# Introduction to Linear Algebra

Nathaniel E. Helwig

Assistant Professor of Psychology and Statistics
University of Minnesota (Twin Cities)

Updated 04-Jan-2017

# Copyright

# Outline of Notes

1) Basic Definitions:
- Vector and matrix
- Transpose and trace
- Symmetric and diagonal
- Special matrices

2) Basic Calculations:
- Matrix equality
- Addition/Subtraction
- Vector products
- Matrix products

3) Matrix Decompositions:
- Eigenvalue (Spectral)
- Cholesky
- Singular Value
- QR

4) Miscellaneous Topics:
- Definiteness
- Determinants
- Inverses/singularity
- R code

# Basic Definitions

# Vectors and Matrices

A vector is a one-dimensional array: $\mathbf{a} = \begin{pmatrix} a_1 \\ \vdots \\ a_n \end{pmatrix}_{n \times 1}$

A matrix is a two-dimensional array: $\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1p} \\ a_{21} & a_{22} & \cdots & a_{1p} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{np} \end{pmatrix}_{n \times p}$

The order of a matrix refers the to number of rows and columns:

- $\mathbf{a}$ has order $n$-by-1
- $\mathbf{A}$ has order $n$-by-$p$

# Rank of a Matrix

The rank of $\mathbf{A}$ is the number of linearly independent rows/columns.

- column rank of $\mathbf{A}$ is number of linearly independent columns
- row rank of $\mathbf{A}$ is number of linearly independent rows

We say that $\mathbf{A}$ is full rank if $\text{rank}(\mathbf{A}) = \min(n, p)$.

- If $n < p$, full rank implies full row rank, i.e., $\text{rank}(\mathbf{A}) = n$
- If $n > p$, full rank implies full column rank, i.e., $\text{rank}(\mathbf{A}) = p$

# Rank Example

The matrix **A** is NOT full rank

$$\mathbf{A} = \begin{pmatrix} 1 & 3 \\ 2 & 6 \\ 5 & 15 \end{pmatrix}$$

because we have $3\mathbf{a}_1 = \mathbf{a}_2$ where $\mathbf{a}_j$ denotes the $j$-th column of **A**.

In contrast, the matrix **A** is full rank

$$\mathbf{A} = \begin{pmatrix} 1 & 3 \\ 2 & 6 \\ 4 & 15 \end{pmatrix}$$

because we cannot write $\sum_{j=1} b_j \mathbf{a}_j = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$ unless we set $b_j = 0 \; \forall j$.

# Matrix Transpose: Definition

We will denote the transpose with a prime symbol (i.e., $'$).

The transpose of a vector turns a column vector into a row vector:

$$\mathbf{a} = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix}_{n\times 1} \iff \mathbf{a}' = \begin{pmatrix} a_1 & a_2 & \cdots & a_n \end{pmatrix}_{1\times n}$$

The transpose of a matrix exchanges rows and columns, such as

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1p} \\ a_{21} & a_{22} & \cdots & a_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{np} \end{pmatrix}_{n\times p} \iff \mathbf{A}' = \begin{pmatrix} a_{11} & a_{21} & \cdots & a_{n1} \\ a_{12} & a_{22} & \cdots & a_{n2} \\ \vdots & \vdots & \ddots & \vdots \\ a_{1p} & a_{2p} & \cdots & a_{np} \end{pmatrix}_{p\times n}$$

# Matrix Transpose: Example

The transpose of $\mathbf{a} = \begin{pmatrix} 1 \\ 7 \\ 5 \\ 9 \end{pmatrix}_{4\times 1}$ is given by $\mathbf{a}' = \begin{pmatrix} 1 & 7 & 5 & 9 \end{pmatrix}_{1\times 4}$

The transpose of $\mathbf{A} = \begin{pmatrix} 1 & 3 \\ 7 & 2 \\ 5 & 7 \\ 9 & 4 \end{pmatrix}_{4\times 2}$ is given by $\mathbf{A}' = \begin{pmatrix} 1 & 7 & 5 & 9 \\ 3 & 2 & 7 & 4 \end{pmatrix}_{2\times 4}$

# Matrix Transpose: Properties

Some useful properties of matrix transposes include:

- $(\mathbf{A}')' = \mathbf{A}$
- $(\mathbf{A} + \mathbf{B})' = \mathbf{A}' + \mathbf{B}'$ (where $\mathbf{A} + \mathbf{B}$ is matrix addition, later defined)
- $(b\mathbf{A})' = b\mathbf{A}'$ (where $b\mathbf{A}$ is scalar multiplication, later defined)
- $(\mathbf{AB})' = \mathbf{B}'\mathbf{A}'$ (where $\mathbf{AB}$ is matrix multiplication, later defined)
- $(\mathbf{A}^{-1})' = (\mathbf{A}')^{-1}$ (where $\mathbf{A}^{-1}$ is matrix inverse, later defined)

# Matrix Trace: Definition

The trace of a square matrix $\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1p} \\ a_{21} & a_{22} & \cdots & a_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ a_{p1} & a_{p2} & \cdots & a_{pp} \end{pmatrix}_{p \times p}$ is

$$\mathrm{tr}(\mathbf{A}) = \sum_{j=1}^{p} a_{jj} \tag{1}$$

which is the sum of the diagonal elements.

# Matrix Trace: Example

The trace of the matrix $\mathbf{A} = \begin{pmatrix} 1 & 4 & 8 & 13 \\ 2 & 8 & 11 & 2 \\ 7 & 2 & 6 & 9 \\ 5 & 9 & 4 & 3 \end{pmatrix}$ is

$$\begin{aligned} \mathrm{tr}(\mathbf{A}) &= 1 + 8 + 6 + 3 \\ &= 18 \end{aligned}$$

# Matrix Trace: Properties

Some useful properties of matrix traces include:

- $\mathrm{tr}(\mathbf{A}) = \mathrm{tr}(\mathbf{A}')$
- $\mathrm{tr}(\mathbf{A} + \mathbf{B}) = \mathrm{tr}(\mathbf{A}) + \mathrm{tr}(\mathbf{B})$
- $\mathrm{tr}(b\mathbf{A}) = b\mathrm{tr}(\mathbf{A})$
- $\mathrm{tr}(\mathbf{AB}) = \mathrm{tr}(\mathbf{BA})$ if both products are defined
- If $\mathbf{A}$ is symmetric, $\mathrm{tr}(\mathbf{A}) = \sum_{j=1}^{p} \lambda_j$ where $\lambda_j$ is $j$-th eigenvalue of $\mathbf{A}$.

# Symmetric Matrix: Definition

A symmetric matrix is square and symmetric along the main diagonal:

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix}_{n \times n} \tag{2}$$

with $a_{ij} = a_{ji}$ for all $i \neq j$.

Note that $\mathbf{A} = \mathbf{A}'$ for all symmetric matrices (by definition).

# Symmetric Matrix: Example

The matrix $\mathbf{A} = \begin{pmatrix} 9 & 1 & 0 & 4 \\ 1 & 4 & 2 & 1 \\ 0 & 2 & 5 & 6 \\ 4 & 1 & 6 & 8 \end{pmatrix}$ is a symmetric $4 \times 4$ matrix.

The matrix $\mathbf{A} = \begin{pmatrix} 9 & 1 & 0 & 4 \\ 1 & 4 & 2 & 1 \\ 0 & 2 & 5 & 6 \\ 3 & 1 & 6 & 8 \end{pmatrix}$ is NOT a symmetric $4 \times 4$ matrix.

# Diagonal Matrix

A diagonal matrix is a square matrix that has zeros in the off-diagonals:

$$\mathbf{D} = \begin{pmatrix} d_1 & 0 & 0 & \cdots & 0 \\ 0 & d_2 & 0 & \cdots & 0 \\ 0 & 0 & d_3 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & d_p \end{pmatrix}_{p \times p} \tag{3}$$

We often write $\mathbf{D} = \mathrm{diag}(d_1, \ldots, d_p)$ to define a diagonal matrix.

# Identity Matrix

The identity matrix of order $p$ is a $p \times p$ matrix that has ones along the main diagonal and zeros in the off-diagonals:

$$\mathbf{I}_p = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{pmatrix}_{p \times p} \tag{4}$$

Note that $\mathbf{I}_p$ is a special type of diagonal matrix.

# Zero and One Matrices

A vector or matrix of all zeros will be denoted using the notation:

$$\mathbf{0}_n = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}_{n \times 1} \qquad \mathbf{0}_{n \times p} = \begin{pmatrix} 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{pmatrix}_{n \times p}$$

A vector or matrix of all ones will be denoted using the notation:

$$\mathbf{1}_n = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}_{n \times 1} \qquad \mathbf{1}_{n \times p} = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ 1 & 1 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 1 \end{pmatrix}_{n \times p}$$

# Basic Calculations

# Matrix Equality

Given two matrices of the same order $\mathbf{A} = \{a_{ij}\}_{n \times p}$ and $\mathbf{B} = \{b_{ij}\}_{n \times p}$, we say that $\mathbf{A}$ is equal to $\mathbf{B}$ (written $\mathbf{A} = \mathbf{B}$) if and only if $a_{ij} = b_{ij}\ \forall i, j$.

If $\mathbf{A} = \begin{pmatrix} 1 & 4 & 8 & 13 \\ 2 & 8 & 11 & 2 \\ 7 & 2 & 6 & 9 \end{pmatrix}$ and $\mathbf{B} = \begin{pmatrix} 1 & 4 & 8 & 13 \\ 2 & 8 & 11 & 2 \\ 7 & 2 & 6 & 9 \end{pmatrix}$, then $\mathbf{A} = \mathbf{B}$.

If $\mathbf{A} = \begin{pmatrix} 1 & 4 & 8 & 13 \\ 2 & 8 & 11 & 2 \\ 7 & 2 & 6 & 9 \end{pmatrix}$ and $\mathbf{B} = \begin{pmatrix} 1 & 4 & 8 & 13 \\ 2 & 8 & 11 & 2 \\ 7 & 2 & 6 & 0 \end{pmatrix}$, then $\mathbf{A} \neq \mathbf{B}$.

# Matrix Addition and Subtraction: Definition

Given two matrices of the same order $\mathbf{A} = \{a_{ij}\}_{n \times p}$ and $\mathbf{B} = \{b_{ij}\}_{n \times p}$, the addition $\mathbf{A} + \mathbf{B}$ produces $\mathbf{C} = \{c_{ij}\}_{n \times p}$ such that $c_{ij} = a_{ij} + b_{ij}$.

Given two matrices of the same order $\mathbf{A} = \{a_{ij}\}_{n \times p}$ and $\mathbf{B} = \{b_{ij}\}_{n \times p}$, the subtraction $\mathbf{A} - \mathbf{B}$ produces $\mathbf{C} = \{c_{ij}\}_{n \times p}$ such that $c_{ij} = a_{ij} - b_{ij}$.

Note: matrix addition and subtraction is only defined for two matrices of the same order.

# Matrix Addition and Subtraction: Example

Given $\mathbf{A} = \begin{pmatrix} 1 & 4 & 8 & 13 \\ 2 & 8 & 11 & 2 \\ 7 & 2 & 6 & 9 \end{pmatrix}$ and $\mathbf{B} = \begin{pmatrix} 5 & 6 & 1 & 7 \\ 1 & 3 & 0 & 2 \\ 2 & 5 & 3 & 5 \end{pmatrix}$, we have that

$$\mathbf{A} + \mathbf{B} = \begin{pmatrix} 1+5 & 4+6 & 8+1 & 13+7 \\ 2+1 & 8+3 & 11+0 & 2+2 \\ 7+2 & 2+5 & 6+3 & 9+5 \end{pmatrix} = \begin{pmatrix} 6 & 10 & 9 & 20 \\ 3 & 11 & 11 & 4 \\ 9 & 7 & 9 & 14 \end{pmatrix}$$

$$\mathbf{A} - \mathbf{B} = \begin{pmatrix} 1-5 & 4-6 & 8-1 & 13-7 \\ 2-1 & 8-3 & 11-0 & 2-2 \\ 7-2 & 2-5 & 6-3 & 9-5 \end{pmatrix} = \begin{pmatrix} -4 & -2 & 7 & 6 \\ 1 & 5 & 11 & 0 \\ 5 & -3 & 3 & 4 \end{pmatrix}$$

# Vector Inner Products: Definition

The inner product of $\mathbf{x} = (x_1, \ldots, x_n)'$ and $\mathbf{y} = (y_1, \ldots, y_n)'$ is

$$\begin{aligned} \mathbf{x}'\mathbf{y} &= \begin{pmatrix} x_1 \cdots x_n \end{pmatrix} \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} \\ &= \left( \sum_{i=1}^{n} x_i y_i \right)_{1 \times 1} \end{aligned} \tag{5}$$

Note that $\mathbf{x}$ and $\mathbf{y}$ must have the same length (i.e., $n$).

# Vector Inner Products: Example

Given $\mathbf{x} = (3, 9, -2, 5)'$ and $\mathbf{y} = (2, 0, 2, 1)'$, we have that

$$\begin{aligned}
\mathbf{x}'\mathbf{y} &= \begin{pmatrix} 3 & 9 & -2 & 5 \end{pmatrix} \begin{pmatrix} 2 \\ 0 \\ 2 \\ 1 \end{pmatrix} \\
&= 3(2) + 9(0) - 2(2) + 5(1) \\
&= 7
\end{aligned}$$

# Vector Outer Products: Definition

The outer product of $\mathbf{x} = (x_1, \ldots, x_m)'$ and $\mathbf{y} = (y_1, \ldots, y_n)'$ is

$$\begin{aligned} \mathbf{x}\mathbf{y}' &= \begin{pmatrix} x_1 \\ \vdots \\ x_m \end{pmatrix} \begin{pmatrix} y_1 \cdots y_n \end{pmatrix} \\ &= \begin{pmatrix} x_1 y_1 & x_1 y_2 & \cdots & x_1 y_n \\ x_2 y_1 & x_2 y_2 & \cdots & x_2 y_n \\ \vdots & \vdots & \ddots & \vdots \\ x_m y_1 & x_m y_2 & \cdots & x_m y_n \end{pmatrix}_{m \times n} \end{aligned} \tag{6}$$

Note that $\mathbf{x}$ and $\mathbf{y}$ can have different lengths (i.e., $m$ and $n$).

# Vector Outer Products: Example

Given $\mathbf{x} = (3, 9, -2, 5)'$ and $\mathbf{y} = (2, 0, 2, 1)'$, we have that

$$\begin{aligned}\mathbf{x}\mathbf{y}' &= \begin{pmatrix} 3 \\ 9 \\ -2 \\ 5 \end{pmatrix} \begin{pmatrix} 2 & 0 & 2 & 1 \end{pmatrix} \\ &= \begin{pmatrix} 6 & 0 & 6 & 3 \\ 18 & 0 & 18 & 9 \\ -4 & 0 & -4 & -2 \\ 10 & 0 & 10 & 5 \end{pmatrix}\end{aligned}$$

# Matrix-Scalar Products: Definition

The matrix-scalar product of $\mathbf{A} = \{a_{ij}\}_{n \times p}$ and $b \in \mathbb{R}$ is

$$\mathbf{A}b = b\mathbf{A} = \begin{pmatrix} ba_{11} & ba_{12} & \cdots & ba_{1p} \\ ba_{21} & ba_{22} & \cdots & ba_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ ba_{n1} & ba_{n2} & \cdots & ba_{np} \end{pmatrix}_{n \times p} \tag{7}$$

which is the matrix $\mathbf{C} = \{c_{ij}\}_{n \times p}$ such that $c_{ij} = ba_{ij}$.

# Matrix-Scalar Products: Example

Given $\mathbf{A} = \begin{pmatrix} 1 & 4 & 8 & 13 \\ 2 & 8 & 11 & 2 \\ 7 & 2 & 6 & 9 \end{pmatrix}$ and $b = 2$, we have that

$$
\begin{aligned}
b\mathbf{A} &= \begin{pmatrix} 1 & 4 & 8 & 13 \\ 2 & 8 & 11 & 2 \\ 7 & 2 & 6 & 9 \end{pmatrix} 2 \\
&= \begin{pmatrix} 2 & 8 & 16 & 26 \\ 4 & 16 & 22 & 4 \\ 14 & 4 & 12 & 18 \end{pmatrix}
\end{aligned}
$$

# Matrix-Vector Products: Definition

The matrix-vector product of $\mathbf{A} = \begin{pmatrix} a_{11} & \cdots & a_{1p} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{np} \end{pmatrix}$ and $\mathbf{x} = \begin{pmatrix} x_1 \\ \vdots \\ x_p \end{pmatrix}$ is

$$\begin{aligned} \mathbf{Ax} &= \begin{pmatrix} a_{11} & \cdots & a_{1p} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{np} \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_p \end{pmatrix} \\ &= \begin{pmatrix} \sum_{j=1}^{p} a_{1j}x_j \\ \vdots \\ \sum_{j=1}^{p} a_{nj}x_j \end{pmatrix}_{n\times 1} \end{aligned} \tag{8}$$

Note that length of **x** must match number of columns of **A** (i.e., $p$).

# Matrix-Vector Products: Example

Given $\mathbf{A} = \begin{pmatrix} 3 & 4 & 1 \\ 4 & 7 & 5 \end{pmatrix}$ and $\mathbf{x} = \begin{pmatrix} 1 \\ 6 \\ 3 \end{pmatrix}$, we have that

$$\begin{aligned} \mathbf{Ax} &= \begin{pmatrix} 3 & 4 & 1 \\ 4 & 7 & 5 \end{pmatrix} \begin{pmatrix} 1 \\ 6 \\ 3 \end{pmatrix} \\ &= \begin{pmatrix} 3(1) + 4(6) + 1(3) \\ 4(1) + 7(6) + 5(3) \end{pmatrix} \\ &= \begin{pmatrix} 30 \\ 61 \end{pmatrix} \end{aligned}$$

# Matrix-Matrix Products: Definition

The matrix-matrix product of $\mathbf{A} = \{a_{ij}\}_{m \times n}$ and $\mathbf{B} = \{b_{jk}\}_{n \times p}$ is

$$\begin{aligned}\mathbf{AB} &= \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{pmatrix} \begin{pmatrix} b_{11} & \cdots & b_{1p} \\ \vdots & \ddots & \vdots \\ b_{n1} & \cdots & b_{np} \end{pmatrix} \\ &= \begin{pmatrix} \sum_{j=1}^n a_{1j}b_{j1} & \sum_{j=1}^n a_{1j}b_{j2} & \cdots & \sum_{j=1}^n a_{1j}b_{jp} \\ \sum_{j=1}^n a_{2j}b_{j1} & \sum_{j=1}^n a_{2j}b_{j2} & \cdots & \sum_{j=1}^n a_{2j}b_{jp} \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{j=1}^n a_{mj}b_{j1} & \sum_{j=1}^n a_{mj}b_{j2} & \cdots & \sum_{j=1}^n a_{mj}b_{jp} \end{pmatrix}_{m \times p}\end{aligned} \tag{9}$$

Note that # of rows of $\mathbf{B}$ must match # of columns of $\mathbf{A}$ (i.e., $n$), and note that $\mathbf{AB} \neq \mathbf{BA}$ even if both products are defined.

# Matrix-Matrix Products: Example

Given $\mathbf{A} = \begin{pmatrix} 3 & 4 & 1 \\ 4 & 7 & 5 \end{pmatrix}$ and $\mathbf{B} = \begin{pmatrix} 1 & 2 \\ 6 & 1 \\ 3 & 4 \end{pmatrix}$, we have that

$$\begin{aligned}
\mathbf{AB} &= \begin{pmatrix} 3 & 4 & 1 \\ 4 & 7 & 5 \end{pmatrix} \begin{pmatrix} 1 & 2 \\ 6 & 1 \\ 3 & 4 \end{pmatrix} \\
&= \begin{pmatrix} 3(1) + 4(6) + 1(3) & 3(2) + 4(1) + 1(4) \\ 4(1) + 7(6) + 5(3) & 4(2) + 7(1) + 5(4) \end{pmatrix} \\
&= \begin{pmatrix} 30 & 14 \\ 61 & 35 \end{pmatrix}
\end{aligned}$$

# Multiplying by Identity Matrix

Given $\mathbf{A} = \{a_{ij}\}_{m \times n}$, pre-multiplying by the identity matrix returns $\mathbf{A}$

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

and post-multiplying by the identity matrix returns $\mathbf{A}$

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

This is the reason we call $\mathbf{I}_m$ and $\mathbf{I}_n$ "identity" matrices.

# Matrix Decompositions

# Overview of Matrix Decompositions

A matrix decomposition decomposes (i.e., separates) a given matrix into a matrix multiplication of two (or more) simpler matrices.

Matrix decompositions are useful for many things:

- Solving systems of equations
- Obtaining low-rank approximations
- Finding important features of data

We will briefly discuss four matrix decompositions:

- Eigenvalue Decomposition
- Cholesky Decomposition
- Singular Value Decomposition
- QR Decomposition

# Eigenvalue (Spectral) Decomposition

The eigenvalue decomposition (EVD) decomposes a symmetric[1] matrix $\mathbf{A} = \{a_{ij}\}_{n \times n}$ into a product of three matrices:

$$\mathbf{A} = \mathbf{\Gamma}\mathbf{\Lambda}\mathbf{\Gamma}' \tag{10}$$

such that

- $\mathbf{\Gamma} = (\gamma_1 \cdots \gamma_n)_{n \times n}$ where $\gamma_j = (\gamma_{1j}, \ldots, \gamma_{nj})'$ is $j$-th eigenvector
- $\mathbf{\Lambda} = \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$ where $\lambda_j$ is $j$-th eigenvalue
- Eigenvalues/vectors are ordered such that $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$

Note that $\mathbf{\Gamma}$ is an orthogonal matrix: $\mathbf{\Gamma}\mathbf{\Gamma}' = \mathbf{\Gamma}'\mathbf{\Gamma} = \mathbf{I}_n$

[1] EVD is defined for asymmetric matrices, but we will only consider symmetric case.

# Cholesky Decomposition

The Cholesky decomposition (CD) decomposes a positive definite matrix $\mathbf{A} = \{a_{ij}\}_{n \times n}$ into a product of a two matrices:

$$\mathbf{A} = \mathbf{L}\mathbf{L}' \tag{11}$$

where

- $\mathbf{L} = \begin{pmatrix} l_{11} & 0 & 0 & \cdots & 0 \\ l_{21} & l_{22} & 0 & \cdots & 0 \\ l_{31} & l_{32} & l_{33} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ l_{n1} & l_{n2} & l_{n3} & \cdots & l_{nn} \end{pmatrix}$ is a lower (left) triangular matrix

# Singular Value Decomposition

The singular value decomposition (SVD) decomposes any matrix $\mathbf{A} = \{a_{ij}\}_{n \times p}$ into a product of three matrices:

$$\mathbf{A} = \mathbf{U}\mathbf{S}\mathbf{V}' \tag{12}$$

such that

- $\mathbf{U} = (\mathbf{u}_1 \cdots \mathbf{u}_r)_{n \times r}$ where $\mathbf{u}_k = \{u_{ik}\}_{n \times 1}$ is $k$-th left singular vector
- $\mathbf{S} = \mathrm{diag}(s_1, \ldots, s_r)$ where $s_k > 0$ is $k$-th singular value
- $\mathbf{V} = (\mathbf{v}_1 \cdots \mathbf{v}_r)_{p \times r}$ where $\mathbf{v}_k = \{v_{jk}\}_{p \times 1}$ is $k$-th right singular vector
- $r \leq \min(m, n)$ and $r = \min(m, n)$ if $\mathbf{A}$ is full-rank

Note that $\mathbf{U}$ and $\mathbf{V}$ are columnwise orthogonal: $\mathbf{U}'\mathbf{U} = \mathbf{V}'\mathbf{V} = \mathbf{I}_r$

# QR Decomposition

The QR decomposition (QRD) decomposes any long (i.e., $n \geq p$) matrix $\mathbf{A} = \{a_{ij}\}_{n \times p}$ into a product of two matrices:

$$\begin{aligned}
\mathbf{A} &= \mathbf{QR} \\
&= \begin{pmatrix} \mathbf{Q}_1 & \mathbf{Q}_2 \end{pmatrix} \begin{pmatrix} \mathbf{R}_1 \\ \mathbf{0}_{(n-p) \times p} \end{pmatrix} \\
&= \mathbf{Q}_1 \mathbf{R}_1
\end{aligned} \tag{13}$$

such that

- $\mathbf{Q}$ is an orthogonal matrix
- $\mathbf{R}_1 = \begin{pmatrix} r_{11} & r_{12} & r_{13} & \cdots & r_{1p} \\ 0 & r_{22} & r_{23} & \cdots & r_{2p} \\ 0 & 0 & r_{33} & \cdots & r_{3p} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & r_{pp} \end{pmatrix}$ is upper (right) triangular matrix

# Miscellaneous Topics

# Quadratic Forms

The quadratic form of a symmetric matrix $\mathbf{A} = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{pmatrix}$ is

$$\begin{aligned} \mathbf{x}'\mathbf{A}\mathbf{x} &= \begin{pmatrix} x_1 & \cdots & x_n \end{pmatrix} \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} \\ &= \left( \sum_{i=1}^n \sum_{j=1}^n x_i x_j a_{ij} \right)_{1\times 1} \end{aligned} \tag{14}$$

where $\mathbf{x} = \begin{pmatrix} x_1 & \cdots & x_n \end{pmatrix}'$ is any arbitrary vector of length $n$.

# Positive, Negative, and Semi-Definite Matrices

A symmetric matrix $\mathbf{A} = \{a_{ij}\}_{n \times n}$ is said to be

- positive definite if $\mathbf{x}'\mathbf{A}\mathbf{x} > 0$ for every $\mathbf{x} \neq \mathbf{0}_n$
- positive semi-definite if $\mathbf{x}'\mathbf{A}\mathbf{x} \geq 0$ for every $\mathbf{x} \neq \mathbf{0}_n$
- negative definite if $\mathbf{x}'\mathbf{A}\mathbf{x} < 0$ for every $\mathbf{x} \neq \mathbf{0}_n$
- negative semi-definite if $\mathbf{x}'\mathbf{A}\mathbf{x} \leq 0$ for every $\mathbf{x} \neq \mathbf{0}_n$

Note if $\mathbf{x}'\mathbf{A}\mathbf{x} \geq 0$ for some $\mathbf{x}$ and $\mathbf{x}'\mathbf{A}\mathbf{x} < 0$ for other $\mathbf{x}$, then $\mathbf{A}$ is said to be an indefinite matrix.

# Matrix Definiteness: Example

The matrix $\mathbf{A} = \begin{pmatrix} 2 & -1 \\ -1 & 2 \end{pmatrix}$ is positive definite:

$$\begin{aligned} \mathbf{x}'\mathbf{A}\mathbf{x} &= \begin{pmatrix} x_1 & x_2 \end{pmatrix} \begin{pmatrix} 2 & -1 \\ -1 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \\ &= \begin{pmatrix} x_1 & x_2 \end{pmatrix} \begin{pmatrix} 2x_1 - x_2 \\ -x_1 + 2x_2 \end{pmatrix} \\ &= 2x_1^2 - 2x_1 x_2 + 2x_2^2 \\ &= x_1^2 + x_2^2 + (x_1 - x_2)^2 \\ &\geq 0 \end{aligned}$$

with the equality holding only when $x_1 = x_2 = 0$.

# Matrix Definiteness: Properties

Let $\lambda_j$ denote the $j$-th eigenvalue of $\mathbf{A}$ for $j \in \{1, \ldots, n\}$.

Some useful properties of matrix definiteness include:

- If $\mathbf{A}$ is positive definite, then $\lambda_j > 0 \; \forall j$
- If $\mathbf{A}$ is positive semi-definite, then $\lambda_j \geq 0 \; \forall j$
- If $\mathbf{A}$ is negative definite, then $\lambda_j < 0 \; \forall j$
- If $\mathbf{A}$ is negative semi-definite, then $\lambda_j \leq 0 \; \forall j$
- If $\mathbf{A}$ is indefinite, then $\lambda_i > 0$ and $\lambda_j < 0$ for some $i \neq j$

# Matrix Determinant: Definition

The determinant of a square matrix $\mathbf{A} \in \mathbb{R}^{p \times p}$ is a real-valued function from $\mathbb{R}^{p \times p} \to \mathbb{R}$, and is typically denoted by $|\mathbf{A}|$ or $\det(\mathbf{A})$.

Determinants provide information about systems of linear equations:

- Suppose that $\mathbf{A} \in \mathbb{R}^{p \times p}$, $\mathbf{x} \in \mathbb{R}^{p \times 1}$, and $\mathbf{b} \in \mathbb{R}^{p \times 1}$
- System $\mathbf{A}\mathbf{x} = \mathbf{b}$ has a unique solution if and only if $|\mathbf{A}| \neq 0$

Determinants provide information about linear transformations:

- Magnitude of $|\mathbf{A}|$ is the transformation's scale factor
- Sign of $|\mathbf{A}|$ is the transformation's orientation

# Matrix Determinant: Calculation

- For $1 \times 1$ matrix $\mathbf{A} = (a)$, we have
  $|\mathbf{A}| = a$

- For $2 \times 2$ matrix $\mathbf{A} = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$, we have
  $|\mathbf{A}| = ad - bc$

- For $3 \times 3$ matrix $\mathbf{A} = \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix}$, we have
  $|\mathbf{A}| = aei + bfg + cdh - (ceg + bdi + afh)$

# Matrix Determinant: Calculation (continued)

For $p \times p$ matrix $\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1p} \\ a_{21} & a_{22} & \cdots & a_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ a_{p1} & a_{p2} & \cdots & a_{pp} \end{pmatrix}$, we have

$|\mathbf{A}| = \sum_{j=1}^{p} (-1)^{i+j} a_{ij} M_{ij} = \sum_{i=1}^{p} (-1)^{i+j} a_{ij} M_{ij}$

where

- $M_{ij} = |\mathbf{A}_{-ij}|$ is the minor corresponding to cell $(i, j)$ of $\mathbf{A}$
- $(-1)^{i+j} M_{ij}$ is the cofactor corresponding to cell $(i, j)$ of $\mathbf{A}$
- $\mathbf{A}_{-ij}$ is the $(p-1) \times (p-1)$ matrix formed by deleting the $i$-th row and $j$-th column of $\mathbf{A}$

Note: can use any column (or row) to define the determinant of **A**.

# Properties of Matrix Determinants

Some useful properties of matrix determinants include:

- $|\mathbf{A}| = |\mathbf{A}'|$
- $|\mathbf{A}^{-1}| = |\mathbf{A}|^{-1}$ (where $\mathbf{A}^{-1}$ is defined on the next slide)
- $|\mathbf{AB}| = |\mathbf{A}||\mathbf{B}|$ (if **A** and **B** are both square)
- $|b\mathbf{A}| = b^p|\mathbf{A}|$ (if $b \in \mathbb{R}$ and **A** is $p \times p$)
- If **A** is square mat., $|\mathbf{A}| = \prod_{j=1}^{p} \lambda_j$ where $\lambda_j$ is $j$-th eigenvalue of **A**.

# Matrix Inverses: Definition

A square (not necessarily symmetric) matrix $\mathbf{A} = \{a_{ij}\}_{n \times n}$ is invertible (or nonsingular) if there exists another matrix $\mathbf{B} = \{b_{ij}\}_{n \times n}$ such that

$$\mathbf{AB} = \mathbf{I}_n \tag{15}$$

where $\mathbf{I}_n$ is the $n \times n$ identity matrix.

If $\mathbf{B}$ exists, the matrix $\mathbf{B}$ is called the inverse of the matrix $\mathbf{A}$ and is denoted by $\mathbf{A}^{-1}$ (so that $\mathbf{AA}^{-1} = \mathbf{A}^{-1}\mathbf{A} = \mathbf{I}_n$).

# Matrix Inverses: Calculation for $2 \times 2$ Case

Claim:

For $2 \times 2$ matrix $\mathbf{A} = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$, we have $\mathbf{A}^{-1} = \frac{1}{ad-bc}\begin{pmatrix} d & -b \\ -c & a \end{pmatrix}$

Proof:

$$\begin{aligned}
\frac{1}{ad-bc}\begin{pmatrix} d & -b \\ -c & a \end{pmatrix}\begin{pmatrix} a & b \\ c & d \end{pmatrix} &= \frac{1}{ad-bc}\begin{pmatrix} da-bc & db-bd \\ -ca+ac & -cb+ad \end{pmatrix} \\
&= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}
\end{aligned}$$

# Matrix Inverses: Example

Given $\mathbf{A} = \begin{pmatrix} 1 & 3 \\ 2 & 1 \end{pmatrix}$, the inverse is $\mathbf{A}^{-1} = \begin{pmatrix} -1/5 & 3/5 \\ 2/5 & -1/5 \end{pmatrix}$:

$$\mathbf{A}\mathbf{A}^{-1} = \begin{pmatrix} 1 & 3 \\ 2 & 1 \end{pmatrix} \begin{pmatrix} -1/5 & 3/5 \\ 2/5 & -1/5 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

$$\mathbf{A}^{-1}\mathbf{A} = \begin{pmatrix} -1/5 & 3/5 \\ 2/5 & -1/5 \end{pmatrix} \begin{pmatrix} 1 & 3 \\ 2 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

# Matrix Inverses: Properties

Some useful properties of matrix inverses include:

- $(\mathbf{A}^{-1})^{-1} = \mathbf{A}$
- $(b\mathbf{A})^{-1} = b^{-1}\mathbf{A}^{-1}$
- $(\mathbf{A}^{-1})' = (\mathbf{A}')^{-1}$
- $\mathbf{A}^{-1} = \mathbf{A}'$ if and only if $\mathbf{A}$ is orthogonal
- $|\mathbf{A}^{-1}| = |\mathbf{A}|^{-1}$
- $(\mathbf{AB})^{-1} = \mathbf{B}^{-1}\mathbf{A}^{-1}$ if both $\mathbf{A}^{-1}$ and $\mathbf{B}^{-1}$ exist
- $\mathbf{A}^{-1}$ exists only if $|\mathbf{A}| \neq 0$
- If $\mathbf{A}$ is symmetric, then $\mathbf{A}^{-1} = \boldsymbol{\Gamma}\boldsymbol{\Lambda}^{-1}\boldsymbol{\Gamma}' = (\mathbf{L}^{-1})'\mathbf{L}^{-1}$, where $\boldsymbol{\Gamma}\boldsymbol{\Lambda}\boldsymbol{\Gamma}'$ and $\mathbf{LL}'$ denote the EVD and CD of $\mathbf{A}$, respectively

# Matrix Function: Overview

To create a matrix in R, we use the `matrix` function.

The relevant inputs of the `matrix` function include

- `data`: the data that will be arranged into a matrix
- `nrow`: the number of rows of the matrix
- `ncol`: the number of columns of the matrix
- `byrow`: logical indicating if the data should be read-in by rows (default reads in data by columns)

# Matrix Function: Example

```
> x = 1:9
> x
[1] 1 2 3 4 5 6 7 8 9
> matrix(x,nrow=3,ncol=3)
     [,1] [,2] [,3]
[1,]    1    4    7
[2,]    2    5    8
[3,]    3    6    9
> matrix(x,nrow=3,ncol=3,byrow=TRUE)
     [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
[3,]    7    8    9
```

# Matrix Function: Warning

R recycles numbers if the dimensions do not conform:

```
> x = 1:9
> x
[1] 1 2 3 4 5 6 7 8 9
> matrix(x,nrow=3,ncol=4)
     [,1] [,2] [,3] [,4]
[1,]    1    4    7    1
[2,]    2    5    8    2
[3,]    3    6    9    3
Warning message:
In matrix(x, nrow = 3, ncol = 4) :
  data length [9] is not a sub-multiple or multiple
  of the number of columns [4]
```

# R Matrix Calculations: Overview

Remember: scalar multiplication is performed using:

```
*
```

In contrast, matrix multiplication is performed using:

```
%*%
```

Note: the matrix multiplication symbol is really three symbols in a row:

- percent sign
- asterisk
- percent sign

# R Matrix Calculations: Example

```
> x = 1:9
> y = 9:1
> X = matrix(x,3,3)
> Y = matrix(y,3,3)
> X
     [,1] [,2] [,3]
[1,]    1    4    7
[2,]    2    5    8
[3,]    3    6    9
> Y
     [,1] [,2] [,3]
[1,]    9    6    3
[2,]    8    5    2
[3,]    7    4    1
```

```
> X * Y
     [,1] [,2] [,3]
[1,]    9   24   21
[2,]   16   25   16
[3,]   21   24    9
```

```
> X %*% Y
     [,1] [,2] [,3]
[1,]   90   54   18
[2,]  114   69   24
[3,]  138   84   30
```

# R Matrix Calculations: Error Messages

```
> x = 1:6
> y = 6:1
> X = matrix(x,2,3)
> Y = matrix(y,3,2)
> X
     [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6
> Y
     [,1] [,2]
[1,]    6    3
[2,]    5    2
[3,]    4    1
```

```
> X * Y
Error in X * Y :
non-conformable arrays
```

```
> X %*% Y
     [,1] [,2]
[1,]   41   14
[2,]   56   20
```

# R Matrix Calculations: Error Messages (continued)

```
> x = 1:6
> y = 6:1
> X = matrix(x,2,3)
> Y = matrix(y,2,3)
> X
     [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6
> Y
     [,1] [,2] [,3]
[1,]    6    4    2
[2,]    5    3    1
```

```
> X * Y
     [,1] [,2] [,3]
[1,]    6   12   10
[2,]   10   12    6
```

```
> X %*% Y
Error in X %*% Y :
non-conformable arguments
```

# Transpose Function

To obtain the transpose of a matrix in R, we use the `t` function.

```
> X = matrix(1:6,2,3)
> X
     [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6
> t(X)
     [,1] [,2]
[1,]    1    2
[2,]    3    4
[3,]    5    6
```

# Dimension Function

To obtain the dimensions of a matrix in R, we use the `dim` function.

```
> X = matrix(1:6,2,3)
> X
     [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6
> dim(X)
[1] 2 3
> dim(t(X))
[1] 3 2
```

# Crossproduct Function

Given $\mathbf{X} = \{x_{ij}\}_{n \times p}$ and $\mathbf{Y} = \{y_{ik}\}_{n \times q}$, we can obtain the crossproduct $\mathbf{X}'\mathbf{Y}$ using the `crossprod` function.

```
> X = matrix(1:6,3,2)
> Y = matrix(1:9,3,3)
> crossprod(X,Y)
     [,1] [,2] [,3]
[1,]   14   32   50
[2,]   32   77  122
> t(X) %*% Y
     [,1] [,2] [,3]
[1,]   14   32   50
[2,]   32   77  122
```

Note that `crossprod` produces same result as using transpose and matrix multiplication symbol.

However, you should prefer `crossprod` because it is faster.

# Transpose-Crossproduct Function

Given $\mathbf{X} = \{x_{ij}\}_{n \times p}$ and $\mathbf{Y} = \{y_{hj}\}_{m \times p}$, we can obtain the transpose-crossproduct $\mathbf{X}\mathbf{Y}'$ using the `tcrossprod` function.

```
> X = matrix(1:6,2,3)
> Y = matrix(1:9,3,3)
> tcrossprod(X,Y)
     [,1] [,2] [,3]
[1,]   48   57   66
[2,]   60   72   84
> X %*% t(Y)
     [,1] [,2] [,3]
[1,]   48   57   66
[2,]   60   72   84
```

Note that `tcrossprod` produces same result as using transpose and matrix multiplication symbol.

However, you should prefer `tcrossprod` because it is faster.

# Row and Column Summation Functions

We can obtain rowwise and columnwise summations using the `rowSums` and `colSums` functions.

```
> X = matrix(1:6,2,3)
> X
     [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6
> rowSums(X)
[1]  9 12
> colSums(X)
[1]  3  7 11
```

# Row and Column Mean Functions

We can obtain rowwise and columnwise means using the `rowMeans` and `colMeans` functions.

```
> X = matrix(1:6,2,3)
> X
     [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6
> rowMeans(X)
[1] 3 4
> colMeans(X)
[1] 1.5 3.5 5.5
```

# Diagonal Function

The `diag` function has multiple purposes:

- If you input a square matrix, `diag` returns the diagonal elements
- If you input a vector, `diag` creates a diagonal matrix
- If you input a scalar, `diag` creates an identity matrix

```
> X = matrix(1:4,2,2)
> X
     [,1] [,2]
[1,]    1    3
[2,]    2    4
> diag(X)
[1] 1 4
```

```
> diag(1:3)
     [,1] [,2] [,3]
[1,]    1    0    0
[2,]    0    2    0
[3,]    0    0    3
> diag(2)
     [,1] [,2]
[1,]    1    0
[2,]    0    1
```

# Functions for Matrix Decompositions

R has built-in functions for popular matrix decompositions:

- Eigenvalue Decomposition: `eigen`
- Cholesky Decomposition: `chol`
- Singular Value Decomposition: `svd`
- QR Decomposition: `qr`

We will not directly use these functions, but some of the methods we will use call these functions internally.

# Eigenvalue Decomposition

```
> X = matrix(1:9,3,3)
> X = crossprod(X)
> xeig = eigen(X,symmetric=TRUE)
> xeig$val
[1] 2.838586e+02 1.141413e+00 6.308738e-15
> xeig$vec
           [,1]       [,2]       [,3]
[1,] -0.2148372  0.8872307  0.4082483
[2,] -0.5205874  0.2496440 -0.8164966
[3,] -0.8263375 -0.3879428  0.4082483
> Xhat = xeig$vec %*% diag(xeig$val) %*% t(xeig$vec)
> sum( (X - Xhat)^2 )
[1] 1.178874e-26
```

# Cholesky Decomposition

```
> set.seed(1)
> X = matrix(runif(9),3,3)
> X = crossprod(X)
> xchol = chol(X)
> t(xchol)
          [,1]      [,2]      [,3]
[1,] 0.7328929 0.0000000 0.0000000
[2,] 1.1336353 0.6224886 0.0000000
[3,] 1.1694863 0.3705306 0.4688907
> Xhat = crossprod(xchol)
> sum( (X - Xhat)^2 )
[1] 0
```

# Singular Value Decomposition

```
> X = matrix(1:6,3,2)
> xsvd = svd(X)
> xsvd$d
[1] 9.5080320 0.7728696
> xsvd$u
           [,1]       [,2]
[1,] -0.4286671  0.8059639
[2,] -0.5663069  0.1123824
[3,] -0.7039467 -0.5811991
> xsvd$v
           [,1]       [,2]
[1,] -0.3863177 -0.9223658
[2,] -0.9223658  0.3863177
> Xhat = xsvd$u %*% diag(xsvd$d) %*% t(xsvd$v)
> sum( (X - Xhat)^2 )
[1] 3.808719e-30
```

# QR Decomposition

```
> X = matrix(1:6,3,2)
> xqr = qr(X)
> Q = qr.Q(xqr)
> Q
           [,1]       [,2]
[1,] -0.2672612  0.8728716
[2,] -0.5345225  0.2182179
[3,] -0.8017837 -0.4364358
> R = qr.R(xqr)
> R
          [,1]      [,2]
[1,] -3.741657 -8.552360
[2,]  0.000000  1.963961
> Xhat = Q %*% R[,sort(xqr$pivot,index=TRUE)$ix]
> sum( (X - Xhat)^2 )
[1] 8.997945e-31
```