

class06handsonworksheet

Quarto

Quarto enables you to weave together content and executable code into a finished document. To learn more about Quarto see <https://quarto.org>.

Running Code

When you click the **Render** button a document will be generated that includes both content and the output of embedded code. You can embed code like this:

```
1 + 1
```

```
[1] 2
```

You can add options to executable code like this

```
[1] 4
```

The `echo: false` option disables the printing of code (only output is displayed).

```
# 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)
```

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]

```
mean(student1)
```

```
[1] 98.75
```

```
mean(student2)
```

```
[1] NA
```

```
##skip over all values "NA" from vectors prior to calculating mean  
mean(student2, na.rm=TRUE)
```

```
[1] 91
```

```
#student 3 only completed one assignment, so their average grade is based off one individual
```

```
#replace all "NA" values with 0
```

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

```
#calculate mean of vector with "NA"=0
```

```
mean(student3)
```

```
[1] 11.25
```

```
#find positions in which vector has value "NA"
```

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

```
[1] 2
```

```
#find which positions in student 3 vector are 0
```

```
which(student3==0)
```

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

It is time to work with new temp object (that I will call “x”) so I don’t screw up the original objects

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

```
[1] 11.25
```

finally, we want to drop the lowest scored before calculating the mean. This is equivalent to allowing the student to drop their worst score

```
x<-student1
x
```

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

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

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

We can use the -which function to exclude the lowest value:

```
x<-student2
x
```

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

```
#replace NA values with 0
x[is.na(x)] <- 0
#Exclude the lowest score & calculate mean
mean(x[-which.min(x)])
```

```
[1] 91
```

cool! this is my working snippet that I can turn into a function called 'grade()'

All function in R have 3 things: 1. name, in our case "grade" 2. input arguments, in our case the student's grade vector 3. body, this is our working snippet above

```

grade <- function(x)
{
  #replace NA values with 0
  x[is.na(x)] <- 0
  #Exclude the lowest score & calculate mean
  mean(x[-which.min(x)])
}

```

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

```

?apply()
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
```

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

```
ave.scores<- which.min(apply(hw, 2, mean, na.rm=TRUE))
tot.scores<- which.min(apply(hw, 2, sum, na.rm=TRUE))
ave.scores
```

```
hw3
3
```

```
tot.scores
```

```
hw2
2
```

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]

```
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 assignments

```
hw$hw2
```

```
[1] 73 64 69 NA 100 78 100 100 100 72 66 70 100 100 65 100 63 NA 68  
[20] 68
```

```
cor(hw$hw2, ans)
```

```
[1] NA
```

I will mask all NA values to 0

```
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

We can use the `apply()` function here on the columns of the `hw` and pass it in the overall scores for the class.

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

hw1	hw2	hw3	hw4	hw5
0.4250204	0.1767780	0.3042561	0.3810884	0.6325982