# Covid-19 Dataset Analysis

Seth Porter

2022-04-16

## Introduction

This report analyses a Johns Hopkins-sourced dataset to examine the correlations between COVID-19 impact and socio-economic metrics.

### Data Sources

The primary data source for this analysis is from the COVID-19 Data Repository by the Center for Systems Science and Engineering (CSSE) at Johns Hopkins University, as published on Github: https://github.com/CSSEGISandData/COVID-19. The original dataset was described in Dong, Du, and Gardner, "An interactive web-based dashboard to track COVID-19 in real time", The Lancet 20.5 (2020) P533-534.

We enrich this data with income, poverty and population data from secondary datasets:

- Small Area Income and Poverty Estimates (SAIPE) Program from the US Census https://www.census.gov/programs-surveys/saipe.html
- County population estimates from the US Census' Population Estimates Program (PEP) https://www.census.gov/programs-surveys/popest.html

Specific data products are referenced in the Data Import and Tidying section, following.

## Data Import and Tidying

This section covers our data sources and the process of importing, tidying, and joining them together. Readers uninterested in the details of the process may skip to the Analysis heading.

### Setup and Dependencies

First, we import a number of libraries we will need.

```r
library(conflicted)

# Tidyverse and sub-projects
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.0     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.1     v tibble    3.2.0
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.1
```

```
# if this fails, please install with `install.packages("tidyverse")` in the console

# The new fancy error messages breaks LaTeX rendering at least on
# my machine (it embeds an Escape char, \x1b, in the output for fancier
# rendering).
conflict_prefer("filter", "dplyr")
```

```
## [conflicted] Will prefer dplyr::filter over any other package.
```

```
conflict_prefer("lag", "dplyr")
```

```
## [conflicted] Will prefer dplyr::lag over any other package.
```

```
library(lubridate)
library(readxl)
library(stringr)

#conflicts_prefer(
#dplyr::filter(),
#dplyr::lag(),
#)

## https://cran.r-project.org/web/packages/ggcorrplot/readme/README.html
library(ggcorrplot)
# if this fails, please install with `install.packages("ggcorrplot")` in the console
```

## Daily Covid Reports: Import and Tidying

For a current snapshot, we use the `csse_covid_19_daily_reports` described in https://github.com/CSSEGISandData/COVID-19/tree/master/csse_covid_19_data#daily-reports-csse_covid_19_daily_reports; specifically the report from April 1, 2022 downloaded from https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_covid_19_daily_reports/04-01-2022.csv.

```
snapshot_url <- "https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/cs
daily_04_01_raw <- read_csv(snapshot_url, show_col_types = FALSE)
us_daily <- daily_04_01_raw %>%
  # Select only US records (FIPS is US-only)
  filter(!is.na(FIPS)) %>%
  # Prune certain cruise ships and non-state territories
  filter(!is.na(Admin2)) %>%
  select(c(FIPS, Admin2, Province_State, Confirmed, Deaths)) %>%
  mutate(
    FIPS=str_pad(FIPS, 5, pad="0")
  ) %>%
  rename(State=Province_State, County=Admin2)
```

Note that we only preserve Confirmed Cases and Deaths. The remaining metrics are either sporadically-updated (if at all), or in the case of `Incident_Rate` they are based on a population estimate and we will be supplying our own (below). This forces consistency (since Johns Hopkins does not provide a Death Rate metric, we cannot rely exclusively on their population computations).

## Poverty / Income: Import and Tidying

We use a US Census data product to provide median income and poverty-level estimates on a county level. This is a modeled dataset based on several indicators to provide a consistent product across the country.

- Overall docs https://www.census.gov/programs-surveys/saipe.html
- Specific product: "SAIPE State and County Estimates for 2020" https://www.census.gov/data/datasets/2020/demo/saipe/2020-state-and-county.html
- Field-level documentation https://www2.census.gov/programs-surveys/saipe/technical-documentation/file-layouts/state-county/2020-estimate-layout.txt
- Dataset URL https://www2.census.gov/programs-surveys/saipe/datasets/2020/2020-state-and-county/est20all.xls

Unlike the Covid data, separate FIPS fields are present for state and county, so must be combined into a single field before the datasets can be joined.

```
if(!file.exists("Data")){dir.create("Data")}
poverty_url <- "https://www2.census.gov/programs-surveys/saipe/datasets/2020/2020-state-and-county/est2
poverty_file <- "Data/saipe-est20all.xls"
if(!file.exists(poverty_file)) {download.file(poverty_url, poverty_file, mode="wb")}
poverty <- read_excel(poverty_file, range="A4:Y3199", .name_repair = "universal", ) %>%
  # Keep the various names for diagnostics
  select(c(State.FIPS.Code, County.FIPS.Code, Postal.Code, Name,
           Poverty.Estimate..All.Ages,
           Median.Household.Income,
           )) %>%
  mutate(Postal.Code=factor(Postal.Code),
         FIPS=paste0(State.FIPS.Code, County.FIPS.Code),
         # Kalawao County has "." for two fields, just allow them to coerce to NA.
         Poverty.Estimate..All.Ages=suppressWarnings(as.numeric(Poverty.Estimate..All.Ages)),
         Median.Household.Income=suppressWarnings(as.numeric(Median.Household.Income))) %>%
  rename(
    PovertyCount=Poverty.Estimate..All.Ages) %>%
  # Prune state-level aggregates
  filter(County.FIPS.Code != "000") %>%
  select(-c(State.FIPS.Code, County.FIPS.Code))
```

```
## New names:
## * 'State FIPS Code' -> 'State.FIPS.Code'
## * 'County FIPS Code' -> 'County.FIPS.Code'
## * 'Postal Code' -> 'Postal.Code'
## * 'Poverty Estimate, All Ages' -> 'Poverty.Estimate..All.Ages'
## * '90% CI Lower Bound' -> '..90..CI.Lower.Bound...6'
## * ‘90% CI Upper Bound‘ -> ‘..90..CI.Upper.Bound...7‘
## * ‘Poverty Percent, All Ages‘ -> ‘Poverty.Percent..All.Ages‘
## * ‘90% CI Lower Bound‘ -> ‘..90..CI.Lower.Bound...9‘
## * ‘90% CI Upper Bound‘ -> ‘..90..CI.Upper.Bound...10‘
## * ‘Poverty Estimate, Age 0-17‘ -> ‘Poverty.Estimate..Age.0.17‘
## * ‘90% CI Lower Bound‘ -> ‘..90..CI.Lower.Bound...12‘
## * ‘90% CI Upper Bound‘ -> ‘..90..CI.Upper.Bound...13‘
## * ‘Poverty Percent, Age 0-17‘ -> ‘Poverty.Percent..Age.0.17‘
## * ‘90% CI Lower Bound‘ -> ‘..90..CI.Lower.Bound...15‘
## * ‘90% CI Upper Bound‘ -> ‘..90..CI.Upper.Bound...16‘
## * ‘Poverty Estimate, Age 5-17 in Families‘ ->
##   ‘Poverty.Estimate..Age.5.17.in.Families‘
## * ‘90% CI Lower Bound‘ -> ‘..90..CI.Lower.Bound...18‘
## * ‘90% CI Upper Bound‘ -> ‘..90..CI.Upper.Bound...19‘
## * ‘Poverty Percent, Age 5-17 in Families‘ ->
##   ‘Poverty.Percent..Age.5.17.in.Families‘
## * ‘90% CI Lower Bound‘ -> ‘..90..CI.Lower.Bound...21‘
## * ‘90% CI Upper Bound‘ -> ‘..90..CI.Upper.Bound...22‘
## * ‘Median Household Income‘ -> ‘Median.Household.Income‘
## * ‘90% CI Lower Bound‘ -> ‘..90..CI.Lower.Bound...24‘
## * ‘90% CI Upper Bound‘ -> ‘..90..CI.Upper.Bound...25‘
```

Note that we retain only the Median Income and the count of people below the poverty line; we will derive per-capita values using a consistent population estimate, below.

We see NA values in this dataset; let us examine them:

```
poverty %>%
  filter(is.na(PovertyCount) | is.na(Median.Household.Income)) %>%
  select(c(Postal.Code, Name))
```

```
## # A tibble: 1 x 2
##   Postal.Code Name
##   <fct>       <chr>
## 1 HI          Kalawao County
```

This is the smallest county in the United States, per https://en.wikipedia.org/wiki/Kalawao_County,_Hawaii, and the second-smallest population, so it is plausible that the Census does not have a good estimate. We can safely drop this single datapoint:

```
poverty <- poverty %>% filter(!is.na(PovertyCount) &
                              !is.na(Median.Household.Income))
```

### Unmatched Records

Before proceeding with combined analysis, we must examine the records which cannot be joined, and confirm that it is acceptable to disregard them.

Considering first the records which are present in the Covid dataset but not the poverty dataset:

```
covid_not_poverty <- us_daily %>%
  left_join(poverty, by="FIPS") %>%
  filter(is.na(Median.Household.Income))
```

```r
covid_not_poverty_totals <- covid_not_poverty %>%
  summarize(total_deaths=sum(Deaths), total_cases=sum(Confirmed))
print(paste0("Covid records without poverty estimates:",
             nrow(covid_not_poverty), " of which:"))
```

```
## [1] "Covid records without poverty estimates:145 of which:"
```

```r
print(paste0("'Unassigned' records: ",
             nrow(covid_not_poverty %>% filter(County=="Unassigned"))))
```

```
## [1] "'Unassigned' records: 51"
```

```r
covid_not_poverty <- covid_not_poverty %>%
  filter(County != "Unassigned")
print(paste0("Out of state records: ",
             nrow(covid_not_poverty %>% filter(str_detect(County, "Out of")))))
```

```
## [1] "Out of state records: 15"
```

```r
covid_not_poverty <- covid_not_poverty %>%
  filter(!str_detect(County, "Out of"))
print(paste0("Puerto Rico records: ",
             nrow(covid_not_poverty %>% filter(State=="Puerto Rico"))))
```

```
## [1] "Puerto Rico records: 78"
```

```r
covid_not_poverty <- covid_not_poverty %>%
  filter(State != "Puerto Rico")
print(paste0(covid_not_poverty$County, covid_not_poverty$State))
```

```
## [1] "Valdez-CordovaAlaska"
```

```r
us_totals <- us_daily %>%
  summarize(total_deaths=sum(Deaths), total_cases=sum(Confirmed))
print(paste0("Unmatched death fraction: ",
             covid_not_poverty_totals$total_deaths / us_totals$total_deaths))
```

```
## [1] "Unmatched death fraction: 0.0141692186221685"
```

```r
print(paste0("Unmatched case fraction: ",
             covid_not_poverty_totals$total_cases / us_totals$total_cases))
```

```
## [1] "Unmatched case fraction: 0.0123800380497136"
```

We see that Of the 145 such records:

- each state has one "Unassigned" row for cases and deaths which could not be localized
- 15 states report out-of-state cases as well as internal ones

- Puerto Rico has 78 records which it is a shame to ignore (Puerto Rico is chronically forgotten!) but whose omission is perhaps understandable in a “state”-oriented dataset like SAIPE
- 1 record for Valdez-Cordova census Area in Alaska, which is reported in Wikipedia to have been abolished in 2019: https://en.wikipedia.org/wiki/Valdez%E2%80%93Cordova_Census_Area,_Alaska

In total, these records account for 1.2% each of total US deaths and confirmed cases, so it is reasonable to disregard them.

Considering the opposite, locations present in the poverty data but not the covid data:

```
us_daily %>%
  right_join(poverty, by="FIPS") %>%
  # County comes from the covid dataset
  filter(is.na(County)) %>%
  select(c(Postal.Code, Name)) %>%
  count(Postal.Code)
```

```
## # A tibble: 3 x 2
##   Postal.Code     n
##   <fct>       <int>
## 1 AK              1
## 2 MA              2
## 3 UT             22
```

We find one county each in Alaska, Hawaii, 2 in Massachusetts, and 22 in Utah. Per a Johns Hopkins FAQ https://coronavirus.jhu.edu/us-map-faq, “Utah is reporting county data somewhat differently than many other states. The larger-population counties are reporting confirmed cases and deaths at the county level. However, the smaller counties are banded together into county groups. This is in an effort to protect identities of individuals.”

Overall, this is a small number of records, primarily in low-population counties, and we will disregard them for this analysis.

```
us_daily_poverty <- us_daily %>%
  inner_join(poverty, by="FIPS") %>%
  select(-c(Postal.Code, Name))
```

## County Populations

The Johns Hopkins dataset only reports deaths as total counts, not scaled by population. To compare death rates across the country, we must join against a population data source. In this analysis we use the US Census Population and Housing Unit Estimates https://www.census.gov/programs-surveys/popest.html, in particular the 2021 vintage population estimates: https://www2.census.gov/programs-surveys/popest/datasets/2020-2021/counties/totals/co-est2021-alldata.csv which has fields described at https://www2.census.gov/programs-surveys/popest/technical-documentation/file-layouts/2020-2021/CO-EST2021-ALLDATA.pdf

```
pop_url <- "https://www2.census.gov/programs-surveys/popest/datasets/2020-2021/counties/totals/co-est202
raw_pop <- read_csv(pop_url, show_col_types=FALSE)
population <- raw_pop %>%
  # County-level summaries
  filter(SUMLEV=="050") %>%
  mutate(FIPS=paste0(STATE, COUNTY)) %>%
  select(c(FIPS, POPESTIMATE2021, STNAME, CTYNAME))
```

### Unmatched Population

We perform the same data-matching tests for population data as we did for poverty. Starting with records in the population dataset but not the joined Covid data:

```r
population %>%
  left_join(us_daily_poverty, by="FIPS") %>%
  filter(is.na(County)) %>%
  count(STNAME)
```

```
## # A tibble: 4 x 2
##   STNAME            n
##   <chr>         <int>
## 1 Alaska            1
## 2 Hawaii            1
## 3 Massachusetts     2
## 4 Utah             22
```

This is the now-familiar list of counties that provide aggregated Covid reporting.

Considering the rows in the joined Covid/Poverty dataset (that is, after discarding the Covid records not found in the poverty data):

```r
print(paste0("Unmatched Covid records: ", nrow(population %>%
  right_join(us_daily_poverty, by="FIPS") %>%
  filter(is.na(STNAME)))))
```

```
## [1] "Unmatched Covid records: 0"
```

there are no further exceptions, so we can safely join the population data.

```r
us_daily_full <- us_daily_poverty %>%
  left_join(population, by="FIPS") %>%
  select(-c(STNAME, CTYNAME)) %>%
  rename(Population=POPESTIMATE2021)
```

## Missing Data

# Analysis

## Per Capita Computations

Before beginning work, we will compute per-capita values for Cases, Deaths, and Poverty:

```r
us_daily_full <- us_daily_full %>%
  mutate(CasesPerCap = Confirmed / Population,
         DeathsPerCap = Deaths / Population,
         PovertyPerCap = PovertyCount / Population)
```

## Summary

Let us consider basic descriptive statistics for the joined dataset, focusing on the columns we will use in our analysis:

```
summary(us_daily_full %>%
          select(-c(FIPS, County, State, Confirmed, Deaths, PovertyCount)))
```

```
##  Median.Household.Income   Population       CasesPerCap      DeathsPerCap
##  Min.   : 22901           Min.   :     57   Min.   :0.0000   Min.   :0.000000
##  1st Qu.: 47740           1st Qu.:  10885   1st Qu.:0.2112   1st Qu.:0.002557
##  Median : 55102           Median :  25845   Median :0.2443   Median :0.003617
##  Mean   : 57406           Mean   : 106168   Mean   :0.2465   Mean   :0.003711
##  3rd Qu.: 63999           3rd Qu.:  68766   3rd Qu.:0.2768   3rd Qu.:0.004722
##  Max.   :160305           Max.   :9829544   Max.   :3.0000   Max.   :0.017544
##  PovertyPerCap
##  Min.   :0.02965
##  1st Qu.:0.09535
##  Median :0.12386
##  Mean   :0.13331
##  3rd Qu.:0.16128
##  Max.   :0.48992
```

These values seem plausible with the possible exception of the 3 cases-per-person maximum, which we can examine in more detail:

```
# Wrap this as a function; we want to do this a lot, but keep it explicit.
# I can't figure out how to take non-strings as "column" but that's okay.
filter_percentile <- function(data, column, min_quant, max_quant) {
  bounds = quantile(data[[column]], c(min_quant, max_quant))
  return(
    data %>% filter(
      (!!sym(column) > bounds[1]) &
      (!!sym(column) <= bounds[2]))
  )
}

us_daily_full %>% filter_percentile("CasesPerCap", 0.999, 1.0)
```

```
## # A tibble: 4 x 11
##   FIPS  County      State Confi~1 Deaths Pover~2 Media~3 Popul~4 Cases~5 Death~6
##   <chr> <chr>       <chr>   <dbl>  <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
## 1 02282 Yakutat     Alas~     539      5      79   68352     704   0.766 0.00710
## 2 13053 Chattahooc~ Geor~    7449     22    1222   51811    9048   0.823 0.00243
## 3 48127 Dimmit      Texas    5747     51    2501   42788    8473   0.678 0.00602
## 4 48301 Loving      Texas     171      1       7   97491      57   3     0.0175
## # ... with 1 more variable: PovertyPerCap <dbl>, and abbreviated variable names
## #   1: Confirmed, 2: PovertyCount, 3: Median.Household.Income, 4: Population,
## #   5: CasesPerCap, 6: DeathsPerCap
```

The extreme value, 3.0 Cases per capita, is in the lowest-population county in the country (Loving County, TX). The small denominator magnifies the underlying signal, which may be partly reinfection but also likely includes transient workers being diagnosed while in the county. This highlights a difference in definition between the "population of this county" and "people whose disease or death would be recorded in this county".

## Pair-wise Analysis

We begin by taking our two measures of Covid impact (cases and deaths), and our two socioeconomic indicators (median income and population below the poverty line), and comparing them.

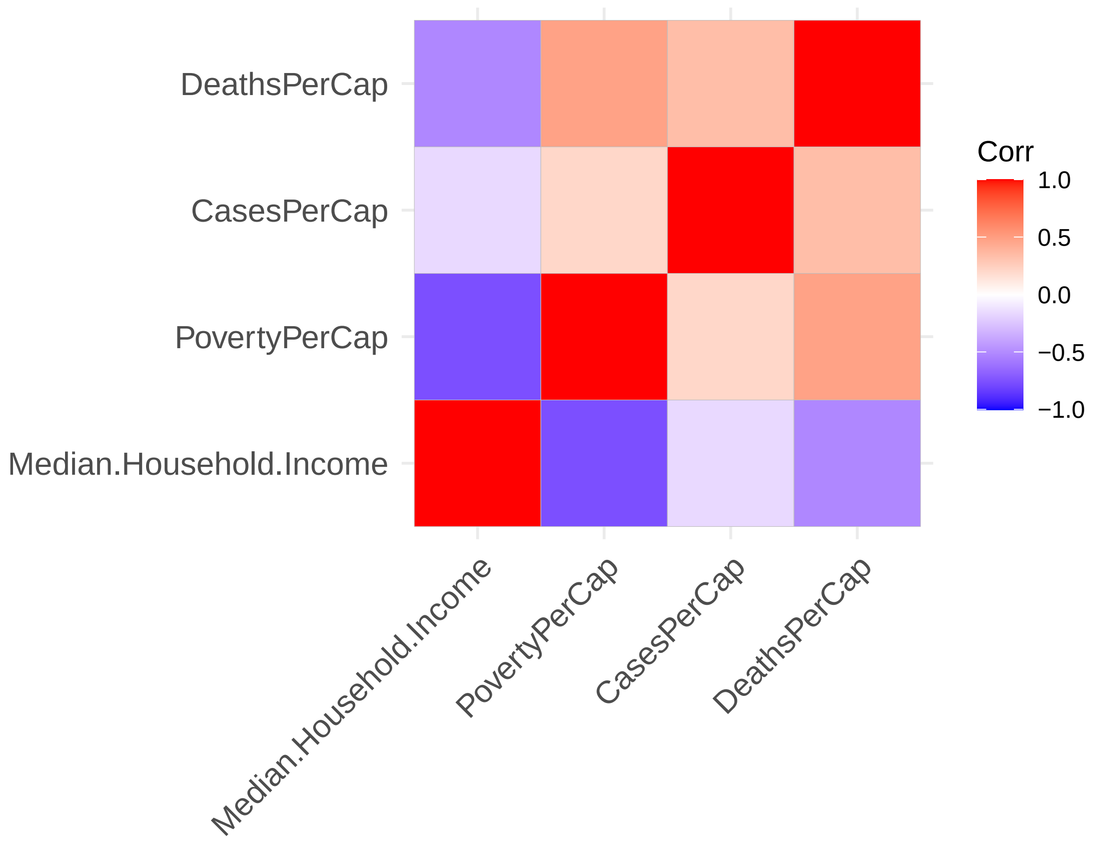
Considering the pairwise correlation matrix:

```r
pairwise <- us_daily_full %>%
  select(Median.Household.Income, PovertyPerCap, CasesPerCap, DeathsPerCap) %>%
  cor()
disp_pairwise <- round(pairwise, 1)

disp_pairwise
```

```
##                         Median.Household.Income PovertyPerCap CasesPerCap
## Median.Household.Income                     1.0          -0.8        -0.2
## PovertyPerCap                              -0.8           1.0         0.2
## CasesPerCap                                -0.2           0.2         1.0
## DeathsPerCap                               -0.5           0.5         0.3
##                         DeathsPerCap
## Median.Household.Income         -0.5
## PovertyPerCap                    0.5
## CasesPerCap                      0.3
## DeathsPerCap                     1.0
```

```r
ggcorrplot(pairwise)
```



Examining these correlations, we see that Poverty Per Capita and the median household income of a county are quite closely related (corr=-0.8).

Interestingly, Cases Per Capita is only weakly related to *any* of the factors including deaths. It is quite weakly correlated (in the expected way) with poverty and income. It is more correlated with Deaths per Capita (which seems reasonable, if a death is necessarily also a case), but only a correlation of 0.3.

By contrast, *Deaths* per capita is the best cross-category (disease and socioeconomic) correlation at 0.5 or -0.5, equally strong for either poverty or income.

### Poverty and Deaths

Choosing the (poverty, death) pair somewhat arbitrarily, we can fit a linear model:

```
deaths_from_poverty = lm(DeathsPerCap ~ PovertyPerCap, us_daily_full)
summary(deaths_from_poverty)
```

```
##
## Call:
## lm(formula = DeathsPerCap ~ PovertyPerCap, data = us_daily_full)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -0.0044799 -0.0009172 -0.0000626  0.0007792  0.0139913
##
## Coefficients:
```

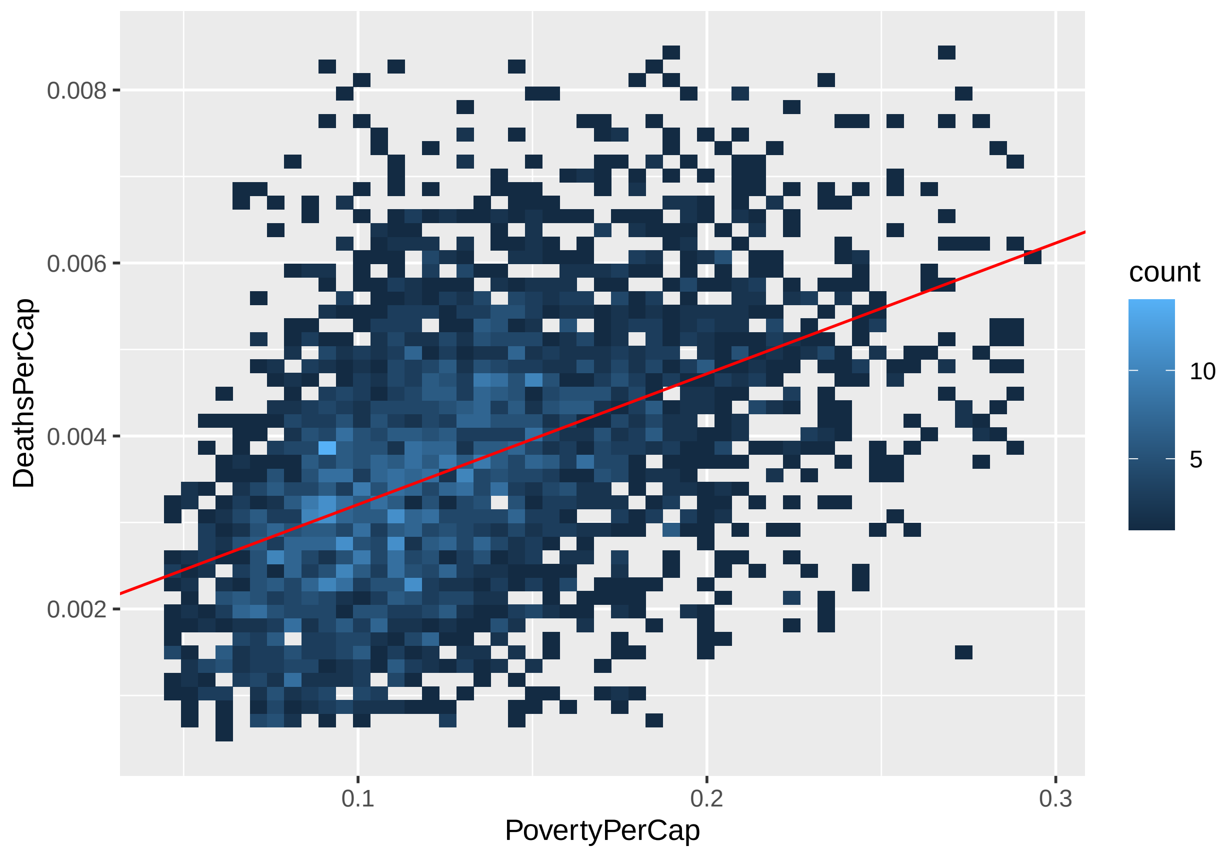
```
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)    1.696e-03  7.142e-05   23.75   <2e-16 ***
## PovertyPerCap 1.512e-02  4.980e-04   30.35   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.001469 on 3115 degrees of freedom
## Multiple R-squared:  0.2282, Adjusted R-squared:  0.228
## F-statistic: 921.1 on 1 and 3115 DF,  p-value: < 2.2e-16
```

```
paste0("Correlation: ",
       cor(us_daily_full$DeathsPerCap, us_daily_full$PovertyPerCap))
```

```
## [1] "Correlation: 0.477720197167518"
```

and plot a 2d histogram of these features against each other:

```
us_daily_full %>%
  filter_percentile("DeathsPerCap", 0.01, 0.99) %>%
  filter_percentile("PovertyPerCap", 0.01, 0.99) %>%
  ggplot(aes(x=PovertyPerCap, DeathsPerCap)) +
    geom_bin_2d(bins=50) +
    geom_abline(slope=coefficients(deaths_from_poverty)["PovertyPerCap"],
                intercept=coefficients(deaths_from_poverty)["(Intercept)"],
                color="red")
```



The slope appears too shallow, visually, but this is an illusion. Among other stability tests, the slope is almost the same if computed on the trimmed data, so the fit is not dominated by the extreme values.

### Many-way

Quite honestly, predicting death from poverty is likely the most practical use case, as poverty information may be known in advance and could be used to predict, whereas case counts are contemporaneous with deaths. However, in some scenarios (perhaps a loss of the ability to collect death data reliably) it could be useful to predict deaths from all three other features.

Fitting a simple linear model, predicting deaths as a sum of three linear components and an intercept:

```r
deaths_from_all = lm(DeathsPerCap ~ PovertyPerCap +
                       CasesPerCap + Median.Household.Income, us_daily_full)
summary(deaths_from_all)
```

```
##
## Call:
## lm(formula = DeathsPerCap ~ PovertyPerCap + CasesPerCap + Median.Household.Income,
##     data = us_daily_full)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -0.0052442 -0.0008756 -0.0001054  0.0007505  0.0105446
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)              4.140e-03  2.372e-04  17.455  < 2e-16 ***
## PovertyPerCap            4.733e-03  7.109e-04   6.657 3.28e-11 ***
## CasesPerCap              5.421e-03  3.294e-04  16.456  < 2e-16 ***
## Median.Household.Income -4.174e-08  2.550e-09 -16.368  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.001356 on 3113 degrees of freedom
## Multiple R-squared:  0.3422, Adjusted R-squared:  0.3416
## F-statistic: 539.9 on 3 and 3113 DF,  p-value: < 2.2e-16
```

we get R^2 up to 0.34 which is a mild improvement. The scales of the features are incommensurate so we cannot interpret coefficients directly as importance.

### Z-scoring for comparable feature scales

If we normalize the columns by z-scoring them we get coefficients which can be interpreted as feature importance:

```r
z_score <- function(column) {
  m = mean(column)
  s = sd(column)
  return((column-m)/s)
}
```

```r
z_scored <- us_daily_full %>%
  mutate(
    ZPoverty=z_score(PovertyPerCap),
    ZIncome=z_score(Median.Household.Income),
    ZCases=z_score(CasesPerCap),
    ZDeaths=z_score(DeathsPerCap)
  )
deaths_from_z = lm(ZDeaths ~ ZPoverty +
                       ZCases + ZIncome, z_scored)
summary(deaths_from_z)
```

```
##
## Call:
## lm(formula = ZDeaths ~ ZPoverty + ZCases + ZIncome, data = z_scored)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.1374 -0.5239 -0.0631  0.4490  6.3085
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.335e-17  1.453e-02   0.000        1
## ZPoverty     1.496e-01  2.247e-02   6.657 3.28e-11 ***
## ZCases       2.449e-01  1.488e-02  16.456  < 2e-16 ***
## ZIncome     -3.641e-01  2.224e-02 -16.368  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8114 on 3113 degrees of freedom
## Multiple R-squared:  0.3422, Adjusted R-squared:  0.3416
## F-statistic: 539.9 on 3 and 3113 DF,  p-value: < 2.2e-16
```

```r
print("Coefficients rescaled to deaths-per-capita:")
```

```
## [1] "Coefficients rescaled to deaths-per-capita:"
```

```r
print(sd(us_daily_full$DeathsPerCap) * coefficients(deaths_from_z))
```

```
##   (Intercept)      ZPoverty        ZCases       ZIncome
##  8.917251e-20  2.500072e-04  4.093647e-04 -6.085546e-04
```

```r
print(paste0("Std Dev of income: ", sd(us_daily_full$Median.Household.Income)))
```

```
## [1] "Std Dev of income: 14580.5339991142"
```

We see that an increase of one standard deviation in median household income, or $14,580, in a county is associated with a decrease of 0.0006 deaths-per-capita, while a one standard deviation increase in cases per capita leads to a slightly smaller increase of 0.0004.

## Conclusions

### Primary Conclusions

Based on these three datasets, it appears that the frequency of Covid cases is largely independent of socioeconomic factors, but the *outcomes* of those infections, in terms of deaths, are highly influenced by poverty. In fact the level of poverty is more influential than the number of cases in a county!

### Bias and Errors

There are many ways that these conclusions could be incorrect, in both the underlying data and the analysis presented here.

- There may be systematic variation in the fidelity of case or death reporting in different areas; note that high rates of reporting would confound with high underlying signal.
- The population and socioeconomic data are not perfectly aligned in time with the case data, and might in some cases reflect the consequences of the disease data rather than causes.
- There are modeling components in both the population and the socioeconomic metrics; assumptions from that level might bleed through and appear as a signal in this analysis.
- Considering only the accumulated case and death counts at a single point in time allows a simpler analysis, but ignores differences in the rates and trends over time. Cases concentrated in spikes can have a sharply worse impact than the same total load distributed over time; aggregate case or death count does not reflect that temporal distribution.
- This analysis discarded several categories of unattributed or aggregated case and death counts. Those were small fractions of the overall totals, but could be systematically concentrated in certain areas. Locally this could lead to an incorrect conclusion about a particular county or area.

All of these scenarios are likely true to at least some degree; the question is whether any of them is so common in the data that it leads us to incorrect conclusions or some systemic error in our models. Future analysis could rule many of these out, by considering stability across multiple snapshots, testing with other data sources, and otherwise reducing the risk that these conclusions are due to some one-off occurrence in this particular dataset.

## Appendix: Session Info

```
sessionInfo()
```

```
## R version 4.2.2 (2022-10-31 ucrt)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 23403)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=English_United States.utf8
## [2] LC_CTYPE=English_United States.utf8
## [3] LC_MONETARY=English_United States.utf8
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.utf8
```

```
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] ggcorrplot_0.1.4 readxl_1.4.2     lubridate_1.9.2  forcats_1.0.0
##  [5] stringr_1.5.0    dplyr_1.1.0      purrr_1.0.1      readr_2.1.4
##  [9] tidyr_1.3.0      tibble_3.2.0     ggplot2_3.4.1    tidyverse_2.0.0
## [13] conflicted_1.2.0
##
## loaded via a namespace (and not attached):
##  [1] tidyselect_1.2.0 xfun_0.37        reshape2_1.4.4   colorspace_2.1-0
##  [5] vctrs_0.5.2      generics_0.1.3   htmltools_0.5.4  yaml_2.3.7
##  [9] utf8_1.2.3       rlang_1.0.6      pillar_1.8.1     glue_1.6.2
## [13] withr_2.5.0      bit64_4.0.5      lifecycle_1.0.3  plyr_1.8.8
## [17] munsell_0.5.0    gtable_0.3.1     cellranger_1.1.0 memoise_2.0.1
## [21] evaluate_0.20    labeling_0.4.2   knitr_1.42       tzdb_0.3.0
## [25] fastmap_1.1.1    parallel_4.2.2   curl_5.0.0       fansi_1.0.4
## [29] highr_0.10       Rcpp_1.0.10      scales_1.2.1     cachem_1.0.7
## [33] vroom_1.6.1      farver_2.1.1     bit_4.0.5        hms_1.1.2
## [37] digest_0.6.31    stringi_1.7.12   grid_4.2.2       cli_3.6.0
## [41] tools_4.2.2      magrittr_2.0.3   crayon_1.5.2     pkgconfig_2.0.3
## [45] ellipsis_0.3.2   timechange_0.2.0 rmarkdown_2.20   rematch_1.0.1
## [49] rstudioapi_0.14  R6_2.5.1         compiler_4.2.2
```