

**Introduction to R**

- None

**Matrix algebra**

- Matrix multiplication:

$$\mathbf{AB} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \\ b_{31} & b_{32} \end{bmatrix} = \begin{bmatrix} a_{11}b_{11} + a_{12}b_{21} + a_{13}b_{31} & a_{11}b_{12} + a_{12}b_{22} + a_{13}b_{32} \\ a_{21}b_{11} + a_{22}b_{21} + a_{23}b_{31} & a_{21}b_{12} + a_{22}b_{22} + a_{23}b_{32} \end{bmatrix}$$

- Inverse: For  $\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$ ,  $\mathbf{A}^{-1} = \frac{1}{a_{11}a_{22} - a_{12}a_{21}} \begin{bmatrix} a_{22} & -a_{12} \\ -a_{21} & a_{11} \end{bmatrix}$

- Trace:  $\text{tr}(\mathbf{A}) = \sum_{i=1}^p a_{ii} = a_{11} + a_{22} + \dots + a_{pp}$

- Determinant of  $2 \times 2$ :  $\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} = a_{11}a_{22} - a_{12}a_{21}$

- Eigenvalues: Roots of the polynomial equation  $|\mathbf{A} - \lambda \mathbf{I}| = 0$  where  $\mathbf{I}$  is an identity matrix
- Eigenvectors: Each eigenvalue of  $\mathbf{A}$  has a corresponding nonzero vector  $\mathbf{b}$  that satisfies  $\mathbf{Ab} = \lambda \mathbf{b}$
- For eigenvalues  $\lambda_i$  of  $\mathbf{A}$ :  $\text{tr}(\mathbf{A}) = \sum_{i=1}^p \lambda_i$  and  $|\mathbf{A}| = \prod_{i=1}^p \lambda_i = \lambda_1 \lambda_2 \dots \lambda_p$

- Quadratic formula: The roots of the equation  $ax^2 + bx + c = 0$  are  $\frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$

- Vector length:  $\sqrt{\sum_{i=1}^p a_i^2}$

Positive definite matrices have all eigenvalues greater than 0 and positive semidefinite matrices are the same but with at least one eigenvalue equal to 0

**Data, distributions, and correlation**

- $\rho_{ij} = \frac{\sigma_{ij}}{\sqrt{\sigma_{ii}\sigma_{jj}}} = \frac{\text{Cov}(x_i, x_j)}{\sqrt{\text{Var}(x_i)\text{Var}(x_j)}}$

- $\boldsymbol{\mu} = E(\mathbf{x}) = \begin{bmatrix} E(x_1) \\ \vdots \\ E(x_p) \end{bmatrix} = \begin{bmatrix} \mu_1 \\ \vdots \\ \mu_p \end{bmatrix}$

- $\boldsymbol{\Sigma} = \text{Cov}(\mathbf{x}) = E[(\mathbf{x} - \boldsymbol{\mu})(\mathbf{x} - \boldsymbol{\mu})'] = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \dots & \sigma_{1p} \\ \sigma_{21} & \sigma_{22} & \dots & \sigma_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{p1} & \sigma_{p2} & \dots & \sigma_{pp} \end{bmatrix}$

- $\boldsymbol{\Sigma} = E(\mathbf{xx}') - \boldsymbol{\mu}\boldsymbol{\mu}'$

- $\mathbf{P} = \text{Corr}(\mathbf{x}) = \begin{bmatrix} 1 & \rho_{12} & \dots & \rho_{1p} \\ \rho_{21} & 1 & \dots & \rho_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ \rho_{p1} & \rho_{p2} & \dots & 1 \end{bmatrix}$

- Multivariate normal distribution,  $\mathbf{x} \sim N_p(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ :  $f(\mathbf{x} | \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{p/2} |\boldsymbol{\Sigma}|^{1/2}} e^{-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})' \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu})}$  for  $-\infty < x_i < \infty$  for  $i=1, \dots, p$  and  $|\boldsymbol{\Sigma}| > 0$

- $\hat{\boldsymbol{\mu}} = \frac{1}{N} \sum_{r=1}^N \mathbf{x}_r = \frac{1}{N} (\mathbf{x}_1 + \mathbf{x}_2 + \dots + \mathbf{x}_N)$

- $\hat{\boldsymbol{\Sigma}} = \frac{1}{N-1} \sum_{r=1}^N (\mathbf{x}_r - \hat{\boldsymbol{\mu}})(\mathbf{x}_r - \hat{\boldsymbol{\mu}})'$

- $\hat{\sigma}_{ij} = \widehat{\text{Cov}}(x_i, x_j) = \frac{1}{N-1} \sum_{r=1}^N (x_{ri} - \bar{x}_i)(x_{rj} - \bar{x}_j)$

- $r_{ij} = \frac{\hat{\sigma}_{ij}}{\sqrt{\hat{\sigma}_{ii}\hat{\sigma}_{jj}}} = \frac{\frac{1}{N-1} \sum_{r=1}^N (x_{ri} - \bar{x}_i)(x_{rj} - \bar{x}_j)}{\sqrt{\left[ \frac{1}{N-1} \sum_{r=1}^N (x_{ri} - \bar{x}_i)^2 \right] \left[ \frac{1}{N-1} \sum_{r=1}^N (x_{rj} - \bar{x}_j)^2 \right]}} = \frac{\sum_{r=1}^N (x_{ri} - \bar{x}_i)(x_{rj} - \bar{x}_j)}{\sqrt{\left[ \sum_{r=1}^N (x_{ri} - \bar{x}_i)^2 \right] \left[ \sum_{r=1}^N (x_{rj} - \bar{x}_j)^2 \right]}}$

- $\mathbf{R} = \begin{bmatrix} 1 & r_{12} & \dots & r_{1p} \\ r_{21} & 1 & \dots & r_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ r_{p1} & r_{p2} & \dots & 1 \end{bmatrix}$

- $z_{rj} = \frac{x_{rj} - \hat{\mu}_j}{\sqrt{\hat{\sigma}_{jj}}}$

**R functions** – These functions are listed mostly in the order they were introduced in the notes

### Introduction to R

Function	Description
<code>pnorm()</code>	Finds a cumulative probability from a univariate normal distribution
<code>qnorm()</code>	Finds a quantile from a univariate normal distribution
<code>ls()</code> and <code>objects()</code>	List items in R's database
<code>c()</code>	Combine items into a vector
<code>sd()</code>	Calculate a standard deviation
<code>var()</code>	Calculate a variance
<code>sqrt()</code>	Calculate a square root
<code>pt()</code>	Finds a cumulative probability from a univariate t distribution
<code>qt()</code>	Finds a quantile from a univariate t distribution
<code>t.test()</code>	Calculates quantities associated with a t-test
<code>read.table(file = "c:\\chris\\datafile.txt", header = TRUE, sep = ",")</code>	Read in a text data file with variable names in the first row and spaces separating the variable names and their values. Comma delimited data files can be read in using the <code>sep=", "</code> option. If the first row does not contain the variable names, use <code>header = FALSE</code> and <code>col.names = c("var1", ..., "var.c")</code> to name the variables yourself. The <code>read.csv()</code> function provides a shortcut way to read in comma delimited data files.
<code>summary()</code>	Summarize information in a data frame or list
<code>head()</code>	Print the first few rows of a data frame
<code>write.table(x = set1, file = "C:\\out_file.csv", quote = FALSE, row.names = FALSE, sep=",")</code>	Save data in a data frame to a file on the hard drive. The data was in the data frame <code>set1</code> and it will be written as a comma delimited file named <code>out_file.csv</code> .
<code>library(RODBC)</code> <code>z&lt;-odbcConnectExcel("C:\\gpa.xls")</code> <code>gpa.excel&lt;-sqlFetch(z, "sheet1")</code> <code>close(z)</code>	Read in data from an Excel file called <code>gpa.xls</code> . The data is located on sheet1 of the Excel file.
<code>plot(x = x, y = y)</code>	Plots <code>y</code> on the y-axis and <code>x</code> on the x-axis
<code>lm(formula = y ~ x, data = set1)</code>	Find the sample regression model (and various other measures) with the response variable <code>y</code> and predictor variable <code>x</code> within <code>set1</code>
<code>names()</code>	Provide the names of items in a list
<code>class()</code>	State the class of an object
<code>win.graph(width = 6, height = 6, pointsize = 10)</code>	Opens a new graphics window that is 6"x6" with font size of 10
<code>segments()</code>	Draw a line segment on a plot
<code>curve()</code>	Plot a function of <code>x</code> , like $f(x) = x^2$
<code>expression()</code>	Can be used to put Greek letters and mathematical symbols on a plot
<code>axis()</code>	Allows for finer control of an x or y-axis in a plot

Function	Description
<code>methods()</code>	List the method or generic functions

### Matrix algebra

Function	Description
<code>matrix(data = c(1, 2, 3, 4, 5, 6), nrow = 2, ncol = 3, byrow = TRUE)</code>	Create a matrix of size 2x3 by row
<code>t()</code>	Transpose a matrix
<code>A+B</code>	Matrix addition for matrices <b>A</b> and <b>B</b>
<code>A%*%B</code>	Matrix multiplication for <b>A</b> and <b>B</b>
<code>A*B</code>	Elementwise multiplication for <b>A</b> and <b>B</b>
<code>cbind()</code>	Combine elements by column
<code>solve(A)</code>	Find the inverse of <b>A</b>
<code>diag(A)</code>	Extract the diagonal elements of <b>A</b>
<code>sum(A)</code>	Sum the elements of <b>A</b>
<code>det(A)</code>	Determinant of <b>A</b>
<code>eigen(A)</code>	Find the eigenvalues and eigenvectors of <b>A</b>
<code>abline(h = y)</code>	Plots a horizontal line at <code>y</code> . A vertical line is plotted with the argument <code>v</code> .
<code>arrows()</code>	Draw an arrow on a plot

### Data, distributions, and correlation

Function	Description
<code>cov2cor()</code>	Calculate a correlation matrix from a covariance matrix
<code>dmvnorm()</code>	$f(\mathbf{x})$ for a multivariate normal distribution; this is in the <code>mvtnorm</code> package
<code>seq()</code>	Create a sequence of numbers
<code>persp3d()</code>	3D surface plot; this function is in the <code>rgl</code> package
<code>contour()</code>	Contour plot
<code>cov()</code>	Calculate estimated covariance matrix
<code>cor()</code>	Calculate estimated correlation matrix
<code>colMeans()</code>	Find the means of each column in a matrix
<code>apply()</code>	Apply a function to every row or column of a matrix
<code>set.seed()</code>	Set a seed number
<code>rmvnorm()</code>	Simulate random vectors from a multivariate normal distribution; this function is in the <code>mvtnorm</code> package
<code>points()</code>	Add points to a plot
<code>scale()</code>	Standardize columns of data
<code>expand.grid()</code>	Create all possible combinations of items within separate vectors
<code>par()</code>	Graphics parameters; <code>pty = "s"</code> creates a square plot, <code>mfrow = c(2,2)</code> creates a 2x2 matrix of plots