class 06

Aaron Liu (PID:A13908620)

##All About Functions: 3 essential things in function: 1. Name 2. Argument 3. The body

```
# 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)

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

```
[1] 91
```

```
?mean
```

Drop the lowest score

```
min(student1)
```

```
[1] 90
```

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

```
[1] 100
```

Modify it for student 2

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

```
[1] 92.83333
```

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

```
[1] 92.83333
```

Modify it for student 3, mask NA for 0

```
#Mask NA for 0
student3[is.na(student3)]<-0
#Drop the min and find the average.
mean(student3[-which.min(student3)], na.rm=TRUE)
```

```
[1] 12.85714
```

##Grade Function

```
grade<-function(x, na.rm=TRUE){
  #The body of the function
  x[is.na(x)]<-0
  return(mean(x[-which.min(x)], na.rm=na.rm))
}
```

Test

```
grade(student3)
```

```
[1] 12.85714
```

1 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. Your final function should be adquately explained with code comments and be able to work on an example class gradebook such as this one in CSV format: "https://tinyurl.com/gradeinput" [3pts]

```
gradebook<-read.csv("https://tinyurl.com/gradeinput", row.names=1)
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
student-7   89 100  74  87 100
student-8   89 100  76  86 100
student-9   86 100  77  88  77
student-10  89  72  79  NA  76
student-11  82  66  78  84 100
student-12 100  70  75  92 100
student-13  89 100  76 100  80
student-14  85 100  77  89  76
student-15  85  65  76  89  NA
student-16  92 100  74  89  77
student-17  88  63 100  86  78
student-18  91  NA 100  87 100
student-19  91  68  75  86  79
student-20  91  68  76  88  76
```

```
?apply
```

Using apply function

```
apply(gradebook, 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
```

Q2 Using your grade() function and the supplied gradebook, Who is the top scoring student overall in the gradebook? [3pts]

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

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

```
hw3
  3
```

mask the NA values to 0

```
mask<-gradebook
mask[is.na(mask)]<-0
which.min(apply(mask,2,mean))
```

```
hw2
  2
```

```
which.min(apply(mask,2,sum))
```

```
hw2
  2
```

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]

```
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
```

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

```
[1] 0.6325982
```

```r
#Now apply it every homework.
which.max(apply(mask, 2, cor, y=ans))
```

```
hw5
  5
```