Class6: R Functions

L.Cruz PID: A59006931

10/15/2021

Quick Rmarkdown Intro

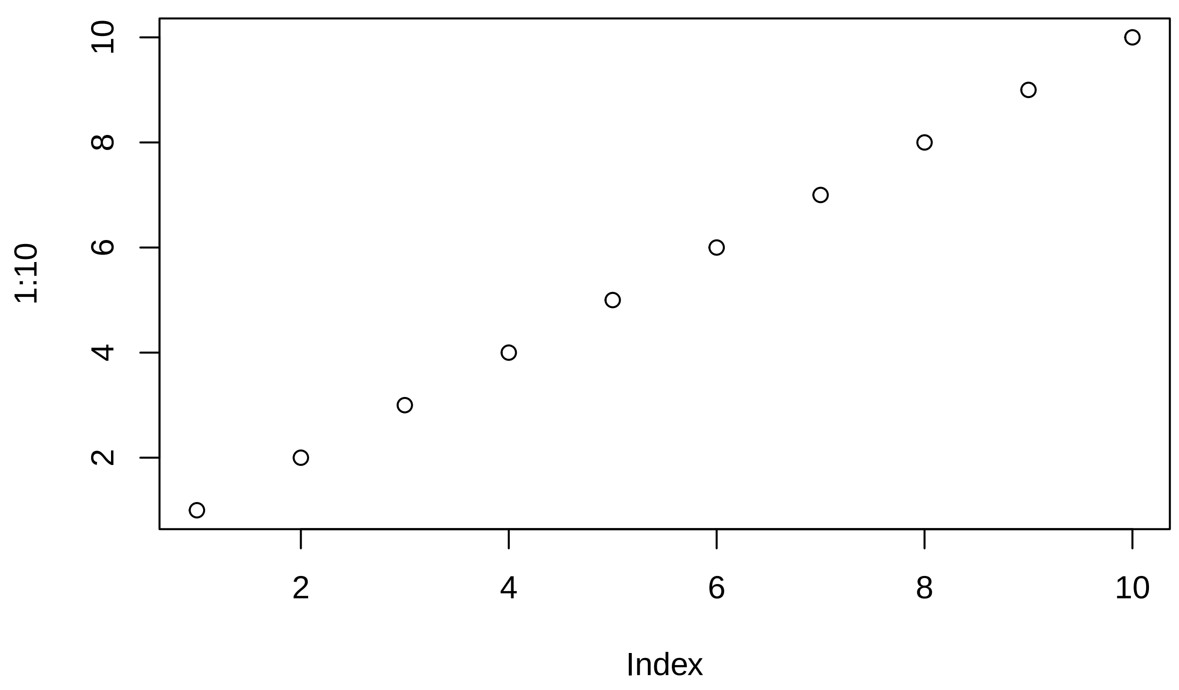
We can write test of course just like any file. We can **style test to be bold** or *italic*

Do:

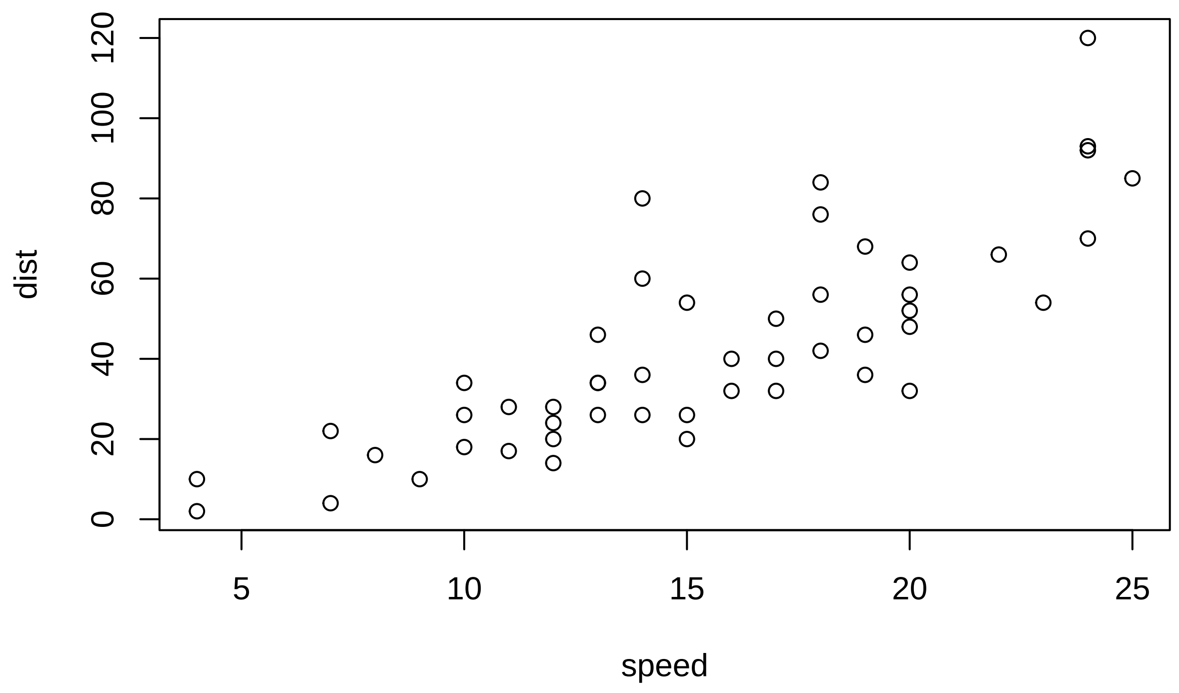
- This
- and that
- and another thing

We can include some code:

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



```
#This is a comment and will not be passed to R
plot(cars)
```



Timet to write a funcrion

> **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)
```

First I want to find the lowest score. I can use the **min()** to find it and the **which.min()** function to find where it is (i.e. its position in the vector)

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

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

Adding the minues symbol means that for all student1 grades we subtract the lowest score I can use minues to get everything in the vector but the lowest

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

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

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

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

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

Does this work for student 2?

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

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

No! Why not?

```
student3
```

```
## [1] 90 NA NA NA NA NA NA NA
```

One great idea is to replace the NA values with zero. Lets do it!

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

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

Lets replace NAs with zero

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

student.prime
```

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

Oke we are so close lets put these bits together to get our mean excluding the lowest score.

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

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

```r
student.prime <- student3
student.prime [is.na(student.prime)]=0
mean(student.prime[-which.min(student.prime)])
```

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

Great! We got it. this works, lets simplify and make as clear as we can.

We can make the variable names more simple.

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

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

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

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

```r
x <- student4
x <- as.numeric(x)
x [is.na(x)]=0
mean(x[-which.min(x)])
```

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

Now we can write our function: All functions have at least 3 things, a name, input args and a body

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

Now all NA emelents have been made into 0 and the function will work

```r
grades (student1)
```

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

now grade a whole class

##apply() this will allow us to apply to all in a sample/table

```r
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
```

we are going to use the super useful **apply()** functions to grade all the students with our **grades()** function

```
apply(scores, 1, 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
```

```
apply(scores, 1, 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
```

```
ans <- apply(scores, 1, grades)
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 the top Student in the Class?

```
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? here we will use **apply()** function again but this time at the columns, which represent different homeworks

```
apply(scores,2, mean)
```

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

I can ignore the NA missing values with na.rm=TRUE

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

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

Replace or 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
```

```r
mask <- scores
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 we can apply on our "masked" scores

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

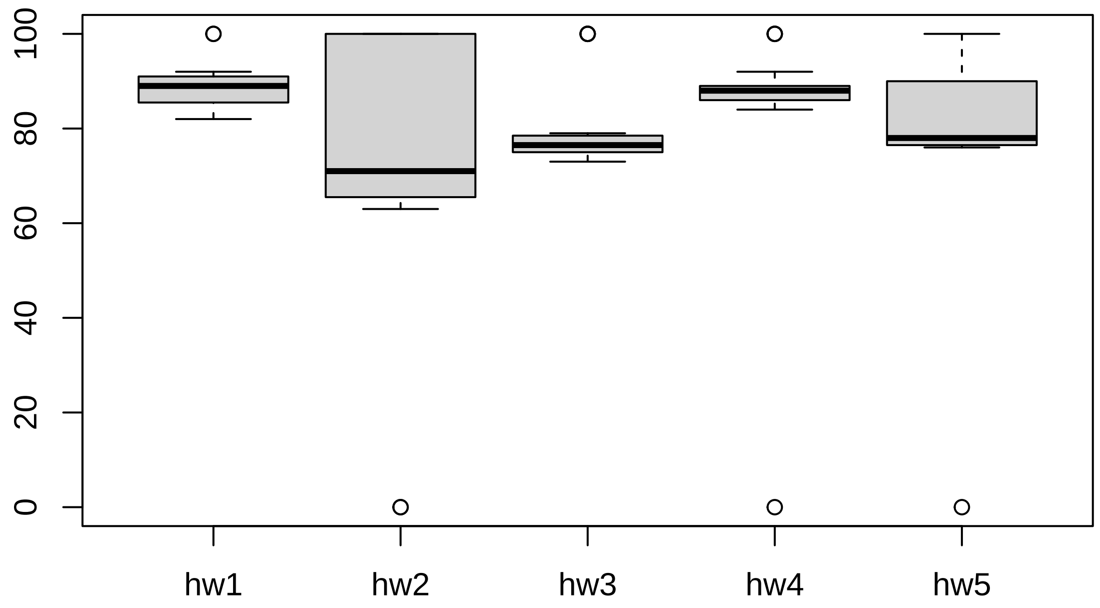
```
##   hw1   hw2   hw3   hw4   hw5
## 89.00 72.80 80.80 85.15 79.25
```

> **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)?

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

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

```r
boxplot(mask)
```



0
20
40
60
80
100
hw1
hw2
hw3
hw4
hw5