Class06: R Functions

Juliane Kwong

```
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)
```

We can use the `mean()` function to calculate the average for a given student vector.

```
mean(student1)
```

```
[1] 98.75
```

Use argument `na.rm=TRUE` to remove NA values from the student scores.

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

```
[1] 91
```

What about student 3?

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

```
[1] 90
```

We can replace the missed assignment NA values with a score of zero. Use the function `is.na()` to identify NA values.

```
student3
```

```
[1] 90 NA NA NA NA NA NA NA
```

```
is.na(student3)
```

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

```
which(is.na(student3))
```

```
[1] 2 3 4 5 6 7 8
```

Now replace the NA values with a score of zero.

```
student3[is.na(student3)]<-0
student3
```

```
[1] 90  0  0  0  0  0  0  0
```

Use temp object (`x`) so that original objects aren't ruined.

```
x<-student3
x[is.na(x)]<-0
mean(x)
```

```
[1] 11.25
```

Finally, we want to drop the lowest score before calculating the mean. This is equivalent to allowing the student to drop their worst assignment score. Use the `min()` function to identify the lowest score.

```
x<-student1
min(x)
```

```
[1] 90
```

Identify the location of the minimum using the function `which.min()`.

```
which.min(x)
```

```
[1] 8
```

Exclude the minimum score from the average.

```
#The 8th vector contains the lowest score, this will exclude the 8th vector from the list
x[-8]
```

```
[1] 100 100 100 100 100 100 100
```

Use the `-which.min()` function to drop the lowest score from the students scores.

```
x[-which.min(x)]
```

```
[1] 100 100 100 100 100 100 100
```

Find the mean that excludes the lowest score.

```
x<-student1

#Map/Replace NA values to zero
x[is.na(x)]<-0

#Exclude the lowest score and calulate the mean
mean(x[-which.min(x)])
```

```
[1] 100
```

Convert the code snippets into a function called `grade()`.

All functions in R have at least 3 things:

- **Name**, in our case "grade"
- Input **arguments**, student1 etc.
- **Body**, this is our working snippet above.

```
grade<-function(x){
  #Map/Replace NA values to zero
  x[is.na(x)]<-0

  #Exclude the lowest score and calulate the mean
  mean(x[-which.min(x)])
}
```

```
grade(student1)
```

```
[1] 100
```

Can also use RStudio to create a function using code snippets. Select `Code > Extract Function`

```
grade <- function(x) {
  x[is.na(x)]<-0
  mean(x[-which.min(x)])
}
```

```
grade(student2)
```

```
[1] 91
```

Read example class gradebook in CSV format: "https://tinyurl.com/gradeinput"

```
url<-"https://tinyurl.com/gradeinput"
read.csv(url)
```

```
           X hw1 hw2 hw3 hw4 hw5
1   student-1 100  73 100  88  79
2   student-2  85  64  78  89  78
3   student-3  83  69  77 100  77
4   student-4  88  NA  73 100  76
5   student-5  88 100  75  86  79
6   student-6  89  78 100  89  77
7   student-7  89 100  74  87 100
8   student-8  89 100  76  86 100
9   student-9  86 100  77  88  77
10 student-10  89  72  79  NA  76
11 student-11  82  66  78  84 100
12 student-12 100  70  75  92 100
13 student-13  89 100  76 100  80
14 student-14  85 100  77  89  76
15 student-15  85  65  76  89  NA
16 student-16  92 100  74  89  77
17 student-17  88  63 100  86  78
```

```
18 student-18  91  NA 100  87 100
19 student-19  91  68  75  86  79
20 student-20  91  68  76  88  76
```

```
hw<-read.csv(url,row.names=1)
```

We can use the `apply()` function to grade all the students in this class with our new `grade()` function.

The `apply()` function allows us to run any function over either the rows or columns of a data.frame.

Format for `apply()`: apply(DATA, MARGIN=1, FUNCTION)

```
apply(hw,1,grade)
```

```
 student-1  student-2  student-3  student-4  student-5  student-6  student-7
     91.75      82.50      84.25      84.25      88.25      89.00      94.00
 student-8  student-9 student-10 student-11 student-12 student-13 student-14
     93.75      87.75      79.00      86.00      91.75      92.25      87.75
student-15 student-16 student-17 student-18 student-19 student-20
     78.75      89.50      88.00      94.50      82.75      82.75
```

```
#Assign temp object to apply function
ans <- apply(hw,1,grade)
ans
```

```
 student-1  student-2  student-3  student-4  student-5  student-6  student-7
     91.75      82.50      84.25      84.25      88.25      89.00      94.00
 student-8  student-9 student-10 student-11 student-12 student-13 student-14
     93.75      87.75      79.00      86.00      91.75      92.25      87.75
student-15 student-16 student-17 student-18 student-19 student-20
     78.75      89.50      88.00      94.50      82.75      82.75
```

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

Use the function `which.max()` to determine the highest score and top scoring student.

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

```
student-18
        18
```

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

```
apply(hw,2,mean, na.rm=TRUE)
```

```
     hw1      hw2      hw3      hw4      hw5
89.00000 80.88889 80.80000 89.63158 83.42105
```

```
ave.scores<-apply(hw,2,mean, na.rm=TRUE)
which.min(ave.scores)
```

```
hw3
  3
```

```
tot.scores<-apply(hw,2,sum, na.rm=TRUE)
which.min(tot.scores)
```

```
hw2
  2
```

```
tot.scores
```

```
 hw1  hw2  hw3  hw4  hw5
1780 1456 1616 1703 1585
```

```
ave.scores
```

```
     hw1      hw2      hw3      hw4      hw5
89.00000 80.88889 80.80000 89.63158 83.42105
```

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]

```
  hw$hw1
```

```
 [1] 100  85  83  88  88  89  89  89  86  89  82 100  89  85  85  92  88  91  91
[20]  91
```

```
  ans
```

```
 student-1  student-2  student-3  student-4  student-5  student-6  student-7
     91.75      82.50      84.25      84.25      88.25      89.00      94.00
 student-8  student-9 student-10 student-11 student-12 student-13 student-14
     93.75      87.75      79.00      86.00      91.75      92.25      87.75
student-15 student-16 student-17 student-18 student-19 student-20
     78.75      89.50      88.00      94.50      82.75      82.75
```

```
  cor(hw$hw1, ans)
```

```
[1] 0.4250204
```

If I try on hw2 I get NA as there are missing homeworks (i.e. NA values)

```
  hw$hw2
```

```
 [1]  73  64  69  NA 100  78 100 100 100  72  66  70 100 100  65 100  63  NA  68
[20]  68
```

I will mask all NA values to zero.

```
  mask<-hw
  mask[is.na(mask)]<-0
```

Now we can find correlation.

```
  cor(mask$hw5, ans)
```

```
[1] 0.6325982
```

We can use the `apply()` function on the hw columns (i.e. individual homeworks) and pass it the overall scores for the class (in my `ans` object as an extra argument).

```r
apply(mask,2,cor,y=ans)
```

```
      hw1       hw2       hw3       hw4       hw5
0.4250204 0.1767780 0.3042561 0.3810884 0.6325982
```