Coronavirus Second Waves

Tyler DeGroff

Import New York Times Coronavirus Data

```
setwd("/Users/tylerdegroff/Documents/Github/NYTimes COVID-19 Data")

nytimes <- read.csv(
  file = "us-counties.csv",
  colClasses = c("Date", "factor", "factor", "character", "numeric", "numeric")
)

setwd("/Users/tylerdegroff/Documents/Github/Coronavirus")
```

Source: New York Times, The, Smith, M., Yourish, K., Almukhtar, S., Collins, K., Ivory, D., & Harmon, A. (2020, January–July). *Coronavirus (Covid-19) Data in the United States* [Cumulative counts of coronavirus cases in the United States, at the county level, over time (daily frequency).]. Github. https://raw.githubusercontent.com/nytimes/covid-19-data/master/us-counties.csv

```
data <- nytimes[nytimes$state %in% c(
  "Alabama",              "Alaska",               "Arizona",
  "Arkansas",             "California",           "Colorado",
  "Connecticut",          "Delaware",             "District of Columbia",
  "Florida",              "Georgia",              "Hawaii",
  "Idaho",                "Illinois",             "Indiana",
  "Iowa",                 "Kansas",               "Kentucky",
  "Louisiana",            "Maine",                "Maryland",
  "Massachusetts",        "Michigan",             "Minnesota",
  "Mississippi",          "Missouri",             "Montana",
  "Nebraska",             "Nevada",               "New Hampshire",
  "New Jersey",           "New Mexico",           "New York",
  "North Carolina",       "North Dakota",         "Ohio",
  "Oklahoma",             "Oregon",               "Pennsylvania",
  "Rhode Island",         "South Carolina",       "South Dakota",
  "Tennessee",            "Texas",                "Utah",
  "Vermont",              "Virginia",             "Washington",
  "West Virginia",        "Wisconsin",            "Wyoming"
), ]

data <- droplevels(data)

data <- data %>%
  within(., fips[county == "New York City"] <- 36999) # treats NYC as a county
```

Import Census Bureau FIPS Code Metadata

```
fips.state <- read_excel("fips.state.xlsx", skip = 4)
```

Source: United States Census Bureau. (2020, May 20). *2019 Census Bureau Region and Division Codes and State FIPS Codes* [Reference file for vintage 2019 Census Bureau state-level FIPS codes.]. United States Department of Commerce. https://www2.census.gov/programs-surveys/popest/geographies/2019/state-geocodes-v2019.xlsx

```
fips.state <- fips.state %>%
  rename(fips.state = "State (FIPS)", state = "Name") %>%
  select(fips.state, state) %>%
  filter(fips.state != "00") %>%
  arrange(fips.state)
```

```
fips.granular <- read_excel("fips.granular.xlsx", skip = 4)
```

Source United States Census Bureau. (2020, May 20). *2019 State, County, Minor Civil Division, and Incorporated Place FIPS Codes* [Reference file for vintage 2019 Census Bureau county-level FIPS codes.]. United States Department of Commerce. https://www2.census.gov/programs-surveys/popest/geographies/2019/all-geocodes-v2019.xlsx

```
fips.county <- fips.granular %>%
  rename(
    fips.state = "State Code (FIPS)",
    fips.county = "County Code (FIPS)",
    county = "Area Name (including legal/statistical area description)"
  ) %>%
  mutate(fips = as.numeric(paste0(
    as.character(fips.state),
    as.character(fips.county)
  ))) %>%
  filter(fips.county != "000") %>%
  select(fips, fips.state, fips.county, county)
```

```
fips <- merge(x = fips.county, y = fips.state, by = "fips.state", all.x = TRUE)
fips <- fips[complete.cases(fips[, "state"]), ]
```

Import Census Bureau Population Estimates

```
pop <- read_excel("pop.xlsx", skip = 3)
```

```
## New names:
## * `` -> ...1
```

Source: United States Census Bureau. (2010–2019, April 1–July 1). *County Population Totals: 2010-2019* [Annual estimates of the county-level resident population, over time (annual frequency).]. United States Department of Commerce. https://www2.census.gov/programs-surveys/popest/tables/2010-2019/counties/totals/co-est2019-annres.xlsx

```
pop <- pop %>%
  rename(countyState = "...1", population = "2019") %>%
  filter(countyState != "United States") %>%
  select(countyState, population)
```

```
fips <- fips %>% mutate(countyState = paste0(county, ", ", state))
pop <- merge(x = pop, y = fips, by = "countyState", all.x = TRUE)
```

Aggregate and Treat NYC as Its Own County

```
pop.nyc <- pop %>%
  filter(
    fips == 36005 | # Bronx County (Bronx)
    fips == 36047 | # Kings County (Brooklyn)
    fips == 36061 | # New York County (Manhattan)
    fips == 36081 | # Queens County (Queens)
    fips == 36085   # Richmond County (Staten Island)
  )

pop <- rbind(
  pop,
  data.frame(
    fips.state = "36",    # actual New York State state-level FIPS code
    fips.county = "999", # synthetic county-level FIPS code
    fips = "36999",      # synthetic FIPS code
    county = "New York City",
    state = "New York",
    countyState = "New York, New York",
    population = sum(pop.nyc$population)
  )
)
```

```
data <- merge(
  x = data,
  y = pop %>% select(fips, population),
  by = "fips",
  all.x = TRUE
)

data <- data %>%
  mutate(
    cases.percap = cases / population,
    deaths.percap = deaths / population
  )

data <- data[order(data$date, data$state, data$county), ]

data <- data %>%

  # mutate across dates by unique county/state combinations

  mutate(countyState = paste0(county, ", ", state)) %>%
  group_by(countyState) %>%

  mutate(
    cases.new = c(cases[1], diff(cases)),
    deaths.new = c(deaths[1], diff(deaths)),

    cases.new.percap = c(cases.percap[1], diff(cases.percap)),
    deaths.new.percap = c(deaths.percap[1], diff(deaths.percap)),

    cases.new.perM = cases.new.percap * 1000000,
```

```
    deaths.new.perM = deaths.new.percap * 1000000
  ) %>%

  mutate(
    cases.new.7dsma = rollmean(cases.new, k = 7, fill = 0, align = "right"),
    deaths.new.7dsma = rollmean(deaths.new, k = 7, fill = 0, align = "right"),

    cases.new.percap.7dsma =
      rollmean(cases.new.percap, k = 7, fill = 0, align = "right"),
    deaths.new.percap.7dsma =
      rollmean(deaths.new.percap, k = 7, fill = 0, align = "right"),

    cases.new.perM.7dsma =
      rollmean(cases.new.perM, k = 7, fill = 0, align = "right"),
    deaths.new.perM.7dsma =
      rollmean(deaths.new.perM, k = 7, fill = 0, align = "right"),
  ) %>%

  mutate(
    cases.active = rollsum(cases.new, k = 9, fill = 0, align = "right"),
    cases.active.percap = rollsum(cases.new.percap, k = 9, fill = 0, align = "right"),
    cases.active.perM = rollsum(cases.new.perM, k = 9, fill = 0, align = "right")
  )

write_csv(data, "data.csv")
```

Binomial Probability Distriubtion: Cumulative Density Function

```
x = 0
n = 100
p = as.numeric(data[
  data$state == "New York" & data$county == "Westchester" & data$date == max(data$date),
  "cases.active.perM"
])
```

```
p
```

```
## [1] 5878
```

$$1 - P\left(q > x | n, prob\right) = 1 - \binom{n}{x} \cdot prob^{x} \cdot (1 - prob)^{(n-x)} \tag{1}$$

$$1 - P\left(q > 0 | n = 100, prob = 5878 \cdot 10^{-6}\right) = 1 - \frac{100!}{0!\,(100-0)!} \cdot \left(5878 \cdot 10^{-6}\right)^{0} \cdot \left(1 - 5878 \cdot 10^{-6}\right)^{(100-0)} \tag{2}$$

```
cdf <- data.frame()
for (i in c(1, 5, 10, 25, 100, 250, 500, 1000, 2500, 5000, 10000)) {

  temp <- data.frame(
    n = i,
    prob = pbinom(q = x, size = i, prob = p * 10^-6, lower.tail = FALSE) * 100
  )

  cdf <- rbind(cdf, temp)

}
```

Table 1: Probability of Exposure to AT LEAST n Actively Contageous Persons. Assumes 9-Day Infectious Period

n	prob
1	0.6
5	2.9
10	5.7
25	13.7
100	44.5
250	77.1
500	94.8
1000	99.7
2500	100.0
5000	100.0
10000	100.0