IST687 – Cleaning/munging Dataframes

Often, in data science, when you get a dataset, it is not in the exact format you want/need. So, you have to refine the dataset into something more useful - this is often called "data munging".

In this lab, you need to read in a dataset and work on that dataset (in a dataframe) so that it can be useful. Then, we will explore the distribution within the dataset.

Step 1: Create a function (named readStates) to read a CSV file into R

1. Note that you are to read a URL, not a file local to your computer.
2. The file is a dataset on state populations (within the United States).

The URL is:

http://www2.census.gov/programs-surveys/popest/tables/2010-2011/state/totals/nst-est2011-01.csv

Hint: google "read.csv" and "url" with respect to R commands

Step 2: Clean the dataframe

3. Note the issues that need to be fixed (removing columns, removing rows, changing column names).
4. Within your function, make sure there are 51 rows (one per state + the district of Columbia). Make sure there are only 5 columns with the columns having the following names (stateName, base2010, base2011,Jul2010, Jul2011).
5. Make sure the last four columns are numbers (i.e. not strings).

Step 3: Store and Explore the dataset

6. Store the dataset into a dataframe, called dfStates.
7. Test your dataframe by calculating the mean for the July2011 data, by doing:
   mean(dfStates$Jul2011)
   → you should get an answer of 6,109,645

Step 4: Find the state with the Highest Population

8. Based on the July2011 data, what is the population of the state with the highest population? What is the name of that state?
9. Sort the data, in increasing order, based on the July2011 data.

Step 5: Explore the distribution of the states

10. Write a function that takes two parameters. The first is a vector and the second is a number.
11. The function will return the percentage of the elements within the vector that is less than the same (i.e. the cumulative distribution below the value provided).
12. For example, if the vector had 5 elements (1,2,3,4,5), with 2 being the number passed into the function, the function would return 0.2 (since 20% of the numbers were below 2).
13. Test the function with the vector 'dfStates$Jul2011Num', and the mean of dfStates$Jul2011Num'.

There are many ways to write this function (described in #10 above) – so please try to write multiple versions of this function – which do you think is best?