

**Instructor: Nara Yoon, Ph.D. ([yoonnx@jmu.edu](mailto:yoonnx@jmu.edu))**

**Spring 2024 | Thursdays 4-7pm, Hart 3008**

# **Data Analytics for Managerial Leadership**

**WK 7: R basic functions**

# PLAN FOR TODAY

## Overview of core concepts

- Importing data
- Variable types
- Building blocks of data structure

# READINGS

**Matloff, N. (2011). The art of R programming a tour of statistical software design. No Starch Press.**

- **Ch 2: Vectors (2.1 through 2.9)**
- **Ch 3: Matrices and arrays (3.1 through 3.4)**
- **Ch 4: Lists (4.1 through 4.4)**

# CASE IN THE NEWS

**Arnold P., Bargagliotti, A., Franklin, C., and Gould, R. (Jul 28, 2022). Bringing complex data into the classroom. Harvard Data Science Review.**

# Importing data

# DATA IMPORTING

We can use various packages to load files

- **readr – to read csv file (comma or tab delimited files)**
- read.table – to read text file
- haven – to read spss, stata, and sas files
- readxl – to read excel files
- fread – to read large data

# Attribute/variable types

# FOUR TYPES OF ATTRIBUTES

1. Nominal
2. Ordinal
3. Numeric
4. Interval and ratio

|         | Attribute 1: | Attribute 2: | Attribute 3: | Attribute 4:   |
|---------|--------------|--------------|--------------|----------------|
| Name    | Age          | Gender       | Bloodtype    | Highest degree |
| Charlie | 10           | Male         | AB           | High school    |
| Cindy   | 15           | Female       | A            | College        |
| David   | 20           | Male         | B            | Grad school    |
| Flower  | 25           | Female       | O            | High school    |

# ATTRIBUTE TYPES (1): NOMINAL



|             | Attribute 1: | Attribute 2:  | Attribute 3:     | Attribute 4:          |
|-------------|--------------|---------------|------------------|-----------------------|
| <b>Name</b> | <b>Age</b>   | <b>Gender</b> | <b>Bloodtype</b> | <b>Highest degree</b> |
| Charlie     | 10           | Male          | AB               | High school           |
| Cindy       | 15           | Female        | A                | College               |
| David       | 20           | Male          | B                | Grad school           |
| Flower      | 25           | Female        | O                | High school           |

## ATTRIBUTE TYPES (2): ORDINAL



|             | Attribute 1: | Attribute 2:  | Attribute 3:     | Attribute 4:          |
|-------------|--------------|---------------|------------------|-----------------------|
| <b>Name</b> | <b>Age</b>   | <b>Gender</b> | <b>Bloodtype</b> | <b>Highest degree</b> |
| Charlie     | 10           | Male          | AB               | High school           |
| Cindy       | 15           | Female        | A                | College               |
| David       | 20           | Male          | B                | Grad school           |
| Flower      | 25           | Female        | O                | High school           |

# ATTRIBUTE TYPES (3): NUMERIC



|             | Attribute 1: | Attribute 2:  | Attribute 3:     | Attribute 4:          |
|-------------|--------------|---------------|------------------|-----------------------|
| <b>Name</b> | <b>Age</b>   | <b>Gender</b> | <b>Bloodtype</b> | <b>Highest degree</b> |
| Charlie     | 10           | Male          | AB               | High school           |
| Cindy       | 15           | Female        | A                | College               |
| David       | 20           | Male          | B                | Grad school           |
| Flower      | 25           | Female        | O                | High school           |

## ATTRIBUTE TYPES (4): INTERVAL VS. RATIO

**Interval attribute: The “difference” between two value has a useful meaning (scaled variable)**

# Building blocks of data structure

# Building blocks of data structure:

## 1. Objects

# OBJECTS

Building blocks of R element (everything is an object!)

Assign object names

```
g_numbers <- c(2,4,6,8)
g_name <- c("Bill", "Nancy", "Kelly", "Tom")
```

# GIVING NAMES TO AN OBJECT

Case sensitive (upper vs. lower case)

Use letters, underscores, & periods (**always begin with letters!**)

Good examples are:

```
thisData <- c(1,2,3,4)  #camel caps  
this_data <- c(1,2,3,4) #underscore  
this.data <- c(1,2,3,4) #periods
```

# CHECKING ALL OBJECTS IN YOUR WORKSPACE: **objects()** and **ls()**

```
objects()
```

**## show the objects in the workspace**

```
ls()
```

**## equivalent**

# ASSIGNING AN OBJECT

Variable assignment operator <-

Assignment can take the form of...

**Entire data set**

```
x <- read.csv( "FileName.csv" )
```

**Descriptive statistics**

```
x <- mean( y )
```

**Other types of data**

```
x <- c("Adam","Susie","Tony")
```

# PRINTING ASSIGNED OBJECT

**Type** the name of variable

```
x <- seq(1,5)
```

```
x
```

```
## [1] 1 2 3 4 5
```

← After creating object, be sure to type your output to see what it looks like

# TRANSFORMATION

## Creation of new variable using existing variables

```
a <- 1:10
```

What do you think will be the output of this?

```
a+10
```

```
## [1] 11 12 13 14 15 16 17 18 19 20
```

# IDENTIFYING OBJECTS' CHARACTERISTICS

**typeof()** – what is the object's data type?

**length()** – how long is it?

```
Y <- 1:10 # ranges from 1 to 10  
Y
```

```
## [1] 1 2 3 4 5 6 7 8 9 10
```

```
typeof(Y)
```

```
## [1] "integer"
```

```
length(Y)
```

```
## [1] 10
```

# Building blocks of data structure:

## 2. Vectors

# VECTORS

Most basic data structure in R

Vectors can have **only one data type**  
(e.g., numeric, character, factor, logical)

# DATA TYPES (VECTOR TYPES)

**Numeric** – numbers, integers

**Character** – letters (strings; with **quotes** “ ”)

**Factor** – categorical variables (strings with **order**)

**Logical** – TRUE or FALSE values

# CREATING VECTORS

`c()` – to put elements

`rep()` – to repeat elements

`seq()` – to generate sequences of elements

To create vector, mostly use combine function, `c()`

```
g_numeric <- c(10,11,12,13)
g_character <- c("x","y","z")
```

# ADDING ELEMENTS TO EXISTING VECTOR

```
g_logical <- c(TRUE, TRUE, FALSE, FALSE)  
g_logical
```

```
## [1] TRUE TRUE FALSE FALSE
```

```
g_logical <- c(FALSE, g_logical)  
g_logical
```

```
## [1] FALSE TRUE TRUE FALSE FALSE
```

# VECTORS: BASIC FUNCTIONS

```
g_numeric <- c(10,11,12,13)
```

← We created a numeric vector

```
typeof(g_numeric)
```

← typeof(): object storage mode

```
## [1] "double"
```

```
length(g_numeric)
```

← length(): vector length

```
## [1] 4
```

```
class(g_numeric)
```

← class(): class attribute

```
## [1] "numeric"
```

# INDEXING VECTOR ELEMENTS: [ ] OPERATOR

```
name <- c("Charlie","Cindy","David","Flower")  
age <- c(10,15,20,25)
```

```
name
```

```
## [1] "Charlie" "Cindy"  "David"   "Flower"
```

```
name[2]
```

```
## [1] "Cindy"
```

# CHARACTER & FACTOR VECTOR

**Character** – letters (strings; with quotes “ ”)

**Factor** – categorical variables (strings with order)

Character vector

```
X <- c("middle_school", "high_school", "college", "grad_school")
```

Factor vector

```
X <- factor(X)  
levels(X)
```

```
## [1] "college"      "grad_school"  "high_school"  "middle_school"
```

# **In-class activity: Changing data type**

# ACTIVITY: MAKING CHANGES TO DATA TYPE

Create numerical vector `c(1,1,0,0)`

```
g.vector1 <- c(1,1,0,0)
# g.vector1<-as.numeric(g.vector1)
str(g.vector1)
```

```
##  num [1:4] 1 1 0 0
```

# ACTIVITY: MAKING CHANGES TO DATA TYPE

Now, make vector `c(1,1,0,0)` to logical TRUE and FALSE values

```
g.vector2 <- as.logical(g.vector1)
typeof(g.vector2)
```

```
## [1] "logical"
```

```
g.vector2
```

```
## [1] TRUE TRUE FALSE FALSE
```

# ACTIVITY: MAKING CHANGES TO DATA TYPE

Make vector `c(1,1,0,0)` to character values

```
g.vector3 <- as.character(g.vector1)  
typeof(g.vector3)
```

```
## [1] "character"
```

```
g.vector3
```

```
## [1] "1" "1" "0" "0"
```

# Building blocks of data structure:

## 3. Lists

# CREATING LISTS: list()

```
x <- list(1:10, "hello", TRUE, 1 + 3i)
x
```

← We create a list x containing four object groups

```
## [[1]]
##  [1]  1  2  3  4  5  6  7  8  9 10
##
## [[2]]
##  [1] "hello"
##
## [[3]]
##  [1] TRUE
##
## [[4]]
##  [1] 1+3i
```

← We can also assign names to each object groups

```
names(x) <- c("number", "character", "logical", "complex")
```