Logit and Probit Models

Week 5
POLS 8830: Advanced Quantitative Methods

Ryan Carlin
Georgia State University
rcarlin@gsu.edu

Application of Logit (or Probit) — Dependent Variable

- Logit/Probit models should only be used with a dichotomous dependent variable
- Norm is to code values as (0) and (1)
- Must have approximately same number of 0s and 1s in data
- Logit and Probit start to break down when balance becomes worse than 60/40 (consider Skewed Logit as an alternative; package **glogis**, call `glogis`)
- Logit/Probit estimates are practically worthless at 90/10 split (consider Rare Events Logit; package **Zelig**, call `relogit`)

Logit Model

- The logit model is calculated using the Logistic distribution
- As you may recall, the probability density function (pdf) for the logistic distribution is:

$$\Lambda(\mathbf{X}\beta) = \frac{e^{\mathbf{X}\beta}}{[1 + e^{\mathbf{X}\beta}]^2}$$

Logit Model

- MLE requires calculating joint probabilities (likelihood of observing the entire dataset), rather than the probability of observing a single observation
- This requires us to work with the cumulative density function (cdf) rather than the pdf of the logistic distribution:

$$\Lambda(\mathbf{X}\beta) = \int_{-\infty}^{\mathbf{X}\beta} \frac{e^{\mathbf{X}\beta}}{[1 + e^{\mathbf{X}\beta}]^2}$$

- Which reduces to:

$$\Lambda(\mathbf{X}\beta) = \frac{1}{1 + e^{-\mathbf{X}\beta}}$$

Logit: Assumptions

- Many of the OLS assumptions are relaxed
- Logit/Probit Assumptions/Requirements:
  1. Binary Dependent Variable
  2. Independent Observations
  3. No perfect multicollinearity
  4. Linearity of relationship between IVs and Log Odds
  5. Larger sample sizes
     - At minimum of 10 cases for the least frequent outcome for each independent variable in your model
     - Alternatively, the average tends to results in desirable behavior with 100 cases per IV

Logit Model: Estimation

- The basic syntax for estimating a logit model in R is:
  - `glm(formula, family=binomial(link="logit"), data, weights, subset, ...)`
  - `formula:` DV $\sim$ IV_1 + IV_2 ...IV_n
- `subset` is used to estimate for a subset of the data based on user defined conditions
- A variety of options exists with logit models (see ?glm() and the glm documentation for a full list)

Logit Model: Estimation

- The most common issue relates to how the standard errors are calculated (e.g. **sandwich**/`vcov(...)` for Huber-White robust standard errors, clustered standard errors, etc.)
  - This solution for heteroskedasticty is not theoretically appropriate for logit/probit
  - Best practice is to explicitly model the source(s) of heteroskedasticity and/or modify the distributional form to account for this influece
  - We'll cover this in the week on MLM/hierarchical modeling

Logit Model: Estimation

- An example:

```
Call:
glm(formula = lfp ~ k5 + k618 + age + wc + inc, family = binomial(link = "logit"))

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.0883  -1.1176   0.6386   0.9954   2.0975

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  3.799878   0.622838   6.101 1.05e-09 ***
k5          -1.463093   0.194636  -7.517 5.60e-14 ***
k618        -0.087589   0.067266  -1.302    0.193
age         -0.063660   0.012579  -5.061 4.18e-07 ***
wcyes        1.062298   0.198689   5.347 8.97e-08 ***
inc         -0.030765   0.007685  -4.003 6.24e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1029.75  on 752  degrees of freedom
Residual deviance:  922.61  on 747  degrees of freedom
AIC: 934.61

Number of Fisher Scoring iterations: 3
```

Probit Model

- Calculated using the normal distribution.
- As with logit models, we must work with the cumulative density function (cdf) rather than the pdf of the normal distribution (note: the normal cdf is bounded by 0 and 1):

$$\Phi(y|\mathbf{X}\beta) = \int_{-\infty}^{\mathbf{X}\beta} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\left[\frac{(y-\mu)^2}{\sigma^2}\right]}$$

- The complexity of the above calculation made use of probit models impractical for early researchers due to the lack of an analytical solution and the lack of computing power

Probit Model: Estimation

- The basic syntax for estimating a probit model in Stata is:
  - `glm(formula, family=binomial(link="probit"), data, weights, subset, ...)`
  - `formula:` DV $\sim$ IV_1 + IV_2 ...IV_n

Probit Model: Estimation

- An example:

```
Call:
glm(formula = lfp ~ k5 + k618 + age + wc + inc, family = binomial(link = "probit"))

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.1127  -1.1226   0.6403   0.9991   2.1176

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  2.282634   0.367479   6.212 5.24e-10 ***
k5          -0.878650   0.113308  -7.755 8.87e-15 ***
k618        -0.051854   0.040551  -1.279    0.201
age         -0.038137   0.007487  -5.094 3.51e-07 ***
wcyes        0.637413   0.117418   5.429 5.68e-08 ***
inc         -0.018499   0.004569  -4.049 5.14e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1029.75  on 752  degrees of freedom
Residual deviance:  923.04  on 747  degrees of freedom
AIC: 935.04

Number of Fisher Scoring iterations: 4
```

Comparing Logit and Probit

- Both distributions are symmetric
- Differences occur in the tails of the distributions
  - Probit has thinnest tail
  - Logit has slightly thicker tails
  - Note: logit coefficients are approximately 1.7 times higher than probit coefficients

Comparing Logit and Probit

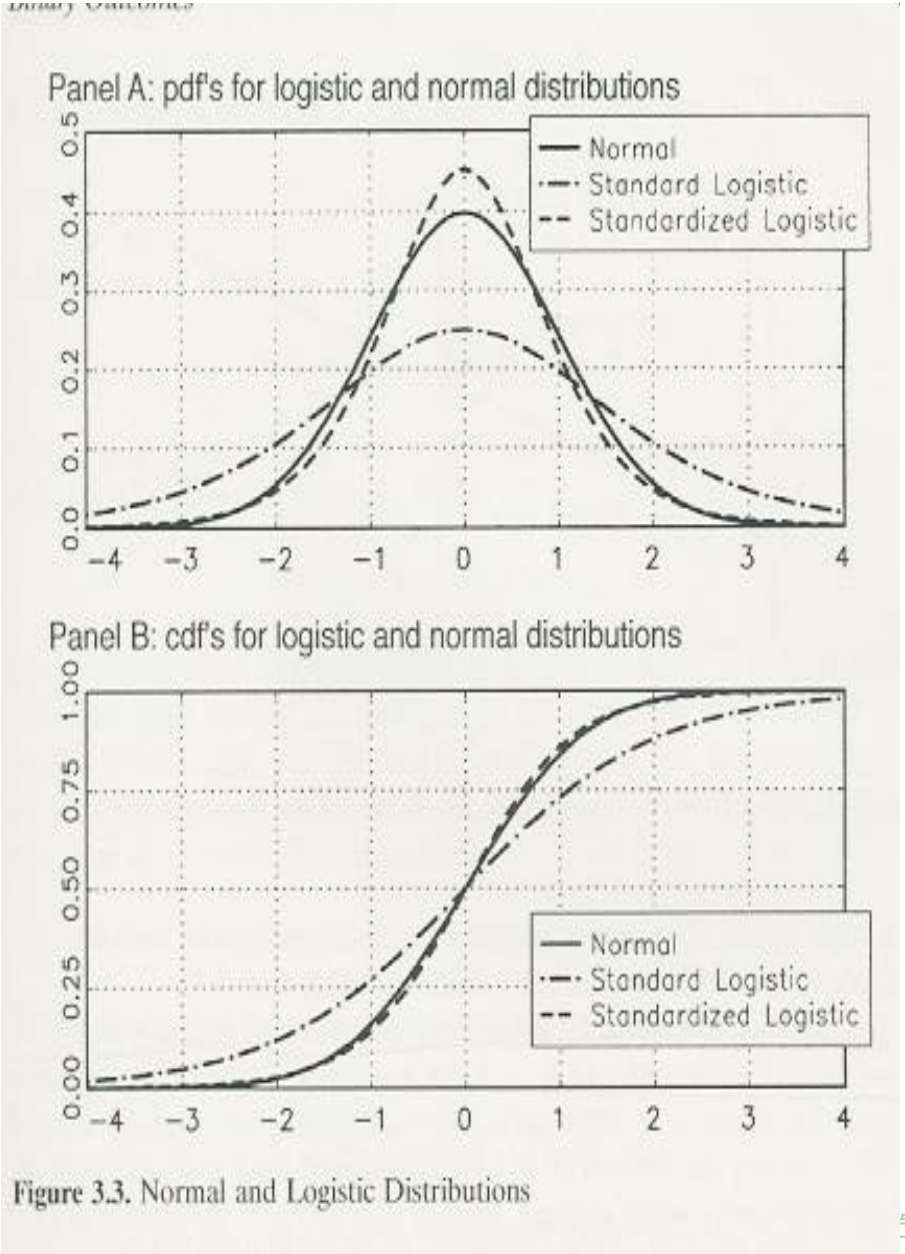

Figure 3.3. Normal and Logistic Distributions

Comparing Logit and Probit

Table: Effect of Various Factors on Likelihood of Female Labor Force Participation

	Logit	Probit
Children:	-0.879^{***}	-0.543^{***}
Under 5	(0.158)	(0.095)
Children	0.027	0.016
6 to 18	(0.058)	(0.036)
Constant	0.445^{***}	0.278^{***}
	(0.111)	(0.069)
Observations	753	753
Log Likelihood	-497.263	-497.260
Akaike Inf. Crit.	1,000.525	1,000.519
Wald χ^2	30.912 ***	32.763 ***
P.R.E.	0.126	0.126
Note:	*p<0.1; **p<0.05; ***p<0.01	

Goodness-of-Fit: Wald χ^2

- A variety of R^2 type measures have been developed for the MLE context
- However, these measures are virtually useless and no one relies on them in practice (in fact they are rarely even reported)
- Wald χ^2 is a basic measure of overall model fit, intuitively similar to F-test in OLS context *(Note: always report this)*

Goodness-of-Fit: Wald χ^2

- The Wald χ^2 test functions like a simultaneous series of likelihood-ratio tests
- The Wald χ^2 test essentially provides a comparison of model fit by constraining covariates to 0, comparing the results of the model if the covariate influence was 0, then measuring if the models are statistically different through the use of a χ^2 statistic
- Significant results indicate that the difference between the original model and the model with (functionally) omitted variables is not zero

Goodness-of-Fit: Wald χ^2

- The Wald χ^2 test functions like a simultaneous series of likelihood-ratio tests
- The Wald χ^2 test essentially provides a comparison of model fit by constraining covariates to 0, comparing the results of the model if the covariate influence was 0, then measuring if the models are statistically different through the use of a χ^2 statistic
- Significant results indicate that the difference between the original model and the model with (functionally) omitted variables is not zero
- Substantively: are the covariates adding explanatory power to the model above 0

Goodness-of-Fit: Wald χ^2

- Can calculate this in R with the command `wald.test` from the **aod** package
- `wald.test(Sigma, b, Terms = NULL, L = NULL, H0 = NULL, df = NULL, verbose = FALSE)`
- `wald.test(b = coef(model_name), Sigma = vcov(model_name), Terms = 2:length(model_name$coefficients))`
  - `This is testing the entirety of the model for the goodness of fit -- not practically useful for model comparison and specification tests`
  - For `Terms =`, we want to start at 2 because 1 refers to the intercept
  - We want to end at `length(model_name$coefficients)` for repeatability. You can specify the number of coefficients (IVs) in the model plus 1 (the intercept).

Goodness-of-Fit: Proportional Reduction of Error

- A good alternative involves comparing the observed values to the predicted values
- The logic behind this approach is to see how well your predictions fare against simply observing the data
- If the predicted values of the model provide a better explanation than simply "looking" at the data, then the model is useful

Goodness-of-Fit: Proportional Reduction of Error

- Let the observed dependent variable ($\mathbf{y}$) equal 0 or 1 (this is generalizable beyond two categories)
- Let π represent the predicted probability that $(\mathbf{y}_i) = 1$
- And $\pi_i = \Pr(\mathbf{y} = 1|\mathbf{X}_i) = f(\mathbf{X}_i\beta)$
- where f = the cdf for the Normal distribution in probit and the cdf for the Logistic distribution in logit

Goodness-of-Fit: Proportional Reduction of Error

- Define the expected value for $\hat{\mathbf{y}}$ as:

$$\hat{\mathbf{y}} = \begin{cases} 0 & \text{if } \hat{\pi}_i \leq 0.5 \\ 1 & \text{if } \hat{\pi}_i > 0.5 \end{cases}$$

- A table is helpful for comparison purposes and helps visualize the intuition behind this approach

	Observed Values	
	0	1
Predicted 0	+	-
Predicted 1	-	+

Goodness-of-Fit: Proportional Reduction of Error

- The Proportional Reduction of Error (PRE) statistic calculates the proportion of + versus the proportion of - to determine the predictive accuracy, using this formula:

$$\text{PRE} = \frac{\%\text{ correctly predicted} - \%\text{ in modal category}}{100 - \%\text{ in modal category}}$$

Goodness-of-Fit: Proportional Reduction of Error

- There is no current implementation of PRE in R
- You'll find our user defined function on iCollege in both the R tutorial and its own separate file
  - You may wish to save the PRE.R file for use later
- Does all the calculations for you, but relies on the default .5 threshold. Only works with binary models.
- You can change this by changing the 0.5 in line 13 of the PRE.R file
  - `predict<-ifelse(model_name$fitted.values> 0.5, 1, 0)`

Goodness-of-Fit: Proportional Reduction of Error

- `pre(model)`
- `model` is a call to your glm (logit/probit) object
  - `model<-glm(DV ~ . , family = binomial(link = "logit"))`
- Provides `pre.object$PRE` if you wish to use the numerical value of PRE for other purposes
- Also includes `stargazer.pre()` for use in `stargazer`
  - See tutorial for usage
- An Example:

```
> pre(m2)

   Proportion Correctly Predicted:      0.681
   Proportion Modal Category:           0.568
   Proportional Reduction in Error:     0.262

   Percent Correctly Predicted:         68.127
   Percent Modal Category:              56.839
   Percent Reduction in Error:          26.154

         PRE
1 0.2615385
```

Model Comparison: Likelihood Ratio Test

- Likelihood Ratio Test (LR Test)
  - Compares β estimates from a constrained and unconstrained model
  - Assesses the imposed constraint by comparing the log-likelihoods of the constrained model to the unconstrained one
  - H_0: $\beta_u = \beta_c$
  - If the null hypothesis is rejected, then the unconstrained model is a significant improvement over the constrained
    - Thinking in terms of explanatory power, a rejected null indicates the loss in degrees of freedom is worth the addition of the additional covariates

Model Comparison: Likelihood Ratio Test

- Likelihood Ratio Test (LR Test)
  - Compares β estimates from a constrained and unconstrained model
  - Assesses the imposed constraint by comparing the log-likelihoods of the constrained model to the unconstrained one
  - H_0: $\beta_u = \beta_c$
  - If the null hypothesis is rejected, then the unconstrained model is a significant improvement over the constrained
    - Thinking in terms of explanatory power, a rejected null indicates the loss in degrees of freedom is worth the addition of the additional covariates
  - Constrained meaning a subset of the IVs from the unconstrained model
    - Unconstrained: DV $\sim$ IV1 + IV2 + IV3 ...
    - Constrained: DV $\sim$ IV1 + IV2 ...

Model Comparison: Likelihood Ratio Test

- From the **lmtest** package, call `lrtest()`
- `lrtest(unconstrained_model_object, constrained_model_object)`

```
> m3$coefficients
(Intercept)          k5        k618
 0.27790495 -0.54301046  0.01641896
> m4$coefficients
(Intercept)          k5        k618         age       wcyes         inc
 2.28263351 -0.87865032 -0.05185438 -0.03813694  0.63741286 -0.01849870
> lrtest(m4, m3)
Likelihood ratio test

Model 1: lfp ~ k5 + k618 + age + wc + inc
Model 2: lfp ~ k5 + k618
  #Df  LogLik Df  Chisq Pr(>Chisq)
1   6 -461.52
2   3 -497.26 -3 71.483  2.055e-15 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Model Comparison: Wald χ^2

- A replication of the LR test can be performed through the Wald χ^2 test
- Can set different combinations of covariates to examine their influence with the (`Terms = `) call
- Functionally, you are setting individual or sets of covariates to 0, and thus their influence to 0, to see if their inclusion adds explanatory power to the model
- This is asymptotically equivalent to the LR test, but can be quicker and simpler in R as it only requires the estimation of a single model

Model Comparison: Wald χ^2

```
> summary(m2)

Call:
glm(formula = lfp ~ k5 + k618 + age + wc + inc, family = binomial(link = "logit"))

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.0883  -1.1176   0.6386   0.9954   2.0975

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  3.799878   0.622838   6.101 1.05e-09 ***
k5          -1.463093   0.194636  -7.517 5.60e-14 ***
k618        -0.087589   0.067266  -1.302    0.193
age         -0.063660   0.012579  -5.061 4.18e-07 ***
wcyes        1.062298   0.198689   5.347 8.97e-08 ***
inc         -0.030765   0.007685  -4.003 6.24e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1029.75  on 752  degrees of freedom
Residual deviance:  922.61  on 747  degrees of freedom
AIC: 934.61

Number of Fisher Scoring iterations: 3
```

```
> wald.test(b = coef(m2), Sigma = vcov(m2), Terms = 2:2)
Wald test:
----------

Chi-squared test:
X2 = 56.5, df = 1, P(> X2) = 5.6e-14
> wald.test(b = coef(m2), Sigma = vcov(m2), Terms = 3:3)
Wald test:
----------

Chi-squared test:
X2 = 1.7, df = 1, P(> X2) = 0.19
> wald.test(b = coef(m2), Sigma = vcov(m2), Terms = 4:4)
Wald test:
----------

Chi-squared test:
X2 = 25.6, df = 1, P(> X2) = 4.2e-07
> wald.test(b = coef(m2), Sigma = vcov(m2), Terms = 5:5)
Wald test:
----------

Chi-squared test:
X2 = 28.6, df = 1, P(> X2) = 9e-08
> wald.test(b = coef(m2), Sigma = vcov(m2), Terms = 2:3)
Wald test:
----------

Chi-squared test:
X2 = 56.5, df = 2, P(> X2) = 5.3e-13
```

- We can see that the third covariate (k618) does not add explanatory power to the model worth its inclusion

Model Comparison

- One may wish to know the added explanatory power for additional covariates
- Informational measures such as the Akaike's Information Criterion (AIC) and Bayesian Information Criterion (BIC) exist for these model comparison purposes
- These only function for models analyzed on the same data set
  - Long disagrees
- These balance model fit with parsimony – are additional covariates worth adding on the basis of added analytical leverage against the cost of their inclusion
- BIC more strongly penalizes the addition of further IVs

Model Comparison

- AIC is generally provided after logit and probit models in `glm()`
- If you wish to see the BIC, use package **stats4** with the call `BIC`
- Lower AIC or BIC values speak to a 'better' model
  - Compare like-to-like: AIC to AIC, BIC to BIC, never AIC to BIC, they are calculated differently

Model Comparison

- Cannot compare AIC or BIC on models with different covariates
- Can compare:
  - DV $\sim$ IV1 + IV2 + IV3 ...
  - DV $\sim$ IV1 + IV2 ...
- But these cannot be compared to:
  - DV $\sim$ IV1 + IV4 ...
- You'd need to compare:
  - DV $\sim$ IV1 + IV2 + IV3 + IV4 ...
  - DV $\sim$ IV1 + IV4 ...

Model Comparison

```
> m1

Call:  glm(formula = lfp ~ k5 + k618, family = binomial(link = "logit"))

Coefficients:
(Intercept)           k5         k618
     0.4448      -0.8793       0.0273

Degrees of Freedom: 752 Total (i.e. Null);  750 Residual
Null Deviance:      1030
Residual Deviance: 994.5        AIC: 1001
> m2

Call:  glm(formula = lfp ~ k5 + k618 + age + wc + inc, family = binomial(link = "logit"))

Coefficients:
(Intercept)           k5         k618          age        wcyes          inc
    3.79988     -1.46309     -0.08759     -0.06366      1.06230     -0.03076

Degrees of Freedom: 752 Total (i.e. Null);  747 Residual
Null Deviance:      1030
Residual Deviance: 922.6        AIC: 934.6
>
> BIC(m1)
[1] 1014.398
> BIC(m2)
[1] 962.3515
```

- As we can see, even though the second model contains more terms, the BIC is lower. This means that the explanatory power provided by the additional terms outweighs the cost of their inclusion