

# Week 1 Assignments

*January 17, 2019*

Each solution should be placed in its own .R file, which should be emailed to me. You can run your code any which way you wish, but if you don't know how to source it, copying and pasting into the terminal will work.

## Problem 1

Write a function that calculates the mean of each numeric column in a data frame. The means should be stored in an atomic vector whose names are equal to the names in the corresponding columns of the data frame. Use `sapply` instead of explicit loops or `colMeans`.

Do note that `sapply`, when applied to a dataframe, treats each column as an individual element that the function is applied to.

Load the Iris dataset by type `data("iris")`. Test your function on this dataset, storing the resulting means in a new variable.

## Problem 2

Write a function that takes one parameter, `data`, that represents a matrix or data frame. This function should return a closure that has one parameter, an integer vector. This closure should return the corresponding columns of `data`. If any of the specified indices are out of bounds (see `?any`, which can help you here), your function should throw an exception (error) via the `stop` function, e.g., `stop("Index out of bounds.")`.

Test your functions using the Iris data set and the index vectors `1:3` and `c(5, 10)`.

In the global environment, modify the Iris data set by removing the first column. Again, test your closure using `1:3`. Did the result change?

## Problem 3

Write a function that takes 1 parameter, an integer vector. This function should have the same behavior as the closure in Problem 2, except that it works on the global dataset `iris` instead of the local variable `data`.

Load the Iris data set. Test your function using the same index vectors as in Problem 2.

In the global environment, modify the Iris data set by removing the first column. Again, test your function using `1:3`. Did the result change?