Gov 2000 - 9. Multiple Linear Regression: Regression in Matrix Form

Matthew Blackwell
Harvard University
mblackwell@gov.harvard.edu

Where are we? Where are we going?

- Last few weeks: regression estimation and inference with one and two independent variables
- This week: the general regression model with arbitrary covariates
- Next few weeks: what to do when the regression assumptions go wrong

Nunn & Wantchekon

- Are there long-term, persistent effects of slave trade on Africans today?
- Basic idea: compare levels of interpersonal trust (Y_i) across different levels of historical slave exports for a respondent's ethnic group
- Problem: ethnic groups and respondents might differ in their interpersonal trust in ways that correlate with the severity of slavery
- One solution: try to control for relevant differences between groups via multiple regression:

VOL. 101 NO. 7 NUNN AND WANTCHEKON: THE ORIGINS OF MISTRUST IN AFRICA 3231

III. Estimating Equations and Empirical Results

A. *OLS Estimates*

We begin by estimating the relationship between the number of slaves that were taken from an individual's ethnic group and the individual's current level of trust. Our baseline estimating equation is:

$$(1) \quad trust_{i,e,d,c} = \alpha_c + \beta\, slave\ exports_e + \mathbf{X}'_{i,e,d,c}\boldsymbol{\Gamma} + \mathbf{X}'_{d,c}\boldsymbol{\Omega} + \mathbf{X}'_e\boldsymbol{\Phi} + \varepsilon_{i,e,d,c},$$

where i indexes individuals, e ethnic groups, d districts, and c countries. The variable $trust_{i,e,d,c}$ denotes one of our five measures of trust, which vary across individuals. α_c denotes country fixed effects, which are included to capture country-specific factors, such as government regulations, that may affect trust (e.g., Philippe Aghion et al. 2010; Aghion, Algan, and Cahuc 2008). $slave\ exports_e$ is a measure of the number of slaves taken from ethnic group e during the slave trade. (We discuss this variable in more detail below.) Our coefficient of interest is β, the estimated relationship between the slave exports of an individual's ethnic group and the individual's current level of trust.

- Our goal: Be able to understand what's we're saying in this equation.

```
nunn <- foreign::read.dta("Nunn_Wantchekon_AER_2011.dta")
mod <- lm(trust_neighbors ~ exports + age + male + urban_dum + malaria_ecology, data = nunn)
summary(mod)
```

```
##
## Call:
## lm(formula = trust_neighbors ~ exports + age + male + urban_dum +
##     malaria_ecology, data = nunn)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.5954 -0.7491  0.1440  0.8735  1.9964
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      1.503e+00  2.183e-02  68.844   <2e-16 ***
## exports         -1.021e-03  4.094e-05 -24.935   <2e-16 ***
## age              5.045e-03  4.724e-04  10.680   <2e-16 ***
## male             2.784e-02  1.382e-02   2.015   0.0439 *
```

```
## urban_dum       -2.739e-01  1.435e-02 -19.079   <2e-16 ***
## malaria_ecology  1.941e-02  8.712e-04  22.279   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9778 on 20319 degrees of freedom
##   (1497 observations deleted due to missingness)
## Multiple R-squared:  0.06039,    Adjusted R-squared:  0.06016
## F-statistic: 261.2 on 5 and 20319 DF,  p-value: < 2.2e-16
```

```
head(model.matrix(mod), 20)
```

```
##    (Intercept)  exports age male urban_dum malaria_ecology
## 1            1 854.9581  40    0         0        28.14704
## 2            1 854.9581  25    1         0        28.14704
## 3            1 854.9581  38    1         1        28.14704
## 4            1 854.9581  37    0         1        28.14704
## 5            1 854.9581  31    1         0        28.14704
## 6            1 854.9581  45    0         0        28.14704
## 7            1 854.9581  20    1         0        28.14704
## 8            1 854.9581  31    0         0        28.14704
## 9            1 854.9581  24    1         0        28.14704
## 10           1 854.9581  52    0         0        28.14704
## 11           1 854.9581  29    1         0        28.14704
## 12           1 854.9581  18    0         0        28.14704
## 13           1 854.9581  50    1         0        28.14704
## 14           1 854.9581  35    0         0        28.14704
## 15           1 854.9581  47    1         0        28.14704
## 16           1 854.9581  29    0         0        28.14704
## 17           1 854.9581  21    1         0        28.14704
## 18           1 854.9581  23    0         0        28.14704
## 19           1 854.9581  25    1         0        28.14704
## 20           1 854.9581  29    0         0        28.14704
```

```
dim(model.matrix(mod))
```

```
## [1] 20325     6
```

Why matrices and vectors?

- Here's one way to write the full multiple regression model:

$$y_i = \beta_0 + x_{i1}\beta_1 + x_{i2}\beta_2 + \cdots + x_{iK}\beta_K + u_i$$

- Notation is going to get needlessly messy as we add variables.

MATRIX ALGEBRA REVIEW

Matrices and vectors

- A matrix is just a rectangular array of numbers. We say that a matrix is $n \times K$ ("n by K") if it has n rows and K columns.
- Uppercase bold denotes a matrix:

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1K} \\ a_{21} & a_{22} & \cdots & a_{2K} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nK} \end{bmatrix}$$

- We will often need to refer to some generic entry (or cell) of a matrix and we can do this with a_{ik} where this is the entry in row i and column k.
- There is nothing special about these matrices. They are basically just like spreadsheets in Excel or the like. It's a way to group numbers.

Examples of matrices

- One example of a matrix that we'll use a lot is the **design matrix**, which has a column of ones, and then each of the subsequent columns is each independent variable in the regression.

$$\mathbf{X} = \begin{bmatrix} 1 & \text{exports}_1 & \text{age}_1 & \text{male}_1 \\ 1 & \text{exports}_2 & \text{age}_2 & \text{male}_2 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & \text{exports}_n & \text{age}_n & \text{male}_n \end{bmatrix}$$

```
head(model.matrix(mod), 20)
```

```
##    (Intercept)  exports age male urban_dum malaria_ecology
## 1            1 854.9581  40    0         0        28.14704
## 2            1 854.9581  25    1         0        28.14704
## 3            1 854.9581  38    1         1        28.14704
## 4            1 854.9581  37    0         1        28.14704
## 5            1 854.9581  31    1         0        28.14704
## 6            1 854.9581  45    0         0        28.14704
## 7            1 854.9581  20    1         0        28.14704
## 8            1 854.9581  31    0         0        28.14704
## 9            1 854.9581  24    1         0        28.14704
## 10           1 854.9581  52    0         0        28.14704
## 11           1 854.9581  29    1         0        28.14704
## 12           1 854.9581  18    0         0        28.14704
## 13           1 854.9581  50    1         0        28.14704
## 14           1 854.9581  35    0         0        28.14704
## 15           1 854.9581  47    1         0        28.14704
## 16           1 854.9581  29    0         0        28.14704
## 17           1 854.9581  21    1         0        28.14704
## 18           1 854.9581  23    0         0        28.14704
## 19           1 854.9581  25    1         0        28.14704
## 20           1 854.9581  29    0         0        28.14704
```

```
dim(model.matrix(mod))
```

```
## [1] 20325     6
```

Vectors

- A **vector** is just a matrix with only one row or one column.
- A **row vector** is a vector with only one row, sometimes called a $1 \times K$ vector:

$$\boldsymbol{\alpha} = \begin{bmatrix} \alpha_1 & \alpha_2 & \alpha_3 & \cdots & \alpha_K \end{bmatrix}$$

- A **column vector** is a vector with one column and more than one row. Here is a $n \times 1$ vector:

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

- Unless otherwise stated, we'll assume that a vector is column vector and vectors will be written with lowercase bold lettering ($\mathbf{b}$)

Vector examples

- One really common vector that we will work with are individual variables, such as the dependent variable, which we will represent as $\mathbf{y}$:

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

Vectors in R

- We can always get a column or row vector from a data frame or matrix in R using the usual subset rules. Note, though, that R always prints a vector in row form, even if it is a column in the original data:

```
model.matrix(mod)[1,]
```

```
##    (Intercept)        exports            age           male
##        1.00000      854.95807       40.00000        0.00000
##      urban_dum malaria_ecology
##        0.00000        28.14704
```

```
head(nunn$trust_neighbors)
```

```
## [1] 3 3 0 0 1 1
```

- **Gotcha** In R, vectors aren't the same as matrices. If try to use `dim()` on a vector, R is confused:

```
dim(nunn$trust_neighbors)
```

```
## NULL
```

- Vectors in R are special constructs and you have to use `length()` to see how many entries there are in the vector:

```r
length(nunn$trust_neighbors)
```

```
## [1] 21822
```

- You can convert vectors to be matrices using `as.matrix()` (with one row or one column, like our definition), but beware that R assumes all vectors are column vectors:

```r
dim(as.matrix(nunn$trust_neighbors))
```

```
## [1] 21822     1
```

Transpose

- There are many operations we'll do on vectors and matrices, but one is very fundamental: the transpose.
- The **transpose** of a matrix $\mathbf{A}$ is the matrix created by switching the rows and columns of the data and is denoted $\mathbf{A}'$. That is, the kth column becomes the kth row.

$$\mathbf{Q} = \begin{bmatrix} q_{11} & q_{12} \\ q_{21} & q_{22} \\ q_{31} & q_{32} \end{bmatrix} \quad \mathbf{Q}' = \begin{bmatrix} q_{11} & q_{21} & q_{31} \\ q_{12} & q_{22} & q_{32} \end{bmatrix}$$

- If $\mathbf{A}$ is $j \times k$, then $\mathbf{A}'$ will be $k \times j$.

Transposing vectors

- Transposing will turn a $k \times 1$ column vector into a $1 \times k$ row vector and vice versa:

$$\boldsymbol{\omega} = \begin{bmatrix} 1 \\ 3 \\ 2 \\ -5 \end{bmatrix} \qquad \boldsymbol{\omega}' = \begin{bmatrix} 1 & 3 & 2 & -5 \end{bmatrix}$$

Transposing in R

```
a <- matrix(1:6, ncol = 3, nrow = 2)
a
```

```
##      [,1] [,2] [,3]
## [1,]    1    3    5
## [2,]    2    4    6
```

```
t(a)
```

```
##      [,1] [,2]
## [1,]    1    2
## [2,]    3    4
## [3,]    5    6
```

Write matrices as vectors

- Sometimes it will be easier to refer to matrices as a group of column or row vectors:
- As a row vector:

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{bmatrix} = \begin{bmatrix} \mathbf{a}_1' \\ \mathbf{a}_2' \end{bmatrix}$$

- with row vectors $\mathbf{a}_1' = \begin{bmatrix} a_{11} & a_{12} & a_{13} \end{bmatrix}$ $\mathbf{a}_2' = \begin{bmatrix} a_{21} & a_{22} & a_{23} \end{bmatrix}$
- Or we can define it in terms of column vectors:

$$\mathbf{B} = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \\ b_{31} & b_{32} \end{bmatrix} = \begin{bmatrix} \mathbf{b_1} & \mathbf{b_2} \end{bmatrix}$$

  where $\mathbf{b_1}$ and $\mathbf{b_2}$ represent the columns of $\mathbf{B}$.

- It should be clear what is what: matrices defined by column will be written horizontally, whereas matrices defined by row will be written vertically with transposes.
- Also, we'll use k and j as subscripts for columns of a matrix: $\mathbf{x}_j$ or $\mathbf{x}_k$, whereas i and t will be used for rows $\mathbf{x}_i'$.

Addition and subtraction

- How do we add or subtract matrices and vectors?
- First, the matrices/vectors need to be **comformable**, meaning that the dimensions have to be the same.
- Let $\mathbf{A}$ and $\mathbf{B}$ both be 2×2 matrices. Then, let $\mathbf{C} = \mathbf{A} + \mathbf{B}$, where we add each cell together:

$$\mathbf{A}+\mathbf{B} = \left[\begin{array}{cc} a_{11} & a_{12} \\ a_{21} & a_{22} \end{array}\right] + \left[\begin{array}{cc} b_{11} & b_{12} \\ b_{21} & b_{22} \end{array}\right] = \left[\begin{array}{cc} a_{11}+b_{11} & a_{12}+b_{12} \\ a_{21}+b_{21} & a_{22}+b_{22} \end{array}\right] = \left[\begin{array}{cc} c_{11} & c_{12} \\ c_{21} & c_{22} \end{array}\right] = \mathbf{C}$$

Scalar multiplication

- A scalar is just a single number: you can think of it sort of like a 1 by 1 matrix.
- When we multiply a scalar by a matrix, we just multiply each element/cell by that scalar:

$$\alpha\mathbf{A} = \alpha \left[\begin{array}{cc} a_{11} & a_{12} \\ a_{21} & a_{22} \end{array}\right] = \left[\begin{array}{cc} \alpha \times a_{11} & \alpha \times a_{12} \\ \alpha \times a_{21} & \alpha \times a_{22} \end{array}\right]$$

The linear model with new notation

- Remember that we wrote the linear model as the following for all $i \in [1, \ldots, n]$:

$$y_i = \beta_0 + x_i\beta_1 + z_i\beta_2 + u_i$$

- Imagine we had an n of 4. We could write out each formula:

$$\begin{aligned} y_1 &= \beta_0 + x_1\beta_1 + z_1\beta_2 + u_1 \quad \text{(unit 1)} \\ y_2 &= \beta_0 + x_2\beta_1 + z_2\beta_2 + u_2 \quad \text{(unit 2)} \\ y_3 &= \beta_0 + x_3\beta_1 + z_3\beta_2 + u_3 \quad \text{(unit 3)} \\ y_4 &= \beta_0 + x_4\beta_1 + z_4\beta_2 + u_4 \quad \text{(unit 4)} \end{aligned}$$

- We can write this as:

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \beta_0 + \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \beta_1 + \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{bmatrix} \beta_2 + \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix}$$

- Hopefully it's clear in this notation that the column vector of the outcomes is a linear combination of the independent variables and the error, with the β coefficients acting as the weights.
- Can we write this in a more compact form? Yes! Let $\mathbf{X}$ and $\boldsymbol{\beta}$ be the following:

$$\underset{(4\times3)}{\mathbf{X}} = \begin{bmatrix} 1 & x_1 & z_1 \\ 1 & x_2 & z_2 \\ 1 & x_3 & z_3 \\ 1 & x_4 & z_4 \end{bmatrix} \qquad \underset{(3\times1)}{\boldsymbol{\beta}} = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \end{bmatrix}$$

Matrix multiplication by a vector

- We will define multiplication of a matrix by a vector in the following way:

$$\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \beta_0 + \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \beta_1 + \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{bmatrix} \beta_2 = \mathbf{X}\boldsymbol{\beta}$$

- Thus, multiplication of a matrix by a vector is just the **linear combination** of the columns of the matrix with the vector elements as weights/coefficients.
- And the left-hand side here only uses scalars times vectors, which is easy!
- In general, let's say that we have a $n \times K$ matrix $\mathbf{A}$ and a $K \times 1$ column vector $\mathbf{b}$ (notice that the number of columns of the matrix is the same as the number of rows of the vector)
- Let $\mathbf{a}_k$ be the kth column of $\mathbf{A}$. Then we can write:

$$\underset{(n\times1)}{\mathbf{c}} = \mathbf{A}\mathbf{b} = b_1\mathbf{a}_1 + b_2\mathbf{a}_2 + \cdots + b_K\mathbf{a}_K$$

Back to regression

- Thus, now let $\mathbf{X}$ be the $n \times (K+1)$ matrix of independent variables and $\boldsymbol{\beta}$ be the $(K+1) \times 1$ column vector of coefficients. Then:

$$\mathbf{X}\boldsymbol{\beta} = \beta_0 + \beta_1 \mathbf{x}_1 + \beta_2 \mathbf{x}_2 + \cdots + \beta_K \mathbf{x}_K$$

- Thus, we can compactly write the linear model as the following:

$$\underset{(n\times 1)}{\mathbf{y}} = \mathbf{X}\boldsymbol{\beta} + \underset{(n\times 1)}{\mathbf{u}}$$

Matrix multiplication

- What if, instead of a column vector b, we have a matrix $\mathbf{B}$ with dimensions $K \times M$.
- How do we do multiplication like so $\mathbf{C} = \mathbf{AB}$?
- Each column of the new matrix is just matrix by vector multiplication:

$$\mathbf{C} = [\mathbf{c}_1 \quad \mathbf{c}_2 \quad \cdots \quad \mathbf{c}_M] \qquad \mathbf{c}_k = \mathbf{A}\mathbf{b}_k$$

- Thus, each column of $\mathbf{C}$ is a linear combination of the columns of $\mathbf{A}$.

Special multiplications

- The **inner product** of a two column vectors $\mathbf{a}$ and $\mathbf{b}$ (of equal dimension, $K \times 1$) is just the transpose of the first multiplied by the second:

$$\mathbf{a}'\mathbf{b} = a_1 b_1 + a_2 b_2 + \cdots + a_K b_K$$

- This is a special case of the stuff above since $\mathbf{a}'$ is a matrix with K columns and just 1 row, so the "columns" of $\mathbf{a}'$ are just scalars.
- Example: let's say that we have a vector of residuals, $\widehat{\mathbf{u}}$, then the inner product of the residuals is:

$$\widehat{\mathbf{u}}'\widehat{\mathbf{u}} = \begin{bmatrix} \widehat{u}_1 & \widehat{u}_2 & \cdots & \widehat{u}_n \end{bmatrix} \begin{bmatrix} \widehat{u}_1 \\ \widehat{u}_2 \\ \vdots \\ \widehat{u}_n \end{bmatrix}$$

$$\widehat{\mathbf{u}}'\widehat{\mathbf{u}} = \widehat{u}_1\widehat{u}_1 + \widehat{u}_2\widehat{u}_2 + \cdots + \widehat{u}_n\widehat{u}_n = \sum_{i=1}^{n} \widehat{u}_i^2$$

- It's just the sum of the squared residuals!
- We can use the inner product to define matrix multiplication. Let $\mathbf{C} = \mathbf{AB}$, then
$$c_{ij} = \mathbf{a}_i'\mathbf{b}_j = a_{i1}b_{1j} + a_{i2}b_{2j} + \cdots + a_{iK}b_{Kj}$$

Special matrices and jargon

- $\mathbf{1}$ is an $n \times 1$ column vector of ones (a "ones vector"):

$$\mathbf{1}'\mathbf{x} = 1 \times x_1 + 1 \times x_2 + \cdots + 1 \times x_n = \sum_{i=1}^{n} x_i$$

- A **square matrix** is one with equal numbers of rows and columns.
- The **diagonal** of a square matrix are the values in which the row number is equal to the column number: a_{11} or a_{22}, etc.

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

- To get the diagonal of a matrix in R, use the `diag()` function:

```
b <- matrix(1:4, nrow = 2, ncol = 2)
b
```

```
##      [,1] [,2]
## [1,]    1    3
## [2,]    2    4
```

```
diag(b)
```

```
## [1] 1 4
```

- The **identity matrix**, $\mathbf{I}$ is a square matrix, with 1s along the diagonal and 0s everywhere else.

$$\mathbf{I} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- The identity matrix multiplied by any matrix just returns the matrix: $\mathbf{AI} = \mathbf{A}$.
- To create an identity matrix in R, you can also use the `diag()` function, but this time just pass it a number instead of a matrix:

```
diag(3)
```

```
##      [,1] [,2] [,3]
## [1,]    1    0    0
## [2,]    0    1    0
## [3,]    0    0    1
```

REGRESSION IN MATRIX FORM

Multiple linear regression in matrix form

- Let $\widehat{\boldsymbol{\beta}}$ be the matrix of estimated regression coefficients:

$$\widehat{\boldsymbol{\beta}} = \begin{bmatrix} \widehat{\beta}_0 \\ \widehat{\beta}_1 \\ \vdots \\ \widehat{\beta}_k \end{bmatrix}$$

- Now, then our estimated regression fits will be:

$$\widehat{\mathbf{y}} = \mathbf{X}\widehat{\boldsymbol{\beta}}$$

- It might be helpful to see this again more written out:

$$\widehat{\mathbf{y}} = \begin{bmatrix} \widehat{y}_1 \\ \widehat{y}_2 \\ \vdots \\ \widehat{y}_n \end{bmatrix} = \mathbf{X}\widehat{\boldsymbol{\beta}} = \begin{bmatrix} 1\widehat{\beta}_0 + x_{11}\widehat{\beta}_1 + x_{12}\widehat{\beta}_2 + \cdots + x_{1K}\widehat{\beta}_K \\ 1\widehat{\beta}_0 + x_{21}\widehat{\beta}_1 + x_{22}\widehat{\beta}_2 + \cdots + x_{2K}\widehat{\beta}_K \\ \vdots \\ 1\widehat{\beta}_0 + x_{n1}\widehat{\beta}_1 + x_{n2}\widehat{\beta}_2 + \cdots + x_{nK}\widehat{\beta}_K \end{bmatrix}$$

- Just a tad bit more tidy, I'd say!

Residuals

- We can easily write the **residuals** in matrix form:

$$\widehat{\mathbf{u}} = \mathbf{y} - \mathbf{X}\widehat{\boldsymbol{\beta}}$$

- Our goal as usual is to minimize the sum of the squared residuals, which we saw earlier we can write:

$$\widehat{\mathbf{u}}'\widehat{\mathbf{u}} = (\mathbf{y} - \mathbf{X}\widehat{\boldsymbol{\beta}})'(\mathbf{y} - \mathbf{X}\widehat{\boldsymbol{\beta}})$$

OLS estimator in matrix form

- By finding the values of $\widehat{\boldsymbol{\beta}}$ that minimizes the sum of the squared residuals, we arrive at the following formula for the OLS estimator:

$$\mathbf{X}'\mathbf{X}\widehat{\boldsymbol{\beta}} = \mathbf{X}'\mathbf{y}$$

- In order to isolate $\widehat{\boldsymbol{\beta}}$, we need to move the $\mathbf{X}'\mathbf{X}$ term to the other side of the equals sign.
- We've learned about matrix multiplication, but what about matrix "division"?

Scalar inverses

- What is division in its simplest form? $\frac{1}{a}$ is the value such that $a\frac{1}{a} = 1$:
- For some algebraic expression: $au = b$, let's solve for u:

$$\begin{aligned} \frac{1}{a}au &= \frac{1}{a}b \\ u &= \frac{b}{a} \end{aligned}$$

- Need a matrix version of this: $\frac{1}{a}$.

Matrix inverses

- **Definition** If it exists, the **inverse** of square matrix $\mathbf{A}$, denoted $\mathbf{A}^{-1}$, is the matrix such that $\mathbf{A}^{-1}\mathbf{A} = \mathbf{I}$.

- We can use the inverse to solve (systems of) equations:

$$\begin{aligned} \mathbf{A}\mathbf{u} &= \mathbf{b} \\ \mathbf{A}^{-1}\mathbf{A}\mathbf{u} &= \mathbf{A}^{-1}\mathbf{b} \\ \mathbf{I}\mathbf{u} &= \mathbf{A}^{-1}\mathbf{b} \\ \mathbf{u} &= \mathbf{A}^{-1}\mathbf{b} \end{aligned}$$

- If the inverse exists, we say that $\mathbf{A}$ is **invertible** or **nonsingular**.

Back to OLS

- Let's assume, for now, that the inverse of $\mathbf{X}'\mathbf{X}$ exists (we'll come back to this)
- Then we can write the OLS estimator as the following:

$$\widehat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}$$

- Memorize this: "x prime x inverse x prime y" sear it into your soul.

OLS by hand in R

- Let's skip the `lm()` function and compute the coefficients directly:
- First we need to get the design matrix:

```
X <- model.matrix(trust_neighbors ~ exports + age + male + urban_dum + malaria_ecology, data = nunn)
dim(X)
```

```
## [1] 20325     6
```

```
## model.frame always puts the response in the first column
y <- model.frame(trust_neighbors ~ exports + age + male + urban_dum + malaria_ecology, data = nunn)[,1]

## solve() does inverses
## and %*% is matrix multiplication
solve(t(X) %*% X) %*% t(X) %*% y
```

```
##                         [,1]
## (Intercept)      1.503037046
## exports         -0.001020836
```

```
## age              0.005044682
## male             0.027836875
## urban_dum       -0.273871917
## malaria_ecology  0.019410561
```

```
coef(mod)
```

```
##     (Intercept)         exports             age            male
##     1.503037046    -0.001020836     0.005044682     0.027836875
##       urban_dum malaria_ecology
##    -0.273871917     0.019410561
```

Intuition for the OLS in matrix form

- What's the intuition here?
- First, note that the "numerator" $\mathbf{X}'\mathbf{y}$ is roughly composed of the covariances between the columns of X and y
- Next, the "denominator" $\mathbf{X}'\mathbf{X}$ is roughly composed of the sample variances and covariances of variables within $\mathbf{X}$
- Thus, we have something like:

$$\widehat{\boldsymbol{\beta}} \approx (\text{variance of } \mathbf{X})^{-1}(\text{covariance of } \mathbf{X} \ \& \ \mathbf{y})$$

- This is a rough sketch and isn't strictly true, but it can provide intuition.
- We're also sidestepping the issues of what the variance of a matrix is for now.

Most general OLS assumptions

1. Linearity: $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{u}$
2. Random/iid sample: $(y_i, \mathbf{x}_i')$ are a iid sample from the population.
3. No perfect collinearity: $\mathbf{X}$ is an $n \times (K+1)$ matrix with rank $K+1$
4. Zero conditional mean: $\mathbb{E}[\mathbf{u}|\mathbf{X}] = \mathbf{0}$
5. Homoskedasticity: $\text{var}(\mathbf{u}|\mathbf{X}) = \sigma_u^2\mathbf{I}_n$
6. Normality: $\mathbf{u}|\mathbf{X} \sim N(\mathbf{0}, \sigma_u^2\mathbf{I}_n)$

No perfect collinearity

- In matrix form: $\mathbf{X}$ is an $n \times (K+1)$ matrix with rank $K+1$
- **Definition** The **rank** of a matrix is the maximum number of linearly independent columns.

- If $\mathbf{X}$ has rank $K+1$, then all of its columns are linearly independent
- …and none of its columns are linearly dependent $\implies$ no perfect collinearity
- $\mathbf{X}$ has rank $K+1 \implies (\mathbf{X}'\mathbf{X})$ is invertible
- Just like variation in X led us to be able to divide by the variance in simple OLS

Expected values of vectors

- The expected value of the vector is just the expected value of its entries.
- Using the zero mean conditional error assumptions:

$$\mathbb{E}[\mathbf{u}|\mathbf{X}] = \begin{bmatrix} \mathbb{E}[u_1|\mathbf{X}] \\ \mathbb{E}[u_2|\mathbf{X}] \\ \vdots \\ \mathbb{E}[u_n|\mathbf{X}] \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \mathbf{0}$$

OLS is unbiased

- Under matrix assumptions 1-4, OLS is unbiased for $\boldsymbol{\beta}$:

$$\mathbb{E}[\widehat{\boldsymbol{\beta}}] = \boldsymbol{\beta}$$

Variance-covariance matrix of random vectors

- The homoskedasticity assumption is different: $\text{var}(\mathbf{u}|\mathbf{X}) = \sigma_u^2\mathbf{I}_n$
- In order to investigate this, we need to know what the variance of a vector is.
- The variance of a vector is actually a matrix:

$$\text{var}[\mathbf{u}] = \Sigma_u = \begin{bmatrix} \text{var}(u_1) & \text{cov}(u_1, u_2) & \dots & \text{cov}(u_1, u_n) \\ \text{cov}(u_2, u_1) & \text{var}(u_2) & \dots & \text{cov}(u_2, u_n) \\ \vdots & & \ddots & \\ \text{cov}(u_n, u_1) & \text{cov}(u_n, u_2) & \dots & \text{var}(u_n) \end{bmatrix}$$

- This matrix is symmetric since $\text{cov}(u_i, u_j) = \text{cov}(u_i, u_j)$

Matrix version of homoskedasticity

- Once again: $\text{var}(\mathbf{u}|\mathbf{X}) = \sigma_u^2\mathbf{I}_n$
- Visually:

$$\text{var}[\mathbf{u}] = \sigma_u^2 \mathbf{I}_n = \begin{bmatrix} \sigma_u^2 & 0 & 0 & \dots & 0 \\ 0 & \sigma_u^2 & 0 & \dots & 0 \\ & & & \vdots & \\ 0 & 0 & 0 & \dots & \sigma_u^2 \end{bmatrix}$$

- In less matrix notation:
  - $\text{var}(u_i) = \sigma_u^2$ for all i (constant variance)
  - $\text{cov}(u_i, u_j) = 0$ for all $i \neq j$ (implied by iid)

Sampling variance for OLS estimates

- Under assumptions 1-5, the sampling variance of the OLS estimator can be written in matrix form as the following:

$$\text{var}[\widehat{\boldsymbol{\beta}}] = \sigma_u^2 (\mathbf{X}'\mathbf{X})^{-1}$$

- This matrix looks like this:

	$\widehat{\beta}_0$	$\widehat{\beta}_1$	$\widehat{\beta}_2$	$\cdots$	$\widehat{\beta}_K$
$\widehat{\beta}_0$	$\text{var}[\widehat{\beta}_0]$	$\text{cov}[\widehat{\beta}_0, \widehat{\beta}_1]$	$\text{cov}[\widehat{\beta}_0, \widehat{\beta}_2]$	$\cdots$	$\text{cov}[\widehat{\beta}_0, \widehat{\beta}_K]$
$\widehat{\beta}_1$	$\text{cov}[\widehat{\beta}_0, \widehat{\beta}_1]$	$\text{var}[\widehat{\beta}_1]$	$\text{cov}[\widehat{\beta}_1, \widehat{\beta}_2]$	$\cdots$	$\text{cov}[\widehat{\beta}_1, \widehat{\beta}_K]$
$\widehat{\beta}_2$	$\text{cov}[\widehat{\beta}_0, \widehat{\beta}_2]$	$\text{cov}[\widehat{\beta}_1, \widehat{\beta}_2]$	$\text{var}[\widehat{\beta}_2]$	$\cdots$	$\text{cov}[\widehat{\beta}_2, \widehat{\beta}_K]$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
$\widehat{\beta}_K$	$\text{cov}[\widehat{\beta}_0, \widehat{\beta}_K]$	$\text{cov}[\widehat{\beta}_K, \widehat{\beta}_1]$	$\text{cov}[\widehat{\beta}_K, \widehat{\beta}_2]$	$\cdots$	$\text{var}[\widehat{\beta}_K]$

Inference in the general setting

- Under assumption 1-5 in large samples:

$$\frac{\widehat{\beta}_k - \beta_k}{\widehat{SE}[\widehat{\beta}_k]} \sim N(0,1)$$

- In small samples, under assumptions 1-6,

$$\frac{\widehat{\beta}_k - \beta_k}{\widehat{SE}[\widehat{\beta}_k]} \sim t_{n-(K+1)}$$

- Thus, under the null of $H_0 : \beta_k = 0$, we know that

$$\frac{\widehat{\beta}_k}{\widehat{SE}[\widehat{\beta}_k]} \sim t_{n-(K+1)}$$

- Here, the estimated SEs come from:

$$\widehat{\text{var}}[\widehat{\boldsymbol{\beta}}] = \widehat{\sigma}^2_u(\mathbf{X}'\mathbf{X})^{-1}$$
$$\widehat{\sigma}^2_u = \frac{\widehat{\mathbf{u}}'\widehat{\mathbf{u}}}{n-(k+1)}$$

- We can access this estimated covariance matrix in R:

```
vcov(mod)
```

```
##                   (Intercept)       exports           age          male
## (Intercept)      4.766593e-04  1.163698e-07 -7.956151e-06 -6.675717e-05
## exports          1.163698e-07  1.676040e-09 -3.658689e-10  7.282947e-09
## age             -7.956151e-06 -3.658689e-10  2.231299e-07 -7.764680e-07
## male            -6.675717e-05  7.282947e-09 -7.764680e-07  1.908894e-04
## urban_dum       -9.658428e-05 -4.861159e-08  7.107867e-07 -1.711373e-06
## malaria_ecology -6.909410e-06 -2.124140e-08  2.324132e-10 -1.017404e-07
##                     urban_dum malaria_ecology
## (Intercept)     -9.658428e-05   -6.909410e-06
## exports         -4.861159e-08   -2.124140e-08
## age              7.107867e-07    2.324132e-10
## male            -1.711373e-06   -1.017404e-07
## urban_dum        2.060633e-04    2.723938e-09
## malaria_ecology  2.723938e-09    7.590439e-07
```

- Note that the diagonal are the variances. So the square root of the diagonal is are the standard errors:

```
sqrt(diag(vcov(mod)))
```

```
##     (Intercept)         exports             age            male
##    2.183253e-02    4.093947e-05    4.723663e-04    1.381627e-02
##       urban_dum malaria_ecology
##    1.435491e-02    8.712313e-04
```

```r
coef(summary(mod))[, "Std. Error"]
```

```
##     (Intercept)         exports             age            male 
##    2.183253e-02    4.093947e-05    4.723663e-04    1.381627e-02 
##       urban_dum malaria_ecology 
##    1.435491e-02    8.712313e-04
```

APPENDIX

Covariance/variance interpretation of matrix OLS

$$\mathbf{X}'\mathbf{y} = \sum_{i=1}^{n} \begin{bmatrix} y_i \\ y_i x_{i1} \\ y_i x_{i2} \\ \vdots \\ y_i x_{iK} \end{bmatrix} \approx \begin{bmatrix} n\overline{y} \\ \widehat{\text{cov}}(y_i, x_{i1}) \\ \widehat{\text{cov}}(y_i, x_{i2}) \\ \vdots \\ \widehat{\text{cov}}(y_i, x_{iK}) \end{bmatrix}$$

$$\mathbf{X}'\mathbf{X} = \sum_{i=1}^{n} \begin{bmatrix} 1 & x_{i1} & x_{i2} & \cdots & x_{iK} \\ x_{i1} & x_{i1}^2 & x_{i2}x_{i1} & \cdots & x_{i1}x_{iK} \\ x_{i2} & x_{i1}x_{i2} & x_{i2}^2 & \cdots & x_{i2}x_{iK} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_{iK} & x_{i1}x_{iK} & x_{i2}x_{iK} & \cdots & x_{iK}x_{iK} \end{bmatrix} \approx \begin{bmatrix} n & n\overline{x}_1 & n\overline{x}_2 & \cdots & n\overline{x}_K \\ n\overline{x}_1 & \widehat{\text{var}}(x_{i1}) & \widehat{\text{cov}}(x_{i1}, x_{i2}) & \cdots & \widehat{\text{cov}}(x_{i1}, x_{iK}) \\ n\overline{x}_2 & \widehat{\text{cov}}(x_{i2}, x_{i1}) & \widehat{\text{var}}(x_{i2}) & \cdots & \widehat{\text{cov}}(x_{i2}, x_{iK}) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ n\overline{x}_K & \widehat{\text{cov}}(x_{iK}, x_{i1}) & \widehat{\text{cov}}(x_{iK}, x_{i2}) & \cdots & \widehat{\text{var}}(x_{iK}) \end{bmatrix}$$