Review of Matrix Algebra

Eric Zivot

4/12/2021

Matrices and Vectors

Matrix

$$\underset{(n \times m)}{\mathbf{A}} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{bmatrix}$$

$$n = \text{ \# of rows, } m = \text{ \# of columns}$$

Square matrix : $n = m$

Matrices and Vectors

Vector

$$\underset{(n\times 1)}{\mathbf{x}} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

R and Excel

- R is a matrix oriented programming language
- Excel can handle matrices and vectors in formulas and some functions
- Excel has special functions for working with matrices. They are called *array* functions.
- You evaluate an Excel array function using `<ctrl> - <shift> - <enter>`

R Examples

You create matrices using the `matrix()` constructor function:

```
matA = matrix(data=c(1,2,3,4,5,6), nrow=2, ncol=3)
matA
```

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

```
class(matA)
```

```
## [1] "matrix" "array"
```

Note: The matrix is filled columnwise.

R Examples

You can fill the matrix rowwise using the optional argument byrow=TRUE:

```
matA = matrix(data=c(1,2,3,4,5,6), nrow=2, ncol=3,
              byrow=TRUE)
matA
```

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

Matrices have dimension attrubutes

```
dim(matA)
```

```
## [1] 2 3
```

R Examples

Vectors in R do not have dimension attributes. They appear as row vectors but that is just how they are printed on screen.

```
xvec = c(1,2,3)
xvec
```

```
## [1] 1 2 3
```

```
class(xvec)
```

```
## [1] "numeric"
```

```
dim(xvec)
```

```
## NULL
```

Transpose of a Matrix

Interchange rows and columns of a matrix

$$\underset{(m\times n)}{\mathbf{A}}{}' = \text{transpose of } \underset{(n\times m)}{\mathbf{A}}$$

Example

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix},\ \mathbf{A}' = \begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix}$$

$$\mathbf{x} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix},\ \mathbf{x}' = \begin{bmatrix} 1 & 2 & 3 \end{bmatrix}$$

R and Excel functions

- R function: `t(A)`
- Excel function `TRANSPOSE(MATRIX)`

R Examples

```
matA = matrix(data=c(1,2,3,4,5,6), nrow=2, ncol=3,
              byrow=TRUE)
t(matA)
```

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

```
xvec = c(1,2,3)
t(xvec)
```

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

Symmetric Matrix

A square matrix $\mathbf{A}$ is symmetric if

$$\mathbf{A} = \mathbf{A}'$$

Example

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

Remark: Covariance and correlation matrices are symmetric

Basic Matrix Operations

Addition and Subtraction (element-by-element)

$$\begin{aligned} \begin{bmatrix} 4 & 9 \\ 2 & 1 \end{bmatrix} + \begin{bmatrix} 2 & 0 \\ 0 & 7 \end{bmatrix} &= \begin{bmatrix} 4+2 & 9+0 \\ 2+0 & 1+7 \end{bmatrix} \\ &= \begin{bmatrix} 6 & 9 \\ 2 & 8 \end{bmatrix} \end{aligned}$$

$$\begin{aligned} \begin{bmatrix} 4 & 9 \\ 2 & 1 \end{bmatrix} - \begin{bmatrix} 2 & 0 \\ 0 & 7 \end{bmatrix} &= \begin{bmatrix} 4-2 & 9-0 \\ 2-0 & 1-7 \end{bmatrix} \\ &= \begin{bmatrix} 2 & 9 \\ 2 & -6 \end{bmatrix} \end{aligned}$$

R Examples

```
matA = matrix(c(4,9,2,1),2,2, byrow=TRUE)
matA
```

```
##      [,1] [,2]
## [1,]    4    9
## [2,]    2    1
```

```
matB = matrix(c(2,0,0,7),2,2, byrow=TRUE)
matB
```

```
##      [,1] [,2]
## [1,]    2    0
## [2,]    0    7
```

R Examples

```
matC = matA + matB
matC
```

```
##      [,1] [,2]
## [1,]    6    9
## [2,]    2    8
```

```
matC = matA - matB
matC
```

```
##      [,1] [,2]
## [1,]    2    9
## [2,]    2   -6
```

Scalar Multiplication (element-by-element)

$$
\begin{aligned}
c &= 2 = \text{scalar} \\
A &= \left[\begin{array}{cc} 3 & -1 \\ 0 & 5 \end{array} \right] \\
2 \cdot A &= \left[\begin{array}{cc} 2 \cdot 3 & 2 \cdot (-1) \\ 2 \cdot 0 & 2 \cdot 5 \end{array} \right] = \left[\begin{array}{cc} 6 & -2 \\ 0 & 10 \end{array} \right]
\end{aligned}
$$

R Examples

```
matA = matrix(c(3,-1,0,5), 2, 2,
              byrow=TRUE)
matA
```

```
##      [,1] [,2]
## [1,]    3   -1
## [2,]    0    5
```

```
matC = 2*matA
matC
```

```
##      [,1] [,2]
## [1,]    6   -2
## [2,]    0   10
```

Matrix Multiplication (not element-by-element)

$$\underset{(3\times 2)}{\mathbf{A}} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{bmatrix}, \quad \underset{(2\times 3)}{\mathbf{B}} = \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \end{bmatrix}$$

Note: $\mathbf{A}$ and $\mathbf{B}$ are comformable matrices: $\#$ of columns in $\mathbf{A} = \#$ of rows in $\mathbf{B}$

$$\begin{aligned} &\underset{(3\times 2)}{\mathbf{A}} \cdot \underset{(2\times 3)}{\mathbf{B}} \\ &= \begin{bmatrix} a_{11}b_{11} + a_{12}b_{21} & a_{11}b_{12} + a_{12}b_{22} & a_{11}b_{13} + a_{12}b_{23} \\ a_{21}b_{11} + a_{22}b_{21} & a_{21}b_{12} + a_{22}b_{22} & a_{21}b_{13} + a_{22}b_{23} \\ a_{31}b_{11} + a_{32}b_{21} & a_{31}b_{12} + a_{32}b_{22} & a_{31}b_{13} + a_{32}b_{23} \end{bmatrix} \end{aligned}$$

Remark: In general, $\mathbf{A} \cdot \mathbf{B} \neq \mathbf{B} \cdot \mathbf{A}$

Example

$$\mathbf{A} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, \ \mathbf{B} = \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix}$$

$$\mathbf{A} \cdot \mathbf{B} = \begin{bmatrix} 5+14 & 6+16 \\ 15+28 & 18+32 \end{bmatrix} = \begin{bmatrix} 19 & 22 \\ 43 & 50 \end{bmatrix}$$

- R operator: `A%*%B`
- Excel function: `MMULT(matrix1, matrix2)`

R Examples

```
matA = matrix(1:4,2,2, byrow=TRUE)
matA
```

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

```
matB = matrix(c(1,2,1,3,4,2),2,3, byrow=TRUE)
matB
```

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

R Examples

```
matC = matA%*%matB
matC
```

```
##      [,1] [,2] [,3]
## [1,]    7   10    5
## [2,]   15   22   11
```

Note: `matB%*%matA` will return an error.

Identity Matrix

The $n-$ dimensional identity matrix has all diagonal elements equal to 1, and all off diagonal elements equal to 0.

Example

$$\mathbf{I}_2 = \left[\begin{array}{cc} 1 & 0 \\ 0 & 1 \end{array} \right]$$

- R function: `diag(n)` creates $n-$ dimensional identity matrix

Identity Matrix

The identity matrix plays the roll of 1 in matrix algebra

$$
\begin{aligned}
\mathbf{I}_2 \cdot \mathbf{A} &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \\
&= \begin{bmatrix} a_{11} + 0 & a_{12} + 0 \\ 0 + a_{21} & 0 + a_{22} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \\
&= \mathbf{A}
\end{aligned}
$$

$$
\begin{aligned}
\mathbf{A} \cdot \mathbf{I}_2 &= \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \\
&= \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = \mathbf{A}
\end{aligned}
$$

R Examples

```
matI = diag(2)
matI
```

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

Representing Summation Using Matrix Notation

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

$$\underset{(n\times 1)}{\mathbf{x}} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}, \quad \underset{(n\times 1)}{\mathbf{1}} = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}$$

Then

$$\mathbf{x}'\mathbf{1} = \begin{pmatrix} x_1 & x_2 & \cdots & x_n \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}$$

Representing Summation Using Matrix Notation

Equivalently

$$
\begin{aligned}
\mathbf{1}'\mathbf{x} &= \begin{pmatrix} 1 & 1 & \cdots & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \\
&= x_1 + x_2 + \cdots + x_n = \sum_{i=1}^{n} x_i
\end{aligned}
$$

Sum of Squares

$$\sum_{i=1}^{n} x_i^2 = x_1^2 + x_2^2 + \cdots + x_n^2$$

$$\begin{aligned}
\mathbf{x}'\mathbf{x} &= \begin{pmatrix} x_1 & x_2 & \cdots & x_n \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \\
&= x_1^2 + x_2^2 + \cdots + x_n^2 = \sum_{i=1}^{n} x_i^2
\end{aligned}$$

Sums of cross products

$$
\begin{aligned}
\sum_{i=1}^{n} x_i y_i &= x_1 y_1 + x_2 y_2 + \cdots + x_n y_n \\
\mathbf{x}'\mathbf{y} &= \begin{pmatrix} x_1 & x_2 & \cdots & x_n \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} \\
&= x_1 y_1 + x_2 y_2 + \cdots + x_n y_n = \sum_{i=1}^{n} x_i y_i \\
&= \mathbf{y}'\mathbf{x}
\end{aligned}
$$

R and Excel functions

R functions:

- `t(x)%*%y` or `t(y)%*%x`
- `crossprod(x, y)`

Excel functions:

- `MMULT(TRANSPOSE(x), y)`
- `MMULT(TRANSPOSE(y), x)`

Portfolio Math with Matrix Algebra

Three Risky Asset Example. Let R_i denote the return on asset $i = A, B, C$ and assume that R_A, R_B and R_C are jointly normally distributed with means, variances and covariances:

$$\mu_i = E[R_i],\ \sigma_i^2 = \mathrm{var}(R_i),\ \mathrm{cov}(R_i, R_j) = \sigma_{ij}$$

Portfolio vector **x**:

$$x_i = \text{share of wealth in asset } i$$
$$x_A + x_B + x_C = 1$$

Portfolio return

$$R_{p,x} = x_A R_A + x_B R_B + x_C R_C.$$

Portfolio Math with Matrix Algebra

Portfolio expected return

$$\mu_{p,x} = E[R_{p,x}] = x_A\mu_A + x_B\mu_B + x_C\mu_C$$

Portfolio variance

$$\begin{aligned}\sigma^2_{p,x} = \mathrm{var}(R_{p,x}) &= x_A^2\sigma_A^2 + x_B^2\sigma_B^2 + x_C^2\sigma_C^2 \\ &+ 2x_Ax_B\sigma_{AB} + 2x_Ax_C\sigma_{AC} + 2x_Bx_C\sigma_{BC}\end{aligned}$$

Portfolio distribution

$$R_{p,x} \sim N(\mu_{p,x}, \sigma^2_{p,x})$$

Portfolio Math with Matrix Algebra

Matrix algebra representation

$$
\mathbf{R} = \begin{pmatrix} R_A \\ R_B \\ R_C \end{pmatrix}, \ \mu = \begin{pmatrix} \mu_A \\ \mu_B \\ \mu_C \end{pmatrix}, \ \mathbf{1} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}
$$

$$
\mathbf{x} = \begin{pmatrix} x_A \\ x_B \\ x_C \end{pmatrix}, \ \Sigma = \begin{pmatrix} \sigma_A^2 & \sigma_{AB} & \sigma_{AC} \\ \sigma_{AB} & \sigma_B^2 & \sigma_{BC} \\ \sigma_{AC} & \sigma_{BC} & \sigma_C^2 \end{pmatrix}
$$

Portfolio Math with Matrix Algebra

Portfolio weights sum to 1

$$\mathbf{x}'\mathbf{1} = \begin{pmatrix} x_A & x_B & x_C \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$
$$= x_A + x_B + x_C = 1$$

Digression on Covariance Matrix

Using matrix algebra, the variance-covariance matrix of the $N \times 1$ return vector $\mathbf{R}$ is defined as

$$\underset{N \times N}{\mathrm{var}(\mathbf{R})} = \mathrm{cov}(\mathbf{R}) = E[(\mathbf{R} - \mu)(\mathbf{R} - \mu)'] = \Sigma$$

Because $\mathbf{R}$ has N elements, Σ is the $N \times N$ matrix

$$\Sigma = \begin{pmatrix} \sigma_1^2 & \sigma_{12} & \cdots & \sigma_{1n} \\ \sigma_{12} & \sigma_2^2 & \cdots & \sigma_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{1n} & \sigma_{2n} & \cdots & \sigma_n^2 \end{pmatrix}$$

Digression on Covariance Matrix

For the case $N = 2$, we have

$$
\begin{aligned}
E[(\underset{2\times 1}{\mathbf{R}} - \mu)(\underset{1\times 2}{\mathbf{R}} - \mu)'] &= E\left[\begin{pmatrix} R_1 - \mu_1 \\ R_2 - \mu_2 \end{pmatrix} \cdot (R_1 - \mu_1, R_2 - \mu_2)\right] \\
&= E\left[\begin{pmatrix} (R_1 - \mu_1)^2 & (R_1 - \mu_1)(R_2 - \mu_2) \\ (R_2 - \mu_2)(R_1 - \mu_1) & (R_2 - \mu_2)^2 \end{pmatrix}\right] \\
&= \begin{pmatrix} E[(R_1 - \mu_1)^2] & E[(R_1 - \mu_1)(R_2 - \mu_2)] \\ E[(R_2 - \mu_2)(R_1 - \mu_1)] & E[(R_2 - \mu_2)^2] \end{pmatrix} \\
&= \begin{pmatrix} \mathrm{var}(R_1) & \mathrm{cov}(R_1, R_2) \\ \mathrm{cov}(R_2, R_1) & \mathrm{var}(R_2) \end{pmatrix} = \begin{pmatrix} \sigma_1^2 & \sigma_{12} \\ \sigma_{12} & \sigma_2^2 \end{pmatrix} = \Sigma
\end{aligned}
$$

Portfolio Return and Expected Return

$$\begin{aligned} R_{p,x} &= \mathbf{x}'\mathbf{R} = \begin{pmatrix} x_A & x_B & x_C \end{pmatrix} \begin{pmatrix} R_A \\ R_B \\ R_C \end{pmatrix} \\ &= x_A R_A + x_B R_B + x_C R_C \\ &= \mathbf{R}'\mathbf{x} \end{aligned}$$

$$\begin{aligned} \mu_{p,x} &= \mathbf{x}'\mu = \begin{pmatrix} x_A & x_B & x_X \end{pmatrix} \begin{pmatrix} \mu_A \\ \mu_B \\ \mu_C \end{pmatrix} \\ &= x_A \mu_A + x_B \mu_B + x_C \mu_C \\ &= \mu'\mathbf{x} \end{aligned}$$

R and Excel Formulas

R formulas

- `crossprod(x, mu)`
- `t(x)%*%mu`
- `t(mu)%*%x`

Excel formulas

- `MMULT(TRANSPOSE(xvec), muvec)`
- `MMULT(TRANSPOSE(muvec), xvec)`

Portfolio variance

$$
\begin{aligned}
\sigma_{p,x}^2 &= \mathrm{var}(\mathbf{x}'\mathbf{R}) = E[(\mathbf{x}'\mathbf{R} - \mathbf{x}'\mu)^2] = E[(\mathbf{x}'(\mathbf{R} - \mu))^2] \\
&= E[\mathbf{x}'(\mathbf{R} - \mu)\mathbf{x}'(\mathbf{R} - \mu)] = E[\mathbf{x}'(\mathbf{R} - \mu)(\mathbf{R} - \mu)'\mathbf{x}] = \\
&= \mathbf{x}'E[(\mathbf{R} - \mu)(\mathbf{R} - \mu)']\mathbf{x} = \mathbf{x}'\Sigma\mathbf{x} \\
&= \begin{pmatrix} x_A & x_B & x_C \end{pmatrix} \begin{pmatrix} \sigma_A^2 & \sigma_{AB} & \sigma_{AC} \\ \sigma_{AB} & \sigma_B^2 & \sigma_{BC} \\ \sigma_{AC} & \sigma_{BC} & \sigma_C^2 \end{pmatrix} \begin{pmatrix} x_A \\ x_B \\ x_C \end{pmatrix} \\
&= x_A^2\sigma_A^2 + x_B^2\sigma_B^2 + x_C^2\sigma_C^2 \\
&+ 2x_Ax_B\sigma_{AB} + 2x_Ax_C\sigma_{AC} + 2x_Bx_C\sigma_{BC}
\end{aligned}
$$

R and Excel Formulas

R formula

- `t(x)%*%Sigma%*%x`

Excel formula

- `MMULT(TRANSPOSE(xvec), MMULT(Sigma, xvec))`
- `MMULT(MMULT(TRANSPOSE(xvec), Sigma), xvec)`

Covariance Between 2 Portfolio Returns

portfolios

$$\mathbf{x} = \begin{pmatrix} x_A \\ x_B \\ x_C \end{pmatrix}, \ \mathbf{y} = \begin{pmatrix} y_A \\ y_B \\ y_C \end{pmatrix}$$
$$\mathbf{x}'\mathbf{1} = 1, \ \mathbf{y}'\mathbf{1} = 1$$

Portfolio returns

$$R_{p,x} = \mathbf{x}'\mathbf{R}$$
$$R_{p,y} = \mathbf{y}'\mathbf{R}$$

Covariance Between 2 Portfolio Returns

Covariance

$$\begin{aligned} \mathrm{cov}(R_{p,x}, R_{p,y}) &= \mathbf{x}'\Sigma\mathbf{y} \\ &= \mathbf{y}'\Sigma\mathbf{x} \end{aligned}$$

Derivation

$$\begin{aligned} \mathrm{cov}(R_{p,x}, R_{p,y}) &= \mathrm{cov}(\mathbf{x}'\mathbf{R}, \mathbf{y}'\mathbf{R}) \\ &= E[(\mathbf{x}'\mathbf{R} - \mathbf{x}'\mu])(\mathbf{y}'\mathbf{R} - \mathbf{y}'\mu])] \\ &= E[\mathbf{x}'(\mathbf{R} - \mu)\mathbf{y}'(\mathbf{R} - \mu)] \\ &= E[\mathbf{x}'(\mathbf{R} - \mu)(\mathbf{R} - \mu)'\mathbf{y}] \\ &= \mathbf{x}'E[(\mathbf{R} - \mu)(\mathbf{R} - \mu)']\mathbf{y} \\ &= \mathbf{x}'\Sigma\mathbf{y} \end{aligned}$$

R and Excel Formulas

R formulas

- `t(x)%*%Sigma%*%y`
- `t(y)%*%Sigma%*%x`

Excel formulas

- `MMULT(TRANSPOSE(xvec), MMULT(Sigma, yvec))`
- `MMULT(MMULT(TRANSPOSE(xvec), Sigma), yvec)`

Matrix Inverse

Let $\underset{(n\times n)}{\mathbf{A}}$ = square matrix.

$\mathbf{A}^{-1}$ = "inverse of" $\mathbf{A}$ satisfies

$$\mathbf{A}^{-1}\mathbf{A}=\mathbf{I}_n,\ \mathbf{A}\mathbf{A}^{-1}=\mathbf{I}_n$$

Remark: $\mathbf{A}^{-1}$ is similar to the inverse of a number:

$$\begin{aligned} a = 2,\ a^{-1} &= \frac{1}{2} \\ a \cdot a^{-1} &= 2 \cdot \frac{1}{2} = 1 \\ a^{-1} \cdot a &= \frac{1}{2} \cdot 2 = 1 \end{aligned}$$

R and Excel Functions

- R function: `solve(A)`
- Excel function: `MINVERSE(matrix)`

R Examples

```
matA = matrix(c(1,2,3,4), 2, 2, byrow=TRUE)
matA
```

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

```
matA.inv = solve(matA)
matA.inv
```

```
##      [,1] [,2]
## [1,] -2.0  1.0
## [2,]  1.5 -0.5
```

R Examples

```
matA%*%matA.inv
```

```
##      [,1]         [,2]
## [1,]    1 1.110223e-16
## [2,]    0 1.000000e+00
```

```
matA.inv%*%matA
```

```
##      [,1]         [,2]
## [1,]    1 4.440892e-16
## [2,]    0 1.000000e+00
```

Systems of Linear Equations

Consider the system of two linear equations

$$
\begin{aligned}
x + y &= 1 \\
2x - y &= 1
\end{aligned}
$$

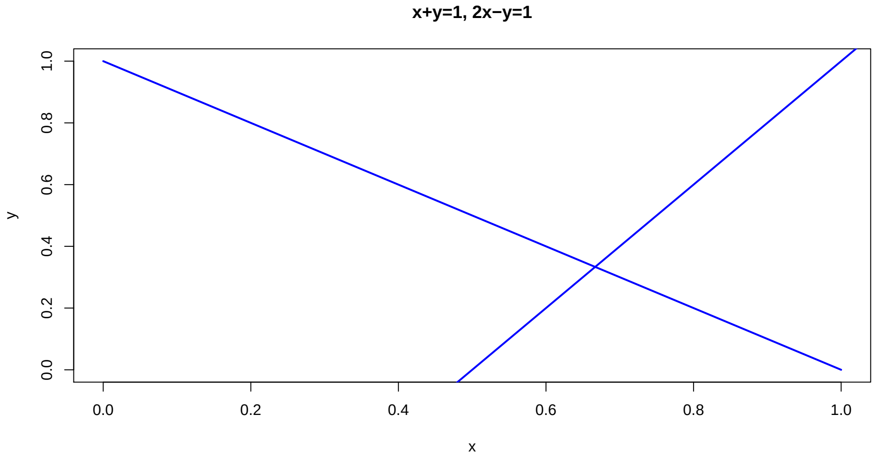
The equations represent two straight lines which intersect at the point

$$x = \frac{2}{3},\ y = \frac{1}{3}$$

Matrix algebra representation:

$$\begin{bmatrix} 1 & 1 \\ 2 & -1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

System of Linear Equations



Systems of Linear Equations

We can write the system as

$$\mathbf{A} \cdot \mathbf{z} = \mathbf{b}$$

where

$$\mathbf{A} = \begin{bmatrix} 1 & 1 \\ 2 & -1 \end{bmatrix}, \ \mathbf{z} = \begin{bmatrix} x \\ y \end{bmatrix} \text{ and } \mathbf{b} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}.$$

Systems of Linear Equations

We can solve for $\mathbf{z}$ by multiplying both sides by $\mathbf{A}^{-1}$

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

or

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 2 & -1 \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

Sytems of Linear Equations

- As long as we can determine the elements in $\mathbf{A}^{-1}$, we can solve for the values of x and y in the vector $\mathbf{z}$.
- Since the system of linear equations has a solution as long as the two lines intersect, we can determine the elements in $\mathbf{A}^{-1}$ provided the two lines are not parallel.

Sytems of Linear Equations

There are general numerical algorithms for finding the elements of $\mathbf{A}^{-1}$ and programs like Excel and R have these algorithms available. However, if $\mathbf{A}$ is a (2×2) matrix then there is a simple formula for $\mathbf{A}^{-1}$. Let

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}.$$

Then

$$\mathbf{A}^{-1} = \frac{1}{\det(\mathbf{A})} \begin{bmatrix} a_{22} & -a_{12} \\ -a_{21} & a_{11} \end{bmatrix}.$$

where

$$\det(\mathbf{A}) = a_{11}a_{22} - a_{21}a_{12} \neq 0$$

Example

Let's apply the above rule to find the inverse of $\mathbf{A}$ in our example:

$$\mathbf{A}^{-1} = \frac{1}{-1-2}\left[\begin{array}{cc} -1 & -1 \\ -2 & 1 \end{array}\right] = \left[\begin{array}{cc} \frac{1}{3} & \frac{1}{3} \\ \frac{2}{3} & \frac{-1}{3} \end{array}\right].$$

Notice that

$$\mathbf{A}^{-1}\mathbf{A} = \left[\begin{array}{cc} \frac{1}{3} & \frac{1}{3} \\ \frac{2}{3} & \frac{-1}{3} \end{array}\right]\left[\begin{array}{cc} 1 & 1 \\ 2 & -1 \end{array}\right] = \left[\begin{array}{cc} 1 & 0 \\ 0 & 1 \end{array}\right].$$

Example

Our solution for $\mathbf{z}$ is then

$$\begin{aligned}\mathbf{z} &= \mathbf{A}^{-1}\mathbf{b} \\ &= \left[\begin{array}{cc} \frac{1}{3} & \frac{1}{3} \\ \frac{2}{3} & \frac{-1}{3} \end{array}\right]\left[\begin{array}{c} 1 \\ 1 \end{array}\right] \\ &= \left[\begin{array}{c} \frac{2}{3} \\ \frac{1}{3} \end{array}\right] = \left[\begin{array}{c} x \\ y \end{array}\right]\end{aligned}$$

so that $x = \frac{2}{3}$ and $y = \frac{1}{3}$. This is the same solution we got before.

R Examples

```
matA = matrix(c(1,1,2,-1), 2, 2, byrow=TRUE)
vecB = c(1,1)
matA.inv = solve(matA)
z = matA.inv%*%vecB
z
```

```
##           [,1]
## [1,] 0.6666667
## [2,] 0.3333333
```

Solving Systems of Linear Equations

In general, if we have n linear equations in n unknown variables we may write the system of equations as

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= b_2 \\ \vdots &= \vdots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n &= b_n \end{aligned}$$

Solving Systems of Linear Equations

We may express this system in matrix form as

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & & & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

Solving Systems of Linear Equations

We may write the system compactly as

$$\underset{(n\times n)}{\mathbf{A}} \cdot \underset{(n\times 1)}{\mathbf{x}} = \underset{(n\times 1)}{\mathbf{b}}.$$

The solution to the system of equations is given by

$$\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$$

where $\mathbf{A}^{-1}\mathbf{A} = \mathbf{I}$ and $\mathbf{I}$ is the $(n \times n)$ identity matrix. If the number of equations is greater than two, then we generally use numerical algorithms to find the elements in $\mathbf{A}^{-1}$.

Bivariate Normal Distribution

Let X and Y be distributed bivariate normal. The joint pdf is given by

$$f(x,y) = \frac{1}{2\pi\sigma_X\sigma_Y\sqrt{1-\rho_{XY}^2}} \times$$

$$\exp\left\{-\frac{1}{2(1-\rho_{XY}^2)}\left[\left(\frac{x-\mu_X}{\sigma_X}\right)^2 + \left(\frac{y-\mu_Y}{\sigma_Y}\right)^2 - \frac{2\rho_{XY}(x-\mu_X)(y-\mu_Y)}{\sigma_X\sigma_Y}\right]\right\}$$

where $E[X]=\mu_X$, $E[Y]=\mu_Y$, $\mathrm{sd}(X)=\sigma_X$, $\mathrm{sd}(Y)=\sigma_Y$, and $\rho_{XY}=\mathrm{cor}(X,Y)$.

Bivariate Normal Distribution

Define

$$\mathbf{X} = \begin{pmatrix} X \\ Y \end{pmatrix},\ \mathbf{x} = \begin{pmatrix} x \\ y \end{pmatrix},\ \mu = \begin{pmatrix} \mu_X \\ \mu_Y \end{pmatrix},\ \Sigma = \begin{pmatrix} \sigma_X^2 & \sigma_{XY} \\ \sigma_{XY} & \sigma_Y^2 \end{pmatrix}$$

Then the bivariate normal distribution can be compactly expressed as

$$f(\mathbf{x}) = \frac{1}{2\pi \det(\Sigma)^{1/2}} e^{-\frac{1}{2}(\mathbf{x}-\mu)'\Sigma^{-1}(\mathbf{x}-\mu)}$$

where

$$\begin{aligned} \det(\Sigma) &= \sigma_X^2\sigma_Y^2 - \sigma_{XY}^2 = \sigma_X^2\sigma_Y^2 - \sigma_X^2\sigma_Y^2\rho_{XY}^2 \\ &= \sigma_X^2\sigma_Y^2(1-\rho_{XY}^2). \end{aligned}$$

Bivariate Normal Distribution

We use the shorthand notation

$$\mathbf{X} \sim N(\mu, \Sigma)$$

This notation extends to n dimensions to define the general multivariate normal distribution

Derivatives of Simple Matrix Functions

Let $\mathbf{A}$ be an $n \times n$ symmetric matrix, and let $\mathbf{x}$ and $\mathbf{y}$ be an $n \times 1$ vectors. Then

$$\frac{\partial}{\partial \mathbf{x}}\mathbf{x}'\mathbf{y} = \begin{pmatrix} \frac{\partial}{\partial x_1}\mathbf{x}'\mathbf{y} \\ \vdots \\ \frac{\partial}{\partial x_n}\mathbf{x}'\mathbf{y} \end{pmatrix} = \mathbf{y},$$

$$\frac{\partial}{\partial \mathbf{x}}\mathbf{A}\mathbf{x} = \begin{pmatrix} \frac{\partial}{\partial x_1}(\mathbf{A}\mathbf{x})' \\ \vdots \\ \frac{\partial}{\partial x_n}(\mathbf{A}\mathbf{x})' \end{pmatrix} = \mathbf{A},$$

$$\frac{\partial}{\partial \mathbf{x}}\mathbf{x}'\mathbf{A}\mathbf{x} = \begin{pmatrix} \frac{\partial}{\partial x_1}\mathbf{x}'\mathbf{A}\mathbf{x} \\ \vdots \\ \frac{\partial}{\partial x_n}\mathbf{x}'\mathbf{A}\mathbf{x} \end{pmatrix} = 2\mathbf{A}\mathbf{x}.$$

Derivatives of Simple Matrix Functions

We will demonstrate these results with simple examples. Let

$$\mathbf{A}=\begin{pmatrix} a & b \\ b & c \end{pmatrix},\ \mathbf{x}=\begin{pmatrix} x_1 \\ x_2 \end{pmatrix},\mathbf{y}=\begin{pmatrix} y_1 \\ y_2 \end{pmatrix}$$

For the first result we have

$$\mathbf{x}'\mathbf{y}=x_1y_1+x_2y_2.$$

Then

$$\frac{\partial}{\partial \mathbf{x}}\mathbf{x}'\mathbf{y}=\begin{pmatrix} \frac{\partial}{\partial x_1}\mathbf{x}'\mathbf{y} \\ \frac{\partial}{\partial x_2}\mathbf{x}'\mathbf{y} \end{pmatrix}=\begin{pmatrix} \frac{\partial}{\partial x_1}(x_1y_1+x_2y_2) \\ \frac{\partial}{\partial x_2}(x_1y_1+x_2y_2) \end{pmatrix}=\begin{pmatrix} y_1 \\ y_2 \end{pmatrix}=\mathbf{y}$$

Derivatives of Simple Matrix Functions

Next, consider the second result. Note that

$$\mathbf{Ax} = \begin{pmatrix} a & b \\ b & c \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} ax_1 + bx_2 \\ bx_1 + cx_2 \end{pmatrix}$$

and

$$(\mathbf{Ax})' = (ax_1 + bx_2, bx_1 + cx_2)$$

Then

$$\frac{\partial}{\partial \mathbf{x}} \mathbf{Ax} = \begin{pmatrix} \frac{\partial}{\partial x_1}(ax_1 + bx_2, bx_1 + cx_2) \\ \frac{\partial}{\partial x_2}(ax_1 + bx_2, bx_1 + cx_2) \end{pmatrix} = \begin{pmatrix} a & b \\ b & c \end{pmatrix} = \mathbf{A}$$

Derivatives of Simple Matrix Functions

Finally, consider the third result. We have

$$\mathbf{x}'\mathbf{A}\mathbf{x} = \begin{pmatrix} x_1 & x_2 \end{pmatrix} \begin{pmatrix} a & b \\ b & c \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = ax_1^2 + 2bx_1x_2 + cx_2^2.$$

Then

$$\begin{aligned} \frac{\partial}{\partial \mathbf{x}}\mathbf{x}'\mathbf{A}\mathbf{x} &= \begin{pmatrix} \frac{\partial}{\partial x_1}\left(ax_1^2 + 2bx_1x_2 + cx_2^2\right) \\ \frac{\partial}{\partial x_2}\left(ax_1^2 + 2bx_1x_2 + cx_2^2\right) \end{pmatrix} = \begin{pmatrix} 2ax_1 + 2bx_2 \\ 2bx_1 + 2cx_2 \end{pmatrix} \\ &= 2\begin{pmatrix} a & b \\ b & c \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = 2\mathbf{A}\mathbf{x}. \end{aligned}$$