

Class 6: R Functions

Shitian Li (PID: A13294481)

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*.

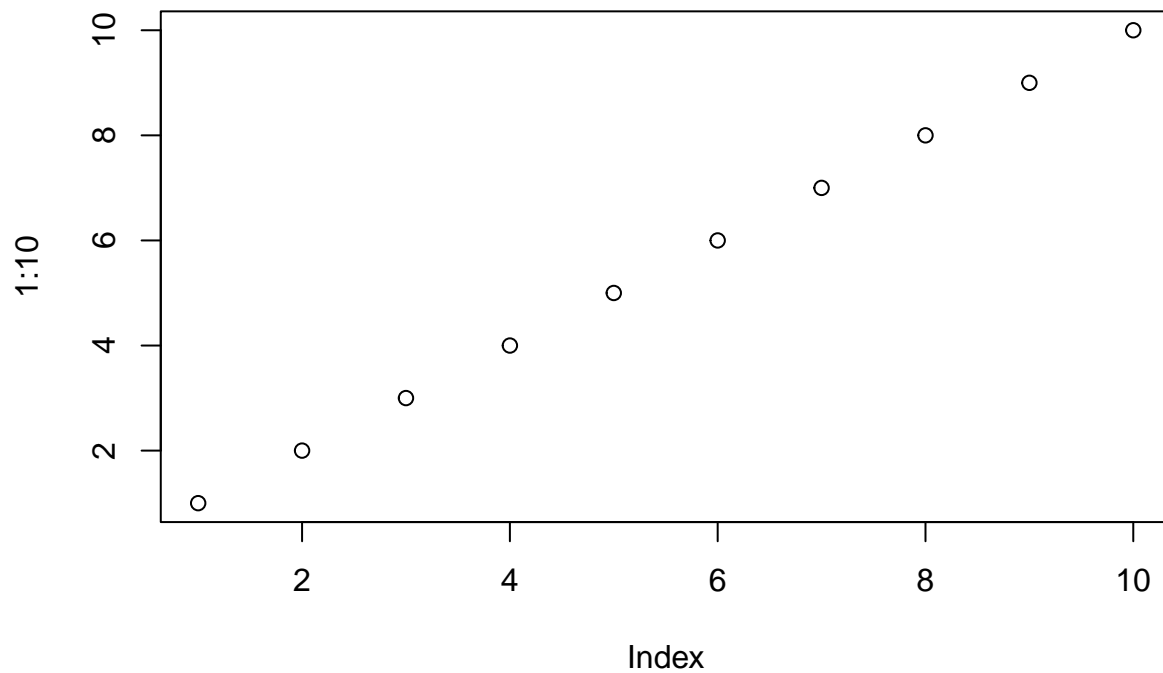
Do:

- this
- and that
- and another thing

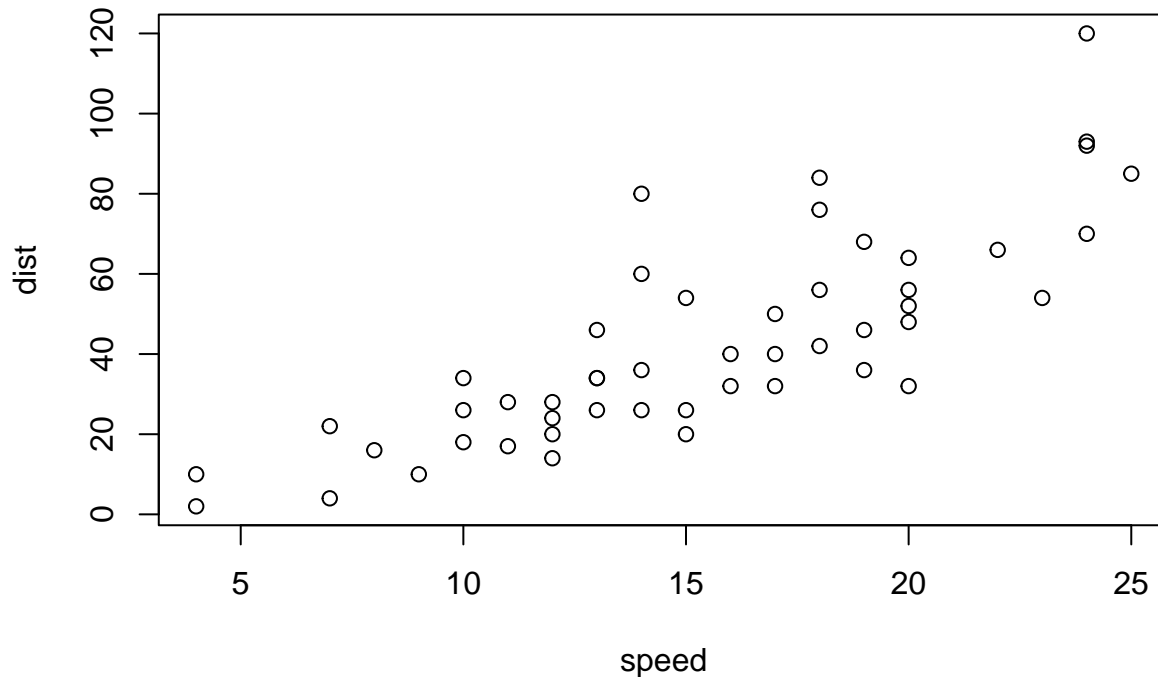
This is more text
and this is a new line.

We can include some code:

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



```
# 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 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]

```
# import some example vectors to test
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, want to find lowest score. We can use `min()` and `which.min()` functions to find them.

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

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

We can use `-` to delete a value

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

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

Now call the `mean()` function:

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

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

This does not work on student 2 because of the *NA* value:

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

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

OK, let's change those NA values into 0

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

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

OK. I'll skip Barry's instructions and directly work on student3 haha.

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

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

```
x <- 1:5  
x
```

```
## [1] 1 2 3 4 5
```

```
x[2] <- 100  
x
```

```
## [1] 1 100 3 4 5
```

```
x[3] <- 200  
x
```

```
## [1] 1 100 200 4 5
```

Great! we got it. Now let's simplify and make it as simple as possible.

We can make variable names more clear

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

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

What if the data is wrong:

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

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

OK, finally we're writing the function:

All functions have at least 3 things:

A name, input args, and a body.

```
grade <- function(x) {
  x <- as.numeric(x)
  x[is.na(x)] = 0
  return(mean(x[-which.min(x)]))
}
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)
grade(student1)
```

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

```
grade(student2)
```

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

```
grade(student3)
```

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

Now grade a whole class

First, read the gradebook for the class.

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

We're going to use **apply()** function to grade all the students with the **grade()** function

```
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 the top scoring student?

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

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

Q3. Which homework hardest?

Here I'll use the **apply()** function again but look at columns.

Maybe we can ignore the NA values with na.rm=T?

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

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

Let's replace or mask NA values to zero

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

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

```
which.min(apply(mask, 2, mean))
```

```
## hw2
##    2
```

Q4. Which homework is the most representative?

Here we'll use the **cor()** function here

```
ans <- apply(scores, 1, grade)
hw_cor <- apply(mask, 2, cor, ans)
hw_cor
```

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

```
which.max(hw_cor)
```

```
## hw5
##    5
```