Lab 1: `R` and Rmarkdown

0.1 Basic R

1. Use the `rep()` function to construct the following vector: `1 1 2 2 3 3 4 4 5 5`
2. Use `rep()` to construct this vector: `1 2 3 4 5 1 2 3 4 5 1 2 3 4 5`
3. Create a vector of 1300 values evenly spaced between 1 and 100.
4. How many of these values are greater than 91? (Hint: see `sum()` as a helpful functions.)

   Using the vector you created of 1300 values evenly spaced between 1 and 100,
5. Modify the elements greater than 90 to equal 9999.
6. View (not modify) the first 10 values in your vector.
7. View (not modify) the last 10 values in your vector.

0.2 Data Frames

1. Make a data frame with column 1: `1,2,3,4,5,6` and column 2: `a,b,a,b,a,b`
2. Select only rows with value "a" in column 2 using logical vector
3. `mtcars` is a built-in data set like `iris`: Extract the 4th row of the `mtcars` data.

0.3 Functions

1. Make a function called `my_mean()` that takes a vector of numbers as input and returns the mean of the vector.
2. Alter your `my_mean()` function to take a second argument (`na.rm`) with default value `FALSE` that removes `NA` values if `TRUE`.
3. Add checks to your function to make sure the input data is either numeric or logical. If it is logical convert it to numeric (Hint: look at the `stopifnot()` function).
4.

The diamonds data set is included in the `ggplot2` package (not by default in `R`). It can be read into your environment with the following function.

```
data("diamonds", package = "ggplot2")
```

Loop over the columns of the diamonds data set and apply your mean function to all of the numeric columns (Hint: look at the `class()` function).

0.4 Rmarkdown

1. Create a new Rmarkdown document.
2. Alter the template to specify the size to be a height of `6`.
3. Add a caption to your figure.
4. Add a chunk that includes your `my_mean()` function and calculate the mean of 100 randomly generated numbers (see `runif()` or `rnorm()`).
5. Compile your document to pdf.