Linear Probability Model

Note on required packages:
The following code requires the packages sandwich and lmtest to estimate regression error variance that may change with the explanatory variables. If you have not done so on your machine, download the package lmtest. This should only need to be done once on your computer.

```r
install.packages("lmtest")
```

You can call the libraries sandwich and lmtest with the following calls to library:

```r
library("sandwich")
library("lmtest")
```

```r
## Loading required package: zoo
##
## Attaching package: 'zoo'
##
## The following objects are masked from 'package:base':
##
## as.Date, as.Date.numeric
```

Recall from a previous tutorial that binary variables can be used to estimate proportions or probabilities that an event will occur. If a binary variable is equal to 1 for when the event occurs, and 0 otherwise, estimates for the mean can be interpreted as the probability that the event occurs.

A linear probability model (LPM) is a regression model where the outcome variable is a binary variable, and one or more explanatory variables are used to predict the outcome. Explanatory variables can themselves be binary, or be continuous.

1. Example: Mortgage loan applications

The dataset, loanapp.RData, includes actual data from 1,989 mortgage loan applications, including whether or not a loan was approved, and a number of possible explanatory variables including variables related to the applicant’s ability to pay the loan such as the applicant’s income and employment information, value of the mortgaged property, and credit history. Also included in the dataset are variables measuring the applicant’s race and ethnicity.

1.1 Estimating the linear probability model

The code below loads the R dataset, which creates a dataset called data, and a list of descriptions for the variables called desc.
Let us estimate a simple linear regression model with loan approval status as a binary outcome variable (`approve`) and total housing expenditure relative to total income as the sole explanatory variable (`hrat`).

```r
lmapp <- lm(approve ~ hrat, data=data)
summary(lmapp)
```

```
##
## Call:
## lm(formula = approve ~ hrat, data = data)
##
## Residuals:
##    Min     1Q   Median     3Q    Max
## -0.9489  0.1013  0.1202  0.1328  0.2710
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.95520   0.02660  35.900  < 2e-16 ***
## hrat        -0.00314   0.00103  -3.045  0.00236 **
## ---
## Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
##
## Residual standard error: 0.3275 on 1987 degrees of freedom
## Multiple R-squared:  0.004645,   Adjusted R-squared:  0.004144
## F-statistic: 9.273 on 1 and 1987 DF,  p-value: 0.002356
```

### 1.2 Visualizing the linear probability model

Let us visualize the actual and predicted outcomes with a plot. The call to `plot()` below produces a scatter plot of the housing expenses ratio relative (`hrat`) to the approval variable (`approve` equal to 0 or 1). The call to `abline()` plots the linear regression equation.

```r
plot(y=data$approve, x=data<hrat)
abline(lmapp)
```
It is a strange looking scatter plot because all the values for approve are either at the top (=1) or at the bottom (=0). The best fitting regression line does not visually appear to describe the behavior of the values, but it still is chosen to minimize the average squared vertical distance between all the observations and the predicted value on the line.

1.3 Predicting marginal effects

Since the average of the binary outcome variable is equal to a probability, the predicted value from the regression is a prediction for the probability that someone is approved for a loan.

Since the regression line is sloping downwards, we see that as an applicant’s housing expenses increase relative to his/her income, the probability that he/she is approved for a loan decreases.

The coefficient on $hrat$ is the estimated marginal effect of $hrat$ on the probability that the outcome variable is equal to 1. Our model predicts that for every 1 percentage point increase in housing expenses relative to income, the probability that the applicant is approved for a mortgage loan decreases by 0.31%.

2. Heteroskedasticity

All linear probability models have heteroskedasticity. Because all of the actual values for $y_i$ are either equal to 0 or 1, but the predicted values are probabilities anywhere between 0 and 1, the size of the residuals grow or shrink as the predicted values grow or shrink.

Let us plot the predicted values against the squared residuals to see this:

```r
plot(y=lmapp$residuals^2, x=lmapp$fitted.values, ylab="Squared Residuals", xlab="Predicted probabilities")
```
In order to conduct hypothesis tests and confidence intervals for the marginal effects an explanatory variable has on the outcome variable, we must first correct for heteroskedasticity. We can use the White estimator for correcting heteroskedasticity.

We compute the White heteroskedastic variance/covariance matrix for the coefficients with the call to `vcovHC` (which stands for Variance / Covariance Heteroskedastic Consistent):

```r
vv <- vcovHC(lmapp, type="HC1")
```

The first parameter in the call above is our original output from our call to `lm()` above, and the second parameter `type="HC1"` tells the function to use the White correction.

Then we call `coeftest()` to use this estimate for the variance / covariance to properly compute our standard errors, t-statistics, and p-values for the coefficients.

```r
coefftest(lmapp, vcov = vv)
```

```
## t test of coefficients:
##
## Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.9552042  0.0311474 30.6672  <2e-16 ***
## hrat        -0.0031414  0.0012522  -2.5086  0.0122 *
## ---
## Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```
The coefficient is in fact negative (-0.003) and the p-value is equal to $0.012 / 2 = 0.006$ (one-sided test). This is less than 0.05, so we reject the null hypothesis. We found sufficient statistical evidence that having higher housing expenses relative to income leads to lower probability for being accepted for a mortgage loan.

3. Problems using the Linear Probability Model

There are some problems using a binary dependent variable in a regression.

1. There is heteroskedasticity. But that’s ok, we know how to correct for it.

2. A linear model for a probability will eventually be wrong for probabilities which are by definition bounded between 0 and 1. Linear equations (i.e. straight lines) have no bounds. They continue eventually upward to positive infinity in one direction, and negative infinity in the other direction. It is possible for the linear probability model to predict probabilities greater than 1 and less than 0.

   Use caution when the predicted values are near 0 and 1. It is useful to examine the predicted values from your regression to see if any are near these boundaries. In the example above, all the predicted values are between 0.7 and 0.95, so fortunately our regression equation is not making any mathematically impossible predictions.

   Also, be cautious when using the regression equation to make predictions outside of the sample. The predicted values in your regression may have all fallen between 0 and 1, but maybe a predicted value will move outside the range.

3. The error term is not normal. When it is, then with small or large sample sizes, the sampling distribution of your coefficient estimates and predicted values are also normal.

   While the residuals and the error term are never normal, with a large enough sample size, the central limit theorem does deliver normal distributions for the coefficient estimates and the predicted values. This problem that the error term is not normal, is really only a problem with small samples.

3.1. Alternatives to the linear probability model

There are fancier methods out there for estimating binary dependent regression models that force the predicted values between 0 and 1, imposing some curvature on the regression model instead of a straight line.

One such model, called the logistic regression model imposes the logistic function. A logistic function has the form:

$$f(x) = \frac{e^x}{1 + e^x}$$

We can plot this function with the following call to curve():

```r
curve((exp(x) / (1 + exp(x))),
     from=-5, to=5,
     ylab="f(x)", main="Logistic Function")
```
To get an idea for how well a straight line can approximate the logistic function, we add to the plot an equation for a straight line with slope equal to 0.2 and vertical intercept equal to 0.5:

```r
curve((exp(x) / (1 + exp(x))),
     from=-5, to=5,
     ylab="f(x)", main="Logistic vs Linear Function")
abline(a=0.5, b=0.2)
```
Over a large range of values for $x$, including the values that are likely near the mean and median, the slope and predicted values of the linear equation approximate well the more complicated logistic function.

3.2. Which model should you use? Logistic regression or linear probability model?

1. Logistic function is mathematically correct. The linear probability model is only an approximation of the truth.
2. I’m not going to teach you the logistic regression model :)
3. Estimate both, compare your predictions. Often enough they are very similar.
4. There is a lot more you can do with the linear probability model.
5. A lot of things, like marginal effects, are much easier with the linear probability model.