Mapping

John M. Drake

May 23, 2022

Maps that georeference non-spatial quantitative information may be especially difficult for the casual observer to properly interpret. Maps of COVID-19 incidence were produced during the pandemic and used by many people for both public and personal decision-making. One study looked at how knowledge risk perception and behavioral intentions were influenced by six different maps of COVID-19 prevalence in the US on May 11, 2022.

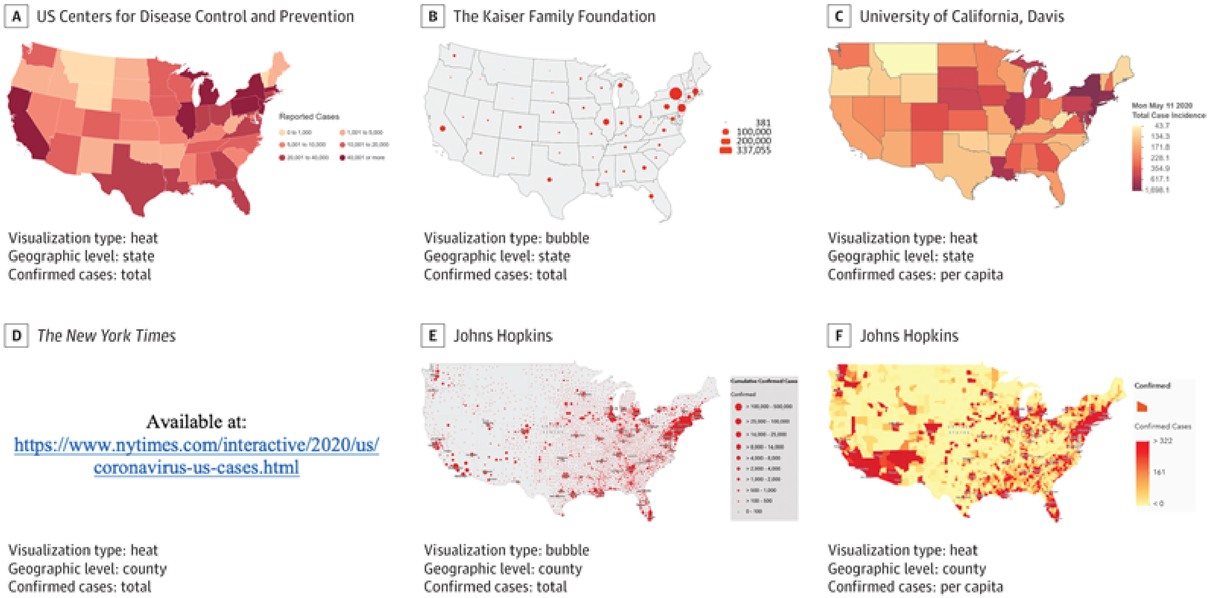


Figure 1 Six maps of COVID-19 prevalence in the US on May 11, 2020 used in a study assessing the effectiveness of different map features on the public understanding of the COVID-19 pandemic.

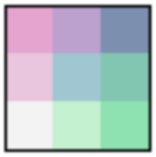
Some key findings of the study included:

- Maps of prevalance (per capita cases) translated poorly to knowledge about total cases.
- Choropleth maps were superior to bubble plots, perhaps because they de-emphasize the relative importance of large population centers and rural areas.
- Small areas with high prevalence are hard to indicate on national maps.
- Popular understanding was improved by aggregating information to the state level.
- Maps of per capita prevalence were more informative than maps of total cases.
- Viewing maps (versus not view maps) had no measurable impact on individual risk perception.
- Viewing maps (versus not view maps) made viewers slightly more optimistic about societal risk.
- Viewing maps (versus not view maps) had no influence on behavior.
- Viewing maps led to less accurate knowledge about total cases compared with having no information at all.

Perhaps most problematic is the difficulty distinguishing between places where transmission is high, indicated by high prevalence, versus where mortality is high, which might be locations with more vulnerable populations or where health systems are being compromised.

=======

Here is an interesting mapping trick! The following code uses a two dimensional (bivariate) color scale to map both variation in prevalence (pink) and variation in mortality (green). The pink and green combine to make a deeper blue color, so we can distinguish locations that are high in prevalance (pink), mortality (green), or both (blue).



Stevens, Josh. 2015. "Bivariate Choropleth Maps: A How-to Guide." Blog. Josh Stevens. https://www.joshuastevens.net/cartography/make-a-bivariate-choropleth-map/.

https://cran.r-project.org/web/packages/pals/vignettes/bivariate_choropleths.html

Exercise 1. Use this code to create a map showing both prevalence and mortality. Look closely at the map. What patterns do you see? Are there "hot spots"? What might explain the variation that you observe? Annotate the map to indicate any key features.

Exercise 2. Choose some different dates and plot the change in the two-color map as a series of small multiples. Do you observe any temporal trends?

Exercise 3. This map is based on government boundaries (counties) and is not a regular tiling. As a result, the *resolution* of the map differs from place to place. How does this affect your interpretation of these results? Are there any ways in which this variation may be misleading?

Covid Data

Download this file from dropbox, and place it in the `data` folder of your R Project.

https://www.dropbox.com/s/umv6887fzxozf6g/covid_data_20220519.rds?dl=0

Data

```
covid_data <- readRDS("../data/covid_data_20220519.rds") # edit path as needed

covid_county <- covid_data$county_dat # US county level data
```

Solution

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.1.1     v forcats 0.5.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
library(latticeExtra) # USCancerRates, mapplot
```

```
## Loading required package: lattice
```

```
##
## Attaching package: 'latticeExtra'
```

```
## The following object is masked from 'package:ggplot2':
##
##     layer
```

```r
library(maps) # map
```

```
##
## Attaching package: 'maps'
```

```
## The following object is masked from 'package:purrr':
##
##     map
```

```r
library(classInt) # classIntervals, findCols
library(grid) # viewport, pushViewport
library(stringr) # string substitutions
library(pals) # stevens.pinkgreen (a bivariate color palette)

mapdate <- "2020-05-18"

# Population data
county_metadata <- read_csv("../data/county_metadata.csv")
```

```
## Rows: 3143 Columns: 9
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (5): fips, state.fips, county.fips, state.name, county.name
## dbl (4): population.2020, mean.pop.center.lat, mean.pop.center.lon, HHS.region
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
# need a location column in the format "countyname,statename"
county_metadata <- county_metadata %>%
  mutate(location = paste0(tolower(state.name),",",tolower(county.name)))

covid_county <- covid_county %>%
  mutate(location = paste0(tolower(state),",",tolower(county)))

covid_county_wide <- covid_county %>%
  filter(date == mapdate) %>%
  pivot_wider(names_from = variable, values_from = value) %>%
  select(location, Total_Cases, Total_Deaths) %>%
  left_join(county_metadata, by="location") %>%
  mutate(Total_Cases_per100k = 100000 * Total_Cases / population.2020) %>%
  mutate(Total_Deaths_per100k = 100000 * Total_Deaths / population.2020)

# Colors
```

```
colorfunction <- stevens.pinkgreen # bivariate color function
nbins <- 3

breaks.cases <- classIntervals(na_if(log(covid_county_wide$Total_Cases_per100k),-Inf),
                               n=nbins,
                               style='equal') # equal breaks across range of data
```

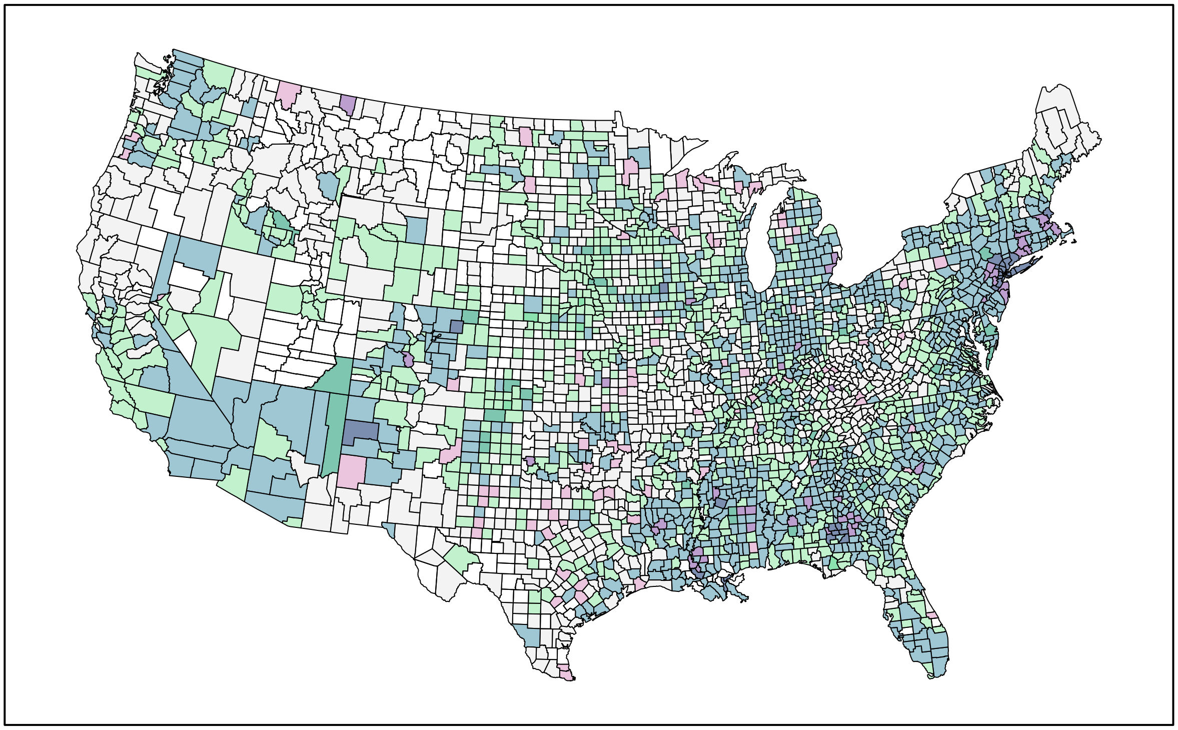
```
## Warning in classIntervals(na_if(log(covid_county_wide$Total_Cases_per100k), :
## var has missing values, omitted in finding classes
```

```
breaks.deaths <- classIntervals(na_if(log(covid_county_wide$Total_Deaths_per100k),-Inf),
                               n=nbins,
                               style='equal') # equal breaks across range of data
```

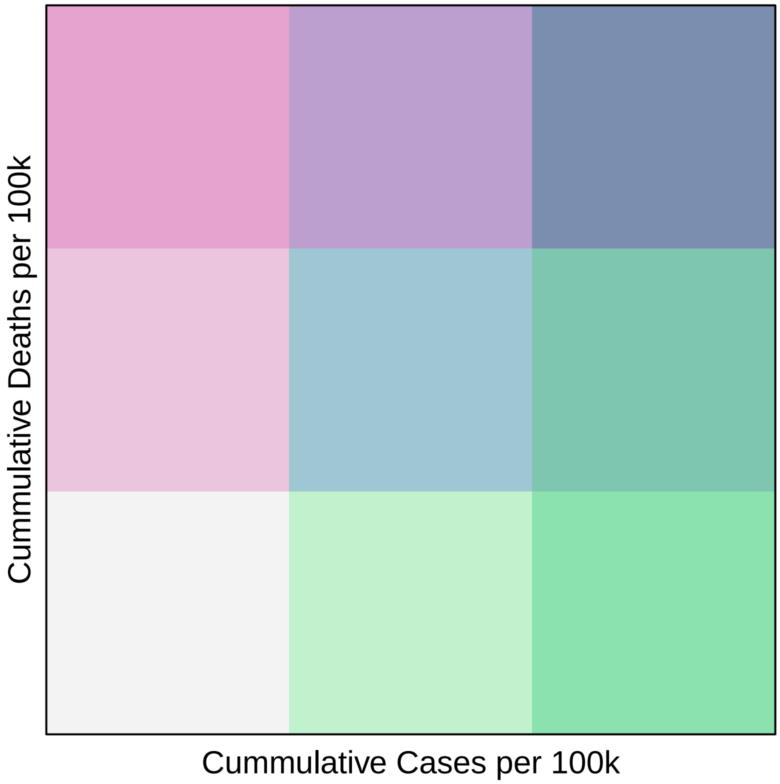
```
## Warning in classIntervals(na_if(log(covid_county_wide$Total_Deaths_per100k), :
## var has missing values, omitted in finding classes
```

```
c <- findCols(breaks.cases)
d <- findCols(breaks.deaths)
j <- replace_na(c,0) + replace_na(nbins*(d-1),0)
covid_county_wide$class.joint <- j %>% na_if(0)

covid_county_df <- as.data.frame(covid_county_wide)
# rownames(covid_county_df) <- covid_county_df$location

covidmap <- mapplot(location ~ class.joint, data = covid_county_wide,
              colramp=colorfunction,
              breaks=seq(from=0.5, by=1, length=nbins*nbins+1),
              xlab="",
              colorkey=FALSE,
              map = map("county", plot = FALSE, fill = TRUE,
                        projection = "tetra"),
              scales = list(draw = FALSE))

covidlegend <- levelplot(matrix(1:(nbins*nbins), nrow=nbins),
                         axes=FALSE, col.regions=colorfunction(),
                         xlab="Cummulative Cases per 100k", ylab="Cummulative Deaths per 100k",
                         cuts=8, colorkey=FALSE,
                         scales=list(draw=0))

par(mfrow=c(1,2))
covidmap
```

```
## Warning in (function (x, y, map, breaks, colramp, exact = FALSE, lwd = 0.5, :
## 109 unmatched regions: alabama,dekalb, alabama,st. clair, alaska,aleutians
## east....
```



```
covidlegend
```


Cummulative Deaths per 100k
Cummulative Cases per 100k