

# Class 06: R Function

Emily Rodriguez

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 adequately 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]

```
# 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
```

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

```
[1] 90
```

We can replace the missed assignment NA values with a score of zero. - How do I do this?

We can use the `is.na()` function to help perhaps?

```
student2
```

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

```
is.na(student2)
```

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

```
student2[ is.na(student2)]
```

```
[1] NA
```

```
is.na(student2)
```

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

```
which(is.na(student2))
```

```
[1] 2
```

It is time to work with new temp objects (that I will call `x`) so I don't screw up my original objects

```
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.

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

```
x <- student1
x
```

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

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

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

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

```
x <- student3
x
```

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

```
# 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.

```
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?

```
grade(student1)
```

```
[1] 100
```

Read a gradebook from online:

```
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.

They `apply()` functions allows us to run any function over either 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
```

**student-18** is the top scoring student overall in the gradebook.

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

```
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
```

**Homework 2** was the toughest for the students.

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
```

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

```
[1] 0.3042561
```

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
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

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

```
[1] 0.6325982
```

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

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

hw1	hw2	hw3	hw4	hw5
0.4250204	0.1767780	0.3042561	0.3810884	0.6325982

**Homework 5** was the most predictive of overall score.