Turn Covid-19 Dates in R column names and Select variables

Brandon Tao

9/21/2020

The USAfacts Covid-19 case and death csv files provides a good resource for class room examples, homework assignments and possibly class projects.

The csv files pose a challenge because the first row has dates that are not allowed as R variable names. This script produces dates that are valid column names and selects few dates to provide smaller csv files for student use.

The important class topics are the often used data wrangling functions and graphics. The need to modify dates is relatively uncommon. This script is shown in class and provided for possible future reference. Modification of dates is not a quiz topic.

library(tidyverse)

## -- Attaching packages ----------------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.2 v purrr 0.3.4  
## v tibble 3.0.3 v dplyr 1.0.0  
## v tidyr 1.1.0 v stringr 1.4.0  
## v readr 1.3.1 v forcats 0.5.0

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

library(anytime)

### 1. Read the USAfacts US county cases and deaths csv files

The US county files are in the Data directory of this project.

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

## Parsed with column specification:  
## cols(  
## .default = col\_double(),  
## `County Name` = col\_character(),  
## State = col\_character()  
## )

## See spec(...) for full column specifications.

deaths <- read\_csv(  
 file='Data/deaths.csv')

## Parsed with column specification:  
## cols(  
## .default = col\_double(),  
## `County Name` = col\_character(),  
## State = col\_character()  
## )  
## See spec(...) for full column specifications.

#check <- names(cases)  
# check

The first two variable names, countyFIPs and “County Name”, are county identifiers. The next two names, state and stateFIPS, are state identifiers. The remaining column names are dates from 1/22/2020 to 9/20/2020.

In R, variable names must start with a letter and have only letters, numbers, an underscore, or a period.

The County Name has a blank that can be omitted.

The dates start with a number. How can this be fixed?

#### 2. Make valid variable names for data

We can fix date by putting a three letter month first and by separating the month, day and year with two underscores. Here we use ’\_’ but we could use ‘.’.

# Get the column names as vector  
tmp <- names(cases)  
  
# Fix the County Name in the first 4 names  
first4 <- tmp[1:4]   
first4[2] <- 'CountyName'  
  
# get the dates by removing first 4 names   
rest <- tmp[-(1:4)]  
rest[1:2] # look as the first two

## [1] "1/22/2020" "1/23/2020"

# put numbers in the standard year month day order   
rest <- anydate(rest)  
rest[1:2]

## [1] "2020-01-22" "2020-01-23"

# Change the format  
# Below the symbol %b means use the 3 letter month name.  
# The symbol %d means use the number.  
# The symbol %Y means use the 4 digit number.  
# the separators '\_' could be replaced by '.'.  
  
rest<- format(rest,"%b\_%d\_%Y") # Reformat '.'   
rest[1:2]

## [1] "Jan\_22\_2020" "Jan\_23\_2020"

names(cases) <- c(first4,rest) # rename all columns

Now we fix the deaths column names. Note, this script and edited copy of the case script above. It could be turned into a function. All that differs is the csv file name. This file namee would the function argument.

deaths <- read\_csv(file='Data/deaths.csv')

## Parsed with column specification:  
## cols(  
## .default = col\_double(),  
## `County Name` = col\_character(),  
## State = col\_character()  
## )

## See spec(...) for full column specifications.

tmp <- names(deaths)  
  
# Fix the County Name in the first for names  
first4 <- tmp[1:4]   
first4[2] <- 'CountyName'  
  
# get the dates  
rest <- tmp[-(1:4)]  
   
# put numbers in the standard year month day order   
rest <- anydate(rest)  
  
# Change the format  
# Below the symbol %b means use 3 letter month name.  
# The symbol %d means use the number  
# The symbol %Y means use the 4 digit number  
# the separators '\_' could be replaced by '.'.  
rest<- format(rest,"%b\_%d\_%Y") # Reformat '.'   
rest[1:2]

## [1] "Jan\_22\_2020" "Jan\_23\_2020"

names(deaths) <- c(first4,rest)

### 3. Select first four variables and the last 3 days that are 7 days apart

# names(cases)  
  
cases\_Sep6\_13\_20 <- select(cases, countyFIPS:stateFIPS,  
 Sep\_06\_2020, Sep\_13\_2020, Sep\_20\_2020)  
names(cases\_Sep6\_13\_20)

## [1] "countyFIPS" "CountyName" "State" "stateFIPS" "Sep\_06\_2020"  
## [6] "Sep\_13\_2020" "Sep\_20\_2020"

deaths\_Sep6\_13\_20 <- select(deaths, countyFIPS:stateFIPS,  
 Sep\_06\_2020, Sep\_13\_2020, Sep\_20\_2020)  
  
  
names(cases\_Sep6\_13\_20)

## [1] "countyFIPS" "CountyName" "State" "stateFIPS" "Sep\_06\_2020"  
## [6] "Sep\_13\_2020" "Sep\_20\_2020"

names(deaths\_Sep6\_13\_20)

## [1] "countyFIPS" "CountyName" "State" "stateFIPS" "Sep\_06\_2020"  
## [6] "Sep\_13\_2020" "Sep\_20\_2020"

View(cases\_Sep6\_13\_20)  
View(deaths\_Sep6\_13\_20)

### 4. Save the csv files in Data folder for class use

write\_csv(cases\_Sep6\_13\_20,path='Data/cases\_Sep6\_13\_20.csv')  
write\_csv(deaths\_Sep6\_13\_20,path='Data/deaths\_Sep6\_13\_20.csv')