

10_13_21_writing_R_functions

Kelly_F

10/15/2021

Quick R Markdown intro

We can write text to be **bolded text**, or *italics*.

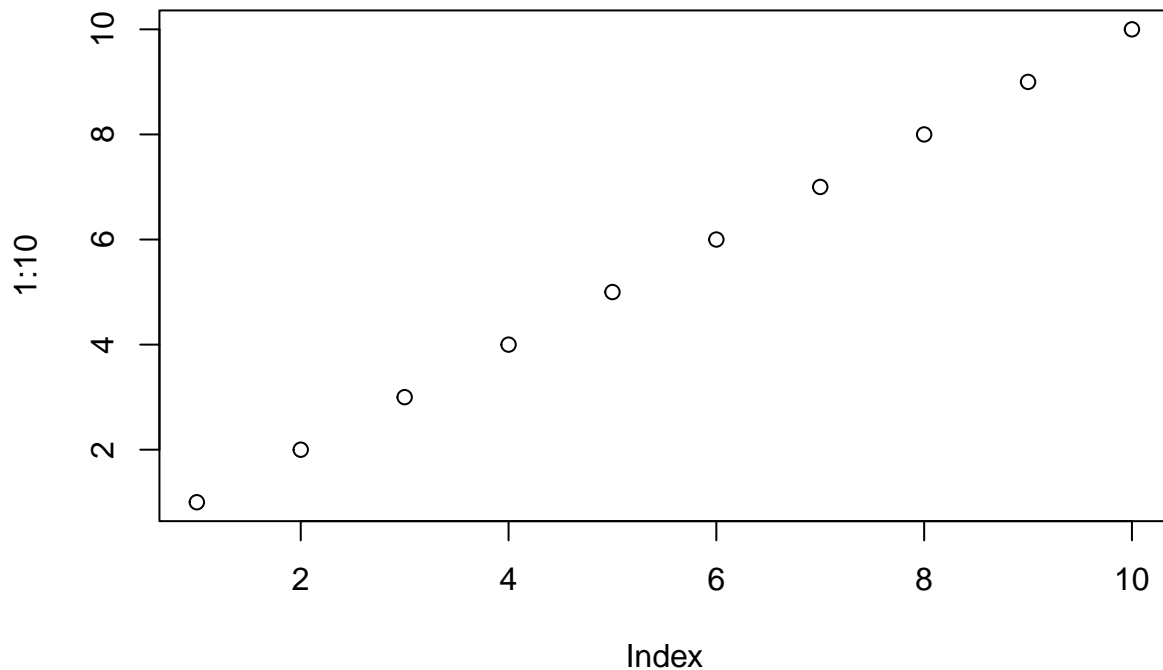
Do (my bullet list example):

- this
- and that
- and another thing

Note, two lines between text are needed to create a new line.

We can include some code:

```
#option+CMD+I to insert a new code block  
#My comment that will not be passed into R  
  
#My Plot  
plot(1:10)
```



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]

```
# 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)

# Finding the lowest score of student 1
min(student1, na.rm=TRUE)
```

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

```
# Find the position of the minimum score
which.min(student1)
```

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

```
# Remove lowest score from student1
student1 <- student1[-which.min(student1)]
student1
```

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

```
# Now take the mean
student1_avg <- mean(student1)
student1_avg
```

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

```
# Repeat example but for student 2. Must remove NA values to calculate mean
# Remove minimum score then calculate mean
student2_avg <- mean(student2[-which.min(student2)],
                     na.rm=TRUE)
student2_avg
```

```
## [1] 92.83333
```

```
# Assign zeros to NA values instead of dropping so that grades aren't inflated. Can then treat zero as 0
student3[is.na(student3)] <- 0
student3
```

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

```
which.min(student3)
```

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

```
student3_avg <- mean(student3[-which.min(student3)])
student3_avg
```

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

```
# Now write a function to calculate overall homework grade

grade <- function(student_grades){ # A function titled "grade" that needs a vector of student grade scores
  student_grades[is.na(student_grades)] <- 0 # Replace NA values with zero
  overall_grade <- mean(student_grades[-which.min(student_grades)]) # Drop lowest score and calculate t
}
```

```
# Let's test our function
student2 <- c(100, NA, 90, 90, 90, 90, 97, 80)
student3 <- c(90, NA, NA, NA, NA, NA, NA, NA)
student2_grade <- grade(student2)
student2_grade
```

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

```
student3_grade <- grade(student3)
student3_grade
```

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

```
# Now lets apply our grade function this to a df of grades
```

```
# Import data
```

```
df <- read.csv("/Volumes/GoogleDrive/My Drive/Courses/BGGN_213/10_15_21_writing_functions/student_homeworks.csv")
```

```
#Test out "grade" function on one row of data
```

```
# Take first row for student1 grades
```

```
student1_grades <- as.numeric(df[1,])
student1_grades
```

```
## [1] 100 73 100 88 79
```

```
# Apply grade function to student1 to calculate overall grade
```

```
student1_overall_grade <- grade(student1_grades)
student1_overall_grade
```

```
## [1] 91.75
```

```
#Calculate overall grade for all students in the df
```

```
for (i in 1:nrow(df)){
  student_grades <- as.numeric(df[i,])
  df$overall.grade[i] <- grade(student_grades)
}
df
```

```
##           hw1 hw2 hw3 hw4 hw5 overall.grade
## student-1 100 73 100 88 79           91.75
## student-2 85 64 78 89 78           84.35
## student-3 83 69 77 100 77           85.75
## student-4 88 NA 73 100 76           85.75
## student-5 88 100 75 86 79           88.95
## student-6 89 78 100 89 77           89.55
## student-7 89 100 74 87 100          93.55
## student-8 89 100 76 86 100          93.35
## student-9 86 100 77 88 77           88.55
## student-10 89 72 79 NA 76           81.55
## student-11 82 66 78 84 100          87.15
## student-12 100 70 75 92 100          91.75
## student-13 89 100 76 100 80          92.15
## student-14 85 100 77 89 76           88.55
## student-15 85 65 76 89 NA           81.35
## student-16 92 100 74 89 77           89.95
## student-17 88 63 100 86 78           88.75
## student-18 91 NA 100 87 100          93.95
## student-19 91 68 75 86 79           84.55
## student-20 91 68 76 88 76           84.55
```

```
# Lets try an alternative approach with the "apply" function
df2 <- read.csv("/Volumes/GoogleDrive/My Drive/Courses/BGGN_213/10_15_21_writing_functions/student_homescores.csv")

overall_grades <- apply(df2, 1, grade)
```

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

Answer: Student #18

```
# Determine which student the maximum grade comes from
which.max(overall_grades)
```

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

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

Answer: homework 2

```
# Replace NA values with zeros in the data frame
mask <- df2
mask[is.na(mask)] <- 0

# Calculate average score per homework
avg_homework <- apply(mask, 2, mean)
avg_homework
```

```
## 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)? [1pt]

Answer: homework 5 was the most predictive

```
# Pearson correlation between homework 1 and overall grades
cor(mask$hw1, overall_grades)
```

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

```
# Use apply function to do all correlations at once
apply(mask, 2, cor, overall_grades)
```

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

```
# Make a graph of our grade scores
boxplot(mask)
```

