1. To answer this question, we will create 3 vectors named *first, second*, and *third*.

   (a) Create the vectors first, second, and third. Each vector should be made up of the following elements: first=(6,10,9,10,8,4,9,1,3,6), second=(4,2,3,2,2,2,5,5,1,2), third=(12,12,13,11,13,10). Make sure to print your vectors to make sure that you have entered the correct values in the correct locations.

   (b) Try adding the vectors *first* and *second* together. Is the result what you expect?

   (c) Now try adding the vectors *first* and *third* together. What is the length of *third* compared to *first*? How were we able to add these two vectors together?

   (d) Let's use some of the vector specific functions. What is the sum of the elements in each of the vectors? Add these sums together to get a grand total?

   (e) Feel free to play around with some of the functions for vectors, and how they can be combined into equations involving vectors.

   (f) Based on your experiences with vectors, can you compare and contrast using R and Excel to perform data calculations? For example, Excel allows you to perform calculations like =SUM(firstcell:lastcell) and others. How does performing this type of calculation in R compare to performing this type of calculation in Excel?

2. Now we are going to play with different ways of creating vectors in R. See if you can predict what the following commands will do, and then check your answer using R.

   (a) seq(from=1,to=10,by=2)

   (b) seq(from=5,to=1,by=-1)

   (c) seq(from=1,to=5,length=11)

   (d) x=c(1,2,3), rep(x, times=3)

   (e) rep(x,each=3)

   (f) y=x, rep(x,times=y)

3. Recall that we made some vectors named *first, second*, and *third* in question 1. We will be using them again here to answer parts of this exercise.

   (a) What happens if we bind the vectors together using the command `c(first,second,third)`?

   (b) We are going to use a new function that is very similar to `c()` in this part. This function is called `cbind()` which stands for *column bind*. Type the command `cbind(first,second)`. What does this create? What happens if we type `cbind(first,second,third)`? What do you think the error message means?

   (c) We are going to use a new function that is very similar to `c()` in this part. This function is called `rbind()` which stands for *row bind*. Type the command `rbind(first,second,third)`. How does this compare to what happened when we used `cbind()`? What happens if we type the command `rbind(first,second,5)`?

4. Using what you know about the commands discussed in lecture and introduced above, we are going to create a data layout for an experimental scenario.

   Imagine that we are sampling from 5 different landfill sites. We are planning to sample once a month for the months of June, July, August, and September (4 sampling times). In each sample, we are testing for 3 quantities of interest: dissolved solids; suspended solids; and phytoplankton levels. Create 3 vectors named *Site, Time*, and *Analyte*. These vectors should be constructed such that if we use the command `cbind(Site,Time,Analyte,NA)` it creates a table representing all of our sampling combinations and has a column of NA's to represent where we will fill in our observation values.

   hint: There is more than one way to create this table that will work.

   hint: It can be extremely useful to think about how the table should look using a pen and paper before using any commands in R.

5. You are going to have to read some data into R in order to complete this question. Look for the data set called H20Qex.csv that has been provided for you in the workshop folder. Choose a name for this data, and read it into R using the `read.table()` function.

(a) After reading in the data, test the command `dim(dataname)`. Write down the values that this function gives us.

(b) One of the observation values is missing. We don't want this missing observation to ruin our analysis, so please remove this row of data.

(c) If we use the command `dim(dataname)` now, what values does this function give us? What does the change reflect about our data?

(d) Try typing `dim(dataname)[2]`. What does the returned value suggest about the type of object that is created by the `dim()` function?

(e) Find the minimum, maximum, and mean of the *observation* variable?

(f) More sensibly, find the mean observation for each of the different *analyte* categories

(g) The 28th observation is a mistake! It should be 1.578 rather than 15.578. Correct this and find a new mean for this analyte.

(h) It turns out that the lab that produced the results couldn't accurately detect values below 0.7. Values below this are considered below the *Method Detection Limit*. Create a vector of TRUE and FALSE called *Below* that indicates whether a given observation is below the detection limit (TRUE), or above the detection limit(FALSE). Bind this column to the data set.

(i) Using R, see if you can find a way to determine the proportion of values that fall below the method detection limit.

hint: You can sum the T/F column

(j) What proportion of observations are censored in the 2nd analyte group?