Logistic Regression Analysis Using STATA[*]

22nd August 2004

NOTE: Some of the commands used in this quick guide are not part of STATA. It is required to install the package SPOST, which can be downloaded for free from the Internet. See STATA help “spost”.

1 Running a Logistic Regression with STATA

1.1 Estimation of the model

To ask STATA to run a logistic regression use the logit or logistic command. The differences between those two commands relates to the output they generate. While logit presents by default the coefficients of the independent variables measured in logged odds, logistic presents the coefficients in odds ratios. The odds coefficients can be obtained also with the logit command by using the option or after the command.

```
. logit depvar indep_var1 indep_var2 indep_var3
. logit depvar indep_var1 indep_var2 indep_var3, or
. logistic depvar indep_var1 indep_var2 indep_var3
```

1.2 Residuals and predicted values

To get the predicted values of the dependent variable according to the latest model estimated, we can use the command predict after an estimation. This command allows us to create a new variable that will store either the predicted values or the residuals:

```
. predict new_predicted_values
. predict new_residual_values, resid
. predict standardized_residuals, rstand
* useful for residual analysis
. predict dx2, dx2
* useful for influential cases analysis
```

2 Coefficients

Appart from having the results of the coefficients in terms of logged odds (unstandardized coefficients) and odds, we can also explore other ways to interpret the strength of the coefficients, mainly the semi-standardized coefficients (column $bStdX$) and the fully standardized coefficients (column $bStdXY$). The command listcoef presents semi-standardized coefficients for the odds

To contact the authors email xavier.fernandez@upf.edu.

ratios. When the option `std` is invoked, it presents the semi-standardized and fully standardized coefficients for the logged odds.

```
. listcoef
. listcoef, std
```

3 Dealing with probabilities

When dealing with probabilities, the most useful command is `prchange`.

```
. prchange
```

The first column shows the change in the probabilities when the independent variable varies from its minimum to its maximum. The second shows the change when the independent variable varies from 0 to 1. This is the most useful when analysing dummy variables. The third and fourth columns show the change in probabilities when the independent variable varies one unit in real value or in standard deviations, respectively. The last column presents the marginal changes of the independent variable. All this values are calculated at the predicted probability when the independent variables take their mean values, which are listed just below this table.

3.1 Marginal effects

Other way to get the marginal effects and the strength of the coefficients is to use the `mfx compute` command.

```
. mfx compute
```

The table presents the marginal effects on the probabilities of each of the continuous variables of the model. Those marginal effects in the probabilities are calculated when the changes in the probability are greater: when the dependent variable is at its mean value.

3.2 Generate predicted probabilities

The command `prgen` allows us to create a set of values derived from our coefficients.

```
. prgen indep_continuous, x(indep_dummy=0) from(0) to(20) ncases(21) gen (newvar1)
. prgen indep_continuous, x(indep_dummy=1) from(0) to(20) ncases(21) gen (newvar2)
```

Moreover, we can plot the two newly generated variables to have a graphical idea of their effects.

```
. graph7 newvar1p1 newvar2p1 newvar2dx
* when using STATA 7.0 or less:
. graph newvar1p1 newvar2p1 newvar2dx
```

3.3 Advanced plotting of the effects of the variables

The `praccum` command is a very powerful tool that in combination with other commands allows us to plot probabilities from models with interaction terms. The example given below will consider that the interaction term is built with a dichotomous variable (with values 0 and 1) and a (quasi)interval variable.

We will graph the probabilities for the different values of the interval variable and for each category of the dichotomous variable, holding the rest of the independent variables fixed at their mean values[1].

```
.  capture matrix drop whatever_name1
.  forvalues count=lowest(interval)highest {
prvalue, x(dichotomous_var=0 interval_var=‘count’ interaction_var=0) rest(mean)
praccum, using(whatever_name1) xis(‘count’)
}

.  pracum, using(whatever_name1) gen(firstprob)
.  capture matrix drop whatever_name2
.  forvalues count= lowest(interval)highest
prvalue, x(dichotomous_var =1 interval_var =‘count’ interaction_var =‘count’) rest(mean)
praccum, using(whatever_name2) xis(‘count’)


.  praccum, using(whatever_name2) gen(secondprob)
.  label var firstprobp1 "Label_of_first_category_of_dichotomous_variable"
.  label var secondprobp1 "Label_of_second_category_of_dichotomous_variable"

.  graph7 firstprobp1 secondprobp1 firstprobx, xlabel(lowest,highest) ylabel(0 .2 to 1) c(ss)


* lowest: the lowest category of the interval variable
* highest: the highest category of the interval variable
* interval: defines the interval used to go from the lowest to the highest
*  values of the interval variable. If you don’t know what is it about and
*  feel completely lost, just try typing value 1.
```

4 Model fit statistics for nested models

To determine if a set of dummy or continuous variables is statistically significant tatistically, we can use the command `chi2tail` to make a simple test of hypothesis.

```
. display chi2tail(df2-df1, var1-var2)
* when df2 is the degrees of freedom of the second model
*      df1 are the degrees of freedom of the nested model
*      var1 is the deviance of the nested model
*      var2 is the deviance of the second model
```

5 Goodness of fit

STATA presents different statistics to get an idea of the goodness of fit of our model with the command `fitstat`.

```
. fitstat
```

5.1 Classification tables

The command `lstat` presents a table with the correctly and incorrectly predicted results of the model. It has strong limitations when the dependent variable is not centered arount a .5

[1] Remember that this example only holds for models that include interaction terms

distribution of both 0 and 1.

```
. lstat
```

6 Residual analysis

6.1 Outliers

To analyse the outliers of the model we must save the standardized residuals using the `predict` command, as we have seen above.

One way to detect some patterns in the worst predicted cases is to list the independent variables for those cases that have at least 2.0 in the standardized residual. The second way is to tabulate the independent variables but asking for the same condition

```
. list indep1 indep2 indep3 predict zres if (zres>=2 & e(sample)), nolabel
* with the 'e(sample)' option we avoid to get the missing values
. tab indep1 if (zres>=2 & e(sample))
```

6.2 Influential cases

The option `dx2` of the `predict` saves the needed values to get an idea of the sign and pattern of the influential cases. As with the outliers, we can both inspect the list of values of tabulate them.

```
. list indep1 indep2 indep3 predict dx2 if (dx2>=7 & e(sample)), nolabel
. tab indep1 if (dx2>=7 & e(sample))
```