Class 06: R Functions

Chloe J. Welch

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*.

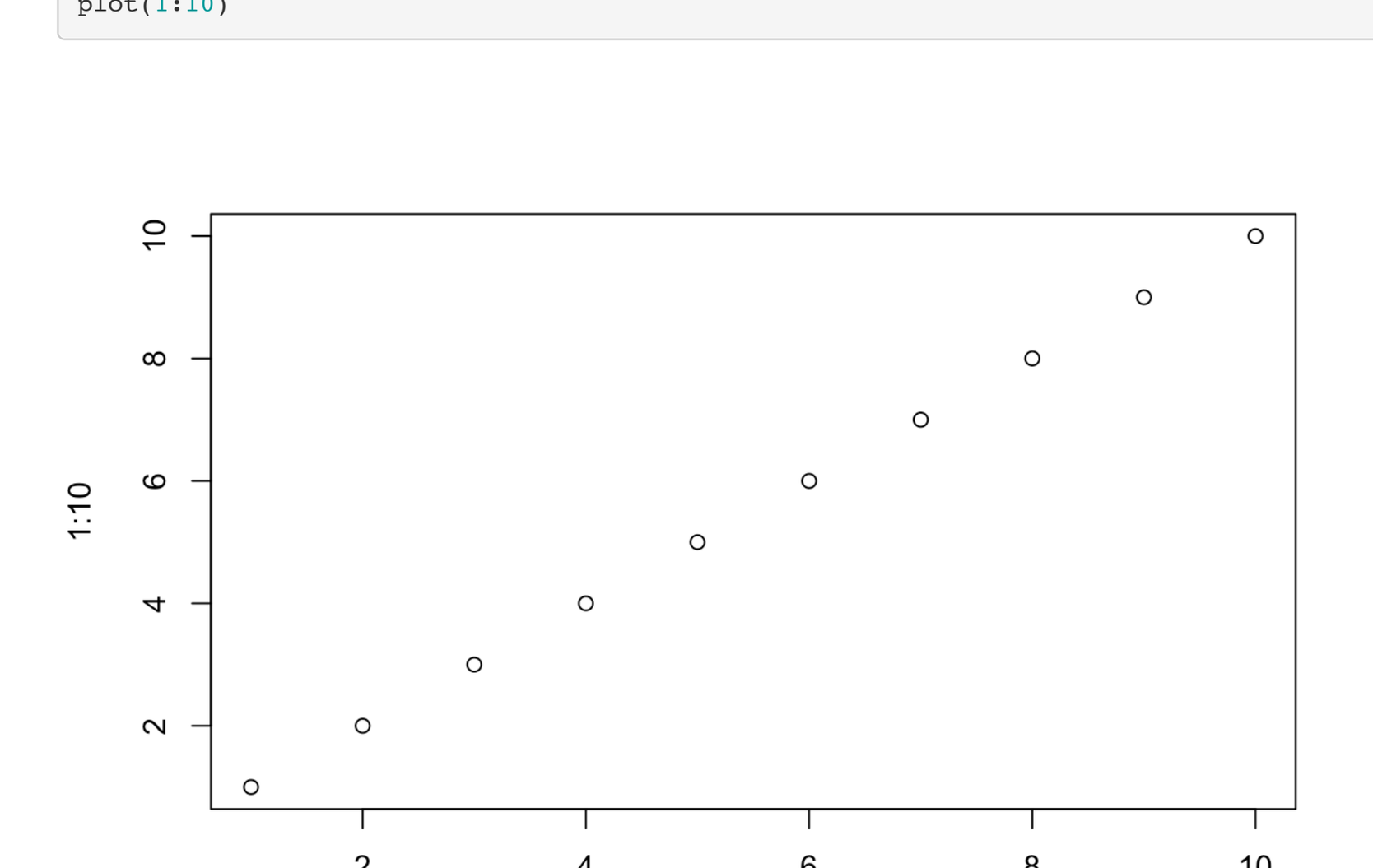
Do:

- this
- and that
- and another thing

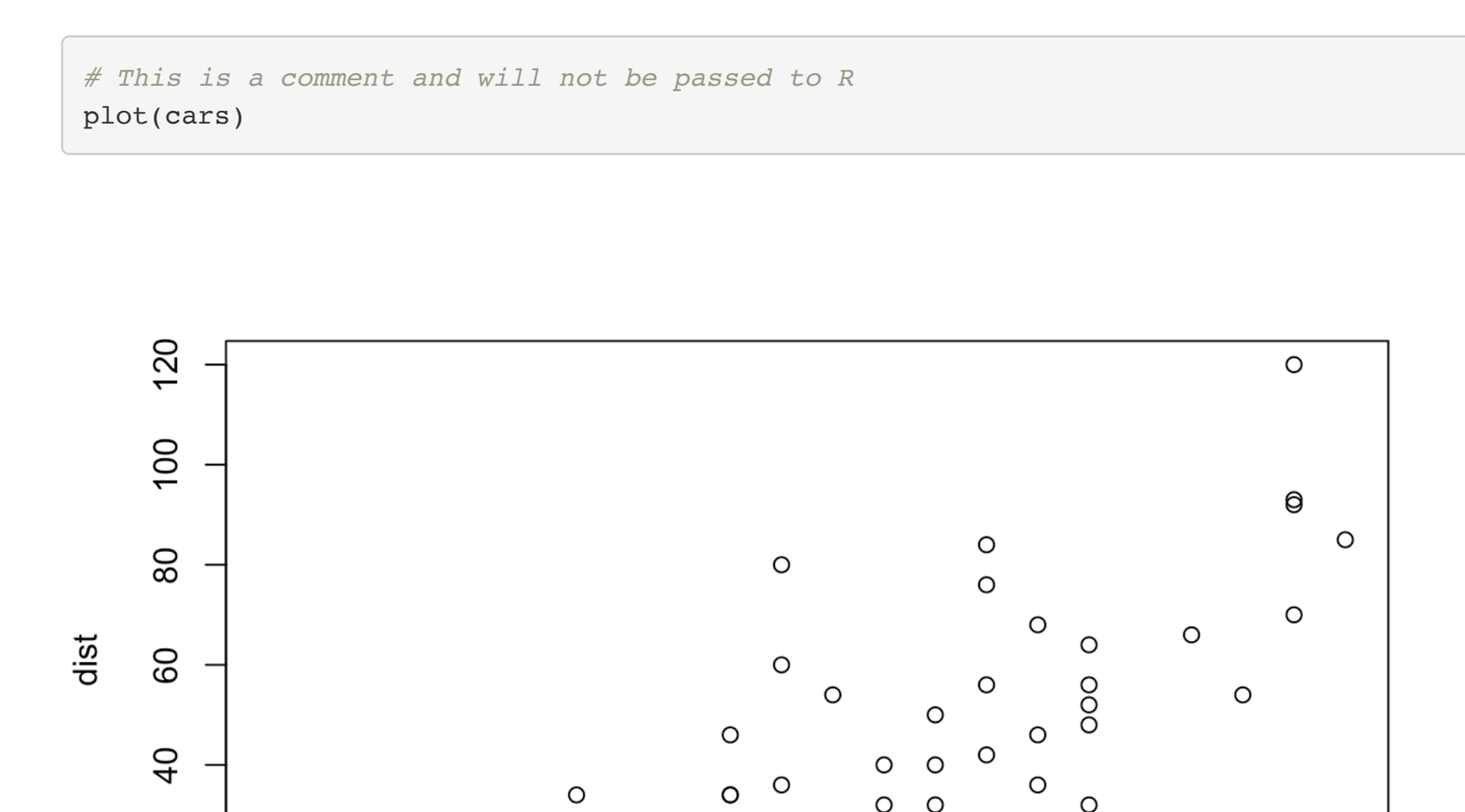
This is more text and this is a new line

We can include some R 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)
```

First, I want to find the lowest score. I can use the **min()** function to find it and the **which.min()** function to find where it is (in other words, what is the position in the vector)?

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

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

I can use "minus" to get everything in the vector, except for the lowest score.

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

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

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

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

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

Does this work for student2?

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

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

NO! Why does this not work?

```
student2
```

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

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

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

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

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

```
student3
```

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

To identify if there are NA's (missing values) present, we can try replacing all NA values with zero. Let's try!

Give this command a try:

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

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

The is.na() function returns a logical vector where TRUE elements indicate the presence of NA values.

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

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

```
student2[is.na(student2)]
```

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

Let's replace NA values with zero

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

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

```
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
```

Next, let's try putting everything 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
```

How about student3?

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

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

Great! We did it! This works. Now, let's try simplifying everything and making things as clear as possible.

We can make the object names more clear

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

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

What if the data is entered incorrectly?

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

The as.numeric function creates or coerces objects of type "numeric". is.numeric is a more general test of an object being interpretable as numbers.

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

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

Now, let's write out function. We need a name, input arguments, and a body.

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

```
grade(student1)
```

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

Now, we can grade a whole class

First, we must read the gradebook

```
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
```

Next, we are going to use the "Apply function" feature: **apply()** function to grade all the students within our **grade()** function.

```
apply(scores, 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
```

```
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**. From your analysis of the gradebook, which homework was toughest on students (i.e. obtained the lowest scores overall)?

Here, I will use the **apply()** function again except this time, look at the columns which represent different homework assignments.

```
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
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  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
```

```
mask <- scores
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  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
```

```
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
```

Remember, in this case, TRUE means the student did NOT submit that specific homework assignment.

```
mask <- scores
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  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
```

```
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 use apply on our "masked" scores

```
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)?

Here, we will use the **cor()** function

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

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

I ca call the **cor()** for every homework and get a value for each, but what a tedious process. Instead, I will use **apply()** and do everything with one command

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

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

Make a boxplot

```
boxplot(scores)
```