Linear Models Lecture 9: Probit

Robert Gulotty

University of Chicago

February 25, 2026

Binary Outcomes and the BLP

Recall from Lecture 2, the **best linear predictor** (BLP):

$$\beta = \arg\min_b \mathbb{E}\left[(Y - X'b)^2\right]$$

What happens when $Y \in \{0, 1\}$?
The conditional expectation function is now a probability:

$$m(x) = \mathbb{E}[Y \mid X = x] = P(Y = 1 \mid X = x)$$

The BLP still exists — OLS estimates β by solving the same normal equations:

$$\sum_{i=1}^{n} X_i(Y_i - X_i'\hat{\beta}) = 0$$

This is the **linear probability model** (LPM).

Binary Outcomes and the BLP

Recall from Lecture 2, the **best linear predictor** (BLP):

$$\beta = \arg\min_b \mathbb{E}\left[(Y - X'b)^2\right]$$

What happens when $Y \in \{0,1\}$?

The conditional expectation function is now a probability:

$$m(x) = \mathbb{E}[Y \mid X = x] = P(Y = 1 \mid X = x)$$

The BLP still exists — OLS estimates β by solving the same normal equations:

$$\sum_{i=1}^{n} X_i(Y_i - X_i'\hat{\beta}) = 0$$

This is the **linear probability model** (LPM).

Binary Outcomes and the BLP

Recall from Lecture 2, the **best linear predictor** (BLP):

$$\beta = \arg\min_b \mathbb{E}\left[(Y - X'b)^2\right]$$

What happens when $Y \in \{0, 1\}$?

The conditional expectation function is now a probability:

$$m(x) = \mathbb{E}[Y \mid X = x] = P(Y = 1 \mid X = x)$$

The BLP still exists — OLS estimates β by solving the same normal equations:

$$\sum_{i=1}^{n} X_i(Y_i - X_i'\hat{\beta}) = 0$$

This is the **linear probability model** (LPM).

Binary Outcomes and the BLP

Recall from Lecture 2, the **best linear predictor** (BLP):

$$\beta = \arg\min_b \mathbb{E}\left[(Y - X'b)^2\right]$$

What happens when $Y \in \{0, 1\}$?

The conditional expectation function is now a probability:

$$m(x) = \mathbb{E}[Y \mid X = x] = P(Y = 1 \mid X = x)$$

The BLP still exists — OLS estimates β by solving the same normal equations:

$$\sum_{i=1}^{n} X_i(Y_i - X_i'\hat{\beta}) = 0$$

This is the **linear probability model** (LPM).

LPM: Consistent for Average Marginal Effects

Even if the true CEF $P(Y = 1 \mid X)$ is nonlinear, OLS estimates a useful object.

Recall the BLP–CEF distinction from Lecture 2:

- The BLP $X'\beta$ need not equal the CEF $m(x)$
- But β is consistent for the **average marginal effect** under mild conditions

Built-in heteroskedasticity (recall Lecture 6):

If $Y_i \in \{0, 1\}$, then

$$\mathrm{Var}(Y_i \mid X_i) = p(X_i)(1 - p(X_i))$$

where $p(X_i) = P(Y_i = 1 \mid X_i)$.

The variance depends on X_i by construction. The LPM **always** requires heteroskedasticity-robust standard errors (HC2).

LPM: Consistent for Average Marginal Effects

Even if the true CEF $P(Y = 1 \mid X)$ is nonlinear, OLS estimates a useful object.
Recall the BLP–CEF distinction from Lecture 2:

- The BLP $X'\beta$ need not equal the CEF $m(x)$
- But β is consistent for the **average marginal effect** under mild conditions

Built-in heteroskedasticity (recall Lecture 6):
If $Y_i \in \{0, 1\}$, then

$$\text{Var}(Y_i \mid X_i) = p(X_i)(1 - p(X_i))$$

where $p(X_i) = P(Y_i = 1 \mid X_i)$.
The variance depends on X_i by construction. The LPM **always** requires heteroskedasticity-robust standard errors (HC2).

LPM: Consistent for Average Marginal Effects

Even if the true CEF $P(Y = 1 \mid X)$ is nonlinear, OLS estimates a useful object.
Recall the BLP–CEF distinction from Lecture 2:

- The BLP $X'\beta$ need not equal the CEF $m(x)$
- But β is consistent for the **average marginal effect** under mild conditions

Built-in heteroskedasticity (recall Lecture 6):
If $Y_i \in \{0, 1\}$, then

$$\mathrm{Var}(Y_i \mid X_i) = p(X_i)(1 - p(X_i))$$

where $p(X_i) = P(Y_i = 1 \mid X_i)$.
The variance depends on X_i by construction. The LPM **always** requires heteroskedasticity-robust standard errors (HC2).

LPM: Consistent for Average Marginal Effects

Even if the true CEF $P(Y = 1 \mid X)$ is nonlinear, OLS estimates a useful object.
Recall the BLP–CEF distinction from Lecture 2:

- The BLP $X'\beta$ need not equal the CEF $m(x)$
- But β is consistent for the **average marginal effect** under mild conditions

Built-in heteroskedasticity (recall Lecture 6):
If $Y_i \in \{0, 1\}$, then

$$\text{Var}(Y_i \mid X_i) = p(X_i)(1 - p(X_i))$$

where $p(X_i) = P(Y_i = 1 \mid X_i)$.
The variance depends on X_i by construction. The LPM **always** requires heteroskedasticity-robust standard errors (HC2).

Limitations of the Linear Probability Model

Problems:

- Predictions outside $[0, 1]$
- Poor approximation in tails
- Marginal effects constant (may be unrealistic)

Motivation: A model that respects the $[0, 1]$ constraint.

Single-index model:

$$P(Y = 1 \mid X) = G(X'\beta)$$

where $G : \mathbb{R} \to [0, 1]$ is a known link function.

Limitations of the Linear Probability Model

Problems:

- Predictions outside $[0, 1]$
- Poor approximation in tails
- Marginal effects constant (may be unrealistic)

Motivation: A model that respects the $[0, 1]$ constraint.

Single-index model:

$$P(Y = 1 \mid X) = G(X'\beta)$$

where $G : \mathbb{R} \to [0, 1]$ is a known link function.

Limitations of the Linear Probability Model

Problems:

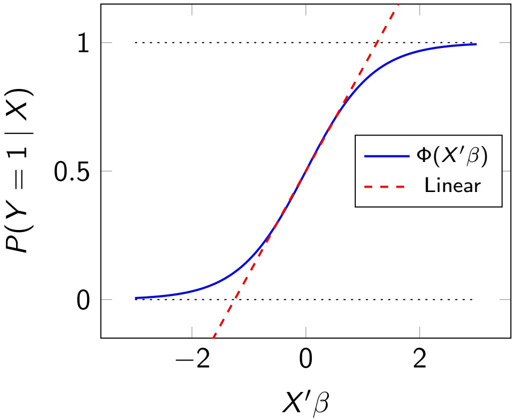
- Predictions outside $[0, 1]$
- Poor approximation in tails
- Marginal effects constant (may be unrealistic)

Motivation: A model that respects the $[0, 1]$ constraint.

Single-index model:

$$P(Y = 1 \mid X) = G(X'\beta)$$

where $G : \mathbb{R} \to [0, 1]$ is a known link function.



From LPM to Probit

Two common link functions:

Probit: $G = \Phi$ (standard normal CDF)
Logit: $G = \Lambda$ (logistic CDF: $e^x/(1+e^x)$)

Both map $\mathbb{R} \to (0,1)$, are symmetric about $1/2$, and produce nearly identical fitted values.

Why Probit?

- Natural latent variable interpretation ($\varepsilon \sim N(0,1)$)
- Connects to the normal distribution theory from Lectures 1 and 7
- Extends naturally to measurement models (IRT, later today)

From LPM to Probit

Two common link functions:

Probit: $G = \Phi$ (standard normal CDF)
Logit: $G = \Lambda$ (logistic CDF: $e^x/(1+e^x)$)

Both map $\mathbb{R} \to (0,1)$, are symmetric about $1/2$, and produce nearly identical fitted values.

Why Probit?

- Natural latent variable interpretation ($\varepsilon \sim N(0,1)$)
- Connects to the normal distribution theory from Lectures 1 and 7
- Extends naturally to measurement models (IRT, later today)

From LPM to Probit

Two common link functions:

Probit: $G = \Phi$ (standard normal CDF)
Logit: $G = \Lambda$ (logistic CDF: $e^x/(1+e^x)$)

Both map $\mathbb{R} \to (0,1)$, are symmetric about $1/2$, and produce nearly identical fitted values.

Why Probit?

- Natural latent variable interpretation ($\varepsilon \sim N(0,1)$)
- Connects to the normal distribution theory from Lectures 1 and 7
- Extends naturally to measurement models (IRT, later today)

Latent Variable Representation

Suppose there exists an unobserved variable:

$$Y_i^* = X_i'\beta + \varepsilon_i$$

Observed outcome:

$$Y_i = \mathbf{1}\{Y_i^* > 0\}$$

Assume:

$$\varepsilon_i \sim N(0,1)$$

Then:

$$P(Y_i = 1 \mid X_i) = P(\varepsilon_i > -X_i'\beta) = \Phi(X_i'\beta)$$

This is the **Probit model**.

Latent Variable Representation

Suppose there exists an unobserved variable:

$$Y_i^* = X_i'\beta + \varepsilon_i$$

Observed outcome:

$$Y_i = \mathbf{1}\{Y_i^* > 0\}$$

Assume:

$$\varepsilon_i \sim N(0,1)$$

Then:

$$P(Y_i = 1 \mid X_i) = P(\varepsilon_i > -X_i'\beta) = \Phi(X_i'\beta)$$

This is the **Probit model**.

Latent Variable Representation

Suppose there exists an unobserved variable:

$$Y_i^* = X_i'\beta + \varepsilon_i$$

Observed outcome:

$$Y_i = \mathbf{1}\{Y_i^* > 0\}$$

Assume:

$$\varepsilon_i \sim N(0, 1)$$

Then:

$$P(Y_i = 1 \mid X_i) = P(\varepsilon_i > -X_i'\beta) = \Phi(X_i'\beta)$$

This is the **Probit model**.

Latent Variable Representation

Suppose there exists an unobserved variable:

$$Y_i^* = X_i'\beta + \varepsilon_i$$

Observed outcome:

$$Y_i = \mathbf{1}\{Y_i^* > 0\}$$

Assume:

$$\varepsilon_i \sim N(0, 1)$$

Then:

$$P(Y_i = 1 \mid X_i) = P(\varepsilon_i > -X_i'\beta) = \Phi(X_i'\beta)$$

This is the **Probit model**.

Latent Variable Representation

Suppose there exists an unobserved variable:

$$Y_i^* = X_i'\beta + \varepsilon_i$$

Observed outcome:

$$Y_i = \mathbf{1}\{Y_i^* > 0\}$$

Assume:

$$\varepsilon_i \sim N(0,1)$$

Then:

$$P(Y_i = 1 \mid X_i) = P(\varepsilon_i > -X_i'\beta) = \Phi(X_i'\beta)$$

This is the **Probit model**.

Identification and Scale Normalization

In the latent model:

$$Y_i^* = X_i'\beta + \varepsilon_i, \qquad \varepsilon_i \sim N(0, \sigma^2)$$

Then:

$$P(Y_i = 1 \mid X_i) = \Phi\left(\frac{X_i'\beta}{\sigma}\right)$$

Only β/σ is identified — the data cannot distinguish (β, σ) from $(c\beta, c\sigma)$.
Normalization: Set $\text{Var}(\varepsilon_i) = 1$.

Key Point

Probit coefficients are identified only up to scale. We cannot interpret β_j the same way as in OLS. This is fundamentally different from the linear model.

Identification and Scale Normalization

In the latent model:

$$Y_i^* = X_i'\beta + \varepsilon_i, \qquad \varepsilon_i \sim N(0, \sigma^2)$$

Then:

$$P(Y_i = 1 \mid X_i) = \Phi\left(\frac{X_i'\beta}{\sigma}\right)$$

Only β/σ is identified — the data cannot distinguish (β, σ) from $(c\beta, c\sigma)$.
Normalization: Set $\text{Var}(\varepsilon_i) = 1$.

Key Point

Probit coefficients are identified only up to scale. We cannot interpret β_j the same way as in OLS. This is fundamentally different from the linear model.

Identification and Scale Normalization

In the latent model:

$$Y_i^* = X_i'\beta + \varepsilon_i, \qquad \varepsilon_i \sim N(0, \sigma^2)$$

Then:

$$P(Y_i = 1 \mid X_i) = \Phi\left(\frac{X_i'\beta}{\sigma}\right)$$

Only β/σ is identified — the data cannot distinguish (β, σ) from $(c\beta, c\sigma)$.

Normalization: Set $\mathrm{Var}(\varepsilon_i) = 1$.

Key Point

Probit coefficients are identified only up to scale. We cannot interpret β_j the same way as in OLS. This is fundamentally different from the linear model.

Identification and Scale Normalization

In the latent model:

$$Y_i^* = X_i'\beta + \varepsilon_i, \qquad \varepsilon_i \sim N(0, \sigma^2)$$

Then:

$$P(Y_i = 1 \mid X_i) = \Phi\left(\frac{X_i'\beta}{\sigma}\right)$$

Only β/σ is identified — the data cannot distinguish (β, σ) from $(c\beta, c\sigma)$.
Normalization: Set $\text{Var}(\varepsilon_i) = 1$.

Key Point

Probit coefficients are identified only up to scale. We cannot interpret β_j the same way as in OLS. This is fundamentally different from the linear model.

Identification and Scale Normalization

In the latent model:

$$Y_i^* = X_i'\beta + \varepsilon_i, \qquad \varepsilon_i \sim N(0, \sigma^2)$$

Then:

$$P(Y_i = 1 \mid X_i) = \Phi\left(\frac{X_i'\beta}{\sigma}\right)$$

Only β/σ is identified — the data cannot distinguish (β, σ) from $(c\beta, c\sigma)$.
Normalization: Set $\text{Var}(\varepsilon_i) = 1$.

Key Point

Probit coefficients are identified only up to scale. We cannot interpret β_j the same way as in OLS. This is fundamentally different from the linear model.

Marginal Effects

In the linear model: $\partial \mathbb{E}[Y \mid X]/\partial x_j = \beta_j$.

In the Probit model:

$$\frac{\partial P(Y = 1 \mid X)}{\partial x_j} = \phi(X'\beta) \cdot \beta_j$$

The marginal effect **depends on** X through $\phi(X'\beta)$.

Average Marginal Effect (AME):

$$\widehat{\text{AME}}_j = \frac{1}{n} \sum_{i=1}^{n} \phi(X_i'\hat{\beta})\,\hat{\beta}_j$$

- The AME averages over the observed distribution of X
- This is the natural analog of the OLS coefficient for binary outcomes
- Computing the AME requires the **delta method** (Lecture 10) for standard errors

Marginal Effects

In the linear model: $\partial\mathbb{E}[Y \mid X]/\partial x_j = \beta_j$.

In the Probit model:

$$\frac{\partial P(Y=1 \mid X)}{\partial x_j} = \phi(X'\beta) \cdot \beta_j$$

The marginal effect **depends on** X through $\phi(X'\beta)$.

Average Marginal Effect (AME):

$$\widehat{\text{AME}}_j = \frac{1}{n}\sum_{i=1}^{n} \phi(X_i'\hat{\beta})\,\hat{\beta}_j$$

- The AME averages over the observed distribution of X
- This is the natural analog of the OLS coefficient for binary outcomes
- Computing the AME requires the **delta method** (Lecture 10) for standard errors

Marginal Effects

In the linear model: $\partial \mathbb{E}[Y \mid X] / \partial x_j = \beta_j$.

In the Probit model:

$$\frac{\partial P(Y = 1 \mid X)}{\partial x_j} = \phi(X'\beta) \cdot \beta_j$$

The marginal effect **depends on** X through $\phi(X'\beta)$.

Average Marginal Effect (AME):

$$\widehat{\text{AME}}_j = \frac{1}{n} \sum_{i=1}^{n} \phi(X_i'\hat{\beta})\, \hat{\beta}_j$$

- The AME averages over the observed distribution of X
- This is the natural analog of the OLS coefficient for binary outcomes
- Computing the AME requires the **delta method** (Lecture 10) for standard errors

Marginal Effects

In the linear model: $\partial \mathbb{E}[Y \mid X]/\partial x_j = \beta_j$.
In the Probit model:

$$\frac{\partial P(Y = 1 \mid X)}{\partial x_j} = \phi(X'\beta) \cdot \beta_j$$

The marginal effect **depends on** X through $\phi(X'\beta)$.

Average Marginal Effect (AME):

$$\widehat{\mathrm{AME}}_j = \frac{1}{n} \sum_{i=1}^{n} \phi(X_i'\hat{\beta})\, \hat{\beta}_j$$

- The AME averages over the observed distribution of X
- This is the natural analog of the OLS coefficient for binary outcomes
- Computing the AME requires the **delta method** (Lecture 10) for standard errors

Marginal Effects

In the linear model: $\partial \mathbb{E}[Y \mid X]/\partial x_j = \beta_j$.

In the Probit model:

$$\frac{\partial P(Y = 1 \mid X)}{\partial x_j} = \phi(X'\beta) \cdot \beta_j$$

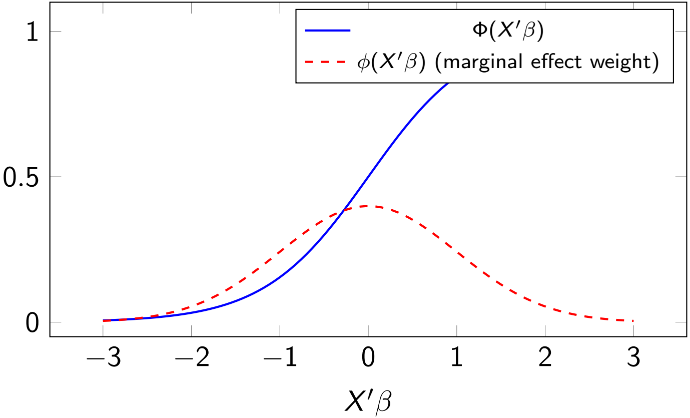
The marginal effect **depends on** X through $\phi(X'\beta)$.

Average Marginal Effect (AME):

$$\widehat{\mathrm{AME}}_j = \frac{1}{n} \sum_{i=1}^{n} \phi(X_i'\hat{\beta})\, \hat{\beta}_j$$

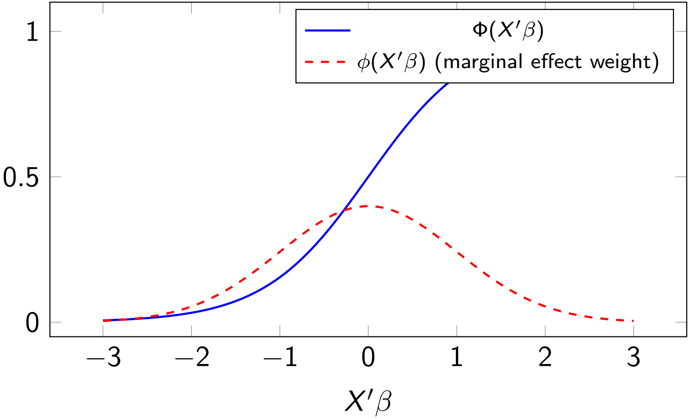
- The AME averages over the observed distribution of X
- This is the natural analog of the OLS coefficient for binary outcomes
- Computing the AME requires the **delta method** (Lecture 10) for standard errors

Marginal Effects: Graphical Intuition



Marginal effects are largest near $X'\beta = 0$ (where the CDF is steepest) and vanish in the tails.

Marginal Effects: Graphical Intuition



Marginal effects are largest near $X'\beta = 0$ (where the CDF is steepest) and vanish in the tails.

Likelihood for the Probit Model

Recall from Lecture 7: we developed the general MLE framework under normality. Now we apply it to a **nonlinear** model.

Assuming conditional independence, the likelihood is:

$$L_n(\beta) = \prod_{i=1}^{n} \Phi(X_i'\beta)^{Y_i} \left(1 - \Phi(X_i'\beta)\right)^{1-Y_i}$$

Log-likelihood:

$$\ell_n(\beta) = \sum_{i=1}^{n} \left[Y_i \log \Phi(X_i'\beta) + (1 - Y_i) \log \left(1 - \Phi(X_i'\beta)\right) \right]$$

We estimate $\hat{\beta}_{\text{MLE}} = \arg\max_{\beta} \ell_n(\beta)$.

Likelihood for the Probit Model

Recall from Lecture 7: we developed the general MLE framework under normality. Now we apply it to a **nonlinear** model.

Assuming conditional independence, the likelihood is:

$$L_n(\beta) = \prod_{i=1}^{n} \Phi(X_i'\beta)^{Y_i} \left(1 - \Phi(X_i'\beta)\right)^{1-Y_i}$$

Log-likelihood:

$$\ell_n(\beta) = \sum_{i=1}^{n} \left[Y_i \log \Phi(X_i'\beta) + (1 - Y_i) \log \left(1 - \Phi(X_i'\beta)\right) \right]$$

We estimate $\hat{\beta}_{\text{MLE}} = \arg\max_\beta \ell_n(\beta)$.

Likelihood for the Probit Model

Recall from Lecture 7: we developed the general MLE framework under normality. Now we apply it to a **nonlinear** model.

Assuming conditional independence, the likelihood is:

$$L_n(\beta) = \prod_{i=1}^{n} \Phi(X_i'\beta)^{Y_i} \left(1 - \Phi(X_i'\beta)\right)^{1-Y_i}$$

Log-likelihood:

$$\ell_n(\beta) = \sum_{i=1}^{n} \left[Y_i \log \Phi(X_i'\beta) + (1 - Y_i) \log \left(1 - \Phi(X_i'\beta)\right)\right]$$

We estimate $\hat{\beta}_{\text{MLE}} = \arg\max_\beta \ell_n(\beta)$.

Likelihood for the Probit Model

Recall from Lecture 7: we developed the general MLE framework under normality. Now we apply it to a **nonlinear** model.

Assuming conditional independence, the likelihood is:

$$L_n(\beta) = \prod_{i=1}^{n} \Phi(X_i'\beta)^{Y_i} \left(1 - \Phi(X_i'\beta)\right)^{1-Y_i}$$

Log-likelihood:

$$\ell_n(\beta) = \sum_{i=1}^{n} \left[Y_i \log \Phi(X_i'\beta) + (1 - Y_i) \log \left(1 - \Phi(X_i'\beta)\right)\right]$$

We estimate $\hat{\beta}_{\text{MLE}} = \arg\max_{\beta} \ell_n(\beta)$.

Score Function

Let $\phi(\cdot)$ denote the standard normal pdf, $\Phi(\cdot)$ the CDF.

The individual score contribution:

$$s_i(\beta) = \frac{\partial \ell_i}{\partial \beta} = \frac{\phi(X_i'\beta)}{\Phi(X_i'\beta)(1-\Phi(X_i'\beta))}\left(Y_i - \Phi(X_i'\beta)\right) X_i$$

The total score:

$$S_n(\beta) = \sum_{i=1}^{n} s_i(\beta) = \sum_{i=1}^{n} w_i \left(Y_i - \Phi(X_i'\beta)\right) X_i$$

where $w_i = \phi(X_i'\beta)/[\Phi(X_i'\beta)(1-\Phi(X_i'\beta))]$.

Setting $S_n(\hat{\beta}) = 0$ defines the MLE. This is a **nonlinear** system — no closed-form solution.

Score Function

Let $\phi(\cdot)$ denote the standard normal pdf, $\Phi(\cdot)$ the CDF.
The individual score contribution:

$$s_i(\beta) = \frac{\partial \ell_i}{\partial \beta} = \frac{\phi(X_i'\beta)}{\Phi(X_i'\beta)(1 - \Phi(X_i'\beta))} \left(Y_i - \Phi(X_i'\beta)\right) X_i$$

The total score:

$$S_n(\beta) = \sum_{i=1}^{n} s_i(\beta) = \sum_{i=1}^{n} w_i \left(Y_i - \Phi(X_i'\beta)\right) X_i$$

where $w_i = \phi(X_i'\beta)/[\Phi(X_i'\beta)(1 - \Phi(X_i'\beta))]$.

Setting $S_n(\hat{\beta}) = 0$ defines the MLE. This is a **nonlinear** system — no closed-form solution.

Score Function

Let $\phi(\cdot)$ denote the standard normal pdf, $\Phi(\cdot)$ the CDF.
The individual score contribution:

$$s_i(\beta) = \frac{\partial \ell_i}{\partial \beta} = \frac{\phi(X_i'\beta)}{\Phi(X_i'\beta)(1 - \Phi(X_i'\beta))} \left(Y_i - \Phi(X_i'\beta)\right) X_i$$

The total score:

$$S_n(\beta) = \sum_{i=1}^{n} s_i(\beta) = \sum_{i=1}^{n} w_i \left(Y_i - \Phi(X_i'\beta)\right) X_i$$

where $w_i = \phi(X_i'\beta)/[\Phi(X_i'\beta)(1 - \Phi(X_i'\beta))]$.

Setting $S_n(\hat{\beta}) = 0$ defines the MLE. This is a **nonlinear** system — no closed-form solution.

Score Function

Let $\phi(\cdot)$ denote the standard normal pdf, $\Phi(\cdot)$ the CDF.
The individual score contribution:

$$s_i(\beta) = \frac{\partial \ell_i}{\partial \beta} = \frac{\phi(X_i'\beta)}{\Phi(X_i'\beta)(1 - \Phi(X_i'\beta))} \left(Y_i - \Phi(X_i'\beta)\right) X_i$$

The total score:

$$S_n(\beta) = \sum_{i=1}^{n} s_i(\beta) = \sum_{i=1}^{n} w_i \left(Y_i - \Phi(X_i'\beta)\right) X_i$$

where $w_i = \phi(X_i'\beta)/[\Phi(X_i'\beta)(1 - \Phi(X_i'\beta))]$.

Setting $S_n(\hat{\beta}) = 0$ defines the MLE. This is a **nonlinear** system — no closed-form solution.

Score as Moment Condition

Key insight: Under correct specification, $\mathbb{E}[s_i(\beta_0)] = 0$.

Compare three estimating equations side by side:

Model	**Estimating Equation**	**Weights**
OLS	$\sum_i X_i(Y_i - X_i'\beta) = 0$	Equal
Probit MLE	$\sum_i w_i(Y_i - \Phi(X_i'\beta))\,X_i = 0$	$w_i = \frac{\phi}{\Phi(1-\Phi)}$
General	$\sum_i g_i(\theta) = 0$	$\to$ GMM

All three are **sample moment conditions**. GMM (coming later) is the general framework:

Score as Moment Condition

Key insight: Under correct specification, $\mathbb{E}[s_i(\beta_0)] = 0$.

Compare three estimating equations side by side:

Model	Estimating Equation	Weights
OLS	$\sum_i X_i(Y_i - X_i'\beta) = 0$	Equal
Probit MLE	$\sum_i w_i(Y_i - \Phi(X_i'\beta))\,X_i = 0$	$w_i = \frac{\phi}{\Phi(1-\Phi)}$
General	$\sum_i g_i(\theta) = 0$	$\rightarrow$ GMM

All three are **sample moment conditions**. GMM (coming later) is the general framework:

Score as Moment Condition

Key insight: Under correct specification, $\mathbb{E}[s_i(\beta_0)] = 0$.

Compare three estimating equations side by side:

Model	Estimating Equation	Weights
OLS	$\sum_i X_i(Y_i - X_i'\beta) = 0$	Equal
Probit MLE	$\sum_i w_i(Y_i - \Phi(X_i'\beta))\,X_i = 0$	$w_i = \frac{\phi}{\Phi(1-\Phi)}$
General	$\sum_i g_i(\theta) = 0$	$\rightarrow$ GMM

All three are **sample moment conditions**. GMM (coming later) is the general framework: find θ such that $\frac{1}{n}\sum_i g(W_i, \theta) = 0$

Iterative Estimation: Newton–Raphson

Unlike OLS, the Probit MLE has no closed-form solution.

Newton–Raphson iterates:

$$\beta^{(t+1)} = \beta^{(t)} - \left[H_n(\beta^{(t)})\right]^{-1} S_n(\beta^{(t)})$$

where $H_n(\beta) = \partial S_n / \partial \beta' = \partial^2 \ell_n / \partial \beta \partial \beta'$ is the Hessian.

Algorithm:

1. Start with an initial guess $\beta^{(0)}$ (e.g., OLS estimates)
2. Compute score $S_n(\beta^{(t)})$ and Hessian $H_n(\beta^{(t)})$
3. Update: $\beta^{(t+1)} = \beta^{(t)} - H_n^{-1} S_n$
4. Repeat until convergence: $\|\beta^{(t+1)} - \beta^{(t)}\| < \varepsilon$

In practice, `glm()` in R handles this automatically via Fisher scoring (a variant of Newton–Raphson).

Iterative Estimation: Newton–Raphson

Unlike OLS, the Probit MLE has no closed-form solution.

Newton–Raphson iterates:

$$\beta^{(t+1)} = \beta^{(t)} - \left[H_n(\beta^{(t)})\right]^{-1} S_n(\beta^{(t)})$$

where $H_n(\beta) = \partial S_n/\partial\beta' = \partial^2\ell_n/\partial\beta\partial\beta'$ is the Hessian.

Algorithm:

1. Start with an initial guess $\beta^{(0)}$ (e.g., OLS estimates)
2. Compute score $S_n(\beta^{(t)})$ and Hessian $H_n(\beta^{(t)})$
3. Update: $\beta^{(t+1)} = \beta^{(t)} - H_n^{-1} S_n$
4. Repeat until convergence: $\|\beta^{(t+1)} - \beta^{(t)}\| < \varepsilon$

In practice, `glm()` in R handles this automatically via Fisher scoring (a variant of Newton–Raphson).

Iterative Estimation: Newton–Raphson

Unlike OLS, the Probit MLE has no closed-form solution.

Newton–Raphson iterates:

$$\beta^{(t+1)} = \beta^{(t)} - \left[H_n(\beta^{(t)})\right]^{-1} S_n(\beta^{(t)})$$

where $H_n(\beta) = \partial S_n / \partial \beta' = \partial^2 \ell_n / \partial \beta \partial \beta'$ is the Hessian.

Algorithm:

1. Start with an initial guess $\beta^{(0)}$ (e.g., OLS estimates)
2. Compute score $S_n(\beta^{(t)})$ and Hessian $H_n(\beta^{(t)})$
3. Update: $\beta^{(t+1)} = \beta^{(t)} - H_n^{-1} S_n$
4. Repeat until convergence: $\|\beta^{(t+1)} - \beta^{(t)}\| < \varepsilon$

In practice, `glm()` in R handles this automatically via Fisher scoring (a variant of Newton–Raphson).

Iterative Estimation: Newton–Raphson

Unlike OLS, the Probit MLE has no closed-form solution.

Newton–Raphson iterates:

$$\beta^{(t+1)} = \beta^{(t)} - \left[H_n(\beta^{(t)})\right]^{-1} S_n(\beta^{(t)})$$

where $H_n(\beta) = \partial S_n / \partial \beta' = \partial^2 \ell_n / \partial \beta \partial \beta'$ is the Hessian.

Algorithm:

1. Start with an initial guess $\beta^{(0)}$ (e.g., OLS estimates)
2. Compute score $S_n(\beta^{(t)})$ and Hessian $H_n(\beta^{(t)})$
3. Update: $\beta^{(t+1)} = \beta^{(t)} - H_n^{-1} S_n$
4. Repeat until convergence: $\|\beta^{(t+1)} - \beta^{(t)}\| < \varepsilon$

In practice, `glm()` in R handles this automatically via Fisher scoring (a variant of Newton–Raphson).

Why Not Just Use OLS?

	LPM (OLS)	**Probit (MLE)**
Closed-form solution	Yes	No
Predictions in $[0, 1]$	Not guaranteed	Yes
Consistent for β	—	Yes (if correctly specified)
Consistent for AME	Yes (always)	Yes (if correctly specified)
Robust to misspecification	Yes (BLP always exists)	Requires sandwich SEs
Efficiency	Lower	Higher (if correct)

Practical guidance: LPM is a robust baseline. Probit gains efficiency if the model is correct, but misspecification has consequences.

Why Not Just Use OLS?

	LPM (OLS)	Probit (MLE)
Closed-form solution	Yes	No
Predictions in $[0, 1]$	Not guaranteed	Yes
Consistent for β	—	Yes (if correctly specified)
Consistent for AME	Yes (always)	Yes (if correctly specified)
Robust to misspecification	Yes (BLP always exists)	Requires sandwich SEs
Efficiency	Lower	Higher (if correct)

Practical guidance: LPM is a robust baseline. Probit gains efficiency if the model is correct, but misspecification has consequences.

Fisher Information for Probit

The Fisher information matrix is:

$$\mathcal{I}(\beta) = \mathbb{E}\left[\frac{\phi(X'\beta)^2}{\Phi(X'\beta)(1-\Phi(X'\beta))} XX'\right]$$

Derivation: By the information matrix equality,

$$\mathcal{I}(\beta) = \mathbb{E}[s_i(\beta)s_i(\beta)'] = -\mathbb{E}\left[\frac{\partial^2 \ell_i(\beta)}{\partial\beta\partial\beta'}\right]$$

The weight $\frac{\phi(X'\beta)^2}{\Phi(X'\beta)(1-\Phi(X'\beta))}$ is largest when $X'\beta \approx 0$ (where we have the most "information" about β) and smallest in the tails.

Compare with OLS: $\mathbb{E}[XX']/\sigma^2$. In Probit, the effective "variance" changes with X.

Fisher Information for Probit

The Fisher information matrix is:

$$\mathcal{I}(\beta) = \mathbb{E}\left[\frac{\phi(X'\beta)^2}{\Phi(X'\beta)(1-\Phi(X'\beta))} XX'\right]$$

Derivation: By the information matrix equality,

$$\mathcal{I}(\beta) = \mathbb{E}[s_i(\beta)s_i(\beta)'] = -\mathbb{E}\left[\frac{\partial^2 \ell_i(\beta)}{\partial\beta\partial\beta'}\right]$$

The weight $\frac{\phi(X'\beta)^2}{\Phi(X'\beta)(1-\Phi(X'\beta))}$ is largest when $X'\beta \approx 0$ (where we have the most "information" about β) and smallest in the tails.

Compare with OLS: $\mathbb{E}[XX']/\sigma^2$. In Probit, the effective "variance" changes with X.

Fisher Information for Probit

The Fisher information matrix is:

$$\mathcal{I}(\beta) = \mathbb{E}\left[\frac{\phi(X'\beta)^2}{\Phi(X'\beta)(1-\Phi(X'\beta))} XX'\right]$$

Derivation: By the information matrix equality,

$$\mathcal{I}(\beta) = \mathbb{E}[s_i(\beta)s_i(\beta)'] = -\mathbb{E}\left[\frac{\partial^2 \ell_i(\beta)}{\partial\beta\partial\beta'}\right]$$

The weight $\frac{\phi(X'\beta)^2}{\Phi(X'\beta)(1-\Phi(X'\beta))}$ is largest when $X'\beta \approx 0$ (where we have the most "information" about β) and smallest in the tails.

Compare with OLS: $\mathbb{E}[XX']/\sigma^2$. In Probit, the effective "variance" changes with X.

Fisher Information for Probit

The Fisher information matrix is:

$$\mathcal{I}(\beta) = \mathbb{E}\left[\frac{\phi(X'\beta)^2}{\Phi(X'\beta)(1-\Phi(X'\beta))} XX'\right]$$

Derivation: By the information matrix equality,

$$\mathcal{I}(\beta) = \mathbb{E}[s_i(\beta)s_i(\beta)'] = -\mathbb{E}\left[\frac{\partial^2 \ell_i(\beta)}{\partial\beta\partial\beta'}\right]$$

The weight $\frac{\phi(X'\beta)^2}{\Phi(X'\beta)(1-\Phi(X'\beta))}$ is largest when $X'\beta \approx 0$ (where we have the most "information" about β) and smallest in the tails.

Compare with OLS: $\mathbb{E}[XX']/\sigma^2$. In Probit, the effective "variance" changes with X.

Hessian and Information Matrix Equality

The expected Hessian can be computed by differentiating the score:

$$-\mathbb{E}\left[\frac{\partial^2 \ell_i}{\partial\beta\partial\beta'}\right] = \mathbb{E}\left[\frac{\phi(X'\beta)^2}{\Phi(X'\beta)(1-\Phi(X'\beta))}XX'\right]$$

Meanwhile, the outer product of scores:

$$\mathbb{E}[s_i(\beta_0)s_i(\beta_0)'] = \mathbb{E}\left[\frac{\phi(X'\beta)^2}{\Phi(X'\beta)^2(1-\Phi(X'\beta))^2}\underbrace{(Y_i-\Phi)^2}_{\Phi(1-\Phi)}XX'\right]$$

Using $\mathbb{E}[(Y_i - \Phi(X_i'\beta))^2 \mid X_i] = \Phi(X_i'\beta)(1-\Phi(X_i'\beta))$, one factor cancels.

Information Matrix Equality

Under **correct specification**: $\mathbb{E}[s_i s_i'] = -\mathbb{E}\left[\frac{\partial^2 \ell_i}{\partial\beta\partial\beta'}\right]$. The outer product of scores equals the negative expected Hessian.

Hessian and Information Matrix Equality

The expected Hessian can be computed by differentiating the score:

$$-\mathbb{E}\left[\frac{\partial^2 \ell_i}{\partial\beta\partial\beta'}\right] = \mathbb{E}\left[\frac{\phi(X'\beta)^2}{\Phi(X'\beta)(1-\Phi(X'\beta))}XX'\right]$$

Meanwhile, the outer product of scores:

$$\mathbb{E}[s_i(\beta_0)s_i(\beta_0)'] = \mathbb{E}\left[\frac{\phi(X'\beta)^2}{\Phi(X'\beta)^2(1-\Phi(X'\beta))^2}\underbrace{(Y_i-\Phi)^2}_{\Phi(1-\Phi)}XX'\right]$$

Using $\mathbb{E}[(Y_i - \Phi(X_i'\beta))^2 \mid X_i] = \Phi(X_i'\beta)(1-\Phi(X_i'\beta))$, one factor cancels.

Information Matrix Equality

Under **correct specification**: $\mathbb{E}[s_i s_i'] = -\mathbb{E}\left[\frac{\partial^2 \ell_i}{\partial\beta\partial\beta'}\right]$. The outer product of scores equals the negative expected Hessian.

Hessian and Information Matrix Equality

The expected Hessian can be computed by differentiating the score:

$$-\mathbb{E}\left[\frac{\partial^2 \ell_i}{\partial\beta\partial\beta'}\right] = \mathbb{E}\left[\frac{\phi(X'\beta)^2}{\Phi(X'\beta)(1-\Phi(X'\beta))}XX'\right]$$

Meanwhile, the outer product of scores:

$$\mathbb{E}[s_i(\beta_0)s_i(\beta_0)'] = \mathbb{E}\left[\frac{\phi(X'\beta)^2}{\Phi(X'\beta)^2(1-\Phi(X'\beta))^2}\underbrace{(Y_i-\Phi)^2}_{\Phi(1-\Phi)}XX'\right]$$

Using $\mathbb{E}[(Y_i - \Phi(X_i'\beta))^2 \mid X_i] = \Phi(X_i'\beta)(1-\Phi(X_i'\beta))$, one factor cancels.

Information Matrix Equality

Under **correct specification**: $\mathbb{E}[s_i s_i'] = -\mathbb{E}\left[\frac{\partial^2 \ell_i}{\partial\beta\partial\beta'}\right]$. The outer product of scores equals the negative expected Hessian.

Asymptotic Normality: Derivation Sketch

Taylor-expand the score around β_0:

$$0 = S_n(\hat{\beta}) \approx S_n(\beta_0) + H_n(\beta_0)(\hat{\beta} - \beta_0)$$

Rearranging:

$$\sqrt{n}(\hat{\beta} - \beta_0) \approx \left[-\frac{1}{n}H_n(\beta_0)\right]^{-1} \frac{1}{\sqrt{n}} S_n(\beta_0)$$

By the law of large numbers: $\frac{1}{n}H_n(\beta_0) \xrightarrow{p} -\mathcal{I}(\beta_0)$.

By the CLT: $\frac{1}{\sqrt{n}}S_n(\beta_0) \xrightarrow{d} N(0, \mathcal{I}(\beta_0))$.

Combining:

$$\boxed{\sqrt{n}(\hat{\beta} - \beta_0) \xrightarrow{d} N(0,\ \mathcal{I}(\beta_0)^{-1})}$$

Same structure as Lecture 7, but now for a nonlinear model. The asymptotic tools (WLLN, CLT) will be developed formally in Lecture 10.

Asymptotic Normality: Derivation Sketch

Taylor-expand the score around β_0:

$$0 = S_n(\hat{\beta}) \approx S_n(\beta_0) + H_n(\beta_0)(\hat{\beta} - \beta_0)$$

Rearranging:

$$\sqrt{n}(\hat{\beta} - \beta_0) \approx \left[-\frac{1}{n} H_n(\beta_0)\right]^{-1} \frac{1}{\sqrt{n}} S_n(\beta_0)$$

By the law of large numbers: $\frac{1}{n} H_n(\beta_0) \xrightarrow{p} -\mathcal{I}(\beta_0)$.

By the CLT: $\frac{1}{\sqrt{n}} S_n(\beta_0) \xrightarrow{d} N(0, \mathcal{I}(\beta_0))$.

Combining:

$$\boxed{\sqrt{n}(\hat{\beta} - \beta_0) \xrightarrow{d} N(0,\ \mathcal{I}(\beta_0)^{-1})}$$

Same structure as Lecture 7, but now for a nonlinear model. The asymptotic tools (WLLN, CLT) will be developed formally in Lecture 10.

Asymptotic Normality: Derivation Sketch

Taylor-expand the score around β_0:

$$0 = S_n(\hat{\beta}) \approx S_n(\beta_0) + H_n(\beta_0)(\hat{\beta} - \beta_0)$$

Rearranging:

$$\sqrt{n}(\hat{\beta} - \beta_0) \approx \left[-\frac{1}{n}H_n(\beta_0)\right]^{-1} \frac{1}{\sqrt{n}} S_n(\beta_0)$$

By the law of large numbers: $\frac{1}{n}H_n(\beta_0) \xrightarrow{p} -\mathcal{I}(\beta_0)$.

By the CLT: $\frac{1}{\sqrt{n}}S_n(\beta_0) \xrightarrow{d} N(0, \mathcal{I}(\beta_0))$.

Combining:

$$\boxed{\sqrt{n}(\hat{\beta} - \beta_0) \xrightarrow{d} N(0,\ \mathcal{I}(\beta_0)^{-1})}$$

Same structure as Lecture 7, but now for a nonlinear model. The asymptotic tools (WLLN, CLT) will be developed formally in Lecture 10.

Asymptotic Normality: Derivation Sketch

Taylor-expand the score around β_0:

$$0 = S_n(\hat{\beta}) \approx S_n(\beta_0) + H_n(\beta_0)(\hat{\beta} - \beta_0)$$

Rearranging:

$$\sqrt{n}(\hat{\beta} - \beta_0) \approx \left[-\frac{1}{n} H_n(\beta_0)\right]^{-1} \frac{1}{\sqrt{n}} S_n(\beta_0)$$

By the law of large numbers: $\frac{1}{n} H_n(\beta_0) \xrightarrow{p} -\mathcal{I}(\beta_0)$.

By the CLT: $\frac{1}{\sqrt{n}} S_n(\beta_0) \xrightarrow{d} N(0, \mathcal{I}(\beta_0))$.

Combining:

$$\boxed{\sqrt{n}(\hat{\beta} - \beta_0) \xrightarrow{d} N(0,\ \mathcal{I}(\beta_0)^{-1})}$$

Same structure as Lecture 7, but now for a nonlinear model. The asymptotic tools (WLLN, CLT) will be developed formally in Lecture 10.

Asymptotic Normality: Derivation Sketch

Taylor-expand the score around β_0:

$$0 = S_n(\hat{\beta}) \approx S_n(\beta_0) + H_n(\beta_0)(\hat{\beta} - \beta_0)$$

Rearranging:

$$\sqrt{n}(\hat{\beta} - \beta_0) \approx \left[-\frac{1}{n} H_n(\beta_0)\right]^{-1} \frac{1}{\sqrt{n}} S_n(\beta_0)$$

By the law of large numbers: $\frac{1}{n} H_n(\beta_0) \xrightarrow{p} -\mathcal{I}(\beta_0)$.

By the CLT: $\frac{1}{\sqrt{n}} S_n(\beta_0) \xrightarrow{d} N(0, \mathcal{I}(\beta_0))$.

Combining:

$$\boxed{\sqrt{n}(\hat{\beta} - \beta_0) \xrightarrow{d} N\left(0,\ \mathcal{I}(\beta_0)^{-1}\right)}$$

Same structure as Lecture 7, but now for a nonlinear model. The asymptotic tools (WLLN, CLT) will be developed formally in Lecture 10.

Asymptotic Normality: Derivation Sketch

Taylor-expand the score around β_0:

$$0 = S_n(\hat{\beta}) \approx S_n(\beta_0) + H_n(\beta_0)(\hat{\beta} - \beta_0)$$

Rearranging:

$$\sqrt{n}(\hat{\beta} - \beta_0) \approx \left[-\frac{1}{n}H_n(\beta_0)\right]^{-1} \frac{1}{\sqrt{n}}S_n(\beta_0)$$

By the law of large numbers: $\frac{1}{n}H_n(\beta_0) \xrightarrow{p} -\mathcal{I}(\beta_0)$.

By the CLT: $\frac{1}{\sqrt{n}}S_n(\beta_0) \xrightarrow{d} N(0, \mathcal{I}(\beta_0))$.

Combining:

$$\boxed{\sqrt{n}(\hat{\beta} - \beta_0) \xrightarrow{d} N\left(0,\ \mathcal{I}(\beta_0)^{-1}\right)}$$

Same structure as Lecture 7, but now for a nonlinear model. The asymptotic tools (WLLN, CLT) will be developed formally in Lecture 10.

The Sandwich Under Misspecification

What if Φ is the wrong link function?

If the true CEF is $G_0(X'\beta)$ but we estimate Probit, the information matrix equality **fails**:

$$\mathbb{E}[s_i s_i'] \neq -\mathbb{E}\left[\frac{\partial^2 \ell_i}{\partial\beta\partial\beta'}\right]$$

The asymptotic variance becomes the **sandwich** (recall Lectures 5–6):

$$\text{Var}(\sqrt{n}(\hat{\beta} - \beta_0)) = \underbrace{H^{-1}}_{\text{bread}} \underbrace{\mathbb{E}[s_i s_i']}_{\text{meat}} \underbrace{H^{-1}}_{\text{bread}}$$

where $H = -\mathbb{E}[\partial^2 \ell_i / \partial\beta\partial\beta']$.

Quasi-MLE (QMLE): Even with the wrong G, the index $X'\beta$ can be consistently estimated (up to scale) — a form of semi-parametric resilience.

Practical implication: Use robust (sandwich) standard errors for Probit, just as you do for OLS. Same `sandwich` package in R.

The Sandwich Under Misspecification

What if Φ is the wrong link function?

If the true CEF is $G_0(X'\beta)$ but we estimate Probit, the information matrix equality **fails**:

$$\mathbb{E}[s_i s_i'] \neq -\mathbb{E}\left[\frac{\partial^2 \ell_i}{\partial\beta\partial\beta'}\right]$$

The asymptotic variance becomes the **sandwich** (recall Lectures 5–6):

$$\text{Var}(\sqrt{n}(\hat{\beta} - \beta_0)) = \underbrace{H^{-1}}_{\text{bread}} \underbrace{\mathbb{E}[s_i s_i']}_{\text{meat}} \underbrace{H^{-1}}_{\text{bread}}$$

where $H = -\mathbb{E}[\partial^2 \ell_i / \partial\beta\partial\beta']$.

Quasi-MLE (QMLE): Even with the wrong G, the index $X'\beta$ can be consistently estimated (up to scale) — a form of semi-parametric resilience.

Practical implication: Use robust (sandwich) standard errors for Probit, just as you do for OLS. Same `sandwich` package in R.

The Sandwich Under Misspecification

What if Φ is the wrong link function?

If the true CEF is $G_0(X'\beta)$ but we estimate Probit, the information matrix equality **fails**:

$$\mathbb{E}[s_i s_i'] \neq -\mathbb{E}\left[\frac{\partial^2 \ell_i}{\partial\beta\partial\beta'}\right]$$

The asymptotic variance becomes the **sandwich** (recall Lectures 5–6):

$$\text{Var}(\sqrt{n}(\hat{\beta} - \beta_0)) = \underbrace{H^{-1}}_{\text{bread}} \underbrace{\mathbb{E}[s_i s_i']}_{\text{meat}} \underbrace{H^{-1}}_{\text{bread}}$$

where $H = -\mathbb{E}[\partial^2 \ell_i / \partial\beta\partial\beta']$.

Quasi-MLE (QMLE): Even with the wrong G, the index $X'\beta$ can be consistently estimated (up to scale) — a form of semi-parametric resilience.

Practical implication: Use robust (sandwich) standard errors for Probit, just as you do for OLS. Same `sandwich` package in R.

The Sandwich Under Misspecification

What if Φ is the wrong link function?

If the true CEF is $G_0(X'\beta)$ but we estimate Probit, the information matrix equality **fails**:

$$\mathbb{E}[s_i s_i'] \neq -\mathbb{E}\left[\frac{\partial^2 \ell_i}{\partial\beta\partial\beta'}\right]$$

The asymptotic variance becomes the **sandwich** (recall Lectures 5–6):

$$\text{Var}(\sqrt{n}(\hat{\beta} - \beta_0)) = \underbrace{H^{-1}}_{\text{bread}} \underbrace{\mathbb{E}[s_i s_i']}_{\text{meat}} \underbrace{H^{-1}}_{\text{bread}}$$

where $H = -\mathbb{E}[\partial^2 \ell_i / \partial\beta\partial\beta']$.

Quasi-MLE (QMLE): Even with the wrong G, the index $X'\beta$ can be consistently estimated (up to scale) — a form of semi-parametric resilience.

Practical implication: Use robust (sandwich) standard errors for Probit, just as you do for OLS. Same `sandwich` package in R.

The Sandwich Under Misspecification

What if Φ is the wrong link function?

If the true CEF is $G_0(X'\beta)$ but we estimate Probit, the information matrix equality **fails**:

$$\mathbb{E}[s_i s_i'] \neq -\mathbb{E}\left[\frac{\partial^2 \ell_i}{\partial\beta\partial\beta'}\right]$$

The asymptotic variance becomes the **sandwich** (recall Lectures 5–6):

$$\text{Var}(\sqrt{n}(\hat{\beta} - \beta_0)) = \underbrace{H^{-1}}_{\text{bread}} \underbrace{\mathbb{E}[s_i s_i']}_{\text{meat}} \underbrace{H^{-1}}_{\text{bread}}$$

where $H = -\mathbb{E}[\partial^2 \ell_i / \partial\beta\partial\beta']$.

Quasi-MLE (QMLE): Even with the wrong G, the index $X'\beta$ can be consistently estimated (up to scale) — a form of semi-parametric resilience.

Practical implication: Use robust (sandwich) standard errors for Probit, just as you do for OLS. Same `sandwich` package in R.

What if the Regressor is Unobserved?

In Probit:

$$P(Y_i = 1 \mid X_i) = \Phi(X_i'\beta)$$

But suppose the key regressor is **latent**.

Let:

- i = individuals
- j = items
- θ_i = latent trait (ability, ideology)

Model:

$$Y_{ij} = \mathbf{1}\{a_j\theta_i - b_j + \varepsilon_{ij} > 0\}$$

What if the Regressor is Unobserved?

In Probit:

$$P(Y_i = 1 \mid X_i) = \Phi(X_i'\beta)$$

But suppose the key regressor is **latent**.

Let:

- i = individuals
- j = items
- θ_i = latent trait (ability, ideology)

Model:

$$Y_{ij} = \mathbf{1}\{a_j\theta_i - b_j + \varepsilon_{ij} > 0\}$$

What if the Regressor is Unobserved?

In Probit:

$$P(Y_i = 1 \mid X_i) = \Phi(X_i'\beta)$$

But suppose the key regressor is **latent**.

Let:

- i = individuals
- j = items
- θ_i = latent trait (ability, ideology)

Model:

$$Y_{ij} = \mathbf{1}\{a_j\theta_i - b_j + \varepsilon_{ij} > 0\}$$

What if the Regressor is Unobserved?

In Probit:

$$P(Y_i = 1 \mid X_i) = \Phi(X_i'\beta)$$

But suppose the key regressor is **latent**.
Let:

- i = individuals
- j = items
- θ_i = latent trait (ability, ideology)

Model:

$$Y_{ij} = \mathbf{1}\{a_j\theta_i - b_j + \varepsilon_{ij} > 0\}$$

The IRT Model

Assume:

$$\varepsilon_{ij} \sim N(0,1)$$

Then:

$$P(Y_{ij} = 1 \mid \theta_i) = \Phi(a_j\theta_i - b_j)$$

Parameters:

- θ_i : latent ability
- a_j : discrimination
- b_j : difficulty

This is a **latent Probit model**.

The IRT Model

Assume:

$$\varepsilon_{ij} \sim N(0, 1)$$

Then:

$$P(Y_{ij} = 1 \mid \theta_i) = \Phi(a_j \theta_i - b_j)$$

Parameters:

- θ_i : latent ability
- a_j : discrimination
- b_j : difficulty

This is a **latent Probit model**.

The IRT Model

Assume:

$$\varepsilon_{ij} \sim N(0,1)$$

Then:

$$P(Y_{ij} = 1 \mid \theta_i) = \Phi(a_j\theta_i - b_j)$$

Parameters:

- θ_i : latent ability
- a_j : discrimination
- b_j : difficulty

This is a **latent Probit model**.

The IRT Model

Assume:

$$\varepsilon_{ij} \sim N(0,1)$$

Then:

$$P(Y_{ij} = 1 \mid \theta_i) = \Phi(a_j\theta_i - b_j)$$

Parameters:

- θ_i : latent ability
- a_j : discrimination
- b_j : difficulty

This is a **latent Probit model**.

Likelihood with Latent Traits

We do not observe θ_i.

Assume:

$$\theta_i \sim N(0, 1)$$

Individual likelihood contribution:

$$L_i(a, b) = \int \prod_j \Phi(a_j\theta - b_j)^{Y_{ij}} (1 - \Phi(a_j\theta - b_j))^{1-Y_{ij}} \phi(\theta) d\theta$$

Total likelihood:

$$L = \prod_i L_i$$

Now estimation requires numerical integration or EM.

Likelihood with Latent Traits

We do not observe θ_i.
Assume:

$$\theta_i \sim N(0, 1)$$

Individual likelihood contribution:

$$L_i(a, b) = \int \prod_j \Phi(a_j\theta - b_j)^{Y_{ij}}(1 - \Phi(a_j\theta - b_j))^{1-Y_{ij}}\phi(\theta)d\theta$$

Total likelihood:

$$L = \prod_i L_i$$

Now estimation requires numerical integration or EM.

Likelihood with Latent Traits

We do not observe θ_i.

Assume:

$$\theta_i \sim N(0,1)$$

Individual likelihood contribution:

$$L_i(a,b) = \int \prod_j \Phi(a_j\theta - b_j)^{Y_{ij}}(1-\Phi(a_j\theta - b_j))^{1-Y_{ij}}\phi(\theta)d\theta$$

Total likelihood:

$$L = \prod_i L_i$$

Now estimation requires numerical integration or EM.

Likelihood with Latent Traits

We do not observe θ_i.
Assume:

$$\theta_i \sim N(0,1)$$

Individual likelihood contribution:

$$L_i(a,b) = \int \prod_j \Phi(a_j\theta - b_j)^{Y_{ij}}(1-\Phi(a_j\theta - b_j))^{1-Y_{ij}}\phi(\theta)d\theta$$

Total likelihood:

$$L = \prod_i L_i$$

Now estimation requires numerical integration or EM.

Likelihood with Latent Traits

We do not observe θ_i.
Assume:

$$\theta_i \sim N(0,1)$$

Individual likelihood contribution:

$$L_i(a,b) = \int \prod_j \Phi(a_j\theta - b_j)^{Y_{ij}}(1-\Phi(a_j\theta - b_j))^{1-Y_{ij}}\phi(\theta)d\theta$$

Total likelihood:

$$L = \prod_i L_i$$

Now estimation requires numerical integration or EM.

Identification in IRT

Just as in Probit:

If we rescale:

$$\theta_i^* = c\theta_i$$

then:

$$a_j^* = \frac{a_j}{c}$$

The likelihood is unchanged.
We impose normalizations:

$$\mathbb{E}[\theta_i] = 0, \quad \text{Var}(\theta_i) = 1$$

Location and scale are not identified without normalization.

Identification in IRT

Just as in Probit:
If we rescale:

$$\theta_i^* = c\theta_i$$

then:

$$a_j^* = \frac{a_j}{c}$$

The likelihood is unchanged.
We impose normalizations:

$$\mathbb{E}[\theta_i] = 0, \quad \text{Var}(\theta_i) = 1$$

Location and scale are not identified without normalization.

Identification in IRT

Just as in Probit:

If we rescale:

$$\theta_i^* = c\theta_i$$

then:

$$a_j^* = \frac{a_j}{c}$$

The likelihood is unchanged.

We impose normalizations:

$$\mathbb{E}[\theta_i] = 0, \quad \mathrm{Var}(\theta_i) = 1$$

Location and scale are not identified without normalization.

Identification in IRT

Just as in Probit:
If we rescale:

$$\theta_i^* = c\theta_i$$

then:

$$a_j^* = \frac{a_j}{c}$$

The likelihood is unchanged.
We impose normalizations:

$$\mathbb{E}[\theta_i] = 0, \quad \mathrm{Var}(\theta_i) = 1$$

Location and scale are not identified without normalization.

Identification in IRT

Just as in Probit:
If we rescale:

$$\theta_i^* = c\theta_i$$

then:

$$a_j^* = \frac{a_j}{c}$$

The likelihood is unchanged.
We impose normalizations:

$$\mathbb{E}[\theta_i] = 0, \quad \text{Var}(\theta_i) = 1$$

Location and scale are not identified without normalization.

Probit in R: glm()

```
# Linear Probability Model
lpm <- lm(Y ~ X1 + X2, data = dta)

# Probit Model
probit <- glm(Y ~ X1 + X2, data = dta,
              family = binomial(link = "probit"))

# Compare coefficients
cbind(LPM = coef(lpm), Probit = coef(probit))
```

- glm() uses Fisher scoring (iteratively reweighted least squares)
- family = binomial(link = "probit") specifies $G = \Phi$
- For logit: family = binomial(link = "logit")

Probit in R: glm()

```r
# Linear Probability Model
lpm <- lm(Y ~ X1 + X2, data = dta)

# Probit Model
probit <- glm(Y ~ X1 + X2, data = dta,
              family = binomial(link = "probit"))

# Compare coefficients
cbind(LPM = coef(lpm), Probit = coef(probit))
```

- `glm()` uses Fisher scoring (iteratively reweighted least squares)
- `family = binomial(link = "probit")` specifies $G = \Phi$
- For logit: `family = binomial(link = "logit")`

Marginal Effects in R

Probit coefficients $\hat{\beta}_j$ are **not** marginal effects. Compute them manually:

```
# Average Marginal Effect (AME)
xb <- predict(probit, type = "link")  # X'beta-hat
ame <- mean(dnorm(xb)) * coef(probit)
ame
```

Or using the margins package:

```
library(margins)
summary(margins(probit))
```

The AME is directly comparable to the LPM coefficient — both estimate $\mathbb{E}[\partial P/\partial x_j]$.

Marginal Effects in R

Probit coefficients $\hat{\beta}_j$ are **not** marginal effects. Compute them manually:

```
# Average Marginal Effect (AME)
xb <- predict(probit, type = "link")  # X'beta-hat
ame <- mean(dnorm(xb)) * coef(probit)
ame
```

Or using the `margins` package:

```
library(margins)
summary(margins(probit))
```

The AME is directly comparable to the LPM coefficient — both estimate $\mathbb{E}[\partial P/\partial x_j]$.

Marginal Effects in R

Probit coefficients $\hat{\beta}_j$ are **not** marginal effects. Compute them manually:

```
# Average Marginal Effect (AME)
xb <- predict(probit, type = "link")  # X'beta-hat
ame <- mean(dnorm(xb)) * coef(probit)
ame
```

Or using the `margins` package:

```
library(margins)
summary(margins(probit))
```

The AME is directly comparable to the LPM coefficient — both estimate $\mathbb{E}[\partial P/\partial x_j]$.

Robust Standard Errors for Probit

Same `sandwich` tools from Lecture 6:

```
library(sandwich)
library(lmtest)

# Default (model-based) SEs
coeftest(probit)

# Robust (sandwich) SEs
coeftest(probit, vcov = vcovHC(probit, type = "HC1"))
```

- Model-based SEs rely on the information matrix equality (correct specification)
- Sandwich SEs are valid even under misspecification (QMLE)
- If model-based and robust SEs differ substantially, this signals possible misspecification

Robust Standard Errors for Probit

Same `sandwich` tools from Lecture 6:

```
library(sandwich)
library(lmtest)

# Default (model-based) SEs
coeftest(probit)

# Robust (sandwich) SEs
coeftest(probit, vcov = vcovHC(probit, type = "HC1"))
```

- Model-based SEs rely on the information matrix equality (correct specification)
- Sandwich SEs are valid even under misspecification (QMLE)
- If model-based and robust SEs differ substantially, this signals possible misspecification

Comparison Table

	Linear	**LPM**	**Probit**	**IRT**
Outcome	Continuous	Binary	Binary	Binary
CEF	$X'\beta$	$X'\beta$	$\Phi(X'\beta)$	$\Phi(a\theta - b)$
Moment Cond.	$\mathbb{E}[Xe] = 0$	$\mathbb{E}[Xe] = 0$	$\mathbb{E}[s(\beta)] = 0$	EM
Estimation	OLS	OLS	MLE	MLE+Integ.
Regressor	Observed	Observed	Observed	Latent

The GMM generalization: All of these are special cases of

$$\mathbb{E}[g(W_i, \theta_0)] = 0$$

Find $\hat{\theta}$ such that $\frac{1}{n}\sum_{i=1}^{n} g(W_i, \hat{\theta}) \approx 0$. This is the **Generalized Method of Moments**.

Comparison Table

	Linear	**LPM**	**Probit**	**IRT**
Outcome	Continuous	Binary	Binary	Binary
CEF	$X'\beta$	$X'\beta$	$\Phi(X'\beta)$	$\Phi(a\theta - b)$
Moment Cond.	$\mathbb{E}[Xe] = 0$	$\mathbb{E}[Xe] = 0$	$\mathbb{E}[s(\beta)] = 0$	EM
Estimation	OLS	OLS	MLE	MLE+Integ.
Regressor	Observed	Observed	Observed	Latent

The GMM generalization: All of these are special cases of

$$\mathbb{E}[g(W_i, \theta_0)] = 0$$

Find $\hat{\theta}$ such that $\frac{1}{n}\sum_{i=1}^{n} g(W_i, \hat{\theta}) \approx 0$. This is the **Generalized Method of Moments**.

Key Takeaways and Looking Ahead

Today:

- Binary outcomes $\Rightarrow$ nonlinear CEF, but LPM (BLP) remains a useful benchmark
- Probit: latent variable model estimated by MLE
- Coefficients $\neq$ marginal effects; compute AME
- Score equations are moment conditions — same logic as OLS normal equations
- Sandwich SEs handle misspecification, just as in the linear model
- IRT extends Probit to latent regressors

Next (Lecture 10): The formal asymptotic tools — WLLN, CLT, delta method — that justify everything we did today.

Key Takeaways and Looking Ahead

Today:

- Binary outcomes $\Rightarrow$ nonlinear CEF, but LPM (BLP) remains a useful benchmark
- Probit: latent variable model estimated by MLE
- Coefficients $\neq$ marginal effects; compute AME
- Score equations are moment conditions — same logic as OLS normal equations
- Sandwich SEs handle misspecification, just as in the linear model
- IRT extends Probit to latent regressors

Next (Lecture 10): The formal asymptotic tools — WLLN, CLT, delta method — that justify everything we did today.