Class 06

BIMM 143 Gen Dantay

```
# 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)
```

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

```
mean(student1)
```

```
[1] 98.75
```

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

```
[1] 91
```

We used `na.rm=TRUE` argument to remove NA values before calculating the mean. > what about student 3?

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

```
[1] 90
```

There are too many missed homeworks, and it only includes the homework that they did do. It also isn't fair for the other students. > to fix this we could do:

We can replace the missed assignment NA values with a score of zero. First I need to find where the NA values are.

```
student3
```

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

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

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

```
student3[is.na(student3)]
```

```
[1] 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
```

I can now make these values be anything I want

```
student3
```

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

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

It is time to work with new temp object (that I will call `x`) so I don't mess up my original projects.

```
x<- student3
x
```

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

```r
x[is.na(x)]
```

```
numeric(0)
```

```r
x
```

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

```r
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: I can use the minus sign together with `which.in()` to exclude the lowest value:

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

```
[1] 2
```

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

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

Now I need to put this all back together to make our working snippet.

```r
x<- student3
# map/replace NA values to zero
x[is.na(x)] <- 0

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

```
[1] 12.85714
```

Cool! This is my working snippet that I can turn 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.

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

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

Can I use this function now?

```r
grade(student1)
```

```
[1] 100
```

Read a gradebook from online:

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

```
           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
```

We can use the `apply()` function to grade all the students in this class with our new `grade()` function. The `apply()` functions allows us to run any function over the rows or columns of a data frame. let's see how it works:

```
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? [3pts]

```
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]

```
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 
```

```r
tot.scores
```

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

```r
ave.scores
```

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

Therefore, homework 2 was the toughest homework with the lowest scores overall.

> 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]

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

```
[1] 0.4250204
```

```r
cor(hw$hw3, ans)
```

```
[1] 0.3042561
```

If I try on Hw2 I get NA as there are missing homeworks (i.e. NA values)

```r
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.

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

```
           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   0  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   0  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   0
student-16  92 100  74  89  77
student-17  88  63 100  86  78
student-18  91   0 100  87 100
student-19  91  68  75  86  79
student-20  91  68  76  88  76
```

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

```
[1] 0.6325982
```

We can use the `apply()` function here on the columns of the (i.e. the individual homeworks) and pass it to 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 
```

therefore, the answer is hw2.