All county nday daygap case and death rate time series

### Directions

Folder Contents:

The folder should have:  
two downloaded Usafacts csv files called cases and deaths and two RData workspaces called fixDates.RData and County\_and\_State\_2019\_Pop.RData

In the Setup section below, CHANGE the number of days (nday), and the daygap to your preference. For example to show two 28 day change arrows use:

nday <- 3  
daygap <- 28

To show a 48 week times series for same day of the week use:  
nday <- 48  
daygap <- 7

The specification provides regularly spaced days ending with the last day in the USAfacts files.

Consider the nday=3 daygap=28 example.  
If the last day in the file was the 400th column, integers to pick day columns would be 400-2\*28, 400-28, and 400.

If the case and death files ended on /03/31/2021/ the dates returned would be:  
Feb\_03\_2021, Mar\_03\_2021, Mar\_31\_2021.

The county counts in the USAfacts files are converted to rates per 100,000 using the County\_2019 population counts. They can be converted to other rates in the graphics scripts.

### Two ways to run RMD script.

Click on the Run menu near the top right and then click on Run All.

Alternatively click on knit to also produce a word document.

### What the script does for both cases and deaths files.

It reads Usafacts Covid-19 county case and death count csv files. For each file it extracts the analyst specified days, joins the 2019 county population data, restructures the data using pivot\_longer, computes rates and removes the no longer needed column.

The script saves two tibbles named caseRates and deathRates in in a named RData workspace. An example name is  
“Cnty rates days=3 gap=28 end\_Mar\_26\_2021.RData.” This suffices to to document the date selection.

### Next steps

The RData are typically copied and pasted into folders that address graphics production.

For county data the caseRates and deathRates tibbles produced are left in the pivot\_longer format. Then, in graphics or modeling scripts, it is easy to filter rows to choose states and dates. The use pivot\_wider produces the needed tibbles (or data.frames) for graphics or modeling. For states with many counties, class examples typically pick the counties with highest recent rates.

The tibbles have been to make time series plots. There is a North Dakota class example. For states with many counties it is reasonable to look at 25 or so times series on a page. The North Dakota example picks the counties with the highest rates on last day.

### 0. Setup

nday <- 3  
daygap <- 28  
  
library(tidyverse)

## -- Attaching packages ---------------------------------- tidyverse 1.3.0.9000 --

## v ggplot2 3.3.3 v purrr 0.3.4  
## v tibble 3.0.4 v dplyr 1.0.5  
## v tidyr 1.1.2 v stringr 1.4.0  
## v readr 1.4.0 v forcats 0.5.0

## -- Conflicts ------------------------------------------ tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

load('Data/fixDates.RData')  
load('Data/County\_and\_State\_2019\_Pop.RData')

### 1. Read csv files, pick column and rename dates.

The first 4 variables are county Fips codes and names, and state postal codes and fip codes.

The rest of the variables are USAfacts date column names that are not acceptable variable names in R.

The script below selects data columns using numbers and then provides acceptable column names. Accepable names start with a letter. They can contain letters and numbers but only two special characters, ‘.’ and ’\_’.  
For example, “Mar\_02\_2021” is an acceptable variable name. Subsequent graphics scripts convert such names into dates that are used in time series plots.

cases <- read\_csv(file='cases.csv')

##   
## -- Column specification --------------------------------------------------------  
## cols(  
## .default = col\_double(),  
## `County Name` = col\_character(),  
## State = col\_character()  
## )  
## i Use `spec()` for the full column specifications.

nams <- names(cases)  
lastvar <- length(nams)  
subs <- lastvar- daygap\*((nday-1):0)  
newnams <- c('Fips','County','Postal',  
 fixDates(nams[subs]))  
subs <- c(1:3,subs)  
cases <- cases[,subs]  
names(cases) <- newnams  
  
deaths <- read\_csv(file='deaths.csv')

##   
## -- Column specification --------------------------------------------------------  
## cols(  
## .default = col\_double(),  
## `County Name` = col\_character(),  
## State = col\_character()  
## )  
## i Use `spec()` for the full column specifications.

nams <- names(deaths)  
lastvar <- ncol(deaths)  
subs <- lastvar- daygap\*((nday-1):0)  
newnams <- c('Fips','County','Postal',  
 fixDates(nams[subs]))  
subs <- c(1:3,subs)  
deaths <- deaths[,subs]  
names(deaths) <- newnams  
lastDay<-newnams[length(newnams)]

### 2. Filter to remove unwanted rows

Filter rows:  
Omit based on Fips codes Fips = 2270. This Alaska region was rename and numbered  
Fips = 6000. This is a Cruise ship that docked in California Fips = 0. These rows have state counts are not attributed to counties

cases <- cases %>%  
 filter(Fips != 2270 &   
 Fips != 6000 & Fips > 999)  
  
deaths <- deaths %>%  
 filter(Fips != 2270 &   
 Fips != 6000 & Fips > 999)

### 3. Join the county population column

One column is called FIPS and the other Fips. Rename one of them so they match in the left\_join functions.

workPop <-select(County\_2019\_Pop,FIPS,Pop)  
names(workPop) <- c('Fips','Pop')  
cases <- left\_join(cases,workPop,by='Fips')  
deaths <- left\_join(deaths,workPop,by='Fips')

### 4. Pivot\_longer, Compute rates, Remove Pop, Counts

Only stack the day columns. The first 3 columns such as Fips, County, and Postal, and the last column,Pop, will recycle for every day.

When the rates are computed omit the Pop and Counts columns. They have served their purpose.

# Omit FIPS and Pop from stacking columns, they recycle   
  
nams <- names(cases)[4:(ncol(cases)-1)]  
tmp <- pivot\_longer(cases,all\_of(nams),  
 names\_to = "Days", values\_to ="Counts")  
  
cnty\_cRates<- tmp %>%   
 mutate(Rates=100000\*Counts/Pop) %>%  
 select(!c(Pop,Counts))  
  
# deaths  
nams <-names(deaths)[4:(ncol(cases)-1)]  
tmp <- pivot\_longer(deaths,all\_of(nams),  
 names\_to = "Days", values\_to ="Counts")  
  
cnty\_dRates<- tmp %>%   
 mutate(Rates=100000\*Counts/Pop) %>%  
 select(!c(Pop,Counts))

### 5. Save County case and death rate time series in a RData workspace.

save(cnty\_cRates,cnty\_dRates,  
file = paste0('Cnty rates days=',nday,' gap=',daygap,' end=',lastDay,'.RData'))