Class 6 R Functions

Jada Ruiz

Example Student Grades:

```r
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)
```

> Q1. Write a function grade() to determine an overall grade from a vector of student homework assignment scores dropping the lowest single score. If a student misses a homework (i.e. has an NA value) this can be used as a score to be potentially dropped.

Start by using the mean.

```r
mean(student1)
```

```
[1] 98.75
```

Find the minimum value of the student's grade because that is what we plan to drop.

```r
min(student1)
```

```
[1] 90
```

Different type of minimum function tell you what position the minimum value is in.

```r
which.min(student1)
```

```
[1] 8
```

```
student1
```

```
[1] 100 100 100 100 100 100 100  90
```

Get a vector without the minimum value either by hard coding, which wont work for all students, or by using another method the minus index trick.

```
student1[1:7]
```

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

```
student1[-8]
```

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

So I will combine the output of ‘which_min()’ with the minus index trick to get the student’s score without the lowest value.

```
mean(student1[-which.min(student1)])
```

```
[1] 100
```

This will not work for student 2 specifically because student 2 has NA values which student 1 doesn’t have so we didn’t consider that while writing the code for student 1.

```
mean(student2[-which.min(student2)])
```

```
[1] NA
```

There is a ‘na.rm=FALSE’ which is by default, putting it equal to ‘TRUE’ will allow the code to actually work.

```
mean(student2[-which.min(student2)], na.rm=TRUE)
```

```
[1] 92.83333
```

```r
student2
```

```
[1] 100  NA  90  90  90  90  97  80
```

We see this doesn't exactly work for student 3 though because they have a lot of missing grades, more than one NA. We must replace all of the NA (missing values) values with zero.

```r
student3[is.na(student3)] <- 0
mean(student3[-which.min(student3)])
```

```
[1] 12.85714
```

```r
student2[is.na(student2)] <- 0
mean(student2[-which.min(student2)])
```

```
[1] 91
```

Now turn into a function:

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

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

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

Heres the gradebook:

```r
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
```

Now time to use the 'apply()' function.

```
results <- apply(gradebook, 1, grade)
results
```

```
 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 
```

Now time to find the highest scoring student:

```
which.max(apply(gradebook, 1, grade))
```

```
student-18 
        18 
```

```
max(apply(gradebook, 1, grade))
```

```
[1] 94.5
```

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

```
which.min(apply(gradebook, 2, sum, na.rm=TRUE))
```

```
hw2 
  2 
```

> 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)?

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

cor(mask$hw5, results)
```

```
[1] 0.6325982
```

Or use apply:

```
apply(mask, 2, cor, y=results)
```

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