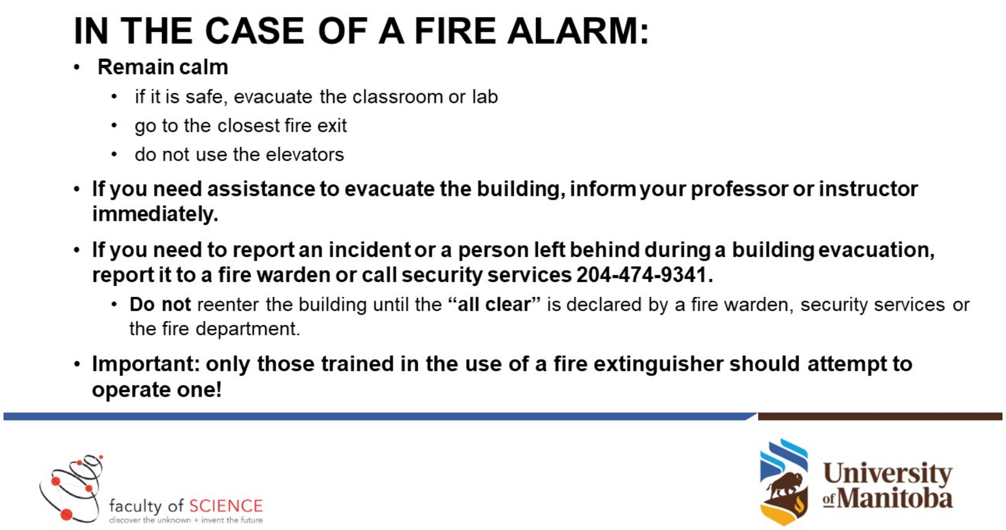
STAT 3690 Lecture Note

Week One (Jan 9, 11, & 13, 2023)

Zhiyang Zhou (zhiyang.zhou@umanitoba.ca, zhiyanggeezhou.github.io)

2023/Jan/08 17:14:33



Syllabus

Contact

- Instructor: Zhiyang (Gee) Zhou, PhD, Asst. Prof.
  - Email: zhiyang.zhou@umanitoba.ca
  - Homepage: zhiyanggeezhou.github.io
- Marker: Masudul Islam
  - Email: islamm8@myumanitoba.ca

Timeline

- Lectures
  - Mon/Wed/Fri 9:30–10:20 am
- Office Hour
  - Wed 10:30–11:20 am
- Assessments
  - Assignments (4/5 times)
  - Midterm
  - Final project

Grading

- Assignments (30%)
    - Scanned/photographed and submitted to Crowdmark
    - Attaching both outputs and source codes (if applicable)
    - Including necessary interpretation
    - Organized in a clear and readable way
    - Accepting NO late submission
- Midterm (35%)
    - Open-book
    - In-person on Mar 10 6–8 pm OR take-home (webcam-invigilated) NOT later than Mar. 20
- Final project (35%)
    - Individual report analying recently collected datasets
    - See the Project Guideline posted at UM Learn

Meterials

- Reading list (recommended but not required)
    - [J&W] R. A. Johnson & D. W. Wichern. (2007). *Applied Multivariate Statistical Analysis*, 5/6th Ed. London: Pearson Education.
        * 2HR print reserve in the Sciences and Technology Library
    - [R&C] A. C. Rencher & W. F. Christensen. (2012). *Methods of Multivariate Analysis*, 3rd Ed. Hoboken: Wiley.
        * Digital copy accessible via the library
    - D. Salsburg (2001). *The Lady Tasting Tea: How Statistics Revolutionized Science in the Twentieth Century*. New York: WH Freeman.
- Lecture notes and beyond
    - zhiyanggeezhou.github.io
    - UM Learn

Outline

- Topics to be covered
    - Matrix manipulation
    - Basics of statistical modeling
    - Multivariate normal distribution
    - Inference on a mean vector
    - Comparisons of several multivariate means
    - Multivariate linear regression
    - Principal component analysis
    - Factor analysis
    - Canonical correlation analysis
    - and so forth

R basics

- Installation
    - download and install BASE *R* from https://cran.r-project.org
    - download and install *Rstudio* from https://www.rstudio.com
    - download and install packages via *Rstudio*
- Working directory
    - When you ask *R* to open a certain file, it will look in the working directory for this file.
    - When you tell *R* to save a data file or figure, it will save it in the working directory.
- Packages

  - installation: install.packages()
  - loading: library()
- Help manual: help(), ?, google, stackoverflow, etc.

- *R* is free but not cheep
  - Open-source
  - Citing packages
  - NO quality control
  - Requiring statistical sophistication
  - Time-consuming to become a master

- References for *R*
  - M. L. Rizzo (2019) Statistical Computing with R, 2nd Ed. (forthcoming)
  - O. Jones, R. Maillardet, A. Robinson (2014) Introduction to Scientific Programming and Simulation Using R, 2nd Ed.
  - ......
- Courses online
  - https://www.pluralsight.com/search?q=R
  - ......
- Data types: let str() or class() tell you
  - numbers (integer, real, or complex)
  - characters ("abc")
  - logical (TRUE or FALSE)
  - date & time
  - factor (commonly encountered in this course)
  - NA (different from Inf, " ' ', 0, NaN etc.)

- Data structures: let str() or class() tell you
  - vector: an ordered collection of the same data type
  - matrix: two-dimensional collection of the same data type
  - array: more than two dimensional collection of the same data type
  - data frame: collection of vectors of same length but of arbitrary data types
  - list: collection of arbitrary objects

- Data input and output
  - create
    * vector: c(), seq(), rep()
    * matrix: matrix(), cbind(), rbind()
    * data frame
  - output: write.table(), write.csv(), write.xlsx()
  - import: read.table(), read.csv(), read.xlsx()
    * header: whether or not assume variable names in first row
    * stringsAsFactors: whether or not convert character string to factors
  - scan(): a more general way to input data
  - save.image() and load(): save and reload workspace
  - source(): run R script

- Parenthesis in *R*
  - paenthesis () to enclose inputs for functions

- square brackets [], [[]] for indexing
- braces {} to enclose forloop or statements such as if or ifelse

---

- Elementary arithmetic operators
  - +, -, *, /, ^
  - log, exp, sin, cos, tan, sqrt
  - FALSE and TRUE becoming 0 and 1, respectively
  - sum(), mean(), median(), min(), max(), var(), sd(), summary()
- Matrix calculation
  - element-wise multiplication: A * B
  - matrix multiplication: A %*% B
  - singlar value decomposition: eigen(A)
- Loops: for() and while()

- Probabilities
  - normal distribution: dnorm(), pnorm(), qnorm(), rnorm()
  - uniform distribution: dunif(), punif(), qunif(), runif()
  - multivariate normal distribution: dmvnorm(), rmvnorm()

- Basic plots
  - strip chart, histogram, box plot, scatter plot
  - Package ggplot2 (RECOMMENDED)

Matrix basics

Matrix decomposition

- Eigendecomposition (for square $n \times n$ matrix $\mathbf{A}_{n\times n}$): $\mathbf{A} = \mathbf{V}\Lambda\mathbf{V}^{-1}$
  - $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_n)$
    - $\lambda_1 \geq \cdots \geq \lambda_n$ are the eigenvalues of $\mathbf{A}$, i.e., n roots of characteristic equation $\det(\lambda\mathbf{I}_n - \mathbf{A}) = 0$
  - $\mathbf{V} = [\boldsymbol{v}_1, \dots, \boldsymbol{v}_n]_{n\times n}$
    - $\boldsymbol{v}_1, \dots, \boldsymbol{v}_n$ are (right) eigenvectors of $\mathbf{A}$, i.e., $\mathbf{A}\boldsymbol{v}_i = \lambda_i\boldsymbol{v}_i$
  - Implementation in *R*: eigen()

- Spectral decomposition (for symmetric $\mathbf{A}$): $\mathbf{A} = \mathbf{V}\Lambda\mathbf{V}^\top$
  - $\mathbf{V}$ is orthogonal, i.e., $\mathbf{V}^\top = \mathbf{V}^{-1}$

- Singular value decomposition (SVD) for $n \times p$ matrix $\mathbf{B}$: $\mathbf{B} = \mathbf{U}\mathbf{S}\mathbf{W}^\top$
  - $\mathbf{U} = [\boldsymbol{u}_1, \dots, \boldsymbol{u}_n]_{n\times n}$ with $\boldsymbol{u}_i$ the ith eigenvector of $\mathbf{B}\mathbf{B}^\top$
    - $\mathbf{U}$ is orthogonal
  - $\mathbf{W} = [\boldsymbol{w}_1, \dots, \boldsymbol{w}_p]_{p\times p}$ with $\boldsymbol{w}_i$ the ith eigenvector of $\mathbf{B}^\top\mathbf{B}$
    - $\mathbf{W}$ is orthogonal

-

$$\mathbf{S} = \left[\begin{array}{c|c} \mathbf{S}_1 & \mathbf{0}_{n\times(p-n)} \end{array}\right]_{n\times p} \text{ if } n \le p \text{ AND } \left[\begin{array}{c} \mathbf{S}_1 \\ \hline \mathbf{0}_{(n-p)\times p} \end{array}\right]_{n\times p} \text{ if } n > p$$

  * $\mathbf{S}_1 = \text{diag}(s_1, \ldots, s_n)$ if $n \le p$ and $\text{diag}(s_1, \ldots, s_p)$ if $n > p$
  * $s_1 \ge \cdots \ge s_n$ are squre roots of eigenvalues of $\mathbf{B}\mathbf{B}^\top$
  * $s_1 \ge \cdots \ge s_p$ are squre roots of eigenvalues of $\mathbf{B}^\top\mathbf{B}$
- Thin/compact SVD for $n \times p$ matrix $\mathbf{B}$:

$$\mathbf{B} = [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_r]\text{diag}(s_1, \ldots, s_r)[\boldsymbol{w}_1, \ldots, \boldsymbol{w}_r]^\top = s_1\boldsymbol{u}_1\boldsymbol{w}_1^\top + \cdots + s_r\boldsymbol{u}_r\boldsymbol{w}_r^\top$$

  * $r = \text{rank}(\mathbf{B}) \le \min\{n, p\}$
  * $s_1 \ge \cdots \ge s_r > 0$ are square roots of non-zero eigenvalues of $\mathbf{B}^\top\mathbf{B}$ or $\mathbf{B}\mathbf{B}^\top$
  * Implementation via *R*: svd()

- Exercise: Is it feasible to apply eigen() only in conducting the thin SVD for a matrix with non-negative singular values (λ_i's)?

Square root of positive (semi-)definite matrix

- $\mathbf{A}$ is positive semi-definite (say $\mathbf{A} \ge 0$) iff $\mathbf{A}$ is symmetric and its eigenvalues are all non-negative
  - Equiv., $\boldsymbol{u}^\top\mathbf{A}\boldsymbol{u} \ge 0$ for any $n \times 1$ real matrix $\boldsymbol{u}$ (say $\boldsymbol{u} \in \mathbb{R}^{n\times 1}$ OR $\boldsymbol{u} \in \mathbb{R}^n$)
- $\mathbf{A}$ is positive definite (say $\mathbf{A} > 0$) iff $\mathbf{A}$ is symmetric and its eigenvalues are all positive
  - Equiv., $\boldsymbol{u}^\top\mathbf{A}\boldsymbol{u} > 0$ for all $\boldsymbol{u} \in \mathbb{R}^n$
- If $\mathbf{A} = \mathbf{V}\Lambda\mathbf{V}^\top$ is the spectral decomposition of positive semi-definite $\mathbf{A}$, then $\mathbf{A}^{1/2} = \mathbf{V}\Lambda^{1/2}\mathbf{V}^\top$ satisfies that
  - $\Lambda^{1/2} = \text{diag}(\lambda_1^{1/2}, \ldots, \lambda_n^{1/2})$
  - $\mathbf{A}^{1/2}\mathbf{A}^{1/2} = \mathbf{A}$
- If $\mathbf{A} = \mathbf{V}\Lambda\mathbf{V}^\top$ is the spectral decomposition of positive definite $\mathbf{A}$, then $\mathbf{A}^{-1/2} = \mathbf{V}\Lambda^{-1/2}\mathbf{V}^\top$ satisfies that
  - $\Lambda^{-1/2} = \text{diag}(\lambda_1^{-1/2}, \ldots, \lambda_n^{-1/2})$
  - $\mathbf{A}^{-1/2}\mathbf{A}^{-1/2} = \mathbf{A}^{-1}$
  - $\mathbf{A}^{1/2}\mathbf{A}^{-1/2} = \mathbf{I}_n$

Determinant and trace

- Applicable only to square matrices
- Properties for determinant
  - $|\mathbf{A}| = \prod_i \lambda_i$
  - $|\mathbf{A}^\top| = |\mathbf{A}|$
  - $|\mathbf{A}^{-1}| = |\mathbf{A}|^{-1}$
  - $|c\mathbf{A}| = c^n|\mathbf{A}|$ for $n \times n$ matrix $\mathbf{A}$ and scalar c
  - $|\mathbf{A}\mathbf{B}| = |\mathbf{A}||\mathbf{B}|$ if $\mathbf{A}$ and $\mathbf{B}$ are square matrices of the identical dimension
- Properties for trace
  - $\text{tr}(\mathbf{A}) = \sum_i \lambda_i$
  - $\text{tr}(c\mathbf{A}) = c\text{tr}(\mathbf{A})$ for scalar c
  - $\text{tr}(\mathbf{A} + \mathbf{B}) = \text{tr}(\mathbf{A}) + \text{tr}(\mathbf{B})$ if $\mathbf{A}$ and $\mathbf{B}$ are square matrices of the identical dimension
  - $\text{tr}(\mathbf{A}\mathbf{B}) = \text{tr}(\mathbf{B}\mathbf{A})$ for $m \times n$ $\mathbf{A}$ and $n \times m$ $\mathbf{B}$
- Remark:$|\mathbf{A}|$ and $\text{tr}(\mathbf{A})$ can be taken as measures of the size of $\mathbf{A}$ when $\mathbf{A}$ is positive definite (i.e., its eigenvalues are all positive).

- Exercise: Prove that
  1. $\mathrm{tr}(\mathbf{AB}) = \mathrm{tr}(\mathbf{BA})$ for $m \times n$ $\mathbf{A}$ and $n \times m$ $\mathbf{B}$.
  2. (The trace trick) $\mathrm{tr}(\mathbf{A}_1 \cdots \mathbf{A}_k) = \mathrm{tr}(\mathbf{A}_{k'+1} \cdots \mathbf{A}_k \mathbf{A}_1 \cdots \mathbf{A}_{k'})$ for $1 < k' < k$.
  3. $\mathrm{tr}(\mathbf{A}) = \sum_i \lambda_i$.
  4. $|\mathbf{A}| = \prod_i \lambda_i$. Hint: Jordan matrix decomposition, i.e., there exists a Jordan normal (or canonical) form $\mathbf{J}$ and invertible $\mathbf{U}$ such that $\mathbf{A} = \mathbf{UJU}^{-1}$ for any square $\mathbf{A}$.

Block/partitioned matrix

- A partition of matrix: Suppose $\mathbf{A}_{11}$ is of $p \times r$, $\mathbf{A}_{12}$ is of $p \times s$, $\mathbf{A}_{21}$ is of $q \times r$ and $\mathbf{A}_{22}$ is of $q \times s$. Make a new $(p+q) \times (r+s)$-matrix by organizing $\mathbf{A}_{ij}$'s in a 2 by 2 way:

$$\mathbf{A} = \left[\begin{array}{c|c} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \hline \mathbf{A}_{21} & \mathbf{A}_{22} \end{array}\right]$$

  e.g.,

$$\mathbf{A} = \left[\begin{array}{cc|c} 1 & 0 & 2 \\ 0 & 1 & 3 \\ \hline 4 & 5 & 6 \end{array}\right]$$

  if

$$\mathbf{A}_{11} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \mathbf{A}_{12} = \begin{bmatrix} 2 \\ 3 \end{bmatrix}, \quad \mathbf{A}_{21} = \begin{bmatrix} 4 & 5 \end{bmatrix}, \quad \text{and} \quad \mathbf{A}_{22} = \begin{bmatrix} 6 \end{bmatrix}.$$

- Operations with block matrices
  - Working with partitioned matrices just like ordinary matrices
  - Matrix addition: if dimensions of $\mathbf{A}_{ij}$ and $\mathbf{B}_{ij}$ are quite the same, then

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

  - Matrix multiplication: if $\mathbf{A}_{ij}\mathbf{B}_{jk}$ makes sense for each i, j, k, then

$$\mathbf{AB} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{B}_{11} & \mathbf{B}_{12} \\ \mathbf{B}_{21} & \mathbf{B}_{22} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{11}\mathbf{B}_{11} + \mathbf{A}_{12}\mathbf{B}_{21} & \mathbf{A}_{11}\mathbf{B}_{12} + \mathbf{A}_{12}\mathbf{B}_{22} \\ \mathbf{A}_{21}\mathbf{B}_{11} + \mathbf{A}_{22}\mathbf{B}_{21} & \mathbf{A}_{21}\mathbf{B}_{12} + \mathbf{A}_{22}\mathbf{B}_{22} \end{bmatrix}$$

  - Inverse: if $\mathbf{A}$, $\mathbf{A}_{11}$ and $\mathbf{A}_{22}$ are all invertible, then

$$\mathbf{A}^{-1} = \begin{bmatrix} \mathbf{A}_{11.2}^{-1} & -\mathbf{A}_{11.2}^{-1}\mathbf{A}_{12}\mathbf{A}_{22}^{-1} \\ -\mathbf{A}_{22}^{-1}\mathbf{A}_{21}\mathbf{A}_{11.2}^{-1} & \mathbf{A}_{22.1}^{-1} \end{bmatrix}$$

    * $\mathbf{A}_{11.2} = \mathbf{A}_{11} - \mathbf{A}_{12}\mathbf{A}_{22}^{-1}\mathbf{A}_{21}$
    * $\mathbf{A}_{22.1} = \mathbf{A}_{22} - \mathbf{A}_{21}\mathbf{A}_{11}^{-1}\mathbf{A}_{12}$