Problem Set 1

Your name here

date here

- Due date: August 24 at 12:00 AM PST
- 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 yu 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 hte return or enter key so that the code runs onto the next line.

First, try pressing Alt + Shift + K. You should see that your file has knitted into a pdf. You can also knit by pressing the knit button above next to the magnifiying glass.

Let's load the reprex library.

```
# note: to install packages for the first time you run this function
# install.packages("reprex")
library(reprex)
```

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 <- eight

#my_variable_ + my_Other_variable
```

Question 2

Create five variables with the following type of data types:

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

```
# your code here
```

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

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

# your code here

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

# your code here
```

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 of becoming ill due to cheesecake.

```
# your code here
```

Question 5

For this question use the reprex function discussed in lecture for each of the variable assignments.

Create four variables to represent the 2x2 table in Question 2. We suggest using a, b, c, and d to match the 2x2 format.

Using these variables, calculate the odds ratio again and assign it to a new variable.

```
# your code here
```

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
```

Question 7

In this question you will create a data frame. Below is code for how to do this:

```
# data frame with three columns and three rows
# notice how we start with the column name and then the row values
# each column, in this example, has three values
df_example <- data.frame("column_1" = 1:3,
                         "column_2" = c("string_1", "string_2", "2"),
                         # here we add an NA value due to missing data
                         "column_3" = c(NA, "string_3", 50))
df_example
```

```
##   column_1 column_2 column_3
## 1        1 string_1     <NA>
## 2        2 string_2 string_3
## 3        3        2       50
```

Now you try! Create a data frame with three columns and the following values:

- Column 1: ID
  1, 2, 3
- Column 2: NAME
  "Pam", "Jim", "Dwight"
- Column 3: AGE
  40, NA, 48

```
# your code here
```

Question 8

With your new data frame, find the following values:

- length =
- typeof =
- class =

```
# your code here
```

Question 9

Missing data, ugh! Missing data are ubiquitous part of social, behavioral, and health sciences. R can help us find where missing data is present. For this question, please read the help documentation of this useful tool.

```
# pull up help documentation by putting a question mark before the function name
?is.na()
```

Use is.na() to find the missing data in the data frame created in Question 7. List this observation's available information and which column is NA.

```
# your code here
```

Challenge

Create a function that lets you add two numbers together. For example, if I had two arguments, 1 and 2, the output should result in 3. You can use the ?function help documentation and/or google.

```
# your code here
```

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