Lab 6: Binary Regression and Data Generation

EDP 380C.28: Simulation in R

Fall 2022 – Unique: 12104

Brian T. Keller, Ph.D.
bk@utexas.edu
University of Texas at Austin
November 7, 2022

Objectives

- Learn about the specification of logistic and probit regressions.
- Learn to estimate logistic and probit regression models via maximum likelihood.
- Learn to generate binary data via the probit model.

Lab 6 builds on what we have done in Lab 5 by extending the methods to binary regression. To begin, let us first understand the nature of how we will model binary variables. First consider a single observation of a random variable that takes on a '0' for a failure and a '1' for a success. In addition, the rate by which this process generates successes is defined by its probability, π. Conversely, the rate of failures is determined by $1-\pi$. Such a variable follows a Bernoulli distribution, which is a special case of the binomial distribution for only one trial (i.e., $n = 1$). In terms of distributions, we will write this as follows.

$$Y \sim \mathrm{B}\left(1, \pi\right) \tag{1}$$

The binomial distribution is represented as $\mathrm{B}\left(n, \pi\right)$, with n independent trials with a probability of π success.

Starting from the above foundation, we can construct a regression model to relate the outcome (Y) to a set of predictors. We will create this link by relating the probability of each observation's success to the predictors via a function and a linear combination of regression slopes and the predictors.

$$\begin{aligned} y_i &\sim \mathrm{B}\left(1, \pi_i\right) \\ \pi_i &= f\left(\alpha + \mathbf{x}'\boldsymbol{\beta}\right) \end{aligned} \tag{2}$$

Note, the $\mathbf{x}$ vector does not include a 1 to represent the intercept of the model. In this instance, we are using the functional notation ($f\left(\cdot\right)$) to represent the link function that relates our linear combination ($\alpha + \mathbf{x}'\boldsymbol{\beta}$) to the predicted probability for observation i (π_i). In the subsequent portions of the lab, we will discuss two ways to link the probability to the linear combination of predictors: probit and logistic link functions.

1 Logistic Regression

Probit regression relates the probability to a linear combination of predictors (i.e., regression) using the natural logarithm of the odds.

$$\pi_i = \frac{\exp\{\alpha + \mathbf{x}_i'\boldsymbol{\beta}\}}{1 + \exp\{\alpha + \mathbf{x}_i'\boldsymbol{\beta}\}} = \frac{1}{1 + \exp\left\{-(\alpha + \mathbf{x}_i'\boldsymbol{\beta})\right\}} \tag{3}$$

Sometimes this is referred to as the "logit" or the "sigmoid" function in machine learning.

$$\pi_i = \text{logit}\left(\alpha + \mathbf{x}_i'\boldsymbol{\beta}\right) \tag{4}$$

Equation (4) is equivalent to Equation (3) and simply acts as a shorthand.

To write out the likelihood function for our logistic model, we apply the probability mass function (PMF) for a one trial Binomial distribution. For one observation the PMF is given in Equation (5).

$$\begin{aligned} P\left(\, y_i \mid \mathbf{x}_i, \boldsymbol{\beta}\,\right) &= \pi_i^{y_i}\left[1 - \pi_i\right]^{1-y_i} \\ &= \text{logit}(\alpha + \mathbf{x}_i'\boldsymbol{\beta})^{y_i}\left[1 - \text{logit}\left(\alpha + \mathbf{x}_i'\boldsymbol{\beta}\right)\right]^{1-y_i} \end{aligned} \tag{5}$$

The second line of Equation (5) is given by substituting Equation (4) for π_i. Conceptually, when $y_i = 1$, the probability is given by the logit of our predicted score. Conversely, when $y_i = 0$, the probability is given by one minus the logit of our predicted score. With the PMF specified for a single observation, we can construct the likelihood function by multiplying across each observation:

$$\mathcal{L}\left(\alpha, \boldsymbol{\beta} \mid y, \mathbf{X}\right) = \prod_{i=1}^{N} P\left(\, y_i \mid \mathbf{x}_i, \alpha, \boldsymbol{\beta}\,\right) \tag{6}$$

Taking the natural logarithm of Equation (6) provides the log likelihood.

$$\begin{aligned} \ell\left(\alpha, \boldsymbol{\beta} \mid y, \mathbf{X}\right) = \ln\left\{\mathcal{L}\left(\alpha, \boldsymbol{\beta} \mid y, \mathbf{X}\right)\right\} &= \ln\left\{\prod_{i=1}^{N} P\left(\, y_i \mid \mathbf{x}_i, \alpha, \boldsymbol{\beta}\,\right)\right\} \\ &= \sum_{i=1}^{N} \ln\left\{P\left(\, y_i \mid \mathbf{x}_i, \alpha, \boldsymbol{\beta}\,\right)\right\} \end{aligned} \tag{7}$$

1.1 Fitting a Logistic Regression in R

First we will use an example data set provided in the `data` folder in the Lab repository. To read in the data use the `read.csv()` function.

```
> binary <- read.csv('data/binary.csv')
> head(binary)
  admit gre  gpa rank
1     0 380 3.61    3
2     1 660 3.67    3
3     1 800 4.00    1
4     1 640 3.19    4
5     0 520 2.93    4
6     1 760 3.00    2
```

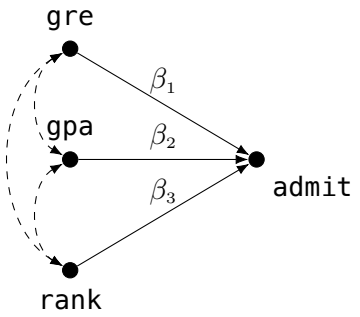

Figure 1: Graph of the Probit Regression Example Model

With the hypothetical data set, we will construct a model to predict graduate school admittance (`admit`) from an applicants GRE (`gre`), GPA (`gpa`), and prestige of undergraduate institution (`rank`). The DAG for this model is presented in Figure 1.

Before implementing the logistic estimation algorithm, first we need to implement a numerically stable `logit()` function. Implement the logit function using the following step function:

$$\text{logit}\,(x) = \begin{cases} x < 0 & \frac{\exp\{x\}}{1+\exp\{x\}} \\ x \geq 0 & \frac{1}{1+\exp\{-x\}} \end{cases} \tag{8}$$

1.a In R:

(1) Fit the logistic regression model given by Figure 1 in R via the `glm()` function. To accomplish this, you will need to set the `family = binomial("logit")`.

(2) Obtain the variance/covariance parameters via the `vcov()` function.

(3) Implement the numerically stable `logit` function in Equation (8).

Answer:

In the comments of the code, provide the following:

(1) The entire output of the `summary()` function for the `glm` model you fit.

(2) The variance/covariance matrix produced by `vcov()`

(3) The difference between the predicted probabilities from the `glm` function (i.e., use `predict(model, type = "response")`) and the predicted probabilities from your `logit` function (i.e., `logit(predict(model))`). Summarize this comparison by taking the sum of the differences across all observations. If implemented correctly, this should be near 0.

1.2 Estimating Logistic Regression via Newton-Raphson/Fisher Scoring

Logistic regression is a special case where the expected and observed information are identical. Therefore, implementing the Newton-Raphson method will provide the same results as Fisher

scoring. Recall that a single iteration of the Netwon-Raphson method is updated by using the ratio of the first derivative of the log likelihood and the second derivative of the log likelihood.

$$\theta^{(t)} = \theta^{(t-1)} - \left(\frac{\partial^2 \ell}{\partial \theta^2}\right)^{-1} \frac{\partial \ell}{\partial \theta} \tag{9}$$

We can write the above update step in terms of the Hessian matrix (i.e., matrix of second order partial derivatives) and the score vector (i.e., gradient / vector of first order partial derivatives) for the logistic regression as follows:

$$\boldsymbol{\theta}^{(t)} = \boldsymbol{\theta}^{(t-1)} - \left[\mathbf{H}\left(\boldsymbol{\theta}^{(t-1)}\right)\right]^{-1} s\left(\boldsymbol{\theta}^{(t-1)}\right) \tag{10}$$

where $\boldsymbol{\theta}' = [\alpha, \boldsymbol{\beta}]$ (the parameters in our model), $\mathbf{H}(\boldsymbol{\theta})$ is the Hessian matrix, and $s(\boldsymbol{\theta})$ is the score vector.

To compute the score vector and Hessian matrix, first we will construct the design matrix (i.e., the first column is a vector of ones for the intercept followed by the predictors), henceforth denoted as $\widetilde{\mathbf{X}}$. Using this design matrix, we can compute the score vector for the intercept and regression coefficients with the following computation.

$$\frac{\partial \ell}{\partial \boldsymbol{\theta}} = \widetilde{\mathbf{X}}' \left[\mathbf{y} - \text{logit}\left(\widetilde{\mathbf{X}}\boldsymbol{\theta}\right)\right] \tag{11}$$

The result of this computation will be a $p+1$ vector of partial derivatives that we can use in Equation (10). Similarly, we can compute the Hessian matrix using the design matrix via the following computation

$$\mathbf{H}(\boldsymbol{\theta}) = -\widetilde{\mathbf{X}}'\mathbf{W}\widetilde{\mathbf{X}} \tag{12}$$

where $\mathbf{W}$ is a $N \times N$ diagonal weight matrix of the model implied variance for an observation.

$$w_{ii} = \text{logit}(\alpha + \mathbf{x}_i'\boldsymbol{\beta})\left[1 - \text{logit}(\alpha + \mathbf{x}_i'\boldsymbol{\beta})\right] \tag{13}$$

Finally, we can obtain the standard errors for our maximum likelihood estimates. This process is straightforward using the Newton-Raphson algorithm. Recall from the previous lab that the variance/covariance matrix of the parameters is equal to the the inverse of the Hessian matrix multiplied by -1:

$$\text{Var}(\boldsymbol{\theta}) = \left[-\mathbf{H}(\boldsymbol{\theta})\right]^{-1} \tag{14}$$

and the standard error is equal to the square root of the diagonal elements (i.e., square root of the parameter variance matrix).

1.b In R:

Create functions that implements computing the score vector and the Hessian matrix for logistic regression. Use these functions to then create a function that implements Fisher scoring to produce the logistic regression estimates. The input of the function should use the following form.

```
function(y, X, tol = 1e-8, maxiter = 25)
```

The output should be a `list()` that includes the following items (names in brackets).

- number of iterations (`niter`)
- final log likelihood value (`loglik`)
- Model summary with the following form (`summary`):

	Estimate	SE	z value	Pr(>|z|)
b0	—	—	—	—
b1	—	—	—	—
⋮	⋮	⋮	⋮	⋮
bp	—	—	—	—

Hint: The results from your Hessian function evaluated at the estimated parameters should map onto the `vcov` function from the previous answer.

Answer:

Analyze the model in Figure 1 and add the resulting output from your function as a comment in the code. The results should map onto the R `glm` results exactly.

2 Probit Regression

Probit regression relates the probability to a linear combination of predictors (i.e., regression) using the cumulative distribution function (CDF) for the standard normal distribution (Φ).

$$\pi_i = \Phi\left(\alpha + \mathbf{x}_i'\boldsymbol{\beta}\right) \tag{15}$$

By applying the standard cumulative distribution function, we are mapping the sample space of a probability (i.e., 0 to 1) to the shape of a normal distribution.

To write out the likelihood function for our probit model, we apply the probability mass function (PMF) for a one trial Binomial distribution. For one observation the PMF is given in Equation (16).

$$\begin{aligned} P\left(y_i \mid \mathbf{x}_i, \boldsymbol{\beta}\right) &= \pi_i^{y_i}\left[1-\pi_i\right]^{1-y_i} \\ &= \Phi\left(\alpha + \mathbf{x}_i'\boldsymbol{\beta}\right)^{y_i}\left[1-\Phi\left(\alpha + \mathbf{x}_i'\boldsymbol{\beta}\right)\right]^{1-y_i} \end{aligned} \tag{16}$$

The second line of Equation (16) is given by substituting Equation (15) for π_i. Conceptually, when $y_i = 1$, the probability is given by evaluating the predicted score via a standard normal CDF.

Conversely, when $y_i = 0$, the probability is given by one minus evaluating the predicted score via a standard normal CDF. With the PMF specified for a single observation, we can construct the likelihood function by multiplying across each observation:

$$\mathcal{L}\left(\boldsymbol{\beta} \mid y, \mathbf{X}\right) = \prod_{i=1}^{N} P\left(y_i \mid \mathbf{x}_i, \boldsymbol{\beta}\right) \tag{17}$$

Taking the natural logarithm of Equation (17) provides the log likelihood.

$$\begin{aligned} \ell\left(\alpha, \boldsymbol{\beta} \mid y, \mathbf{X}\right) = \ln\left\{\mathcal{L}\left(\alpha, \boldsymbol{\beta} \mid y, \mathbf{X}\right)\right\} &= \ln\left\{\prod_{i=1}^{N} P\left(y_i \mid \mathbf{x}_i, \alpha, \boldsymbol{\beta}\right)\right\} \\ &= \sum_{i=1}^{N} \ln\left\{P\left(y_i \mid \mathbf{x}_i, \alpha, \boldsymbol{\beta}\right)\right\} \end{aligned} \tag{18}$$

2.a In R:

(1) Fit the probit regression model given by Figure 1 in R via the `glm()` function. To accomplish this, you will need to set the `family = binomial("probit")`.

(2) Obtain the variance/covariance parameters via the `vcov()` function.

Answer:

In the comments of the code, provide the following:

(1) The entire output of the `summary()` function for the `glm` model you fit.

(2) The variance/covariance matrix produced by `vcov()`

2.1 Estimating Probit Regression via Newton-Raphson/Fisher Scoring

Similar to logistic regression, probit regression is a special case where the expected and observed information are identical. Therefore, by implementing the Newton-Raphson method we will obtain the same results as Fisher scoring. Recall that a single iteration of the Netwon-Raphson method is updated by using the ratio of the first derivative of the log likelihood and the second derivative of the log likelihood, Equation (9).

To compute the score vector and Hessian matrix, we will again use the design matrix ($\widetilde{\mathbf{X}}$) and vector of all parameters ($\boldsymbol{\theta}$). The score vector is given by a complicated function that we will break down to ease the computation. First, we must evaluate the predicted score using a standard normal CDF ($\phi\left[\cdot\right]$) and PDF ($\Phi\left[\cdot\right]$)—i.e., `dnorm` and `pnorm` respectively in R).

$$\begin{aligned} \mathbf{a} &= \phi\left(\widetilde{\mathbf{X}}\boldsymbol{\theta}\right) \\ \mathbf{b} &= \Phi\left(\widetilde{\mathbf{X}}\boldsymbol{\theta}\right) \end{aligned} \tag{19}$$

Each of the above vectors will result in a $n \times 1$ column vector. Next, we can use these column vectors and some elementwise matrix functions (i.e., $\odot$ is element wise multiplication and $\oslash$ is elementwise division).

$$\frac{\partial \ell}{\partial \boldsymbol{\theta}} = \widetilde{\mathbf{X}}'\left(\mathbf{a} \odot \left[\mathbf{y} \oslash \left(\mathbf{b} - \mathbf{b} \odot \mathbf{b}\right) - \frac{1}{1 - \mathbf{b}}\right]\right) \tag{20}$$

If computed correctly, the result of this computation will be a $p + 1$ vector of partial derivatives that we can use in Equation (10). Similarly, we can compute the Hessian matrix using the design matrix via the following computation

$$\mathbf{H}(\boldsymbol{\theta}) = -\widetilde{\mathbf{X}}'\mathbf{W}\widetilde{\mathbf{X}} \tag{21}$$

where $\mathbf{W}$ is a $N \times N$ diagonal weight matrix of the model implied variance for an observation. The diagonal is given by the following matrix expression.

$$\text{diag}(\mathbf{W}) = \mathbf{a} \odot \mathbf{a} \oslash (\mathbf{b} - \mathbf{b} \odot \mathbf{b}) \tag{22}$$

2.b In R:

Create functions that implements computing the score vector and the Hessian matrix for probit regression. Use these functions to then create a function that implements Fisher scoring to produce the logistic regression estimates. The input of the function should use the following form.

```
function(y, X, tol = 1e-8, maxiter = 25)
```

The output should be a `list()` that includes the following items (names in brackets).

- number of iterations (`niter`)
- final log likelihood value (`loglik`)
- Model summary with the following form (`summary`):

	Estimate	SE	z value	Pr(>|z|)
b0	—	—	—	—
b1	—	—	—	—
⋮	⋮	⋮	⋮	⋮
bp	—	—	—	—

Hint: The results from your Hessian function evaluated at the estimated parameters should map onto the `vcov` function from the previous answer.

Answer:

Analyze the model in Figure 1 and add the resulting output from your function as a comment in the code. The results should map onto the R `glm` results exactly.

3 Generating Data via Probit Model

An alternative way we can conceptualize a probit regression is by using a latent (unobserved) variable to represent the propensity of a single observation being observed as a 0 or a 1. We will denote this latent variable with a superscripted star ($\star$). We can write this latent variable as follows.

$$y_i^\star \sim \mathcal{N}(\alpha + \mathbf{x}_i'\boldsymbol{\beta}, 1) \tag{23}$$

where we define an indicator function between the latent $Y^\star$ and the manifest Y as follows.

$$Y = \begin{cases} 1 & Y^\star > 0 \\ 0 & \text{otherwise} \end{cases} \tag{24}$$

The two specifications can be shown to be equivalent.

$$\begin{aligned} P\left(Y_i = 1 \mid \mathbf{x}_i\right) &= P\left(Y^\star > 0\right) \\ &= P\left(\alpha + \mathbf{x}_i'\boldsymbol{\beta} + \epsilon_i > 0\right) \\ &= P\left(\epsilon_i < \alpha + \mathbf{x}_i'\boldsymbol{\beta}\right) \\ &= \Phi\left(\alpha + \mathbf{x}_i'\boldsymbol{\beta}\right) \end{aligned} \tag{25}$$

The result of the above equation matches Equation (15). Just as there are two ways to think of the probit model, there are two ways to generate the data. We can either specify the probability and draw Y from a binomial distribution or simulate the $Y^\star$ and categorize it based upon the indicator function in Equation (24). We will focus on the latent variable specification because it is helpful to diagnose the data generating process by analyzing the latent $Y^\star$ values.

For our discussion, we will assume that you have already generated the $\mathbf{X}$ variables from a multivariate normal distribution with a mean and covariance matrix,

$$\mathbf{X} \sim \mathcal{N}_p\left(\boldsymbol{\mu}_x, \boldsymbol{\Sigma}_x\right) \tag{26}$$

and we are interested in generating the Y values. Additionally, we will specify the relationship between our Y and $\mathbf{X}$ variables via the proportion of variance explained (R^2) in the latent $Y^\star$.The advantage of this approach is that it will allow for the specification of nonlinear relationships where a correlation does not accurately capture the association.

To begin, we will generate a proxy variable, denoted as Z, that is, the linear combination of our predictors. This variable will act as our latent propensity and be transformed into $Y^\star$ in the subsequent step. The model for p predictors is as follows.

$$z_i \sim \mathcal{N}\left(\mathbf{x}_i'\boldsymbol{\gamma}, 1\right) \tag{27}$$

Note, $\boldsymbol{\gamma}$ is the vector of regression coefficients. Notably, there is no intercept in the model, and we have yet to select the coefficients and residual variance parameters. By focusing on the variance explained, the values of the regression coefficients themselves are less important. Just like with Method 1 of linear regression data generation, we will leave all coefficients equal to 1, which denotes that each variable equally contributes to the explained variance. Based on linear regression, we can compute the variance explained using the familiar following equation.

$$\sigma^2_{\text{explained}} = \boldsymbol{\gamma}'\boldsymbol{\Sigma}_x\boldsymbol{\gamma} \tag{28}$$

Next, we need to select some R^2 that we want to generate and compute the residual variance based on the derived formula.

$$\sigma^2_e = \sigma^2_{\text{explained}}\left(\frac{1}{R^2} - 1\right) \tag{29}$$

With σ^2_e solved, we can generate our Z variable using draws from a normal distribution based upon Equation (27). Next, we will standardize Z using its model implied variance and mean. The Z-score formula gives the standardized score:

$$Z^\star = \frac{Z - \mu_z}{\sigma_z} \tag{30}$$

where the model implied variance is:

$$\sigma_z^2 = \sigma_e^2 + \sigma_{\text{explained}}^2 \tag{31}$$

and the mean is:

$$\mu_z = \boldsymbol{\mu}_x' \boldsymbol{\gamma} \tag{32}$$

With the proxy $Z^\star$ variable generated, we will then transform it to map onto $Y^\star$ that is specified by the probit specification in R. As part of this process, we will define the scale of $Y^\star$ to have a residual variance equal to one (as implied by the model in Equation (23)) and the intercept value (which will define the proportion of ones in the population; denoted as τ). We rescale $Z^\star$ based on the following formula.

$$Y^\star = \frac{Z^\star + \Phi^{-1}(\tau)}{\sqrt{1 - R^2}} \tag{33}$$

Note that Φ^{-1} is the inverse cumulative standard normal distribution function (i.e., `qnorm()` in R), and τ is the population proportion that we want. Finally, with the $Y^\star$ generated, we can categorize the latent response into the manifest observed variable using the indicator function in Equation (24).

3.1 Obtaining Population Parameters

The final step is to calculate the population values of the regression coefficients for the probit model. To do this, we need to transform the intercept and the $\boldsymbol{\gamma}$ parameters we specified in the data generation. The definition of the regression slopes is given using the following transformation from linear regression.

$$\boldsymbol{\beta} = \frac{\boldsymbol{\gamma}}{\sigma_z \sqrt{1 - R^2}} \tag{34}$$

Conceptually, the denominator is the product of the two scaling terms we used to transform Z to $Y^\star$.

The intercept, α, is defined by the proportion (τ) using the inverse cumulative standard normal distribution function scaled by the total standard deviation of $Y^\star$. The definition of the regression intercept is given using the following transformation.

$$\alpha = \Phi^{-1}(\tau) \times \sqrt{1 - \frac{R^2}{R^2 - 1}} - \mu_x \boldsymbol{\beta} \tag{35}$$

The above equation uses the standard regression formula for an intercept. The first product is the mean of $Y^\star$. To reiterate, we scale the cumulative normal distribution function based upon the square root of the total variance of $Y^\star$ defined by the known R^2 and the residual variance (which is equal to one). The second product is the means of our predictors multiplied by the regression coefficients. If our predictors are centered, this term drops out.

3.a In R:

(1) Create a function that generates p-predictors that are multivariate normally distributed and a binary outcome that is generated in accordance to the probit model. This function should map onto the following inputs:

```
function(n, p_x, p_y)
```

where `p_x` is the parameters for the predictors (i.e., a mean vector and covariance matrix) and `p_y` is the parameters for the outcome model (i.e., population proportion, fixed regression weights (i.e., γ in the equations), and variance explained of the latent propensity (R^2). The output should include the categorized Y, the latent $Y^\star$, and the p-predictors.

(2) Set the seed to `1249872`.

(3) Run a 500 replication by 2 condition simulation that evaluates your data generation function using your probit regression function. Use the following model parameters: $\boldsymbol{\mu}_x' = [1, 2, 3, 4, 5]$, the correlation of all predictors are 0.10, the standard deviations of the predictors start at 1 and go to 5, the γ all are equal to 1, and $\mathsf{R}^2 = 0.6$. Set the population proportion to 0.25 in the first condition and 0.5 in the second.

Hint: You will need to solve for the population parameters yourself.

Answer:

Construct the following table for each condition and include it as a comment in your code.

	Pop.	Y Star	Est.	P. Bias	PB JSE	Coverage	Coverage JSE
b0	—	—	—	—	—	—	—
b1	—	—	—	—	—	—	—
b2	—	—	—	—	—	—	—
b3	—	—	—	—	—	—	—
b4	—	—	—	—	—	—	—
b5	—	—	—	—	—	—	—

Column Defintions:

- `Pop.` = Population values
- `Y Star` = Linear regression estimate between $Y^\star$ and predictors (use `lm`).
- `Est.` = Probit regression estimate
- `P. Bias` = Percent bias of the probit estimate
- `PB JSE` = Jackknife standard errors of percent bias estimate
- `Coverage` = Coverage of the probit estimate
- `Coverage JSE` = Jackknife standard errors of the coverage estimate