Class 6

Max Strul

10/14/2022

Table of contents


```
#fname <- function(arg1, arg2) {paste(arg1,arg2)}
# 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)
# gradebook_1 <- data.frame(student_1=student1,student_2=student2,student_3=student3)
url <- "https://tinyurl.com/gradeinput"
example_gradebook_1 <- read.csv(url,row.names = 1)
```

Writing the grade function

```
#grade() function --> Needs to
#first sort all numbers
#pop or get rid of lowest value (based on how it sorts)
grade <- function(gradebook,studentnumber){
  gradebook[is.na(gradebook)]<-0
  student <- gradebook[studentnumber,]
  single_grade <- student[,-which.min(student)]
  #this is a vector of a student's grade book minus their lowest grade
  gradebook_size <- length(single_grade)
  reported_grade <- sum(single_grade)/gradebook_size
}
```

Creating a gradebook

```
class_size <- nrow(example_gradebook_1)
final_grades <- data.frame(student=1:class_size,final_grade=1)
for(i in 1:class_size){
  final_grades[i,2]<-(grade(example_gradebook_1,i))
}
print(final_grades)
```

```
   student final_grade
1        1       91.75
2        2       82.50
3        3       84.25
4        4       84.25
5        5       88.25
6        6       89.00
7        7       94.00
8        8       93.75
9        9       87.75
10      10       79.00
11      11       86.00
12      12       91.75
13      13       92.25
14      14       87.75
15      15       78.75
```

```
16      16       89.50
17      17       88.00
18      18       94.50
19      19       82.75
20      20       82.75
```

Apply function

`apply()` function is a useful function that can be used to create a gradebook via the restraints of a given matrix.

```
#print(apply(example_gradebook_1,1,grade))

#This does not work for the given function grade

#because it was made to apply to an entire spreadsheet of data.
```

Which student scored the highest?

```
y = final_grades[,2]
names(y) = rownames(example_gradebook_1)
sort(y)
```

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

```
#Here we see student #18 scored the highest.
#This function has an issue
#if multiple students score the same value then the sort function is better
```

Which homework was toughest on students?

```
#Essentially we want to find the specific homework
#that was the lowest scoring assignment for the most students
#We want to take the minimum position of each student's grade book
#then take the mode of that value (most frequently occurring one)
#I dont want to consider NA as a hard homework, thus I will not include such values.
full_gradebook <- read.csv(url,row.names = 1)
list_of_hardest_homeworks <- data.frame(student=1:class_size,hardest_hw=1)
toughest_homework <- function(gradebook){
  for(i in 1:nrow(gradebook)){
    list_of_hardest_homeworks[i,2] <- which.min(gradebook[i,])
  }
  list_of_homeworks <- sort(list_of_hardest_homeworks[,-1])
}

toughest_homework_array <- toughest_homework(full_gradebook)
print(toughest_homework_array)
```

```
 [1] 2 2 2 2 2 2 2 2 2 2 3 3 3 3 3 3 3 4 5 5
```

Here we see the mode of this data vector is 2

Alternatively:

```
sort(apply(example_gradebook_1,2,sum, na.rm=TRUE))
```

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

Which homework was most predictive of overall score?

```
#Which homework was most predictive of overall score?
#essentially means which homework for each student is closest to their final grade
#then taking the mode of that

#look at the gradebook for each student
#use modulo to find the value with the lowest difference in grade
```

```
prediction <- example_gradebook_1
predictive_hw <- function(gradebook){
                      for(i in 1:nrow(gradebook)){
                          for(x in 1:ncol(gradebook)){
                            prediction[i,x]<-final_grades[i,2]%%gradebook[i,x]
                          }
                      }
  print(prediction)
}

predicted_array <- predictive_hw(example_gradebook_1)
```

```
             hw1   hw2   hw3   hw4   hw5
student-1  91.75 18.75 91.75  3.75 12.75
student-2  82.50 18.50  4.50 82.50  4.50
student-3   1.25 15.25  7.25 84.25  7.25
student-4  84.25    NA 11.25 84.25  8.25
student-5   0.25 88.25 13.25  2.25  9.25
student-6   0.00 11.00 89.00  0.00 12.00
student-7   5.00 94.00 20.00  7.00 94.00
student-8   4.75 93.75 17.75  7.75 93.75
student-9   1.75 87.75 10.75 87.75 10.75
student-10 79.00  7.00  0.00    NA  3.00
student-11  4.00 20.00  8.00  2.00 86.00
student-12 91.75 21.75 16.75 91.75 91.75
student-13  3.25 92.25 16.25 92.25 12.25
student-14  2.75 87.75 10.75 87.75 11.75
student-15 78.75 13.75  2.75 78.75    NA
student-16 89.50 89.50 15.50  0.50 12.50
student-17  0.00 25.00 88.00  2.00 10.00
student-18  3.50    NA 94.50  7.50 94.50
student-19 82.75 14.75  7.75 82.75  3.75
student-20 82.75 14.75  6.75 82.75  6.75
```

```
#The array above tells us that if there is a student
#whos homework grade is the same as any of their final grade
#the modulo would be 0.
#We can compare this with their actual grade
print(example_gradebook_1)
```

```
          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
```

```
#And their final grade
print(final_grades)
```

```
   student final_grade
1        1       91.75
2        2       82.50
3        3       84.25
4        4       84.25
5        5       88.25
6        6       89.00
7        7       94.00
8        8       93.75
9        9       87.75
10      10       79.00
11      11       86.00
12      12       91.75
13      13       92.25
14      14       87.75
15      15       78.75
16      16       89.50
17      17       88.00
```

```
18      18       94.50
19      19       82.75
20      20       82.75
```

```
#For a better output we may want this as a list
#of which homework was most correlated with each student's final grade


#To do this we want to find the position of the lowest value
#within the row of the predicted_array
#apply(predicted_array,1,which.min())

vector_of_lowest_modulo <- data.frame(values=1:nrow(final_grades))

for(i in 1:nrow(final_grades)){
  vector_of_lowest_modulo[i,1] <- which.min(predicted_array[i,])
}
vector_of_lowest_modulo
```

```
   values
1       4
2       3
3       1
4       5
5       1
6       1
7       1
8       1
9       1
10      3
11      4
12      3
13      1
14      1
15      3
16      4
17      1
18      1
19      5
20      3
```

```
sort(vector_of_lowest_modulo$values)
```

```
 [1] 1 1 1 1 1 1 1 1 1 1 3 3 3 3 3 4 4 4 5 5
```

```
#This tells us that homework 1 on average has the highest correlation
#which comparing to the correlation function is incorrect.
```

Alternatively... use `apply()`

```
No_nas <- example_gradebook_1
No_nas[is.na(No_nas)] <- 0
sort(apply(No_nas,2,cor,final_grades[,2]),decreasing=TRUE)
```

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

Here we see homework 2 is the least correlative with grades, and homework 5 is the most correlated with grades.