STAT 3355
Introduction to Data Analysis

Lecture 04: R Basics III

Created by Dr. Qiwei Li
Presented by Dr. Octavious Smiley

Department of Mathematical Sciences
The University of Texas at Dallas

Last Class

- Data in `R`
  - Basic data modes: numeric, integer, character, logical
  - Basic data classes: data vector, data matrix, data frame
- Difference among `()`, `[]`, and `{}`
- Structured data vector
  - Simple sequence via `:`
  - Repeated sequence via `rep()`
  - Arithmetic sequence via `seq()`
- Loop statements
  - `for(){}`
  - `while(){}`

Learning Goals

- Know basic R data classes
  - A single variable
  - Data vector
  - Data matrix
  - Data frame
- Know basic data types in mathematics/statistics

Matrix Assignment

- A 2-dimension array of variables that have the same type

$$\boldsymbol{X} = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1p} \\ x_{21} & x_{22} & \dots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \dots & x_{np} \end{bmatrix}$$

- Input a matrix via the function `matrix()`
  - X `<- matrix(c(`x_{11}`,`x_{21} `...,` x_{n1}`, ...,` x_{1p}`,`x_{2p}`,...,` x_{np}`), nrow =` n`, ncol =` p`, byrow = FALSE)`
  - Number of rows is n
  - Number of columns is p
  - Each entry is a numeric, integer, character, or logical
  - If mixing the type, it will be coerced into one type

Matrix Assignment

- Obtain a matrix from multiple vectors via the function `rbind()` or `cbind()`
  - X `<- rbind(`x`, `y`, `$\ldots$`)` or X `<- cbind(`x`, `y`, `$\ldots$`)`
  - If mixing the type, it will be coerced into one type
- Can contain only one entry, one row, or one column
  - X `<- matrix(`x`, nrow = 1, ncol = 1)`
  - X `<- matrix(c(`x_1`,`x_2 `...,` x_p`), nrow = 1, ncol = `p`)`
  - X `<- matrix(c(`x_1`,`x_2 `...,` x_n`), nrow = `n`, ncol = 1)`
- Bind all columns in X to a vector via the function `c(`X`)` or `as.vector(`X`)`

Matrix Assignment

- Examples

```
# Create two 5-by-5 matrices
X <- matrix(1:25, nrow = 5, ncol = 5, byrow
    = FALSE)
Y <- matrix(1:25, nrow = 5, ncol = 5, byrow
    = TRUE)

# Convert a matrix into a vector
x_1 <- c(X)
x_2 <- as.vector(X)
```

Matrix Assignment

- Examples

```
# Inputting whales data
whales_tx <- c(74, 122, 235, 111, 292, 111,
    211, 133, 156, 79)
names(whales_tx) <- 1990:1999

whales_fl <- c(89, 254, 306, 292, 274, 233,
    294, 204, 204, 90)
names(whales_fl) <- 1990:1999

# Create a whale matrix
whales <- rbind(whales_tx, whales_fl)
whales <- cbind(whales_tx, whales_fl)
```

Matrix Name

- Name the matrix via the function `rownames(`X`)` and `colnames(`X`)`
  - Automatic coercion to character
  - Always ensure the completeness of the data

Matrix Name

- Examples

```
whales <- cbind(whales_tx, whales_fl)
colnames(whales) <- c("texas", "florida")
rownames(whales)

whales <- rbind(whales_tx, whales_fl)
rownames(whales) <- c("texas", "florida")
colnames(whales)
```

Single Entry Access

- Access a specified entry via
  - X`[`i`, `j`]`, where i is an integer between 1 and n, and j is an integer between 1 and p
  - X`[-`i`, -`j`]`, negative indices return all entries of the matrix but the i-th row and the j-th column, which is also called (i, j) minor or first minor of the matrix X if it is square
  - X`[a, b]`, where a is an entry in `rownames(`X`)`, and b is an entry in `colnames(`X`)`
  - X`[`k`]`, the k-th entry in `c(`X`)`
    - X`[`k`] == `X`[1 + (`$k-1$`)%%`p`, 1 + (`$k-1$`)%/%`p`]`
- X`[]` is equivalent to X

Row and Column Access

- Access a row via
  - X`[`i`,]`, where i is an integer between 1 and n
  - X`[-`i`,]`, negative index returns all rows of the matrix but the i-th row
  - X`[a,]`, where a is an entry in `rownames(`X`)`
- Access a column via
  - X`[, `j`]`, where j is an integer between 1 and p
  - X`[, -`j`]`, negative index returns all columns of the matrix but the j-th column
  - X`[, b]`, where b is an entry in `colnames(`X`)`

Example

- Data

```
# Input the number of whales beachings per
    year in Texas during the 1990s
whales_tx <- c(74, 122, 235, 111, 292, 111,
    211, 133, 156, 79)
names(whales_tx) <- 1990:1999

# Input the number of whales beachings per
    year in Florida during the 1990s
whales_fl <- c(89, 254, 306, 292, 274, 233,
    294, 204, 204, 90)
names(whales_fl) <- 1990:1999

whales <- rbind(whales_tx, whales_fl)
```

You Turn

- Work on the matrix of whale, where rows correspond to states and columns correspond to years, and answer the following questions
  - What is the number of whales in Florida in 1998?
  - What are the numbers of whales in Texas between 1995 and 1998?
  - What are the numbers of whales in Florida between 1990 and 1999 excluding the year of 1998?

Your Turn

- Solutions

```
whales <- rbind(whales_tx, whales_fl)
rownames(whales) <- c("texas", "florida")

# First question
whales["florida", "1998"]

# Second question
whales["texas", as.character(1995:1998)]

# Third quesiton
whales["florida", -which(colnames(whales) ==
    "1998")]
```

Common Functions

- Common functions for a numeric matrix
  - `sum(`X`)` and `mean(`X`)`
  - `min(`X`)`, `max(`X`)`, `range(`X`)`
  - `sort(`X`)` and `sort(`X`, decreasing = TRUE)`
  - `which.min(`X`)` and `which.max(`X`)`
- All the above functions treat X as a numeric vector `c(`X`)`

Common Functions

- Basic matrix operations for a numeric matrix
  - Transpose: `t(`X`)`
  - Addition and subtraction: X `+` Y and X `-` Y, where X and Y should have the same dimension
  - Scalar multiplication: c`*`X, where $c \in \mathbb{R}$
  - Multiplication: X `%*%` Y, where the number of columns in X should equal to the number of rows in Y
  - Exponentiation: X`^`c, where $c \in \mathbb{R}$
  - Diagonal matrix: `diag(c(`x_1`, ..., `x_n`))`
  - Determinant: `det(`X`)`, where X is a square matrix
  - Inverse: `solve(`X`)`, where X is a square matrix, of which determinant is not zero
  - Eigendecomposition: `eigen(`X`)`, where X is a square matrix

Common Functions

- Common functions to summarize a numeric matrix column-wise or row-wise
  - `rowSums(`X`)` and `colSums(`X`)`
  - `rowMeans(`X`)` and `colMeans(`X`)`
  - `apply(`X`, MARGIN = 1, function)` and `apply(`X`, MARGIN = 2, function)`
- Mathematical operators and functions are applied entry-wise

Common Functions

- Examples

```
# Use the whale matrix where rows correspond
    to years and columns corrspond to states
whales <- t(whales)

colMeans(whales)
colSums(whales)

apply(whales, MARGIN = 2, median)
apply(whales, 2, median)
```

Common Functions

- Other functions
  - `dim()`
  - `mode()`
  - `class()`
  - `is.matrix()` and `as.matrix()`
  - `which(, arr.ind = TRUE)`

Common Functions

- Examples

```
mode(whales)
class(whales)

which(whales == max(whales))
which(whales == max(whales), arr.ind = TRUE)
```

Data Frame

- A 2-dimension array of variables that have the same type within each column

$$
\boldsymbol{X} = \begin{array}{c} \\ \text{Sample } 1 \\ \text{Sample } 2 \\ \vdots \\ \text{Sample } n \end{array}
\begin{array}{c}
\begin{array}{cccc} \text{Variable } 1 & \text{Variable } 2 & \cdots & \text{Variable } p \end{array} \\
\begin{pmatrix}
x_{11} & x_{12} & \dots & x_{1p} \\
x_{21} & x_{22} & \dots & x_{2p} \\
\vdots & \vdots & \ddots & \vdots \\
x_{n1} & x_{n2} & \dots & x_{np}
\end{pmatrix}
\end{array}
$$

- The fundamental data structure by most of `R` functions

Data Frame

$$
\boldsymbol{X} = \begin{array}{c c}
 & \begin{array}{cccc} \text{Variable } 1 & \text{Variable } 2 & \cdots & \text{Variable } p \end{array} \\
\begin{array}{l} \text{Sample } 1 \\ \text{Sample } 2 \\ \vdots \\ \text{Sample } n \end{array} & \begin{pmatrix} x_{11} & x_{12} & \dots & x_{1p} \\ x_{21} & x_{22} & \dots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \dots & x_{np} \end{pmatrix}
\end{array}
$$

- Number of rows is n, also called the number of observations/samples/subjects
- Number of columns is p, also called the number of features/variables/attributes
- Each column is a numeric, integer, character, factor, or logical vector

Data Frame Assignment

- Input the data.frame via the function `data.frame(x_name =` x`, y_name =` y`, z_name =` z`, ...)`
- Convert a matrix $\boldsymbol{X}$ into a data frame via the function
  - `data.frame(`X`)`
  - `melt(`X`)` in the package `reshape2`

Data Frame Assignment

- Examples

```
X <- data.frame(whales)

install.packages("reshape2")
library(reshape2)

X <- melt(whales)
X <- melt(t(whales))
```

Data Frame Name

- Name the variables in a data frame via the function `names(`X`)` and `colnames(`X`)`
- Name the samples in a data frame via the function `rownames(`X`)`
  - Not recommended
  - Let the sample identity number be a variable
- Automatic coercion to character
- Always ensure the completeness of the data

Data Frame

- Examples

```
data_whales <- melt(whales)
names(data_whales) <- c("year", "state", "
    amount")
```

Data Frame Entries Access

- Single entry access: As the same as matrix
- Row access: As the same as matrix
- Column access: As the same as matrix, and
  - X`$`variable_name
  - Access a single entry of a specific variable via X`$`variable_name`[`i`]`, where i is an integer between 1 and n

Data Frame Management

- Sample
  - Delete a sample via X `<-` X`[-`i`,]`
  - Add a sample via X `<-` `rbind(`X`,` x`)`
- Variable
  - Delete a variable via X `<-` X`[, -`j`]`
  - Add a variable via X `<-` `cbind(`X`,` x`)` or X`$x_name` `<-` x

Common Functions

- Common functions for summarize a data frame
  - `str(`X`)`
  - `summary(`X`)`
  - `head(`X`)`
  - `tail(`X`)`
- Most functions for a matrix will be also applicable for a data frame
  - Be aware of data type

Data Frame

- Examples

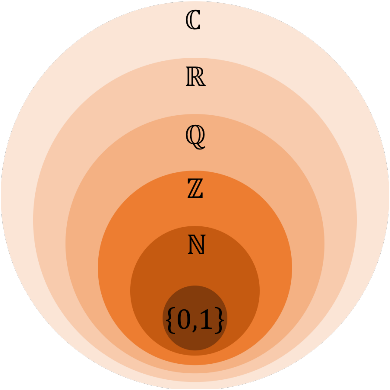
```
str(data_whales)
summary(data_whales)
head(data_whales)
tail(data_whales)
```

Dataset

- A set of numbers, characters, and logical values after a data collection process
- A *variable* is some measurement or characteristic of an item of interest

Number System

- Data is information in digital form that can be transmitted or processed
- The number system



Complex Numbers

- The set of $\{a + bi\}$, where $a, b \in \mathbb{R}$ and $i^2 = -1$
  - Real part: a
  - Imaginary part: b
- If $b = 0$, then the complex number reduces to a real number, of which value is a
- Define a compelx number in `R` via
  - a `+` b`i`
  - The function `complex(real =` a`, imaginary =` b`)`
- Common functions: is.complex(), as.complex(), Re(), Im()

Real Numbers

- The set of all rational numbers ($\mathbb{Q}$) and irrational numbers
  - Rational numbers: The numbers constructed from ratios (or fractions) of integers
  - Irrational numbers: All the real numbers that are not rational numbers, e.g. e, π, $\sqrt{2}$
- Since all values are stored as groupings of bits in a computer, all real numbers in `R` are their approximate rational numbers with finite decimal representations
- Examine the range of a whole real number in `R` via
  - .Machine$double.xmax, which is 2^{1024}
  - The infinity: Inf
- Common functions: is.numeric(), as.numeric()

Continuous Data

- Numeric data: Measurable information that is always collected in number form
- Continuous data: Can only be described on $\mathbb{R}$
- Examples
  - The height of a person in cm
  - The weight of a person in kg or lb
  - Body mass index (BMI): The ratio of the weight to the squared height
  - The age of a person in year, month, day, etc.
  - The learning time for this course per week in minute

Integer Numbers

- The set of all natural numbers (denoted by $\mathbb{N}$) and their additive inverses
  - Natural numbers: $\{0, 1, 2, \ldots\}$, which is countable
- Examine the range of an integer number in R via
  - .Machine$integer.max, which is 2^{31}
- Common functions: is.integer(), as.integer()

Discrete Data

- Discrete data
  - Can be counted
  - Can be turned from continuous data by truncating
- For discrete data we expect that samples share values, whereas for continuous data this will be unlikely
- Examples
  - The whole number of age of a person in year
  - The whole number of learning hours for this course per week

Binary Numbers

- The set of two natural numbers $\{0, 1\}$
- Binary data can take on only two possible states
  - Traditionally labeled as 0 and 1
- Logical data can be viewed as binary data
  - Labeled as false (0) and true (1)
  - as.numeric(TRUE) $==$ 1 and as.logical(1) $==$ TRUE
  - as.numeric(FALSE) $==$ 0 and as.logical(0) $==$ FALSE

Binary Data

- Examples
  - Adult or Nonadult of a person
  - The outcome of an experiment: Failure or success
  - The response to a yes-no question: No or yes
  - The presence or absence of some feature: Absent or present
  - The truth or falsehood of a proposition: False or true

Categorical Data

- Data that records categories
- Take on exactly K possible states, where $K \geq 2$
- Assigned numeric indices, e.g. $\{0, 1, \ldots, K-1\}$
  - May not be meaningfully ordered
  - Cannot be manipulated as numbers
- A character vector x can be viewed as categorical data
  - The possible states can be obtained via the function unique(x)
  - The number of states K can be obtained via length(unique(x))
- A numeric vector x can be turned into as categorical data by binning

Categorical Data

- Examples
  - The political party that a person vote for: Democratic, republican, etc.
  - The blood type of a person: A, B, AB, or O
  - The state that a person was born in: One of the 50 states
  - The age stages of a person: infant $(0, 1]$, toddler $[1, 3)$, Preschooler $[3, 5)$, Gradeschooler $[5, 12)$, teen $[12, 18)$, youth $[18, 30)$, thirties $[30, 40)$, middle-aged $[40, 60)$, elderly $[60, \infty)$

Ordinal Data

- A special categorical data
  - The categories are naturally ordered
  - The distance between the categories may be unknown
- Take on exactly K possible states, where $K \geq 2$
- Assigned numeric indices, e.g. $\{0, 1, \ldots, K-1\}$
  - Meaningfully ordered
  - May not be manipulated as numbers
- Encode a vector x into a factor vector via the function factor(x, levels $=$, labels $=$)
  - Levels: a vector of the unique values (in order) that x might have taken
  - Labels: an optional character vector for the levels

Ordinal Data

- Examples
  - The age stages of a person: Infant $(0, 1]$, toddler $[1, 3)$, Preschooler $[3, 5)$, Gradeschooler $[5, 12)$, teen $[12, 18)$, youth $[18, 30)$, thirties $[30, 40)$, middle-aged $[40, 60)$, elderly $[60, \infty)$
  - The response to a typical survey question: Dislike, dislike somewhat, neutral, like somewhat, like
  - Education levels: Less than 9th grade, high school graduate, associate degree, bachelor's degree, master's degree, doctoral degree

Ordinal Data

- Difference between character, factor, and ordered factor

	Possible values	Order
Character	Anything	Alphabetical
Factor	Fixed and finite	Fixed with alphabetical
Ordered factor	Fixed and finite	Fixed and meaningful

Which Data Type to Choose

- The age of a person:
  - Numeric: In year with finite decimal presentations, e.g. $x = 20.333$
  - Integer: In year with whole numbers, e.g. $x = 20$
  - Categorical: elderly, gradeschooler, middle-aged, preschooler, stage of infant, teen, thirties, toddler, youth, e.g. $x =$'youth'
  - Ordinal: Stage of infant, toddler, preschooler, gradeschooler, teen, youth, thirties, middle-aged, elderly, e.g. $x = 6$
  - Binary: Adult or not, e.g. $x =$TRUE
- Research subjects
- Data privacy

Quiz 2

- Decide which data type is most appropriate for each of the following variables collected in a medical experiment:
  - Subject ID, Name, Treatment, Gender, Number of siblings, Address, Race, Eye color, Birth city

Quiz 2

- Answers:
  - Subject ID: character
  - Name: character
  - Treatment: (ordered) factor
  - Gender: (unordered) factor or logical
  - Number of siblings: integer
  - Address: character
  - Race: (unordered) factor
  - Eye color: (unordered) factor
  - Birth city: character

Which Data Type to Use

- What is the average value?
  - Yes: Continuous numeric
  - Make sense but not be an answer: Discrete numeric
  - Make no sense: Are the values naturally ordered?
    - Yes: Ordinal or logical
    - No: Categorical

Which Data Type to Use in R

- What is the average value?
  - Yes: Numeric
  - Make sense but not be an answer: Integer
  - Make no sense: Are the values naturally ordered?
    - Yes: Factor or logical
    - No: Character