

class06:R functions

Trinity Lee A16639698

##All about functions in R

Functions in R require starting with small defined input vectors and then building up to more complex vectors.

Today in lab we will look at developing a functions for calculating grades of students in class.

#input vectors of students 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)
```

To find out the average of scores we can use the `mean` function

```
mean(student1)
```

```
[1] 98.75
```

Dropping the lowest score should give us an average of 100 The lowest value can be found using the `min` function

```
min(student1)
```

```
[1] 90
```

Using the `which.min` function

```
student1
```

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

```
which.min(student1)
```

```
[1] 8
```

Using the minus syntax trick I can get everything but the element with the min value. The first working snippet of code is created.

```
student1[-which.min(student1)]
```

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

```
mean(student1[-which.min(student1)])
```

```
[1] 100
```

Testing code on student2

```
student2
```

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

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

```
[1] NA
```

Finding the problem - NA input in the `mean()`

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

```
[1] 91
```

looking at student 3

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

```
[1] 90
```

Want to stop working with `student1`,`student2`,`student3` so instead work with an input called `x`.

```
x<-student2
x
```

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

We want to overwrite the NA values with zero - if you miss a homework you score zero on the homework.

AI told us about the `is.na()` function.

```
x
```

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

```
is.na(x)
```

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

Checking the function for all students to turn NA to 0 and drop lowest score when averaged

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

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

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

```
[1] 91
```

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

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

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

```
[1] 12.85714
```

```
x<-student1
# mask NA values to zero
x[is.na(x)]<-0
#Drop lowest score and get the mean
mean(x[-which.min(x)])
```

[1] 100

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]

```
grade<-function(x){
# mask NA values to zero
x[is.na(x)]<-0
#Drop lowest score and get the mean
mean(x[-which.min(x)])}
```

```
grade(student1)
```

[1] 100

Now we need to read the grade book

```
gradebook<-read.csv('https://tinyurl.com/gradeinput',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

```

The `apply()` function can be used to figure out the question

```

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

```

Answer

```

ans<-apply(gradebook,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]

```

which.max(ans)

```

```
student-18
18
```

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

```
mask<-gradebook
mask[is.na(mask)]<-0
ans3<-apply(mask,2,mean)
ans3
```

```
hw1 hw2 hw3 hw4 hw5
89.00 72.80 80.80 85.15 79.25
```

```
which.min(ans3)
```

```
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]

```
ans4<-apply(mask,2,cor,ans)
ans4
```

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

```
which.max(ans4)
```

```
hw5
5
```