Statistics 133 - Homework 9 part 2

The purpose of part 2 of this assignment is to give you experience data cleaning with command line tools. The data for this assignment come from the Gapminder Foundation, at `http://www.gapminder.org`. Gapminder is a non-profit organization that promotes visualization of data about global development topics, based on the work of Hans Rosling.

Keep track of the UNIX commands you use to do each of the following tasks (use history command). When you are done, enter your commands at the bottom of your .Rmd file for Assigment 9 part 1. You do not need to upload any other files.

Before you start, download the files `lifeexpectancy.csv` and `makemaps.R` from bCourses and temporarily put them on the desktop of whatever computer you are using. Note that `lifeexpectancy.csv` is a CSV (comma separated value) file that we could open in a program like Microsoft Excel, but we're going to manipulate it directly from the UNIX command line. You should have downloaded the Data Science Toolbox already for this assignment.

1. From the command line, create a new directory called `lifespan` somewhere within your home directory, or wherever you want to put it. Change your working directory to be this `lifespan` directory.

2. From the command line, move the downloaded files on the desktop into your `lifespan` directory. Use relative pathnames. When you are done, list the contents of `lifespan` to make sure that the files are there.

3. Look at the file `lifeexpectancy.csv` using `less`. Practice moving up and down in the file using the `SPACE` key to go down and `b` to go up. Here are some things to pay attention to as you look at this file:

   - Each line in the file is quite long and will wrap around in your terminal window; try resizing it to see what happens. Note that this file is essentially in matrix format: rows of the matrix are rows in the file, and columns are separated by commas.
   - The first line of the file is a header showing the years. The first column of this row is missing; see how it says ",1800" with nothing before the comma.
   - Each subsequent line in the file has a country name and then the data on life expectancy in that country for each of the years indicated in the first row. Missing data in this particular file is indicated by an NA, just like in `R`. There is a lot of missing data in this file! You will see some countries have no data at all, and other countries have data only for certain years.

   Type `q` when you are done to exit from `less`.

4. How many countries are represented in this file (i.e. how many lines are in the file) ? Write a `UNIX` command to find out. (Remember the top line of the file contains years, not country data, so we don't want to count it. For what to turn in, you can just write the `UNIX` command you'd use to get the total number from which you'd subtract one.)

5. Suppose we want to do some analysis for the years 1950, 1975, and 2000. Use `head` to look at *only the first line* of this file again.

Based on the fact that the first column is missing and subsequent columns start from 1980 and increase by one each time, figure out what column numbers correspond to 1950, 1975, and 2000. (You don't need to write a UNIX command to do this, just do the math.)

6. Write a UNIX command to keep only these columns of `life expectancy.csv`. For now, just print this to the screen. (*Hint: take the results of* `cat lifeexpectancy.csv` *and pipe it into* `cut`. In addition to the columns you found above, also be sure to keep the first column with the country names.) If you've done this step correctly, the first few lines of what gets printed to the screen should look like this:

```
,1950,1975,2000
Abkhazia,NA,NA,NA
Afghanistan,28.801,38.438,42.129
Akrotiri and Dhekelia,NA,NA,NA
Albania,55.23,68.93,75.651
Algeria,43.077,58.014,70.994
```

7. Create a clean data file that contains only the data for these years, *also removing any line that contains no data for these years.* (*Hint: Use the filter* `egrep "[0-9]"` *to keep lines that contain numbers.*) Carry out this step in one line of UNIX code, using pipes, and redirect the output into a new file called `lifeexpectancy.clean.csv`. Use `less` again to check that `lifeexpectancy.clean.csv` contains what you want. If not, go back and modify your last command. This is what the first few lines of my file look like:

```
,1950,1975,2000
Afghanistan,28.801,38.438,42.129
Albania,55.23,68.93,75.651
Algeria,43.077,58.014,70.994
Angola,30.015,39.483,41.003
Argentina,62.485,68.481,74.34
```

   Note how Abkhazia and Akrotiri and Dhekelia were both removed from the dataset because they had no data.

8. The file `makemaps.R` contains some code to visualize the data file you just created. You do not need to modify this file, but use `cat` to look at it, and notice three things:

   - the two packages that are needed; check to make sure both are installed on the machine you're using (Type R, to start up R in your DataScienceToolbox window, then type install.packages("maps") and install.packages("fields"). Type Ctrl D to exit R.)
   - the use of the `system` function to create a directory; `system` lets you run UNIX commands from within `R`.
   - the creation of plots using the `pdf` and `dev.off` functions; I need these because we're going to run `R` in `BATCH` mode and I won't be able to manually save the plots

   Still within the `lifespan` directory, run `R` in `BATCH` mode to execute this file. Use `cat` to look at the output file and check for any error messages. If you've done everything correctly so far, your `lifespan` directory should now contain a new subdirectory called `plots` with three `pdf` maps inside. Use `ls` to look at the contents of the `lifespan` directory again, then use `ls` to look at the contents of the `plots` directory, while still located in the `lifespan` directory.

If you want to look at the pdf plots you created copy them to the MyDataScienceToolbox folder in your home directory (type cp * /vagrant/.).

Extra Credit

A video showing some of Hans Rosling's work is at `http://www.youtube.com/watch?v=jbkSRLYSojo`. For a possible two bonus points (so max score of 12) on this assignment, watch this video enter your answers to the following on bCourses.

1. What is the primary plot type that Rosling uses in the video?
2. We could create something similar in `R` by first making a series of plots in separate files and then stringing them together to make an animation. The `UNIX` function `convert` can do this, as can the commercial software Quicktime Pro. If we wanted to do this, what additional data would we need, beyond what you used in this assignment?
3. Name three other techniques that Rosling uses to enhance his plots. Which of them do you think you could do in `R`?