Exercise 4: Models of Binary Dependent Variable – Solution

In this exercise the interest is in a binary dependent variable. We will first discuss the linear probability model. We then discuss its limitations which will lead us to the probit and logit models.

1. Open the data, label and examine the variables

Open the data file. We are interested in two variables. *deny* is a binary variable, which equals 1 if the mortgage application is denied and equals 0 if it is accepted. *piratio* is a continuous variable, which is the ratio of the applicant's anticipated total monthly loan payments to his or her monthly income. Label these variables as below. Examine the variables.

```
use "C:\exercisefour - data"
label variable deny "Mortgage Application Decision"
label variable piratio "Payment Income Ratio"
tabulate deny
histogram piratio
tabulate black
```

2. Compute summary statistics

The scatter plot of *deny* against *piratio* seems to indicate that applicants with a high ratio of debt payments to income are more likely to have their applications denied. In this exercise we will attempt to model this likelihood.

```
pwcorr deny piratio black, sig
scatter deny piratio
```

3. Linear probability model

We are interested in the simple model $E(y|x) = \beta_0 + \beta_1 x_1$, where the dependent variable takes a value of 1 or 0. In this case the conditional expectation becomes $E(y|x) = 1 * P(y = 1|x) + 0 * P(y = 0|x) = P(y = 1|x)$. Hence, we have $P(y = 1|x) = \beta_0 + \beta_1 x_1$. The dependent variable is called the response probability and it represents the probability of success, where success refers to value 1. β_1 measures the change in the probability of success with respect to a change in x_1. The OLS estimate of this model is consistent and unbiased.

Limitations of the model

The linear probability model is limited in three respects. First, we stated the conditional expectation function as $E(y|x) = P(y = 1|x) = \beta_0 + \beta_1 x_1$ but we did not specify a range of values that x can take. If x takes extreme values, the probability will be forced to lie outside its range $[0, 1]$, which makes no sense. Or, consider the unconditional expectation function $E(y) = P(y = 1) = \beta_0 + \beta_1 E(x_1)$. The probability is defined at the mean value of x. If x departs from its mean considerably then y will also depart from its mean forcing the probability to lie outside its defined range.

Second, when the dependent variable is a binary variable, its conditional variance can be shown equal to $Var(y|x) = P(y|x) * (1 - P(y|x))$, where $P(y|x) = \beta_0 + \beta_1 x_1$. That is, the conditional variance of y depends on the values of x. This is the definition of heteroskedasticity. Hence, by its construction the linear probability model is heteroskedastic. We should either correct the standard errors of the estimates in the estimated linear probability model, or carry out the weighted least squares estimation.

The third limitation is not mathematical but economical. The stated probability is linear in the independent variable. This is restrictive because it imposes a linear relation for all possible values of the independent variable. This is saying constant marginal effects. In many cases we expect the probability to have a nonlinear relation with its determinants.

Regression

We model the probability of denial as $P(deny = 1|piratio) = \beta_0 + \beta_1 piratio$. To estimate the model we run the command `regress deny piratio`. The estimated model is $-0.13 + 0.81 piratio$.

Partial effect of *piratio*

The coefficient of *piratio* indicates that a 10% increase in debt to income ratio increases the probability of denial by about 8%.

The model has two problems and we can visualize them. For the first problem, type `predict phat, xb` and then `summarize phat, detail` to observe those predictions outside the unit interval. For the second problem, visualize how the variance of the residuals progress with values of *piratio*. Obtain the residuals using the command `predict ehat, residuals` and then plot them against *piratio* using the command `twoway (scatter ehat piratio)`. The plot shows that the deviation of the residuals from mean 0 is rather changing at high values of *piratio*. This is evidence of heteroskedasticity.

Inference on the effect of *piratio*

Recall that the variance of a coefficient estimate depends on the variance of the errors, which are homoskedastic and hence have a constant variance. In the estimated model the variance of the coefficient estimates are not valid because the residuals are heteroskedastic by the definition of the linear probability model. Therefore we cannot use an invalid standard error in the t-statistic. But we wish to make inference. We can react in two ways.

First, we can calculate the heteroskedasticity robust standard errors. See endnote 1 for a comparison of the usual and the robust variance of a coefficient estimate. To do this in Stata add the option `robust` at the end of the regression command: `regress deny piratio, robust`. The standard errors slightly decrease when we correct them for heteroskedasticity. The t-statistic is 9.74 and the p-value is virtually 0. $\hat{\beta}_1$ is significant at 1%.

Second, we can proceed with the weighted least squares estimation. For this, we first need to obtain an estimate of the conditional variance function which defines the heteroskedasticity in the current model. As stated above, the function we need to estimate is $Var(y|x) = P(y|x) * (1 - P(y|x))$. But $P(y|x) = \beta_0 + \beta_1 x_1$ is just the linear probability model and the estimate of $P(y|x)$ is the prediction of this model. We need to avoid predictions outside the unit interval. Replace the guilty predictions as follows: `replace phat = 0.999 if phat >= 1` and as `replace phat = 0.001 if phat <= 0`. Now we construct $1/(\hat{P}(y|x) * (1 - \hat{P}(y|x)))$, where the denominator is the estimated variance function and the ratio itself is the *weight* that

will weigh the least squares. See endnote 2 for the idea of weighed the least squares. Create this ratio using the command `generate weight = 1/(phat * (1 - phat))`. Estimate the model using the constructed weights. The command is `regress deny piratio [aweight = weight]`, where `aweight` stands for *analytic weight* as Stata defines it.[3] The standard errors decrease compared to the OLS estimation. But also the coefficient estimates are different. $\hat{\beta}_1$ is significant at 1%.

Prediction

Graph the fitted line by typing `twoway (scatter deny piratio) (connected phat piratio, sort)`. For a debt to income ratio of 1, the probability that the mortgage application will be denied is predicted to be about 70%. To find the exact prediction and its standard deviation, type `lincom _b[_cons] + _b[piratio] * 1`. The predicted probability is about 68% and it is significant at 1%.

Model Fit

In models of binary dependent variable the value of R^2 is usually much lower than 1. To see the reason recall the formula of R^2. It is the ratio of model sum of squares, $\sum(\hat{y}_i - \bar{y})^2$, to total sum of squares $\sum(y_i - \bar{y})^2$. Now consider two different scatter plots. Produce the first one as we did above `twoway (scatter deny piratio) (connected phat piratio, sort)`. Imagine the second one as there was a cloud of observations around the regression line in this plot. In these two cases the $\bar{y}_i$, and consequently the distance between $\hat{y}_i$ and $\bar{y}_i$ would be about same, or at least not very different. However, the distance between y_i and $\bar{y}_i$ would be almost always larger, making R^2 smaller, in the first case than in the second case. An implication of this is that we could consider another functional form, to fit the line better to the scatter of binary observations. But it is hard to think of a functional form that can capture the observations lying along the two parallel lines. All these mean that R^2 does not seem to be a useful measure when the dependent variable is binary.

Therefore we rely on another measure, called *percent correctly predicted* or *count* $- R^2$. It is given by the ratio of *number of correct predictions/total number of observations*. But how do we decide that a prediction is correct? We need to set a threshold. The choice is arbitrary and you can set your own level of correctness. But an objective choice would be 0.5. All we need to do is check if a prediction is smaller or bigger than the threshold and compare it to the actual observation. If the prediction is bigger (smaller) than 0.5 and if the actual observation is 1 (0) we count it as *correctly predicted*. The translation of this to the Stata syntax is `count if (deny == 1 & phat > 0.5) | (deny == 0 & phat < 0.5)`. This command counts that the number of correct predictions is 2343. The total number of observations is 2708. `display 2343/2708` indicates that 86% of the time our model makes right predictions. The problem with this measure is that not only a predicted value of 0.99 but also a value of 0.51 is treated as a correct prediction. For this reason, we shall consider a more restrictive threshold, such as `count if (deny == 1 & phat > 0.8) | (deny == 0 & phat < 0.2)`. `display 2163/2708` indicates almost 80%. The model is still predictive.

```
regress deny piratio
predict phat, xb
summarize phat, detail
predict ehat, residuals
twoway (scatter ehat piratio)
```

```
regress deny piratio, robust
replace phat = 0.999 if phat >= 1
replace phat = 0.001 if phat <= 0
generate weight = 1/(phat * (1-phat))
regress deny piratio [aweight = weight]
twoway (scatter deny piratio) (connected phat piratio, sort)
lincom _b[_cons] + _b[piratio] * 1
count if (deny == 1 & phat > 0.5) | (deny == 0 & phat < 0.5)
display 2343/2708
count if (deny == 1 & phat > 0.8) | (deny == 0 & phat < 0.2)
display 2163/2708
```

4. Probit and logit models

Derivation of the probit and logit models depends on an underlying unobserved, or latent, model. The model is stated as $y^* = \beta_0 + \beta_1 x_1 + u$. We view this model as the propensity for an event to occur. As x_1 increases, the propensity that y^* occurs increases. But x_1 can increase indefinitely and hence the propensity. When will the event y^* occur? We need to set a threshold. Let that threshold be 0. Hence, if $\beta_0 + \beta_1 x_1 + u > 0$ we consider that the event occurs and indicate this occurrence as $y = 1$.[4] If the propensity is smaller than 0, we indicate it as $y = 0$. We observe y. But what determines y is an unobserved model. We specify y as $y = I(y^* \geq 0)$, where I is an indicator function that returns 1 if the propensity is larger than the threshold, and 0 otherwise. The interest is in the probability that the event occurs: $P(y = 1|x) = P(y^* > 0|x)$. Hence, $P(y = 1|x) = P(y^* > 0|x) = P(\beta_0 + \beta_1 x_1 + u > 0|x) = P(u > -(\beta_0 + \beta_1 x_1)|x)$. It is here that we make an assumption on how u is distributed. If it has a standard normal distribution we have the probit model, if it has a standard logistic distribution we have the logit model.[5] It follows that $P(u > -(\beta_0 + \beta_1 x_1)|x) = 1 - \Phi(-(\beta_0 + \beta_1 x_1)) = \Phi(\beta_0 + \beta_1 x_1)$, where Φ is the cumulative distribution function of the standard normal or the logistic distribution. The former is given by

$$\Phi(\beta_0 + \beta_1 x_1) = 1/\sqrt{2\pi} \int_{-\infty}^{\beta_0 + \beta_1 x_1} \exp(-z^2/2) dz,$$

and the latter by

$$\Lambda(\beta_0 + \beta_1 x_1) = exp(\beta_0 + \beta_1 x_1)/1 + exp(\beta_0 + \beta_1 x_1).$$

Because the models are nonlinear in the parameters, we use the maximum likelihood method for estimation of these parameters.

The partial effect of x_1 is $\delta P(y^* > 0|x_1)/\delta x_1 = \delta\Phi(\beta_0 + \beta_1 x_1)/\delta x_1 = \phi(\beta_0 + \beta_1 x_1)\beta_1$. The partial effect always has the same sign as β_1. Note that, due to nonlinearity, the partial effect is not only the coefficient but it is scaled with $\phi(\beta_0 + \beta_1 x_1)$. Hence, a probit or logit coefficient estimate does not have a useful interpretation. To calculate the partial effect we need to estimate this scale factor.

Merits of the model

We stated the linear model as $P(y = 1|x) = \beta_0 + \beta_1 x_1$ and the current nonlinear model as $P(y = 1|x) = \Phi(\beta_0 + \beta_1 x_1)$. How does the nonlinear model solves the problems of the linear model? The probability in the linear model can take values out of the probability range. The

probability in the nonlinear model is restricted to the probability range as the cumulative distribution function is defined on the unit interval $(0, 1)$. In the linear model marginal effects are constant. That is, the change in the linear function $P(y = 1|x) = \beta_0 + \beta_1 x_1$ is constant. In the nonlinear model marginal effects are changing. That is, the change in the cumulative distribution function $P(y = 1|x) = \Phi(\beta_0 + \beta_1 x_1)$ is not constant.

Regression

We model the probability of denial as $P(deny = 1|piratio) = \Phi(\beta_0 + \beta_1 piratio)$. To estimate the model we run the command `probit deny piratio`. Alternatively in the program menu follow the route Statistics > Binary outcomes > Probit Regression. In the regression output a series of `Iterations` appear. These are a record of the steps in maximizing the log-likelihood function. The reported log-likelihood values are negative as always they are. This is because in the log-likelihood function we have the logarithm of the cumulative distribution function which is always negative. Also, instead of the t value we see the z value in the output. Recall that the t distribution has a z (standard normal) distribution asymptotically (that is, as $n \to \infty$). We need the z distribution because the maximum likelihood estimation results are all asymptotic. The estimated model is $-2.18 + 3.18 piratio$.

Partial effect of *piratio*

Coefficient of *piratio* is positive which means *piratio* is positively related to the probability of denial. We cannot conclude more than this. For further interpretation we need to calculate the marginal effect of *piratio*.

The change in the probability of denial for a small change in *piratio* can be calculated as $\phi(-2.18 + 3.18 * piratio) * 3.18$. This is the tangent slope and it is an approximate change. The change depends on the value of *piratio*. Suppose our interest is at a starting level of debt income ratio of 0.4. Hence, we evaluate the standard normal density function at $-2.18 + 3.18 * 0.4$. We will then multiply this figure by 3.18. The sequence of commands are the following. `scalar thisisit = _b[_cons] + _b[piratio] * 0.4` computes the stated scalar, and `display "Marginal effect of piratio = " normalden(thisisit) * _b[piratio]` evaluates the scalar at the density, multiplies it with 3.18 and displays the result. The marginal effect of *piratio* is 0.83. This means that if *piratio* increases by 0.1 unit, probability of denial will increase by 0.083, or 8.3%. To understand what we are doing we calculated step by step the marginal effect given its mathematical expression. We can instead ask Stata to directly provide us the marginal effect. The command for this is `mfx, at (piratio = 0.4)`. This command also produces the predicted probability `0.18023103`, when $piratio = 0.4$ as stated in the last column in the output. But we will exercise the calculation of prediction below.

We may wish to compute the exact change for increasing *piratio* from 0.4 to 0.5. This is the secant slope and it can be calculated as $\Phi(-2.18+3.18*0.5)-\Phi(-2.18+3.18*0.4)$. This is equal to $\Phi(-0.59) - \Phi(-0.91) = 0.095$, or 9.5%. The sequence of Stata commands that give this figure is the following. `scalar one = _b[_cons] + _b[piratio] * 0.5` and `scalar two = _b[_cons] + _b[piratio] * 0.4` compute the scalars that we evaluate in the cumulative distribution function. `display "Difference in probabilities = " normal(one)-normal(two)` computes the difference between the two probability amounts and displays the result.

Inference on the effect of *piratio*

To make inference on the marginal effect, we need a standard error. Since the effect is non-linear in the estimates, we will use the *nlcom* command. It is `nlcom normalden(_b[_cons] + _b[piratio] * 0.4) * _b[piratio]`. The marginal effect appears significant at 1%.

We can test the overall significance of the model using the (LR) (likelihood ratio) test. The statistic is $LR = 2 * (L_{ur} - L_r)$, where L_{ur} is the log-likelihood value for the unrestricted model and L_r is for the restricted model. We compare the log likelihood values from the unrestricted model to the restricted model. Dropping variables in the log-likelihood function decreases the log-likelihood value. The test measures whether the decrease is large enough to conclude that the dropped variables are important. $L_{ur} = -1024.5$ from the output. We find that $L_r = -1083.6$ if we run `probit deny`. This gives $LR = 2 * (-1024.5 + 1085.6) = 122.14$. This is same as the `LR chi2(1)` in the Stata output. $2 * (L_{ur} - L_r)$ has an approximate chisquare distribution with degrees of freedom equal to the number of restrictions. The critical value is find by typing `scalar chi = invchi2tail(1,0.05)` and `display chi`, where `invchi2tail(1,0.05)` returns the inverse of the reverse cumulative (upper-tail) chi-squared distribution. Since 122.14 is larger than 3.84 we reject the null hypothesis at 5% level that the slope coefficient is 0.

Prediction

Rerun the regression `probit deny piratio` because Stata has in memory the last regression `probit deny` we run above. Graph the fitted curve by obtaining the predictions using the command `predict pphat` and then the command `twoway (scatter deny piratio) (connected pphat piratio, sort)`. Unlike the linear probability model, the probabilities lie between 0 and 1. Also, the marginal effects are not constant anymore. For a debt to income ratio of 1, the probability that the mortgage application will be denied is predicted to be about 90%. The exact prediction is given by $\Phi(-2.18 + 3.18 * 1) = \Phi(1)$, which represents the area under the standard normal density up to value 1. We can calculate it by typing `display normal(1)`. This gives 84.1%. We can use the `nlcom` command to get the standard deviation of this predicted probability. Type `nlcom normal(_b[_cons] + _b[piratio] * 1)`. The predicted probability is significant at 1%.

Model Fit

We discussed the reason that R^2 is low in models of binary dependent variable. That the function here is not linear but S shaped does not change the reasoning we made. Convince yourself that it does not. As before, instead of the R^2 we can use here the $count - R^2$ as a measure of model fit. Since the idea is same we will not repeat it here. Yet another measure is the $pseudo - R^2$ presented in the estimation output. The value is very small perhaps because we are considering only a single covariate.

```
probit deny piratio
scalar thisisit = _b[_cons] + _b[piratio] * 0.4
display "Marginal effect of piratio = " normalden(thisisit) * _b[piratio]
mfx, at (piratio = 0.4)
scalar one = _b[_cons] + _b[piratio] * 0.5
scalar two = _b[_cons] + _b[piratio] * 0.4
display "Difference in probabilities = " normal(one)-normal(two)
nlcom normalden(_b[_cons] + _b[piratio] * 0.4) * _b[piratio]
```

```
probit deny
scalar chi = invchi2tail(1,0.05)
display chi
probit deny piratio
predict pphat
twoway (scatter deny piratio) (connected pphat piratio, sort)
display normal(1)
nlcom normal(_b[_cons] + _b[piratio] * 1)
```

5. Programming probit

In your do-file is a sample Stata program of maximum likelihood estimation. This short program demonstrates that it is very feasible to program and estimate your own likelihood function in Stata. You will note that the results we get from this program are the same as the results we obtained above.

†Endnotes

1. In a two variable regression model where the errors are *homoskedastic*, the variance of the least square estimate is given by

$$Var(\hat{\beta}_1) = \frac{\sum(x_i - \bar{x})^2\sigma^2}{(\sum(x_i - \bar{x})^2)^2}$$

In a two variable model where the errors are *heteroskedastic*, the robust variance of the least square estimate is given by

$$Var(\hat{\beta}_1) = \frac{\sum(x_i - \bar{x})^2\sigma_i^2}{(\sum(x_i - \bar{x})^2)^2}$$

Note the subscript of σ^2 in the latter expression. For each x_i there is an associated σ_i^2. This takes into account the correlation between the errors and the independent variable. The σ^2 in the former expression does not depend on any x_i and hence can factor out of the sum. White's heteroskedasticity test (1980) relies on the stated two variances (Murray, Econometrics, 2006, p. 395). 2. Start with the original model $y = \beta_0 + \beta_1 x_1 + u$. Divide the equation through by the square root of the conditional variance function which is *known* as

$$Var(u|x) = P(y|x) * (1 - P(y|x))$$

For ease of exposition let $h = Var(u|x)$. This will result in the following error $u/\sqrt{h}$. Consider the conditional variance of this transformed error. $Var(u/\sqrt{h}|x) = 1/h * Var(u|x)$. Plugging $Var(u|x)$ gives 1 which is a constant. We find that

$$Var(u/\sqrt{P(y|x) * (1 - P(y|x))}|x) = 1$$

The new error is homoskedastic because it has a constant variance. Note that in dealing with heteroskedasticity, the *robust variance* approach has no influence on the regression equation. The *weighted least squares* has a direct influence. 3. Actually, the denominator should have been square rooted. In Stata the analytic weight is required to be inversely proportional to the variance. Stata will take care of the square rooting itself. Type `help weight` for the relevant explanation of Stata. 4. The threshold could be nonzero. But as long as there is a constant

term in the latent model, any nonzero threshold can be made zero. 5. Logistic distribution has slightly fatter tails than the standard normal distribution. There is no sound reason to choose one over the other.

§Humor

> "... the word econometrics should not be confused with economystics or economic-tricks ..." (Kennedy, A Guide to Econometrics, 2008, p. 6)

† Please send your questions, comments or possible corrections to kantarci@uvt.nl.