

Class6

Sabrina Koldinger(A16368238)

2024-01-25

R Functions

Functions are how we get stuff done. We call functions to do everything useful in R.

One cool thing about R is that it makes writing your own functions simpler.

All functions in R have at least three things:

- A name (that we pick)
- One or more input arguments (the input to our function)
- The body (lines of code that do the work)

```
funname= function(input1, input2){The body with R code}
```

Silly first function to add two numbers:

```
x=5  
y=1  
x+y
```

```
[1] 6
```

```
addme=function(x,y=1){x+y}
```

```
addme(1,1)
```

```
[1] 2
```

```
addme(10,)
```

```
[1] 11
```

Today's Lab

Question 1:

```
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
```

```
mean(student1)
```

```
[1] 98.75
```

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

```
[1] 91
```

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

```
[1] 90
```

```
min(student1)
```

```
[1] 90
```

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

```
[1] 8
```

Found which.min() from min. It is the 8th value in the vector

```
#Find lowest score
student1[which.min(student1)]
```

```
[1] 90
```

```
#Exclude lowest score
student1[-which.min(student1)]
```

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

Put all the parts together

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

```
[1] 100
```

Use a common shortcut and use 'x' as my input

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

```
[1] 100
```

Could replace NA values with 0.

```
y= c(1, 2, NA, 4, 5)
y==NA
```

```
[1] NA NA NA NA NA
```

```
is.na(y)
```

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

how do I remove the NA element from the vector

```
!c(F,F,F)
```

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

```
y[is.na(y)]
```

```
[1] NA
```

```
y[!is.na(y)]
```

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

actually make NA 0

```
y[is.na(y)]=0
```

Putting this together

```
x=student2
#Change NA to 0
x[is.na(x)]=0
#Find and remove min value and get mean
mean(x[-which.min(x)])
```

```
[1] 91
```

```
x=student1
#Change NA to 0
x[is.na(x)]=0
#Find and remove min value and get mean
mean(x[-which.min(x)])
```

```
[1] 100
```

Last step now that I have my woking code snippet is to make my `grade()` function.

```
grade= function(x){x[is.na(x)]=0
  mean(x[-which.min(x)])}
```

```
grade(student1)
```

```
[1] 100
```

```
grade(student2)
```

```
[1] 91
```

```
grade(student3)
```

```
[1] 12.85714
```

Now read gradebook file

```
url= "https://tinyurl.com/gradeinput"
gradebook= read.csv(url, 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

```
mean_grades=apply(gradebook, 1 ,grade)
mean_grades
```

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	

Question 2:

```
max(mean_grades)
```

```
[1] 94.5
```

```
which.max(mean_grades)
```

```
student-18  
18
```

Student 18 has the highest grade.

Question 3:

```
overall_grades_for_assignment=apply(gradebook, 2 , mean, na.rm=T)  
which.min(overall_grades_for_assignment)
```

```
hw3  
3
```

Homework 3 is the most difficult.

Question 4: which homework was most predictive of overall score (i.e. highest correlation with average grade score)

```
#make NAs to zero  
mask= gradebook  
mask[is.na(mask)]=0
```

Use `cor()` function for correlation analysis

```
cor(mask$hw5, mean_grades)
```

```
[1] 0.6325982
```

```
cor(mask$hw3,mean_grades)
```

```
[1] 0.3042561
```

Use `apply()` function to run this analysis over the whole course.

```
apply(mask, 2, cor, mean_grades )
```

	hw1	hw2	hw3	hw4	hw5
	0.4250204	0.1767780	0.3042561	0.3810884	0.6325982