Class 6: R function

San Luc (PID: A59010657)

10/15/2021

Quick Rmarkdown intro

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

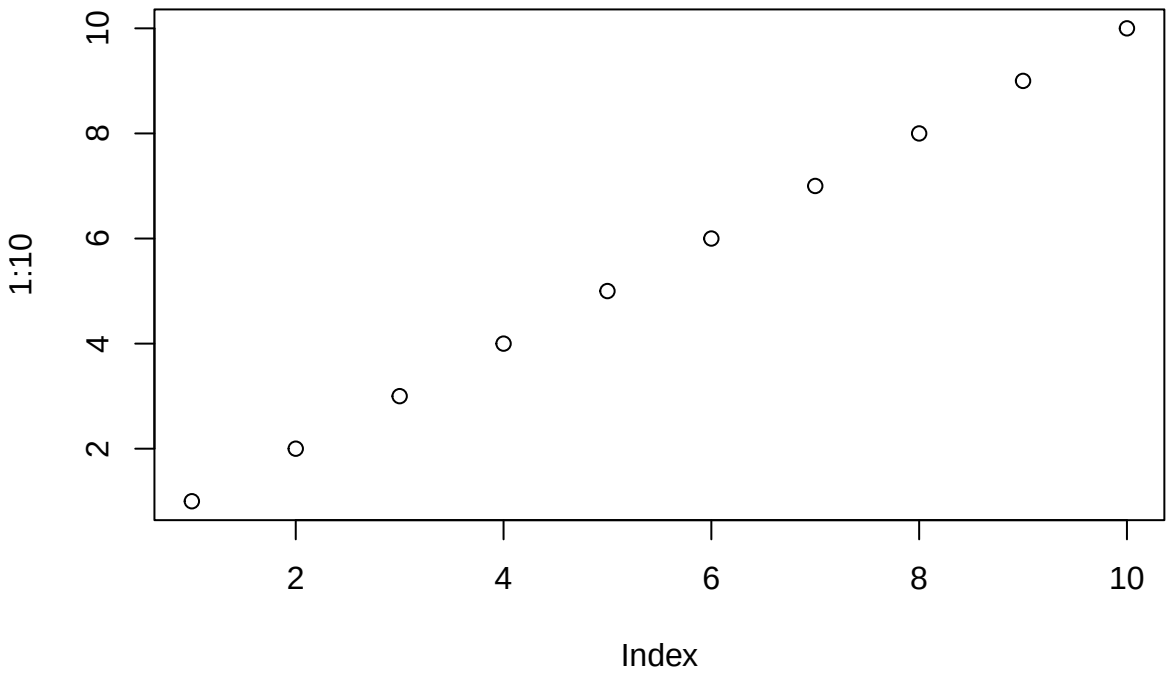
DO:

- this
- and that
- and another thing

This is more text and this is a new line

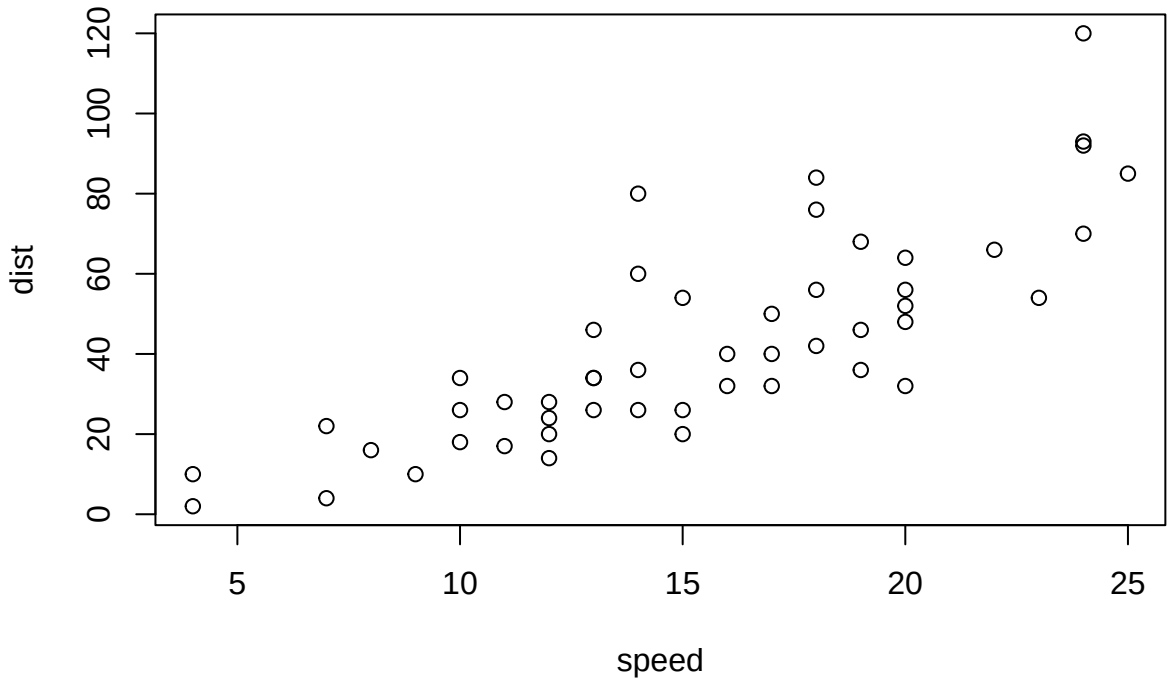
We can include some code:

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




```
#this is a comment and will not be passed to R
```

```
plot(cars)
```



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)
```

For Student 1

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

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

I can use minus to get everything in the vector but the lowest score

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

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

```
new_student1 <- student1[-which.min(student1)]
```

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

```
mean(new_student1)
```

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

For student 2, we cannot do use the same functions because of the NA

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

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

Honestly, we can use the function, right?

For student 2 and 3, we will have to convert all the NA to 0... (that's a lot of zeroes)

Google is our best friend, so ask away!

They suggest us to try the is.na() function. is.na() results in a vector where a TRUE indicates an NA value.

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

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

Let's replace NA with zero.

```
New_student2 <- student2
New_student2[is.na(student2)]=0
mean(New_student2[-which.min(New_student2)])
```

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

We can use the same thing for student 3!

```
New_student3 <- student3
New_student3[is.na(student3)]=0
mean(New_student3[-which.min(New_student3)])
```

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

Ah, what if our data is entered wrong?

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

We can use as.numeric to switch the logical value to numerical.

```
New_student4 <- student4
New_student4 <- as.numeric(New_student4)
New_student4[is.na(student4)]=0
mean(New_student4[-which.min(New_student4)])
```

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

Now we can finally write our function!

All functions have at least three things: - a name - input arguments - a body

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

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

```
##            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
```

```
apply(Score,1,grade)
```

```
##  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
```

```
Overall <-apply(Score,1,grade)
```

Q2. Using your grade() function and the supplied gradebook, Who is the top scoring student overall in the gradebook? [3pts]

```
which.max(Overall)
```

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

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

We can ignore the NA, but it's not too accurate....

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

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

Or we can do the same thing and replace all the NA with zero!

```
Mask <- Score
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
```

```
apply(Mask,2,mean)
```

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

So it's hw 2!

> **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]

We can use **cor()** function to see the correlation between the highest with the average grade score.

```
cor(Mask$hw1, Overall)
```

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

It will be a pain to do them individually, so we will apply

```
apply(Mask,2,cor,Overall)
```

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

Let's make a boxplot for fun (ahaha)

```
boxplot(Score)
```