

Class6

Laura Biggs

Quarto

Q1

```
# Q1: Write a function grade() to determine an overall grade from a vector of student home
```

```
#Practice student inputs
```

```
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)
```

```
mean(student1)
```

```
[1] 98.75
```

```
#Both return NA b/c of NA in list, need to somehow exclude
```

```
mean(student2)
```

```
[1] NA
```

```
mean(student3)
```

```
[1] NA
```

How to exclude the lowest value?

```
#Tells us what position in a list has the lowest value
#which.min()

which.min(student1)
```

```
[1] 8
```

```
#If we want to remove the lowest value use the minus sign to extract it from the individual
#Removes the lowest score, 90, from student 1
student1[-which.min(student1)]
```

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

```
#Can we find the mean from this which.min() operation?
mean(student1[-which.min(student1)])
```

```
[1] 100
```

Test cope snippet on other student examples

```
#Student 2
#Doesnt work b/c of NA in the list, try excluding with na.rm=TRUE
mean(student2[-which.min(student2)])
```

```
[1] NA
```

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

```
[1] 91
```

```
#Student 3
#na.rm distorts the grade and is inadequate in calculating the total grade
mean(student3[-which.min(student3)])
```

```
[1] NA
```

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

```
[1] 90
```

Identify and replace all NA's with a value of 0

```
#Find all NA values w/the element is.na; returns FALSE if not and TRUE if NA
is.na(student2)
```

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

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

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

```
#OR identify NA position w/which() function
which(is.na(student2))
```

```
[1] 2
```

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

```
[1] 2 3 4 5 6 7 8
```

```
#Replace NA with zero
student2[is.na(student2)] <- 0
mean(student2)
```

```
[1] 79.625
```

```
#Produces the expected low grade from the missing assignments
student3[is.na(student3)] <- 0
mean(student3)
```

```
[1] 11.25
```

We must drop the lowest score

```
#Use the previous code that excluded the lowest score using the which.min() operator
student2[is.na(student2)] <- 0
student2
```

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

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

```
[1] 91
```

```
#Test on student 3; it WORKS!
student3[is.na(student3)] <- 0
student3
```

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

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

```
[1] 12.85714
```

Function time!

```
#Create grade function w/mean variable, and return of mean variable.
#na.rm = TRUE excludes NA values from function calculations
#Use which.min function to exclude the lowest value
grade <- function(x) {
  x[is.na(x)] <- 0
  mean(x[-which.min(x)])
}
```

Test function with student example input

```
grade(student1)
```

```
[1] 100
```

```
grade(student2)
```

```
[1] 91
```

```
grade(student3)
```

```
[1] 12.85714
```

Code documentation

```
#' Calculation of average score from vector of student homework grades,  
# dropping the lowest score. Missing values are treated as 0.  
#'  
#' @param x A numeric vector of homework scores  
#'  
#' @return An average score  
#' @export  
#'  
#' @examples  
#' student <- c(100, NA, 90, 97)  
#' grade(student)  
#'  
  
grade <- function(x) {  
  # Treat input NA values as zero  
  x[is.na(x)] <- 0  
  # Excluding the lowest score with -which.min()  
  mean(x[-which.min(x)])  
}
```

Use the function on the whole class data

```
#Create variable and load in gradeinput file  
#Covert students from X column to rows  
url <- 'https://tinyurl.com/gradeinput'  
gradebook <- read.csv(url, 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

```

```

#Test function with apply()
#Inputs to apply is the dataframe/x values, the row or column choice, and the name of the
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

```

# Q2: Using your grade() function and the supplied gradebook, Who is the top scoring student?

#Find the answer with code using which.max()
which.max(apply(gradebook, 1, grade))

```

```
student-18
      18
```

Q3

```
# Q3: From your analysis of the gradebook, which homework was toughest on students (i.e. o
```

```
#Mean calculation
avg_hw <- apply(gradebook, 2, mean, na.rm=TRUE)
avg_hw
```

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

```
which.min(avg_hw)
```

```
hw3
3
```

```
#Median calculation
# Use apply() to calculate the grade function on each column which represents the individu
# Row = 1 and column = 2
# na.rm=TRUE again to exclude the NA values
med_hw <- apply(gradebook, 2, median, na.rm=TRUE)
med_hw
```

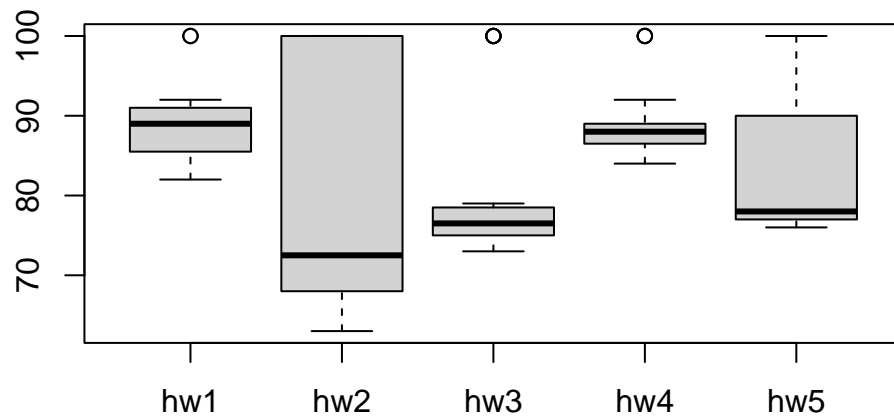
```
      hw1  hw2  hw3  hw4  hw5
89.0 72.5 76.5 88.0 78.0
```

```
which.min(med_hw)
```

```
hw2
2
```

Is mean or median more representative of the lowest scoring homework? Median

```
boxplot(gradebook)
```



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]

Are the final results of the students correlated with a particular homework?

```
final_results <- apply(gradebook, 1, grade)
cor(final_results, gradebook$hw5)
```

[1] NA

```
#Correlation cannot work with NA values; mask values with is.na
masked_gradebook <- gradebook
masked_gradebook[is.na(masked_gradebook)] <- 0
masked_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   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

```

Correlation w/o NA values

```
cor(final_results, masked_gradebook$hw5)
```

```
[1] 0.6325982
```

Apply correlation across the entire masked gradebook

```
apply(masked_gradebook, 2, cor, x=final_results)
```

```

      hw1      hw2      hw3      hw4      hw5
0.4250204 0.1767780 0.3042561 0.3810884 0.6325982

```

Q5. Make sure you save your Quarto document and can click the “Render” (or Rmark-down”Knit”) button to generate a PDF format report without errors. Finally, submit your PDF to gradescope. [1pt]