# Homework 6: Loops/Functional Programming

## For-Loops

1. Set the seed using the `set.seed` function to 1992.
2. Create a vector called `random_length` that is a random integer between 1 and 100.
3. Create a new vector of 100 random integers called `random_int` that can have values that range from 0 to 1000.
4. Without using a pre-made function (e.g., use a for-loop), count up all of the even integers in the `random_int` vector. Save this number as `number_even`.
5. Without using a pre-made function, find the maximum number of the `random_int` vector.

## Lapply and Map

Recall that the `lapply` function and the `purrr::map` functions are functions that allow for "functional programming". This means that we can loop through lists without actually going through the hassle of writing a for-loop. These next problems will get you more aquainted with the use of these.

### Map/Lapply

1. Copy the following code:

```
input <- c("Illness", "Lions", "Love", "Universe", "Michael",
    "Intense", "Night", "Apple", "Tiger", "Icon")
```

2. Using the `lapply` function, and a function and the `str_sub` function, extract the first letter form each of the words in the `input` vector and make a new VECTOR (not list!) called `secret_word` that contains each of these first letters. For instance, your vector should have 10 elements, the first element being "I", the second being "L" etc.
3. Using a for-loop, paste together the elements of `secret_word` into a new variable named `word`. This `word` variable should be a single word.
4. Repeat number 2, but use the `purrr::map` function.

### Looping through files (without looping)

The purpose of this exercise is to loop through many files of data sets and combine them together within a single script. This can actually be done quite quickly.

1. Download the 3 excel files: 2013.xlsx, 2014.xlsx, 2015.xlsx. Save them in the same folder.

2. Using the `list.files` function, create a new list called `files` which exclusively contains all of the xlsx files specified above.

3. Now we'll start with for-loops: using a for-loop, loop through each of these filenames. The for-loop should do the following: (1) read in the data (2) save the data as `crime_2013` for the 2013.xlsx file and`crime_2014` for the 2014.xlsx file etc.

4. Now bind together the `crime_2013`, `crime_2014`, `crime_2015` data frames together into one large data frame called `crime`.

5. Note that we can do this MUCH faster using the `lapply` and `purrr::map_df` functions. Choose whichever function you want to use, and loop through the files with the goal of creating the `crime` data frame you created from 4. You should be able to do this in just a line or so of code.