Homework # 4

Attached are the following files

```
county_population.csv: population of each county in the US
covid_deaths.csv: number of deaths due to covid
                  split by date and county
county_facts.csv : demographic information by county
```

Also included, but not really part of the data, is `county_facts_dictionary.csv` which explains the columns of `county_facts.csv`.

In completing this homework, you should implement the data workflow described in the lecture. Separate the data into raw and processed. Define different scripts to munge the data, access the data, and analyze the data. Use Rmarkdown to create a pdf file. Include the code of each of your scripts in the Rmarkdown file, specifying which code was from which script. You should have the following R scripts,

```
data_munge.R
data.R
analysis.R
configuration.R
```

You can of course change the names of these scripts to suite your tastes; the point is that scripts should exist reflecting these tasks. Submit a pdf file as well as a zipped/compressed version of your workflow folder. BE SURE TO INCLUDE YOUR NAME IN YOUR ZIPPED FOLDER NAME.

1. The columns giving FIPS, county, and state values for the three datasets are not named consistently. Create new data.frame's using `write.csv` in which these columns are consistently named across the datasets. Do this in your `data_munge.R` file using R code, not by hand. (Hint: If `d` is a data.frame, then

   ```
   names(d)[c(1,3)] <- c("banana", "kiwi")
   ```

   will give columns 1 and 3 the column names banana and kiwi, respectively.) Save the altered csv files in a processed data folder as described in the lecture.

2. For your `data.R` script, write functions that return each of the data.frames. For example,

   ```
   get_county_facts_df()
   ```

   should read the processed county facts csv file and return a data.frame

3. Write a function `get_counties(d)` which returns a data.frame with a column county and a column state containing all counties, and their respective state, in the three datasets we are considering. The csv files of the three datasets have exactly the same counties, so you can use either of the three datasets to do this, and `d` can represent any of the three, depending on which you pick.

4. Write a function `map_counties(cd, cp, cf, demographic)` which returns a data.frame with columns county, state, deathRate, statistic. The data.frame should have a row for each county (and its respective state). For a given county, the deathRate is the per-capita death rate of the county over the whole range of days in the covid deaths dataset. For a given county, statistic is the value for that county of a particular demographic from the county facts dataset. The variable `demographic` is a particular column name in the `county_facts.csv` dataset representing a demongraphic. `cd, cp, cf` are the county death, county population, and county facts data.frames. For example, `map_counties(cd, cp, cf, "AGE135214")` would have as value the ”persons under 5 years, percent, 2014” (this is the meaning of the AGE135214 column in the county facts csv file as described in the county facts dictionary csv file). Hint: This is a mapping from the counties to the per-captia death rate (this code you have from the previous homework) and to a particular column in the county facts data.frame.

5. Pick 5 demographics that you find interesting. For each one, plot the death rate (y value) against the demographic value (x value) across all the counties. In each case use the lm function to apply a linear regression and compute the slope of the best fit line. Which of your demographics has the strongest correlation (i.e. the slope of the best fit line is the correlation) with death rate?