lab_6_R Functions

Longmei Zhang A17012012

Goal today: explore R functions and begin writing our own functions

Every function in R has three components: - a **Name**, we pick this - one or more **input arguments** - the **body**, where the work actually happens

Start simple: first function, adding some numbers

```r
##the default value for y is 1

add <- function(x, y = 1) {
  x + y
}
```

```r
add(10, 1)
```

```
[1] 11
```

Lab Sheet Work

> Q1. Write a function grade() to determine an overall grade from a vector of student homework assignment scores dropping the lowest single score.

```r
# Example input vectors to start with
student1 <- c(100, 100, 100, 100, 100, 100, 100, 90)
student2 <- c(100, NA, 90, 90, 90, 90, 97, 80)
student3 <- c(90, NA, NA, NA, NA, NA, NA, NA)
```

begin by calculating the avergae for students

```r
mean(student1)
```

```
[1] 98.75
```

```r
mean(student2, na.rm = TRUE)
```

```
[1] 91
```

```r
mean(student3, na.rm = TRUE)
```

```
[1] 90
```

Need to drop the lowest score and fix the NA issue. Try the min function. Try to find the location with `which.min()`

```r
min_ind <- which.min(student1)
mean(student1[-min_ind])
```

```
[1] 100
```

```r
x <- student2

##find NAs in vector and make them 0
x[is.na(x)] <- 0

##finds the mean
mean(x[-which.min(x)])
```

```
[1] 91
```

Turn it into a function

```r
grade <- function(x){
  ##find NAs in vector and make them 0
  x[is.na(x)] <- 0

  ##finds the mean
  mean(x[-which.min(x)])
}
```

```r
grade(student1)
```

```
[1] 100
```

```r
grade(student2)
```

```
[1] 91
```

```r
grade(student3)
```

```
[1] 12.85714
```

Load the entire gradebook

```r
gradebook <- read.csv("https://tinyurl.com/gradeinput", row.names=1)

head(gradebook)
```

```
          hw1 hw2 hw3 hw4 hw5
student-1 100  73 100  88  79
student-2  85  64  78  89  78
student-3  83  69  77 100  77
student-4  88  NA  73 100  76
student-5  88 100  75  86  79
student-6  89  78 100  89  77
```

Use the `apply()` function on the `gradebook` dataset. Need to decide apply the function over row(1) or column(2)

```r
ans <- apply(gradebook, 1, grade)
```

> Q2. Using your grade() function and the supplied gradebook, Who is the top scoring student overall in the gradebook?

```r
which.max(ans)
```

```
student-18
        18
```

```
ans[which.max(ans)]
```

```
student-18
      94.5
```

Q3. From your analysis of the gradebook, which homework was toughest on students (i.e. obtained the lowest scores overall? [2pts]

```
##change the NA in gradebook into 0
masked_gb <- gradebook
masked_gb[is.na(gradebook)] <- 0

##find mean by column
ans2 <- apply(masked_gb, 2, mean)
which.min(ans2)
```

```
hw2
  2
```

We can also modify the `grade()` function to make drop lowest optional

```
grade2 <- function(x, drop.low = T){


  ##find NAs in vector and make them 0
  x[is.na(x)] <- 0

  if(drop.low){

    ##drop the lowest and finds the mean
    out <- mean(x[-which.min(x)])
  }else{
    out <- mean(x)
  }
  return(out)
}

##find the mean without dropping lowest score
apply(gradebook, 2, grade2, drop.low=F)
```

```
  hw1   hw2   hw3   hw4   hw5 
89.00 72.80 80.80 85.15 79.25 
```

> Q4. Optional Extension: From your analysis of the gradebook, which homework was most predictive of overall score (i.e. highest correlation with average grade score)? [1pt]

The function to calculate correlation in R is called `cor()`

```r
stu_mean = apply(gradebook, 1, grade2, drop.low = T)
cor(stu_mean, masked_gb$hw1)
```

```
[1] 0.4250204
```

want to `apply()` the `cor()` function over the `masked_gb` and use the `stu_mean`

```r
cor_hw <- apply(masked_gb, 2, cor, stu_mean)
which.max(cor_hw)
```

```
hw5 
  5 
```