

# class06

Ziyuan\_Han

10/15/2021

## quick R Markdown intro

We can write text of course just like any file, we can **style the text to be bold** or *italic*

Do: -this -that -and another thing

this is new text (two whitespaces to create a new line)

and this is a new line

— we can include some code:

```
# this is a comment
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)
```

short cut for adding code chunk option + command + I (for R code)

**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] First, find the lowest score using **`min()`**, and the **`which.min()`** function to find where it is (ie. the position in the vector)

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

```
## [1] 8
```

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

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

Now I can call the **`mean()`** function to get the average.

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

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

```
student2
```

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

```
mean(student2)
```

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

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

```
## [1] 91
```

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

```
## [1] 2
```

```
!is.na(student2) #try to reverse the TRUE and FALSE
```

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

```
student2[which(is.na(student2))] = 0  
student3[which(is.na(student3))] = 0  
mean(student3[-which.min(student3)])
```

```
## [1] 12.85714
```

```
student4 <-c(100,NA,90,"90",90,90,97,80)  
student4 <-as.numeric(student4)  
student4[is.na(student4)]=0  
mean(student4[-which.min(student4)])
```

```
## [1] 91
```

```
#write a function
```

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

```
grade(student1)
```

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

```
gradebook<-"https://tinyurl.com/gradeinput"
scores<-read.csv(gradebook,row.names = 1)
scores
```

```
##           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
```

```
avg_perstudent <- apply(scores,1,grade)
```

```
##Q2## who is the toppest student: student 18
```

```
avg_perstudent[which.max(avg_perstudent)]
```

```
## student-18
##           94.5
```

```
##From your analysis of the gradebook, which homework was toughest on students (i.e. obtained the lowest scores overall?## : Homework 2
```

```
mask <- scores
mask[is.na(mask)] = 0
avg_perhw <- apply(mask,2,mean)
avg_perhw[which.min(avg_perhw)]
```

```
## hw2
## 72.8
```

```
##Optional Extension: From your analysis of the gradebook, which homework was most predictive of overall score (i.e. highest correlation with average grade score)? ## Homework5
```

```
ans <- apply(scores,1,grade)
cor(mask$hw1,ans)
```

```
## [1] 0.4250204
```

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

```
## [1] 0.176778
```

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

```
## [1] 0.3042561
```

```
cor(mask$hw4,ans)
```

```
## [1] 0.3810884
```

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

```
## [1] 0.6325982
```

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

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

```
boxplot(scores)
```

