# Big Data and Security

**Jeffrey Borowitz, PhD**
*Lecturer*
Sam Nunn School of International Affairs

## Generalizing Linear Regression to Binary Outcomes

# Predicting Discrete Outcomes

- Lets say you are a bank and want to determine whether a transaction is fraudulent
- The strategy is to pretend you're modeling a "propensity" and then if the propensity is high enough, guess that it is fraudulent
- You can look at a bunch transactions (i) and make:

$$y_i = \begin{cases} 1, & if\ fraudulent \\ 0, & if\ not\ fraudulent \end{cases}$$

- And then you can just run a regression!
- Clearly bigger values of $\widehat{y_i}$ mean a transaction is more likely to be fraudulent
    - The problem is you want something between 100% sure and 0% sure that a transaction is a fraud
    - Theoretically, your $\widehat{y_i}$ can be anything: bigger than 1, less than 0

Georgia Tech

# Predicting Discrete Outcomes: Logistic Regression

- We know our outcome will be either 0 or 1 (a fraud or not a fraud)
- So let's say there's some true underlying propensity to be fraudulent.
- So lets say we take our $\hat{y}$ and then draw a random ε from a particular distribution.
  - "Logistic" regression actually means we've picked a distribution for ε
- We can think of our X variable as moving around a probability distribution
  - If you take a draw of ε so that $\hat{y}$ + ε < 0, then the outcome is 0
  - If you take a draw of ε so that $\hat{y}$ + ε > 0, then the outcome is 1
- We can have an objective function like:

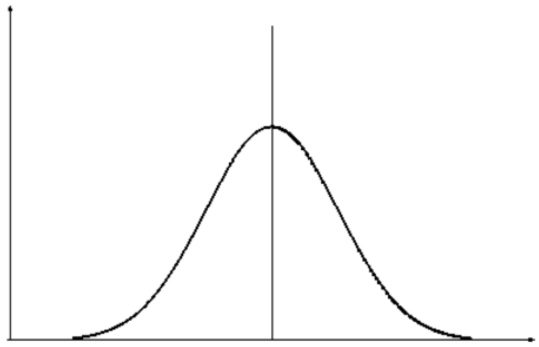
$$\sum_{\mathrm{i}} \mathrm{p}(\varepsilon_{\mathrm{i}}\ \text{gives observed}\ \mathrm{y}_{\mathrm{i}})$$

  - Here the function p(·) depends on the shape of our logistic distribution
  - And our parameters are allowing X to shift the probability up and down

Georgia Tech

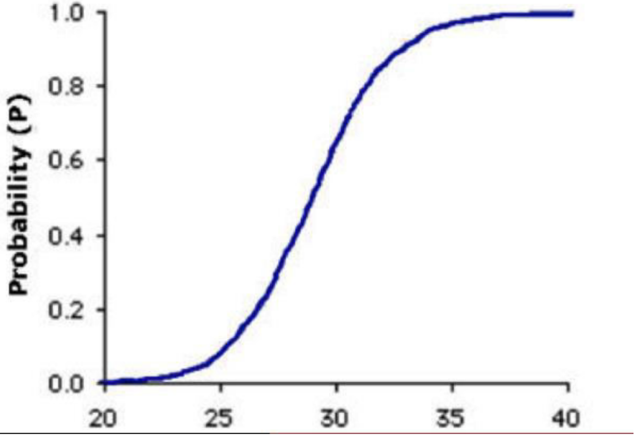
# Predicting Discrete Outcomes: Logistic Regression

- The center of the distribution depends on the X variables you care about
- If X is really big, curve moves way right, so you would never get a draw of ε so that $\alpha + \beta X + \varepsilon < 0$



# Logistic Regression

- Logistic regression is super common
- It has a nice role for uncertainty: because if you increase X, you can keep increasing your likelihood more and more and never getting to 100%



# Logistic Regression

- To estimate this, we change our objective function from

$$\min_{\alpha,\beta} F(\alpha,\beta) = \sum_i (y_i - \alpha - \beta x_i)^2$$

to

$$\min_{\alpha,\beta} F(\alpha,\beta) = \sum_i H(y_i - G(\alpha - \beta x_i))$$

where G and H are functions about the specific shape of the logistic function

- Note: this is NOT the traditional formulation, but is equivalent and I write it this way

# Probit Regression

- Instead of the logistic function, we could also use other functions
- For a **probit**, you use functions for G and H related to the shape of the normal distribution

$$\min_{\alpha,\beta} F(\alpha, \beta) = \sum_i H(y_i - G(\alpha - \beta x_i))$$

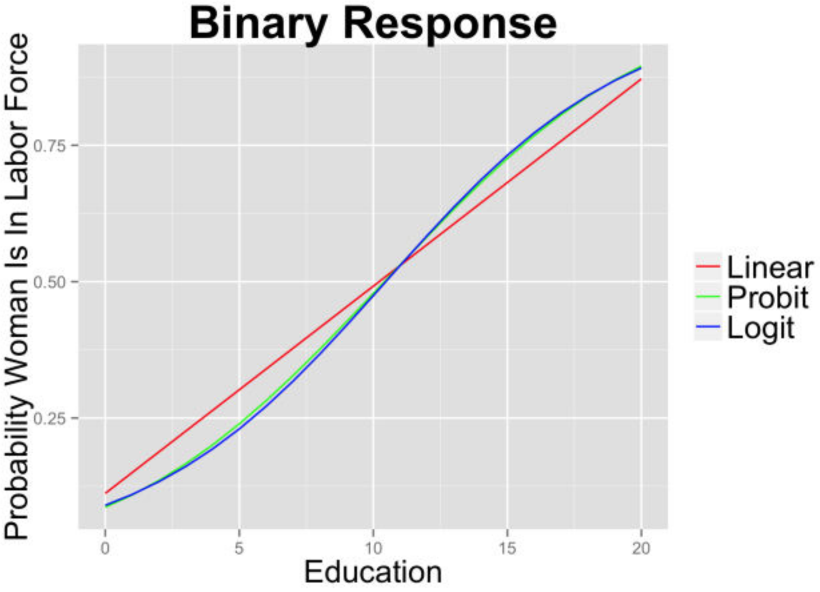
# Example: Predicting Whether a Woman Works for Pay

- Whether a woman works is binary: it’s either yes or no
- What things might affect whether a woman works for pay?
    - If she has young children, that might make her less likely to work
    - If she’s more educated, she might want to work (didn’t spend all that time in school if she didn’t want to use it)

Georgia Tech

# Logistic Regression vs. Linear Regression

- Logistic regression is used often in practice, but often many models give similar outputs



# Lesson Summary

- Logistic regression is useful for binary outcome, as you can increase likelihood of a variable without getting to 100%
- A probit is another regression option for binary outcomes, based on a normal distribution assumption