Homework 1

For this homework you will create an R program and upload it to wolfware (.Rmd is ok if you know that already). Be sure that your R file has a header with your name and comments throughout. Note that a comment can be created in R using `#`.

The purpose of this homework is to get you some practice with basic R objects and some R functions.

To Do - Write an R script (or markdown file if you know that already) that corresponds with each step below.

Subsetting vectors

1) Create a vector `vec` with elements 2, 4, 6, . . . , 12.
2) Write code to print only the odd elements of vec to the console.
3) You can exclude elements from a vector by using a '`-`' sign in front of their indices. Write code to print the `vec` object without the first element and also the `vec` object without the first and third elements to the console.
4) Utilize the `length` function to print `vec` to the console without its last element.
5) You can reverse the order of elements in a vector using the `rev` function. Write code save the reversed `vec` object as `revVec`.
6) Print the first element of vec to the console 5 times (Hint: use the `rep` function to create your index).

Subsetting matrices

1) Create a matrix called 'unifMat' of random values between 0 and 1 (use `runif`). There should be 4 rows and 3 columns.
2) Overwrite the rows of the `unifMat` object as follows:

- For odd rows, replace the entries by two times the current entries
- For the even rows repalce the entries by one-half times the current entries.

3) Convert `vec` from earlier into a matrix by using `as.matrix` and call it `matVec`. Print the dimensions (`dim`) of `matVec` to the console.
4) When subsetting a matrix, there is an additional argument called `drop`. The default value of drop is `FALSE`. Try subsetting the first row of `unifMat` using `drop = TRUE` and `drop=FALSE` (put an extra comma in the `[`, `[1, , drop = FALSE]`). Explain the impact of using `drop` in a comment.

Subsetting data frames

1) The `iris` dataframe is a built-in R object. Write code to show three ways to access the 4th column of the `iris` dataset.
2) Add to the `iris` object by creating a new column (list element really) called `Extra.Col` whose entries are all 1's. Hint: `nrow` can give you the correct number of 1's to create with the `rep` function.
3) You can rearrange the order of columns by subsetting their colnames in the order that you want. For example,

```
iris[, c("Species","Sepal.Width","Petal.Width","Sepal.Length","Petal.Length", "Extra.Col")]
```

Use the built-in `mtcars` dataset. Write code to print the dataset out so that the columns are in alphabetical order (Hint: you can use the `sort` function to order the column names).

4) You can delete a column from a data frame by setting its value to `NULL`. Write code to overwrite the `iris` object with the `Sepal.Width` column removed.

5) This is a tough one! The `gsub` function allows you to search for a pattern of letters or symbols and replace it with a different character. Use this on the `iris` column names to change the all periods to underscores. The help for `gsub` gives the following info:

```
gsub(pattern, replacement, x, ignore.case = FALSE, perl = FALSE, fixed = FALSE, useBytes
= FALSE)
```

The first argument should be the pattern we want to find - here a '.'. Period is kind of a special symbol so we need to actually look for `\\.` so R knows we actually want to find the period. (\ is an escape character and we need to use two of them here.)

The second argument is what we want to replace the periods found with - here an '_'.

Lastly (since all other arguments have a default value), we need to give it the object to search through. In this case we want to look at the `colnames(iris)`.

Subsetting lists

1) Create a list `lst1` with 3 elements - `vec`, `revVec`, and `unifMat`. Name these elements `a`, `b`, and `c`. Show 2 ways to access the 2nd element of `lst1`.

2) Create a list `lst2` that contains `lst1` as its only element. How can you access 'b' by subsetting lst2? Write code to do so.

3) In a comment, describe what happens if you try to access an element of a list using single square brackets, [], instead of double brackets, [[]]? (Hint: use the `str` function on the result to explore!)

You are now ready to submit to wolfware. Nice!