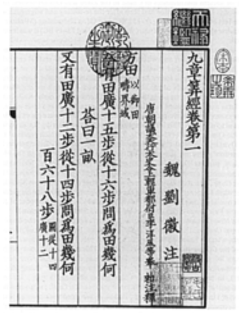
Mathematics for Economists

Chapters 4-5

Linear Models and Matrix Algebra

Johann Carl Friedrich Gauss (1777–1855)



九章筭經卷第一

魏 劉徽 注

方田 以御田疇界域

田廣十五步從十六步問爲田幾何

荅曰一畝

又有田廣十二步從十四步問爲田幾何

百六十八步 圖從十四廣十二

The Nine Chapters on the Mathematical Art (1000-200 BC)

Objectives of Math for Economists

- To study economic problems with the formal tools of math.
- To understand mathematical economics problems by stating the unknown, the data and the restrictions/conditions.
- To plan solutions to these problems by finding a connection between the data and the unknown
- To carry out your plans for solving mathematical economics problems
- To examine the solutions to mathematical economics problems for general insights into current and future problems.

- Remember: Math econ is like love – a simple idea but it can get complicated.

2

4. Linear Algebra

- Some early history:
- The beginnings of matrices and determinants goes back to the second century BC although traces can be seen back to the fourth century BC. But, the ideas did not make it to mainstream math until the late 16^{th} century
- The Babylonians around 300 BC studied problems which lead to simultaneous linear equations.
- The Chinese, between 200 BC and 100 BC, came much closer to matrices than the Babylonians. Indeed, the text *Nine Chapters on the Mathematical Art* written during the Han Dynasty gives the first known example of matrix methods.
- In Europe, 2x2 determinants were considered by *Cardano* at the end of the 16^{th} century and larger ones by *Leibniz* and, in Japan, by *Seki* about 100 years later.

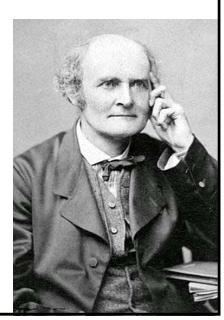
4. What is a Matrix?

- A matrix is a set of elements, organized into rows and columns

$$\begin{array}{cc} & \text{rows} \\ \text{columns} & \begin{bmatrix} a & b \\ c & d \end{bmatrix} \end{array}$$

- *a* and *d* are the diagonal elements.
- *b* and *c* are the off-diagonal elements.

- Matrices are like plain numbers in many ways: they can be added, subtracted, and, in some cases, multiplied and inverted (divided).



Arthur Cayley (1821 – 1895, England)

4. Matrix: Details

- Examples:

$$A = \begin{bmatrix} a_{11} & a_{21} \\ a_{12} & a_{22} \end{bmatrix}; \qquad b = \begin{bmatrix} b_1 & b_2 & b_3 \end{bmatrix}$$

- Dimensions of a matrix: numbers of rows by numbers of columns. The Matrix **A** is a 2x2 matrix, **b** is a 1x3 matrix.

- A matrix with only 1 column or only 1 row is called a *vector*.

- If a matrix has an equal numbers of rows and columns, it is called a *square* matrix. Matrix **A**, above, is a square matrix.

- Usual Notation: Upper case letters $\Rightarrow$ matrices
  Lower case $\Rightarrow$ vectors

5

4. Matrix: Details

- In econometrics, we have data, say *T* (or *N*) observations, on a dependent variable, **Y**, and on *k* explanatory variables, **X**.

- Under the usual notation, vectors will be column vectors: **y** and $\mathbf{x}_k$ are *T*x1 vectors:

$$\mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_T \end{bmatrix} \qquad \& \qquad \boldsymbol{x}_j = \begin{bmatrix} x_{j1} \\ \vdots \\ x_{jT} \end{bmatrix} \qquad j = 1, \ldots, k$$

X is a *T*x*k* matrix: $$\boldsymbol{X} = \begin{bmatrix} x_{11} & \cdots & x_{k1} \\ \vdots & \ddots & \vdots \\ x_{T1} & \cdots & x_{k1} \end{bmatrix}$$

Its columns are the *k* *T*x1 vectors $\mathbf{x}_j$. It is common to treat $\mathbf{x}_1$ as vector of ones, $\boldsymbol{\imath}$.

6

4.1 Special Matrices: Identity and Null

- *Identity Matrix:* A square matrix with 1's along the diagonal and 0's everywhere else. Similar to scalar "1."

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- *Null matrix:* A matrix in which all elements are 0's. Similar to scalar "0."

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

- Both are *diagonal* matrices $\Rightarrow$ off-diagonal elements are zero.
- Both are examples of *symmetric* and *idempotent* matrices. As we will see later:
  - Symmetric: $A = A^T$
  - Idempotent: $A = A^2 = A^3 = \dots$

7

4.1 Matrix: Elementary Row Operations

- Elementary row operations:
  - Switching: Swap the positions of two rows
  - Multiplication: Multiply a row by a non-zero scalar
  - Addition: Add to one row a scalar multiple of another.
- An *elementary matrix* is a matrix which differs from the identity matrix by one single elementary row operation.
- If the matrix subject to elementary row operations is associated to a system of linear equations, then these operations do not change the solution set. Row operations can make the problem easier.
- Elementary row operations are used in Gaussian elimination to reduce a matrix to row echelon form.

8

4.1 Matrix multiplication: Details

- Multiplication of matrices requires a *conformability condition*
- The conformability condition for multiplication is that the column dimensions of the lead matrix **A** must be equal to the row dimension of the lag matrix **B**.
- If **A** is an (*m*x*n*) and **B** an (*n*x*p*) matrix (**A** has the same number of columns as **B** has rows), then we define the product of **AB**. **AB** is (*m*x*p*) matrix with its ij-th element is $\sum_{j=1}^{n} a_{ij} b_{jk}$
- What are the dimensions of the vector, matrix, and result?

$$aB = [a_{11} a_{12}] \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & {}_{22} & b_{23} \end{bmatrix} = c = [c_{11} \quad c_{12} \quad c_{13}]$$
$$= [a_{11}b_{11} + a_{12}b_{21} \quad a_{11}b_{12} + a_{12}b_{22} \quad a_{11}b_{13} + a_{12}b_{23}]$$

- Dimensions: *a*(1x2), B(2x3) $\Rightarrow$ c(1x3)

9

4.1 Transpose Matrix

- The transpose of a matrix **A** is another matrix $\mathbf{A}^{\mathrm{T}}$ (also written **A'**) created by any one of the following equivalent actions:

  - write the rows (columns) of **A** as the columns (rows) of $\mathbf{A}^{\mathrm{T}}$

  - reflect **A** by its main diagonal to obtain $\mathbf{A}^{\mathrm{T}}$
- Formally, the (*i,j*) element of $\mathbf{A}^{\mathrm{T}}$ is the (*j,i*) element of **A**:

  $[\boldsymbol{A}^{\mathrm{T}}]_{ij} = [\boldsymbol{A}]_{ji}$
- If **A** is a $m \times n$ matrix $\Rightarrow$ $\mathbf{A}^{\mathrm{T}}$ is a $n \times m$ matrix.
- $(\mathbf{A}')' = \mathbf{A}$
- Conformability changes unless the matrix is square.

$$\text{Example:} \quad A = \begin{bmatrix} 3 & 8 & -9 \\ 1 & 0 & 4 \end{bmatrix} \quad \Rightarrow A' = \begin{bmatrix} 3 & 1 \\ 8 & 0 \\ -9 & 4 \end{bmatrix}$$

10

4.1 Transpose Matrix: Example – X'

- In econometrics, an important matrix is **X'X**. Recall **X**:

$$\boldsymbol{X} = \begin{bmatrix} x_{11} & \cdots & x_{k1} \\ \vdots & \ddots & \vdots \\ x_{T1} & \cdots & x_{k1} \end{bmatrix}$$ a (Txk) matrix

Then,

$$\boldsymbol{X'} = \begin{bmatrix} x_{11} & \cdots & x_{T1} \\ \vdots & \ddots & \vdots \\ x_{T1} & \cdots & x_{k1} \end{bmatrix}$$ a (kxT) matrix

11

4.1 Basic Operations

- Addition, Subtraction, Multiplication

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} + \begin{bmatrix} e & f \\ g & h \end{bmatrix} = \begin{bmatrix} a+e & b+f \\ c+g & d+h \end{bmatrix}$$ Just add elements

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} - \begin{bmatrix} e & f \\ g & h \end{bmatrix} = \begin{bmatrix} a-e & b-f \\ c-g & d-h \end{bmatrix}$$ Just subtract elements

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} e & f \\ g & h \end{bmatrix} = \begin{bmatrix} ae+bg & af+bh \\ ce+dg & cf+dh \end{bmatrix}$$ Multiply each row by each column and add

$$k\begin{bmatrix} a & b \\ c & d \end{bmatrix} = \begin{bmatrix} ka & kb \\ kc & kd \end{bmatrix}$$ Multiply each element by the scalar

12

4.1 Basic Matrix Operations: Examples

- Matrix addition

$$\begin{bmatrix} 2 & 1 \\ 7 & 9 \end{bmatrix} + \begin{bmatrix} 3 & 1 \\ 0 & 2 \end{bmatrix} = \begin{bmatrix} 5 & 2 \\ 7 & 11 \end{bmatrix}$$

$$A_{2x2} + B_{2x2} = C_{2x2}$$

- Matrix subtraction

$$\begin{bmatrix} 2 & 1 \\ 7 & 9 \end{bmatrix} - \begin{bmatrix} 1 & 0 \\ 2 & 3 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 5 & 6 \end{bmatrix}$$

- Matrix multiplication

$$\begin{bmatrix} 2 & 1 \\ 7 & 9 \end{bmatrix} \text{x} \begin{bmatrix} 1 & 0 \\ 2 & 3 \end{bmatrix} = \begin{bmatrix} 4 & 3 \\ 26 & 27 \end{bmatrix}$$

$$A_{2x2} \text{x} B_{2x2} = C_{2x2}$$

- Scalar multiplication

$$\frac{1}{8}\begin{bmatrix} 2 & 4 \\ 6 & 1 \end{bmatrix} = \begin{bmatrix} 1/4 & 1/2 \\ 3/4 & 1/8 \end{bmatrix}$$

13

4.1 Basic Matrix Operations: X'X

- A special matrix in econometrics, **X'X** (a *k*x*k* matrix):
- Recall **X** (*T*x*k*): $\boldsymbol{X} = \begin{bmatrix} \boldsymbol{x_{11}} & \cdots & \boldsymbol{x_{k1}} \\ \vdots & \ddots & \vdots \\ \boldsymbol{x_{1T}} & \cdots & \boldsymbol{x_{kT}} \end{bmatrix}$ & $\boldsymbol{X'} = \begin{bmatrix} \boldsymbol{x_{11}} & \cdots & \boldsymbol{x_{1T}} \\ \vdots & \ddots & \vdots \\ \boldsymbol{x_{k1}} & \cdots & \boldsymbol{x_{kT}} \end{bmatrix}$

$$\mathbf{X'X} = \begin{bmatrix} \Sigma_{i=1}^n x_{i1}^2 & \Sigma_{i=1}^n x_{i1} x_{i2} & \dots & \Sigma_{i=1}^n x_{i1} x_{iK} \\ \Sigma_{i=1}^n x_{i2} x_{i1} & \Sigma_{i=1}^n x_{i2}^2 & \dots & \Sigma_{i=1}^n x_{i2} x_{iK} \\ \dots & \dots & \dots & \dots \\ \Sigma_{i=1}^n x_{iK} x_{i1} & \Sigma_{i=1}^n x_{iK} x_{i2} & \dots & \Sigma_{i=1}^n x_{iK}^2 \end{bmatrix} = \Sigma_{i=1}^n \begin{bmatrix} x_{i1}^2 & x_{i1} x_{i2} & \dots & x_{i1} x_{iK} \\ x_{i2} x_{i1} & x_{i2}^2 & \dots & x_{i2} x_{iK} \\ \dots & \dots & \dots & \dots \\ x_{iK} x_{i1} & x_{iK} x_{i2} & \dots & x_{iK}^2 \end{bmatrix}$$

$$= \Sigma_{i=1}^n \begin{bmatrix} x_{i1} \\ x_{i2} \\ \dots \\ x_{ik} \end{bmatrix} \begin{bmatrix} x_{i1} & x_{i2} & \dots & x_{iK} \end{bmatrix}$$

$$= \Sigma_{i=1}^n \mathbf{x}_i \mathbf{x}_i'$$

14

4.1 Basic Matrix Operations: $\imath'$X

- Recall $\imath$ is a column vector of ones (in this case, a Tx1 vector):

$$\imath = \begin{bmatrix} 1 \\ 1 \\ \dots \\ 1 \end{bmatrix}$$

- Given **X** (Txk), then $\imath'$ **X** is a 1xk vector:

$$\imath' \boldsymbol{X} = [1 \quad \dots \quad 1] \begin{bmatrix} x_{11} & \cdots & x_{k1} \\ \vdots & \ddots & \vdots \\ x_{1T} & \cdots & x_{kT} \end{bmatrix} = [\sum_{t=1}^{T} x_{1t} \quad \dots \quad \sum_{t=1}^{T} x_{kt}]$$

Note: If $\mathbf{x}_1$ is a vector of ones (representing a constant in the linear classical model), then:

$$\imath' \mathbf{x}_1 = \sum_{t=1}^{T} x_{1t} = \sum_{t=1}^{T} 1 = T \qquad \text{("dot product")}$$

15

4.1 Basic Matrix Operations: R

- Many ways to create a vector (c, 2:7, seq, rep, etc) or a matrix (c, cbind, rbind). Usually, matrices will be data –i.e., read as inpu:

```
> v <- c(1, 3, 5)
> v
[1] 1 3 5
> A <- matrix(c(1, 2, 3, 7, 8, 9), ncol = 3)
> A
    [,1] [,2] [,3]
[1,]    1    3    8
[2,]    2    7    9
> B <- matrix(c(1, 3, 1, 1, 2, 0), nrow = 3)
> B
    [,1] [,2]
[1,]    1    1
[2,]    3    2
[3,]    1    0
```

16

16

4.1 Basic Matrix Operations: R

- Matrix addition/substraction: +/- --element by element

- Matrix multiplication: %*%

```
> C <- A%*%B                 #A is 2x3; B is 3x2
> C
     [,1] [,2]
[1,]   18    7
[2,]   32   16
```

- Scalar multiplication: * --elementwise multiplication of two matrices/vectors

```
> 2*C
     [,1] [,2]
[1,]   36   14
[2,]   64   32
```

17

17

4.1 Basic Matrix Operations: R

- Matrix transpose: t

```
> t(B)              #B is 3x2; t(B) is 2x3
     [,1] [,2] [,3]
[1,]    1    3    1
[2,]    1    2    0
```

- X'X

```
> t(B)%*%B          # command crossprod(B) is more efficient
     [,1] [,2]
[1,]   11    7
[2,]    7    5
```

- dot product

```
> i <- c(1,1,1); t(i)%*%v       # v <- c(1, 3, 5)
     [,1]
[1,]    9
```

18

18

4.1 Laws of Matrix Addition & Multiplication

- Commutative law of Matrix Addition: **A** + **B** = **B** + **A**

$$A+B=\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}+\begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}=\begin{bmatrix} a_{11}+b_{11} & b_{12}+a_{12} \\ a_{21}+a_{21} & b_{22}+a_{22} \end{bmatrix}$$

$$B+A=\begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}+\begin{bmatrix} a_{11} & a_{12} \\ b_{21} & b_{22} \end{bmatrix}=\begin{bmatrix} b_{11}+a_{11} & b_{12}+a_{12} \\ b_{21}+a_{21} & b_{22}+a_{22} \end{bmatrix}$$

- Matrix Multiplication is distributive across Additions:

A (**B**+ **C**) = **AB** + **AC** (assuming comformability applies).

19

4.1 Matrix Multiplication

- Matrix multiplication is generally not commutative. That is,

AB ≠ **BA** even if **BA** is conformable

(because different dot product of rows or col. of **A**&**B**)

$$A=\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, B=\begin{bmatrix} 0 & -1 \\ 6 & 7 \end{bmatrix}$$

$$AB=\begin{bmatrix} 1(0)+2(6) & 1(-1)+2(7) \\ 3(0)+4(6) & 3(-1)+4(7) \end{bmatrix}=\begin{bmatrix} 12 & 13 \\ 24 & 25 \end{bmatrix}$$

$$BA=\begin{bmatrix} 0(1)+(-1)(3) & 0(2)+(-1)4 \\ 6(1)+7(3) & 6(2)+(7)4 \end{bmatrix}=\begin{bmatrix} -3 & -4 \\ 27 & 40 \end{bmatrix}$$

20

4.1 Matrix multiplication

- Exceptions to non-commutative law:

  AB=**BA** iff

  B = a scalar,

  B = identity matrix **I**, or

  B = the inverse of **A** -i.e., $\mathbf{A}^{-1}$

- **Theorem**: It is not true that **AB** = **AC** => **B**=**C**

  Proof:

$$A = \begin{bmatrix} 1 & 2 & 1 \\ 1 & 0 & -1 \\ 1 & -2 & -3 \end{bmatrix};\ B = \begin{bmatrix} 1 & -1 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \end{bmatrix};\ C = \begin{bmatrix} 2 & 1 & 2 \\ -1 & -1 & -1 \\ 2 & 3 & 1 \end{bmatrix}$$

Note: If **AB** = **AC** for all matrices **A**, then **B** = **C**.

21

4.1 Inverse of a Matrix

- Identity matrix: **AI = A**

$$I_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Notation: $\mathbf{I}_j$ is a *j*x*j* identity matrix.

- Given **A** (*m*x*n*), the matrix **B** (*n*x*m*) is a *right-inverse* for **A** iff

  $\mathbf{AB} = \mathbf{I}_m$

- Given **A** (*m*x*n*), the matrix **C** (*m*x*n*) is a *left-inverse* for **A** iff

  $\mathbf{CA} = \mathbf{I}_n$

22

4.1 Inverse of a Matrix

- **Theorem**: If **A** (mxn), has both a *right-inverse* **B** and a *left-inverse* **C**, then **C** = **B**.

  Proof:

  We have $\mathbf{AB}=\mathbf{I_m}$ and $\mathbf{CA}=\mathbf{I_n}$.

  Thus,

  $\mathbf{C}(\mathbf{AB}) = \mathbf{C\,I_m} = \mathbf{C}$ and $\mathbf{C}(\mathbf{AB})=(\mathbf{CA})\mathbf{B} = \mathbf{I_n B} = \mathbf{B}$

  $\Rightarrow \mathbf{C}(n\text{x}m)=\mathbf{B}(m\text{x}n)$

Note:

- This matrix is unique. (Suppose there is another left-inverse **D**, then **D**=**B** by the theorem, so **D**=**C**.).

- If **A** has both a right and a left inverse, it is a square matrix. It is usually called *invertible*. We say "the matrix **A** is *non-singular*."

4.1 Inverse of a Matrix

- Inversion is tricky:
  $(\mathbf{ABC})^{-1} = \mathbf{C}^{-1}\mathbf{B}^{-1}\mathbf{A}^{-1}$

- **Theorem**: If **A** (mxn) and **B** (nxp) have inverses, then **AB** is invertible and $(\mathbf{AB})^{-1} = \mathbf{B}^{-1}\mathbf{A}^{-1}$

  Proof:

  We have $\mathbf{AA}^{-1}=\mathbf{I_m}$ and $\mathbf{A}^{-1}\mathbf{A}=\mathbf{I_n}$

  $\mathbf{BB}^{-1}=\mathbf{I_n}$ and $\mathbf{B}^{-1}\mathbf{B}=\mathbf{I_p}$

  Thus,

  $\mathbf{B}^{-1}\mathbf{A}^{-1}(\mathbf{AB}) = \mathbf{B}^{-1}(\mathbf{A}^{-1}\mathbf{A})\,\mathbf{B} = \mathbf{B}^{-1}\,\mathbf{I_n B} = \mathbf{B}^{-1}\,\mathbf{B} = \mathbf{I_p}$

  $(\mathbf{AB})\,\mathbf{B}^{-1}\mathbf{A}^{-1} = \mathbf{A}\,(\mathbf{BB}^{-1})\,\mathbf{A}^{-1} = \mathbf{A\,I_n}\,\mathbf{A}^{-1} = \mathbf{A}\,\mathbf{A}^{-1} = \mathbf{I_m}$

  $\Rightarrow$ **AB** is invertible and $(\mathbf{AB})^{-1} = \mathbf{B}^{-1}\mathbf{A}^{-1}$

- More on this topic later.

24

4.1 Transpose and Inverse Matrix

- $(\mathbf{A} + \mathbf{B})' = \mathbf{A}' + \mathbf{B}'$
- If $\mathbf{A}' = \mathbf{A}$, then $\mathbf{A}$ is called a *symmetric* matrix.

- **Theorems**:
  - Given two comformable matrices **A** and **B**, then $(\mathbf{AB})' = \mathbf{B}'\mathbf{A}'$
  - If **A** is invertible, then $(\mathbf{A}^{-1})' = (\mathbf{A}')^{-1}$ (and $\mathbf{A}'$ is also invertible).

25

4.1 Partitioned Matrix

- A *partitioned matrix* is a matrix which has been broken into sections called *blocks* or *submatrices* by horizontal and/or vertical lines extending along entire rows or columns. For example, the 3x*m* matrix can be partitioned as:

$$\begin{bmatrix} a_{11} & a_{12} & | & \Lambda & a_{1m} \\ a_{21} & a_{22} & | & \Lambda & a_{2m} \\ - & - & | & - & - \\ a_{31} & a_{32} & | & \Lambda & a_{3m} \end{bmatrix} = \begin{bmatrix} A_{11}(2\text{x}2) & A_{12}(2\text{x}(m-2)) \\ A_{21}(1\text{x}2) & A_{22}(1\text{x}(m-2)) \end{bmatrix}$$

- Augmented matrices are also partitioned matrices. They have been partitioned vertically into two blocks.

- Partitioned matrices are used to simplify the computation of inverses.

26

4.1 Partitioned Matrix

- If two matrices, **A** and **B**, are partitioned the same way, addition can be done by blocks. Similarly, if both matrices are *comformable partitioned*, then multiplication can be done by blocks.

- A *block diagonal matrix* is a partitioned square matrix, with main diagonal blocks square matrices and the off-diagonal blocks are null matrices.

<u>Nice Property</u>: The inverse of a block diagonal matrix is just the inverse of each block.

$$\begin{bmatrix} A_1 & 0 & \Lambda & 0 \\ 0 & A_2 & \Lambda & 0 \\ \Lambda & \Lambda & \Lambda & \Lambda \\ 0 & 0 & \Lambda & A_n \end{bmatrix} \Rightarrow \begin{bmatrix} A_1^{-1} & 0 & \Lambda & 0 \\ 0 & A_2^{-1} & \Lambda & 0 \\ \Lambda & \Lambda & \Lambda & \Lambda \\ 0 & 0 & \Lambda & A_n^{-1} \end{bmatrix}$$

27

4.1 Partitioned Matrix: Partitioned OLS Solution

- In the Classical Linear Model, we have the OLS solution:

$$b = (X'X)^{-1}X'y \qquad \Rightarrow \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} X_1'X_1 & X_1'X_2 \\ X_2'X_1 & X_2'X_2 \end{bmatrix}^{-1} \begin{bmatrix} X_1'y \\ X_2'y \end{bmatrix}$$

- Use of the partitioned inverse result produces a fundamental result, the Frisch-Waugh (1933) Theorem: To calculate $\mathbf{b}_2$ (or $\mathbf{b}_1$) we do not need to invert the whole matrix. For this result, we need the southeast element in the inverse of $(\mathbf{X'X})^{-1}$:

$[\,]^{-1}_{(2,1)}$ $\qquad$ $[\,]^{-1}_{(2,2)}$

$$\begin{bmatrix} \mathbf{X_1'X_1} & \mathbf{X_1'X_2} \\ \mathbf{X_2'X_1} & \mathbf{X_2'X_2} \end{bmatrix}^{\mathbf{-1}}$$

- With the partitioned inverse, we get:

$$\mathbf{b}_2 = [\,]^{-1}_{(2,1)}\ \mathbf{X}_1'\mathbf{y} + [\,]^{-1}_{(2,2)}\ \mathbf{X}_2'\mathbf{y}$$

28

4.1 Partitioned Matrix: Partitioned OLS Solution

- From partitioned inverse: $\mathbf{b}_2 = [\,]^{-1}_{(2,1)}\ \mathbf{X}_1'\mathbf{y} + [\,]^{-1}_{(2,2)}\ \mathbf{X}_2'\mathbf{y}$

- As we will derive later:

$$1.\ \text{Matrix } X'X = \begin{bmatrix} X_1'X_1 & X_1'X_2 \\ X_2'X_1 & X_2'X_2 \end{bmatrix}$$

$$2.\ \text{Inverse} = \begin{bmatrix} (X_1'X_1)^{-1} + (X_1'X_1)^{-1}X_1'X_2DX_2'X_1(X_1'X_1)^{-1} & (X_1'X_1)^{-1}X_1'X_2D \\ -DX_2'X_1(X_1'X_1)^{-1} & D \end{bmatrix}$$

$$\text{where } D = [X_2'X_2 - X_2'X_1(X_1'X_1)^{-1}X_1'X_2]^{-1} = [X_2'(I - X_1(X_1'X_1)^{-1}X_1)'X_2]^{-1}$$

$$\Rightarrow D = [X_2'M_1X_2]^{-1}$$

- The algebraic result is: $[\,]^{-1}_{(2,1)} = \text{-D}\ \mathbf{X}_2'\mathbf{X}_1(\mathbf{X}_1'\mathbf{X}_1)^{-1}$
  $[\,]^{-1}_{(2,2)} = \text{D} = [\mathbf{X}_2'\mathbf{M}_1\mathbf{X}_2]^{-1}$

$\Rightarrow \quad \mathbf{b}_2 = [\,]^{-1}_{(2,1)}\ \mathbf{X}_1'\mathbf{y} + [\,]^{-1}_{(2,2)}\ \mathbf{X}_2'\mathbf{y} = [\mathbf{X}_2'\mathbf{M}_1\mathbf{X}_2]^{-1}\mathbf{X}_2'\mathbf{M}_1\mathbf{y}$

4.1 Properties of Symmetric Matrices

- Definition:
  If $\mathbf{A}' = \mathbf{A}$, then $\mathbf{A}$ is called a *symmetric* matrix.

- **Theorems**:
  - If $\mathbf{A}$ and $\mathbf{B}$ are nxn symmetric matrices, then $(\mathbf{AB})' = \mathbf{BA}$
  - If $\mathbf{A}$ and $\mathbf{B}$ are nxn symmetric matrices, then $(\mathbf{A}+\mathbf{B})' = \mathbf{B}+\mathbf{A}$
  - If $\mathbf{C}$ is any nxn matrix, then $\mathbf{B} = \mathbf{C}'\mathbf{C}$ is symmetric.

- Useful symmetric matrices:
  $\mathbf{V} = \mathbf{X}'\mathbf{X}$
  $\mathbf{P} = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'$ — **P**: Projection matrix
  $\mathbf{M} = \mathbf{I} - \mathbf{P} = \mathbf{I} - \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'$ — **M**: Residual maker

30

4.1 Application 1: Linear System

- There is a functional form relating a dependent variable, y, and k explanatory variables, **X**. The functional form is linear, but it depends on k unknown parameters, $\boldsymbol{\beta}$. The relation between y and **X** is not exact. There is an error, $\boldsymbol{\varepsilon}$. We have T observations of y and **X**.

- Then, the data is generated according to:

$$y_i = \Sigma_{j=1,..k}\, x_{k,i}\, \beta_k + \varepsilon_i \qquad\qquad i=1, 2, ..., T.$$

Or using matrix notation:

$$\mathbf{y} = \mathbf{X}\, \boldsymbol{\beta} + \boldsymbol{\varepsilon}$$

where $\mathbf{y}$ & $\boldsymbol{\varepsilon}$ are (Tx1); $\mathbf{X}$ is (Txk); and $\boldsymbol{\beta}$ is (kx1).

- We will call this relation *data generating process* (DGP).

- The goal of econometrics is to estimate the unknown vector $\boldsymbol{\beta}$.

31

4.1 Application 2: System of Equations

- Assume an economic model as system of linear equations with:
  a_{ij} parameters, where i = 1,.., m rows, j = 1,.., n columns
  x_i endogenous variables (n),
  d_i exogenous variables and constants (m).

$$\begin{cases} a_{11}\, x_1 + a_{12}\, x_2 + \dots + a_{1n}\, x_n = d_1 \\ a_{21}\, x_1 + a_{22}\, x_2 + \dots + a_{2n}\, x_n = d_2 \\ \dots\dots \qquad \dots\dots \qquad \dots\dots \qquad \dots \\ a_{m1}\, x_1 + a_{m2}\, x_2 + \dots + a_{mn}\, x_n = d_m \end{cases}$$

- We can write this system using linear algebra notation: $\mathbf{A}\,\mathbf{x} = \boldsymbol{d}$

$$\begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix} \begin{bmatrix} x_1 \\ \dots \\ x_n \end{bmatrix} = \begin{bmatrix} d_1 \\ \dots \\ d_m \end{bmatrix}$$

$\boldsymbol{d}$ = column vector

$\mathbf{A}$ = (mxn) matrix

$\mathbf{x}$ = column vector

- Q: What is the nature of the set of solutions to this system?

32

4.1 Application 2: System of Equations

- System of linear equations: **Ax** = ***d***

where
A = (mxn) matrix of parameters
x = column vector of endogenous variables (nx1)
d = column vector of exogenous variables and constants (mx1)

- Solve for x*

- Questions:

- For what combinations of **A** and ***d*** there will zero, one, many or an infinite number of solutions?
- How do we compute (characterize) those sets of solutions?

33

4.1 Solution of a General Equation System

- **Theorem**: Given **A** (mxn). If **A** has a right-inverse, then the equation **Ax** = ***d*** has at least one solution for every ***d*** (mx1).

Proof:
Pick an arbitrary ***d***. Let **H** be a right-inverse (so $\mathbf{AH}=\mathbf{I}_m$).
Define **x*=H*d***.
Thus,
$\mathbf{Ax}^* = \mathbf{A\,H}\boldsymbol{d} = \mathbf{I}_m\boldsymbol{d} = \boldsymbol{d}$ => x* is a solution. ■

34

4.1 Solution of a General Equation System

- **Theorem**: Given **A** (mxn). If **A** has a left-inverse, then the equation **A**x=$\boldsymbol{d}$ has at most one solution for every $\boldsymbol{d}$ (mx1). That is, if **A**x=$\boldsymbol{d}$ has a solution x* for a particular $\boldsymbol{d}$, then x* is unique.

Proof:

Suppose x* is a solution and z* is another solution. Thus, **A**x*=$\boldsymbol{d}$ and **A**z*=$\boldsymbol{d}$. Let **G** be a left-inverse for **A** (so $\mathbf{GA}=\mathbf{I}_n$).

$$\mathbf{A}x^*=\boldsymbol{d} \quad \Rightarrow \mathbf{GA}\,x^*=\mathbf{G}\boldsymbol{d}$$
$$\Rightarrow \mathbf{I}_n x^* = x^* = \mathbf{G}\boldsymbol{d}.$$

$$\mathbf{A}z^*= \boldsymbol{d} \quad \Rightarrow \mathbf{GA}\,z^* = \mathbf{G}\boldsymbol{d}$$
$$\Rightarrow \mathbf{I}_n z^* = z^* = \mathbf{G}\boldsymbol{d}.$$

Thus,

$$x^*=z^*=\mathbf{G}\boldsymbol{d}. \ \blacksquare$$

35

4.1 Solution of a General Equation System

- Problem with the previous proof? We're assuming the left-inverse exists (and there's always a solution).

- Assume the 2x2 model

2x + y = 12

4x + 2y = 24

Find x*, y*:

y = 12 – 2x

4x + 2(12 – 2x) = 24

4x +24 – 4x = 24

0 = 0 ? indeterminante!

- Why?

4x + 2y =24

2(2x + y) = 2(12)

- one equation with two unknowns

2x + y = 12

<u>Conclusion</u>: Not all simultaneous equation models have solutions

(not all matrices have inverses).

36

4.1 Solution of a General Equation System

- **Theorem**: Given **A** (mxn) invertible. Then, the equation **Ax** = ***d*** has one and only one solution for every ***d*** (mx1).

Proof:
Trivial from previous two theorems.

- Given an invertible matrix, **A**, use the "solve" command:

```
> A
     [,1] [,2]
[1,]   18    7
[2,]   32   16
> d <- c(2, 1)
> x <- solve(A, d)
> x
 [1]  0.390625 -0.718750
```

37

4.1 Linear dependence and Rank: Example

- A set of vectors is *linearly dependent* if any one of them can be expressed as a linear combination of the remaining vectors; otherwise, it is linearly independent.

- Formal definition: Linear independence (LI)

The set $\{u_1,...,u_k\}$ is called a *linearly independent* set of vectors iff

$$c_1\, u_1 + ... + c_k u_k = \theta \qquad \Rightarrow \; c_1 = c_2 = ... = c_k, = 0.$$

- Notes:

- Dependence prevents solving a system of equations. More unknowns than independent equations.

- The number of linearly independent rows or columns in a matrix is the *rank* of a matrix (rank(**A**)).

38

4.1 Linear dependence and Rank: Example

- Examples:

$$v_1' = [5 \quad 12]$$
$$v_2' = [10 \quad 24]$$
$$A = \begin{bmatrix} 5 & 10 \\ 12 & 24 \end{bmatrix} = \begin{bmatrix} v_1' \\ v_2' \end{bmatrix}$$
$$2v_1' - v_2' = 0' \qquad \Rightarrow rank(A) = 1$$

$$v_1 = \begin{bmatrix} 2 \\ 7 \end{bmatrix}; v_2 = \begin{bmatrix} 1 \\ 8 \end{bmatrix}; v_3 = \begin{bmatrix} 4 \\ 5 \end{bmatrix}; \qquad A = \begin{bmatrix} 2 & 1 & 4 \\ 7 & 8 & 5 \end{bmatrix}$$
$$3v_1 - 2v_2$$
$$= [6 \quad 21] - [2 \quad 16]$$
$$= [4 \quad 5] = v_3$$
$$3v_1 - 2v_2 - v_3 = 0 \quad \Rightarrow rank(A) = 2$$

39

4.2 Application 1: One Commodity Market Model (2x2 matrix)

- Economic Model

1) $Q_d = a - bP$ (a,b >0)
2) $Q_s = -c + dP$ (c,d >0)
3) $Q_d = Q_s$

- Find P* and Q*

$$P^* = \frac{a+c}{b+d}$$
$$Q^* = \frac{ad-bc}{b+d}$$

Scalar Algebra form

(Endogenous Vars :: Constants)

4) $1Q + bP = \quad a$

5) $1Q - dP = \quad -c$

40

4.2 Application 1: One Commodity Market Model (2x2 matrix)

Matrix algebra

$$\begin{bmatrix} 1 & b \\ 1 & -d \end{bmatrix} \begin{bmatrix} Q \\ P \end{bmatrix} = \begin{bmatrix} a \\ -c \end{bmatrix}$$

$$Ax = d$$

$$\begin{bmatrix} Q^* \\ P^* \end{bmatrix} = \begin{bmatrix} 1 & b \\ 1 & -d \end{bmatrix}^{-1} \begin{bmatrix} a \\ -c \end{bmatrix}$$

$$x^* = A^{-1} d$$

41

4.2 Application 2: Finite Markov Chains

- Markov processes are used to measure movements over time.

Employees at time 0 are distributed over two plants A & B

$$x_0' = [A_0 \quad B_0] = [100 \quad 100]$$

The employees stay and move between each plants w/ a known probability

$$M = \begin{bmatrix} P_{AA} & P_{AB} \\ P_{BA} & P_{BB} \end{bmatrix} = \begin{bmatrix} .7 & .3 \\ .4 & .6 \end{bmatrix}$$

At the end of one year, how many employees will be at each plant?

$$[A_1 \quad B_1] = x_0' M = [A_0 \quad B_0] \begin{bmatrix} P_{AA} & P_{AB} \\ P_{BA} & P_{BB} \end{bmatrix} = [A_0 P_{AA} + A_0 P_{BA} \quad B_0 P_{AB} + B_0 P_{BB}]$$

$$= [100 \quad 100] \begin{bmatrix} .7 & .3 \\ .4 & .6 \end{bmatrix} = [.7*100 + .4*100, \qquad .3*100 + .6*100]$$

$$= [110 \quad 90]$$

42

4.2 Application 2: Finite Markov Chains

At the end of two years, how many employees will be at each plant?

$$[A_1 \quad B_1] = x_0' M = [A_0 \quad B_0]\begin{bmatrix} P_{AA} & P_{AB} \\ P_{BA} & P_{BB} \end{bmatrix} = [110 \quad 90]$$

$$[A_2 \quad B_2] = x_0' M^2 = [A_0 \quad B_0]\begin{bmatrix} P_{AA} & P_{AB} \\ P_{BA} & P_{BB} \end{bmatrix}\begin{bmatrix} P_{AA} & P_{AB} \\ P_{BA} & P_{BB} \end{bmatrix}$$

$$= [110 \quad 90]\begin{bmatrix} .7 & .3 \\ .4 & .6 \end{bmatrix} = [.7*110 + .4*90, \quad .3*110 + .6*90]$$

$$= [113 \quad 87]$$

M

After k years : $[A_k \quad B_k] = x_0' M^k$

43

4.3 Definite Matrices - Forms

- A form is a polynomial expression in which each component term has a uniform degree. A quadratic form has a uniform second degree.

Examples:

$9x + 3y + 2z$	-first degree form.
$6x^2 + 2xy + 2y^2$	-second degree (quadratic) form.
$x^2z + 2yz^2 + 2y^3$	-third degree (cubic) form.

- A quadratic form can be written as: **x'A x**, where **A** is a symmetric matrix.

44

4.3 Definite Matrices - Forms

- For one variable, a quadratic form is the familiar: $y = a\,x^2$

If $a>0$, then $a\,x^2$ is always non-negative, and equals 0 only when $x=0$. We call a form like this *positive definite*.

If $a<0$, then $a\,x^2$ is always non-positive, and equals 0 only when $x=0$. We call a form like this *negative definite*.

There are two intermediate cases, where the form can be equal to 0 for some non-zero values of x: *negative/positive semidefinite*.

- For a general quadratic form, $y = \mathbf{x'A\,x}$, we say the form is
  - *Positive definite* if y is invariably positive ($y > 0$)
  - *Positive semi-definite* if y is invariably non-negative ($y \geq 0$)
  - *Negative semi-definite* if y is invariably non-positive ($y \leq 0$)
  - *Negative definite* if y is invariably negative ($y < 0$)
  - *Indefinite* if y changes signs.

45

4.3 Definite Matrices - Definition

- A quadratic form is said to be indefinite if y changes signs.

- A symmetric $(n\times n)$ ***A*** is called *positive definite (pd), positve semidefinite (psd), negative semidefinite (nsd) and negative definite (nd)* according to the corresponding sign of the quadratic form, y.

For example, if $y = \mathbf{x'A\,x}$, is positive, for any non-zero vector $\mathbf{x}$ of n real numbers; we say $\mathbf{A}$ is positive definite.

<u>Example</u>: Let $\mathbf{A} = \mathbf{X'X}$.

Then, $\mathbf{z'A\,z} = \mathbf{z'X'X\,z} = \mathbf{v'v} > 0$. $\Rightarrow$ $\mathbf{X'X}$ is pd

- In general, we use eigenvalues to determine the definiteness of a matrix (and quadratic form).

46

4.4 Upper and Lower Triangular Matrices

- A square (nxn) matrix **C** is:
  -Upper Triangular (UT) iff C_{ij}=0 for i>j
(if the diagonal elements are all equal to 1, we have a upper-unit triangular (UUT) matrix)

$$\begin{bmatrix} 1 & 2 & 5 \\ 0 & 0 & 6 \\ 0 & 0 & 1 \end{bmatrix} \text{UT}$$

  -Lower Triangular (LT) iff C_{ij}=0 for i<j
(if the diagonal elements are all equal to 1, we have a lower-unit triangular (LUT) matrix)

$$\begin{bmatrix} 0 & 0 & 0 \\ 2 & 1 & 0 \\ 1 & 2 & 0 \end{bmatrix} \text{LT}$$

  -Diagonal (D) iff C_{ij}=0 for i≠j

- **Theorems**:
The product of the two UT (UUT) matrices is UT (UUT).
The product of the two LT (LUT) matrices is LT (LUT).
The product of the two D matrices is D.

47

4.4 UT & LT Matrices – LU Factorization

- An (nxn) matrix **A** can be factorized, with proper row and/or column permutations, into two factors, an LT matrix ***L*** and an UT matrix ***U***:

$$A = LU = \begin{bmatrix} l_{11} & 0 & 0 \\ l_{21} & l_{22} & 0 \\ l_{31} & l_{32} & l_{33} \end{bmatrix} \text{x} \begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix}$$

- Without permutations in **A**, the factorization may fail. We have an n^2 by n^2 system. For example, given $a_{11} = l_{11}\, u_{11}$, if a_{11}=0, then at least one of l_{11} & u_{11} has to be 0, which implies either ***L*** or ***U*** is singular (impossible if ***A*** is non-singular).

- A proper permutation matrix, **P**, is enough for LU factorization. It is called *LU factorization with Partial Pivoting* (or PA = LU).

4.4 UT & LT Matrices – Forward Substitution

• The LU decomposition requires $2n^3/3$ (plus lower order terms) operations or "flops" –i.e., floating point operations (+,-,x,/). When n is large, n^3 dominates, we describe this situation with "order n^3"or $O(n^3)$.

• Q: Why are we interested in these matrices?
Suppose $\mathbf{A}\mathbf{x}=\boldsymbol{d}$, where $\mathbf{A}$ is LT (with non-zero diagonal terms). Then, the solutions are recursive (*forward substitution*).

Example:
$x_1 = d_1/a_{11}$
$a_{21}\, x_1 + a_{22}\, x_2 = d_2$
$a_{31}\, x_1 + a_{32}\, x_2 + a_{33}\, x_3 = d_3$

Note: For an nxn matrix $\mathbf{A}$, this process involves n^2 flops.

49

4.4 UT & LT Matrices – Back Substitution

• Similarly, suppose $\mathbf{A}x=d$, where $\mathbf{A}$ is UT (with non-zero diagonal terms). Then, the solutions are recursive (*backward substitution*).

Example:
$a_{11}\, x_1 + a_{12}\, x_2 + a_{13}\, x_3 = d_1$
$a_{22}\, x_2 + a_{23}\, x_3 = d_2$
$x_3 = d_3/a_{31}$

Note: Again, for $\mathbf{A}(n$x$n)$, this process involves n^2 flops.

50

4.4 UT & LT Matrices – Linear Systems

- Finding a solution to **Ax**=***d***

Given **A** (nxn). Suppose we can decompose **A** into **A**=**LU**, where **L** is LUT and **U** is UUT (with non-zero diagonal).

Then $\quad \mathbf{Ax}=\boldsymbol{d} \Rightarrow \mathbf{LUx} = \boldsymbol{d}$.

Suppose **L** is invertible $\Rightarrow \mathbf{Ux} = \mathbf{L}^{-1}\boldsymbol{d} = \boldsymbol{c}$ $\quad$ (or $\boldsymbol{d} = \mathbf{L}\boldsymbol{c}$)
$\Rightarrow$ solve by forward substitution for ***c***.

Then, $\mathbf{Ux} = \boldsymbol{c}$ *(Gaussian elimination)* $\Rightarrow$ solve by backward substitution for **x**.

- **Theorem**:

If **A** (nxn) can be decomposed **A**=**LU**, where **L** is LUT and **U** is UUT (with non-zero diagonal), then **Ax**=***d*** has a unique solution for every ***d***.

51

4.4 UT & LT Matrices – LDU Decomposition

- We can write a "symmetric" decomposition. Since **U** has non-zero diagonal terms, we can write **U**=**DU***, where **U*** is UUT.

Example:

$$U = \begin{bmatrix} 2 & 4 & 8 \\ 0 & 3 & 6 \\ 0 & 0 & 5 \end{bmatrix}; \quad D = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 5 \end{bmatrix}; \quad => U^* = \begin{bmatrix} 1 & 2 & 4 \\ 0 & 1 & 2 \\ 0 & 0 & 1 \end{bmatrix}$$

- **Theorems**:

- If we can write **A** (nxn) as **A**=**LDU**, where **L** is LUT, **D** is diagonal with non zero diagonal elements, and **U** is UUT, then **L**, **D**, and **U** are unique.

- If we can write **A** (nxn) as **A**=**LDU**, and **A** is symmetric, then we can write **A**=**LDL'**.

52

4.4 Cholesky Decomposition

- **Theorem**: Cholesky decomposition

A is a symmetric positive definite matrix (**A** symmetric, **A=LDL',** and all diagonal elements of D are positive), then **A** = **HH'.**

Proof:

Since **A** is symmetric, then **A=LDL'.**

The product of a LUT matrix and a D matrix is a LUT matrix.

Let **D***=**D**$^{1/2}$ and **L** be a LT matrix.

Then **H**=**LD*** is matrix is LT $\Rightarrow$ **A=HH'.** ■

- **H** is called the *Cholesky factor* of **A** ('square root' of a pd matrix.)

- The Cholesky decomposition is unique. It is used in the numerical solution of systems of equations, non-linear optimization, Kalman filter algorithms, IRF of VARs, etc.

53

4.4 Cholesky decomposition: Algorithm

- Let's partition matrices **A=HH'** as:

$$\begin{bmatrix} a_{11} & A_{21}^T \\ A_{21} & A_{22} \end{bmatrix} = \begin{bmatrix} l_{11} & 0 \\ L_{21} & L_{22} \end{bmatrix} \begin{bmatrix} l_{11} & L_{21}^T \\ 0 & L_{22}^T \end{bmatrix} = \begin{bmatrix} l_{11}^2 & l_{11}L_{21}^T \\ l_{11}L_{21} & L_{21}L_{21}^T + L_{22}L_{22}^T \end{bmatrix}$$

- Algorithm

1. Determine l_{11} and L_{21}: $\quad l_{11} = \sqrt{a_{11}}$ & $L_{21} = (1/l_{11})\ A_{21}$

(if **A** is pd $\Rightarrow a_{11}>0$)

2. Compute L_{22} from $A_{22} - L_{21}\,L_{21}^T = L_{22}\,L_{22}^T$

(if **A** is pd $\Rightarrow A_{22} - L_{21}\,L_{21}^T = A_{22} - A_{21}A_{21}^T/a_{11}$ is pd)

André-Louis Cholesky (1875–1918, France)

4.4 Cholesky decomposition: Algorithm

- Example:

$$\begin{bmatrix} 25 & 15 & -5 \\ 15 & 18 & 0 \\ -5 & 0 & 11 \end{bmatrix} = \begin{bmatrix} l_{11} & 0 & 0 \\ l_{21} & l_{22} & 0 \\ l_{31} & l_{32} & l_{33} \end{bmatrix} \begin{bmatrix} l_{11} & l_{21} & l_{31} \\ 0 & l_{22} & l_{32} \\ 0 & 0 & l_{33} \end{bmatrix}$$

- first column of L

$$\begin{bmatrix} 25 & 15 & -5 \\ 15 & 18 & 0 \\ -5 & 0 & 11 \end{bmatrix} = \begin{bmatrix} 5 & 0 & 0 \\ 3 & l_{22} & 0 \\ -1 & l_{32} & l_{33} \end{bmatrix} \begin{bmatrix} 5 & 3 & -1 \\ 0 & l_{22} & l_{32} \\ 0 & 0 & l_{33} \end{bmatrix}$$

- second column of L

$$\begin{bmatrix} 18 & 0 \\ 0 & 11 \end{bmatrix} - \begin{bmatrix} 3 \\ -1 \end{bmatrix} \begin{bmatrix} 3 & -1 \end{bmatrix} = \begin{bmatrix} l_{22} & 0 \\ l_{32} & l_{33} \end{bmatrix} \begin{bmatrix} l_{22} & l_{32} \\ 0 & l_{33} \end{bmatrix}$$

$$\begin{bmatrix} 9 & 3 \\ 3 & 10 \end{bmatrix} = \begin{bmatrix} 3 & 0 \\ 1 & l_{33} \end{bmatrix} \begin{bmatrix} 3 & 1 \\ 0 & l_{33} \end{bmatrix}$$

55

4.4 Cholesky decomposition: Algorithm

- Example:
  - third column of L: $10 - 1 = l_{33}^2$, $i.e.$, $l_{33} = 3$

conclusion:

$$\begin{bmatrix} 25 & 15 & -5 \\ 15 & 18 & 0 \\ -5 & 0 & 11 \end{bmatrix} = \begin{bmatrix} 5 & 0 & 0 \\ 3 & 3 & 0 \\ -1 & 1 & 3 \end{bmatrix} \begin{bmatrix} 5 & 3 & -1 \\ 0 & 3 & 1 \\ 0 & 0 & 3 \end{bmatrix}$$

Note: Again, for $\mathbf{A}$(nxn), the Cholesky decomposition involves $n^3/3$ flops.

56

4.4 Cholesky decomposition: Application

- System of Equations

If $\mathbf{A}$ is a positive definite matrix, then we can solve $\mathbf{Ax} = \boldsymbol{d}$ by
(1) Compute the Cholesky decomposition $\mathbf{A}=\mathbf{HH}'$.
(2) Solve $\mathbf{Hy} = \boldsymbol{d}$ for $\mathbf{y}$, (*forward solution*)
(3) With $\mathbf{y}$ known, solve $\mathbf{H'x} = \mathbf{y}$ for $\mathbf{x}$. (*backward solution*)
Q: How many flops? Step (1): $n^3/3$ flops, Steps (2)+(3): $2n^2$ flops.

Note: $\mathbf{A}^{-1}$ is not computed (Gauss-Jordan methods needs $4n^3$ flops)

- Ordinary Least Squares (OLS)

Systems of the form $\mathbf{Ax} = \boldsymbol{d}$ with $\mathbf{A}$ symmetric and pd are common in economics. For example, the *normal equations* in OLS problems are of this form (the unknown is $\mathbf{b}$):

$$(\mathbf{y} - \mathbf{Xb})'\,\mathbf{X} = 0 \qquad \Rightarrow \mathbf{X'X}\,\mathbf{b} = \mathbf{X'}\,\mathbf{y}$$

No need to compute $(\mathbf{X'X})^{-1}$ $(=\mathbf{A}^{-1})$ to solve for $\mathbf{b}$.

57

4.5 Inverse matrix (Again)

- Review
  - $\mathbf{AA}^{-1} = \mathbf{I}$
  - $\mathbf{A}^{-1}\mathbf{A}=\mathbf{I}$
  - Necessary for matrix to be square to have *unique* inverse.
  - If an inverse exists for a square matrix, it is unique
  - $(\mathbf{A}')^{-1}=(\mathbf{A}^{-1})'$
  - If $\mathbf{A}$ is pd, then $\mathbf{A}^{-1} = \mathbf{H}'^{-1}\mathbf{H}^{-1}$
  - Solution to $\mathbf{A}\,\mathbf{x} = \boldsymbol{d}$

$$\mathbf{A}^{-1}\mathbf{A}\,\mathbf{x}^* = \mathbf{A}^{-1}\,\boldsymbol{d}$$

$$\mathbf{I}\,\mathbf{x}^* = \mathbf{A}^{-1}\,\boldsymbol{d} \qquad \Rightarrow \mathbf{x}^* = \mathbf{A}^{-1}\,\boldsymbol{d} \text{ (solution depends on } A^{-1})$$

  - Linear independence a problem to get $\mathbf{x}^*$
  - Determinant test! (coming soon)

58

4.5 Inverse of a Matrix: Calculation

- **Theorem**: Let **A** be an invertible (*n*x*n*) matrix. Suppose that a sequence of elementary row-operations reduces **A** to the identity matrix. Then, the same sequence of elementary row-operations when applied to the identity matrix yields $\mathbf{A}^{-1}$.

$$\begin{bmatrix} a & b & c & 1 & 0 & 0 \\ d & e & f\,| & 0 & 1 & 0 \\ g & h & i & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 & r & s & t \\ 0 & 1 & 0\,| & u & v & w \\ 0 & 0 & 1 & x & y & z \end{bmatrix}$$

Process:

- Append the identity matrix to **A**.
- Subtract multiples of the other rows from the first row to reduce the diagonal element to 1.
- Transform **I** as you go.
- When the original **A** matrix becomes **I**, the original identity has become $\mathbf{A}^{-1}$.

4.5 Determination of the Inverse (Gauss-Jordan Elimination)

$\mathbf{AX} = \mathbf{I}$ — all **A**, **X** and **I** are (*n*x*n*) square matrices — $\mathbf{X} = \mathbf{A}^{-1}$

1) Augmented matrix

2) Transform (using elementary row operations) augmented matrix

$$[\mathbf{A} \,\vdots\, \mathbf{I}\,] \xrightarrow{} [\,\mathbf{UT} \,\vdots\, \mathbf{H}] \xrightarrow{} [\,\mathbf{I} \,\vdots\, \mathbf{K}]$$

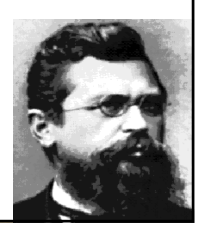
further row operations

Gauss elimination

Gauss-Jordan elimination

$$\mathbf{I\,X} = \mathbf{K}$$

$$\mathbf{I\,X} = \mathbf{X} = \mathbf{A}^{-1} \Rightarrow \mathbf{K} = \mathbf{A}^{-1}$$



Wilhelm Jordan (1842– 1899, Germany)

4.5 Gauss-Jordan Elimination: Example 1

Find $\mathbf{A}^{-1}$ using the Gauss-Jordan method.

$$A = \begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{bmatrix}$$

Process: Expand $\mathbf{A}|\mathbf{I}$. Start scaling and adding rows to get $\mathbf{I}|\mathbf{A}^{-1}$.

1. $$A|I = \begin{bmatrix} 2 & 1 & 1 & 1 & 0 & 0 \\ 1 & 2 & 1 & 0 & 1 & 0 \\ 1 & 1 & 2 & 0 & 0 & 1 \end{bmatrix} \xrightarrow{R_1(1/2)} \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ 1 & 2 & 1 & 0 & 1 & 0 \\ 1 & 1 & 2 & 0 & 0 & 1 \end{bmatrix}$$

2. $$\begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ 1 & 2 & 1 & 0 & 1 & 0 \\ 1 & 1 & 2 & 0 & 0 & 1 \end{bmatrix} \xrightarrow{R_{21}(-1)\,\&\,R_{31}(-1)} \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ 0 & \frac{3}{2} & \frac{1}{2} & -\frac{1}{2} & 1 & 0 \\ 0 & \frac{1}{2} & \frac{3}{2} & -\frac{1}{2} & 0 & 1 \end{bmatrix}$$

4.5 Gauss-Jordan Elimination: Example 1

3. $$\begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ 0 & \frac{3}{2} & \frac{1}{2} & -\frac{1}{2} & 1 & 0 \\ 0 & \frac{1}{2} & \frac{3}{2} & -\frac{1}{2} & 0 & 1 \end{bmatrix} \xrightarrow{R_2(2/3)} A = \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ 0 & 1 & \frac{1}{3} & -\frac{1}{3} & \frac{2}{3} & 0 \\ 0 & \frac{1}{2} & \frac{3}{2} & -\frac{1}{2} & 0 & 1 \end{bmatrix}$$

4. $$\begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ 0 & 1 & \frac{1}{3} & -\frac{1}{3} & \frac{2}{3} & 0 \\ 0 & \frac{1}{2} & \frac{3}{2} & -\frac{1}{2} & 0 & 1 \end{bmatrix} \xrightarrow{R_{32}(-1/2)} \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ 0 & 1 & \frac{1}{3} & -\frac{1}{3} & \frac{2}{3} & 0 \\ 0 & 0 & \frac{4}{3} & -\frac{1}{3} & -\frac{1}{3} & 1 \end{bmatrix}$$

5. $$\begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ 0 & 1 & \frac{1}{3} & -\frac{1}{3} & \frac{2}{3} & 0 \\ 0 & 0 & \frac{4}{3} & -\frac{1}{3} & -\frac{1}{3} & 1 \end{bmatrix} \xrightarrow{R_3(3/4)} \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ 0 & 1 & \frac{1}{3} & -\frac{1}{3} & \frac{2}{3} & 0 \\ 0 & 0 & 1 & -\frac{1}{4} & -\frac{1}{4} & \frac{3}{4} \end{bmatrix}$$

Gauss elimination

4.5 Gauss-Jordan Elimination: Example 1

6. $$\begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ 0 & 1 & \frac{1}{3} & -\frac{1}{3} & \frac{2}{3} & 0 \\ 0 & 0 & 1 & -\frac{1}{4} & -\frac{1}{4} & \frac{3}{4} \end{bmatrix} \xrightarrow{R_{23}(-1/3)\,\&\,R_{13}(-1/2)} \begin{bmatrix} 1 & \frac{1}{2} & 0 & \frac{5}{8} & \frac{1}{8} & \frac{-3}{8} \\ 0 & 1 & 0 & \frac{-1}{4} & \frac{3}{4} & \frac{-1}{4} \\ 0 & 0 & 1 & \frac{-1}{4} & \frac{-1}{4} & \frac{3}{4} \end{bmatrix}$$

7. $$\begin{bmatrix} 1 & \frac{1}{2} & 0 & \frac{5}{8} & \frac{1}{8} & \frac{-3}{8} \\ 0 & 1 & 0 & \frac{-1}{4} & \frac{3}{4} & \frac{-1}{4} \\ 0 & 0 & 1 & \frac{-1}{4} & \frac{-1}{4} & \frac{3}{4} \end{bmatrix} \xrightarrow{R_{12}(-1/2)} \begin{bmatrix} 1 & 0 & 0 & \frac{6}{8} & \frac{2}{8} & \frac{-2}{8} \\ 0 & 1 & 0 & \frac{-1}{4} & \frac{3}{4} & \frac{-1}{4} \\ 0 & 0 & 1 & \frac{-1}{4} & \frac{-1}{4} & \frac{3}{4} \end{bmatrix}$$

Gauss-Jordan elimination

8. $$I \mid A^{-1} = \begin{bmatrix} 1 & 0 & 0 & \frac{6}{8} & \frac{2}{8} & \frac{-2}{8} \\ 0 & 1 & 0 & \frac{-1}{4} & \frac{3}{4} & \frac{-1}{4} \\ 0 & 0 & 1 & \frac{-1}{4} & \frac{-1}{4} & \frac{3}{4} \end{bmatrix} \longrightarrow A^{-1} = \begin{bmatrix} \frac{3}{4} & \frac{-1}{4} & \frac{-1}{4} \\ \frac{-1}{4} & \frac{3}{4} & \frac{-1}{4} \\ \frac{-1}{4} & \frac{-1}{4} & \frac{3}{4} \end{bmatrix}$$

4.5 Gauss-Jordan Elimination: Example 2

Partitioned inverse (using the Gauss-Jordan method).

1. $$\begin{bmatrix} \Sigma_{XX} & \Sigma_{XY} & I & 0 \\ \Sigma_{YX} & \Sigma_{YY} & 0 & I \end{bmatrix} \xrightarrow{\Sigma_{XX}^{-1}R_1} \begin{bmatrix} I & \Sigma_{XX}^{-1}\Sigma_{XY} & \Sigma_{XX}^{-1} & 0 \\ \Sigma_{YX} & \Sigma_{YY} & 0 & I \end{bmatrix}$$

2. $$\xrightarrow{R_2 - \Sigma_{YX}R_1} \begin{bmatrix} I & \Sigma_{XX}^{-1}\Sigma_{XY} & \Sigma_{XX}^{-1} & 0 \\ 0 & \Sigma_{YY} - \Sigma_{YX}\Sigma_{XX}^{-1}\Sigma_{XY} & -\Sigma_{YX}\Sigma_{XX}^{-1} & I \end{bmatrix}$$

3. $$\xrightarrow{[\Sigma_{YY} - \Sigma_{YX}\Sigma_{XX}^{-1}\Sigma_{XY}]^{-1}R_2} \begin{bmatrix} I & \Sigma_{XX}^{-1}\Sigma_{XY} & \Sigma_{XX}^{-1} & 0 \\ 0 & I & D(-\Sigma_{YX}\Sigma_{XX}^{-1}) & D \end{bmatrix}$$

where $D = [\Sigma_{YY} - \Sigma_{YX}\Sigma_{XX}^{-1}\Sigma_{XY}]^{-1}$

4. $$\xrightarrow{R_1 - \Sigma_{XX}^{-1}\Sigma_{XY}R_2} \begin{bmatrix} I & 0 & \Sigma_{XX}^{-1} + \Sigma_{XX}^{-1}\Sigma_{XY}D\Sigma_{YX}\Sigma_{XX}^{-1} & \Sigma_{XX}^{-1}\Sigma_{XY}D \\ 0 & I & -D(\Sigma_{YX}\Sigma_{XX}^{-1}) & D \end{bmatrix}$$

4.5 Gauss-Jordan Elimination: Computations

• Q: How many flops to invert a matrix with the G-J method?
A: Avoid inverses! But, if you must... The process of zeroing out one element of the left-hand matrix requires multiplying the line to be subtracted by a constant ($2n$ flops), and subtracting it ($2n$ flops). This must be done for (approximately) n^2 matrix elements. Thus, the number of flops is about equal to $4n^3$ by the G-J method.

• Using a standard PC (100 Gigaflops, 10^9, per second), for a 30x30 matrix, the time required is less than a millisecond, comparing favorably with 10^{21}+ years for the method of cofactors.

• More sophisticated (optimal) algorithms, taking advantage of zeros –i.e., the sparseness of the matrix-, can improve to n^3 flops.

4.5 Matrix inversion: Note

- It is not possible to divide one matrix by another. That is, we can not write A/B. For two matrices **A** and **B**, the quotient can be written as $\mathbf{AB}^{-1}$ or $\mathbf{B}^{-1}\mathbf{A}$.

- In general, in matrix algebra $\mathbf{AB}^{-1} \neq \mathbf{B}^{-1}\mathbf{A}$.

  Thus, writing A/B does not clearly identify whether it represents $\mathbf{AB}^{-1}$ or $\mathbf{B}^{-1}\mathbf{A}$.

  We'll say $\mathbf{B}^{-1}$ post-multiplies **A** (for $\mathbf{AB}^{-1}$) and

  $\mathbf{B}^{-1}$ pre-multiplies **A** (for $\mathbf{B}^{-1}\mathbf{A}$)

- Matrix division is matrix inversion.

66

4.5 Matrix inversion: R

- To find the inverse of a matrix or solve a system of equations, use "solve"

```
> A
    [,1] [,2]
[1,]   18    7
[2,]   32   16
> solve(A)
      [,1]      [,2]
[1,]  0.25 -0.109375
[2,] -0.50  0.281250
```

- Solve system $\mathbf{Ax} = \mathbf{d}$

```
> d <- c(2, 1)
> x <- solve(A, d); x
    [1]  0.390625 -0.718750
```


4.6 Trace of a Matrix

- The *trace* of an nxn matrix $\boldsymbol{A}$ is defined to be the sum of the elements on the main diagonal of $\boldsymbol{A}$:

  $\text{trace}(\mathbf{A}) = tr(\mathbf{A}) = \Sigma_i \, a_{ii}$.

  where a_{ii} is the entry on the ith row and ith column of A.
- Properties:
  - $tr(\boldsymbol{A} + \boldsymbol{B}) = tr(\boldsymbol{A}) + tr(\boldsymbol{B})$
  - $tr(c\boldsymbol{A}) = c\; tr(\boldsymbol{A})$
  - $tr(\boldsymbol{AB}) = tr(\boldsymbol{BA})$
  - $tr(\boldsymbol{ABC}) = tr(\boldsymbol{CAB})$ (*invariant* under *cyclic* permutations.)
  - $tr(\boldsymbol{A}) = tr(\boldsymbol{A}^T)$
  - $d\; tr(\boldsymbol{A}) = tr(d\boldsymbol{A})$ (differential of trace)
  - $tr(\boldsymbol{A}) = rank(\boldsymbol{A})$ when $\boldsymbol{A}$ is idempotent –i.e., $\boldsymbol{A} = \boldsymbol{A}^2$.

4.6 Application: Rank of the Residual Maker

- We define **M**, the residual maker, as:

$$\mathbf{M} = \mathbf{I}_n - \boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}' = \mathbf{I}_n - \mathbf{P}$$

where $\boldsymbol{X}$ is an nxk matrix, with $rank(\mathbf{X})=k$

- Let's calculate the trace of **M**:

$tr(\mathbf{M}) = tr(\mathbf{I}_n) - tr(\mathbf{P}) = n - k$

- $tr(\mathbf{I}_T) = n$
- $tr(\boldsymbol{P}) = k$

Recall $tr(\boldsymbol{ABC}) = tr(\boldsymbol{CAB})$

$\Rightarrow tr(\boldsymbol{P}) = tr(\boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}') = tr(\boldsymbol{X}'\boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X})^{-1}) = tr(\boldsymbol{I}_k) = k$

- Since **M** is an idempotent matrix –i.e., $\mathbf{M}=\mathbf{M}^2$-, then

$rank(\mathbf{M}) = tr(\mathbf{M}) = n - k$

69

4.7 Determinant of a Matrix

- The determinant is a number associated with any squared matrix.
- If **A** is an nxn matrix, the determinant is $|\mathbf{A}|$ or det(**A**).
- Since the early days, a determinant was used to "determine" if a system of linear equations has a unique solution.
- Cramer (1750) expanded the concept to sets of equations, but a bit later, they were recognized as independent functions, Vandermole (1772).
- Determinants are used to characterize invertible matrices. A matrix is invertible (non-singular) if and only if $|\mathbf{A}|\neq 0$.
- That is, if $|\mathbf{A}|\neq 0 \rightarrow \mathbf{A}$ is invertible or non-singular.
- Can be found using factorials, pivots, and cofactors!
- Lots of interpretations.

70

4.7 Determinant of a Matrix

- When n is small, determinants are used for inversion and to solve systems of equations.

Example: Inverse of a 2x2 matrix:

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \qquad |A| = \det(A) = ad - bc$$

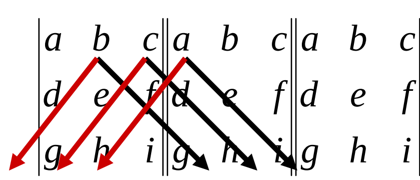
$$A^{-1} = \frac{1}{ad - bc}\begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

This matrix is called the adjugate of A (or adj(A)).

$A^{-1} = \text{adj}(A) / |A|$

4.7 Determinant of a Matrix (3x3)

$$\begin{vmatrix} a & b & c \\ d & e & f \\ g & h & i \end{vmatrix} = aei + bfg + cdh - afh - bdi - ceg$$



Sarrus' Rule: Sum from left to right. Then, subtract from right to left

Note: N! terms

- Q: How many flops? For **A** (3x3), we count 17 operations.

4.7 Determinants: Laplace formula

- The determinant of a matrix of arbitrary size can be defined by the *Leibniz formula* or the *Laplace formula*.
- The *Laplace formula* (or *expansion*) expresses the determinant $|\boldsymbol{A}|$ as a sum of n determinants of $(n\text{-}1) \times (n\text{-}1)$ sub-matrices of **A**. There are n^2 such expressions, one for each row and column of **A**
- Define the *i,j minor* M_{ij} (usually written as $|M_{ij}|$) of **A** as the determinant of the $(n\text{-}1) \times (n\text{-}1)$ matrix that results from deleting the *i*-th row and the *j*-th column of **A**.

Pierre-Simon Laplace (1749–1827, France).

4.7 Determinants: Laplace formula

- Define the $C_{i,j}$ the *cofactor* of **A** as:

$$C_{i,j} = (-1)^{i+j} \, | M_{i,j} |$$

- The cofactor matrix of **A** -denoted by **C**-, is defined as the *n*x*n* matrix whose (*i,j*) entry is the (*i,j*) cofactor of **A**. The transpose of **C** is called the adjugate or adjoint of **A** -adj(**A**).

- **Theorem** (Determinant as a Laplace expansion)
  Suppose **A** = $[a_{ij}]$ is an *nxn* matrix and *i,j*= {1, 2, ...,*n*}. Then the determinant

$$\begin{aligned} |A| &= a_{i1}C_{i1} + a_{i2}C_{i2} + \ldots + a_{in}C_{in} \\ &= a_{ij}C_{ij} + a_{2j}C_{2j} + \ldots + a_{nj}C_{nj} \end{aligned}$$

74

4.7 Determinants: Laplace formula

- Example:

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 0 & -1 & 0 \\ 2 & 4 & 6 \end{bmatrix}$$

$$|A| = 1 \text{ x } C_{11} + 2 \text{ x } C_{12} + 3 \text{ x } C_{13} =$$
$$= 1 \text{ x } (-1 \text{ x } 6) + 2 \text{ x } (-1) \text{ x } (0) + 3 \text{ x } (-(-1) \text{ x } 2)) = 0$$
$$= -2 \text{ x } (0) + (-1)(1\text{x}6 - 3\text{x}2) + -4 \text{ x } (0) = 0$$

- |**A**|=0 ⇒ The matrix is singular. (Check!)

- How many flops? For a **A** (3x3), we count 14 operations (better!). For **A** (nxn), we calculate n subdeterminants, each of which requires (n-1) subdeterminants, etc. Then, computations of order n! (plus some n terms), or O(n!).

75

4.7 Determinants: Properties

- Interchange of rows and columns does not affect |**A**|. (Corollary, |**A**| = |**A'**|.)
- To any row (column) of ***A*** we can add any multiple of any other row (column) without changing |**A**|.

  (Corollary, if we transform **A** into **U** or **L**, |**A**|=|**U**| = |**L**|, which is equal to the product of the diagonal element of **U** or **L**.)
- |**I**| = 1, where **I** is the identity matrix.
- |k**A**| = k^n |**A**|, where k is a scalar.
- |**A**| = |**A'**|.
- |**AB**| = |**A**| |**B**|.
- |$\mathbf{A}^{-1}$|=1/|**A**|.
- Recursive flops formula: $\text{flops}_n = n * (\text{flops}_{n-1} + 2) - 1$

76

4.7 Determinants: R

- Simple command, det(A)
- > M = cbind(rbind(1,2), rbind(6,5))

```
     [,1] [,2]
[1,]    1    6
[2,]    2    5
>det(M)
[1] -7
> det(M*2)
[1] -28
> Minv <-solve(M); M);Minv
           [,1]       [,2]
[1,] -0.7142857  0.8571429
[2,]  0.2857143 -0.1428571
> det(Minv)
[1] -0.1428571
```

77

4.7 Determinants: Computations

- By today's standards, a 30×30 matrix is small. Yet it would be impossible to calculate a 30×30 determinant by Laplace formula. It would require over *n*! ($30! \approx 2.65 \times 10^{32}$) multiplications.

- If a computer performs one quatrillion (1.0×10^{15}) multiplications per second (a Petaflops, the 2008 record), it would have to run for over 8.4 billion years to compute a 30×30 determinant by Laplace's method.

- Using today's fastest computer (2013 China Tianhe-2, 33 petaflops), it would take 254 million years.

- Not a very useful, computationally speaking, method. Avoid factorials!

78

4.7 Determinants: Computations

- Faster way of evaluating the determinant: Bring the matrix to UT (or LT) form by linear transformations. Then, the determinant is equal to the product of the diagonal elements.

- For **A** (nxn), each linear transformation involves adding a multiple of one row to another row, that is, n or fewer additions and n or fewer multiplications. Since there are n rows, this is a procedure of order n^3 -or O(n^3).

<u>Example</u>: For n = 30, we go from 30! = 2.65*10^{32} flops to 30^3 = 27,000 flops.

79

4.7 Determinants: Cramer's Rule - Derivation

- Recall the solution to **Ax**=***d***, where **A** is an nxn matrix:

$$\mathbf{x}^* = \mathbf{A}^{-1}\boldsymbol{d}$$

Using the cofactor method to get the inverse we get:

$$x^* = \frac{1}{|A|}\left(\text{adjoint } A\right) \qquad (d)$$

$$\underset{nx1}{\begin{bmatrix} x_1^* \\ x_2^* \\ \text{M} \\ x_n^* \end{bmatrix}} = \underset{1x1}{\frac{1}{|A|}} \underset{nxn}{\begin{bmatrix} |C_{11}| & |C_{21}|\Lambda & |C_{n1}| \\ |C_{12}| & |C_{22}|\Lambda & |C_{n2}| \\ \text{M} & \text{M} & \text{M} \\ |C_{1n}| & |C_{2n}|\Lambda & |C_{nn}| \end{bmatrix}} \underset{nx1}{\begin{bmatrix} d_1 \\ d_2 \\ \text{M} \\ d_n \end{bmatrix}} = \frac{1}{|A|}\begin{bmatrix} \sum_{i=1}^{n} d_i|C_{i1}| \\ \sum_{i=1}^{n} d_i|C_{i2}| \\ \text{M} \\ \sum_{i=1}^{n} d_i|C_{in}| \end{bmatrix}$$

80

4.7 Determinants: Cramer's Rule - Derivation

- Example: Let **A** be 3x3. Then,

1) $$\begin{bmatrix} x_1^* \\ x_2^* \\ x_3^* \end{bmatrix} = \frac{1}{|A|}\begin{bmatrix} d_1|C_{11}| + d_2|C_{21}| + d_3|C_{31}| \\ d_1|C_{12}| + d_2|C_{22}| + d_3|C_{32}| \\ d_1|C_{13}| + d_2|C_{23}| + d_3|C_{33}| \end{bmatrix} = \frac{1}{|A|}\begin{bmatrix} \sum_{i=1}^{3} d_i|C_{i1}| \\ \sum_{i=1}^{3} d_i|C_{i2}| \\ \sum_{i=1}^{3} d_i|C_{i3}| \end{bmatrix}$$

2) $$\sum_{i=1}^{3} d_i|C_{i1}| = d_1|C_{11}| + d_2|C_{21}| + d_i|C_{31}| \quad \text{where } |C_{ij}| \equiv (-1)^{i+j}|M_{ij}|$$

3) $$\sum_{i=1}^{3} d_i|C_{i1}| = d_1\begin{vmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{vmatrix} + d_2\begin{vmatrix} a_{12} & a_{13} \\ a_{32} & a_{33} \end{vmatrix} + d_3\begin{vmatrix} a_{12} & a_{13} \\ a_{22} & a_{23} \end{vmatrix} = |A_1|$$

4) $$A_1 = \begin{bmatrix} d_1 & a_{12} & a_{13} \\ d_2 & a_{22} & a_{23} \\ d_3 & a_{32} & a_{33} \end{bmatrix}. \quad \text{Find } |A_1| \text{ such that } x_1^* = |A_1|/|A|$$

81

4.7 Determinants: Cramer's Rule - Derivation

$$x_1^* = \frac{\sum_{i=1}^{3} d_i|C_{i1}|}{\sum_{i=1}^{3} a_{i1}|C_{i1}|} = \frac{\begin{vmatrix} d_1 & a_{12} & a_{13} \\ d_2 & a_{22} & a_{23} \\ d_3 & a_{32} & a_{33} \end{vmatrix}}{\begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix}} = \frac{|A_1|}{|A|}$$

$$x_2^* = \frac{\sum_{i=1}^{3} d_i|C_{i2}|}{\sum_{i=1}^{3} a_{i2}|C_{i2}|} = \frac{\begin{vmatrix} a_{11} & d_1 & a_{13} \\ a_{21} & d_2 & a_{23} \\ a_{31} & d_3 & a_{33} \end{vmatrix}}{\begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix}} = \frac{|A_2|}{|A|}$$

82

4.7 Determinants: Cramer’s Rule - Derivation

$$x_3^* = \frac{\sum_{i=1}^{3} d_i |C_{i3}|}{\sum_{i=1}^{3} a_{i3} |C_{i3}|} = \frac{\begin{vmatrix} a_{11} & a_{12} & d_1 \\ a_{21} & a_{22} & d_2 \\ a_{31} & a_{32} & d_3 \end{vmatrix}}{\begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix}} = \frac{|A_3|}{|A|}$$

Gabriel Cramer (1704-1752, Switzerland).

4.7 Determinants: Cramer’s Rule - Derivation

- Following the pattern, we have the general Cramer’s rule:

$$\begin{bmatrix} x_1^* \\ x_2^* \\ x_3^* \\ \vdots \\ x_n^* \end{bmatrix} = \frac{1}{|A|} \begin{bmatrix} \sum_{i=1}^{n} d_i |C_{i1}| \\ \sum_{i=1}^{n} d_i |C_{i2}| \\ \sum_{i=1}^{n} d_i |C_{i3}| \\ \vdots \\ \sum_{i=1}^{n} d_i |C_{in}| \end{bmatrix} = \begin{bmatrix} |A_1|/|A| \\ |A_2|/|A| \\ |A_3|/|A| \\ \vdots \\ |A_n|/|A| \end{bmatrix}$$

84

4.7 Cramer's Rule Application: Macro Model

Matrix form

$$\begin{bmatrix} 1 & -1 & -1 \\ -b & 1 & 0 \\ -g & 0 & 1 \end{bmatrix}\begin{bmatrix} Y \\ C \\ G \end{bmatrix} = \begin{bmatrix} I_0 \\ a-bT_0 \\ 0 \end{bmatrix}$$

The determinant

$$|A| = \begin{vmatrix} 1 & -1 & -1 \\ -b & 1 & 0 \\ -g & 0 & 1 \end{vmatrix} = 1-(b+g)$$

85

4.7 Cramer's Rule Application: Macro Model

- Applying Cramer's rule for the 3x3 case:

$$|A_Y| = \begin{vmatrix} I_0 & -1 & -1 \\ a-bT_0 & 1 & 0 \\ 0 & 0 & 1 \end{vmatrix} = I_0 + a - bT_0 \qquad Y^* = \frac{|A_Y|}{|A|} = \frac{I_0 + a - bT_0}{1-(b+g)}$$

$$|A_C| = \begin{vmatrix} 1 & I_0 & -1 \\ -b & a-bT_0 & 0 \\ -g & 0 & 1 \end{vmatrix} = bI_0 + (1-g)(a-bT_0) \qquad C^* = \frac{|A_C|}{|A|} = \frac{bI_0 + (1-g)(a-bT_0)}{1-(b+g)}$$

$$|A_G| = \begin{vmatrix} 1 & -1 & I_0 \\ -b & 1 & a-bT_0 \\ -g & 0 & 0 \end{vmatrix} = g(a-bT_0+I_0) \qquad G^* = \frac{|A_G|}{|A|} = \frac{g(a-bT_0+I_0)}{1-(b+g)}$$

86

Ch. 4 - Notation and Definitions: Summary

- **A** (Upper case letters) = matrix
- b (Lower case letters) = vector
- *n*x*m* = *n* rows, *m* columns
- rank(**A**) = number of linearly independent vectors of **A**
- trace(**A**) = *tr*(**A**) = sum of diagonal elements of **A**
- Null matrix = all elements equal to zero.
- Diagonal matrix = all off-diagonal elements are zero.
- **I** = identity matrix (diagonal elements: 1, off-diagonal: 0)
- $|\mathbf{A}|$ = det(**A**) = determinant of **A**
- $\mathbf{A}^{-1}$ = inverse of **A**
- $\mathbf{A}' = \mathbf{A}^T$ = Transpose of **A**
- $|M_{ij}|$ = Minor of **A**
- $\mathbf{A} = \mathbf{A}^T$ => Symmetric matrix
- $\mathbf{A}^T \mathbf{A} = \mathbf{A}\,\mathbf{A}^T$ => Normal matrix
- $\mathbf{A}^T = \mathbf{A}^{-1}$ => Orthogonal matrix
- $\mathbf{A} = \mathbf{A}^2$ => Idempotent matrix

87