Class6

Erin Li

```
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(student2)
```

```
[1] NA
```

```
mean(student3)
```

```
[1] NA
```

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

```
[1] 100
```

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

```
[1] NA
```

```
mean(student3[-which.min(student3)])
```

```
[1] NA
```

```
#which.min(student1) gives position 8
```

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

```
[1] 91
```

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

```
[1] 90
```

I want to stop working with ‘student1’, ‘student2’ etc. and typing it out every time so let instead work with an input called ‘x’

```
#Mask NA values to zero
x <- student2
x [is.na(x)]<-0
#Drop lowest score and get the mean
result <- mean(x[-which.min(x)])
result
```

```
[1] 91
```

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]

```
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)])
}
```

Use this function:

```
grade(student1)
```

```
[1] 100
```

```
grade(student2)
```

```
[1] 91
```

```
grade(student3)
```

```
[1] 12.85714
```

```
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
```

Q2.Usingyourgrade()functionandthesuppliedgradebook,Whoisthetopscoringstudent overallinthegradebook?[3pts] Top scoring student is 18, scored 94.5.

```
ans = apply(X= gradebook, MARGIN=1, FUN=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 
```

```
#which.max(list)
#ans[which.max(list)]
#Top scoring student is 18, scored 94.5.
```

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

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

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

```
# Question 2 is the toughest
```

Q4.OptionalExtension:Fromyouranalysisofthegradebook,whichhomeworkwasmost predictiveofoverallscore(i.e.highestcorrelationwithaveragegradescore)?[1pt] hw 5 is the most representative

```
mask <- gradebook
mask[is.na(mask)]<-0
```

```
hw.ave <- apply(mask, 2, mean)
correlate <- apply(mask, 2, cor, y = ans)
correlate
```

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

```
which.max(correlate)
```

```
hw5
  5
```

```
#hw 5 is the most representative
```