DATA 133 - Introduction to Data Science I

Instructor: Renzhi Cao
Computer Science Department
Pacific Lutheran University
Fall 2023

Announcements

- Quiz 1 today on Sakai (20 mins)
- Read books: Page 12 - 22
- Don’t forget to comment on Discord!

Reference book

- R Programming for Data Science. By Roger Peng.
  ISBN-10: 1365056821, April 20, 2016.

Pictures from: https://www.google.com/search?
[illegible]

Learning in today

- R basics

Review Demo

- *Create and print a new variable ‘b’ with value 2021?*

- *Calculate the following formula and save it to variable C:*

  *88 * 75 + 25 / 20*

  print out the value of C

R data types: Objects and Attributes

R has five basic or “atomic” classes of objects:

- numeric (real numbers)
- integer
- character
- complex
- logical (True/False)

1. Numeric and integer

- Decimal values are called **numerics** in R

```
> x <- 10.5       # assign a decimal value
> x              # print the value of x
[1] 10.5
> class(x)       # print the class name of x
[1] "numeric"
```

- If you explicitly want an integer, you need to specify the L suffix. So entering 1 in R gives you a numeric object; entering 1L explicitly gives you an integer object

```
> x <- 30L       # assign a integer value
> x              # print the value of x
[1] 30
> class(x)       # print the class name of x
[1] "integer"
```

Poll: “x <- 133.133L”
A: error
B: x will be an integer
C: x will be numeric

1. Numeric and integer

- Inf represents infinity. e.g. 1/0 is Inf,   1/Inf is 0

- NaN represents an undefined value (“not a number”); e.g. 0 / 0;

NaN can also be thought of as a missing value

```
> x <- Inf      # assign a infinity value
> x             # print the value of x
[1] Inf
> y <- 0/0
[1] "NaN"
```

1. Attribute for Numeric and integer

R objects can have attributes, which are like metadata for the object.

Some examples of R object attributes are

- names, dimnames
- dimensions (e.g. matrices, arrays)
- class (e.g. integer, numeric)
- length
- other user-defined attributes/metadata attributes function to access object's

attributes.

2. Characters

- A **character** object is used to represent string values in R.

```
> c <- “Hello World”     # assign a character value
> c                      # print the value of c

> class(c)               # print the class of c
```

- Two character values can be concatenated with the paste function.

```
> c1 <- “Hello”
> c2 <- “World”

> c <- paste(c1,c2)      #  paste(c1,c2, sep = "")
> print(c)
```

- Extract a substr ?   substr(string,start=*,stop=*)

```
> c <- “Hello World”     # assign a character value
> substr(c,start=0,stop=3)
```

3. Complex

- A **complex** value in R is defined via the pure imaginary value *i*.

```
> c <- 1 + 2i                # assign a complex value
> c                          # print the value of c

> class(c)                   # print the class of c
```

4. Logical

- A **logical** value is often created via comparison between variables.

```
> x <- 1
> y <- 2
> z = x>y
> print(z)

> class(z)
```

- Standard logical operations are "&" (and), "|" (or), and "!" (negation).

```
> x <- (1>2)

> y <- (2>1)

> z <- x&y

> print(z)
```

Practice

1. Use numeric class object to calculate multiplication of 2 and 3 and assign it to x

2. Continue Q1, how about using integer class object?

3. Assign “DATA133” to x and print out the length and type of x

4. Update x in Q3 by adding “ is great” at the end.

5. What should be the value of the following x?

Practice

1. Use numeric class object to calculate multiplication of 2 and 3 and assign it to x
   - x <- 2 * 3
2. Continue Q1, how about using integer class object?
   - x <- 2L * 3L
3. Assign “DATA133” to x and print out the length and type of x
   - x <- “DATA133”
   - print(length(x))
   - class(x)
4. Update x in Q3 by adding “ is great” at the end.
   - x <- paste(x,” is great”)
5. What should be the value of the following x?
   - x <- (5>2) | (6<5)
   - x <- (5>2) & (6<5)

Extra practice for logical

X	Y	X&Y	X|Y	!X
T	F			
T	T			
F	T			
F	F			

Vectors and Lists

- The c() function can be used to create vectors of objects by concatenating things together.

```
> x <- c(0.5, 0.6)        ## numeric
>x <- c(1L,3L)               ## integer
> x <- c(TRUE, FALSE).  ##logical
> x <- c(T, F)                   ##logical
```

```
> x <- c("a", "b", "c") ## character
> x <- 2:13                    ## integer
> x <- c(1+0i, 2+4i)  ## complex
> x[0]                          # print the class type of x
> class(x)          # show the class of variable x
> x[1]                # print the first element of x
```

Vectors and Lists

- Vector operations and subsets

```
> x <- c(1,3,5,6,7)
> y <- c(2,5,7,3,6)
> x - y
> x + y
```

```
> x <- c(1,3,5,6,7,9,10)
> x[2:4]
```

Practice demo

Try the following:

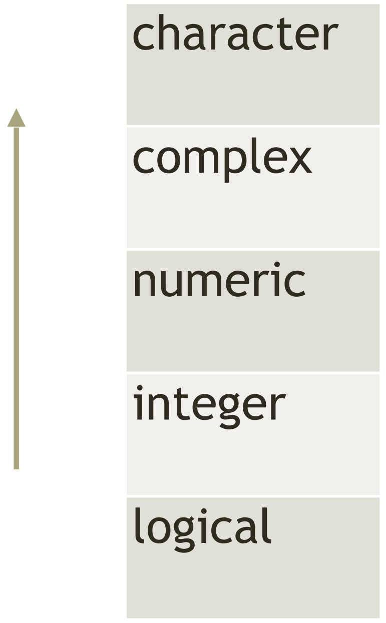
```
> x <- c(10, 20)
> x
> x+3

>x <- c(10, “a”)
>x
>x + 3
```

Vectors and Lists

There are occasions when different classes of R objects get mixed together.

```
> y <- c(1.7, "a") ## character
> y <- c(TRUE, 2) ## numeric
> y <- c("a", TRUE) ## character
```



Vectors and Lists

- >x <- c(“abc”,10)
- >x[2]+3   # not working

- > as.numeric(x)   # convert x to numeric class
- > as.integer(x)   # convert x to integer class
- > as.logical(x)   # convert x to logical class
- > as.character(x) # convert x to character class

- > as.numeric(x[2]) + 3 # convert to numeric class

Vectors and Lists

Lists are special type of vector that contain elements of different classes.

- *x <- list(“abc”,10)*
- *> x[[2]] + 3*

- *> x<-c(“abc”,10)*
- *> as.numeric(x[2]) + 3*

Guided program (Download Inclass3.R on web and use source)

1. **Create a vector v, and add two elements: “hello”, 133**
2. Print the second element of v
3. Convert the second element of v to integer I
4. Create vector v2 which contains number from 1 to 6
5. Create vector v3 which adds integer I to each element of v2
6. Create a list l which contains same element of v
7. Get the length of v, v2, v3 and l
8. Convert vector v to a list vl, and compare the value of vl and l

Practice of R

Work on PairProgram3.R from website

Pair-programming PairProgram3.R and PairProgram4.R will be due by next Monday, submit on Sakai

Read book

Don't forget to take the quiz 1

Break

- Take the quiz on Sakai now