

Heteroskedasticity, Logit and Probit Models, Maximum Likelihood Estimation, and Inference based on Maximum Likelihood

Review Session

CLABE 2025/2026

Marco Rosso

11 December 2025

Review Goals

- Refresh heteroskedasticity and robust inference in linear models
- Revisit the Linear Probability Model (LPM) and its limitations
- Understand Logit and Probit models and how to interpret marginal and discrete effects
- Review Maximum Likelihood estimation and ML-based inference (LR, Wald, LM)
- Practice translating theory into Stata commands and “by hand” exam-style calculations

Heteroskedasticity: Definition and Consequences

Definition: Errors have non-constant variance conditional on regressors

$$\text{Var}(\varepsilon_i | X) = \sigma_i^2 \neq \sigma^2$$

Consequences:

- OLS estimators remain **unbiased and consistent**
- **Bias in OLS variance estimators** — standard errors are wrong
- Gauss-Markov theorem violated — OLS is NO LONGER BLUE
- OLS is inefficient among linear unbiased estimators

When to care: Incorrect inference (t-tests, confidence intervals, hypothesis tests)

Testing for Heteroskedasticity: White Test

Auxiliary Regression (univariate X):

$$\hat{\varepsilon}_i^2 = \alpha_0 + \alpha_1 x_i + \alpha_2 x_i^2 + v_i$$

Hypothesis: $H_0 : \alpha_1 = \alpha_2 = 0$ (homoskedasticity)

Test Statistic:

$$\text{White} = n \cdot R^2 \sim \chi_2^2$$

Alternative (multivariate X):

$$\hat{\varepsilon}_i^2 = \alpha_0 + \alpha_1 \hat{y}_i + \alpha_2 \hat{y}_i^2 + v_i$$

Works for any model specification — preferred approach

White Test in Stata

Step 1: Estimate the main model

```
regress lsales lprice
```

Step 2: Perform White test

```
estat imtest, white
```

Alternative (manual auxiliary regression):

```
predict yhat
predict residuals, residuals
gen residuals_sq = residuals^2
gen lprice_sq = lprice^2
regress residuals_sq lprice lprice_sq
```

Interpretation: Reject H_0 if p-value < 0.05 \Rightarrow heteroskedasticity present

Dealing with Heteroskedasticity: Robust Standard Errors

Heteroskedasticity-Robust Variance Estimator:

$$\widehat{\text{Var}}(\hat{\beta}_j) = \frac{\sum_{i=1}^n (x_i - \bar{x})^2 \hat{\varepsilon}_i^2}{(\text{SST}_x^2)^2}$$

Properties:

- Valid in large samples whether or not heteroskedasticity exists
- Based on Law of Large Numbers and Central Limit Theorem
- Allows valid confidence intervals and hypothesis tests

Stata Implementation:

```
regress lsales lprice , robust
```

Reports standard errors that account for heteroskedasticity

Linear Probability Model: Setup and Issues

Model: Binary dependent variable $Y_i \in \{0, 1\}$

$$Y_i = \beta_0 + \beta_1 X_i + \varepsilon_i$$

Interpretation:

$$E(Y_i|X_i) = \Pr(Y_i = 1|X_i) = \beta_0 + \beta_1 X_i$$

→ β_1 = change in probability of $Y = 1$ when X increases by 1 unit

Issues:

- Predicted probabilities can be < 0 or > 1
- Normality assumption implausible (Y is binary)
- **Heteroskedasticity guaranteed:** $\text{Var}(\varepsilon_i|X) = \pi_i(1 - \pi_i)$
- Marginal effects are constant (unrealistic for probabilities)

When reasonable: Large sample, binary regressors, phenomenon not rare/frequent

LPM Estimation in Stata

```
// Estimate LPM with robust standard errors  
regress y x1 x2 x3, robust  
  
// Predict probabilities  
predict prob_y, xb  
  
// Check for out-of-bounds predictions  
summarize prob_y  
list prob_y if prob_y < 0 | prob_y > 1
```

Output interpretation:

- Coefficients = marginal effects in percentage points
- Use robust option due to heteroskedasticity
- R-squared tells goodness of fit (often low)

Fixing Heteroskedasticity in LPM: Feasible GLS

Known Heteroskedasticity Form in LPM:

$$\text{Var}(\varepsilon_i | X) = \sigma^2 \pi_i (1 - \pi_i)$$

where $\pi_i = \text{Prob}(Y_i = 1 | X_i) = \Pr(Y_i = 1 | X_i)$

FGLS Procedure:

1. Estimate by OLS: $Y_i = \beta_0 + \beta_1 X_i + \varepsilon_i$
2. Compute $\hat{\pi}_i = \hat{Y}_i$ (predicted probabilities)
3. Transform data: $Y_i^* = \frac{Y_i}{\sqrt{\hat{\pi}_i(1-\hat{\pi}_i)}}, X_i^* = \frac{X_i}{\sqrt{\hat{\pi}_i(1-\hat{\pi}_i)}}$
4. Estimate weighted regression: $Y_i^* = \beta_0^* X_i^* + \varepsilon_i^*$

Requirement: All weights must be positive!

FGLS Implementation in Stata

```
// Step 1: Initial OLS estimation  
regress y x  
predict yhat  
predict residuals, residuals  
  
// Step 2: Calculate weights (check for positivity)  
gen weight = yhat * (1 - yhat)  
list weight if weight <= 0  
  
// Step 3: Weighted regression (FGLS)  
gen y_weighted = y / sqrt(weight)  
gen x_weighted = x / sqrt(weight)  
regress y_weighted x_weighted [aw=weight], noconstant
```

Note: noconstant option used because transformation includes intercept

Logit and Probit: Motivation

Problem with LPM: Predicted probabilities unbounded

Solution: Use cumulative distribution functions (CDF) to bound [0, 1]

Model Specification:

$$E(Y_i|X_i) = \Pr(Y_i = 1|X_i) = F(\beta_0 + \beta_1 X_i)$$

- ▶ **Logit:** $F = \frac{e^z}{1+e^z}$ (logistic CDF)
- ▶ **Probit:** $F = \Phi(z)$ (standard normal CDF)

where $z = \beta_0 + \beta_1 X_i$ is the index

Latent Variable Interpretation:

$$\begin{aligned}Y_i^* &= \beta_0 + \beta_1 X_i + \varepsilon_i \\ Y_i &= 1 \text{ if } Y_i^* > 0, \quad 0 \text{ otherwise}\end{aligned}$$

Model Identification in Logit/Probit

Identification Problem:

$$Y_i^* = \beta_0 + \beta_1 X_i + \varepsilon_i \quad \text{and} \quad Z_i^* = k \cdot Y_i^* = k\beta_0 + k\beta_1 X_i + k\varepsilon_i$$

Both models generate identical observed Y_i , but parameters differ by scale!

Solution: Standardize Error Variance

- ▶ **Logit:** $\text{Var}(\varepsilon_i) = \frac{\pi^2}{3} \approx 3.29$ (standardized)
- ▶ **Probit:** $\text{Var}(\varepsilon_i) = 1$ (standard normal)

Consequence: Error variance is **NOT** an unknown parameter to estimate

Implication: Cannot directly compare coefficients between logit and probit (different scales)

Marginal Effects in Logit and Probit

Key Difference from LPM:

$$\frac{\partial \Pr(Y_i = 1|X_i)}{\partial X_j} = f(\beta_0 + \beta_1 X_i) \cdot \beta_j$$

Marginal effects depend on the value of all covariates!

Logit Marginal Effect:

$$ME_{\text{logit}} = \frac{e^z}{(1 + e^z)^2} \cdot \beta_j = \pi(1 - \pi)\beta_j$$

Probit Marginal Effect:

$$ME_{\text{probit}} = \phi(z) \cdot \beta_j$$

where $\phi(z)$ is the standard normal PDF

Typical Choices for Evaluation Point:

- At sample means: $z = \bar{z}$
- At median values
- At particular values of interest

Discrete Changes for Binary Regressors

When X is binary: Marginal effect not well-defined; use discrete change

Procedure:

1. Compute $\Pr(Y = 1|X = 0)$ with all other variables at specified values
2. Compute $\Pr(Y = 1|X = 1)$ with all other variables at specified values
3. Discrete change = $\Pr(Y = 1|X = 1) - \Pr(Y = 1|X = 0)$

Stata Implementation:

```
// Estimate model  
logit y x1 binary_x x3  
  
// Discrete change for binary_x at means  
margins , dydx(binary_x) atmeans
```

Logit and Probit Estimation in Stata

Estimate models:

```
// Logit model  
logit y x1 x2 x3
```

```
// Probit model  
probit y x1 x2 x3
```

Compute marginal effects (at means):

```
// After logit or probit  
margins , dydx(*) atmeans
```

Predict probabilities:

```
predict prob_logit  
predict prob_probit
```

Compare predictions:

```
summarize prob_logit prob_probit
```

Maximum Likelihood Principle

Basic Idea: Choose parameters that maximize probability of observing the data

Likelihood Function:

$$L(\theta|y) = \prod_{i=1}^n p(y_i|x_i; \theta)$$

Log-Likelihood:

$$\ell(\theta|y) = \log L(\theta|y) = \sum_{i=1}^n \log p(y_i|x_i; \theta)$$

ML Estimator:

$$\hat{\theta}_{ML} = \arg \max_{\theta} \ell(\theta|y)$$

First-Order Condition:

$$\frac{\partial \ell}{\partial \theta} = 0 \quad (\text{score function})$$

ML Properties and Inference

Properties of ML Estimator (under regularity conditions):

- **Consistent:** $\hat{\theta}_{ML} \xrightarrow{P} \theta_0$
- **Asymptotically unbiased:** $E(\hat{\theta}_{ML}) \rightarrow \theta_0$
- **Asymptotically efficient:** Achieves Cramér-Rao lower bound

Asymptotic Distribution:

$$\hat{\theta}_{ML} \sim N(\theta_0, I(\theta_0)^{-1})$$

where $I(\theta)$ is the Information Matrix

Variance Estimation:

$$\widehat{\text{Var}}(\hat{\theta}_{ML}) = - \left[\frac{\partial^2 \ell}{\partial \theta \partial \theta'} \right]^{-1}$$

(Negative inverse of Hessian)

Hypothesis Testing with ML

Three Asymptotically Equivalent Tests:

1. Likelihood Ratio (LR) Test:

$$LR = -2[\ell(\hat{\theta}_0) - \ell(\hat{\theta}_{ML})] \sim \chi_q^2$$

Requires estimating both restricted and unrestricted models

2. Wald Test:

$$W = (\hat{\theta}_{ML} - \theta_0)' \widehat{\text{Var}}(\hat{\theta}_{ML})^{-1} (\hat{\theta}_{ML} - \theta_0) \sim \chi_q^2$$

Requires only unrestricted model

3. Lagrange Multiplier Test:

$$LM = s(\hat{\theta}_0)' I(\hat{\theta}_0)^{-1} s(\hat{\theta}_0) \sim \chi_q^2$$

Requires only restricted model

Note: For single coefficient test: $W = t^2$

Manual Computation: Binomial ML & Confidence Intervals

Binomial ML Example (Academic Exams): $p_{ML} = \frac{n_1}{n}$, $SE = \sqrt{\frac{\hat{p}(1-\hat{p})}{n}}$

```
// Example: 71 students passed out of 95
scalar n1 = 71                                // number of successes
scalar n_total = 95                            // total sample size
scalar p_hat = n1/n_total                      // ML estimate
scalar se_p = sqrt(p_hat*(1-p_hat)/n_total)    // standard error
scalar ci_low = p_hat - 1.96*se_p              // 95% CI lower bound
scalar ci_high = p_hat + 1.96*se_p             // 95% CI upper bound

display "p = " %5.3f p_hat " SE = " %5.4f se_p
display "95% CI = [" %5.3f ci_low ", " %5.3f ci_high "]"

// Test H0: p = 0.5
scalar z_test = (p_hat - 0.5)/se_p
scalar pval_test = 2*(1 - normal(abs(z_test)))
display "Test p=0.5: z = " %5.3f z_test " p-value = " %5.4f pval_test
```

ML Inference in Stata (1)

Estimate model and get test on single coefficient:

```
// Logit model with inference  
logit y x1 x2 x3  
  
// Test H0: beta1 = 0 (t-statistic automatic)  
display "t-stat = " _b[x1] / _se[x1]  
  
// Wald test for multiple restrictions  
test x1 x2
```

ML Inference in Stata (2)

Likelihood Ratio test:

```
// Unrestricted model  
logit y x1 x2 x3  
estimates store unrestricted  
  
// Restricted model  
logit y x1  
estimates store restricted  
  
// LR test  
lrtest unrestricted restricted
```

Goodness of Fit in Logit/Probit

Pseudo R-squared:

$$\text{Pseudo } R^2 = 1 - \frac{\ell(\hat{\theta})}{\ell(\hat{\theta}_0)}$$

where $\ell(\hat{\theta}_0)$ = log-likelihood of model with only constant

- ▶ Ranges from 0 to 1
- ▶ Often much lower than OLS R^2
- ▶ Interpretation different from linear models

Fraction of Correctly Predicted:

$$\text{Accuracy} = \frac{n_{11} + n_{00}}{n}$$

where n_{11} = correctly predicted $Y = 1$, n_{00} = correctly predicted $Y = 0$

Confusion Matrix:

		Predicted	
		$\hat{Y} = 1$	$\hat{Y} = 0$
Actual	$Y = 1$	n_{11}	n_{10}
	$Y = 0$	n_{01}	n_{00}

Logit/Probit Goodness of Fit in Stata

```
// Estimate model  
logit y x1 x2 x3  
  
// Obtain pseudo R-squared (in output)  
// Predict probabilities  
predict prob_pred  
  
// Predict binary choice (threshold at 0.5)  
gen y_pred = (prob_pred > 0.5)  
  
// Compute fraction correctly predicted  
gen correct = (y_pred == y)  
summarize correct
```

Output includes:

- Log-likelihood values
- Pseudo R-squared
- Akaike and Bayesian Information Criteria (AIC, BIC)

Manual Computation: APE (Average Partial Effects)

APE for Logit: $\widehat{APE} = \frac{1}{n} \sum_{i=1}^n \pi_i(1 - \pi_i)\hat{\beta}_j$

// After: logit y x1 x2 x3

```
predict xb, xb linear                                // predicted index z_i
gen pi = exp(xb)/(1+exp(xb))                      // pi_i = F(z_i)
gen me_x1 = pi*(1-pi)*_b[x1]                        // marginal effect for each obs
scalar APE_x1 = mean(me_x1)                          // average across all n
display "APE(x1) = " %6.4f APE_x1
```

APE for Probit: $\widehat{APE} = \frac{1}{n} \sum_{i=1}^n \phi(z_i)\hat{\beta}_j$

// After: probit y x1 x2 x3

```
predict xb, xb linear
gen phi = normalden(xb)                            // phi(z_i) = normal PDF
gen me_x1_probit = phi*_b[x1]
scalar APE_probit_x1 = mean(me_x1_probit)
display "APE Probit(x1) = " %6.4f APE_probit_x1
```

Manual Computation: MPE (at means)

MPE at sample means (Logit): $\widehat{MPE} = \hat{\pi}_{\bar{x}}(1 - \hat{\pi}_{\bar{x}})\hat{\beta}_j$

```
// Calculate z at sample means  
  
summarize x1 x2 x3, meanonly  
scalar z_mean = _b[_cons] + _b[x1]*r(mean1) + _b[x2]*r(mean2) \\  
+ _b[x3]*r(mean3)  
scalar pi_mean = exp(z_mean)/(1+exp(z_mean))  
scalar MPE_x1 = pi_mean*(1-pi_mean)*_b[x1]  
  
display "MPE(x1) at means = " %6.4f MPE_x1
```

Manual Computation: Wald Tests

Wald Test Manual ($H_0: \beta_j = 0$): $W = \left(\frac{\hat{\beta}_j}{\text{SE}(\hat{\beta}_j)} \right)^2 \sim \chi^2_1$

```
// After any logit/probit estimation

scalar t_stat = _b[x1]/_se[x1]                                // t-statistic
scalar wald_stat = t_stat^2                                     // Wald = t^2
scalar pval_wald = chi2tail(1, wald_stat)                      // p-value

display "Wald stat = " %6.3f wald_stat " p-value = " %6.4f pval_wald

// OR use direct t-test p-value:
scalar pval_t = 2*ttail(e(df_r), abs(t_stat))
```

Manual Computation: Pseudo R²

Pseudo R² Manual: Pseudo $R^2 = 1 - \frac{\ell(\hat{\theta})}{\ell(\hat{\theta}_0)}$

```
// After logit/probit estimation
```

```
scalar ll_full = e(ll)           // log-likelihood of full model  
scalar ll_null = e(ll_0)         // log-likelihood of null (constant only)  
  
scalar pseudo_R2 = 1 - (ll_full / ll_null)  
  
display "Pseudo R^2 = " %6.4f pseudo_R2
```

Manual Computation: Likelihood Tests

Likelihood Ratio Test Manual: $LR = -2[\ell_{\text{restricted}} - \ell_{\text{unrestricted}}] \sim \chi_q^2$

```
// Estimate unrestricted model
```

```
logit y x1 x2 x3
```

```
scalar ll_unres = e(ll)
```

```
// Estimate restricted model (drop x3)
```

```
logit y x1 x2
```

```
scalar ll_res = e(ll)
```

```
// Compute LR test
```

```
scalar lr_stat = -2*(ll_res - ll_unres) // positive , tests on x3
scalar lr_pval = chi2tail(1, lr_stat) // df = #restrictions
```

```
display "LR stat = " %6.3f lr_stat " p-value = " %6.4f lr_pval
```

Manual Computation: White Test

White Test Manual: $\text{White} = n \cdot R_{\text{aux}}^2 \sim \chi_2^2$

```
// After: regress y x1 x2
```

```
predict yhat          // predicted values
predict resid, residuals // residuals
gen resid_sq = resid^2 // squared residuals
```

```
// Auxiliary regression (for univariate case)
```

```
gen yhat_sq = yhat^2
regress resid_sq yhat yhat_sq
scalar white_nR2 = e(N)*e(r2)
scalar white_pval = chi2tail(2, white_nR2)

display "White test = " %6.2f white_nR2 " p-value = " %6.4f white_pval
```

Manual Computation: Discrete Changes

Discrete Change (Binary Regressor): $\Delta = \Pr(Y = 1|X_j = 1) - \Pr(Y = 1|X_j = 0)$ at fixed X values

```
// After: logit y x1 binary_x x3
```

```
summarize x1 x3, meanonly
```

```
scalar x1_m = r(mean1)
```

```
scalar x3_m = r(mean2)
```

```
// Pr(Y=1) when binary_x=1
```

```
scalar z1 = _b[_cons] + _b[x1]*x1_m + _b[binary_x]*1 + _b[x3]*x3_m
```

```
scalar pr_y1 = exp(z1)/(1+exp(z1))
```

```
// Pr(Y=1) when binary_x=0
```

```
scalar z0 = _b[_cons] + _b[x1]*x1_m + _b[binary_x]*0 + _b[x3]*x3_m
```

```
scalar pr_y0 = exp(z0)/(1+exp(z0))
```

```
// Discrete change
```

```
scalar disc_change = pr_y1 - pr_y0
```

```
display "Discrete change = " %6.4f disc_change
```

Stata Command Summary

Task	Stata Command
Basic OLS	regress y x1 x2
OLS with robust SE	regress y x1 x2, robust
White test	estat imtest, white
Logit model	logit y x1 x2
Probit model	probit y x1 x2
Marginal effects	margins, dydx(*) atmeans
Predict probabilities	predict prob
Test restrictions	test x1 x2
LR test	lrtest model1 model2
Model comparison	estimates table model1 model2

Key Formulas

- **LPM:** $E(Y|X) = \beta_0 + \beta_1 X$
- **Logit:** $E(Y|X) = \frac{e^z}{1+e^z}$ where $z = \beta_0 + \beta_1 X$
- **Probit:** $E(Y|X) = \Phi(z)$ where $z = \beta_0 + \beta_1 X$
- **APE (Logit):** $\frac{1}{n} \sum \pi_i(1 - \pi_i)\beta_j$
- **APE (Probit):** $\frac{1}{n} \sum \phi(z_i)\beta_j$
- **Heteroskedasticity in LPM:** $\text{Var}(\varepsilon_i|X) = \pi_i(1 - \pi_i)$
- **Log-likelihood:** $\ell(\theta) = \sum_{i=1}^n \log p(y_i|x_i; \theta)$
- **Pseudo R-squared:** $1 - \frac{\ell(\hat{\theta})}{\ell(\hat{\theta}_0)}$

Tips

1. **Always check for heteroskedasticity** in binary dependent variable models
2. **Use robust standard errors** when in doubt
3. **APE/MPE depend on evaluation point** — compute at means or medians
4. **Remember the scale problem** — can't compare logit and probit coefficients
5. **For exams without 'margins':** Use scalars to compute APE, MPE, tests manually
6. **For LPM,** use robust SE due to guaranteed heteroskedasticity
7. **Goodness of fit** different for nonlinear models — pseudo R² not comparable to OLS
8. **Interpretation matters:** Coefficients show direction; effects show magnitude

Worked Example: Complete Analysis (1)

Scenario: Model brand choice ($Y=1$ if buy Heinz) using relative price

Stata Workflow:

```
// Data setup  
use ketchup_data.dta
```

```
// 1. Estimate OLS (LPM) with robust SE
```

```
regress choice rprice, robust
```

```
// 2. Test for heteroskedasticity
```

```
estat imtest, white
```

Worked Example: Complete Analysis (2)

```
// 3. Estimate Logit and compute APE manually
```

```
logit choice rprice
predict xb, xb linear
gen pi = exp(xb)/(1+exp(xb))
gen me = pi*(1-pi)*_b[rprice]
scalar APE = mean(me)
display "APE(rprice) = " APE
```

```
// 4. Estimate Probit
```

```
probit choice rprice
predict xb_p, xb linear
gen phi = normalden(xb_p)
gen me_p = phi*_b[rprice]
scalar APE_p = mean(me_p)
```

Worked Example: Complete Analysis (3)

```
// 5. Compare predictions  
  
predict prob_logit  
predict prob_probit  
summarize prob_*
```