6 Writing functions

(AST230) R for Data Science

Md Rasel Biswas

Writing own R functions

- Every R function has three parts:
  - name
  - a body of code
  - a set of arguments
- The basic format of an R function

```
my_fun <- function(arguments) {

  "body of code"

}
```

- Functions are just another R object
- Comas separate more than one argument

Example

- Write a function that calculates the area of a right triangle

Simulating rolling dice

The following codes simulate an experiment “rolling a dice”

```r
sample(x = 1:6, size = 1)
```

```
[1] 3
```

```r
sample(x = 1:6, size = 1)
```

```
[1] 4
```

Write a function `roll()` for the simulation

```r
roll <- function() {
   dice <- 1:6
   out <- sample(x = dice, size = 1)
   return(out)
}
```

- `roll()` has no arguments, and it will return a number between 1 to 6

Simulating rolling dice

- To use the function `roll()`, first run the written R function codes in the R console. Then run `roll()`

```
# Start rolling dice
# roll 1
roll()
```

```
[1] 1
```

```
# roll 2
roll()
```

```
[1] 2
```

```
# See the function codes
roll
```

```
function() {
   dice <- 1:6
   out <- sample(x = dice, size = 1)
   return(out)
}
<bytecode: 0x7fb187b33b60>
```

Functions with arguments

The number of heads in 10 fair coin toss

```
# Toss a fair coin 10 times
s10 <- sample(x = c(0,1), size = 10, replace = TRUE)
sum(s10)
```

```
[1] 5
```

The number of heads in 20 fair coin toss

```
# Toss a fair coin 20 times
s20 <- sample(x = c(0,1), size = 20, replace = TRUE)
sum(s20)
```

```
[1] 12
```

The number of heads in 100 fair coin toss

```
# Toss a fair coin 100 times
s100 <- sample(x = c(0,1), size = 100, replace = TRUE)
sum(s100)
```

```
[1] 67
```

Functions with arguments

- The `size` argument of `sample()` is changing for different number of coin toss
- Can you write a function in R that takes how many times to toss a fair coin and prints us how many times it lands on heads?

Functions with arguments

```r
# Creating the function
toss_fair_coins <- function(times) {
  out <- sample(x = 0:1, size = times, replace = T)
  sum(out)
}
```

Tossing coins 200/500 times

```r
toss_fair_coins(times = 200)
```

```
[1] 107
```

```r
toss_fair_coins(times = 500)
```

```
[1] 241
```

What will happen if you run the function without specifying the argument

```r
toss_fair_coins()
```

Functions with arguments

```r
# Creating R function with default argument values
toss_fair_coins2 <- function(times = 100) {
  out <- sample(x = 0:1, size = times, replace = T)
  sum(out)
}
```

- Default value of an argument can be specified
- E.g., `times = 100` will be considered if `times` is not supplied!

```r
# No argument supplied
toss_fair_coins2()
```

```
[1] 45
```

```r
# If argument supplied
toss_fair_coins2(times=10000)
```

```
[1] 5003
```

Exercise 6.1

- Write an R function `roll_dice()` that rolls a dice `n` times (where `n` is 1 or more) and returns the sum of the dice rolls.
- Write an R function `toss_bias_coins()` that tosses a biased coin `n` times (where `n` is 1 or more times) and returns the number of heads. The probability of getting heads is 0.70.
- Write an R function that can simulate flipping a fair or biased coin. The probability of getting heads should not always be 0.70, but any value between 0 and 1!

Help for R functions

`help("mean")` or “?mean” to check the detail of `mean()`

```
help("mean") # equivalently ?mean
help.search("mean") #equivalently ??mean
```

- The `help.search()` function searches through the help documentation, code demonstrations and package vignettes and displays the results as clickable links for further exploration
- Another useful function is `apropos()`

```
apropos("mean")
```

```
 [1] ".colMeans"     ".rowMeans"     "colMeans"      "kmeans"
 [5] "mean"          "mean.Date"     "mean.default"  "mean.difftime"
 [9] "mean.POSIXct"  "mean.POSIXlt"  "rowMeans"      "weighted.mean"
```

Help for R functions

Pro Tip

RStudio snippet can save your time. Type `fun` and wait some milliseconds to see suggestions (press **Tab** if they don’t appear). Select (or press **Tab**) on the snippet option to insert it. You can see and customize your snippets from **Tools**>**Global Options**>**Code**.