Making State Tibbles To Support Graphical Comparisons

Brandon Tao

9/21/2020

### 0. Setup

library(tidyverse)

State statistics are often available in csv files. Here we read previously produced Covid-19 state case and death count csv files. They have three September days to show short temporal trends.

### 1. Read csv files

Note that this project has Data and Docs directories to provide more organized storage. The data and documents are no long mixed with R scripts in a single directory. One small cost is needing to specify the Data directory as shown in the script below.

state\_pop <- read\_csv('Data/State\_Pop\_2019.csv')  
names(state\_pop) <- c('State','Postal','Fips','Pop')  
state\_pop <- select(state\_pop, Postal, Pop)  
  
cases <- read\_csv('Data/cases\_Sep6\_13\_20.csv')  
nam <- names(cases)  
nam[3] <- 'Postal'  
names(cases) <- nam  
  
deaths <- read\_csv('Data/deaths\_Sep6\_13\_20.csv')  
names(deaths) <- nam # same as above

### 2. Use group\_by and summarize to get state counts for chosen days

ST\_cases <- cases %>% group\_by(Postal)%>%  
 summarize(Sep6=sum(Sep\_06\_2020),  
 Sep13=sum(Sep\_13\_2020),   
 Sep20=sum(Sep\_20\_2020))  
  
  
ST\_deaths<- deaths %>% group\_by(Postal)%>%  
 summarize(Sep6=sum(Sep\_06\_2020),  
 Sep13=sum(Sep\_13\_2020),   
 Sep20=sum(Sep\_20\_2020))

### 3. Use left\_join and mutate to compute rates per 100000

ST\_case\_rates <- left\_join(ST\_cases,state\_pop,by='Postal') %>%   
 mutate(Sep6r=100000\*Sep6/Pop,  
 Sep13r=100000\*Sep13/Pop,  
 Sep20r=100000\*Sep20/Pop,  
 type=rep('Cases',nrow(ST\_cases)))  
  
ST\_death\_rates <- left\_join(ST\_deaths,state\_pop,by='Postal') %>%   
 mutate(Sep6r=100000\*Sep6/Pop,  
 Sep13r=100000\*Sep13/Pop,  
 Sep20r=100000\*Sep20/Pop,  
 type=rep('Deaths',nrow(ST\_cases)))  
  
# A quick look  
ST\_case\_rates

## # A tibble: 51 x 9  
## Postal Sep6 Sep13 Sep20 Pop Sep6r Sep13r Sep20r type   
## <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <chr>  
## 1 AK 5765 6278 6836 731545 788. 858. 934. Cases  
## 2 AL 132313 138753 144962 4903185 2699. 2830. 2956. Cases  
## 3 AR 65377 70217 75723 3017804 2166. 2327. 2509. Cases  
## 4 AZ 205765 208512 214018 7278717 2827. 2865. 2940. Cases  
## 5 CA 737650 760490 785501 39512223 1867. 1925. 1988. Cases  
## 6 CO 58984 61318 64857 5758736 1024. 1065. 1126. Cases  
## 7 CT 53365 54326 55527 3565287 1497. 1524. 1557. Cases  
## 8 DC 14279 14592 14955 705749 2023. 2068. 2119. Cases  
## 9 DE 18043 18849 19566 973764 1853. 1936. 2009. Cases  
## 10 FL 646350 663991 683754 21477737 3009. 3092. 3184. Cases  
## # ... with 41 more rows

ST\_death\_rates

## # A tibble: 51 x 9  
## Postal Sep6 Sep13 Sep20 Pop Sep6r Sep13r Sep20r type   
## <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <chr>   
## 1 AK 42 44 45 731545 5.74 6.01 6.15 Deaths  
## 2 AL 2273 2350 2437 4903185 46.4 47.9 49.7 Deaths  
## 3 AR 893 981 1181 3017804 29.6 32.5 39.1 Deaths  
## 4 AZ 5218 5320 5476 7278717 71.7 73.1 75.2 Deaths  
## 5 CA 13728 14385 15018 39512223 34.7 36.4 38.0 Deaths  
## 6 CO 1970 1988 2014 5758736 34.2 34.5 35.0 Deaths  
## 7 CT 4468 4480 4492 3565287 125. 126. 126. Deaths  
## 8 DC 611 616 620 705749 86.6 87.3 87.8 Deaths  
## 9 DE 609 615 621 973764 62.5 63.2 63.8 Deaths  
## 10 FL 11847 12605 13296 21477737 55.2 58.7 61.9 Deaths  
## # ... with 41 more rows

### 4. Save rate tibbles for use in graphics

Here we save the tibbles into an RData file. This is easy to load into working memory. Alternately we could write csv files and read them back in later.

save(ST\_case\_rates,ST\_death\_rates,file='Data/ST\_rates\_Sep6\_13\_20.RData')