class 6: R function

Jose Chacon (PID A59010515)

10/15/2021

Quick Rmarkdown intro

We can write text of course just like any file. We can **style text to be bold** or *italic*:

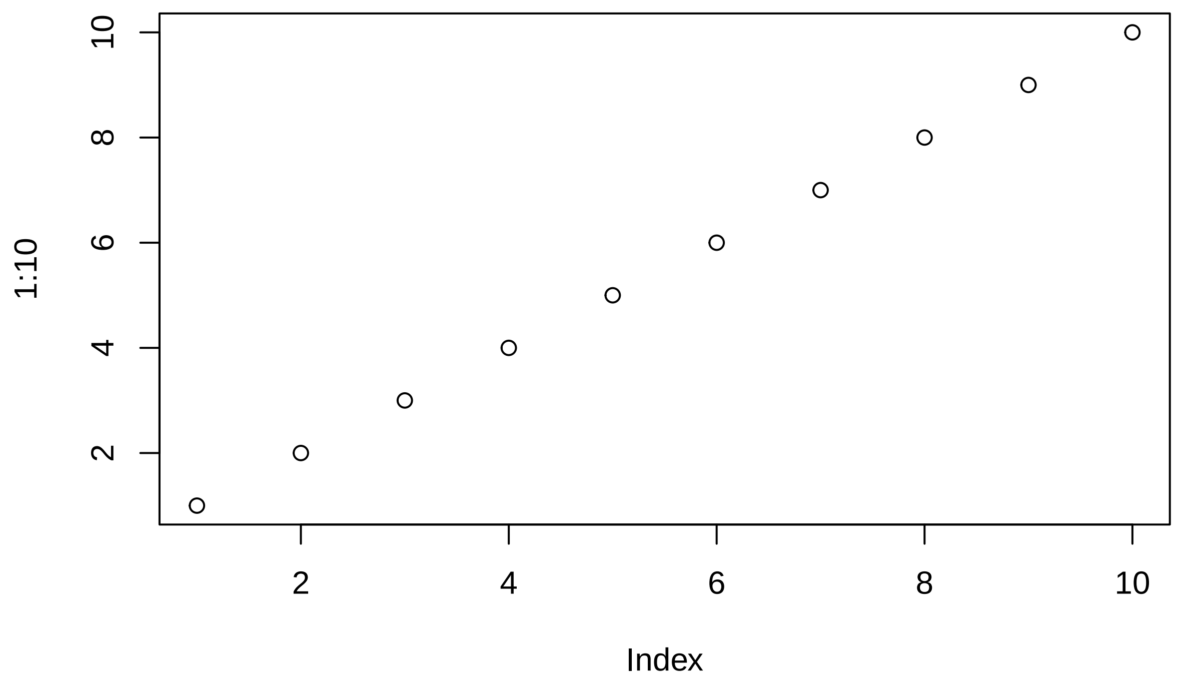
Do:

-this -and that -and another thing

THis is more text and this is a new line

We can include some code:

```
plot(1:10)
```



Time to write a function

> **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 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]

```
# 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)
```

Find lowest score, to exclude from average. Lowest score can be found with min() and position can be found with which.min()

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

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

add '-' in front to exclude data in vector, in this case 'which.min' , excluding the lowest score from the vector.

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

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

Now **mean()** can function can be used to get average

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

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

Does it work for student2?

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

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

```
which.min(student2)
```

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

```
mean(student2)
```

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

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

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

```
mean(student3, )
```

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

Replace NA value with 0

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

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

na() makes it a binary T or F at each position

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

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

! turns it upside down, reverses it

```
!is.na(student2)
```

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

replace NA values with zero and conserve original

```
student.prime <- student2
student.prime[is.na(student.prime)] = 0
student.prime
```

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

get mean excluding lowest score

```
mean(student.prime[ -which.min(student.prime)])
```

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

student 3?

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

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

```r
mean(student.prime[ -which.min(student.prime)])
```

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

Clear up code

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

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

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

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

New student

```r
student4 <- c(100, NA, 90, "90", 90, 90, 97, 80)
student4
```

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

```r
new_student4<- as.numeric(student4)
new_student4
```

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

```r
x <- new_student4
x[is.na(x)] = 0
x
```

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

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

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

Write function function are composed of 3 things a name, input arg, and a body

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

```
grade(student2)
```

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

Now grade the class

```
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
```

Use the **apply()** fuction to grade students with the **grade()** function >**Q1**

```
ans <- apply(scores, 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 who is top scoring student

```
which.max(ans)
```

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

student 18 scored the highest

Q3. which test was toughest

```
q3 <- apply(scores, 2, mean)
q3
```

```
##  hw1  hw2  hw3  hw4  hw5 
## 89.0   NA 80.8   NA   NA
```

ignore NA values # not great since it means if someone didn't turn it it would be zero

```
apply(scores, 2, mean, na.rm=TRUE)
```

```
##      hw1      hw2      hw3      hw4      hw5 
## 89.00000 80.88889 80.80000 89.63158 83.42105
```

replace/mask NA values to zero

```
mask <- scores
is.na(mask)
```

```
##              hw1   hw2   hw3   hw4   hw5
## student-1  FALSE FALSE FALSE FALSE FALSE
## student-2  FALSE FALSE FALSE FALSE FALSE
## student-3  FALSE FALSE FALSE FALSE FALSE
## student-4  FALSE  TRUE FALSE FALSE FALSE
## student-5  FALSE FALSE FALSE FALSE FALSE
## student-6  FALSE FALSE FALSE FALSE FALSE
## student-7  FALSE FALSE FALSE FALSE FALSE
## student-8  FALSE FALSE FALSE FALSE FALSE
## student-9  FALSE FALSE FALSE FALSE FALSE
## student-10 FALSE FALSE FALSE  TRUE FALSE
## student-11 FALSE FALSE FALSE FALSE FALSE
## student-12 FALSE FALSE FALSE FALSE FALSE
## student-13 FALSE FALSE FALSE FALSE FALSE
## student-14 FALSE FALSE FALSE FALSE FALSE
## student-15 FALSE FALSE FALSE FALSE  TRUE
## student-16 FALSE FALSE FALSE FALSE FALSE
## student-17 FALSE FALSE FALSE FALSE FALSE
## student-18 FALSE  TRUE FALSE FALSE FALSE
## student-19 FALSE FALSE FALSE FALSE FALSE
## student-20 FALSE FALSE FALSE FALSE FALSE
```

```
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
```

Now use apply utilizing the "masked" scores!

```
apply(mask, 2, mean)
```

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

Hardest test was hw2

Q4.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]

Here use the **cor()** function

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

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

```
cor(mask, ans)
```

```
##          [,1]
## hw1 0.4250204
## hw2 0.1767780
## hw3 0.3042561
## hw4 0.3810884
## hw5 0.6325982
```

can call **cor()** for every hw and get a value, use apply to do them all

```
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 
```

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

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

hw5 is most predictive