

Class06

- A **name**
- A **body** (where work actually happens) You can add options to executable code like this

```
#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 start by using the `mean()` function to calculate an average.

```
mean(student1)
```

```
[1] 98.75
```

I found the `min()` function to find the minimum value in a vector

```
min(student1)
```

```
[1] 90
```

Looking at the “See Also” section of the `min()` help page I found out about `which.min()`

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

```
[1] 8
```

So I will combine the output of `which.min()` with the minus index trick to get the student scores without the lowest value

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

```
[1] 100
```

Hmm... for student2 this gives NA

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

```
[1] NA
```

I see there is an `na.rm=FALSE` by default argument to the `mean()` function. Will this help us?

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

```
[1] 92.83333
```

Trying out `is.na` to replace NA values with 0.

```
student2[is.na(student2)] <- 0  
student2
```

```
[1] 100  0  90  90  90  90  97  80
```

Now calculating mean with NA being 0.

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

```
[1] 91
```

Doing the same for student3

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

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

```
mean(student3[-which.min(student3)])
```

```
[1] 12.85714
```

Creating a function

```
x <- student1  
x[ is.na(x)] <- 0  
mean( x[ -which.min(x)])
```

```
[1] 100
```

I now have a working snippet of code that I have simplified to work with any student x

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

```
[1] 100
```

Creating function grade

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

```
grade(student1)
```

```
[1] 100
```

New Student Gradebook

```
GradeBook <- read.csv("https://bioboot.github.io/bgg213_S19/class-material/student_homeworks/gradebook.csv")  
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

Attempting to use `grade()` on new dataset

```
#grade(GradeBook)
```

Now trying to use `apply()`

```
#1 because we are doing rows. It would be 2 if columns.
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	

Q2 Finding highest scoring student

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

```
student-18
94.5
```

Q3 Finding hardest homework on students

```
#First setting all NA to 0
```

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

```
#Next using apply to find medians of columns since columns represent hw
#using 2 since it represents columns
HardestHWMedian <- apply(GradeBook, 2, median)
```

```
HardestHWMedian[which.min(HardestHWMedian)]
```

```
hw2  
71
```

```
#using mean to see if theres a notable difference  
HardestHWMean <- apply(GradeBook, 2, mean)  
HardestHWMean[which.min(HardestHWMean)]
```

```
hw2  
72.8
```

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, Results)
```

```
      [,1]  
hw1 0.4250204  
hw2 0.1767780  
hw3 0.3042561  
hw4 0.3810884  
hw5 0.6325982
```

```
apply(Mask, 2, cor, Results)
```

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