Alex Miller

Problem Set 4

1.

(a)

For every given value of x\_i in Excel, the probability that it was drawn from a normal distribution with a mean of μ was calculated with the function:

=(1/(SQRT(2\*PI()\*1)))\*EXP(-((A2-$F$2)^2)/(2\*1))

Where A2 is the value of x\_i and F2 is μ. Similarly, the probability that x\_i was drawn from a Poisson distribution with a mean of μ was calculated with the function:

=(($F$2^A2)\*EXP(-1\*$F$2))/FACT(A2)

Where A2 and F2 are x\_i and λ respectively. The natural log of each of these probabilities were taken for both the normal and Poisson distributions, and then added together to find the log likelihood. For all possible means, given μ or λ >0, both the natural and Poisson distributions returned probabilities bounded between 0 and 1. Since both normal and Poisson distributions have cumulative density functions that approach 1, all individual values in the PDF must be less than 1. The normal distribution also returned probabilities bounded between 0 and 1 for values of μ<0 or μ=0, but the Poisson distribution could not handle these values.

(b)

By plugging in different values for μ and λ (as well as looking at the true mean for the sample), the maximum likelihood estimated mean for both distributions was the true mean: 14.26. For a sigma squared of 1 for the normal distribution, the maximum likelihood function value was -402.81, while the Poisson distribution had a maximum likelihood function value of -122.14.

(c)

Using the Poisson distribution, the restricted log-likelihood value for λ = 1 was -1192.04; for an unrestricted test, we set λ = 14.26 and get a log-likelihood value of -122.14. From here, the likelihood-ratio test statistic is equal to -2 \* (-1192.04 - (-122.14)) = 2139.8. Since the test statistic is distributed as a Chi-squared with a degree of freedom for every restriction, we can infer that the p-value associated with this test is incredibly close to zero. We would therefore reject the null hypothesis that λ = 1.

(d)

As long as our mean is greater than zero, the Poisson distribution is the superior model. The maximum likelihood function value is much higher than that of the normal distribution, so with the same number of parameters, the Poisson distribution would have a lower AIC. If we need to estimate a mean that can take on zero or negative values, however, the Poisson distribution would not function.

2.

(a)

Since people tend to make economic decisions based in part upon the effect prices will have on their income rather then the prices themselves, both the water bill and the price of the program were divided by the household income to calculate the percentage costs. Since the histograms of these percentage variables were all positive values with a large amount of probability stacked near zero (log-normal distribution), the natural log of these variables were used in the following equations so they would be normally distributed. All of the following models had the binary variable “ypay,” indicating that the respondent would pay as the dependent variable. The probability of the respondent answering yes was dependent on if they were a member of an environmental group, a large city dummy, their water bill as a percent of income, and the price the survey asked them to pay as a percent of their income. The models used to estimate this relationship were the linear probability model (OLS), probit, and logit. The commands used to execute the estimates are as follows:

. reg ypay environ urban logwaterbilpercent logbidpercent

. mfx compute

. estimates store ols

. probit ypay environ urban logwaterbilpercent logbidpercent

. mfx compute

. estimates store probit

. fitstat

. logit ypay environ urban logwaterbilpercent logbidpercent

. mfx compute

. estimates store logit

. fitstat

. esttab ols probit logit using parameters.csv, replace se stats(r2 N)

. esttab ols probit logit using marginal.csv, replace margin se stats(r2 N)

This table shows the formatted results from this estimation:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Parameter Estimates** | | | **Marginal Estimates** | | |
| *DV: ypay* | **OLS** | **Probit** | **Logit** | **OLS** | **Probit** | **Logit** |
|  |  |  |  |  |  |  |
| environ | 0.342\*\* | 1.510\*\* | 2.773\*\* | 0.342\*\* | 0.356\*\*\* | 0.347\*\*\* |
|  | (0.101) | (0.475) | (0.913) | (0.101) | (0.0781) | (0.0807) |
|  |  |  |  |  |  |  |
| urban | 0.226\* | 0.904\* | 1.618\* | 0.226\* | 0.331\* | 0.356\* |
|  | (0.0892) | (0.373) | (0.692) | (0.0892) | (0.136) | (0.154) |
|  |  |  |  |  |  |  |
| logwaterbilpercent | 0.138\* | 0.510\* | 0.859 | 0.138\* | 0.174\* | 0.169 |
|  | (0.0540) | (0.246) | (0.444) | (0.0540) | (0.0843) | (0.0875) |
|  |  |  |  |  |  |  |
| logbidpercent | -0.22\*\*\* | -0.91\*\*\* | -1.62\*\*\* | -0.22\*\*\* | -0.31\*\*\* | -0.32\*\*\* |
|  | (0.0309) | (0.173) | (0.339) | (0.0309) | (0.0560) | (0.0625) |
|  |  |  |  |  |  |  |
| constant | -0.554 | -4.645\* | -8.587\* |  |  |  |
|  | (0.384) | (1.849) | (3.527) |  |  |  |
|  |  |  |  |  |  |  |
| (Pseudo) R2 | 0.435 | 0.417 | 0.420 | 0.435 | 0.417 | 0.420 |
| Note: SE in parentheses. For marginal effects, the dummy variables represent a discrete change from 0 to 1. Significance is denoted by: \* p<0.05 \*\* p<0.01 \*\*\* p<0.001 | | | | | | |

Generally speaking, all three models return parameter estimates with similar signs, magnitudes, and significances. Being in an environmental group, living in a city, and paying a larger water bill all have positive influence on the probability of saying yes, while asking respondents to pay more for the program have a negative influence on the probability of saying yes. Since our dependent variable is binary, we can infer that OLS is not the best model to use. It does not necessarily return biased coefficients, but the standard errors are incorrect. Also, while it seems like the parameter estimates for logit and probit might be suggesting stronger correlations, one cannot just compare OLS parameters to logit or probit parameters. You can inference test logit/probit parameters, but making intuitive explanations is easier using the marginal effects at the means. For OLS, the marginal effects are not any different from the parameters because OLS is linear by definition; logit and probit exhibit differences, however. That being said, the observed marginal effects of our x’s on the probability of saying yes at the mean are much closer to the OLS estimated parameters.

(b)

The values for xiβ and the values for p(xiβ) for a logistic distribution, and whether that probability indicates a “yes” were calculated with the following commands after the logit estimation:

. gen xb = \_b[environ]\*environ + \_b[urban]\*urban + \_b[logwaterbilpercent]\*logwaterbilpercent + \_b[logbidpercent]\*logbidpercent + \_b[\_cons]\*\_cons

. gen probyes = (exp(xb))/(1+exp(xb))

. gen ypayhat = probyes>=0.5

The following table displays these values for the first ten observations:

**+---------------------------------------+**

**| ypay xb probyes ypayhat |**

**|---------------------------------------|**

**1. | 1 2.903856 .9480367 1 |**

**2. | 1 2.075157 .888465 1 |**

**3. | 0 2.558921 .9281706 1 |**

**4. | 1 1.730221 .8494406 1 |**

**5. | 1 4.813597 .9919468 1 |**

**|---------------------------------------|**

**6. | 1 2.311855 .9098541 1 |**

**7. | 1 4.322298 .9869044 1 |**

**8. | 1 5.103207 .9939595 1 |**

**9. | 1 6.334551 .9982292 1 |**

**10. | 0 -2.289828 .0919689 0 |**

**+---------------------------------------+**

It did pretty well! For 10 observations, it correctly guessed the outcome in 9 cases. Following this, we can replicate the results of the lstat command with the following commands:

. gen lstat\_sensitivity = ypay\*(probyes>=0.5)

. replace lstat\_sensitivity = . if ypay==0

. gen lstat\_specificity = (1-ypay)\*(probyes<0.5)

. replace lstat\_specificity = . if ypay==1

. gen lstat\_pospredictval = ypay\*(probyes>=0.5)

. replace lstat\_pospredictval = . if probyes<0.5

. gen lstat\_negpredictval = (1-ypay)\*(probyes<0.5)

. replace lstat\_negpredictval = . if probyes>=0.5

. gen lstat\_falseposfortrueneg = (1-ypay)\*(probyes>=0.5)

. replace lstat\_falseposfortrueneg = . if ypay==1

. gen lstat\_falsenegfortruepos = ypay\*(probyes<0.5)

. replace lstat\_falsenegfortruepos = . if ypay==0

. gen lstat\_falseposforclasspos = (1-ypay)\*(probyes>=0.5)

. replace lstat\_falseposforclasspos = . if probyes<0.5

. gen lstat\_falsenegforclassneg = ypay\*(probyes<0.5)

. replace lstat\_falsenegforclassneg = . if probyes>=0.5

. gen lstat\_correctclass = ypay\*(probyes>=0.5)+(1-ypay)\*(probyes<0.5)

\*ssc install fsum

\*for formatted summary stats

. fsum lstat\*, stats(mean) format(%9.4f)

Which returns the resulting table:

**Variable | N Mean**

**---------------------------+------------------**

**lstat\_sensitivity | 59 0.8644**

**lstat\_specificity | 36 0.7500**

**lstat\_pospredictval | 60 0.8500**

**lstat\_negpredictval | 35 0.7714**

**lstat\_falseposfortrueneg | 36 0.2500**

**lstat\_falsenegfortruepos | 59 0.1356**

**lstat\_falseposforclasspos | 60 0.1500**

**lstat\_falsenegforclassneg | 35 0.2286**

**lstat\_correctclass | 95 0.8211**

Compared to the “lstat” results:

**Logistic model for ypay**

**-------- True --------**

**Classified | D ~D | Total**

**-----------+--------------------------+-----------**

**+ | 51 9 | 60**

**- | 8 27 | 35**

**-----------+--------------------------+-----------**

**Total | 59 36 | 95**

**Classified + if predicted Pr(D) >= .5**

**True D defined as ypay != 0**

**--------------------------------------------------**

**Sensitivity Pr( +| D) 86.44%**

**Specificity Pr( -|~D) 75.00%**

**Positive predictive value Pr( D| +) 85.00%**

**Negative predictive value Pr(~D| -) 77.14%**

**--------------------------------------------------**

**False + rate for true ~D Pr( +|~D) 25.00%**

**False - rate for true D Pr( -| D) 13.56%**

**False + rate for classified + Pr(~D| +) 15.00%**

**False - rate for classified - Pr( D| -) 22.86%**

**--------------------------------------------------**

**Correctly classified 82.11%**

**--------------------------------------------------**

(c)

The probit model was able to correctly classify 83.2% of the observations (count R2) with a McFadden’s R2 of 0.416, while the logit model was able to correctly classify 82.1% with a McFadden’s of 0.420. Although both are considered measures of fit, the count R2 is a more intuitive measure of the model’s predictive power; it just simply counts the percent of observations correctly classified as positive or negative by the model. The McFadden’s R2, on the other hand, is intuitively more like the F-test reported at the top of an OLS ANOVA table; it compares the maximum likelihood from a model with only a constant to the maximum likelihood of the unrestricted model. If McFadden’s R2 is low, you know that your independent variables aren’t very correlated with changes in percent.

(d)

Since we have the same number of observations, we can compare logit and probit models using the AIC in the “fitstat” command. The probit model has an AIC of 83.567, while the logit model has an AIC of 83.068. The lower AIC is the better model, so logit is the winner. Indeed, even just looking at the maximum likelihood values (-36.534 for logit and -36.784 for probit), logit is higher than probit.

(e)

Keeping the logit model parameters, we can estimate the effect that $30 higher water bills would have on the probability of saying yes with the following commands:

. gen logwaterbilpercent30 = log((waterbil+30)/hhinc)

. gen xb30 = \_b[environ]\*environ + \_b[urban]\*urban + \_b[logwaterbilpercent]\*logwaterbilpercent30 + \_b[logbidpercent]\*logbidpercent + \_b[\_cons]\*\_cons

. gen probyes30 = (exp(xb30))/(1+exp(xb30))

. gen ypayhat30 = probyes30>=0.5

For the first 10 observations, the following table compares the newly generated probabilities and yhats (with $30 higher water bills) with the previous ones:

**+-------------------------------------------------+**

**| ypay probyes probyes30 ypayhat ypayhat30 |**

**|-------------------------------------------------|**

**1. | 1 .9480367 .9651411 1 1 |**

**2. | 1 .888465 .9235978 1 1 |**

**3. | 0 .9281706 .9514794 1 1 |**

**4. | 1 .8494406 .8954189 1 1 |**

**5. | 1 .9919468 .9946787 1 1 |**

**|-------------------------------------------------|**

**6. | 1 .9098541 .9387141 1 1 |**

**7. | 1 .9869044 .9913319 1 1 |**

**8. | 1 .9939595 .9960114 1 1 |**

**9. | 1 .9982292 .9988324 1 1 |**

**10. | 0 .0919689 .1332273 0 0 |**

**+-------------------------------------------------+**

Increasing the water bill by $30 seems to have slightly increased the probability of saying yes, although it has not increased enough to change the predicted outcomes (ypayhat30). This makes sense with the model, since our marginal effect estimates assigned the water bill a positive, significant 0.174. Taking summary statistics of these new variables give us estimates on what the sample as a whole would do. The mean of our original ypayhat was 0.6316, and now the mean of our new ypayhat30 is 0.7158. This means that an additional 8% of our sample would say yes to the environmental program if all of their water bills were $30 higher.

(f)

Without a “real” error term, the standard errors of a maximum likelihood estimator must be calculated in some other way. We know that regardless of the distribution chosen for the estimator, the maximum likelihood values will start low, increase as we approach the true mean, and decrease once we’ve passed it. The speed in which this curve increases and decreases will tell us something about how confident we are in the values; if the curve is really spread out, there are many values that are very close to the maximum, but if the curve is really steep, we can be relatively sure that that one xβ is the correct one. The second derivative of the likelihood function will give us some measure if it’s curvature, which can be used to calculate standard errors.