to be 0.43? The key to answering this question, and more generally to understanding regression with a binary DV, is to interpret the regression as modeling the probability that the DV equals 1. Thus, the predicted value of 0.43 is interpreted as meaning that when education is 3, then the probability of having a high quality school is estimated to be 43%. Said differently, if there were schools with `avg_ed=3`, then 43% of them would be school of high quality.

This interpretation follows from two facts. First, the regression function tells us the expected value of Y given the regressors: $E(Y|X_1, \dots, X_k)$. Second, if Y is a binary variable, then its expected value (or mean) is the probability that $Y=1$, that is $E(Y)=\Pr(Y=1)$. Thus for a binary variable, $E(Y|X_1, \dots, X_k) = \Pr(1|X_1, \dots, X_k)$. In short, for a binary DV the predicted (expected) value from the regression is the probability that $Y=1$, given (conditional on) X .

Addendum:

The expected value of a random variable Y , denoted $E(Y)$ is the long-run average value of the DV over many repeated trials or occurrences. The expected value of Y is also called the expectation of Y or the mean of Y .

The expected value of a *Bernoulli random variable* (i.e., a binary variable with just two occurrence: 1 with probability p and 0 with probability $(1-p)$) is:

$$E(Y) = 1 \cdot p + 0 \cdot (1-p) = p$$

Thus, the expected value of a Bernoulli random variable is p , which is the probability that it takes on the value "1".

The linear probability model is the name for the OLS when the DV is binary rather than continuous. Because the dependent variable Y is binary, the regression function corresponds to the probability that the DV equals 1, given X . The coefficient β_1 on a regressor X is the change in the probability that $Y=1$ associated with a unit change in X .

Problems with the linear probability model:

1) the R^2 has no meaning here. When the DV is continuous, it is possible to imagine a situation in which $R^2 = 1$ (all the data lie exactly on the regression line). This is impossible when the DV is binary (look at the scatter before!)

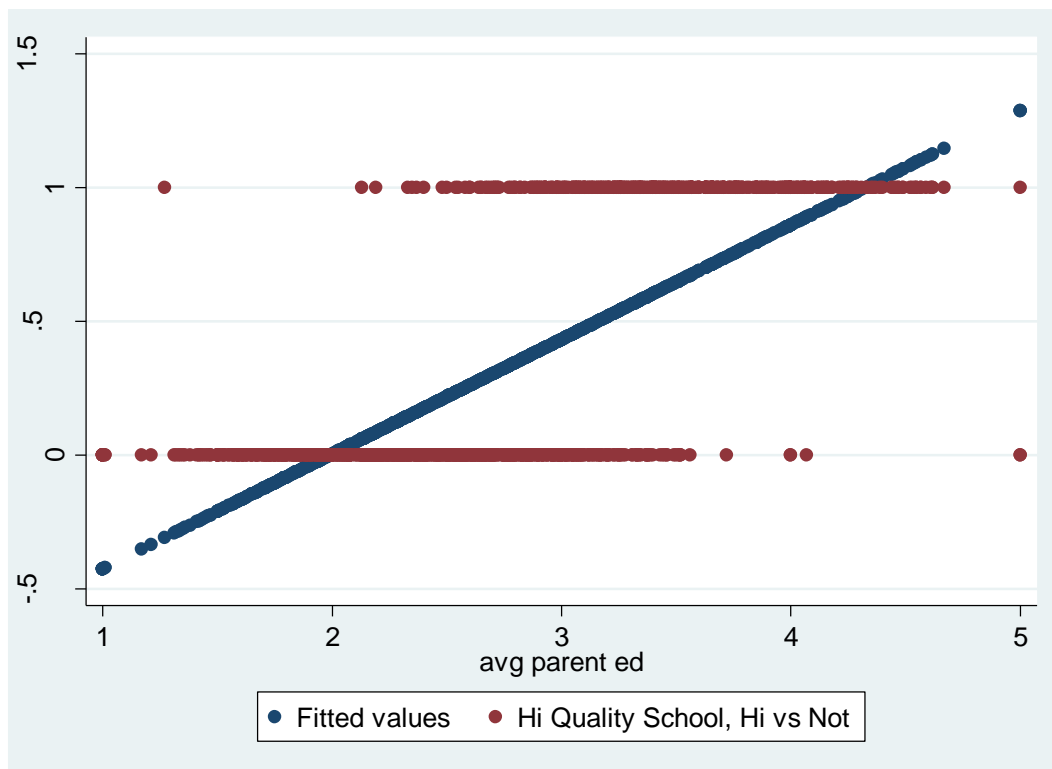
2) Take a look at here:

```
predict yhat
```

```
twoway scatter yhat hiqual avg_ed, ylabel(0 1)
```

In the graph we have plotted the predicted values (called "fitted values" in the legend, the blue line) along with the observed data values (the red dots). Upon inspecting the graph, you will notice that

some things do not make sense. First of all, there are predicted values that are less than zero and other predicted values that are greater than +1. Such values are not possible given the nature of our outcome variable. The estimated line representing the predicted probabilities drops below 0 for very low values of parents' education and exceeds 1 for high values of parents' education. This is non-sense! A probability cannot be less than 0 or greater than 1! This nonsensical feature is an inevitable consequence of the linear regression (i.e., of our attempt to fit a line with a dichotomous DV). Also, the line does a poor job of "fitting" or "describing" the data points.



To address this problem, we need to do something else. In other words, the linear probability model is easier to use and to interpret, but it cannot capture the nonlinear nature of the true population regression function!

Probit and Logit regression

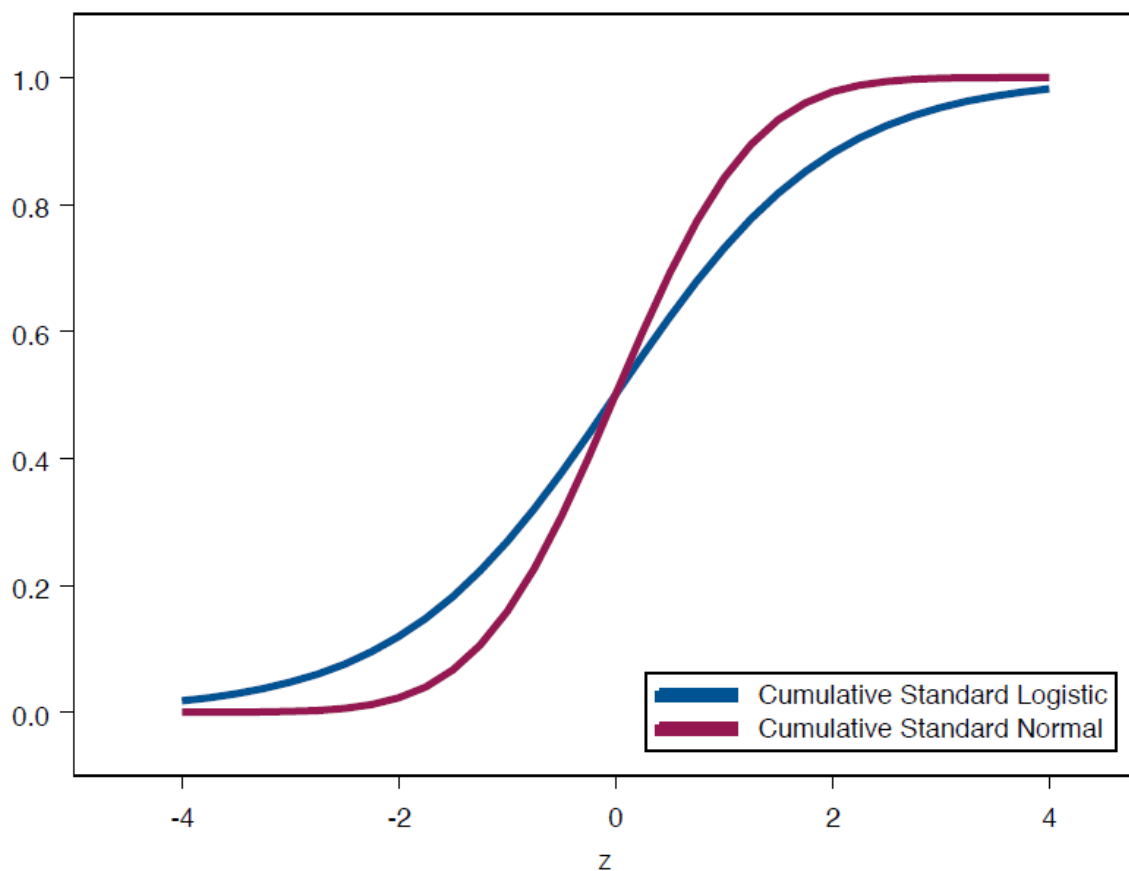
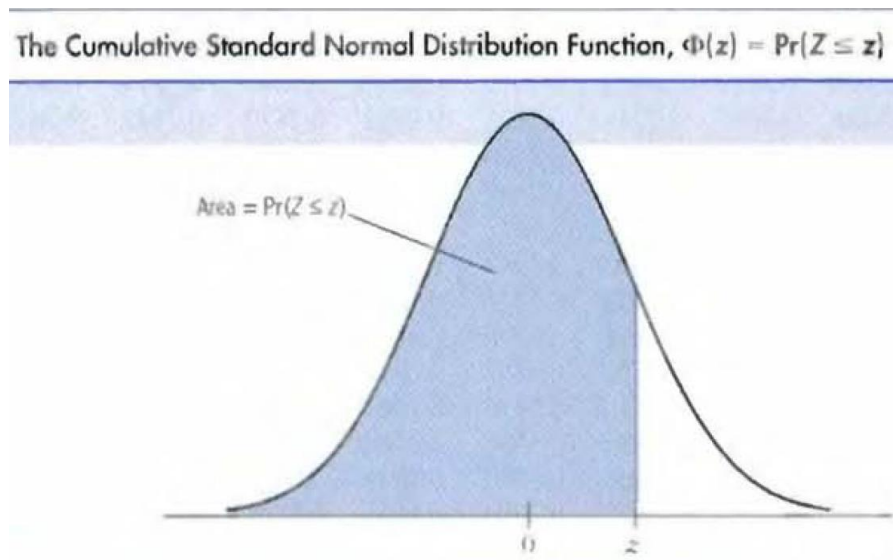
Probit and Logit regression are nonlinear regression models specifically designed for a binary DV. Because a regression with a binary DV models the probability that $Y=1$, it makes sense to adopt a nonlinear formulation that forces the predicted values to be between 0 and 1.

Because CDFs (cumulative probability distribution functions) produce probabilities between 0 and 1, they are used in logit and probit regressions. Probit regression uses the standard normal CDF. Logit regression uses the Logistic CDF.

In both situations, the arguments of the CDF depends on the regressors.

Addendum:

The **cumulative probability distribution** is the probability that a random variable is less than or equal to a particular value. Overall, a CDF ranges between 0 and 1.



Probit regression uses the cumulative standard normal.

Logit regression uses the cumulative standard logistic.

Probit regression

The probit regression model with a single regressor X is

$$\Pr(Y=1|X) = \varphi(\beta_0 + \beta_1 X_i) \quad (1)$$

where φ (phi) is the cumulative standard normal distribution function.

Let's run now the following probit regression model, where Y is the binary variable indicating whether a school is of high quality or not, X is the average education of parents:

```
probit hiqual avg_ed
```

From the result table, we can note that the estimated coefficients are $\beta_0 = -6.42$ and $\beta_1 = 2.03$. What is the probability that a school is of high quality if parents' average education equals 3? According to equation (1), this probability is $\varphi(\beta_0 + \beta_1 \cdot \text{avg_ed}) = \varphi(-6.42 + 2.03 \cdot 3) = \varphi(-0.33)$.

In the probit model, the term $\beta_0 + \beta_1 X_i$ plays the role of “ z ” in the cumulative standard normal distribution table. Thus, the calculation we just saw can be equivalently done by first computing the “ z value”, $z = (\beta_0 + \beta_1 \cdot \text{avg_ed}) = (-6.42 + 2.03 \cdot 3) = (-0.33)$, and then looking up the probability in the tail of the normal distribution to the left of $z = -0.33$, which is 37%.

Taking a look at the cumulative normal distribution table (see the pdf file: “Cumulative Standard Normal Distribution Function”) for $z = -0.33$, the value is equal to 0.3707 (that means 37%). That is, when avg_ed is 3, the predicted probability that the school will be of a good quality is 37%, computed using the probit model with the coefficients $\beta_0 = -6.42$ and $\beta_1 = 2.03$ (remember that, with the linear probability model, a school's probability of being of good quality was 43%!).

To this purpose, the `margins` command can be really useful.

Here the `margins` command gives you the probability $\Pr(Y=1|X=3)$:

```
margins, at(avg_ed=3)
```

```
marginsplot
```

Here the `margins` command gives the linear prediction (not the probability!): we get our usual value: -0.33!

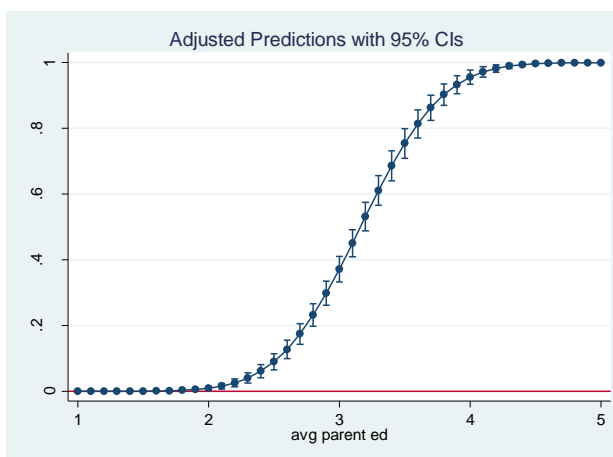
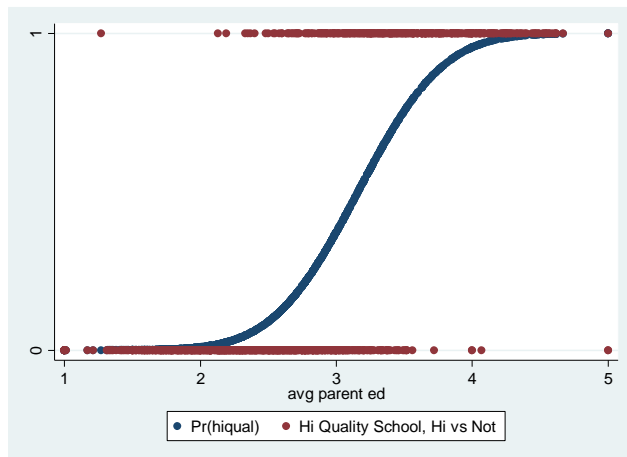
```
margins, at(avg_ed=3) predict(xb)
```

If β_1 in equation (1) is positive, then an increase in X increases the probability that $Y=1$. Conversely, if β_1 in equation (1) is negative, then an increase in X decreases the probability that $Y=1$. Beyond this, however, **it is not easy to interpret the probit coefficients directly**. Instead, the coefficients are best interpreted indirectly by **computing probabilities and/or changes in probabilities**. When there is just one regressor, the easiest way to interpret a probit regression is to plot the probabilities. The estimated probit regression function has a stretched “S” shape: it is nearly zero and flat for small values of avg_ed ; it turns and increases for intermediate level; and it flattens out again and is nearly one for large values.

```
probit hiqual avg_ed
predict yhat1
twoway scatter yhat1 hiqual avg_ed, ylabel(0 1)
```

Alternatively:

```
probit hiqual avg_ed
margins, at(avg_ed=(1 (0.1) 5))
marginsplot
```



For example, for $\text{avg_ed} = 2$, the estimated probability of having a school of high quality is less than 1%. When $\text{avg_ed} = 3$, the estimated probability of having a school of high quality is 37%. When $\text{avg_ed} = 4$, the estimated probability of having a school of high quality is 96%. Note that, unlike the linear probability model, the probit conditional probabilities are always between 0 and 1. Also, the probit regression curve does a much better job of “fitting” or “describing” the data points.

And what about the effect of a change in X ? The expected change is estimated in three steps: first, compute the predicted value at the original value of X using the estimated regression function; next, compute the predicted value at the changed value of X ; then compute the difference between the two predicted values. Why always doing such procedure? Because the probit regression function is non-linear! Therefore, the effect of a change in X depends on the starting value of X (i.e., moving avg_ed from 2 to 3 increases Y by 36 percentage points; moving avg_ed from 3 to 4 increases Y by 59 percentage points).

```
probit hiqual avg_ed
margins, at(avg_ed=(2 3)) contrast(atcontrast(r._at) wald)
margins, at(avg_ed=(3 4)) contrast(atcontrast(r._at) wald)
```

More in detail: the nonlinear models studied in the previous lectures are nonlinear functions of the IVs but are linear functions of the unknown coefficients (parameters). Consequently, the unknown coefficients of those nonlinear regression functions can be estimated by OLS. In contrast, the probit

regression functions are a nonlinear function of the coefficients, given that the probit coefficients appear inside the CDF.

Because the population regression function is a nonlinear function of the coefficients, those coefficients are more complicated to estimate than linear regression functions. The coefficients of the probit model can be estimated via maximum likelihood. The maximum likelihood estimator is consistent and normally distributed in large samples, so that confidence intervals for the coefficients can be constructed in the usual way.

Notice that writing:

```
margins, at(avg_ed=(2 3)) contrast(atcontrast(r._at) wald)
```

is different from writing:

```
margins, dydx(avg_ed) at(avg_ed=(2))
```

because the effect is not linear, then the `dydx(avg_ed)` will tell you the impact on Y when your X is equal to 2; but when passing from 2 to 3, the average effect will be the mean of increasing X when X is equal to: 2; 2.1; 2.2; 2.3; ...; 3

Logit regression

The logit regression model is similar to the probit regression model, except that the cumulative normal distribution function Φ in equation (1) is replaced by the cumulative standard logistic distribution function, which we denote by F. The logistic cumulative distribution function has a specific functional form, defined in terms of the exponential function, which is given in equation (2) below (assuming multiple regressors)

$$\Pr(Y=1|X_1, \dots, X_k) = F(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_k X_k) = \frac{1}{1 + \exp^{-(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_k X_k)}} \quad (2)$$

As with the probit, the logit coefficients are best interpreted by computing predicted probabilities and differences in predicted probabilities.

```
logit h1qual avg_ed
```

To obtain a probability we have put – through the formula above – the estimated coefficients in the logit CDF!

What is the probability of having a school of high quality when parents' education equals 3 now?

```
di 1/(1+exp(-( -12.30054 + 3*3.909635 )))
```

Alternatively:

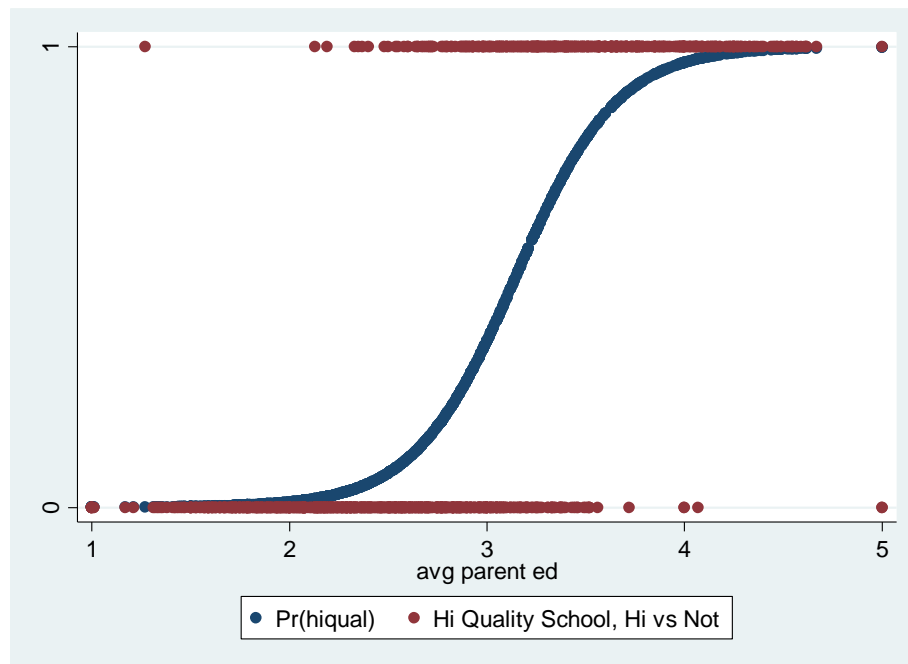
```
di exp( -12.30054 + 3*3.909635 )/(1+exp( -12.30054 + 3*3.909635 ))
```

Alternatively:

```
margins, at(avg_ed=3)
```

That is, with logit the estimated probability of having a school of high quality is 36.1%. With probit, it was 37%.

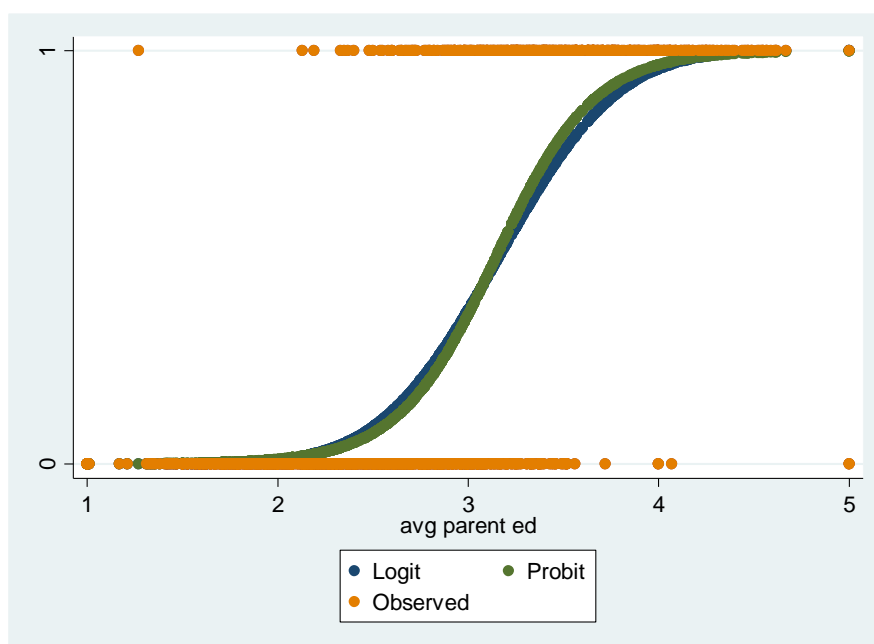
```
logit hiqual avg_ed
predict yhat2
twoway scatter yhat2 hiqual avg_ed, ylabel(0 1)
```



As we did before with probit, we have now calculated the predicted probabilities and have graphed them against the observed values. With the logistic regression (just as with the probit), we get predicted probabilities that make sense: no predicted probabilities is less than zero or greater than one.

Let's compare the probit with the logit prediction:

```
corr yhat1 yhat2
twoway (scatter yhat1 hiqual avg_ed, ylabel(0 1)) ///
(scatter yhat2 hiqual avg_ed, ylabel(0 1)), legend(order(1 "Logit" 3
"Probit" 4 "Observed"))
```

As we can see, the logit and the probit regression functions are pretty similar (correlation = 0.999!!!). So, which should you use in practice? There is no one right answer. Just pick up the one you like most. Historically the main motivation for logit regression was that the logistic cumulative distribution function could be computed faster than the normal CDF. But now this distinction is no longer important.

Addendum: A different (but related interpretation) of the logit coefficients:

Probability is defined as the quantitative expression of the chance that an event will occur. More precisely, it is the number of times the event “occurs” divided by the number of times the event “could occur”. For a simple example, let’s consider tossing a coin. On average, you get heads once out of every two tosses. Hence, the probability of getting heads is 1/2 or 0.5.

Let’s now consider the **odds**. In statistics, probability and odds are not the same. The odds of an event happening is defined as the probability that the event occurs divided by the probability that the event does not occur – that is, $p/(1-p)$. To continue with our coin-tossing example, the probability of getting heads is 0.5 and the probability of not getting heads (i.e., getting tails) is also 0.5. Hence, the odds are $0.5/0.5 = 1$. Note that the probability of an event happening and its complement, the probability of the event not happening, must sum to 1. Now let’s pretend that we alter the coin so that the probability of getting heads is 0.6. The probability of not getting heads is then 0.4. The odds of getting heads is $0.6/0.4 = 1.5$. If we had altered the coin so that the probability of getting heads was 0.2, then the odds of getting heads would have been $0.2/0.8 = 0.25$. As you can see, when *the odds equal one*, the probability of the event happening is equal to the probability of the event not happening. When *the odds are greater than one*, the probability of the event happening is higher than the probability of the event not happening, and when *the odds are less than one*, the probability of the event happening is less than the probability of the event not happening.

Also note that odds can be converted back into a probability: $\text{probability} = \text{odds} / (1 + \text{odds})$.

In summary:

- probability: the number of times the event occurs divided by the number of times the event could occur (possible values range from 0 to 1)

- odds: the probability that an event will occur divided by the probability that the event will not occur: $\text{probability}(\text{success}) / \text{probability}(\text{failure})$

Note that *Stata* has two commands for logistic regression, **logit** and **logistic**. The main difference between the two is that the former displays the coefficients and the latter displays the odds. You can also obtain the odds ratios by using the **logit** command with the **or** option. Which command you use is a matter of personal preference.

From logit coefficients to odds: **Log odds** are the natural logarithm of the odds, that is $\ln(p/(1-p))$. The coefficients reported in the output of the logit regression are given in units of log odds. Therefore, the coefficients indicate the amount of change expected in the log odds when there is a one unit change in the predictor variable.

But then $\exp^{\ln(p/(1-p))} = p/(1-p)$

```
logit hiqua1 avg_ed
```

If you type: `di exp(3.909635)`

then you have the odds ratio of increasing by 1 unit avg_ed!

```
logit hiqua1 avg_ed, or
```

as an alternative:

```
logistic hiqua1 avg_ed
```

An odds ratio of 49.8 means that a school at a given level of average parents' education is 49.8 times more likely to be a school with good quality than a school at the next lower level of average education.

Another example: explaining the turnout in the US Presidential election 2004 (using nes2004.dta):

```
logit vote_2004 educ
```

As already discussed, the logit assumes a nonlinear relationship between years of education and the probability of voting. That is, it is assumed that for people located near the extremes of the IV (respondents with either low or high level of education) a 1-year increase in education will have a weaker effect on the probability of voting than will a 1-year increase for respondents in the middle range of the IV. Let's try to understand it better by using the **predict** command.

```
predict yhat
```

```
tab educ, sum(yhat) nost
```

Notice that the increments are higher in the middle range of education.

```
twoway scatter yhat vote_2004 educ, ylabel(0 1)
```

Addendum: Things to consider when estimating a probit (or logit) model

As we have stated, probit (and logit) regression uses a maximum likelihood to get the estimates of the coefficients. Many of desirable properties of maximum likelihood are found as the sample size increases. The behavior of maximum likelihood with small sample sizes is not well understood. In particular, in small samples one may get high standard errors. In the extreme, if there are too few cases in relation to the number of variables, it may be impossible to converge on a solution. According to Long (1997, pp. 53-54), 100 is a minimum sample size, and you want *at least* 10 observations per predictor. This does not mean that if you have only one predictor you need only 10 observations. Peduzzi et al. (1996) recommend that the smaller of the classes of the dependent variable have at least 10 events per parameter in the model. Pedhazur (1997) recommends that sample size should be at least 30 times the number of parameters being estimated. More observations are needed when the dependent variable is very lopsided; in other words, when there are very few 1's and lots of 0's, or viceversa. Moreover, when you have problems with multicollinearity, you will need a larger sample size.

Similarly, it is always better to check for empty or small cells by doing a crosstab between categorical predictors and the outcome variable. If a cell has very few cases (a small cell), the model may become unstable or it might not run at all.

It is however sometimes possible to estimate models for binary outcomes in datasets with only a small number of cases and/or when some of the cells formed by the outcome and categorical predictor variable have no observations using **exact logistic regression** (the `exlogistic` command).

It is also important to keep in mind that when the outcome is rare, even if the overall dataset is large, it can be difficult to estimate a probit or logit model. In that case it would be more appropriate to use for example a “**Rare event logistic regression**” (the `relogit` command thanks to Gary King! <http://gking.harvard.edu/files/abs/0s-abs.shtml>).

Measures of fit

As a measure of fit after a logit, you can rely on different statistics.

First, you can **compare the initial log-likelihood and the final one** to have an idea of how well a model performs. For example:

```
logit vote_2004 educ
```

If education greatly improves the model's predictive power, then the two logs would be very different, and the final log would be much closer to 0 than the initial one. Recall that 0 is the maximum value of log-likelihood because likelihoods can vary between 0 and 1, the logs of likelihoods can vary between large negative numbers and 0 (the log of 1).

We have two measures to assess the strength of the relationship between the IV and the likelihood of the DV: the likelihood ratio (LR chi2) and pseudo R². The **likelihood ratio** is obtained in the following way: $-2(\text{initial log likelihood} - \text{final log likelihood})$.

```
di -2*(-553.07398 - -509.37393) .
```

Then you have a test on this (where H0 is that your model does not improve anything against a know-nothing model). In particular, you have to compare the result with the Chi Squared distribution to

understand if such difference is significant or not. As an alternative, you can rely on the usual `lrtest` (null hypothesis: the coefficients of the variables excluded from the reduced model are simultaneously equal to 0. This would mean that removing those variables has no effect) :

```
logit vote_2004 educ
est store full_model
logit vote_2004 if e(sample)
lrtest full_model .
```

Alternatively, you can use a **pseudo R^2** to seek to communicate the strength of association between DV and IV (because of the statistical foundations of logistic regression, the notion of explained variance/variation has not a comparable notion in logistic regression). It is a “pseudo” R^2 because it is different from the R^2 found in OLS regression, where R^2 measures the proportion of variance explained by the model. The pseudo R^2 is not *measured* in terms of variance. This is because in logistic regression the variance is fixed as the variance of the standard logistic distribution (i.e., $\pi^2/3$). Moreover, the variance of the standard normal distribution employed in a probit is once again fixed and equals to 1.

Stata reports the measure suggested by McFadden:

(initial log likelihood – final log likelihood) / (initial log likelihood)

```
di (-553.07398 - -509.37393) / (-553.07398)
```

However, note that statisticians have elaborated many versions of the pseudo R^2 . We should also note that different pseudo R^2 s can give very different assessments of a model’s fit, and that there is no one version of pseudo R^2 that is preferred by most data analysts over other versions.

Another commonly used test of model fit is to estimate the “**fraction correctly predicted**”. Such measure uses the following rule: if $Y_i=1$ and the predicted probability exceeds 50% or if $Y_i=0$ and the predicted probability is less than 50%, then Y_i is said to be correctly predicted. Otherwise, Y_i is said to be incorrectly predicted. The “fraction correctly predicted” is the fraction of the n observations in your sample that are correctly predicted by your model. An advantage of this measure of fit is that it is easy to understand. A disadvantage is that it does not reflect the quality of the prediction: if $Y_i=1$, the observation is treated as correctly predicted whether the predicted probability is 51% or 99%.

```
estat classification
di (821+19)/1065
di 821/837
di 19/228
```

In our case, the overall rate of correct classification is estimated to be 78.87%, with 98.09% of the voting group correctly classified (*sensitivity*: 821 out of 837) and only 8.3% of the not-voting group correctly classified (*specificity*: 19 out of 228). **Classification is sensitive to the relative sizes of each component group, and always favors classification into the larger group.** This phenomenon is evident

here. By default, `estat classification` uses a cutoff of 0.5, although you can vary this with the `cutoff()` option.

```
estat class, cut(0.1)
```

Relatedly, a useful way of comparing the predictive ability of different models is to use the **Receiver Operating Characteristic, or ROC curve** (by Hosmer and Lemeshow). This curve plots the probability of detecting a true signal (sensitivity) and false signal (1—specificity) for the entire range of possible cutpoints. The area under the ROC curve (denoted AUC) provides a measure of the model's ability to discriminate. A value of 0.5 indicates no ability to discriminate (might as well toss a coin) while a value of 1 indicates perfect ability to discriminate, so the effective range of AUC is from 0.5 to 1.0. Hosmer-Lemeshow indicate that AUC of 0.5 indicates no discrimination, AUC of between 0.7 and 0.8 indicates acceptable discrimination, AUC of between 0.8 and 0.9 indicates excellent discrimination, and AUC greater than 0.9 is considered outstanding discrimination.

```
lroc
```

Another commonly used test of model fit is the **Hosmer and Lemeshow's goodness-of-fit test**. The idea behind the Hosmer and Lemeshow's goodness-of-fit test is that the predicted frequency and observed frequency should match closely, and that the more closely they match, the better is the fit. The Hosmer-Lemeshow goodness-of-fit statistic is computed as the Pearson chi-squared from the contingency table of observed frequencies and expected frequencies.

The command `estat gof` presents the Pearson chi-squared goodness-of-fit test for the fitted model. The Pearson chi-squared goodness-of-fit test is a test of the observed against expected number of responses using cells defined by the covariate patterns.

```
estat gof, table
```

Why a table with 15 groups?

```
codebook educ if e(sample)
```

```
table educ if vote_2004!=.
```

Similar to a test of association of a two-way table, a good fit as measured by Hosmer and Lemeshow's test will yield a large p-value (i.e., the difference between observed and predicted frequency is not significantly different from 0). With a p-value of 0.1930 (take a look also at the table of the Chi-Squared if you are unsure!), we can say that Hosmer and Lemeshow's goodness-of-fit test indicates that our model fits the data well.

When there are continuous predictors in the model, there will be many cells defined by the predictor variables, making a very large contingency table, which would yield significant result more than often. Therefore, a common practice is to regrouping the data by ordering on the predicted probabilities and then forming, say, 4 nearly equal-sized groups.

```
estat gof, group(4) table
```

Logistic regression with multiple IVs

Let's see what happens if we add age to our model.

```
logit vote_2004 educ
logit vote_2004 educ age
```

The first thing note is that now the education coefficient has increased compared to the previous situation (without age). Why this? Above all note that the correlation between age and education is negative. Thus in our first analysis (without age) we were also comparing older respondents (who, on average, have fewer years of schooling) with younger respondents (who, on average, have more years of schooling). Since younger people are less likely to vote than are older people, the uncontrolled effect of age contributed to weaken the relationship between education and vote_2004. In a situation like this, age is said to be a *suppressor variable*, because it suppresses or attenuates education's true effect on turnout.

Overall, how does the model with education and age performs compared to the model with just age? The pseudo R^2 increases. And the log-likelihood increases. Still, is our second model significantly better than the more parsimonious model with education only? Remember that in the OLS we had the adjusted R^2 . What about here with the logit? Let's use the likelihood ratio test!

To do it, you first run the model that you want to use as the basis for comparison (the full model). Next, you save the estimates with a name using the `estimates store` command. Next, you run the model that you want to compare to your full model, and then use the `lrtest` command with the name of the full model. In our example, we will name our full model "full_model". The output of this is a likelihood ratio test which tests the null hypothesis that the coefficients of the variable(s) left out of the reduced model is/are simultaneously equal to 0. In other words, the null hypothesis for this test is that removing the variable(s) has no effect; it does not lead to a poorer-fitting model.

```
logit vote_2004 educ age
est store full_model
logit vote_2004 educ if e(sample)
lrtest full_model .
```

The chi-square statistic equals 29.26, which is statistically significant. This means that the variable that was removed to produce the reduced model resulted in a model that has a significantly poorer fit, and therefore the variable should be included in the model.

Now let's take a moment to make a few comments on the code used above. For the second logit (i.e. for the reduced model), we have added `if e(sample)`, which tells STATA to only use the cases that were included in the first model. If there were missing data on one of the variables (here age) that was present in the full model but not in the reduced model, there would be more cases used in the reduced model. Using exactly the same cases in both models is important because the `lrtest` assumes that the same cases are used in each model. It is not necessary to include the dot (.) at the end of the `lrtest` command, but we have included it to be explicit about what is being tested. STATA automatically "names" a model as . if you have not specifically named it.

For another example, imagine that you have a model with lots of predictors in it. You could run many variations of the model, dropping one variable at a time or groups of variables at a time. Each time that you run a model, you would use the `estimates store` command and give each model its own name. We will try a mini-example below.

```

* full model
logit vote_2004 educ age income_hh
est store a

* with income_hh removed from the model
logit vote_2004 educ age if e(sample)
est store b

lrtest a b, stats

* with income_hh and age removed from the full model
logit vote_2004 educ if e(sample)
est store c

lrtest a c

lrtest b c

```

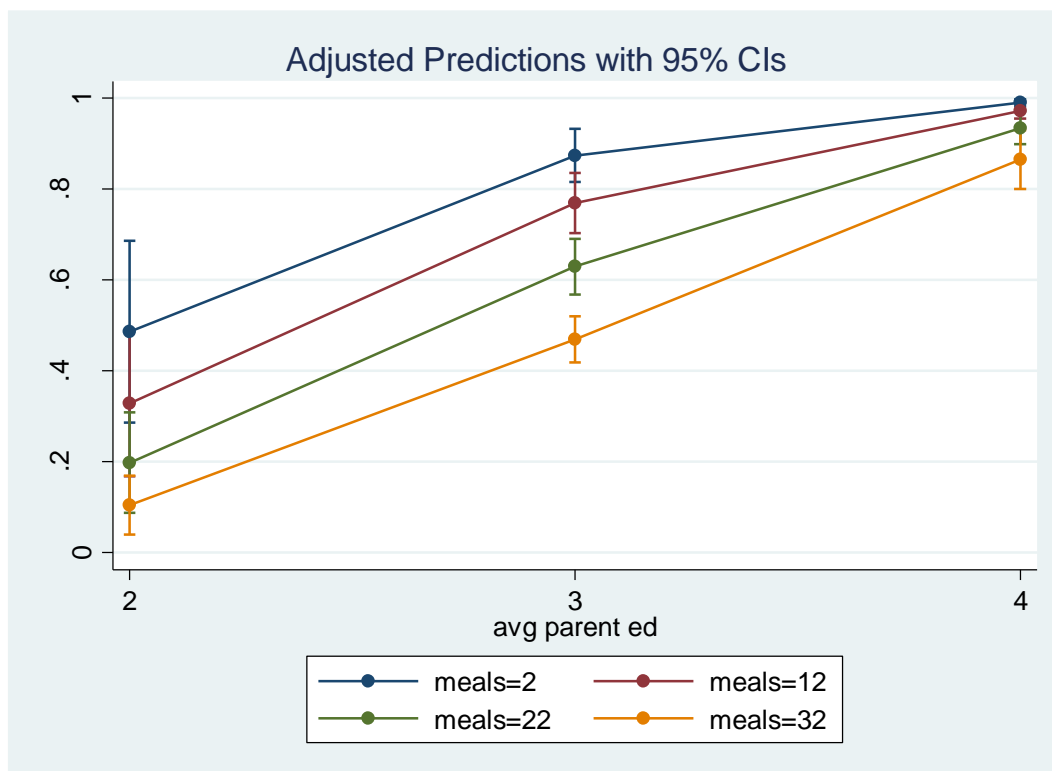
Now, since you have more than one IV, any effect of a unit change of one IV on the predicted probability of your DV will depend on the value of the other IVs (besides being a non-linear relationship between each IV and the DV as we already noted!). In other words, also the relationship between a given IV and DV changes accordingly to the value of the control IVs in a non-linear way.

Let's go back to our previous example with quality of schools (school.dta dataset). And let's run a probit regression with multiple regressors.

```

probit hiqual avg_ed enroll meals
margins, at(avg_ed=(2 3 4) (mean)_all)
margins, at(avg_ed=(2 3 4) meals=2 (mean)_all)
margins, at(avg_ed=(2 3 4) meals=12 (mean)_all)
margins, at(avg_ed=(2 3 4) meals=(2 12 22 32) (mean)_all)
marginsplot

```



As a result, a probit and/or a logit (that is, **non-linear models**) implicitly forces the effect of all the IVs to depend on each other. That is, even in an additive model (i.e. in a model without any interaction), the marginal effect of X depends on the value of the other IVs as well (so we also need to think about the value of the other IVs; moreover, the beta changed according to these values).

However, note that **this** dependence **occurs if the analyst's hypothesis is either conditional or not** – it is just part of deciding to use a non-linear model as logit: it is always **THERE!** If one wants to test a conditional hypothesis in a meaningful way, then the analyst has to include an explicit **interaction** term (or a **quadratic** one!) in the model.

Let's see an example with a **quadratic term**:

```
probit hiqual avg_ed enroll c.meals##c.meals
# Pr(Y=1 | X) at different values of meals from 0 to 100:
margins, at(meals=(0 (10) 100))
marginsplot

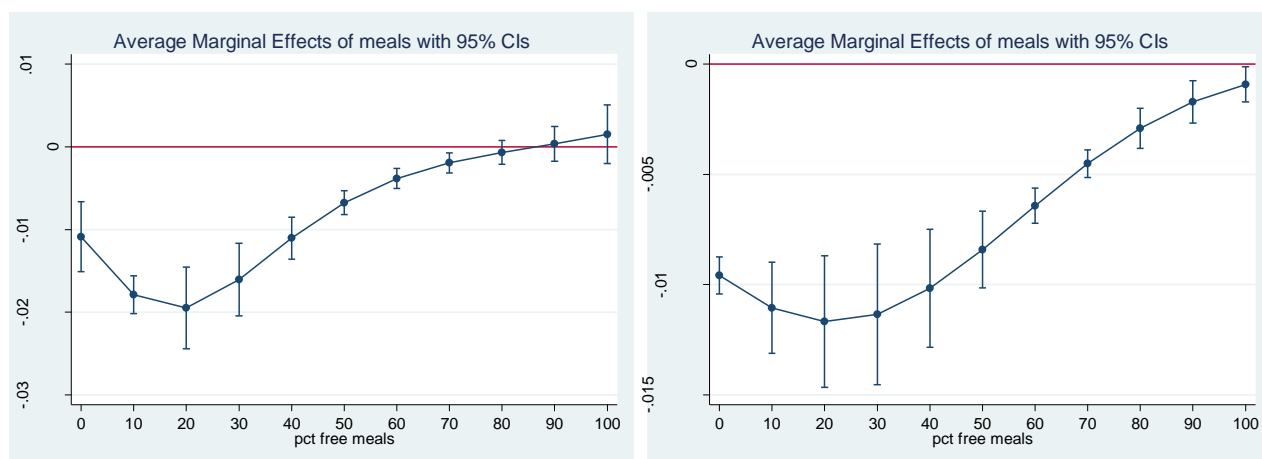
# marginal impact of increasing meals by 1 unit:
probit hiqual avg_ed enroll c.meals##c.meals
margins, dydx(meals) at(meals=(0 (10) 100))
marginsplot, yline(0)
```

***WITHOUT QUADRATIC TERM**


```

probit hiqual avg_ed enroll meals
margins, dydx(meals) at(meals=(0 (10) 100))
marginsplot, yline(0)

```



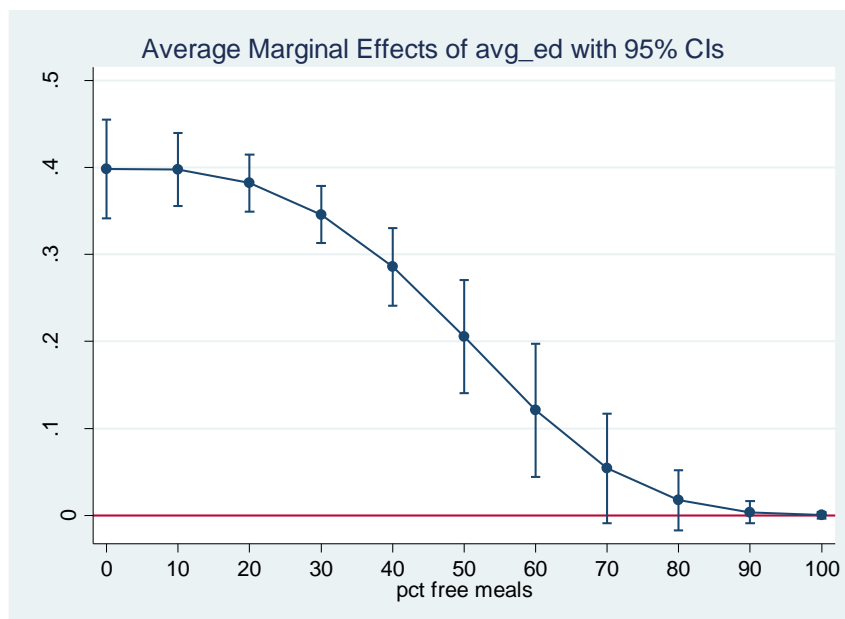
The left panel illustrates the marginal impact of meals when you have a quadratic term in the model. The right panel displays the marginal impact of meals when you DO NOT have a quadratic term in the model. The difference is substantial!

Let's see an example with an **interaction term**:

```

probit hiqual c.avg_ed##c.meals enroll
# marginal impact of increasing avg_ed by 1 unit:
margins, dydx(avg_ed) at(meals=(0 (10) 100))
marginsplot, yline(0)

```

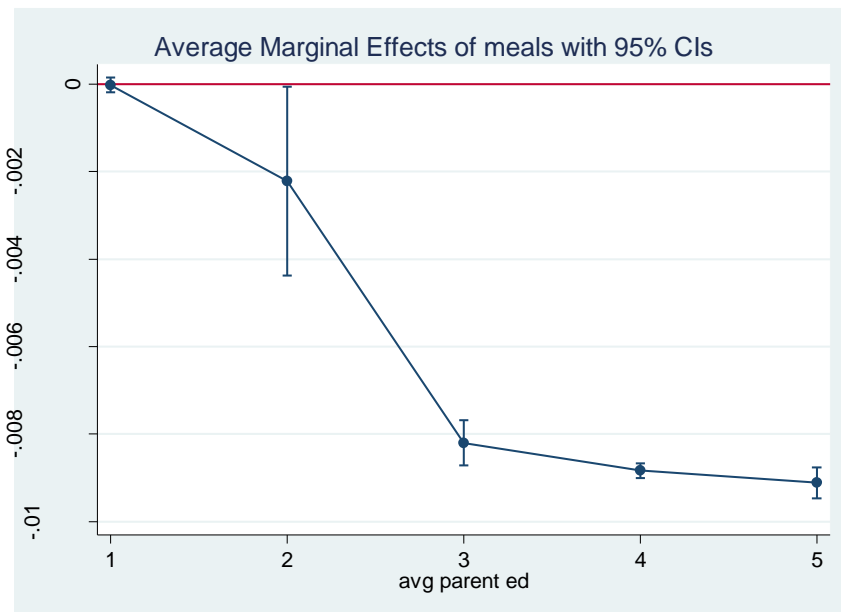


***WITHOUT INTERACTION TERM**

```
probit hiqual c.avg_ed meals enroll
# marginal impact of increasing avg_ed by 1 unit:
margins, dydx(avg_ed) at(meals=(0 (10) 100))
marginsplot, yline(0)
```

***THE REVERSE SIDE OF THE INTERACTION:**

```
probit hiqual c.avg_ed##c.meals enroll
# marginal impact of increasing meals by 1 unit:
margins, dydx(meals) at(avg_ed=(1 (1) 5))
marginsplot, yline(0)
```



Addendum: Be careful with margins when using a non-linear model!

Compare the results of these two margins:

```
probit hiqual avg_ed enroll meals
margins, at(avg_ed=(2 3 4) (mean)_all)
margins, at(avg_ed=(2 3 4) )
```

Why are the two sets of results different? In the second case, technically the margins command for avg_ed is estimated by considering the values of the two other IVs as they are found in the dataset!!! For example, for the first observation enroll = 638 and meals = 78, for the second observation enroll = 308 and meals = 49, and so on.

If we had an OLS this wouldn't matter at all! Why? No matter the value of enroll or meals, the impact of avg_ed is always constant!

```
reg hiqual avg_ed enroll meals
margins, at(avg_ed=(2 3 4) (mean)_all)
margins, at(avg_ed=(2 3 4))
```

But this could matter for non-linear models (it matters if we use a quadratic function or an interaction in an OLS; it matters always with a logit/probit!)

Addendum: Additional and useful command for logit models:

```
findit fitstat
logit vote_2004 educ age
```

```
fitstat
```

The `fitstat` command gives a list of various pseudo R^2 . As you can see from the output, some statistics indicate that the model fit is relatively good, while others indicate that it is not so good. The values are so different because they are measuring different things. We will not discuss the items in this output; rather, our point is to let you know that there is little agreement regarding an R^2 statistic in logistic regression, and that different approaches lead to very different conclusions. If you use an R^2 statistic at all, use it with great care.

You can use `fitstat` also to compare among different nested models:

```
# Comparing nested models with fitstat
```

```
logit vote_2004 educ age income_hh
```

```
fitstat, saving(m1)
```

```
logit vote_2004 educ age if e(sample)
```

```
fitstat, using(m1)
```

Other limited dependent variable models

The binary DV is an example of a DV with a limited range – that is, of a limited DV. Which are the other main models with a limited dependent variable?

1) **censored and truncated regression models**: when you have a ceiling or a floor in the distribution of your data by construction: for example, IQ test – you cannot score more than 200 even if you ideally could have more than 200 as your IQ test! i.e., not all people with an IQ of 200 are the same! But you cannot observe them! Observations are simply unavailable when the DV is above or below a certain cutoff.

2) **count data**: when the DV is a counting number.

3) **ordered responses**: when the DV is an ordinary variable (i.e., which is the degree you have → use ordered logit model).

4) **discrete choice data**: when the DV is a categorical variable (i.e., the mode of transport you select to go to university → use multinomial logit regression models).