The Rose Garden Event:
A Hierarchical Bayesian Approach to Modeling Positive Coronavirus Cases

Jedidiah Harwood$^1$ Eric A. Suess$^2$

$^1$Department of Statistics
University of California, Davis

$^2$Department of Statistics and Biostatistics
California State University, East Bay

August 10, 2021
Motivation

- President Trump’s Rose Garden Event for Supreme Court nominee Amy Coney Barrett, Sept. 26, 2020

- The Rose Garden event:
  1. Approximately 300 attendees
  2. Every guest tested negative, no social distancing
  3. Multiple, subsequent, coronavirus cases

- After the event numerous people tested positive. How can we explain this?
  1. There is a difference between testing positive and someone “truly” having the disease, \( D = 1 \)
  2. Diagnostic tests are imperfect
Implementation

- Built a Bayesian hierarchical model to estimate the number of people who would have tested positive after each event and over all the events.

- Posterior estimation:
  1. MCMC algorithm: Gibbs Sampler.

- R packages:
  1. rjags
  2. runjags
Our model is based on the test results of the individuals attending. We do not have these data. The data was simulated using reasonable values for the prevalence of the disease, sensitivity and specificity of the tests for each event.

To apply our model, we:

1. Collected data on 73 different Trump events (including the Rose Garden event, 9/26/20)
2. *Simulated* the number of positive results for each event

Generated posterior distributions for the model parameters.

Generated posterior predictive distributions using the simulated data.
President Trump’s superspreader events

- From news reports we found 70 such events.

```r
summary(covid19trump[, 1:2])
```

<table>
<thead>
<tr>
<th>Event_Type</th>
<th>Number_of_Participants</th>
</tr>
</thead>
<tbody>
<tr>
<td>Party</td>
<td>Min. : 110</td>
</tr>
<tr>
<td>Rally</td>
<td>1st Qu.: 1200</td>
</tr>
<tr>
<td>Airport</td>
<td>Median : 5000</td>
</tr>
<tr>
<td></td>
<td>Mean : 6117</td>
</tr>
<tr>
<td></td>
<td>3rd Qu.: 7000</td>
</tr>
<tr>
<td></td>
<td>Max. : 30000</td>
</tr>
</tbody>
</table>
Simulating the Number of Positive Cases

- To simulate the number of positive cases:
  1. Gave each event its own randomly determined probability of testing positive from a beta distribution
  2. Used each individual event’s probability of testing positive and size, to randomly generate the number of people who would have tested positive
Simulating the Number of Positive Cases

\[ k <- \text{nrow(na.omit(covid19trump))} \quad \# \text{Number of Events} \]

\[ \text{omega} <- 0.05 \quad \# \text{Mode of the beta Prior} \]

\[ \text{kappa} <- 100 \quad \# \text{Concentration for beta Prior} \]

\[ \text{prob\_nu} <- \text{rbeta(k, omega*(kappa - 2) + 1, (1 - omega)*(kappa - 2) + 1)} \quad \# \text{P(+)} \]

\# Loading different event sizes
\[ n <- \text{na.omit(covid19trump$Number\_of\_Participants)} \]
Simulating the Number of Positive Cases

```r
head(covid19trump[, 1:2])

## Event_Type  Number_of_Participants
## 1  Party         500
## 2  Party         300
## 3  Party         400
## 4  Party         900
## 5  Party        1500
## 6  Party         300

# Simulating the number of positive cases
positive_cases_sim <- function(x){
  rbinom(1, x, prob_nu)
}
positive_cases_sim(covid19trump$Number_of_Participants[1])

## [1] 23
```
What is the Model?

Figure 1: Model Diagram
What is the Model?

- $y_i$ = The number of people who have tested positive for coronavirus, at each event, $i$.

- $y_i$ was modeled using a binomial distribution:

  $$y_i \sim \text{binomial}(\nu_i, N_{total_i})$$
What is the Model?

- $\eta =$ The sensitivity of the coronavirus tests used.
  \[
  \eta = P(+) | D
  \]

- $\theta =$ The specificity of the coronavirus tests used.
  \[
  \theta = P(-) | D^c
  \]

- priors
  \[
  \eta \sim beta(910, 90) \\
  \theta \sim beta(950, 50)
  \]
What is the Model?

- $\pi_i =$ The **true** prevalence of the coronavirus at each event, $i$.

$$\pi_i = P(D|\omega, \kappa)$$

$$\pi_i \sim beta(\omega \star (\kappa - 2) + 1, (1 - \omega) \star (\kappa - 2) + 1)$$
What is the Model?

- $\nu_i = \text{The probability of testing positive for coronavirus, at each event, } i.$

\[
\nu_i = P(+|\eta, \theta, \pi_i) = \eta \pi_i + (1 - \theta)(1 - \pi_i)
\]

- For the simulated data, we set $\eta = \theta = 0.95$

- $N_{total_i} = \text{The number of participants at each event, } i.$

- **Note:** The parameter $\nu_i$ depends on $\eta$ and $\theta$
What is the Model?

- $\omega =$ The mode of the beta prior distribution put upon the prevalence of the coronavirus.

- $\omega$ is used in the model for an alternative parameterization of the beta distribution, rather than use of $\beta$ or $\alpha$.

- prior:

$$\omega \sim beta(6, 95)$$
What is the Model?

- $\kappa = \text{The concentration of the beta prior distribution put upon the prevalence of the coronavirus.}$

$\kappa$ is used in the model for an alternative parameterization of the beta distribution as well.

$$\kappa = (\kappa - 2) + 2$$

- prior:

$$\kappa - 2 \sim \text{gamma}(5.8, 0.48)$$
Posterior Predictive Distributions

- \( Py_i = \) The posterior, predictive distribution for the number of people who tested positive for coronavirus, at each event, \( i \).

- \( Py_i \sim \text{binomial}(\nu_i, N_{total_i}) \)

- \( Py_{tot} = \) The posterior, predictive distribution for the total number of people who tested positive for coronavirus throughout all the events.

- \[ Py_{tot} = \sum_{i=1}^{N} Py_i \]

- While everyone who attended the Rose Garden event tested negative, because of the imperfect diagnostic test used there were some people with a false negative test results.
Model Limitations

- Assumes that the same test was used for every event.
- Assumes that the underlying prevalence distribution for the coronavirus is the same for every event.
Implementation: JAGS code

modelString <- "
model {
  for (i in 1:k){
    y[i] ~ dbin( nu[i], Ntotal[i] )
    nu[i] = eta*pi[i] + (1 - theta)*(1 - pi[i])
    pi[i] ~ dbeta( omega*(kappa - 2) + 1, 
                   (1 - omega)*(kappa - 2) + 1 )
    Py[i] ~ dbin( nu[i], Ntotal[i] )
  }
  omega ~ dbeta( 6, 95)
  kappa = kappaMinusTwo + 2
  kappaMinusTwo ~ dgamma( 5.8, .48 )
  eta ~ dbeta( 910, 90 )
  theta ~ dbeta( 950, 50 )
  Py_tot = sum(Py)
}
"
"
Figure 2: Density Plot of $P_{Y_{tot}}$
The gold bar in the plot, represents the 95% Highest Density Interval (HDI) for the Mode of $P_{y_{tot}}$. 

The HDI indicates a 95% probability that $mode(P_{y_{tot}})$ would fall in between 11,227 and 11,816.
Posterior Estimates

Distributions of Py\[i\]
Subset of i 10 or Lower

Figure 3: Density Plot(s) of \( Py_i \)
The blue bar in the plot represents the 95% HDI for the mode of $P_{yi}$.

The variability in $P_{yi}$ between the different events is evident in the plot above.
Figure 4: Density Plot(s) of $\pi_i$
Posterior Estimates

- The blue bar in the plot represents the 95% HDI for the mode of $\pi_i$

- Posterior estimates for event prevalences appear to remain consistent with the (simulated) data.

- $0.00000842 \leq \text{mode}(\pi_1) \leq 0.0225$
- $0.00000220 \leq \text{mode}(\pi_2) \leq 0.0230$
- $0.000000563 \leq \text{mode}(\pi_3) \leq 0.0119$
- $0.00000180 \leq \text{mode}(\pi_4) \leq 0.0114$
- $0.00000125 \leq \text{mode}(\pi_5) \leq 0.0132$
- $0.000000919 \leq \text{mode}(\pi_6) \leq 0.0209$
- $0.000641 \leq \text{mode}(\pi_7) \leq 0.00939$
Figure 5: Density Plot of $\nu_i$ and $\pi_i$ by Event
Posterior Estimates

As evident from the plot, one’s chance of testing positive is greater than the chance of actually having the coronavirus.

The coronavirus tests are imperfect!
Posterior Estimates

Figure 6: Density Plot of $\eta$
The gold bar represents the 95% HDI for the mode of $\eta$.

As shown in the posterior distribution, the sensitivity for this test was not perfect.

It’s 95% likely that upwards of 11% of tests were false negatives.
Figure 7: Density Plot of $\theta$
Posterior Estimates

- $0.977 \leq \text{mode}(\theta) \leq 0.980$

- The gold bar represents a 95% HDI for the mode of $\theta$.
- Specificity yielded *slightly* better results than sensitivity.
- It is 95% likely that 3% of tests were false positives.
Tests were prone to incorrect results, both false positive and false negative.

At the Rose Garden event people who tested positive were refused entry, but for all of the other events no tests were required for entry. So if the tests were done, some people entering would have had false positive and some would have false negative results.
To ensure that all the MCMC samples had properly converged, we made use trace plots, and used the Gelman-Rubin Statistic.
Diagnostics

Figure 8: Trace Plot for $Py_{tot}$

- As evident from the trace plot, all the chains have seemingly converged.
Diagnostics

Figure 9: Trace Plots for $Py_1 \rightarrow Py_6$

- While there were too many individual events to plot all of the trace plots on the same page, the first six trace plots for $Py_i$ serve as a good representation for the convergence of the chains.
As evident from the trace plots, the estimates for the distribution of $\pi_i$ have seemingly converged as well.

Figure 10: Trace Plots for $\pi_1 \rightarrow \pi_6$
As evident from the trace plot, the estimates for the distribution of $\nu_i$ have seemingly converged as well.
As evident from the trace plots, the chains seem to have successfully converged.
What is the Gelman-Rubin statistic?
- Ratio of variance between MCMC chains vs. within MCMC chains
- Can think of as a sort of ANOVA F-test
- Ideally, we would like the Gelman-Rubin statistic to be around 1 (insignificant)
Gelman-Rubin Statistic

Table 1: Gelman-Rubin Statistics: $nu_i$

<table>
<thead>
<tr>
<th>Point.Estimate</th>
<th>Upper.C.I.</th>
<th>Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0002710</td>
<td>1.0006836</td>
<td>$nu[1]$</td>
</tr>
<tr>
<td>1.0001850</td>
<td>1.0005888</td>
<td>$nu[2]$</td>
</tr>
<tr>
<td>0.9999339</td>
<td>1.0000705</td>
<td>$nu[3]$</td>
</tr>
<tr>
<td>0.9999448</td>
<td>1.0000976</td>
<td>$nu[4]$</td>
</tr>
<tr>
<td>0.9999364</td>
<td>0.9999446</td>
<td>$nu[5]$</td>
</tr>
<tr>
<td>0.9998892</td>
<td>0.9999507</td>
<td>$nu[6]$</td>
</tr>
</tbody>
</table>

Gelman-Rubin statistics appear to be around 1 - further suggesting convergence of chains.
Gelman-Rubin Statistic

Table 2: Gelman-Rubin Statistics: $P_y$ and $Py_{tot}$

<table>
<thead>
<tr>
<th>Point.Estimate</th>
<th>Upper.C.I.</th>
<th>Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0002989</td>
<td>1.0010359</td>
<td>$Py_{tot}$</td>
</tr>
<tr>
<td>0.9999381</td>
<td>0.9999539</td>
<td>$Py[1]$</td>
</tr>
<tr>
<td>1.0000482</td>
<td>1.0002504</td>
<td>$Py[2]$</td>
</tr>
<tr>
<td>0.9999843</td>
<td>1.0002102</td>
<td>$Py[3]$</td>
</tr>
<tr>
<td>1.0001271</td>
<td>1.0006467</td>
<td>$Py[4]$</td>
</tr>
<tr>
<td>0.9999690</td>
<td>1.0001508</td>
<td>$Py[5]$</td>
</tr>
<tr>
<td>1.0000572</td>
<td>1.0002924</td>
<td>$Py[6]$</td>
</tr>
</tbody>
</table>

Gelman-Rubin statistics appear to be around 1 - implying that the estimates for the distribution of $Py_i$ and $Py_{tot}$ have converged.
**Gelman-Rubin Statistic**

Table 3: Gelman-Rubin Statistics: $\pi_i$

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Point.Estimate</th>
<th>Upper.C.I.</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$\pi[1]$</td>
<td>1.0003948</td>
<td>1.0008883</td>
<td>74</td>
</tr>
<tr>
<td>$\pi[2]$</td>
<td>1.0002639</td>
<td>1.0007004</td>
<td>75</td>
</tr>
<tr>
<td>$\pi[3]$</td>
<td>0.9999439</td>
<td>0.9999921</td>
<td>76</td>
</tr>
<tr>
<td>$\pi[4]$</td>
<td>0.9999727</td>
<td>1.0001051</td>
<td>77</td>
</tr>
<tr>
<td>$\pi[5]$</td>
<td>1.0004034</td>
<td>1.0005236</td>
<td>78</td>
</tr>
<tr>
<td>$\pi[6]$</td>
<td>0.9999937</td>
<td>1.0001293</td>
<td>79</td>
</tr>
</tbody>
</table>

Gelman-Rubin statistics are around 1 - implying an insignificant different between the chains - therefore, implying that the chains have converged.
The Gelman-Rubin statistics for $\omega, \kappa, \eta, \text{ and } \theta$ imply that the estimates for the respective posterior distributions have converged.
Through this model, we were able to:

1. Estimate the total number of people who would have tested positive for coronavirus at each of former President Trump’s events in 2020.

2. Estimate the event specific coronavirus prevalence.

3. Estimate the event specific chance of testing positive for coronavirus.

4. Estimate the sensitivity and specificity of the tests used for the events.

5. Determine the Rose Garden as a unique event – in that all participants tested negative, but resulted in multiple coronavirus cases (due to testing imperfections).
References
