Statistical Bioinformatics Lab: More Advanced R

January 18, 2023

(adapted from Jasmine Chong)

Data Manipulation in R

- We will now cover useful functions for data manipulation and dealing with missing data in R
- Knowing these tricks will save you time… and hair!

- Read in “kidney.csv”
- **> kidney <- read.csv("kidney.csv")**
- **> str(kidney) #shows structure of data frame**
- We can see “NA”s – this is R telling us there are missing values

Missing Values in R

- **> mean(kidney$weight) # shows NA because it contains missing values!**
- **[1] NA**

- Need to tell how R to deal with missing values.
- First need to identify how many values are missing.
  - **> is.na(kidney$weight) #logical assessment if values are missing**
  - **> missing1 <- as.numeric(is.na(kidney$weight))**
  - **> sum(missing1)**

Missing Values in R

- **Option 1: Complete Removal**
  - In general, it is bad form to ignore missing data.

- Calculate mean without missing data
  - **> mean(kidney$weight, na.rm=T)**
- Subset kidney data, keeping only samples with no missing data
  - **> complete <- complete.cases(kidney)** #logical vector identifying rows with complete information
  - **> kidney.complete <- kidney[complete,]** #subset only where rows are complete (TRUE)

Missing Values in R

- **Option 2: Replace missing with mean/min values**

- **> missing <- is.na(kidney$weight)** # identify which are missing (true if missing)
- **> mean(kidney$weight[!missing])** # calculate mean weight of not missing values
- **> kidney$weight[missing] <- 72.8** # assign missing values the mean

Demo

Functions in R

- Remember: A function is an organized set of commands to perform a specific task
  - Name
  - Input
  - R commands to do something to the input – wrapped in { }
  - Output

Writing a Function in R

- What will it do?
- What is the input (parameters)?
  - Need to feed it the data/parameters.
- What is the output?
  - What does it return? A single value? The entire matrix?
  - Can only return a single R object.
- Give your function a meaningful name!

Example of a function

- What does it do?
  - Replace missing values with the mean of non-missing values.
- Input?
  - A numerical matrix with missing values.
- Output?
  - A matrix with missing values replaced.

Examples of a Function

```
replace_missing <- function(data=NA){
  missing.inx <- is.na(data)
  non.missing.mean <- mean(data[!missing.inx])
  data[missing.inx] <- non.missing.mean
  return(data)
}

> kidney <- replace_missing(kidney) # run the function!
```

Demo

Conditions in R

- Decision making is important for programming
- Evaluate if statements are TRUE or FALSE
- **If** statements in R

```
> if (condition1) {
      Statement1
  }
```

- If statement is TRUE, statement gets executed. If FALSE, nothing happens. (Example in RStudio)

If - else statements

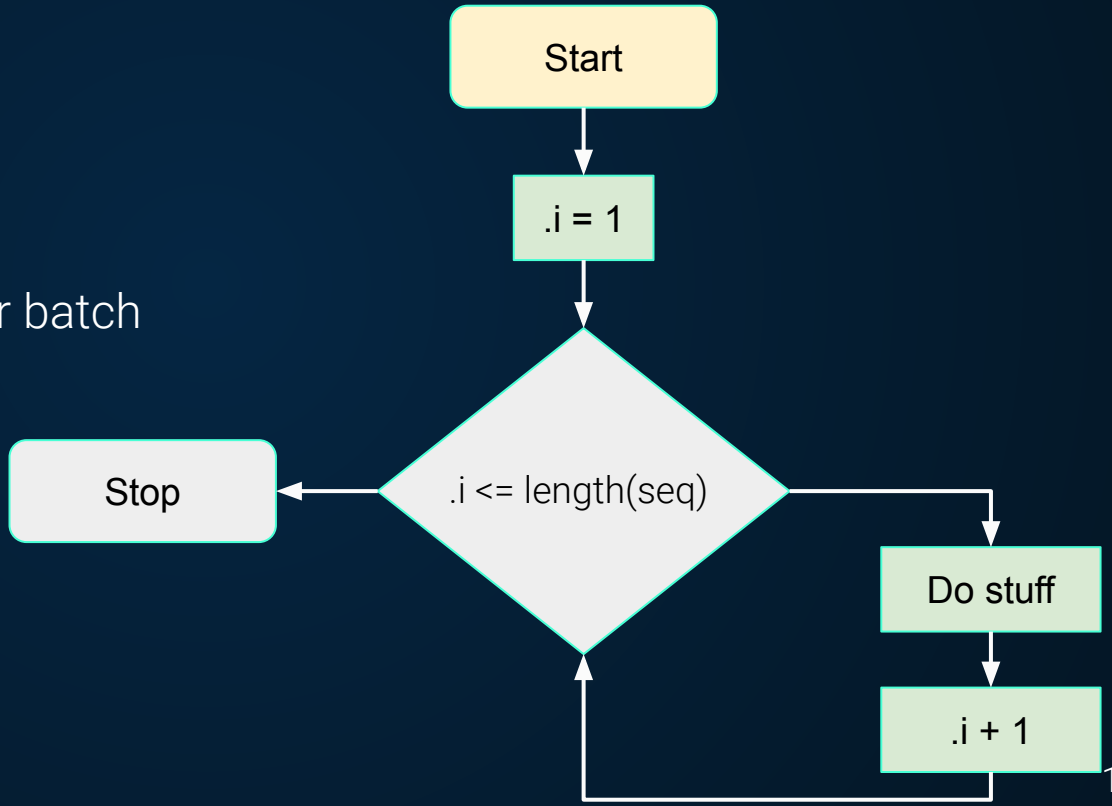
```
>if (condition1){
        statement1
} else {
        statement2
}
```

- Else only executed if condition1 is FALSE.
- Else must be in the same line as the closing brace for the if statement. (Example RStudio)

Demo

For loops in R

- Repetitive execution
- What are loops?
  - Automated repeating instructions - used for batch processes.



For loops in R

```
> for (variable in sequence){
      statement1
}
```

- Variable = loop variable
- Sequence = vector expression, usually a sequence like 1:10
- For each iteration of the loop, the variable gets assigned a value from the sequence and the statement is evaluated.

For loops in R

```
> workshop <- c(“Tim”, “Laura”, “Buzz”, “Charlie”)
> for (student in 1:length(workshop)){
      print(c(“Hello”, student))
}
```

For loops in R… a little trickier

```
> x <- 1:10
> y <- rep(0, 10) # empty vector to contain results of loop

> for (i in length(x)){
>     y[i] <- sqrt(x[i])
>}
```

Demo

Intro to R Apply Family

- **apply(X, MARGIN, FUN, …)**
- X is the matrix, list or dataframe
- MARGIN defines how the function is applied - 1: over rows, 2: over columns
- FUN is the function you want to apply

> matrix <- matrix(rnorm(30), nrow=5, ncol=6) # create a matrix
> output <- apply(matrix, 2, sum) # sum values of each column

Demo

T-tests in R

- One of the most common statistical tests.
- Does the means of two vectors differ significantly?
- In the kidney data:

```
> t.test(kidney$age[1:3], kidney$age[4:6])

        Welch Two Sample t-test

data:  kidney$age[1:3] and kidney$age[4:6]
t = 0.11637, df = 3.3729, p-value = 0.9139
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -41.18939  44.52272
sample estimates:
mean of x mean of y
 61.66667  60.00000
```

Demo + Practice

Mini Activity

- Give row names to the kidney dataset.
- Using an **if-else** condition, check if the 10th patient is young (<18), middle-aged (>18 but < 60), or old (>60).
- Using a **for loop** (for all patients now), if the creatinine level is greater than 1.2, save the patient names.
- Using the **t-test** function, is there a significant difference between creat_conc of people < 50 and people > 50?

Practice