Class 06: R Functions

Alex Cagle

2023-04-25

Lab 6

```
# Loading data
gradebook <- read.csv(url('https://tinyurl.com/gradeinput'), row.names = 1)

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
```

```
# 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)
```

```
# mean(student1)

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

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

```
# x <- student1
# x
# which.min(x)
```

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]

```r
# Question 1:
# Function to calculate average grade with lowest score dropped
grade <- function(x) {
  x[is.na(x)] <- 0

  print(x)

  lowest_index <- which.min(x)

  x <- x[-lowest_index]

  mean(x)

  # avg <- mean(new_gradebook)

  # return(avg)
}
```

```r
# Testing grade() with example vectors
# grade(student1)

# grade(student2)

# grade(student3)
```

```r
# Question 1:
# Applying grade() function to gradebook
avgs <- apply(gradebook, 1, grade)
```

```
## hw1 hw2 hw3 hw4 hw5
## 100  73 100  88  79
## hw1 hw2 hw3 hw4 hw5
##  85  64  78  89  78
## hw1 hw2 hw3 hw4 hw5
##  83  69  77 100  77
## hw1 hw2 hw3 hw4 hw5
##  88   0  73 100  76
## hw1 hw2 hw3 hw4 hw5
##  88 100  75  86  79
## hw1 hw2 hw3 hw4 hw5
##  89  78 100  89  77
## hw1 hw2 hw3 hw4 hw5
##  89 100  74  87 100
## hw1 hw2 hw3 hw4 hw5
##  89 100  76  86 100
## hw1 hw2 hw3 hw4 hw5
##  86 100  77  88  77
## hw1 hw2 hw3 hw4 hw5
##  89  72  79   0  76
## hw1 hw2 hw3 hw4 hw5
##  82  66  78  84 100
## hw1 hw2 hw3 hw4 hw5
## 100  70  75  92 100
```

```
## hw1 hw2 hw3 hw4 hw5
##  89 100  76 100  80
## hw1 hw2 hw3 hw4 hw5
##  85 100  77  89  76
## hw1 hw2 hw3 hw4 hw5
##  85  65  76  89   0
## hw1 hw2 hw3 hw4 hw5
##  92 100  74  89  77
## hw1 hw2 hw3 hw4 hw5
##  88  63 100  86  78
## hw1 hw2 hw3 hw4 hw5
##  91   0 100  87 100
## hw1 hw2 hw3 hw4 hw5
##  91  68  75  86  79
## hw1 hw2 hw3 hw4 hw5
##  91  68  76  88  76
```

```
# Displaying the average homework grade for each student
avg_df <- data.frame(avgs)
avg_df
```

```
##             avgs
## student-1  91.75
## student-2  82.50
## student-3  84.25
## student-4  84.25
## student-5  88.25
## student-6  89.00
## student-7  94.00
## student-8  93.75
## student-9  87.75
## student-10 79.00
## student-11 86.00
## student-12 91.75
## student-13 92.25
## student-14 87.75
## student-15 78.75
## student-16 89.50
## student-17 88.00
## student-18 94.50
## student-19 82.75
## student-20 82.75
```

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

```
# Question 2:
# Highest scoring student
print("The highest score is:")
```

```
## [1] "The highest score is:"
```

```
highest_score <- avg_df[which.max(avg_df[,1]),]
highest_score
```

```
## [1] 94.5
```

```
print("The highest scoring student is:")
```

```
## [1] "The highest scoring student is:"
```

```
rownames(avg_df)[avg_df[, 1] == highest_score]
```

```
## [1] "student-18"
```

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

```
# Question 3:
# Toughest homework
avg_hw_score <- apply(gradebook, 2, sum, na.rm=T)
avg_hw_score
```

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

```
print("The toughest homework is:")
```

```
## [1] "The toughest homework is:"
```

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

```
##  hw2
## 1456
```

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 <- gradebook
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
```

```
hw$hw1
```

```
##  [1] 100  85  83  88  88  89  89  89  86  89  82 100  89  85  85  92  88  91  91
## [20]  91
```

```
cor(hw$hw1, avgs)
```

```
## [1] 0.4250204
```

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

```
## [1] 0.3042561
```

```
# Masking 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
```

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

```
## [1] 0.6325982
```

```r
apply(mask, 2, cor, y=avgs)
```

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