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`

Application

- 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.

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

```
##    Event_Type Number_of_Participants
##  Party  : 6   Min.    :  110
##  Rally  :13   1st Qu.: 1200
##  Airport:54   Median : 5000
##               Mean   : 6117
##               3rd Qu.: 7000
##               Max.   :30000
```

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 <- nrow(na.omit(covid19trump)) # Number of Events

omega <- 0.05 # Mode of the beta Prior

kappa <- 100 # Concentration for beta Prior

prob_nu <- rbeta(k, omega*(kappa - 2) + 1,
                 (1 - omega)*(kappa - 2) + 1) # P(+)

# Loading different event sizes
n <- na.omit(covid19trump$Number_of_Participants)
```

Simulating the Number of Positive Cases

```
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?

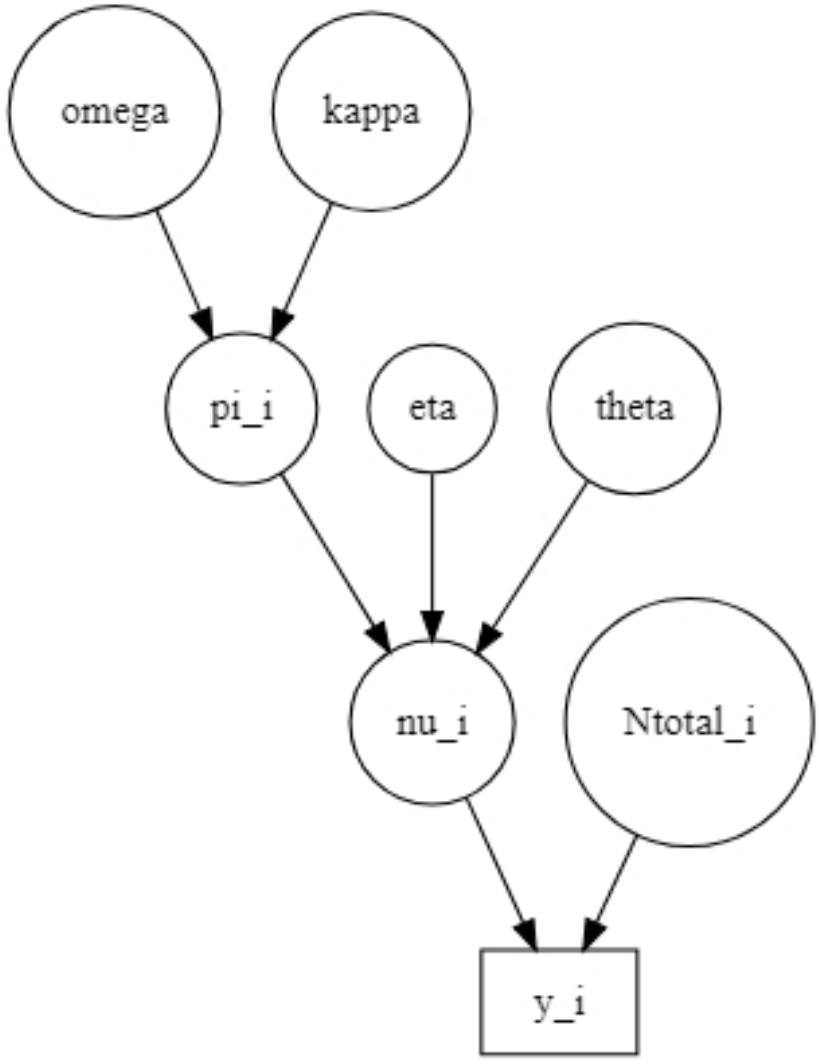
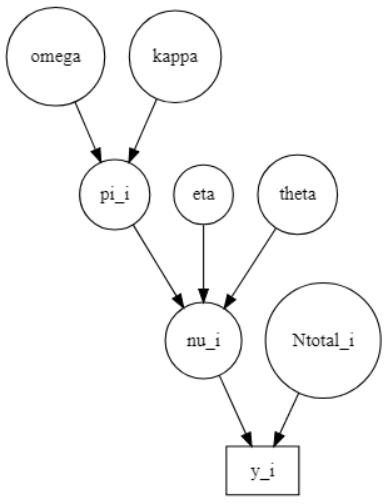


Figure 1: Model Diagram

What is the Model?

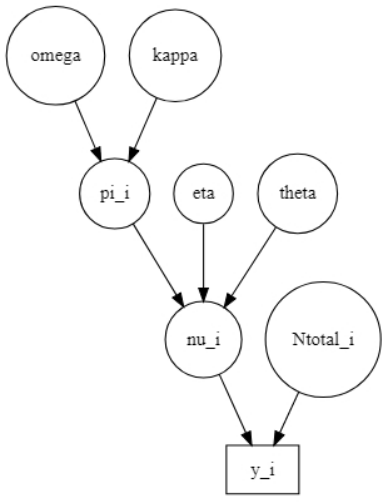


- y_i = The number of people who have tested positive for coronavirus, at each event, i.

- y_i was modeled using a binomial distribution:

$$y_i \sim binomial(\nu_i, Ntotal_i)$$

What is the Model?



- $\eta =$ The sensitivity of the coronavirus tests used.

$$\eta = P(+|D)$$

- $\theta =$ The specificity of the coronavirus tests used.

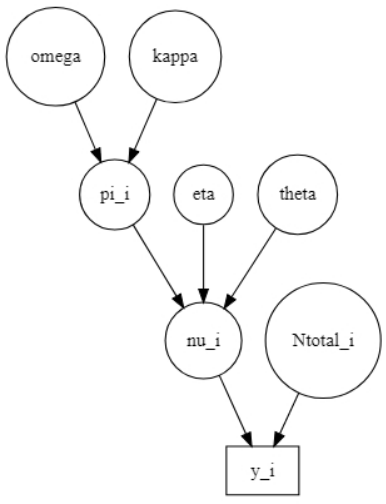
$$\theta = P(-|D^c)$$

- priors

$$\eta \sim beta(910, 90)$$

$$\theta \sim beta(950, 50)$$

What is the Model?

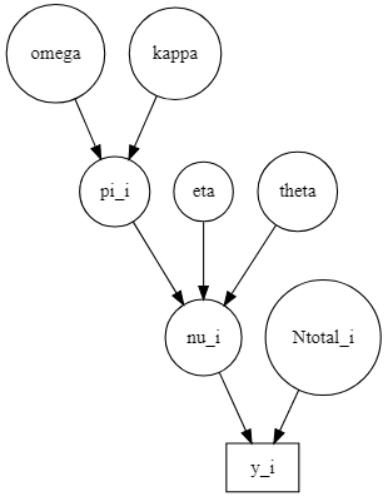


- π_i = The **true** prevalence of the coronavirus at each event, *i*.

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

$$\pi_i \sim beta(\omega * (\kappa - 2) + 1, (1 - \omega) * (\kappa - 2) + 1)$$

What is the Model?



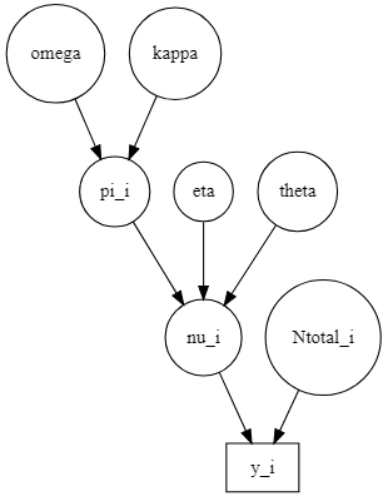
- ν_i = The probability of testing positive for coronavirus, at each event, *i*.

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

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

- $Ntotal_i$ = The number of participants at each event, *i*.
- **Note:** The parameter ν_i dependes on η and θ

What is the Model?



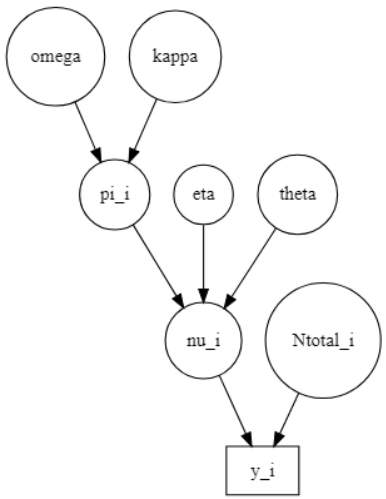
- ω = The mode of the beta prior distribution put upon the prevalence of the coronavirus.

- ω is used in the model for an alternative parameterization of the beta distribution, rather than use of β or α.

- prior:

$$\omega \sim beta(6, 95)$$

What is the Model?



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

- κ is used in the model for an alternative parameterization of the beta distribution as well.

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

- prior:

$$\kappa - 2 \sim \mathit{gamma}(5.8, 0.48)$$

Posterior Predictive Distributions

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

$$Py_i \sim binomial(\nu_i, Ntotal_i)$$

- 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)
}
"
```

Posterior Estimates

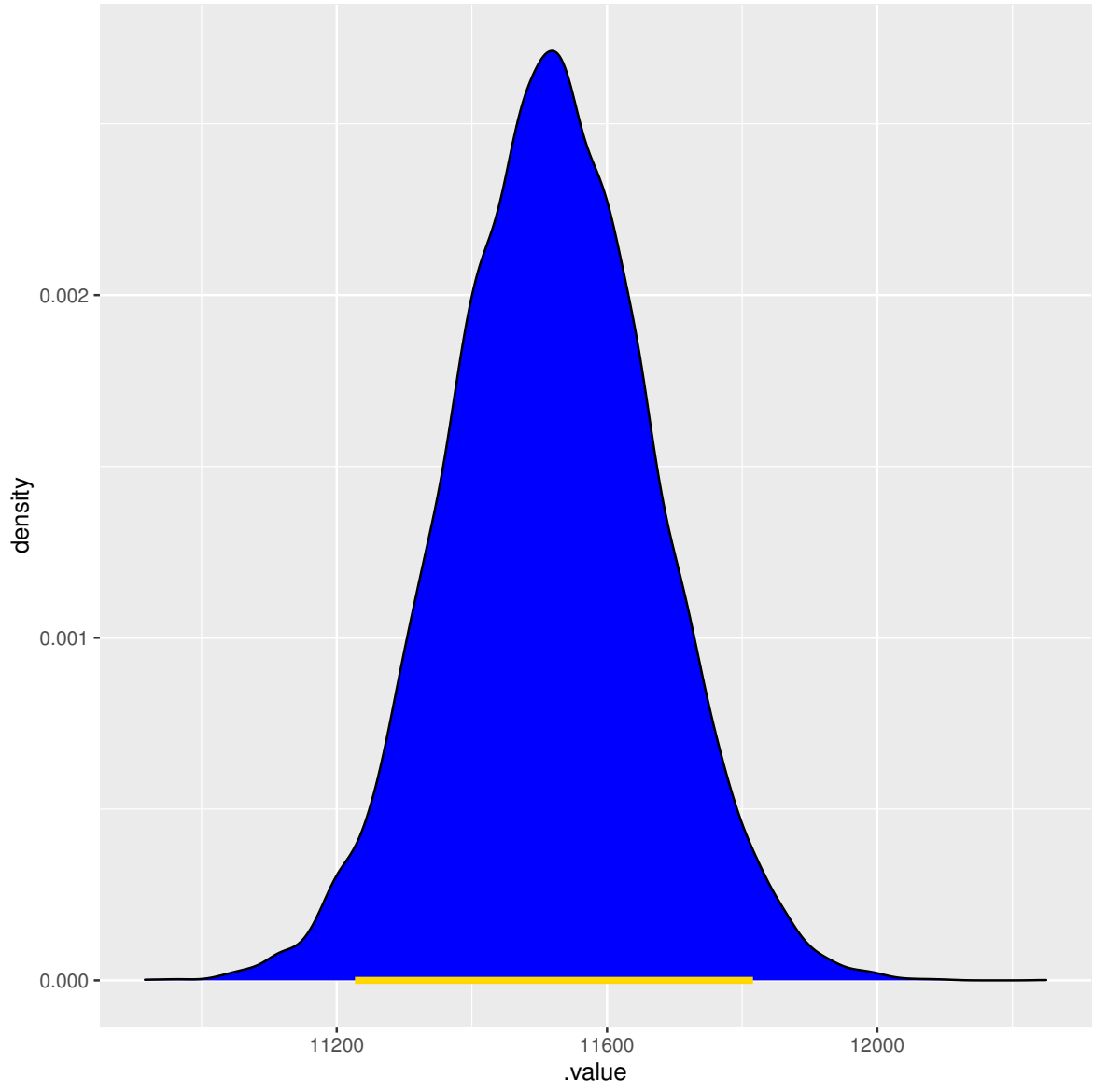
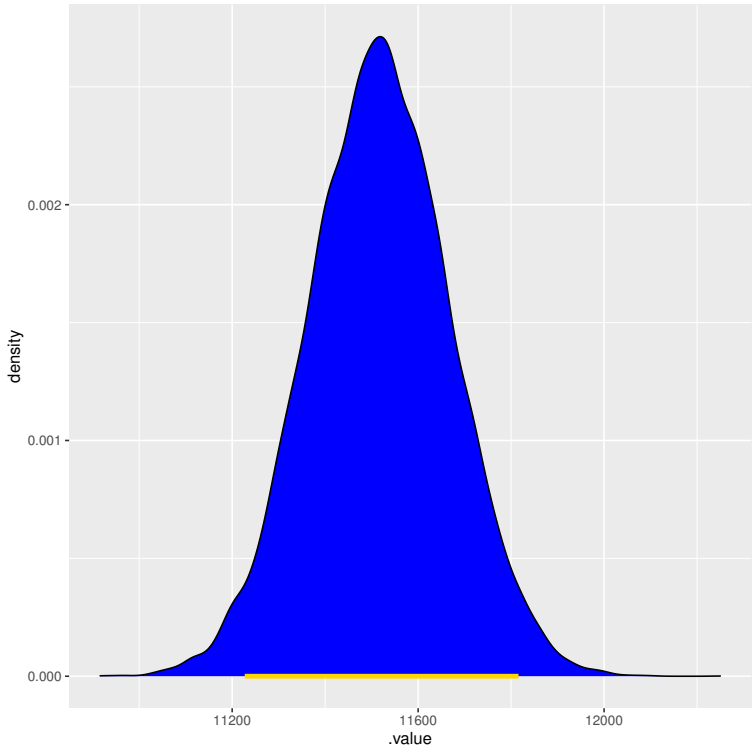


Figure 2: Density Plot of Py_{tot}

Posterior Estimates



$$11,227 \leq mode(Py_{tot}) \leq 11,816$$

- The gold bar in the plot, represents the 95% Highest Density Interval (HDI) for the Mode of Py_{tot}.
- The HDI indicates a 95% probability that $mode(Py_{tot})$ would fall in between 11,227 and 11,816.

Posterior Estimates

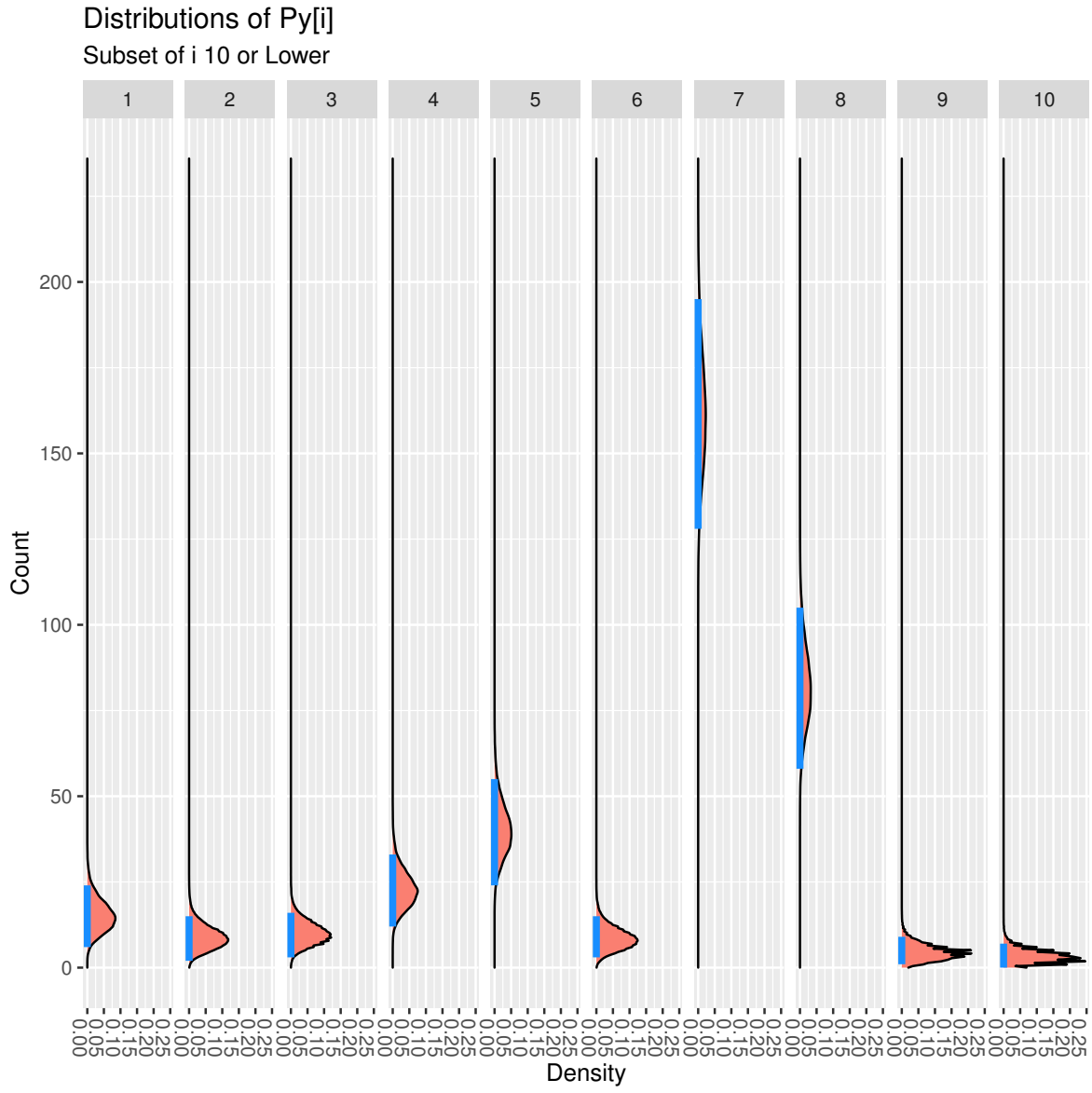
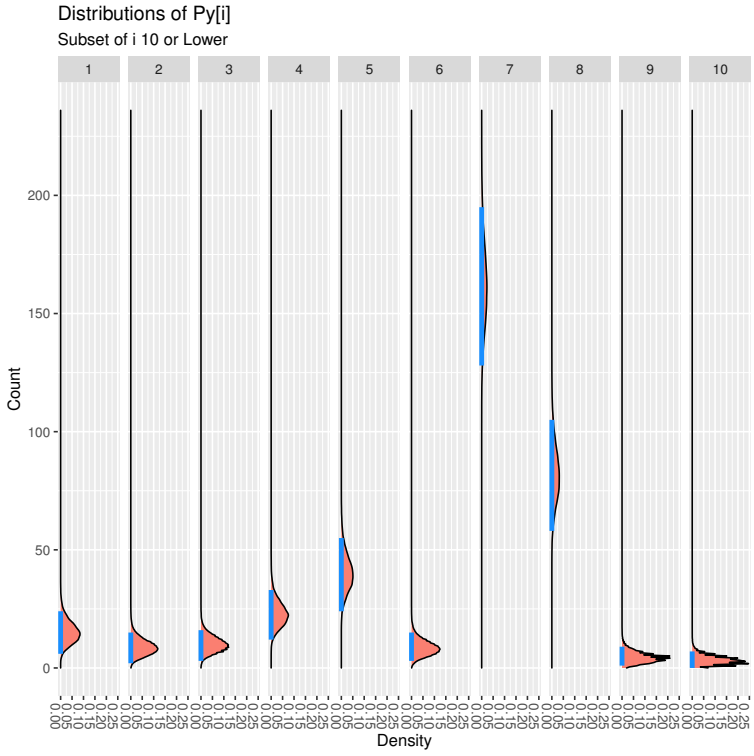


Figure 3: Density Plot(s) of Py_i

Posterior Estimates



$$6 \leq mode(Py_1) \leq 24$$
$$2 \leq mode(Py_2) \leq 15$$
$$4 \leq mode(Py_3) \leq 17$$
$$17 \leq mode(Py_4) \leq 18$$
$$12 \leq mode(Py_5) \leq 33$$
$$24 \leq mode(Py_6) \leq 55$$
$$128 \leq mode(Py_7) \leq 195$$

- The blue bar in the plot represents the 95% HDI for the mode of Py_i.
- The variability in Py_i between the different events is evident in the plot above.

Posterior Estimates

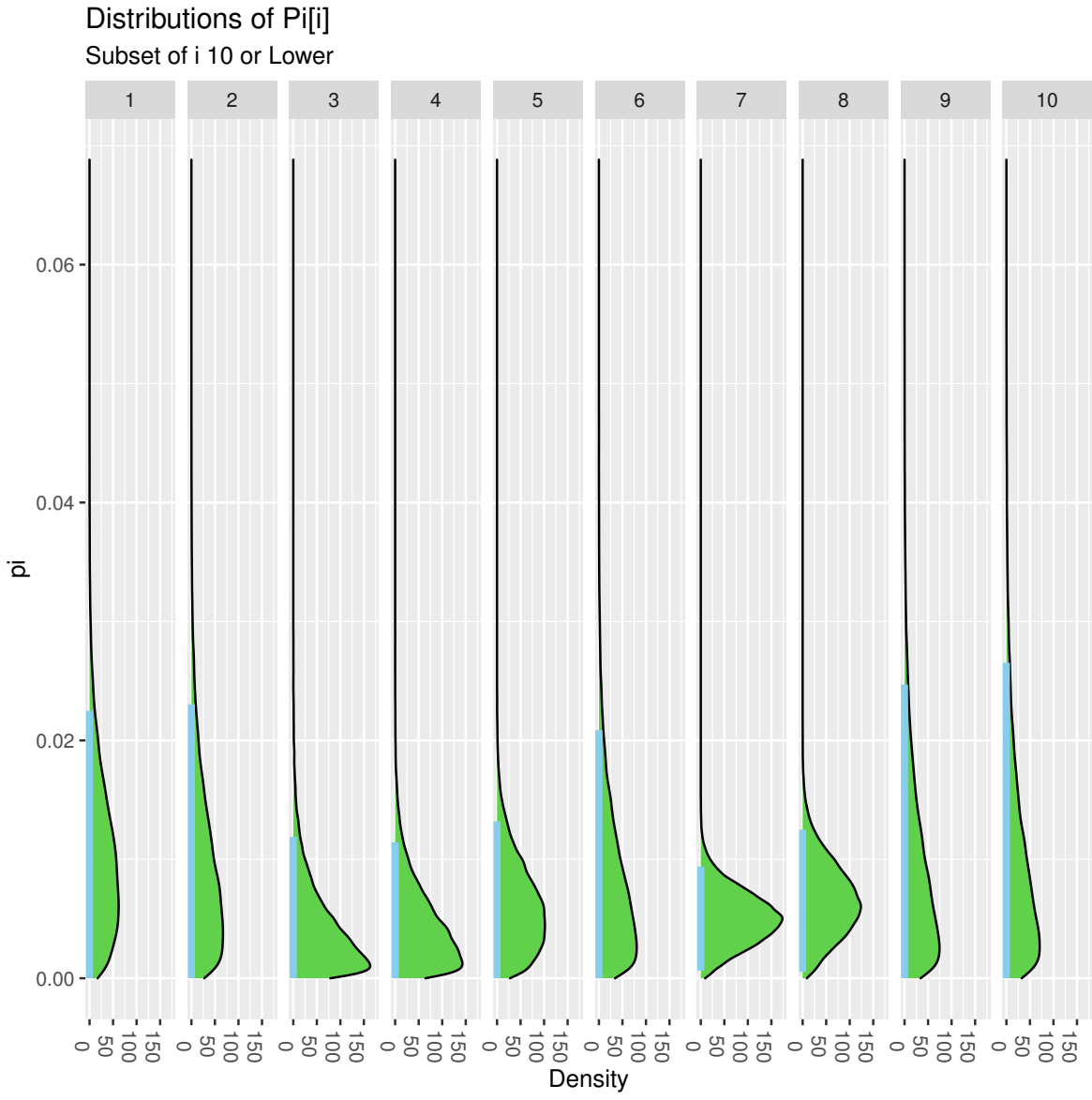
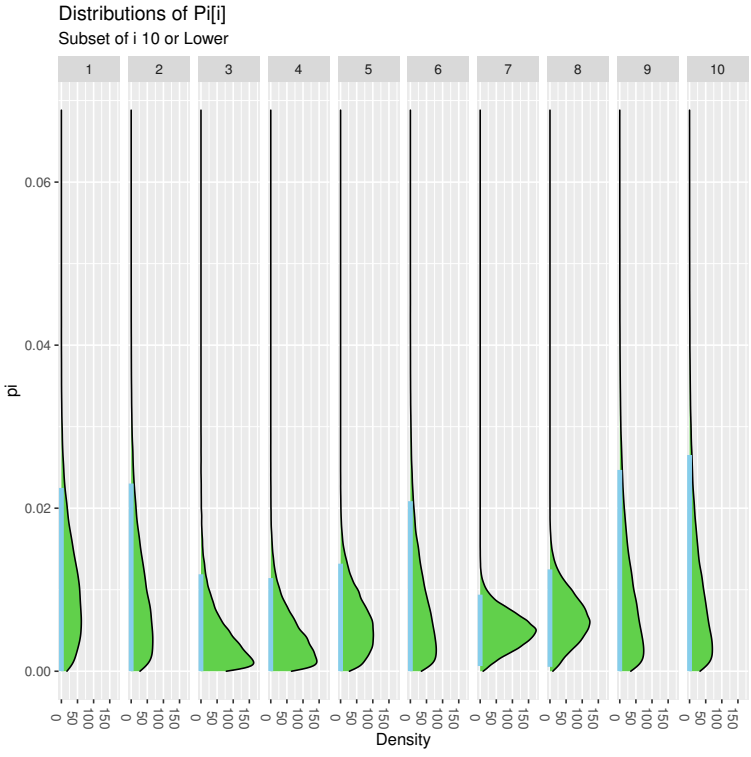


Figure 4: Density Plot(s) of π_i

Posterior Estimates



- $0.00000842 \leq mode(\pi_1) \leq 0.0225$
- $0.00000220 \leq mode(\pi_2) \leq 0.0230$
- $0.000000563 \leq mode(\pi_3) \leq 0.0119$
- $0.00000180 \leq mode(\pi_4) \leq 0.0114$
- $0.00000125 \leq mode(\pi_5) \leq 0.0132$
- $0.000000919 \leq mode(\pi_6) \leq 0.0209$
- $0.000641 \leq mode(\pi_7) \leq 0.00939$

- The blue bar in the plot represents the 95% HDI for the mode of π_i
- Posterior estimates for event prevalences appear to remain consistent with the (simulated) data.

Posterior Estimates

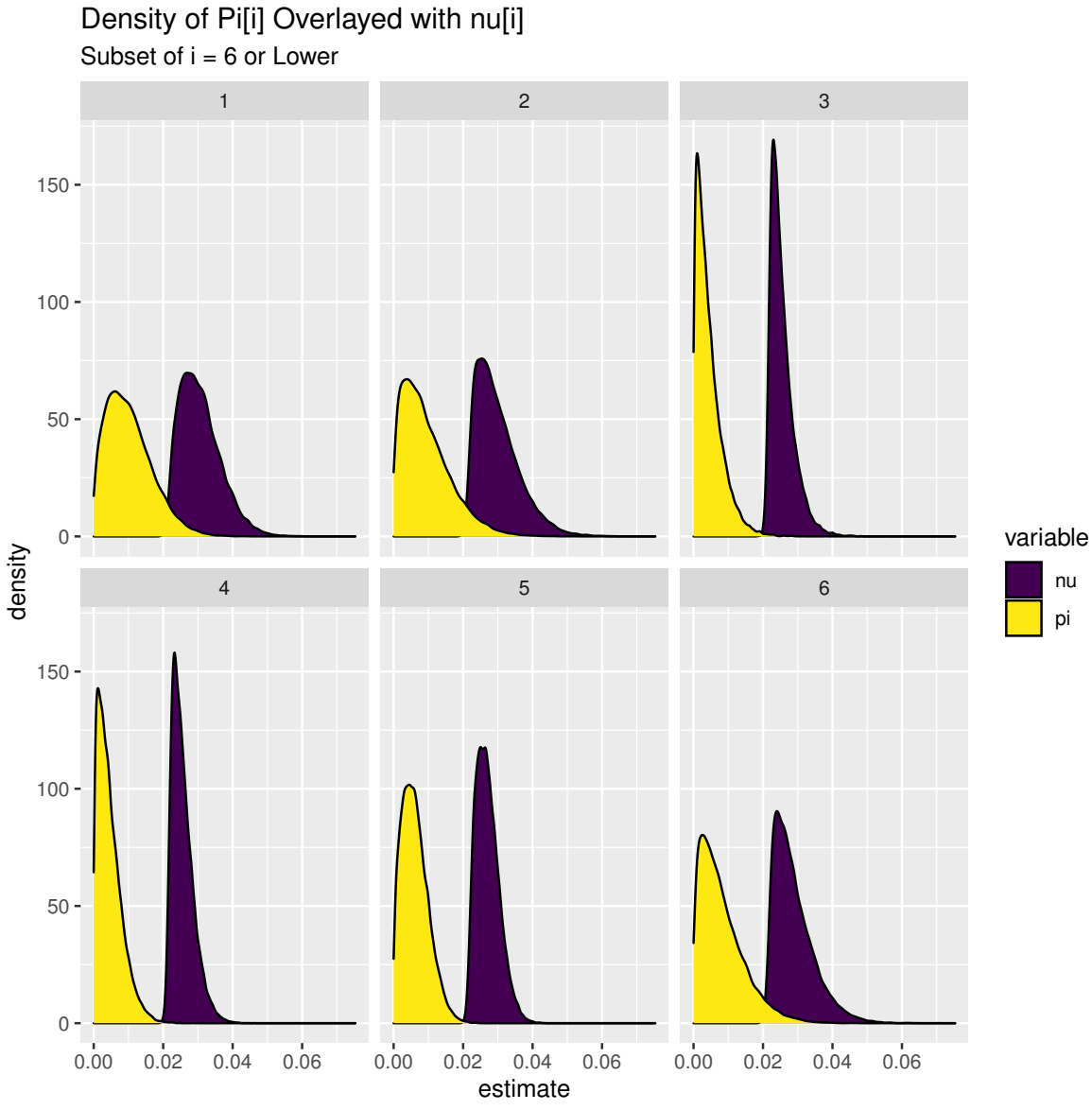
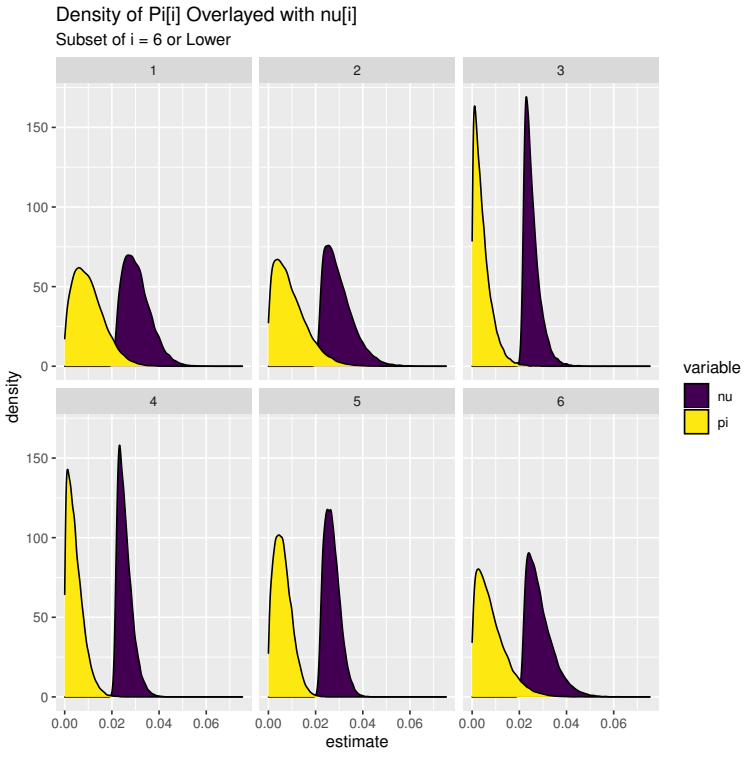


Figure 5: Density Plot of ν_i and π_i by Event

Posterior Estimates



- $0.0214 \leq mode(\nu_1) \leq 0.0418$
- $0.0209 \leq mode(\nu_2) \leq 0.0423$
- $0.020 \leq mode(\nu_3) \leq 0.0324$
- $0.0320 \leq mode(\nu_4) \leq 0.0320$
- $0.0336 \leq mode(\nu_5) \leq 0.0336$
- $0.0209 \leq mode(\nu_6) \leq 0.0404$

- 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

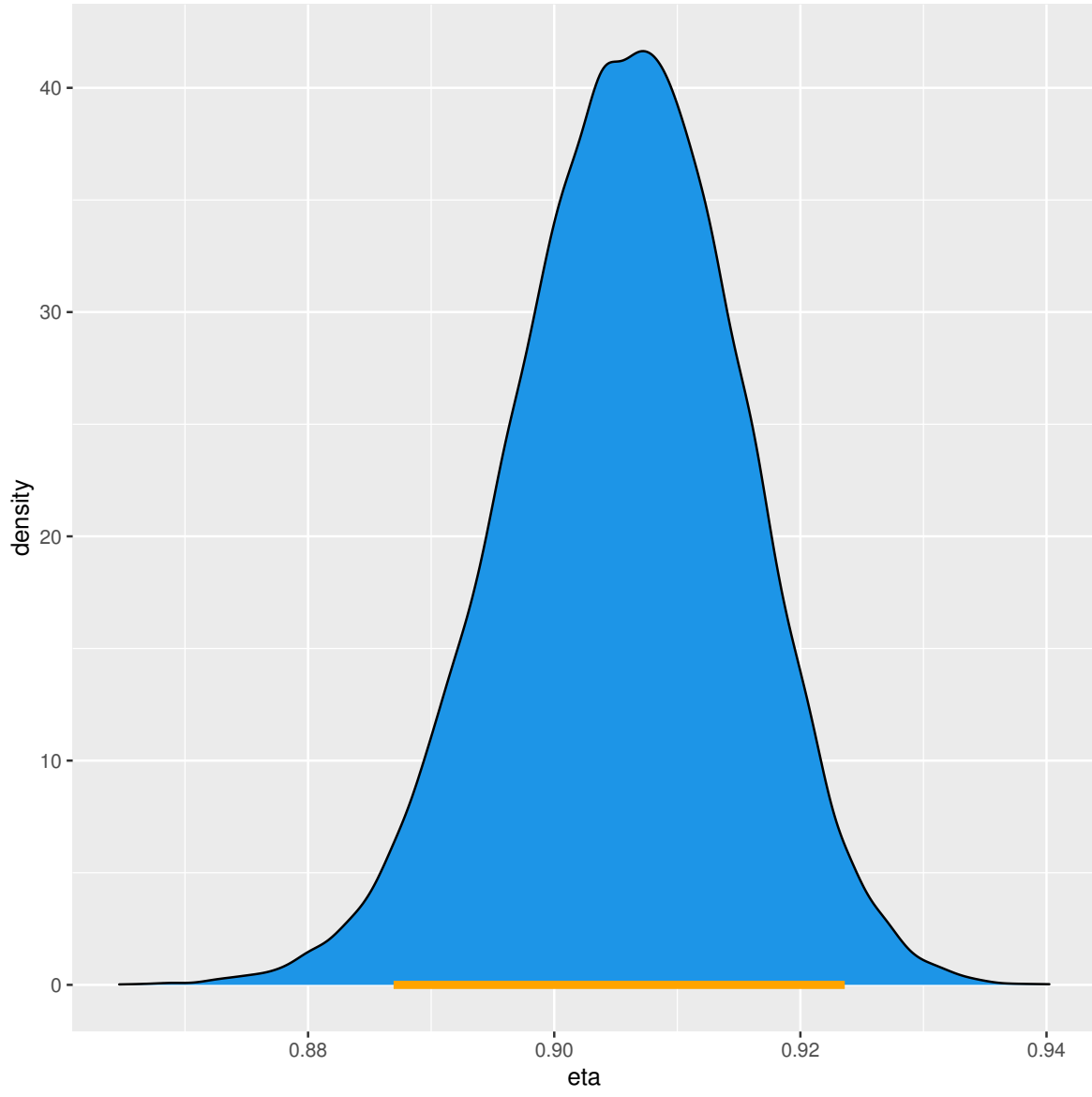
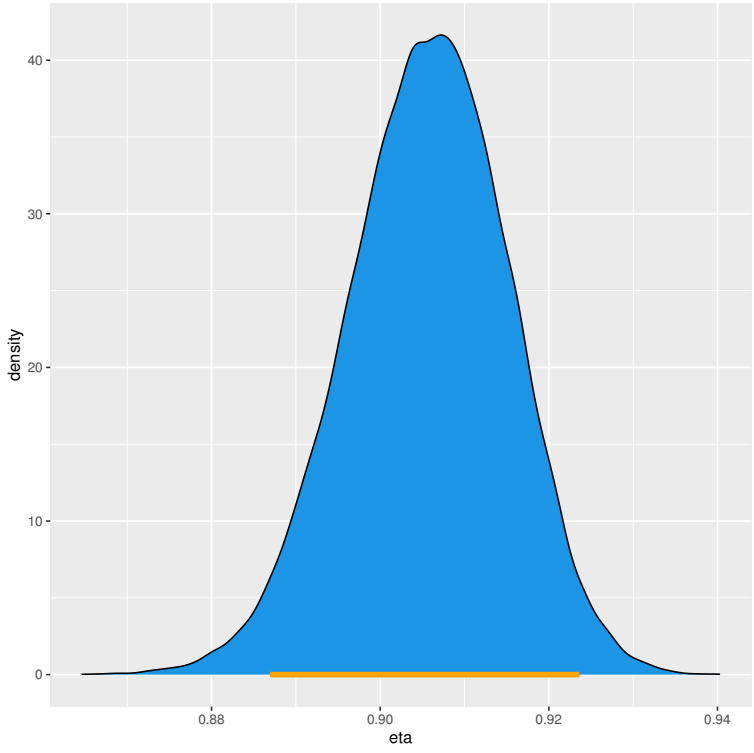


Figure 6: Density Plot of η

Posterior Estimates



- $0.887 \leq mode(\eta) \leq 0.924$

- The gold bar represents the 95% HDI for the mode of η.
- 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.

Posterior Estimates

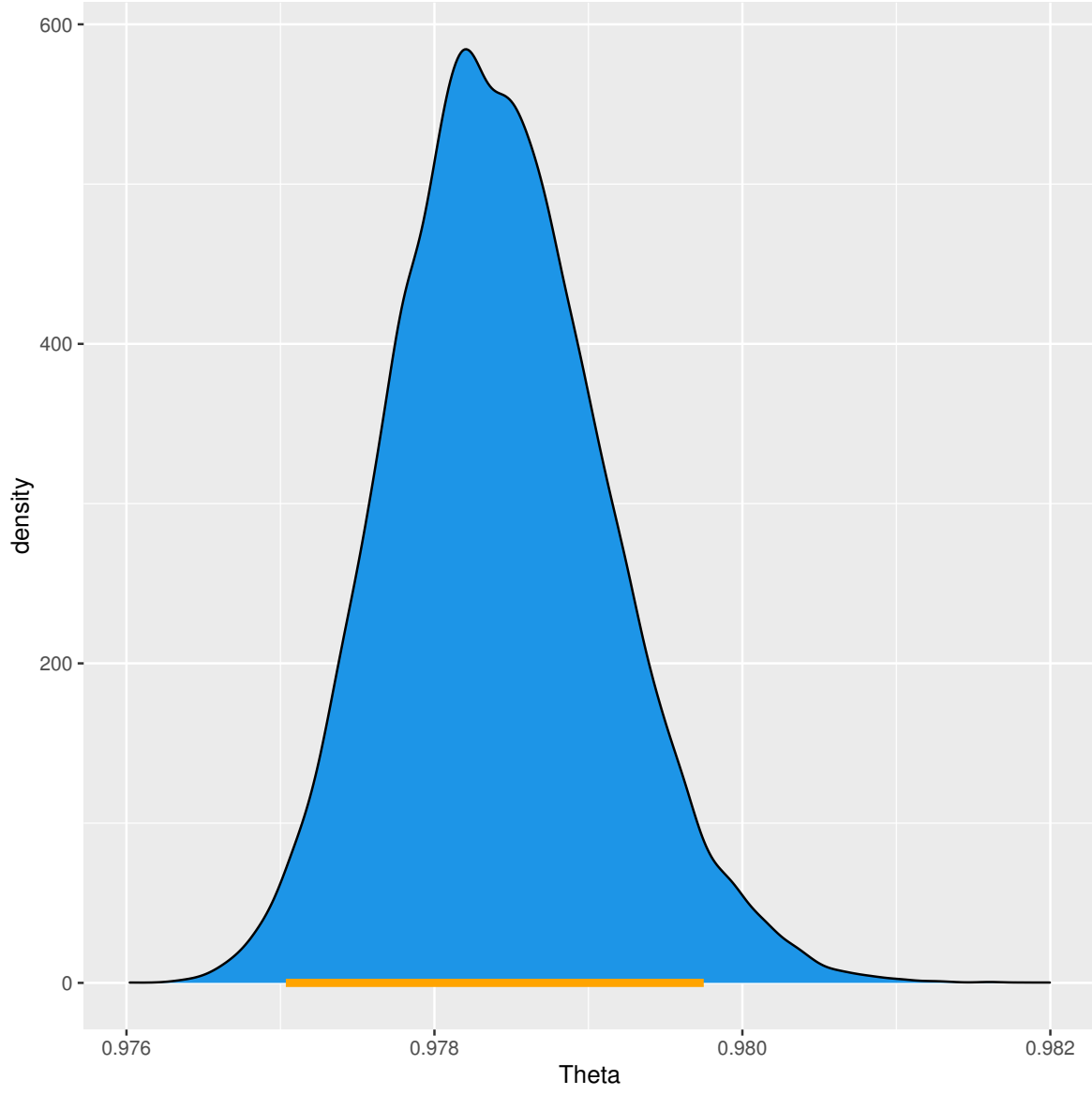
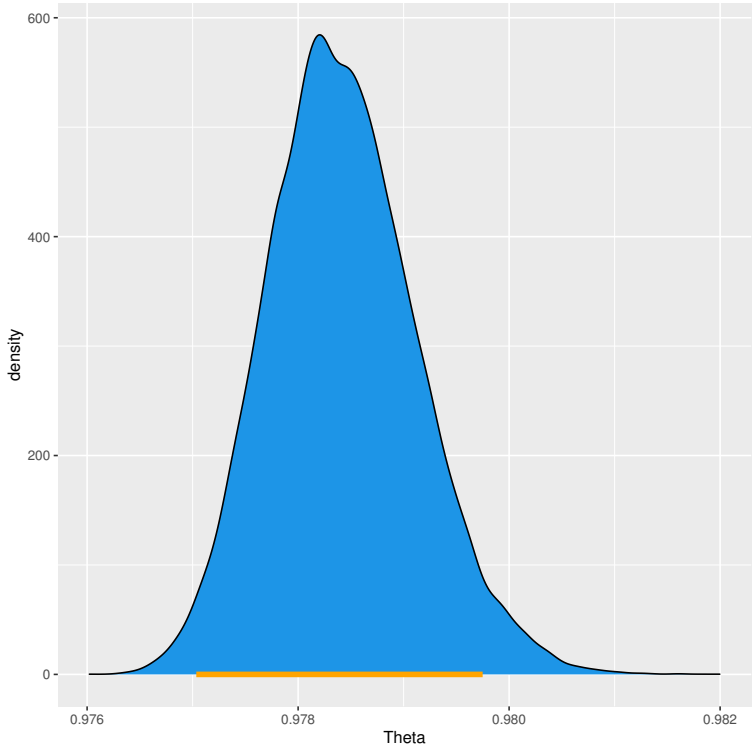


Figure 7: Density Plot of θ

Posterior Estimates



- $0.977 \leq mode(\theta) \leq 0.980$

- The gold bar represents a 95% HDI for the mode of θ.
- Specificity yielded *slightly* better results than sensitivity.
- It is 95% likely that 3% of tests were false positives.

Results

- 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.

Diagnostics

- To ensure that all the MCMC samples had properly converged, we made use trace plots, and used the Gelman-Rubin Statistic.

Diagnostics

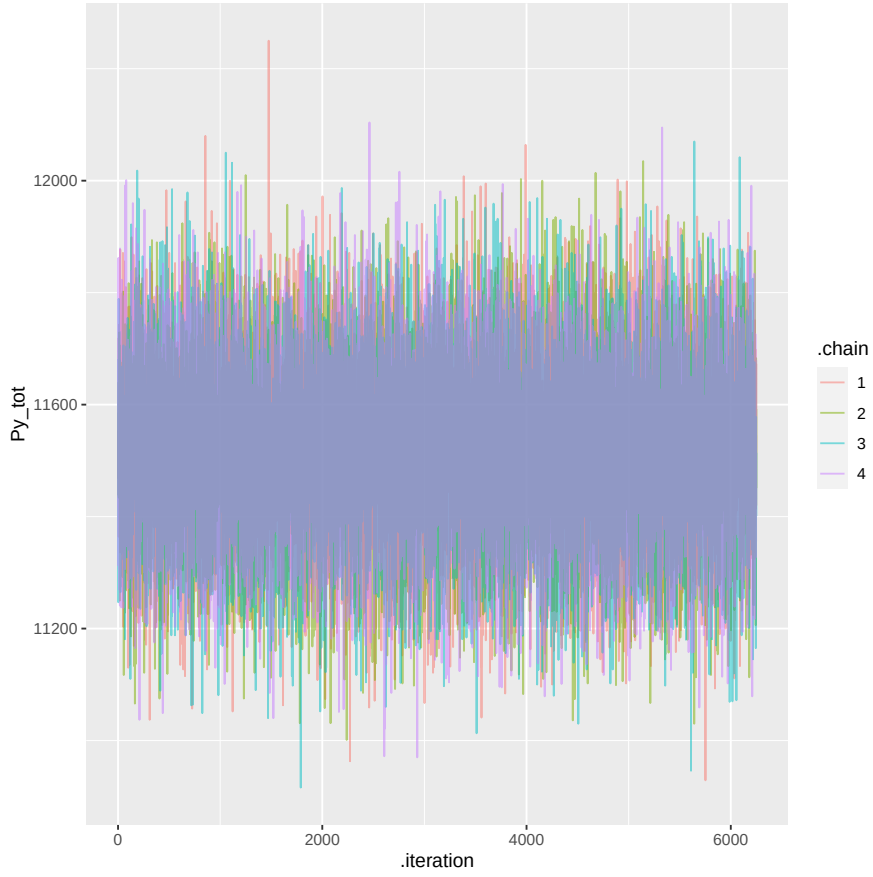


Figure 8: Trace Plot for Py_{tot}

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

Diagnostics

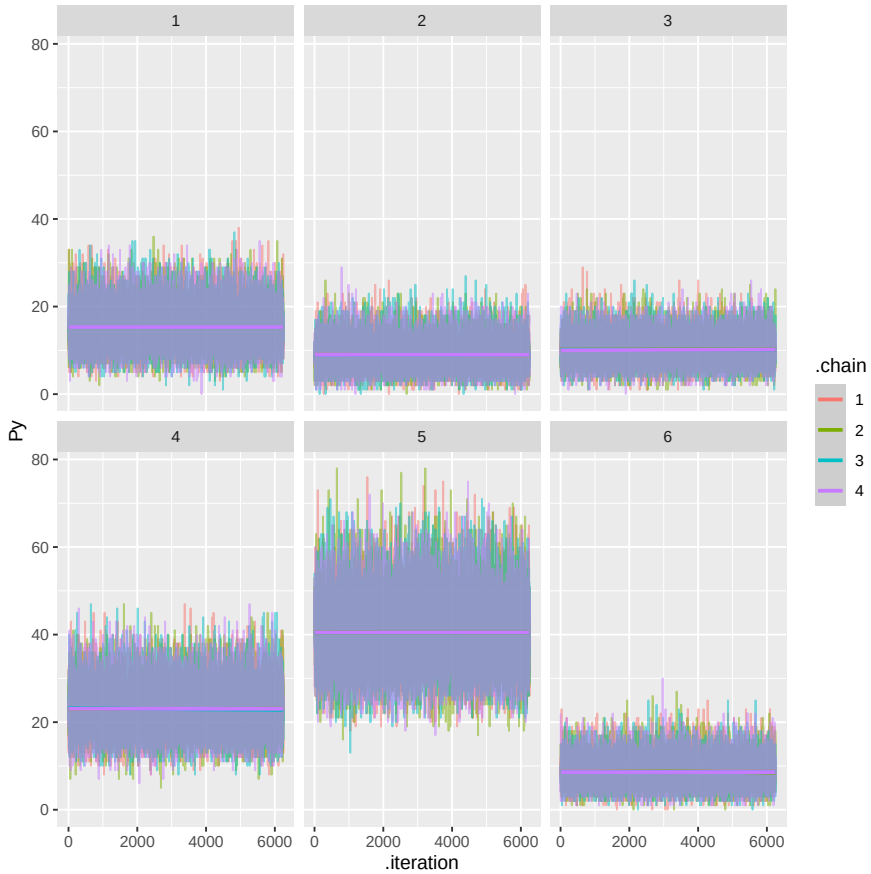


Figure 9: Trace Plots for $Py_1 \rightarrow Py_6$

- 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.

Diagnostics

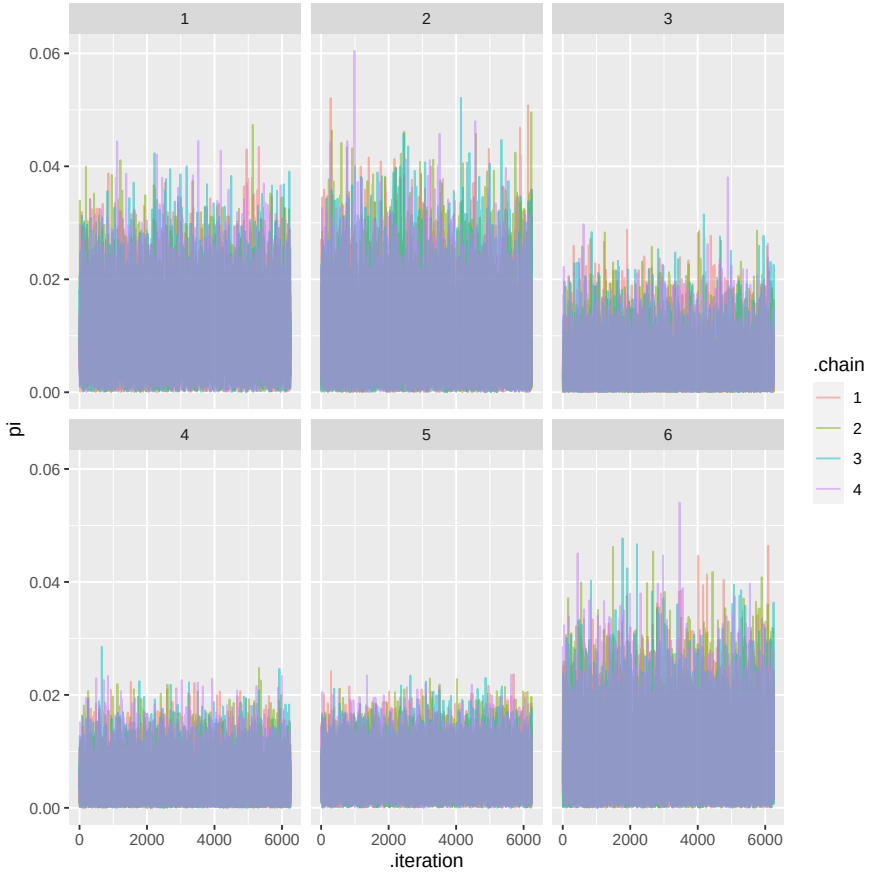


Figure 10: Trace Plots for $\pi_1 \rightarrow \pi_6$

- As evident from the trace plots, the estimates for the distribution of π_i have seemingly converged as well.

Diagnostics

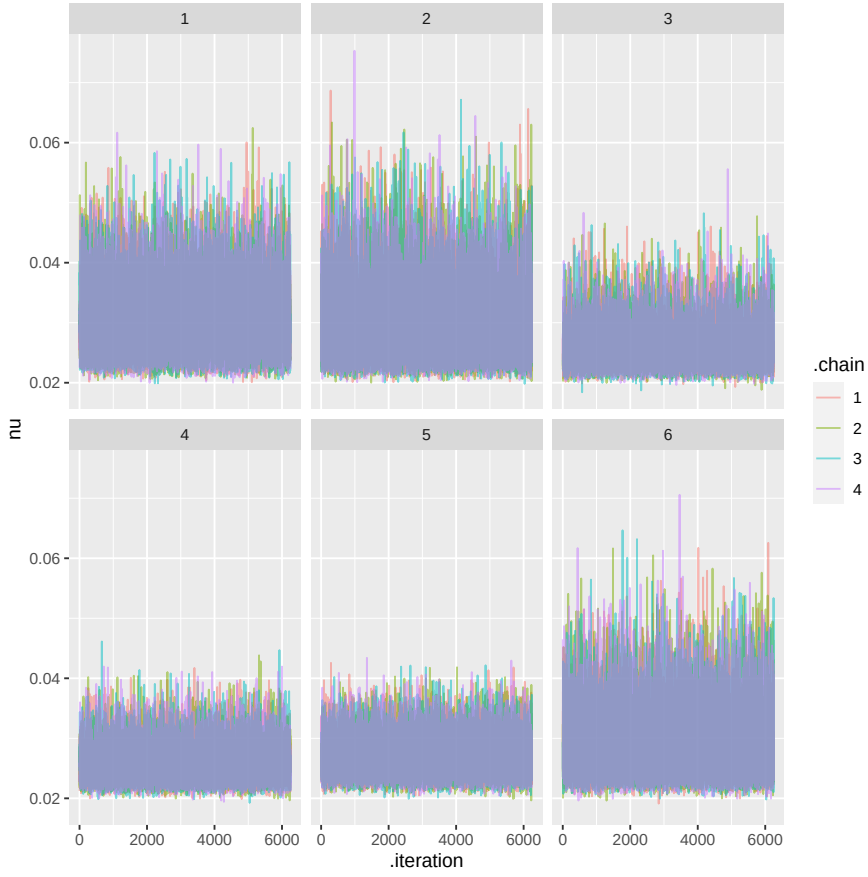


Figure 11: Trace Plots for $\nu_1 \rightarrow \nu_6$

- As evident from the trace plot, the estimates for the distribution of ν_i have seemingly converged as well.

Diagnostics

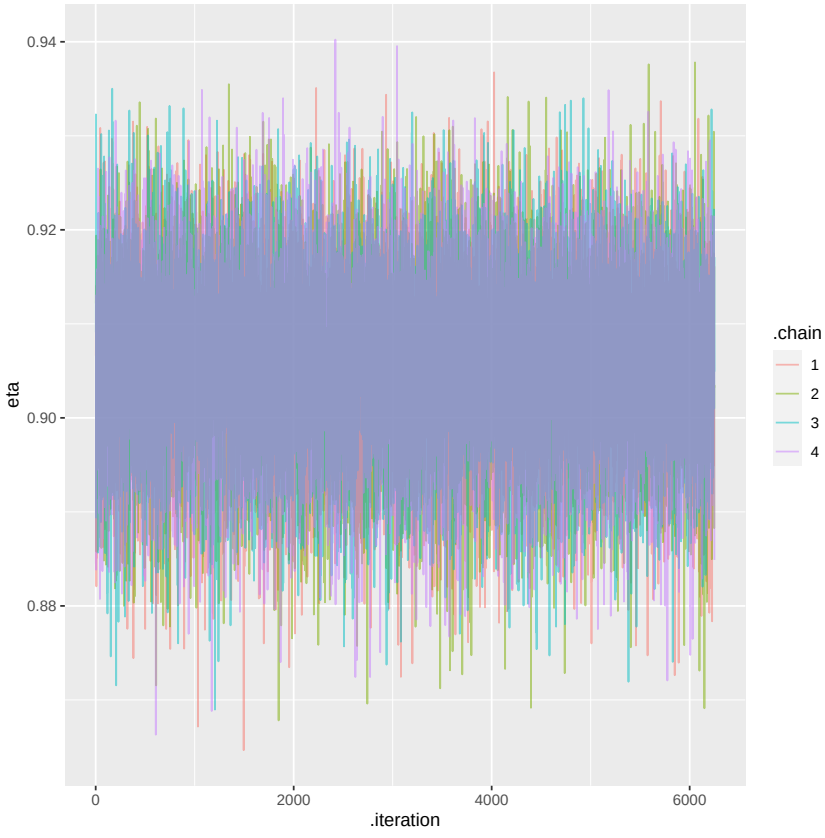


Figure 12: Trace Plot for η

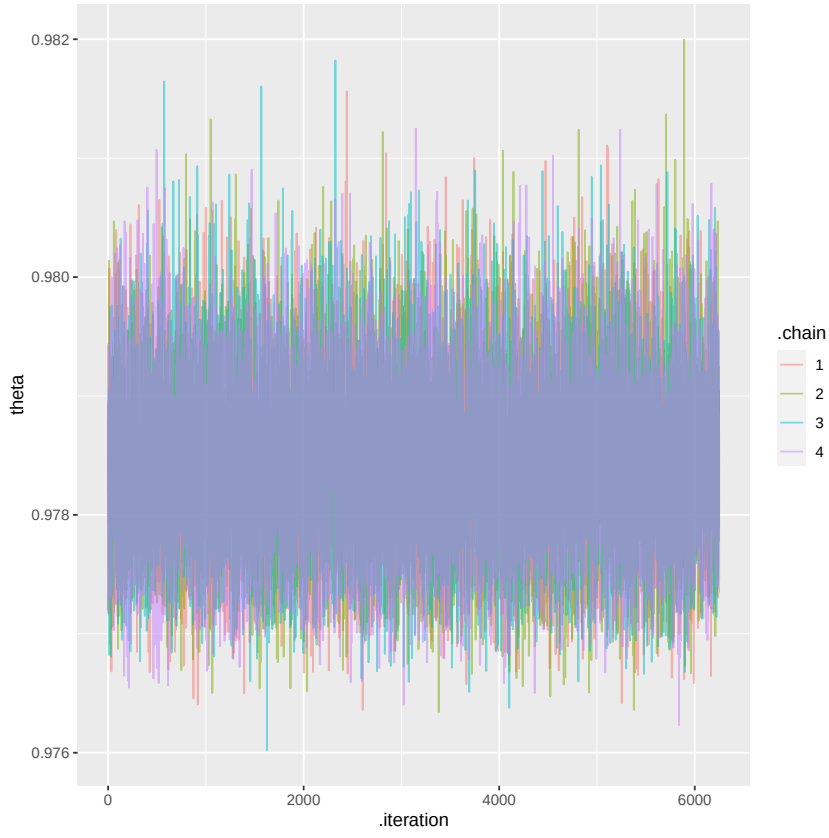


Figure 13: Trace Plot for θ

- As evident from the trace plots, the chains seem to have successfully converged.

Gelman-Rubin Statistic

- 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

Point.Estimate	Upper.C.I.	Parameter
1.0002710	1.0006836	nu[1]
1.0001850	1.0005888	nu[2]
0.9999339	1.0000705	nu[3]
0.9999448	1.0000976	nu[4]
0.9999364	0.9999446	nu[5]
0.9998892	0.9999507	nu[6]

- 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}

Point.Estimate	Upper.C.I.	Parameter
1.0002989	1.0010359	Py_tot
0.9999381	0.9999539	Py[1]
1.0000482	1.0002504	Py[2]
0.9999843	1.0002102	Py[3]
1.0001271	1.0006467	Py[4]
0.9999690	1.0001508	Py[5]
1.0000572	1.0002924	Py[6]

- 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

	Point.Estimate	Upper.C.I.	Parameter
74	1.0003948	1.0008883	pi[1]
75	1.0002639	1.0007004	pi[2]
76	0.9999439	0.9999921	pi[3]
77	0.9999727	1.0001051	pi[4]
78	1.0004034	1.0005236	pi[5]
79	0.9999937	1.0001293	pi[6]

- Gelman-Rubin statistics are around 1 - implying an insignificant different between the chains - therefore, implying that the chains have converged.

Gelman-Rubin Statistic

Table 4: Gelman-Rubin Statistics: *Eta*, *Theta*, *Omega*, and *Kappa*

Point.Estimate	Upper.C.I.	Parameter
0.9999705	1.000109	omega
0.9999903	1.000106	kappa
1.0002021	1.000831	eta
1.0002289	1.000769	theta

- The Gelman-Rubin statistics for $\omega, \kappa, \eta,$ and θ imply that the estimates for the respective posterior distributions have converged.

Conclusion

- 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

- Kruschke, J. (2015). Doing Bayesian Data Analysis: A Tutorial with R, Jags, and Stan (2nd ed.). Academic Press / Elsevier.
- Suess, Eric A. (2000). Certifying Countrywise Disease Freedom in Animal Livestock Populations: A Bayesian Approach. JSM 2000.