STAT 33A Workbook 1

CJ HINES (3034590053)

Sep 3, 2020

This workbook is due **Sep 3, 2020** by 11:59pm PT.

The workbook is organized into sections that correspond to the lecture videos for the week. Watch a video, then do the corresponding exercises *before* moving on to the next video.

Workbooks are graded for completeness, so as long as you make a clear effort to solve each problem, you'll get full credit. That said, make sure you understand the concepts here, because they're likely to reappear in homeworks, quizzes, and later lectures.

As you work, write your answers in this notebook. The first lecture video this week will teach you more about how the notebook works.

Please do not delete the exercises already in this notebook, because it may interfere with our grading tools.

Packages & R Notebooks

Watch the "Packages & Notebooks" lecture video.

Exercise 1

You don't need to write anything in the notebook for this exercise.

Install the `rmarkdown` and `tinytex` packages.

Confirm that your setup works by knitting this file to PDF.

If you run into any trouble, ask on Piazza or in office hours as soon as possible. You'll need to be able to knit to turn in the assignment.

Markdown

Watch the "Markdown" lecture video.

For the remaining exercises, answer questions with complete sentences, and put code in code chunks. You can make as many new code chunks as you like.

In the notebook, you can run the line of code where the cursor is by pressing `Ctrl` + `Enter` on Windows or `Cmd` + `Enter` on Mac OS X. You can run an entire code chunk by clicking on the green arrow in the upper right corner of the code chunk.

Exercise 2

Write 3-5 sentences introducing yourself. Try out some of the Markdown formatting: italics, bold, lists, quotes, code chunks, and links.

After you finish, knit the notebook to check that your formatting was applied correctly.

YOUR ANSWER GOES HERE:

Maria Christina Juilleanne Ceniza (CJ) Hines

UC Berkeley 2023 | She/Her

> Hello, my name is CJ Hines. I am a sophomore at Cal studying Economics and Computer Science. I was born in the *Philippines* and raised in *San Diego, CA*. I love `yoga`, `history`, and am a huge `sports` fan!
>
> - Some of my favorite teams are:
>   1. **The Toronto Maple Leafs**.
>   2. **Paris Saint-Germain F.C.**.
>   3. **The Toronto Raptors**.

```r
c(toupper("go leafs go"), ("!!!"))
```

```
## [1] "GO LEAFS GO" "!!!"
```

Exercise 3

Use Markdown to make:

- A word that's both bold and italic
- A link that's italic
- A nested list (a list with indented subelements)

Try to figure these out by experimentation before asking or searching online.

YOUR ANSWER GOES HERE:

> ***Word***
>
> *Link*

1. List
    1. Is Nested
        1. Even More Nested

Vectors

Watch the "Vectors" lecture video.

Exercise 4

Another way to create vectors is with the `rep()` function. The `rep()` function creates a vector by replicating a value or vector of values.

1. The first parameter of `rep()` is the thing to replicate. The second parameter, `times`, is the number of times to to replicate. Use `rep()` to make a vector with 10 elements, all equal to 78.

2. What happens if you pass a vector as the first argument to `rep()`? Give some examples.

3. Skim the help file `?rep`. What happens if you pass a vector as the second argument to `rep()`? The help file might seem a bit cryptic, so you'll also need to experiment. Give some examples.

YOUR ANSWER GOES HERE:

```
rep(78, times = 10)
```

```
##  [1] 78 78 78 78 78 78 78 78 78 78
```

```
x = c(5, 10, 15)
rep(x, times = 10)
```

```
##  [1]  5 10 15  5 10 15  5 10 15  5 10 15  5 10 15  5 10 15  5 10 15  5 10 15  5
## [26] 10 15  5 10 15
```

```
y = c(3, 6, 9)
rep(x, y)
```

```
##  [1]  5  5  5 10 10 10 10 10 10 15 15 15 15 15 15 15 15 15
```

```
rep(y, x)
```

```
##  [1] 3 3 3 3 3 6 6 6 6 6 6 6 6 6 6 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9
```

Exercise 5

Yet another way to create vectors is with the `seq()` function. The `seq()` function creates a vector that contains a sequence of numbers.

Skim the help file `?seq`. Give some examples of creating vectors with the `seq()` function.

YOUR ANSWER GOES HERE:

```
seq(5, 9)
```

```
## [1] 5 6 7 8 9
```

```
seq(5, 100, by = 15)
```

```
## [1]  5 20 35 50 65 80 95
```

```
seq(5, 15, length.out = 3)
```

```
## [1]  5 10 15
```

Subsets of Vectors

Watch the "Subsets of Vectors" lecture video.

Exercise 6

Try out each of the following:

- Getting a single element of a vector
- Reassigning an element of a vector
- Getting multiple elements of a vector
- Getting multiple elements of a vector using negative indices

YOUR ANSWER GOES HERE:

```
z = seq(30, 70, by = 10)
z[1]
```

```
## [1] 30
```

```
z[3] = 500
z[c(2,4)]
```

```
## [1] 40 60
```

```
z[-5]
```

```
## [1]  30  40 500  60
```

Exercise 7

Consider this code:

```
x = seq(1, 5) * 10
x[c(3, 4, 5)][1]
```

```
## [1] 30
```

Try running the code to see what the result is. Then explain what's happening in the second line (it may be useful to explain what x[c(3, 4, 5)] does first).

YOUR ANSWER GOES HERE:

x[c(3, 4, 5)] creates a new vector of x's 3rd, 4th, and 5th elements which would be 30 40 50. Then x[c(3, 4, 5)][1] would select the 1st element of that vector which is 30.

Exercise 8

The `sort()` function sorts the elements of a vector. For instance:

```r
x = c(4, 5, 1)
sort(x)
```

```
## [1] 1 4 5
```

Another way to sort vectors is by using the `order()` function. The order function returns the *indices* for the sorted values rather than the values themselves:

```r
x = c(4, 5, 1)
order(x)
```

```
## [1] 3 1 2
```

These can be used to sort the vector by subsetting:

```r
x[order(x)]
```

```
## [1] 1 4 5
```

The key advantage of `order()` over `sort()` is that it can also be used to sort one vector based on another, as long as the two vectors have the same length.

Create two vectors with the same length, and use one to sort the elements of the other.

YOUR ANSWER GOES HERE:

```r
vector1 = c(10, 100, 1000, 1)
vector2 = c(202, 333, 44, 55)
vector1[order(vector2)]
```

```
## [1] 1000    1   10  100
```

```r
vector2[sort(vector1)]
```

```
## [1] 202  NA  NA  NA
```

Submitting Your Work

Congratulations, you made it through the first workbook!

You need to submit your work in two places:

- Submit this Rmd file with your edits on bCourses.
- Knit and submit the generated PDF file on Gradescope.