Ordinary Least Squares (OLS)

Dr. Syed Badruddoza

Texas Tech University

May 2, 2025

The Linear Regression Model

The multiple linear regression model:

$$y = X\beta + \varepsilon \tag{1}$$

where:

- y is the dependent variable (vector of observations),
- X is the $n \times K$ matrix of independent variables,
- β is the $K \times 1$ vector of parameters to be estimated,
- ε is the $n \times 1$ vector of disturbances (errors).

OLS Estimator

The OLS estimator of β minimizes the sum of squared errors:

$$S(\beta) = (y - X\beta)'(y - X\beta) \tag{2}$$

Taking the first-order condition:

$$\frac{\partial S}{\partial \beta} = -2X'y + 2X'X\beta = 0 \tag{3}$$

Note the exogeneity condition : $X'(y - X\beta) = 0$

GMM uses this

Solving for β:

$$\hat{\beta} = (X'X)^{-1}X'y \tag{4}$$

Variance-Covariance Matrix of OLS Estimator

The variance of the OLS estimator is given by:

$$Var(\hat{\beta}) = Var[(X'X)^{-1}X'\varepsilon] \tag{5}$$

Using the property $Var(Ax) = AVar(x)A'$, we get:

$$Var(\hat{\beta}) = (X'X)^{-1}X'Var(\varepsilon)X(X'X)^{-1} \tag{6}$$

Since $Var(\varepsilon) = \sigma^2 I_n$, we obtain:

$$Var(\hat{\beta}) = \sigma^2(X'X)^{-1} \tag{7}$$

The standard errors of the estimates are:

$$SE(\hat{\beta}) = \sqrt{\sigma^2(X'X)^{-1}} \tag{8}$$

Assumptions of OLS

1. Linearity: The model is linear in parameters.
2. Full Rank (No Perfect Multicollinearity): X has full column rank.
3. Exogeneity (Zero Conditional Mean): $E[\varepsilon|X] = 0$.
4. Homoskedasticity (Constant Variance): $Var(\varepsilon_i) = \sigma^2$ for all i.
5. No Autocorrelation (Independence of Errors): $Cov(\varepsilon_i, \varepsilon_j) = 0$ for $i \neq j$.
6. Normality (for Inference): $\varepsilon \sim N(0, \sigma^2 I)$.

Note: Assumptions 1–4 constitute the **Gauss-Markov assumptions**, ensuring the OLS estimator is **BLUE** (Best Linear Unbiased Estimator). Adding the **normality assumption** results in the **Classical Linear Model (CLM) assumptions**, which are necessary for valid hypothesis testing.

Does This Variance Satisfy the CRLB?

The likelihood function for $y \sim N(X\beta, \sigma^2 \mathbf{I}_n)$ is:

$$L(\beta, \sigma^2) = \frac{1}{(2\pi\sigma^2)^{n/2}} \exp\left(-\frac{1}{2\sigma^2}(y - X\beta)'(y - X\beta)\right).$$

The log-likelihood function:

$$\ell(\beta, \sigma^2) = -\frac{n}{2}\ln(2\pi) - \frac{n}{2}\ln(\sigma^2) - \frac{1}{2\sigma^2}(y - X\beta)'(y - X\beta).$$

Taking the derivative with respect to β:

$$\frac{\partial \ell}{\partial \beta} = \frac{1}{\sigma^2}X'(y - X\beta).$$

Fisher Information and the CRLB

The negative expectation of the Hessian (second derivative):

$$\frac{\partial^2 \ell}{\partial \beta \partial \beta'} = -\frac{1}{\sigma^2} X'X.$$

The Fisher Information Matrix is:

$$I(\beta) = \mathbb{E}\left[-\frac{\partial^2 \ell}{\partial \beta \partial \beta'}\right] = \frac{1}{\sigma^2} X'X.$$

By the Cramér-Rao Lower Bound (CRLB), the covariance matrix of any unbiased estimator $\tilde{\beta}$ satisfies:

$$\mathrm{Var}(\tilde{\beta}) \geq I(\beta)^{-1} = \sigma^2 (X'X)^{-1}.$$

Conclusion: The OLS estimator attains this bound, meaning its variance satisfies the CRLB.

MLE Estimation of $\hat{\beta}$ given $\epsilon \sim N(0, \sigma^2 I_n)$

Log-Likelihood Function

$$\log L(\beta, \sigma^2) = -\frac{n}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}(y - X\beta)'(y - X\beta).$$

First-Order Condition

$$\frac{\partial \log L}{\partial \beta} = X'y - X'X\beta = 0.$$

$$\hat{\beta}_{MLE} = (X'X)^{-1}X'y.$$

The Maximum Likelihood estimator of β coincides with the OLS estimator.

Simulating Data for $y = X\beta + \epsilon$ in Python

```
import numpy as np
import statsmodels.api as sm
np.random.seed(1234)
n, k = 100, 3
X = np.random.randn(n, k)  # Random predictors
X = np.hstack([np.ones((n, 1)), X])  # With intercept
beta = np.array([1.5, -2.0, 0.5, 1.0])
epsilon = np.random.randn(n)
y = X @ beta + epsilon
model = sm.OLS(y, X).fit()
print(model.summary())
```

OLS Regression Results

Variable	Coefficient	Std. Error	t-Statistic	P-value
Constant	1.6094	0.101	15.973	0.000
X1	-2.0181	0.087	-23.323	0.000
X2	0.3736	0.110	3.406	0.001
X3	0.9388	0.115	8.155	0.000

- $R^2 = 0.868$, Adjusted $R^2 = 0.864$
- F-statistic: 210.2 ($p < 0.0001$)
- Observations: 100
- Interpretation: An increase in X_1 is associated with a decrease in y by 2.0181 units, *ceteris paribus*.

Hypothesis Testing

Testing $H_0 : \beta_k = 0$ (individual significance test)

- Test statistic:

$$t_k = \frac{\text{Estimate} - \text{Hypothesized value}}{\text{Standard Error}} = \frac{\hat{\beta}_k}{SE(\hat{\beta}_k)} \qquad (9)$$

- Follows a $t(n-K)$ distribution.
- Degrees of freedom is $df = n - K$ where n is the sample size and K is the number of estimated parameters, and k=1,2,...,K.

Calculating p-values in OLS

- The p-value is the probability of observing a test statistic as extreme as t_k, assuming the null hypothesis $H_0 : \beta = 0$ is true.
- Or, p-value is the smallest level of significance where the null hypothesis can be rejected.
- For a two-tailed test:

$$p = 2 \times (1 - CDF_t(|t_k|, df))$$

  where CDF is the cumulative distribution function of the t-distribution.

Decision Rule:

- If $p < \alpha$, reject H_0 (typically $\alpha = 0.05$).
- If $p \geq \alpha$, fail to reject H_0 (insufficient evidence).

The Wald Test

- The Wald test is used to test linear restrictions on regression coefficients.
- It evaluates whether a set of parameters $H_0 : R\beta = r$ holds.
- Commonly used in hypothesis testing for nested models.

General Hypothesis:

$$H_0 : R\beta = r, \quad H_1 : R\beta \neq r$$

where:

- R is a $q \times k$ restriction matrix.
- r is a $q \times 1$ vector.

Wald Test (Contd.)

Wald Statistic:

$$W = (R\hat{\beta} - r)'[R(X'X)^{-1}R']^{-1}(R\hat{\beta} - r)$$

- $\hat{\beta}$ is the OLS estimate.
- $R(X'X)^{-1}R'$ captures the covariance of restrictions.
- Under H_0, $W \sim \chi^2_q$, where q is the number of restrictions.
- Reject H_0 if $W > \chi^2_q(\alpha)$ at significance level α.

Wald Test Example

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \varepsilon$$

Hypothesis:

$$H_0 : \beta_2 = \beta_3 = 0 \quad \text{(No effect of } X_2 \text{ and } X_3\text{)}$$

Step 1: Define R and r

$$R = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad r = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Step 2: Compute Wald Statistic

$$W = (R\hat{\beta} - r)'[R(X'X)^{-1}R']^{-1}(R\hat{\beta} - r)$$

Step 3: Compare with χ^2_2 critical value

- If $W > \chi^2_2(\alpha)$, reject H_0.
- Otherwise, fail to reject H_0.

Wald Test Example (Contd.)

```
import numpy as np
import statsmodels.api as sm
np.random.seed(1234)
n = 100
X = np.random.randn(n, 3)
X = sm.add_constant(X)
beta = np.array([1.5, -2.0, 0.5, 1.0])
eps = np.random.randn(n)
y = X @ beta + eps
model = sm.OLS(y, X).fit()
R = np.array([[0, 0, 1, 0], [0, 0, 0, 1]])
r = np.array([0, 0])
model.wald_test((R, r))
```

- The test statistic is compared to a χ^2 critical value.
- A low p-value indicates rejection of H_0.

Likelihood Ratio Test

Test for Nested Models: $\Lambda = -2\,(\ell_0 - \ell_1)$

- $\ell_0 =$ Log-likelihood of restricted model.
- $\ell_1 =$ Log-likelihood of full model.

Test Statistic:

$$\Lambda \sim \chi^2_{df}, \quad df = \text{difference in parameters.}$$

Hypothesis:

- H_0 : Restricted model is sufficient.
- H_1 : Full model significantly improves fit.

Reject H_0 if Λ is large (p-value < 0.05).

MLE distribution

Lagrange Multiplier (LM) Test

Test if a restricted model is significantly different from an unrestricted model.

$$LM = S(\hat{\theta}_0)' I(\hat{\theta}_0)^{-1} S(\hat{\theta}_0)$$

- $S(\hat{\theta}_0)$ is the score function at the restricted estimates.
- $I(\hat{\theta}_0)$ is the Fisher Information Matrix.
- $LM \sim \chi^2_{df}$, where df is the number of constraints.

Hypothesis Testing:

- H_0 : The restricted model is correct.
- H_1 : The unrestricted model is significantly better.

If LM is large ($p < 0.05$), reject H_0 $\rightarrow$ the restricted model is insufficient.

Goodness of Fit

Coefficient of Determination (R^2)

$$R^2 = 1 - \frac{SS_{residual}}{SS_{total}} \tag{10}$$

where:

$$SS_{residual} = \sum (y_i - \hat{y}_i)^2, \quad SS_{total} = \sum (y_i - \bar{y})^2 \tag{11}$$

Adding more predictors to the model reduces $SS_{residual}$, or keeps it the same. **Adjusted** R^2 accounts for model complexity:

$$\bar{R}^2 = 1 - \frac{\frac{SS_{residual}}{n-K}}{\frac{SS_{total}}{n-1}} \tag{12}$$

Log transformations

1. **Log-Lin Model:** $\log(y) = \beta_0 + \beta_1 x + \epsilon$

$$\%\Delta y \approx 100 \cdot \beta_1$$

If $\beta_1 = 0.05$, a 1-unit increase in x is associated with a 5% increase in y.

2. **Log-Log Model:** $\log(y) = \beta_0 + \beta_1 \log(x) + \epsilon$

$$\beta_1 = \frac{\%\Delta y}{\%\Delta x}$$

Elasticity : If $\beta_1 = 1.2$, a 1% increase in x increases y by 1.2%.

3. **Lin-Log Model:** $y = \beta_0 + \beta_1 \log(x) + \epsilon$

$$\Delta y \approx \beta_1 \cdot 0.01 \cdot \%\Delta x$$

If $\beta_1 = 2.5$, a 1% increase in x increases y by 0.025 units.

Violation of the Assumptions

1. **Linearity**: Nonlinear models, supervised machine learning.
2. **Full Rank**: LASSO, clustering or unsupervised learning.
3. **Exogeneity**: Causal models, 2SLS, RDD, TWFE.
4. **Homoscedasticity**: GLS, robust error variance.
5. **No Autocorrelation**: Time series modeling.
6. **Normality**: Other distribution, e.g., Negative Binomial.

Chapter 2

Violation of Homoscedasticity

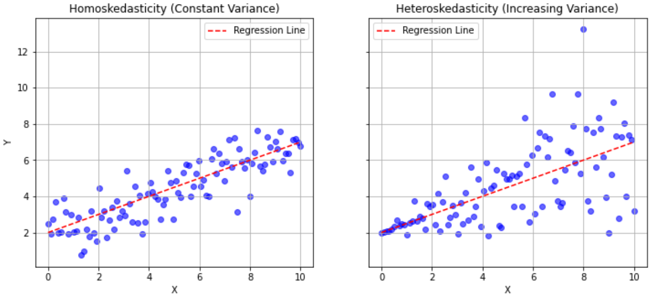
When the Error Variance is Not Constant

- Heteroskedasticity occurs when the variance of errors (ε) is not constant across observations.
- Standard OLS assumptions require $\mathrm{Var}(\varepsilon|X) = \sigma^2 I_n$.
- If violated, OLS estimates remain unbiased and consistent, but no longer BLUE as standard errors and hypothesis tests are unreliable.

Weighted Least Squares (WLS): Assign weights to observations inversely proportional to the variance of their errors.

- Generalized Least Squares
- Robust Standard Errors (Huber-White Sandwich Estimator)
- Clustered Standard Errors (VCE Cluster)

Homoskedasticity vs. Heteroskedasticity



Testing for Heteroskedasticity

1. **Breusch-Pagan Test:** Regress squared residuals on explanatory variables.

$$\hat{\varepsilon}_i^2 = \gamma_0 + \gamma_1 X_{1i} + \cdots + \gamma_k X_{ki} + u_i$$

   - $H_0 : \gamma_1 = \gamma_2 = \cdots = \gamma_k = 0$ (Homoscedasticity).
   - Test Statistic: $LM = nR^2 \sim \chi_k^2$.
2. **White Test:** Like Breusch-Pagan but includes squares and interactions of X. More general but requires a large sample.
3. **Goldfeld-Quandt Test:** Sort data by X, split into low and high X groups (dropping the middle), estimate residual variances s^2_{small} and s^2_{large}. $H_0 =$ Homoscedasticity.

   - Test Statistic: $F = \frac{s^2_{\text{large}}}{s^2_{\text{small}}} \sim F$.

Generalized Least Squares (GLS)

Given the original model: $Y = X\beta + e$, GLS applies the transformation: $Y^* = PY, \quad X^* = PX, \quad e^* = Pe$, where P is a transformation matrix such that: $P'P = \Omega^{-1}$ and after the transformation, the transformed error term satisfies:

$$\mathrm{Var}(e^*) = P \cdot \mathrm{Var}(e) \cdot P' = P\Omega P' = I.$$

The transformed model becomes:

$$Y^* = X^*\beta + e^*,$$

which now satisfies the OLS assumptions. The GLS estimator is given by:

$$\hat{\beta}_{\mathrm{GLS}} = (X'\Omega^{-1}X)^{-1}X'\Omega^{-1}Y.$$

Variance of the GLS Estimator

$$\hat{\beta}_{\text{GLS}} = (X'\Omega^{-1}X)^{-1}X'\Omega^{-1}Y.$$

$$\hat{\beta}_{\text{GLS}} = \beta + (X'\Omega^{-1}X)^{-1}X'\Omega^{-1}\varepsilon.$$

$$\text{Var}(\hat{\beta}_{\text{GLS}}) = \text{Var}[(X'\Omega^{-1}X)^{-1}X'\Omega^{-1}\varepsilon].$$

Using Var$(\varepsilon) = \Omega$**:**

$$\text{Var}(\hat{\beta}_{\text{GLS}}) = (X'\Omega^{-1}X)^{-1}X'\Omega^{-1}\Omega\Omega^{-1}X(X'\Omega^{-1}X)^{-1}.$$

$$\text{Var}(\hat{\beta}_{\text{GLS}}) = (X'\Omega^{-1}X)^{-1}X'\Omega^{-1}X(X'\Omega^{-1}X)^{-1}.$$

$$\text{Var}(\hat{\beta}_{\text{GLS}}) = (X'\Omega^{-1}X)^{-1}.$$

GLS (Contd.)

- **Efficiency**: GLS is more efficient than OLS when heteroskedasticity or correlation is present, as it produces smaller variances for the parameter estimates.
- **Generalization**: GLS is applicable to models with complex error structures, such as heteroskedasticity or autocorrelation.

Difference Between GLS and FGLS

Generalized Least Squares (GLS):

- Assumes the error variance-covariance structure (Ω) is known.
- Transforms to make errors homoscedastic and uncorrelated.
- The transformed model is estimated using OLS.

Feasible Generalized Least Squares (FGLS):

- Unlike GLS, Ω is unknown and estimated from the data.
- Initial estimate (e.g., OLS residuals) to approximate Ω.
- After estimating Ω, applies GLS on the transformed model.
- Iterative procedures (like Cochrane-Orcutt for AR(1) errors) can improve estimates.
- Not unbiased, but consistent and asymptotically more efficient than the OLS $\hat{\beta}$ in heteroskedasticity.

Feasible GLS (FGLS) for Autocorrelation

Model with Autocorrelation

$$\varepsilon_t = \rho\varepsilon_{t-1} + u_t, \quad |\rho| < 1, \quad u_t \sim N(0, \sigma_u^2).$$

Transformation

- To eliminate autocorrelation, transform the model:

  $$y_t^* = \beta_0(1-\rho) + \beta_1 x_{1t}^* + \cdots + \beta_k x_{kt}^* + u_t,$$

  where:

  $$y_t^* = y_t - \rho y_{t-1}, \quad x_{jt}^* = x_{jt} - \rho x_{j(t-1)}.$$

- After transformation, apply OLS to the transformed model.

FGLS for known heteroskedasticity

Suppose the error structure is known:

$$\mathrm{Var}(\varepsilon_i|X_i) = \sigma_i^2 = \sigma^2 h(X_i),$$

where $h(X_i)$ is a known function of X_i.

Transformation:

- Divide both sides of the regression equation by $\sqrt{h(X_i)}$:

$$\frac{y_i}{\sqrt{h(X_i)}} = \beta_0 \frac{1}{\sqrt{h(X_i)}} + \sum_{j=1}^{k} \beta_j \frac{x_{ji}}{\sqrt{h(X_i)}} + \frac{\varepsilon_i}{\sqrt{h(X_i)}}.$$

- The transformed model satisfies the homoscedasticity assumption, allowing OLS to be applied efficiently.

Robust Standard Errors (Huber-White)

Robust Variance-Covariance Matrix:

$$V_{\text{robust}} = (X'X)^{-1} X' \hat{\Omega} X (X'X)^{-1}$$

where:

$$\hat{\Omega} = \text{diag}(\hat{\varepsilon}_i^2).$$

Implementation in Python (Statsmodels):

```
import statsmodels.api as sm
model = sm.OLS(y, X).fit(cov_type='HC0')
print(model.summary())
```

▸ What is HC0

Note:

- Corrects for unknown heteroskedasticity.
- Standard errors are more reliable for hypothesis testing.

Steps of Robust Standard Errors

1. Estimate OLS model. Compute $\hat{\beta}$ and residuals $\hat{\varepsilon}$.
2. Compute robust variance. Use $(X'X)^{-1} \sum X_i' \hat{\varepsilon}_i^2 X_i (X'X)^{-1}$.
3. Extract standard errors. Take square root of diagonal elements of variance matrix.
4. Perform hypothesis testing. Compute t-statistics using robust SEs.
5. Interpret results. If robust SEs differ from OLS SEs, heteroskedasticity affects inference.

Clustered Standard Errors (VCE Cluster)

Use when errors are correlated within groups (e.g., individuals within firms, students within schools). Ordinary robust errors assume independence; clustering accounts for group-level dependence. Clustered Variance-Covariance Matrix with clusters g

$$V_{\text{cluster}} = (X'X)^{-1} \left(\sum_{g=1}^{G} X_g' \hat{\varepsilon}_g \hat{\varepsilon}_g' X_g \right) (X'X)^{-1}$$

Implementation in Python (Statsmodels)

```
import statsmodels.api as sm
model = sm.OLS(y, X).fit(cov_type='cluster',
    cov_kwds={'groups': cluster_variable})
print(model.summary())
```

- Adjusts for correlation within clusters.
- More conservative standard errors than ordinary robust SEs.
- Often used in panel data and experimental studies.

Using Robust Regression when homoskedastic

- **Unbiasedness:** Coefficient estimates ($\hat{\beta}$) remain unbiased.
- **Efficiency Loss:** Robust standard errors are larger than OLS SEs.
- **Inference Impact:**
  - t-statistics decrease.
  - p-values increase (harder to reject H_0).
  - Confidence intervals widen.
- **Rule:** Use robust SEs when heteroskedasticity is suspected. The best practice is to present both OLS and robust regression results.

Chapter 2 Extension

In models with binary dependent variable: $Y \in \{0, 1\}$, the error term is not heteroskedastic in the traditional sense. However, the variance of y depends on the predicted probabilities, leading to a form of non-constant variance that is inherent to the model.

▸ Check Bernoulli and Binomial Distribution

- Linear Probability Model
- Logit Model
- Probit Model

The Linear Probability Model (LPM)

Probability of $Y = 1$ is modeled like the OLS:

$$P(Y = 1|X = x) = x'\beta$$

- $E(Y|X) = 0.P(Y = 0|X) + 1.P(Y = 1|X) = x'\beta$.
- Estimated using Ordinary Least Squares (OLS).
- β_j represents the change in the probability of $P(Y = 1|X)$ for a unit change in X_j.
- Example: $\hat{\beta}_j = 0.05$ implies a 1-unit increase in X_j is associated with an increases the probability of $Y = 1$ by 5 percentage points, *cateris paribus*.

LPM Issues

LPM Issues:

- Predicted probabilities can be outside the $[0,1]$ range.
  Because $0 \leq P(Y=1|X) = X\beta \leq 1$ not always satisfied.
- Heteroskedasticity as the error variance depends on X:

$$\text{Var}(\varepsilon_i|X) = P(Y=1|X)(1 - P(Y=1|X)).$$

- ε is either $1 - x'\beta$ if $Y = 1$ or $-x'\beta$ if $Y = 0$, so ε is Binomial instead of normally distributed so standard errors are incorrect unless corrected. Thus, t and F tests are invalid.

LPM Alternatives

Possible solutions:

- Use robust standard errors to correct for heteroskedasticity.
- Use Logit or Probit models to ensure probabilities remain between $[0, 1]$.
- Consider truncated LPM where predictions are restricted within bounds.

The Logit Model

- **A sigmoid function** produces an S-shaped curve. e.g., the logistic function:

$$\sigma(x) = \frac{1}{1 + e^{-x}} = \frac{e^x}{1 + e^x} = 1 - \sigma(-x).$$

- Properties:
  - Output range: $(0, 1)$.
  - Smooth and differentiable.
  - The inverse of the logistic function is the logit $\text{logit}(p) = \sigma^{-1}(p) = \ln\left(\frac{p}{1-p}\right)$
- Used in logistic regression, neural networks, and probability modeling.

Logit Model (Contd.)

- Probability of $Y = 1$ is modeled using the logistic function:

$$p = P(Y = 1|X = x) = \frac{1}{1 + e^{-X\beta}} = \frac{e^{X\beta}}{1 + e^{X\beta}}$$

- Gives the log of the odds ratio (odds of $Y = 1$) or the logit:

$$\ln\left(\frac{p}{1-p}\right) = X\beta \quad \text{for} \quad p \in (0,1)$$

- Log of odds ratio is linear in parameters in the Logit model.
- Logit is an example of a Generalized Linear Model (GLM), where the **link function** is the logit function that relates the expected value of Y to the predictors X.

Logit Model (Contd.)

Probability Model: $p = \frac{e^{X\beta}}{1+e^{X\beta}}, \quad 1 - p = \frac{1}{1+e^{X\beta}}$

Given $Y_i \sim \text{Bernoulli}(p_i)$, the likelihood function is:

$$L(\beta) = \prod_{i=1}^{n} p_i^{Y_i}(1 - p_i)^{1-Y_i}$$

Log-likelihood: $\ell(\beta) = \sum_{i=1}^{n} \left[Y_i X_i \beta - \ln(1 + e^{X_i\beta}) \right].$

First-Order Condition: $\frac{\partial \ell(\beta)}{\partial \beta} = \sum_{i=1}^{n} X_i \left[Y_i - \frac{e^{X_i\beta}}{1+e^{X_i\beta}} \right] = 0.$

The equation is nonlinear, it is solved using numerical methods.

Logit Model Example

A Python example

```
import numpy as np
import statsmodels.api as sm
np.random.seed(1234)
n = 100  # Sample size
X = np.random.randn(n, 3)  # 3 predictors
X = sm.add_constant(X)  # Add intercept
beta = np.array([1.5, -2.0, 0.5, 1.0])
log_odds = X @ beta
p = 1 / (1 + np.exp(-log_odds))  # Sigmoid function
y = (np.random.rand(n) < p).astype(int)  # Make binary
model = sm.Logit(y, X).fit()
print(model.summary())
```

Logit Model Estimates

Variable	**Coef.**	**Std. Err.**	**z**	$P > \lvert z \rvert$	**[95% CI]**
Constant	1.2870	0.354	3.639	0.000	[0.594, 1.980]
x_1	-2.2326	0.521	-4.285	0.000	[-3.254, -1.211]
x_2	0.7849	0.349	2.249	0.024	[0.101, 1.469]
x_3	1.1972	0.374	3.198	0.001	[0.463, 1.931]

Model Fit: Pseudo $R^2 = 0.4277$, Log-Likelihood = -35.432.

- The intercept represents the log-odds of success ($Y = 1$) when all predictors are zero.
- A one-unit increase in x_1 decreases the log-odds of success by 2.2326.
- Log-odds can take any value in $(-\infty, \infty)$. If log-odds > 0, then $P(Y = 1|X) > 0.5$ (more likely to happen) and vice versa.

Learn about Pesudo R squared

Logit Model Inference

- No explicit error term: The randomness comes from the Bernoulli-distributed response variable $Y \sim \text{Bernoulli}(p)$.
- Variance estimation for $\hat{\beta}$: Based on the Fisher Information Matrix:

$$\text{Var}(\hat{\beta}) = (X'WX)^{-1}, \quad W_i = p_i(1 - p_i).$$

- t-statistics: Since the model is estimated via Maximum Likelihood: $t_k = \frac{\hat{\beta}_k}{\text{SE}(\hat{\beta}_k)}$ follows a standard normal distribution (not a t-distribution).
- p-values: Computed from the normal distribution:

$$p_j = 2 \times (1 - \Phi(|t_k|))$$

where Φ is the standard normal CDF.

Logit Model Assumptions

- Binary outcome: $Y_i \in \{0, 1\}$ for $i = 1, 2, \ldots, n$.
- Linearity in log-odds: $P(Y_i = 1|X_i) = \frac{e^{X_i\beta}}{1+e^{X_i\beta}}$

$$\ln\left(\frac{P(Y_i = 1|X_i)}{1 - P(Y_i = 1|X_i)}\right) = X_i\beta.$$

- Independence: $P(Y_1, \ldots, Y_n|X_1, \ldots, X_n) = \prod_{i=1}^{n} P(Y_i|X_i)$.
- No multicollinearity: $\text{rank}(X) = K$.
- Large n relative to K.
- No influential outliers: Check leverage and Cook's distance.
- Correct specification of the link function and no omitted variables.

Probability Model (Probit) and Link Function

The probability that $Y = 1$ given X is:

$$P(Y = 1|X) = \Phi(X\beta),$$

where $\Phi(z)$ is the cumulative distribution function (CDF) of the standard normal distribution:

$$\Phi(z) = \int_{-\infty}^{z} \frac{1}{\sqrt{2\pi}} e^{-t^2/2} dt.$$

Probit Link Function:

$$\Phi^{-1}(P(Y = 1|X)) = X\beta.$$

This ensures that probabilities remain between 0 and 1.

Probit Model Derivation

Given $Y_i \sim \text{Bernoulli}(p_i)$, the likelihood function is:

$$L(\beta) = \prod_{i=1}^{n} \Phi(X_i\beta)^{Y_i}[1 - \Phi(X_i\beta)]^{1-Y_i}.$$

$$\ell(\beta) = \sum_{i=1}^{n} \left[Y_i \ln \Phi(X_i\beta) + (1 - Y_i) \ln(1 - \Phi(X_i\beta))\right].$$

The score function (first derivative of the log-likelihood) is set$= 0$

$$\frac{\partial \ell(\beta)}{\partial \beta} = \sum_{i=1}^{n} X_i \frac{Y_i - \Phi(X_i\beta)}{\Phi(X_i\beta)} = 0.$$

Solved using numerical methods like Newton-Raphson.

Probit Model Interpretation

The estimated coefficient $\hat{\beta}_j$ represents the change in the z-score (standard normal units) per unit change in X_j:

$$\frac{\partial \Phi^{-1}(P(Y = 1|X))}{\partial X_j} = \beta_j.$$

Marginal Effects:

$$\frac{\partial P(Y = 1|X)}{\partial X_j} = \Phi(X\beta)\beta_j.$$

Since $\Phi(X\beta)$ varies with X, marginal effects are not constant.

Probit Model Example

```
import numpy as np
import statsmodels.api as sm
from scipy.stats import norm  # Import normal CDF
np.random.seed(1234)
n = 100  # Sample size
X = np.random.randn(n, 3)  # 3 predictors
X = sm.add_constant(X)  # Add intercept
beta = np.array([1.5, -2.0, 0.5, 1.0])
z = X @ beta
p = norm.cdf(z)  # Normal CDF for Probit
y = (np.random.rand(n) < p).astype(int)  # Make binary
model = sm.Probit(y, X).fit()
print(model.summary())
z_mean = np.dot(X.mean(axis=0), beta_hat)
phi_mean = norm.pdf(z_mean)
print( phi_mean * beta_hat[1]) #marginal_effect_x1
```

Probit Regression Summary

Variable	Coef.	Std. Err.	z-value	P$>|z|$
const	2.0205	0.519	3.894	0.000
x1	-3.7412	0.968	-3.863	0.000
x2	0.8548	0.348	2.458	0.014
x3	1.5706	0.464	3.388	0.001

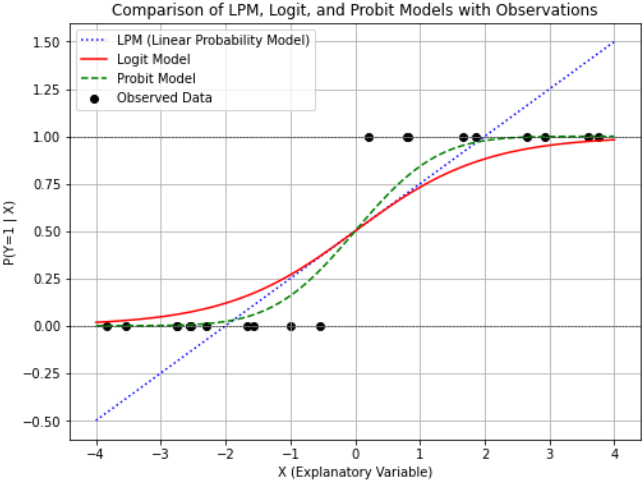
Interpretation of $\hat{\beta}_1$: A one-unit increase in x_1 is significantly associated with 3.7412 unit decreases in the z-score or probit index of y.

Marginal Effects Interpretation: A one-unit increase in x_1 is associated with a decrease in the predicted probability of $P(Y = 1)$, evaluated at the mean of X, by -0.0664 or 6.6 percentage points.

Probit Model Assumptions

- Binary dependent variable: $Y \in \{0, 1\}$.
- Probability modeled as $P(Y = 1|X) = \Phi(X\beta)$, where $\Phi(\cdot)$ is the CDF of the standard normal distribution.
- Linear in parameters with the link function: $X\beta$ enters linearly.
- Error term follows a standard normal distribution: $\varepsilon \sim N(0, 1)$.

Comparison of Popular Models for Binary Y



Latent Variable Approach

Assume an unobservable latent variable y^* follows the regression model:

$$y^* = b_0 + X'b + \varepsilon, \quad \varepsilon \mid X \sim U(-a, a).$$

The probability of observing $y = 1$ is:

$$P(y = 1 \mid X) = P(y^* > 0 \mid X) = P(b_0 + X'b + \varepsilon > 0 \mid X).$$

Note that the latent variable y^* is not binary, but the observed variable y is binary.

LPM from Latent variable

$$P(y = 1 \mid X) = P(\varepsilon > -b_0 - X'b \mid X).$$

Using the uniform CDF:

$$P(y = 1 \mid X) = 1 - F_{\varepsilon|X}(-b_0 - X'b).$$

Since $F_{\varepsilon|X}(\varepsilon) = \frac{\varepsilon+a}{2a}$, we get:

$$P(y = 1 \mid X) = \frac{b_0 + a}{2a} + \frac{X'b}{2a}.$$

This is the Linear Probability Model:

$$P(y = 1 \mid X) = \beta_0 + X'\beta.$$

where:

$$\beta_0 = \frac{b_0 + a}{2a}, \quad \beta = \frac{b}{2a}.$$

Logit from Latent variable

If $\varepsilon \sim \text{Logistic}(0, 1)$, the logistic CDF is:

$$F_{\varepsilon|X}(\varepsilon) = \frac{e^{\varepsilon}}{1 + e^{\varepsilon}}.$$

Then:

$$P(y = 1 \mid X) = 1 - F_{\varepsilon|X}(-b_0 - X'b).$$

Substituting the logistic CDF:

$$P(y = 1 \mid X) = \frac{e^{b_0 + X'b}}{1 + e^{b_0 + X'b}}.$$

This is the logit model:

$$P(y = 1 \mid X) = \frac{1}{1 + e^{-(\beta_0 + X'\beta)}}.$$

Probit from Latent variable

If $\varepsilon \sim N(0,1)$, the normal CDF is:

$$F_{\varepsilon|X}(\varepsilon) = \Phi(\varepsilon).$$

Then:

$$P(y = 1 \mid X) = 1 - F_{\varepsilon|X}(-b_0 - X'b).$$

Using the normal CDF:

$$P(y = 1 \mid X) = \Phi(b_0 + X'b).$$

This is the probit model:

$$P(y = 1 \mid X) = \Phi(\beta_0 + X'\beta).$$

Miscellaneous

Projection in OLS

In OLS, the residuals are given by:

$$e = y - X\hat{\beta} \tag{13}$$

Substituting $\hat{\beta} = (X'X)^{-1}X'y$:

$$e = y - X(X'X)^{-1}X'y \tag{14}$$

Defining the **residual maker** matrix:

$$M = I - X(X'X)^{-1}X' \tag{15}$$

The residuals can be expressed as:

$$e = My \tag{16}$$

Projection in OLS (contd.)

The fitted values from the regression:

$$\hat{y} = X\hat{\beta} = X(X'X)^{-1}X'y \tag{17}$$

Define the **projection matrix** or hat matrix:

$$P = X(X'X)^{-1}X' \tag{18}$$

Note: $X(X'X)^{-1}X' = I$ only if X is a square, full-rank invertible matrix. Thus,

$$\hat{y} = Py \tag{19}$$

where P projects y onto the column space of X.

Properties of Projection Matrices

The projection matrix P and the residual maker matrix M satisfy:

- $P^2 = P$ (idempotent)
- $M^2 = M$ (idempotent)
- $P + M = I$
- $PM = 0$ (orthogonal)

These properties ensure that the residuals are orthogonal to the fitted values:

$$e'\hat{y} = y'M'Py = 0 \tag{20}$$

$$y = Py + My = \text{projection} + \text{residual} \tag{21}$$

Bernoulli and Binomial Distributions

Bernoulli Distribution: e.g., Flipping a coin with probability p for head. $Y \sim \text{Bernoulli}(p)$

$$P(Y = y) = \begin{cases} p, & y = 1, \\ 1 - p, & y = 0. \end{cases}$$

$$E(Y) = p, \quad \text{Var}(Y) = p(1 - p).$$

Binomial Distribution: e.g., Flipping n coins with k heads. $Y \sim \text{Binomial}(n, p)$

$$P(Y = k) = \binom{n}{k} p^k (1 - p)^{n-k}, \quad k = 0, 1, \ldots, n.$$

$$E(Y) = np, \quad \text{Var}(Y) = np(1 - p).$$

Go back to Chapter 2 Extension

Pseudo R^2 (McFadden's R^2)

Formula:

$$R^2 = 1 - \frac{\ell(\hat{\beta})}{\ell(\beta_0)}$$

where:

- $\ell(\hat{\beta})$ = Log-likelihood of the fitted model.
- $\ell(\beta_0)$ = Log-likelihood of the null (intercept-only) model.

Interpretation:

- $R^2 \approx 1 \rightarrow$ Model has strong explanatory power.
- $R^2 \approx 0 \rightarrow$ Model performs similarly to a null model.

Go back to Logit

Asymptotic Normality of MLE

$$\sqrt{n}(\hat{\theta} - \theta_0) \xrightarrow{d} N(0, I(\theta_0)^{-1})$$

where:

- θ_0 is the true parameter.
- $I(\theta_0)$ is the Fisher Information Matrix:

$$I(\theta_0) = -E\left[\frac{\partial^2 \ell(\theta)}{\partial \theta^2}\right].$$

Why is MLE normal?

- Central Limit Theorem (CLT): The score function sums to normality.
- Law of Large Numbers (LLN): Fisher information stabilizes variance.

Score Function in Maximum Likelihood Estimation

The score function is the first derivative of the log likelihood that measures the sensitivity of the log-likelihood function:

$$S(\theta) = \frac{\partial \ell(\theta)}{\partial \theta}$$

MLE First-Order Condition: $S(\hat{\theta}) = 0 \quad \Rightarrow \quad$ MLE solution.

Example: Normal Distribution has a score function that is asymptotically normal if you use the sample mean for μ.

$$\ell(\mu) = -\frac{n}{2} \ln(2\pi\sigma^2) - \frac{1}{2\sigma^2} \sum_{i=1}^{n} (Y_i - \mu)^2$$

$$S(\mu) = \frac{1}{\sigma^2} \sum_{i=1}^{n} (Y_i - \mu).$$

Fisher Information: $\mathrm{Var}[S(\theta)] = I(\theta) = -E\left[\frac{\partial^2 \ell(\theta)}{\partial \theta^2}\right]$.

Key Property: $E[S(\theta)] = 0$.

Why Is the LRT Statistic Chi-Square?

Likelihood Ratio Test (LRT) Statistic:

$$\Lambda = -2\left(\ell_0 - \ell_1\right) \sim \chi^2_{df}$$

Idea:

- The log-likelihood function is approximated by a quadratic form.
- The score function (gradient of log-likelihood) is asymptotically normal.
- The difference in log-likelihoods follows a sum of squared normal variables.

Wilks' Theorem:

$$-2(\ell_0 - \ell_1) \sim \chi^2_{df} \quad \text{(asymptotically)}$$

Go back to LR

Probability Limit (plim)

- Probability Limit (plim) is the limit in probability of a sequence of random variables.
- Denoted as:

$$\text{plim} X_n = c \quad \text{if for any } \varepsilon > 0, \ \lim_{n \to \infty} P(|X_n - c| > \varepsilon) \to 0$$

- It is closely related to the Law of Large Numbers.

Properties of plim

- Linearity:

$$\text{plim}(aX_n + bY_n) = a\text{plim}(X_n) + b\text{plim}(Y_n)$$

- Product Rule:

$$\text{plim}(X_n Y_n) = \text{plim}(X_n) \cdot \text{plim}(Y_n) \quad \text{if both plims exist}$$

- Inverse Property:

$$\text{plim}\left(\frac{1}{X_n}\right) = \frac{1}{\text{plim}(X_n)} \quad \text{if plim}(X_n) \neq 0$$

- Continuous Mapping: If $g(\cdot)$ is continuous,

$$\text{plim}(g(X_n)) = g(\text{plim}(X_n))$$

Key Result: plim of $\frac{1}{n}X'X$

- Consider the matrix X of observations with dimensions $n \times k$.
- The sample second-moment matrix is:

$$\frac{1}{n}X'X$$

- By the Law of Large Numbers:

$$\text{plim}\left(\frac{1}{n}X'X\right) = Q$$

- Q is the population second-moment matrix, defined as: $Q = E[X'X]$ if the data are independently and identically distributed (i.i.d.)

Intuition Behind the Result

- The matrix $\frac{1}{n}X'X$ sums up information from n observations.
- If the observations are i.i.d. and well-behaved (finite variance, etc.), the Law of Large Numbers applies.
- Intuition:

$$\frac{1}{n}X'X \to E[X'X] = Q \quad \text{as } n \to \infty$$

- Q captures the population structure of the explanatory variables.

Applications of plim and Q

- Asymptotic Properties of OLS: - The OLS estimator for β involves $\left(\frac{1}{n}X'X\right)^{-1}$. - Its asymptotic behavior depends critically on the matrix Q.
- Econometric Consistency: - Consistency of estimators relies heavily on plim properties.
- Variance-Covariance Matrices: - As n grows, $\frac{1}{n}X'X$ converges to Q, simplifying asymptotic variance calculations.

Types of Heteroskedasticity Correction

HC0, HC1, HC2, and HC3 are different types of heteroscedasticity-consistent (HC) covariance matrix estimators used in linear regression. They calculate robust standard errors when error term variance is not constant (heteroscedasticity).

- **HC0**: Original White estimator; can be unreliable in small samples.
- **HC1**: Corrects for degrees of freedom for small samples.
- **HC2**: Adjusts based on leverage points.
- **HC3**: Adjusts more conservatively for leverage, preferred for small samples.

HC0: The Original White Estimator

- HC0 is the original heteroscedasticity-consistent estimator.
- It uses residuals squared without any adjustments for sample size or leverage.
- Formula:

$$\hat{V}_{HC0} = (X'X)^{-1} \left[\sum_{i=1}^{n} \hat{\varepsilon}_i^2 x_i x_i' \right] (X'X)^{-1}$$

- Can be biased in small samples.

HC1: Degrees of Freedom Correction

- HC1 adjusts HC0 by incorporating a degrees-of-freedom correction.
- Formula:

$$\hat{V}_{HC1} = \frac{n}{n-p}(X'X)^{-1}\left[\sum_{i=1}^{n}\hat{\varepsilon}_i^2 x_i x_i'\right](X'X)^{-1}$$

- Commonly used in Stata.

HC2: Leverage Adjustment

- HC2 modifies HC0 to account for leverage points.
- Each residual is adjusted by dividing by $(1 - h_{ii})$.
- Formula:

$$\hat{V}_{HC2} = (X'X)^{-1} \left[\sum_{i=1}^{n} \frac{\hat{\varepsilon}_i^2}{1 - h_{ii}} x_i x_i' \right] (X'X)^{-1}$$

- Useful when leverage points may distort variance estimates.

Note: What are Leverage Points

Leverage points (are extreme predictor values) that measure how much influence an observation has on the fitted regression model. They are derived from the **hat matrix** $H = X(X'X)^{-1}X'$ showing variation in X in a row over all variations in X. (Recall that $E(X'X)$ is the 2nd raw moment).

- The leverage of the i-th observation is the i-th diagonal element of H:
$$h_{ii} = X_i(X'X)^{-1}X_i'$$
- h_{ii} ranges between 0 and 1, with higher values indicating greater influence.
- A point is considered high-leverage if: $h_{ii} > \frac{2K}{n}$ where K is number of predictors (including the intercept) and n is sample size (K/n is the mean leverage).

High-leverage points can disproportionately affect model estimates, making robust standard errors essential.

HC3: Conservative Leverage Adjustment

- HC3 further adjusts for leverage by dividing by $(1 - h_{ii})^2$.
- More conservative adjustment compared to HC2.
- Formula:

$$\hat{V}_{HC3} = (X'X)^{-1} \left[\sum_{i=1}^{n} \frac{\hat{\varepsilon}_i^2}{(1 - h_{ii})^2} x_i x_i' \right] (X'X)^{-1}$$

- Preferred option in small samples due to better bias correction.

When to Use Which Estimator?

- **Large Samples:** HC0 or HC1 typically suffice as sample size increases.
- **Small Samples:** HC2 or HC3 are recommended for more reliable results.
- HC3 is often considered the best all-rounder due to its conservative leverage adjustments.

HC4: Recent Development [Cribari-Neto, 2004]

HC4 adjusts the penalty dynamically based on sample size and leverage. It is particularly effective in small samples with influential points.

- Formula:

$$\hat{V}_{HC4} = (X'X)^{-1} \left[\sum_{i=1}^{n} \frac{\hat{\varepsilon}_i^2}{(1 - h_{ii})^{\delta_i}} x_i x_i' \right] (X'X)^{-1}$$

- Where:

$$\delta_i = \min \left\{ 4, \frac{n h_{ii}}{\sum_{j=1}^{n} h_{jj}} \right\}$$

- Explanation:
  - $\hat{\varepsilon}_i$: Residual for observation i
  - h_{ii}: Leverage of the i-th observation
  - δ_i: Penalty that grows with leverage and shrinks with sample size

Back to White's het.

Understanding p-values

- We construct the distribution based on: If H_0 were true, what would have been the **expected** variation in the data.
- A p-value measures how extreme the observed data is in that distribution, **assuming that H_0 is true**.
- So, p-values do not measure the probability that the null hypothesis H_0 is true or false.
- For example, a p-value of 0.03 means that if H_0 (e.g., homoskedasticity) were true, we would observe a test statistic as high as the one calculated (e.g., Calculated $\chi^2 = nR^2 = 12.11$) or higher in approximately 3 out of 100 samples (3% of the time) purely by chance.
- Therefore, if $p = 0.03 < 0.05$, we reject H_0 at the 5% significance level; but if we require a 1% level ($p < 0.01$), we fail to reject H_0 because $0.03 > 0.01$.

Numerical Optimization: Introduction to BFGS

- BFGS stands for **Broyden–Fletcher–Goldfarb–Shanno** algorithm.
- It is a **quasi-Newton method** for solving unconstrained optimization problems.
- BFGS uses an approximation of the **inverse Hessian matrix** to find the minimum.
- It is widely used because it is efficient and does not require calculating the Hessian directly.

$$\min_{\beta} f(\beta)$$

where $f(\beta)$ is a twice-differentiable function.

BFGS Mechanism

- **Gradient Descent Step:**

$$\beta_{t+1} = \beta_t - \alpha_t H_t \nabla f(\beta_t)$$

  where:
  - $\nabla f(\beta_t) =$ Gradient of the objective at iteration t.
  - $H_t =$ Approximation of the inverse Hessian.
  - $\alpha_t =$ Step size (often chosen using line search).

- **Updating the Inverse Hessian Approximation:**

$$H_{t+1} = H_t + \frac{\Delta x_t \Delta x_t'}{\Delta x_t' \Delta g_t} - \frac{H_t \Delta g_t \Delta g_t' H_t}{\Delta g_t' H_t \Delta g_t}$$

  Recall that the Hessian matrix is a square matrix of second-order partial derivatives of a scalar-valued function.
  - $\Delta x_t = \beta_{t+1} - \beta_t$
  - $\Delta g_t = \nabla f(\beta_{t+1}) - \nabla f(\beta_t)$

Advantages of BFGS

- Efficient and fast convergence for moderately-sized problems.
- Does not require computation of the full Hessian matrix.
- Works well when the function is smooth and differentiable.
- Commonly used in machine learning and econometrics.

GMM for OLS Estimation

The Generalized Method of Moments (GMM) can be used to estimate OLS coefficients by utilizing the condition:

$$E\left[X_i\varepsilon_i\right] = E\left[X_i(y_i - X_i'\beta)\right] = 0$$

Moment Conditions
$$g_n(\beta) = \frac{1}{n}\sum_{i=1}^{n} X_i\left(y_i - X_i'\beta\right)$$

In matrix notation, the sample moment conditions are $k \times 1$

$$g_n(\beta) = \frac{1}{n}X'(y - X\beta)$$

Objective Function
$$Q_n(\beta) = g_n(\beta)' W g_n(\beta)$$

with W as a $k \times k$ weighting matrix. For OLS, $W = I$ (identity).

Minimize $Q_n(\beta)$ to get: $\hat{\beta}_{GMM} = (X'X)^{-1}X'y$

Go back to OLS

What can be the optimal weighting matrix $W = W^*$?

Note that, the mean of the moment conditions is $E(g) = 0$. If we choose $W = [Var(g)]^{-1}$, then Q becomes like Z^2. Thus, under some regularity conditions,

$$Q \xrightarrow{d} \chi^2(\text{df}),$$

where df $=$ Number of moments $-$ Number of parameters.

Thus, the optimal weighting matrix in GMM is the inverse of the variance-covariance matrix of the moment conditions:

$$W^* = [\text{Var}(g)]^{-1}.$$

Because this minimizes the asymptotic variance of the GMM estimator, making it efficient.

Example of W^* in OLS

For OLS, the moment conditions are: $g = \frac{1}{n}X'(y - X\beta)$.

The mean of g is $E(g) = 0$

The variance-covariance matrix of g is: $\text{Var}(g) = \frac{1}{n^2}X'\text{Var}(y - X\beta)X$.

Under homoskedasticity, $\text{Var}(y - X\beta) = \sigma^2 I_n \Rightarrow \text{Var}(g) = \frac{\sigma^2}{n^2}X'X$.

Thus, the optimal weighting matrix is: $W^* = \left(\frac{\sigma^2}{n^2}X'X\right)^{-1} = \frac{n^2}{\sigma^2}(X'X)^{-1}$.

Key intuition:

- Q measures how well sample moments match $H_0 : E[g] = 0$.
- Q is a quadratic form in standardized moments, leading to χ^2.
- Note that for OLS we have k regressors and k moment conditions so $df = 0$. OLS is exactly identified.
- If Number of moments $>$ Number of parameters $\Rightarrow$ the model is over-identified (example, instrumental variables or additional restrictions).
- If Number of moments $<$ Number of parameters $\Rightarrow$ the model is under-identified (more unknown parameters than equations).

Regularity Conditions for GMM

The following conditions ensure GMM estimators are consistent, asymptotically normal, and that $Q \xrightarrow{d} \chi^2$:

1. **Valid moments**: $E[g(X_i, \beta)] = 0$.
2. **Identification**: Unique solution at $\beta = \beta_0$.
3. **Smoothness**: $g(X_i, \beta)$ is continuously differentiable in β.
4. **Finite moments**: $E[g(X_i, \beta)g(X_i, \beta)'] < \infty$.
5. **CLT**: $\sqrt{n}\, g_n(\beta_0) \xrightarrow{d} N(0, \Sigma)$.
6. **Consistent W**: $W_n \xrightarrow{p} W^* = [\mathrm{Var}(g)]^{-1}$.
7. **Compact parameter space**: $\beta \in \Theta$.
8. **Rank condition**: $G = E\left[\frac{\partial g(X_i, \beta)}{\partial \beta}\right]$ has full rank at β_0.
9. **Uniform LLN**: Uniform convergence of sample moments.
10. **No perfect multicollinearity**: among regressors/instruments.

These ensure $Q = g' W g \xrightarrow{d} \chi^2(\mathrm{df})$ under $H_0 : E[g] = 0$.

Why dont we use GMM for OLS $\hat{\beta}$

- Under classical assumptions OLS is BLUE and more efficient.
- OLS is linear but GMM is more flexible based on moments.
- OLS has closed form solution, GMM is numerical so computationally intensive.
- GMM can be more efficient when heteroskedasticity or autocorrelation is present ($W \neq I$).

Conclusion to Lecture 1

With the labs posted on GitHub (github.com/Badruddoza), we learned the derivation and implementation from scratch of the estimator $\hat{\beta}$ in a linear model using the following approaches:

- Ordinary Least Squares (OLS)
- Maximum Likelihood Estimation (MLE)
- Generalized Method of Moments (GMM)

We also explored the theory and practical applications of:

- Heteroskedastic errors (GLS and FGLS)
- The Linear Probability Model (LPM)
- The Logit Model
- The Probit Model

In the next lecture, we will discuss the violation of a critical OLS assumption: $E(\varepsilon|X) = 0$.