Test #1 – Formulas and functions
STAT 873

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: $\text{tr}(\mathbf{A}) = \sum_{i=1}^{p} a_{ii} = a_{11} + a_{22} + \ldots + a_{pp}$
- Determinant of 2×2: $\left| \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \right| = a_{11}a_{22} - a_{12}a_{21}$
- Eigenvalues: Roots of the polynomial equation $|\mathbf{A} - \lambda\mathbf{I}| = 0$ where **I** is an identity matrix
- Eigenvectors: Each eigenvalue of **A** has a corresponding nonzero vector **b** that satisfies $\mathbf{Ab} = \lambda\mathbf{b}$
- For eigenvalues λ_i of **A**: $\text{tr}(\mathbf{A}) = \sum_{i=1}^{p} \lambda_i$ and $|\mathbf{A}| = \prod_{i=1}^{p} \lambda_i = \lambda_1\lambda_2\cdots\lambda_p$
- 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

- $\rho_{ij} = \frac{\sigma_{ij}}{\sqrt{\sigma_{ii}\sigma_{jj}}} = \frac{\text{Cov}(x_i, x_j)}{\sqrt{\text{Var}(x_i)\text{Var}(x_j)}}$
- $\boldsymbol{\mu} = E(\mathbf{x}) = \begin{bmatrix} E(x_1) \\ \vdots \\ E(x_p) \end{bmatrix} = \begin{bmatrix} \mu_1 \\ \vdots \\ \mu_p \end{bmatrix}$
- $\boldsymbol{\Sigma} = \text{Cov}(\mathbf{x}) = E[(\mathbf{x}-\boldsymbol{\mu})(\mathbf{x}-\boldsymbol{\mu})'] = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \cdots & \sigma_{1p} \\ \sigma_{21} & \sigma_{22} & \cdots & \sigma_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{p1} & \sigma_{p2} & \cdots & \sigma_{pp} \end{bmatrix}$
- $\boldsymbol{\Sigma} = E(\mathbf{xx}') - \boldsymbol{\mu\mu}'$

- $\mathbf{P} = \text{Corr}(\mathbf{x}) = \begin{bmatrix} 1 & \rho_{12} & \cdots & \rho_{1p} \\ \rho_{21} & 1 & \cdots & \rho_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ \rho_{p1} & \rho_{p2} & \cdots & 1 \end{bmatrix}$
- Multivariate normal distribution, $\mathbf{x} \sim N_p(\boldsymbol{\mu},\boldsymbol{\Sigma})$: $f(\mathbf{x} \mid \boldsymbol{\mu},\boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{p/2}|\boldsymbol{\Sigma}|^{1/2}} e^{-\frac{1}{2}\left[(\mathbf{x}-\boldsymbol{\mu})'\Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu})\right]}$ for $-\infty < x_i < \infty$ for i=1,…,p and $|\boldsymbol{\Sigma}|>0$
- $\hat{\boldsymbol{\mu}} = \frac{1}{N}\sum_{r=1}^{N}\mathbf{x}_r = \frac{1}{N}\left(\mathbf{x}_1 + \mathbf{x}_2 + \ldots + \mathbf{x}_N\right)$
- $\hat{\boldsymbol{\Sigma}} = \frac{1}{N-1}\sum_{r=1}^{N}(\mathbf{x}_r - \hat{\boldsymbol{\mu}})(\mathbf{x}_r - \hat{\boldsymbol{\mu}})'$
- $\hat{\sigma}_{ij} = \text{Cov}(x_i, x_j) = \frac{1}{N-1}\sum_{r=1}^{N}(x_{ri} - \bar{x}_i)(x_{rj} - \bar{x}_j)$
- $r_{ij} = \frac{\hat{\sigma}_{ij}}{\sqrt{\hat{\sigma}_{ii}\hat{\sigma}_{jj}}} = \frac{\frac{1}{N-1}\sum_{r=1}^{N}(x_{ri} - \bar{x}_i)(x_{rj} - \bar{x}_j)}{\sqrt{\left[\frac{1}{N-1}\sum_{r=1}^{N}(x_{ri} - \bar{x}_i)^2\right]\left[\frac{1}{N-1}\sum_{r=1}^{N}(x_{rj} - \bar{x}_j)^2\right]}} = \frac{\sum_{r=1}^{N}(x_{ri} - \bar{x}_i)(x_{rj} - \bar{x}_j)}{\sqrt{\left[\sum_{r=1}^{N}(x_{ri} - \bar{x}_i)^2\right]\left[\sum_{r=1}^{N}(x_{rj} - \bar{x}_j)^2\right]}}$
- $\mathbf{R} = \begin{bmatrix} 1 & r_{12} & \cdots & r_{1p} \\ r_{21} & 1 & \cdots & r_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ r_{p1} & r_{p2} & \cdots & 1 \end{bmatrix}$
- $z_{rj} = \frac{x_{rj} - \hat{\mu}_j}{\sqrt{\hat{\sigma}_{jj}}}$

<u>R functions</u> – These functions are listed mostly in the order they were introduced in the notes.

Introduction to R

Function	**Description**
`pnorm()`	Finds a cumulative probability from a univariate normal distribution
`qnorm()`	Finds a quantile from a univariate normal distribution
`ls()` and `objects()`	List items in R's database
`c()`	Combine items into a vector
`sd()`	Calculate a standard deviation
`var()`	Calculate a variance
`sqrt()`	Calculate a square root
`read.table(file = "c:\\chris\\datafile.txt", header = TRUE, sep = "")`	Read in a text data file with variable names in the first row and spaces separating the variable names and their values.
`read.csv(file = "c:\\chris\\datafile.csv")`	Read in a comma delimited data file.
`summary()`	Summarize information in a data frame or list
`head()`	Print the first few rows of a data frame
`write.table(x = set1, file = "C:\\out_file.csv", quote = FALSE, row.names = FALSE, sep=",")`	Save data in a data frame to a file. The data was in the data frame `set1` and it will be written as a comma delimited file named `out_file.csv`.
`plot(x = x, y = y)`	Plots `y` on the y-axis and `x` on the x-axis
`lm(formula = y ~ x, data = set1)`	Find the sample regression model with the response (dependent) variable `y` and explanatory (independent) variable `x` within `set1`
`names()`	Provide the names of items in a list
`class()`	State the class of an object
`dev.new(width = 6, height = 6, pointsize = 10)`	Opens a new graphics window that is 6"×6" with font size of 10
`segments()`	Draw a line segment on a plot
`curve()`	Plot a function of x, like $f(x) = x^2$
`expression()`	Can be used to put Greek letters and mathematical symbols on a plot
`axis()`	Allows for finer control of an x or y-axis on a plot
`methods()`	List the method or generic functions

Matrix algebra

Function	**Description**
`matrix(data = c(1, 2, 3, 4, 5, 6), nrow = 2, ncol = 3, byrow = TRUE)`	Create a matrix of size 2×3 by row
`t()`	Transpose a matrix
`A+B`	Matrix addition for matrices **A** and **B**
`A%*%B`	Matrix multiplication for **A** and **B**
`A*B`	Elementwise multiplication for **A** and **B**

`cbind()`	Combine elements by column
`solve(A)`	Find the inverse of **A**
`diag(A)`	Extract the diagonal elements of **A**
`sum(A)`	Sum the elements of **A**
`det(A)`	Determinant of **A**
`eigen(A)`	Find the eigenvalues and eigenvectors of **A**
`abline(h = y)`	Plots a horizontal line at `y`. A vertical line is plotted with the argument `v`.
`arrows()`	Draw an arrow on a plot

Data, distributions, and correlation

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