Problem Set 1

Phuong Tseng

8/23/2022

- Submission process: Please submit your assignment directly to Gradescope. You can do this by knitting your file and downloading the PDF to your computer. Then navigate to Gradescope.com or via the link on BCourses to submit your assignment. .

Helpful hints:

- Knit your file early and often to minimize knitting errors! If you copy and paste code from the slides, you are bound to get an error that is hard to diagnose. Typing out the code is the way to smooth knitting. We recommend knitting your file each time after you write a few sentences/add a new code chunk, so you can detect the source of the knitting error more easily. This will save you and the teaching team time!
- Please make sure that your code does not run off the page of the knitted PDF. If it does, we can't see your work. To avoid this, have a look at your knitted PDF and ensure all the code fits in the file. When it doesn't, go back to your .Rmd file and add spaces (new lines) using the return or enter key so that the code runs onto the next line.

First, try knitting by pressing Alt + Shift + K (Windows) or Command + Shift + K (Mac) or by clicking the "Knit" icon above the Source pane. You should see that your file has knitted into a pdf.

Question 1

Un-comment and fix the following code to make it run. Note: You are only using numeric values in this question.

```
my_variable <- 2

my_other_variable <- 8

my_variable + my_other_variable
```

```
## [1] 10
```

Question 2

Create four variables with the following data types:

1. character
2. numeric
3. integer
4. logical

```
char <- "character"
class(char)
```

```
## [1] "character"
```

```
num <- 9
class(num)
```

```
## [1] "numeric"
```

```
int <- as.integer(-20)
typeof(int)
```

```
## [1] "integer"
```

```
logical_x <- 7 == 14
logical_x
```

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

```
list(c(char, num, int, logical_x))
```

```
## [[1]]
## [1] "character" "9"         "-20"       "FALSE"
```

Question 3

Using the variables x and y below, perform the following comparisons.

```
x <- 10; y <- 11

### is x equal to y? ###

# your code here
x == y #false
```

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

```
### does x not equal y? ###

# your code here
x != y #true
```

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

```
### is x greater than or equal to y? ###

# your code here
x >= y
```

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

Question 4

R is a powerful calculator that can help us become more efficient epidemiologist.

Recall that an odds ratio is calculated by the following: (a / c)/(b / d) or (a * d)/(b * c).

Suppose a number of people became ill after exposure to cheesecake. Our two levels of exposure to cheesecake are (1) those who ate cheesecake and (2) those who did not eat cheesecake.

```
# run the following code to view our 2x2 table
# notice how we used one of R's base function called "matrix"
# we directly inputted our values in a list format c("","",...)
# added the argument ncol = 2 to split the list into two columns
# added the argument byrow = TRUE to first complete the rows then the columns

cheesecake_exposure <- matrix(c(15, 36, 18, 25), ncol = 2, byrow = TRUE)
# directly named the two columns
colnames(cheesecake_exposure) <- c("Cases", "Controls")
# directly named the two rows
rownames(cheesecake_exposure) <- c("Exposed", "Not Exposed")
# executed our variable to view the output
cheesecake_exposure
```

```
##             Cases Controls
## Exposed        15       36
## Not Exposed    18       25
```

Calculate the odds ratio of becoming ill due to cheesecake. Save the odds ratio as an object in your environment.

```
# your code here (a / c)/(b / d) or (a * d)/(b * c)
#print(cheesecake_exposure[1,])
#print(cheesecake_exposure[2,])
a = 15
b = 36
c = 18
d = 25
cheesecake_exposure <- matrix(c(a, b, c, d), ncol = 2, byrow = TRUE)
cheesecake_exposure
```

```
##      [,1] [,2]
## [1,]   15   36
## [2,]   18   25
```

```
myresult <-  (a/c)/(b/d)
myresult
```

```
## [1] 0.5787037
```

Question 6

Create two vectors with the following numeric values in the presented order. Then, add them together.

- Use the c() function for the first vector: 1, 2, 3, 4, 5
- Use the colon (":") operator for the second vector: 51, 52, 53, 54, 55

As a logic check, you should expect an output with one vector: 52, 54, 56, 58, 60

```
# your code here
first <- c(1,2,3,4,5)
second <- 51:55
one <- first + second
one
```

```
## [1] 52 54 56 58 60
```

Question 7

(a) Write a line of code that will pull up the documentation for the base R function called "round".
(b) What arguments does the function require?
(c) What is the default value for the second argument?

```
# your code here
help("round")

ceiling(x)
```

```
## [1] 10
```

```
floor(x)
```

```
## [1] 10
```

```
#trunc(x, ...)
round(x, digits = 0)
```

```
## [1] 10
```

```
signif(x, digits = 6)
```

```
## [1] 10
```

It requires x which is a numeric vector or round and signif a complex vector And it requires digits or integer indicating the number of decimal places (round) or significant digits (signif) to be used. Negative values are allowed (see 'Details') then arguments to be passed to methods. Default is 0.

Question 8

Write a single line of code to divide 377 by 120 and round the result to three decimal places. Save this rounded result to a variable called "ptolemy_pi" in your environment. Test if ptolemy_pi is equal to R's built-in approximation of pi (stored by default as the object "pi").

```
pi
```

```
## [1] 3.141593
```

```
# your code here
ptolemy_pi <- round(377/120, 3)
ptolemy_pi
```

```
## [1] 3.142
```

```
pi == ptolemy_pi
```

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

Question 9

Use an if/else statement to print out "greater than 1" if the odds of becoming ill due to cheesecake (from question 4) is higher than 1.0. Print "not greater than 1" if the odds are not higher than 1.0.

You did it! Please knit to a pdf, download, and submit to Gradescope.