PROJECT 1.1 (Cohort Study Smoking)

&

PROJECT 2.2 (Cumulative Incidence)

BY GROUP 19:

Ayunning Tieas - r0825872

Rubing Wang - r0864561

Ishika Jain - r0915387

Shinichi Moribe - r0913014

Fatma Sümer - r0690016

Supervisor:

Professor Christel Faes

Submitted for:

Concept of Bayesian Data Analysis Course

in Masters of Statistics and Data Science, KU Leuven Academic Year 2022–2023

Table of Contents

Executive Summary of PROJECT 1.1 (Cohort Study Smoking)

1. Assuming a **non-informative prior**, the **posterior** probability of disease among **those exposed is $Beta(172\,,3265)$** analytically, meanwhile the **posterior** probability of disease among **those not exposed is $Beta(118\,,4321)$**.

2. Some **summary measures** of the above posterior distributions are provided, **including** their **mean which is 0.500 for** the mean probability of disease among **those exposed and is 0.026 for** the mean probability of disease among **those not exposed**.

3. Based on the Bayesian analysis and the calculated posterior distribution of the relative risk, there is strong statistical evidence to suggest a **significant association between smoking and stroke**.

4. The **code** to obtain an MCMC samples for the above problem can be seen **in Appendix 1.**

5. The **convergence** of samples resulted by the model (which is coded in Appendix 1) **is fulfilled** based on the indication from the traceplots and the result from the Gelman & Rubin, as well as Geweke Convergence Diagnostic Test.

6. The **summary measures** of the three parameters **from the MCMC chain are closely matching** the values obtained through **the analytical solution**.

7. Attributable risk can can be interpreted as the proportion of disease incidence in the exposed group (smokers) that can be directly attributed to the exposure smoking. Based on our results, it is inferred that **approximately 47% of the stroke cases among smokers can be directly attributed to smoking.**

PROJECT 1.1 (Cohort Study Smoking)

1. Assuming a non-informative prior for the probability of disease among those exposed θ+, give the analytical posterior for the probability of disease among those exposed. Do the same for the probability of disease among those not exposed θ–

In their study, Abbott, Yin, Reed, and Yano conducted a 12-year cohort study to investigate the association between smoking and stroke. Among the 3,435 smokers, 171 had a stroke, while among the 4,437 non-smokers, 117 had a stroke (see Table A.1).

Exposure	Stroke	Non-stroke	Total
Smoker	171(x_+)	3264	3435 (n_+)
Non-smoker	117(x_-)	4320	4437(n_-)
Total	288	7584	7872

x_+: The number of smokers who had a stroke
n_+: The total number of smokers
x_-: The number of non-smokers who had a stroke
n_-: The total number of non-smokers

Table A.1. The Confusion Matrix of Study Data

Given observed data where x_+ =171 out of n_+ =3,435 smokers and x_- =117 out of $n_- =$ 4437 non-smokers experienced a stroke, the likelihood function adheres to a binomial distribution. Through the application of Bayesian principles, we can derive the posterior distributions for θ+ (the probability of disease among smokers) and θ- (the probability of disease among non-smokers). We utilize the Beta distribution as the prior for both θ+ and θ-, since it is the the conjugate prior for the binomial likelihood. Assuming a non-informative prior for disease probability, the unit Beta distribution with parameters $\alpha_{prior} = 1$, $\beta_{prior} = 1$ or denoted as Beta(1,1) will serve as a suitable prior. By combining the prior information with the observed data through the likelihood function, we obtain the posterior distributions for θ+ and θ- by following steps:

(1) For θ+

The posterior distribution for the probability of disease among those exposed, denoted as, $p(\theta^+|Y)$, can be derived using the Beta and the Binomial conjugate relationship. The posterior distribution is given by:

$$p(\theta_+|Y) \sim Beta\left(\left[\alpha_{prior} + x_+\right],\ \left[\beta_{prior} + n_+ - x_+\right]\right)$$

Substituting the values from the study, we have:

$$p(\theta_+|Y) \sim Beta([1 + 171\ ], [1 + 3435 - 171])$$

Simplifying and obtaining the posterior distribution for θ+ :

$$p(\theta_+|Y) \sim Beta(172\ , 3265)$$

(2) For θ-

Similarly, the posterior distribution for the probability of disease among those not exposed is given by:

$$p(\theta_-|Y) \sim Beta\left(\left[\alpha_{prior} + x_-\right], \left[\beta_{prior} + n_- - x_-\right]\right)$$

Substituting the values from the study, we have:

$$p(\theta_-|Y) \sim Beta([1 + 117\ ], [1 + 4437 - 117])$$

Simplifying and obtaining the posterior distribution for θ- :

$$p(\theta_-|Y) \sim Beta(118\ , 4321)$$

This gives us the analytical form of the posterior distributions and these two posterior distribution plots are shown in Figure A.1 below.

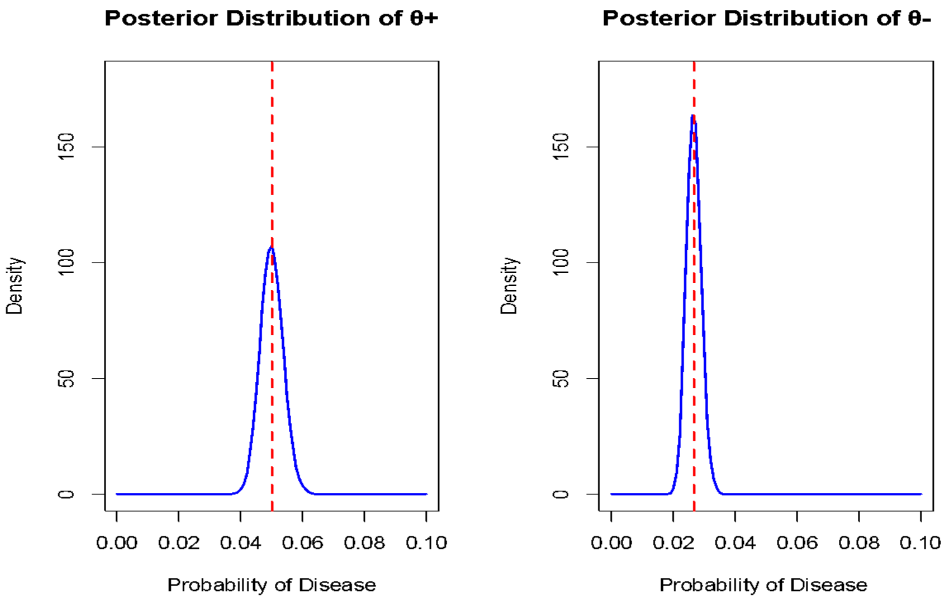

Figure A. 1. Posterior Distribution of θ+ and θ–

2. Give some summary measures of the above posterior distributions.

The summary measures for Beta distribution are analytically straightforward. For a Beta(α_{pos}, β_{pos}) distribution (subscript 'pos' means 'posterior'):

$$Mean = \frac{\alpha_{pos}}{(\alpha_{pos} + \beta_{pos})}$$

$$Mode = \frac{\alpha_{pos} - 1}{(\alpha_{pos} + \beta_{pos} - 2)}$$

$$Variance = \frac{\alpha_{pos}\beta_{pos}}{(\alpha_{pos} + \beta_{pos})^2(\alpha_{pos} + \beta_{pos} + 1)}$$

$$Standard\ Deviation(SD) = \sqrt{variance}$$

In addition, we use the function qbeta() to find the 2.5% and 97.5% quantiles for 95% credible interval (CI) as well as the approximation of median. Furthermore, the function hdi() is used to find the highest posterior density (HPD) interval. Summary measure results are shown below (In Table A.2).

Posterior	α_{pos}	β_{pos}	Mean	Median	Mode	SD	95%CI	95%HPD
$p(\theta_+\|Y)$	172	3265	0.0500	0.0499	0.0498	0.0037	[0.0430, 0.0575]	[0.0425, 0.0566]
$p(\theta_-\|Y)$	118	4321	0.0265	0.0265	0.0263	0.0024	[0.0220, 0.0315]	[0.0218, 0.0310]

Table A.2. The Summary Measure of The Posteriors

In summary, since the posterior distribution for the probability of disease among those exposed follows a Beta(172,3265) distribution, thus it has mean of 0.0500, and standard deviation of 0.0037. Meanwhile, the posterior distribution for the probability of disease among those not exposed follows a Beta(118,4321) distribution so its mean is 0.0265, and standard deviation of 0.0024. Those summary statistics show us that on the average, the probability of disease (stroke) among those exposed (smokers) is indeed higher than those among those not exposed (non-smokers).

3. Can you visualise the posterior distribution of the relative risk, defined as θRR = θ+/θ–? Use a sample from the above derived analytical posterior distribution to answer this question. Give some summary measures of the posterior distribution of the relative risk. Can you conclude that there is an association between smoking and stroke?

The relative risk is defined as $\theta_{RR} = \frac{\theta_+}{\theta_-}$ .To visualize the posterior distribution of the relative risk, we firstly draw samples for θ_+ using the rbeta() function where the distribution is $Beta(172\,,3265)$, as well as for θ_- with $Beta(118\,,4321)$. After obtaining 1,000 samples from each posterior distributions, we then calculate the relative risk by taking the ratio between those samples for θ_+ and θ_-. The result can now represent the posterior distribution of the relative risk.

Mean	Median	Mode	SD	95% Equal Tail CI	95% HPD Interval
1.9008	1.8867	1.8751	0.2274	[1.5066, 2.3572]	[1.4896, 2.3417]

Table A.3. The summary measure of the posterior of relative risk

Based on the summary statistics in Table A.3, both the mean and median values of the posterior distribution of relative risk are almost 2. They indicate that, given this data and settings, the risk of stroke in smokers is nearly twice that in non-smokers. Furthermore, both the 95% equal tail CI for the relative risk ([1.50, 2.35]) and the 95% HPD interval ([1.48, 2.34]) lie above the threshold value of 1. Thus, it suggests a statistically significant association between smoking and the occurrence of stroke. This result can be visualized in a histogram of the posterior distribution for the relative risk in Figure A.2, where the red line shows the mean.

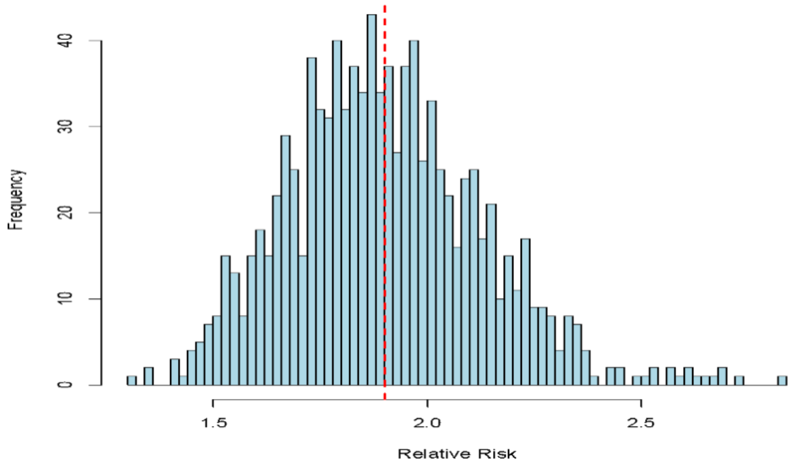

Figure A.2. Posterior Distribution of the Relative Risk

In conclusion, based on the Bayesian analysis and the calculated posterior distribution of the relative risk, there is strong statistical evidence to suggest a significant association between smoking and stroke.

4. Write jags, OpenBugs or Nimble code, to obtain an MCMC samples for the above problem.

For the association between smoking and stroke, we fit the Bayesian model using JAGS to generate MCMC samples from the posterior distribution of the parameters. After we describe the likelihood and prior distributions of the parameters, the JAGS software generates the MCMC samples from the posterior distribution using Gibbs sampling which is a type of MCMC algorithm that samples from the full conditional distributions of the parameters given the other parameters and the data.

As the likelihood in our model, we assume that the incidence of strokes in smokers and non-smokers each follows a binomial distribution with unknown probabilities, θ_+ and θ_-, respectively. The prior distributions for θ_+ and θ_- are Beta distributions with parameters (1,1), which are non-informative priors that reflect the lack of prior knowledge about the probabilities. The model will also compute the relative risk of stroke in smokers compared to non-smokers, which is the ratio of the probabilities of stroke in the two groups. In this project, MCMC samples were obtained by using 10,000 iterations, which included a 2,000 iteration of burn-in period, thinning=1, and we use two chains (n.chains=2). For details on the code, please refer to the Appendix 1.

5. Check convergence of the MCMC chain.

To check the convergence and mixing of the MCMC samples and ensure that the results are reliable, we, firstly, analyse the trace plots visually. Figure A.3 below displays three nice plots with the centre of the chains showing one horizontal band which suggests that the chains have converged and are exploring the posterior distribution in a consistent manner.

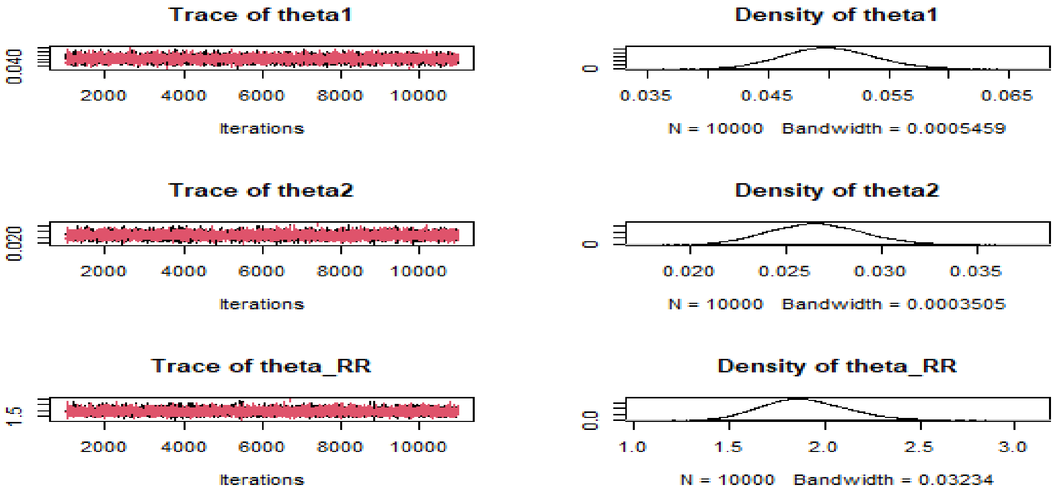

Figure A.3. The trace plots of the samples from the posterior distribution

(In this Figure, theta 1 means θ_+, theta 2 means θ_-, theta_RR means θ_{RR})

To go along with visual inspection, we utilise Gelman-Rubin and Geweke Convergence Diagnostic Tests. The gelman.diag() function computes the potential scale reduction factor (PSRF) for each parameter as a measure of convergence of multiple chains. As can be seen in Table A.4, all PSRF values are equal to 1, suggesting that the chains have converged.

Indicator	Point est.	Upper C.I.
θ_+	1	1
θ_-	1	1
θ_{RR}	1	1
Multivariate psrf	1	1

Table A.4. Potential Scale Reduction Factors (Gelman & Rubin's Statistics)

The plots of Geweke's Diagnostic (Figure A.4) also show that only there is a low number of scores falling outside the interval, that may suggest good convergence.

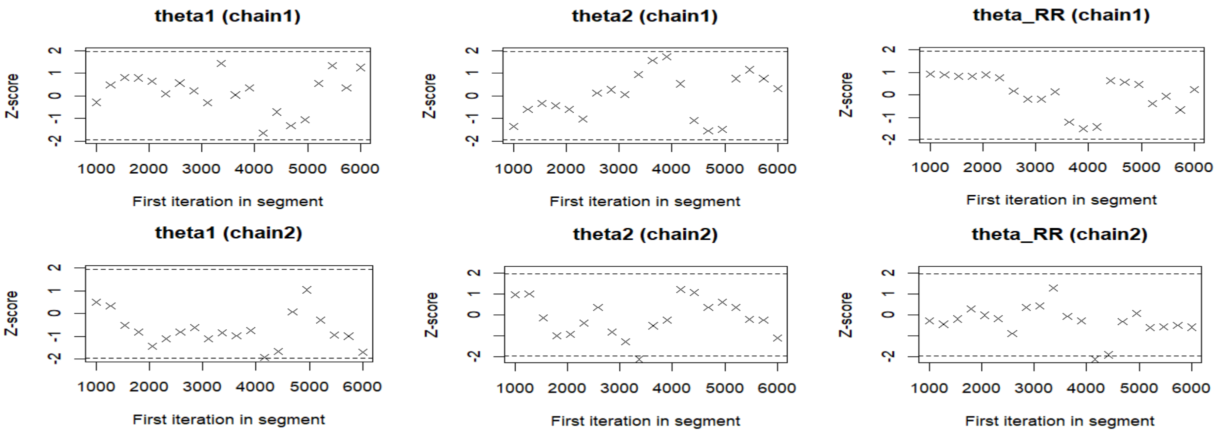

Figure A.4. Geweke's diagnostics

6. Compare the summary measures obtained from the MCMC chain with the results obtained from questions (1)-(3).

Table A.5 presents the summary measures obtained from questions (1)-(3) and also from the MCMC chain. The measures from the MCMC chain were computed using 10,000 iterations, which included a 2,000 iteration burn-in period. As shown in the table, the summary measures of the three parameters from the MCMC chain closely match the values obtained through the analytical solution. Based on the MCMC chain solution, we can also infer a significant association between smoking and the incidence of stroke.

Source	Indicator	Mean	Median	SD	95% Equal-Tail CI	95% HPD-Interval
Analytical Solution (Question 1-3)	θ_+	0.0500	0.0499	0.0037	[0.0430, 0.0575]	[0.0425, 0.0566]
	θ_-	0.0265	0.0265	0.0024	[0.0220, 0.0315]	[0.0218, 0.0310]
	θ_{RR}	1.9008	1.8867	0.2274	[1.5066, 2.3572]	[1.4896, 2.3417]
MCMC	θ_+	0.0501	0.0500	0.0037	[0.0431, 0.0577]	[0.0431, 0.0577]
	θ_-	0.0266	0.0265	0.0024	[0.0220, 0.0316]	[0.0219, 0.0315]
	θ_{RR}	1.9003	1.8840	0.2259	[1.5040, 2.3851]	[1.4731, 2.3442]

Table A.5. Comparison of The Summary Measures of The Analytical Solution and MCMC Chain

7. What is the attributable risk of smoking to the incidence of stroke? The attributable risk is defined as $\theta_{AR} = (\theta_{RR}–1)/\theta_{RR}$. Extend your Bayesian MCMC code to derive the answer.

We have expanded the code for the previous questions to also calculate the attributable risk of smoking on the incidence of stroke, θ_{AR}, as demonstrated in the appendix. Figure A.5 illustrates the trace plot with a horizontal band, which suggests convergence, and it also displays the posterior distribution of θ_{AR}.

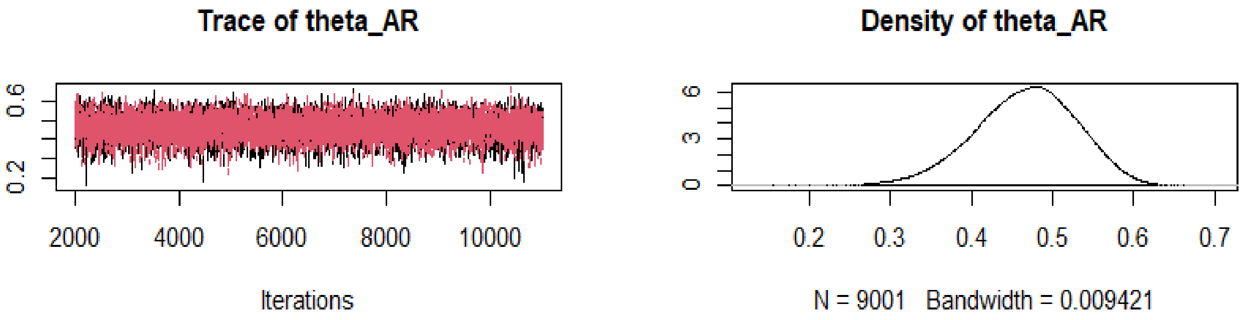

Figure A.5. Trace and Density Of The MCMC Chain For $\boldsymbol{\theta_{AR}}$

The result of Gelman and Rubin diagnostic test also indicates good convergence since the result is less than 1,1 as can be seen in Table A.6 below.

Indicator	Point est.	Upper C.I.
θ_{AR}	1	1

Table A.6. Potential Scale Reduction Factors (Gelman & Rubin's Statistics)

Furthermore, Table A.7 presents a summary statistics for the θ_{AR}.

Indicator	Mean	Median	SD	95% Equal-Tail CI	95% HPD-Interval
θ_{AR}	0.4644	0.4674	0.0624	[0.3312, 0.5763]	[0.3411, 0.5843]

Table A.7. The summary statistics of the Attributable Risk (AR) based on MCMC samples

As the attributable risk can be expressed by the formula $(\theta_{+}$ - $\theta_{-})/\theta_{+}$, it represents the proportion of disease incidence in the exposed group (smokers) that can be directly attributed to the exposure of smoking. Based on our results, it can be inferred that approximately 47% of the stroke cases among smokers can be directly attributed to smoking.

Executive Summary of PROJECT 2.2 (Cummulative Incidence)

1. To estimate region-specific incidence of COVID-19 positively tested individuals, the **Binomial likelihood** is used. And then, the **Uniform(0,1) prior** is chosen as a non-informative prior for the incidence. We use NIMBLE to solve this problem.

2. There are indications of **non-convergence and autocorrelation problems** before the model is updated. However, **by applying burn-in and thinning in the updated model**, those **problems can be solved** as can be seen from the new history plots, autocorrelation plots, and the Gelman and Rubin convergence diagnostic.

3. The **region-specific incidence** of COVID-19 positive tests (p) in New York City, Westchester/Rockland Counties, Long Island, and the Rest of NYS **is 0.22, 0.14, 0.12, and 0.03 respectively**. Using 95% Credible Interval of incidence difference, we find that all pairs of regions have significantly different incidence, **except for Westchester/Rockland Counties and Long Island,** whose **incidence are statistically not different**.

4. Based on the information provided, we choose the prior for sensitivity (Se) as $\boldsymbol{S_e \sim Beta(205, 29)}$**,** meanwhile for specificity (Sp) is $\boldsymbol{S_p \sim Beta(288,\ 2)}$

5. The region-specific cumulative incidences of COVID-19 positive tests (π) can be determined by utilizing the above priors of specificity and sensitivity. The model built has no problem of convergence and autocorrelation. Based on it, we find that the **highest cumulative incidence is in New York City (0.24),** followed by Westchester/Rockland Counties (0.15), Long Island (0.13), and **lastly Rest of NYS (0.03).**

PROJECT 2.2 (Cummulative Incidence)

1. Write out a Bayesian model (likelihood and priors) to estimate the region-specifc incidence. Use jags, OpenBugs or Nimble to solve this problem. Use vague priors to express that you do not have any prior knowledge. Run 2 chains using 2 different sets of initial.

- **The Likelihood**

The data contains the number of individuals tested *N* and the number of COVID-19 positively tested individuals Z, in 4 different regions of New York. We express the probability of *z* successes (having positive test result) out of *n* tests as a Binomial distribution:

$$f_p(z_i) = \binom{n_i}{z_i} {p_i}^{z_i}(1 - p_i)^{n_i - z_i}$$

with: z_i= the number of positive Covid-19 test in the *i*-th region,

p_i= the incidence/prevalence of an individual to have Covid-19 positive test in the *i*-th region,

n_i= the number of individuals tested in the *i*-th region

i = regions, in this case *i* is ranging from 1 to 4 reflecting the regions in New York (1 = New York City, 2 = Westchester/Rockland Counties, 3 = Long Island, 4 = Rest of NYS).

When the data is provided (z_i is given), the region-specific incidence p_i can be considered to have a Binomial likelihood function $L(p_i| z_i)$ as follows:

$$L(p_i|z_i) = \binom{n_i}{z_i} {p_i}^{z_i}(1 - p_i)^{n_i - z_i}$$

- **The Prior**

We use vague priors to express that we do not have any prior knowledge. Thus, we define $p_i \sim \text{Uniform}(0,1)$. This means that we assign equal probability to all possible values of p_i (between 0 and 1). The Uniform prior is indeed often considered as an uninformative prior because it does not favour any particular value of p_i over another.

- **The posterior**

Since the likelihood in our case follows Binomial distribution which has the same kernel as the Beta distribution, and also the prior that we choose is $Uniform(0,1)$ which also basically is a Beta distribution $Beta(1,1)$, so, using the conjugacy approach, the posterior will also be following a Beta distribution. The posterior $f(p_i|z_i)$ now can be expressed as:

$$f(p_i|z_i) = \frac{1}{B(z_i+1, n_i - z_i + 1)} {p_i}^{z_i}(1-p_i)^{n_i - z_i}$$

with $B(z_i+1, n_i - z_i + 1)$ is the Beta distribution with first shape parameter of $(z_i + 1)$ and second shape parameter parameter of $(n_i - z_i + 1)$.

- **The result of running 2 chains with 2 different sets of initial value using NIMBLE**

We choose the 2 extreme initial values (p=0 and p=1). Using NIMBLE, we fit the Bayes model in order to estimate the region-specific incidence p_i. We run 2 chains using the above sets of initial values to get sample from 5,000 iterations (we refer this as the First Model). No burn-in and thinning is applied in this First Model.

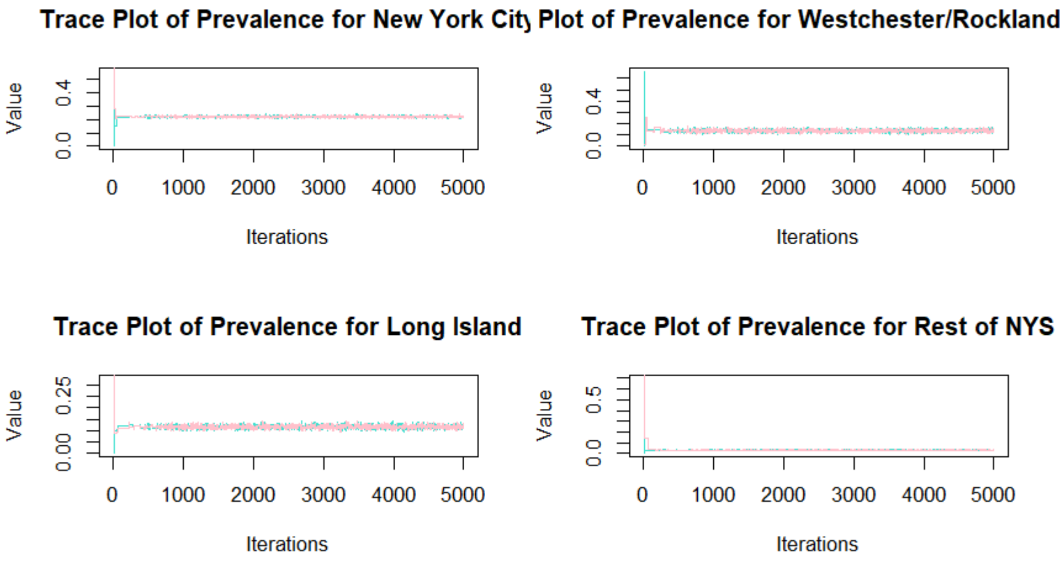

Figure B.1. Trace Plots of the First Model

Figure B.1 above are the trace plots for the sample of each regions' posterior incidence from the First Model (the turquoise denotes samples from chain-1, whereas the pink is for chain-2). Such plots allow us to assess the convergence of the Markov Chain Monte Carlo

(MCMC) samples, thus also to diagnose potential problems such as non-convergence in the posterior samples. We see that all of our trace plots do not show convergence, especially in the beginning of iterations.

We also compute the Gelman and Rubin convergence diagnostic. It tells us the comparison of the within-chain variance to the between-chain variance of a posterior samples and it can also be used as a tool to diagnose convergence. The result can be seen in Table B.1 below which, however, indicates reasonable convergence since the values are lower than 1.1.

Parameter	Point Estimate	Upper C.I.
p[1]	1	1.00
p[2]	1	1.01
p[3]	1	1.02
p[4]	1	1.01

Table B.1. The Gelman and Rubin Convergence Diagnostic Value from the First Model

Next, we also check for the autocorrelation plots of prevalence in the four regions. The result can be seen in Figure B.2 below.

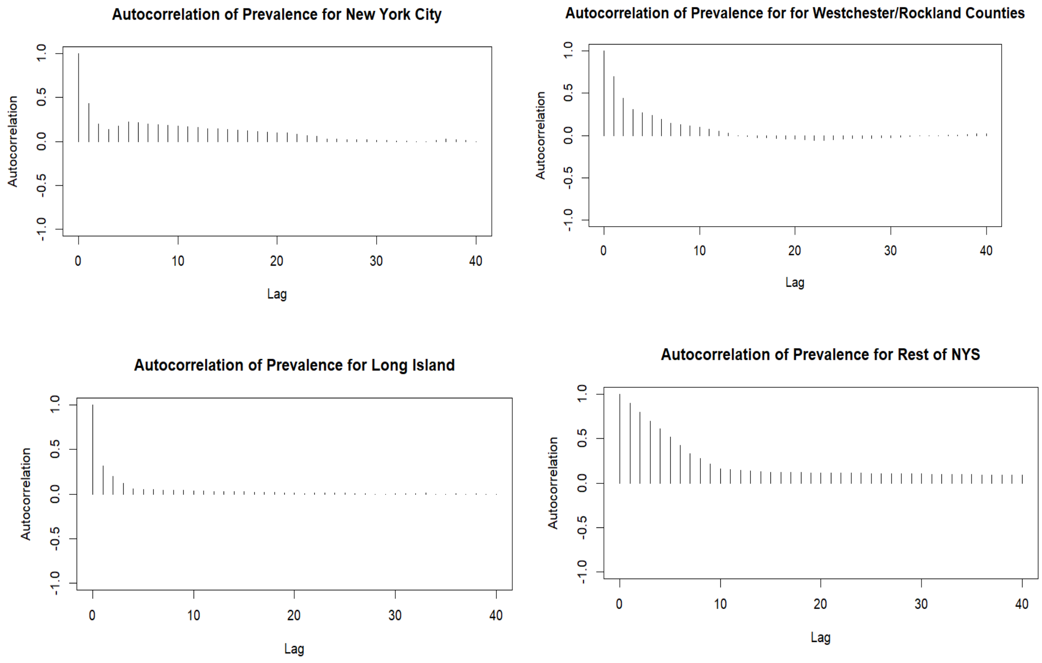

Figure B.2. The Autocorrelation Plot of The First Model

Ideally, the autocorrelation plot should show a rapid decay as the lag increases. However, based on the figure B.2, the condition does not hold, thus there are indications of severe autocorrelations. To solve this problem, we may require burn-in and/or thinning.

2. Run 5000 iterations (updates), then look at history plots and autocorrelation plots of the sample traces and calculate the Gelman and Rubin convergence diagnostic (GR diag) for each of the parameters you have monitored. Do the simulations look like they have converged? If not, carry out some more updates and check again, until you are happy with convergence.

- **The updated model**

Based on analysis in Question 1, it can be concluded that we need to apply burn-in and thinning in order to obtain better convergence and solve the autocorrelation problem. We update the model (we call this as Second Model) by applying 1,000 burn-in, increasing the number of iteration to be 16,000 (in order to have 5,000 iteration after the burn-in), and also applying the thinning of 3. Then, we check again the trace plots and autocorrelation plots for this updated model.

- **The diagnostic of convergence**

Based on the Figure B.3 below, the Second Model seems to have better convergence and mixing.

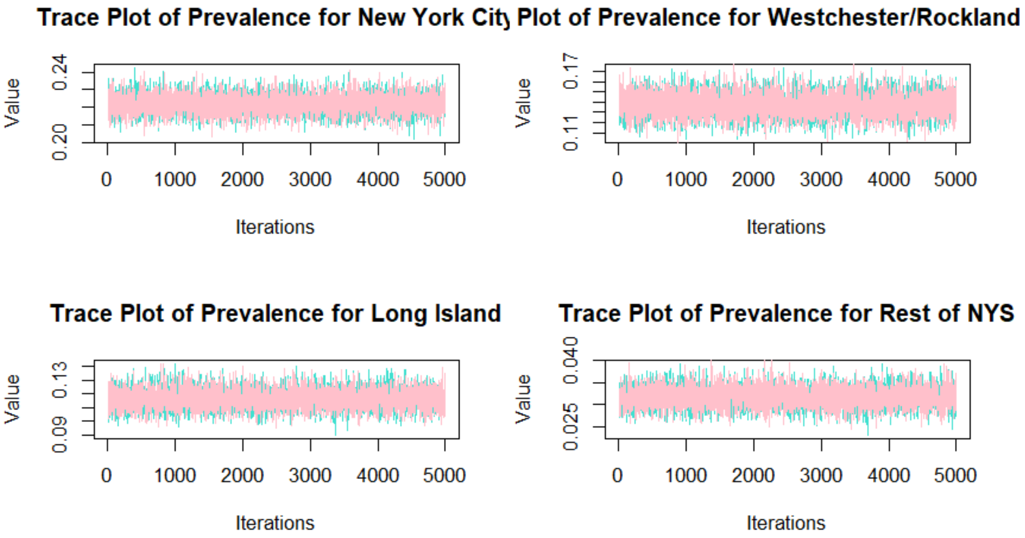

Figure B.3. The Trace Plots of the Second Model

The better convergence of the Second Model is also demonstrated by the Gelman and Rubin diagnostic value as can be seen in Table B.2 below. The values for all the regions are lower than 1.1 thus implying convergence, and good mixing of chains.

Parameter	Point Estimate	Upper C.I.
p[1]	1	1.01
p[2]	1	1.00
p[3]	1	1.00
p[4]	1	1.00

Table B.2. The Gelman and Rubin Convergence Diagnostic Value from the Second Model

- **The autocorrelation**

The autocorrelation plot for the Second Model can be seen in Figure B.4 below, which indicates improvement from the First Model because the autocorrelation decreases faster after only a few lags.

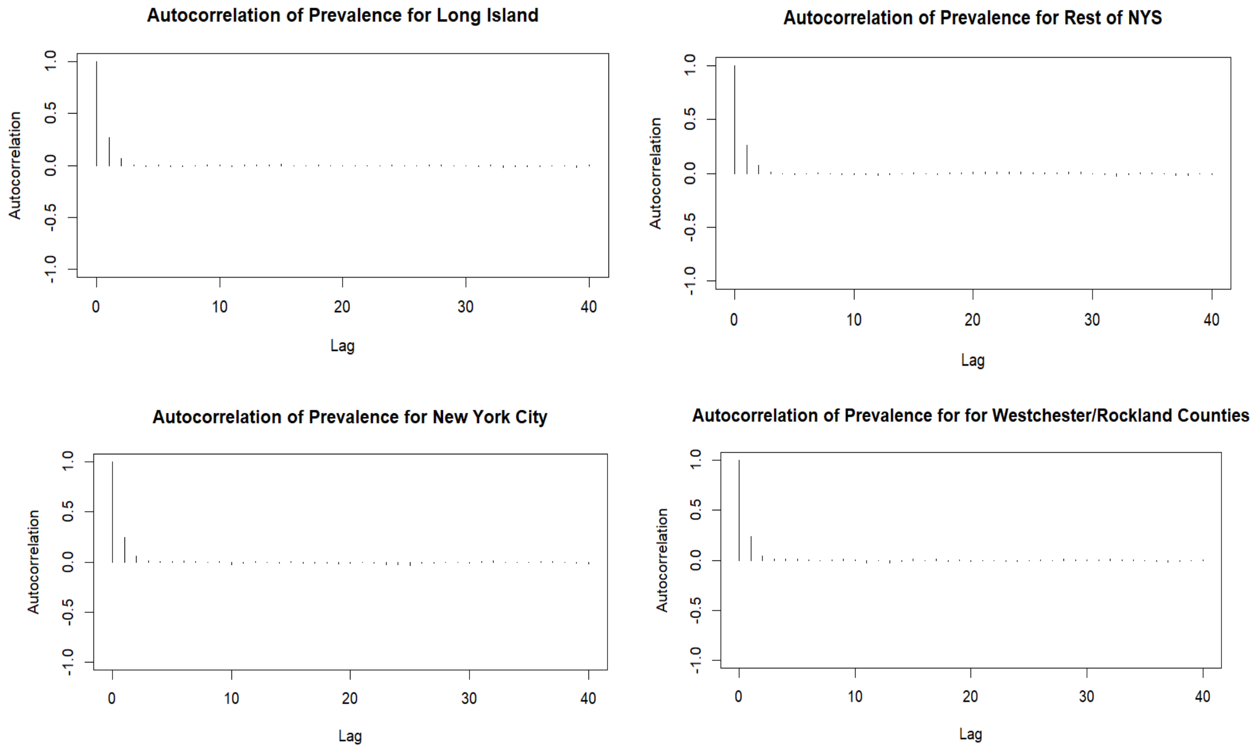

Figure B.4. The Autocorrelation Plots of the Second Model

3. Determine the prevalence *p* based on this survey, and determine whether there are any differences amongst the regions.

- **The region-based prevalence**

Since the Second Model has converged well and has much less problem of autocorrelation, then we will use it to compute the region-based prevalence. The result can be seen in Table B.3 below. It can be seen that for all areas, the prevalence is lower than 0.300. The highest is found in New York City with its prevalence of 0.222, meanwhile the lowest is in the Rest of NYS with its prevalence of 0.032.

No	Region	Prevalence
1	New York City	0.222
2	Westchester/Rockland Counties	0.137
3	Long Island	0.117
4	Rest of NYS	0.032

Table B.3. The Region-Based Prevalence Based On The Second Model

- **Checking difference of prevalence between regions**

We will continue to check the difference between pairs of regions using 95% Credible Interval of their prevalence difference. The result is summarized in the Table B.4 below.

Region	WESTCHESTER/ ROCKLAND COUNTIES		LONG ISLAND		REST OF NYS	
	95% Credible Intervals		95% Credible Intervals		95% Credible Intervals	
	Lower	Upper	Lower	Upper	Lower	Upper
NEW YORK CITY	**Equal Tail**		**Equal Tail**		**Equal Tail**	
	0.059	0.108	0.088	0.123	0.179	0.202
	HPD		**HPD**		**HPD**	
	0.059	0.107	0.087	0.123	0.178	0.201
WESTCHESTER/ ROCKLAND COUNTIES	-		**Equal Tail**		**Equal Tail**	
			-0.004*	0.047*	0.084	0.128
			HPD		**HPD**	
			-0.005*	0.047*	0.084	0.128
LONG ISLAND	-		-		**Equal Tail**	
					0.070	0.100
					HPD	
					0.071	0.100

*Significantly different

Table B.4. The 95% Credible Interval of Prevalence Difference Between Each Pair of Regions

Based on Table B.4, it can be seen that 0 (zero) is not included in the 95% Credible Interval (CI) of prevalence difference for all pairs of regions, except for Westchester/Rockland Counties versus Long Island that includes 0 (zero). Thus, it can be concluded that all pairs of regions in New York have different prevalance, except for the Westchester/Rockland Counties versus the Long Island, whose data do not show enough proof to say that those regions are statistically different (using either the 95% Equal Tail CI or 95% HPD CI).

4. The sensitivity of the antibody test was determined based on an assay in 232 COVID-19 patients, of which 204 tested positive. The specificity of the antibody test was based on two surveys: one in which 195 out of 196 healthy individuals tested negative, and one in which 92 out of 92 healthy individuals tested negative. Translate this information as good as possible in a prior distribution for sensitivity Se and specificity Sp.

If TP, FP, TN, and FN denote the number of true positives, false positives, true negatives, and false negatives from an experiment respectively, then the Sensitivity (S_e) and Specificity (S_p) can be defined as : $S_e = \frac{TP}{TP+FN}$ and $S_p = \frac{TN}{TN+FP}$.

The sensitivity and specificity analysis can be viewed as a binomial experiment. Furthermore, the Beta distribution can be used to describe such binomial experiment. A binomial experiment with y success from n experiments can be interpreted into a prior which follows a Beta distribution with first shape parameter of $y+1$ and second shape parameter paramater of $n-y+1$ or can be denoted as $Beta(y+1, n-y+1)$.

- **The sensitivity**

Based on a survey, in 232 COVID-19 patients, 204 patients are tested positive. Thus, it can be expressed as a binomial experiment with success $y = 204$ from $n = 232$ experiments. In other words, the prior for sensitivity can be defined as Beta distribution with first shape parameters of $y+1 = 204+1 = 205$ and second shape parameter paramater of $n-y+1 = 232-204+1 = 29$ or can be denoted as $S_e \sim Beta(205, 29)$.

- **The specificity**

In a survey of 196 healthy individuals, 195 individuals are tested negative. In another survey of 92 healthy individuals, 92 individuals are tested negative. We will use the information provided in the first survey as a prior for the second survey or in other words, the second survey can be treated as the extra data. If we combine our prior belief from first survey with the data from second survey, we will gain a posterior distribution, which in turns can act as a prior distribution for Specificity in our current study.

From the first survey, the information can be seen as binomial experiment with success $y = 195$ from $n = 196$ experiments. In other words, it can be considered as a prior that follows a Beta distribution with first shape parameters of $y + 1 = 195 + 1 = 196$ and second shape parameter paramater of $n - y + 1 = 196 - 195 + 1 = 2$ or can be denoted as $Beta(\alpha_0 = 196, \beta_0 = 2)$. Meanwhile, the second survey can be seen as binomial experiment with success $y = 92$ from $n = 92$ experiments.

Combining the prior (from survey 1) with the data (from survey 2) will result in a posterior that follows Beta distribution with first shape parameter of $\alpha_0 + y$ (equal to $196 + 92 = 288$) and second shape parameter paramater of $\beta_0 + n - y$ (equal to $2 + 92 - 92 = 2$). The obtained posterior distribution (from combining these 2 surveys) can now be treated as the prior of specificity for our current study which then can be denoted as $S_p \sim B(288,\ 2)$. Figure B.5 below visualizes the Beta distribution that will act as the prior for sensitivity and specificity for our current study along with its mode (in purple).

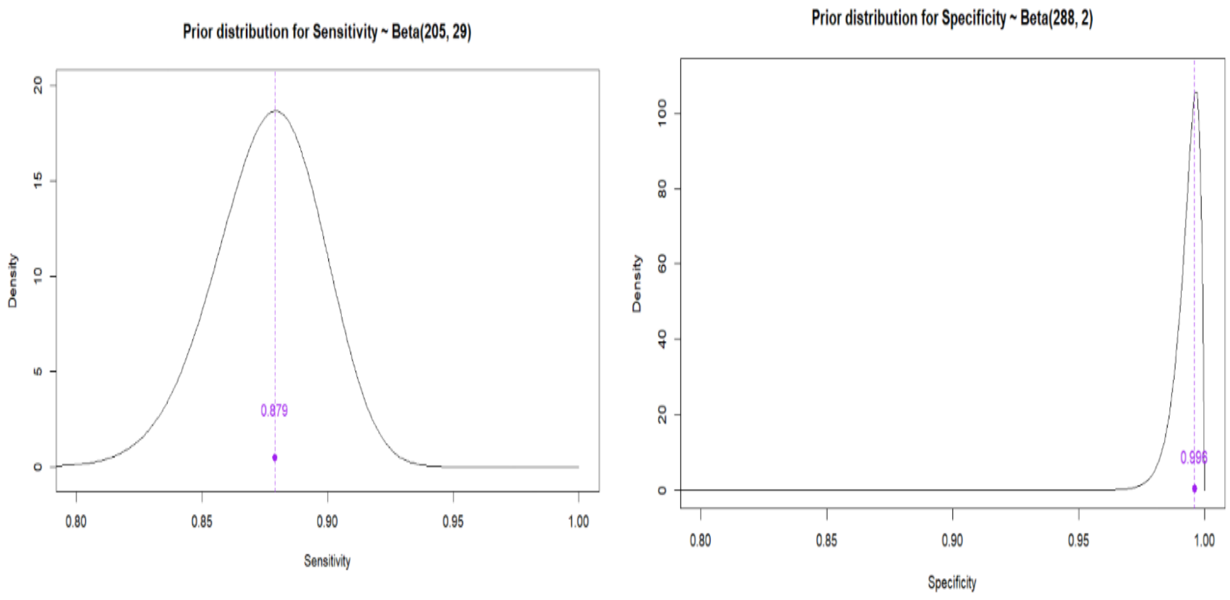

Figure B.5 The Visualization of Prior Distributions for Sensitivity and Specificity

5. Determine the posterior of the region-specific cumulative incidences, defined as $\pi = \frac{p+S_p-1}{S_e+S_p-1}$, with p the observed region-specific incidence, Sp the specificity and Se the sensitivity?

- **Check the convergence and autocorrelation of the model**

In this final part, we will determine the posterior distribution of the region-specific cumulative indices $\pi = \frac{p+S_p-1}{S_e+S_p-1}$ by utilizing the the priors for Specificity and Sensitivity from Question 4. The cumulative incidence can be interperted as the estimate of risk that an individual will experience an event or develop a disease during a specified period of time. To compute it based on our data, we will use NIMBLE again. In this Third Model, we set the initial values for all the parameters as 0 for the first chain, and as 1 for the second chain. We also apply the same specification like in the Second Model which is: 1,000 burn-in; 16,000 number of iteration (in order to have 5,000 iteration after the burn-in), and thinning of 3.

We will also check the convergence and autocorrelation for this Third Model. The trace plot indicates that the posterior samples converge well for all regions in all chains (as can be seen in Figure B.6 below) and also the Gelman and Rubin diagnostics indicates the convergence too (as can be viewed in Table B.5 below).

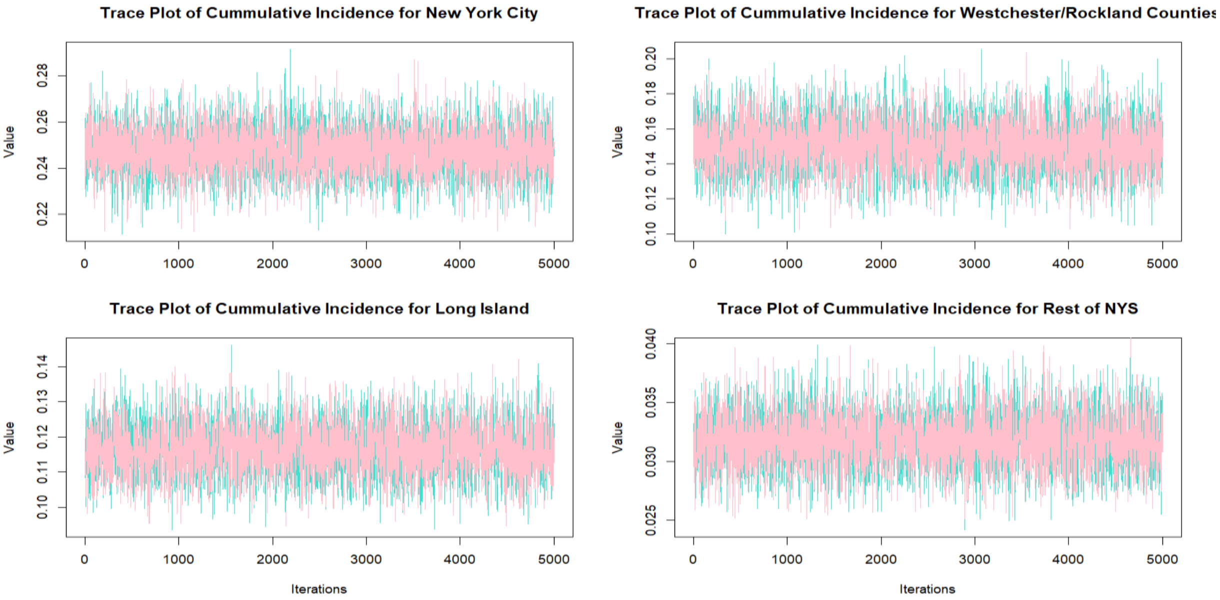

Figure B.6 The Traceplot for The Third Model

Parameter	Point Estimate	Upper C.I.
pi[1]	1	1.00
pi[2]	1	1.00
pi[3]	1	1.01
pi[4]	1	1.00

Table B.5. The Gelman and Rubin Convergence Diagnostic Value For The Third Model

Meanwhile, the autocorrelation plot in Figure B.7 below also shows that there isn't much autocorrelation problem, especially after lag 2 in all the regions. Therefore, we can conclude that we can trust the posterior derived from this Third Model.

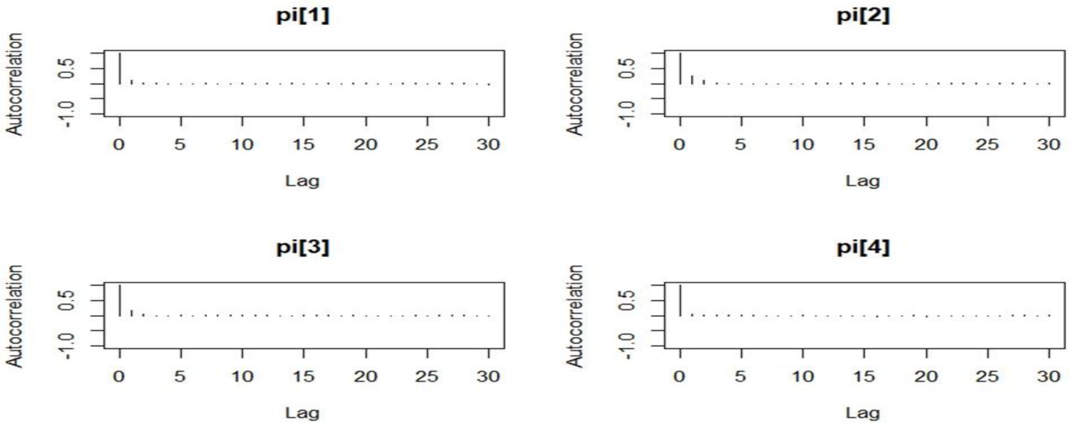

Figure B.7 The Autocorrelation for The Third Model

As the final result of this Third Model, we get the following posterior density plots for the cumulative incidences in each region, as displayed by Figure B.8 below.

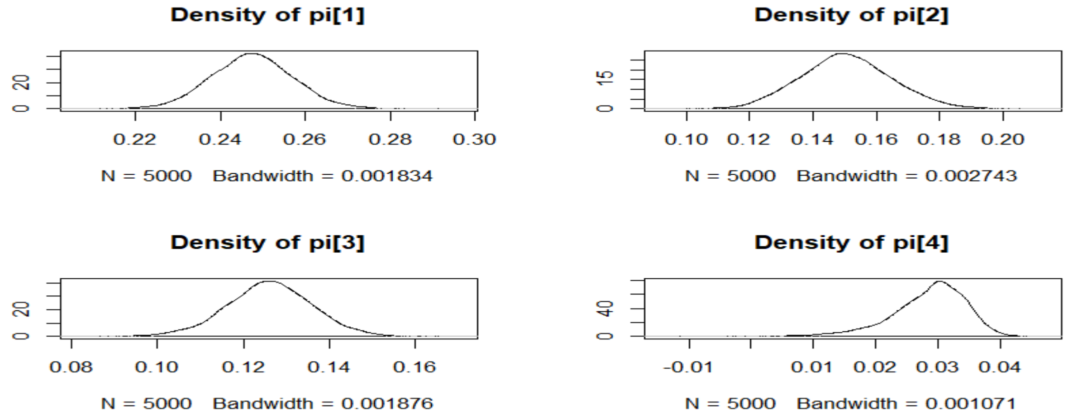

Figure B.8. The Density Plot of Posterior Distribution of Cumulative Incidences in Each Region

- **The result of region-specific cumulative incidences**

The following Table B.6 displays the means of the posterior of the region-specific cumulative incidences in each region. It informs us that the highest cumulative incidences can be found in New York City (0.247), meanwhile the lowest is in the Rest of NYS (0.028). The other regions which are Westchester/Rockland Counties and Long Island show cumulative incidences of 0.150 and 0.126 respectively.

No	Region	Prevalence
1	New York City	0.247
2	Westchester/Rockland Counties	0.150
3	Long Island	0.126
4	Rest of NYS	0.028

Table B.6 The Region-Specific Cumulative Incidences Based On Third Model

APPENDIX

Code For Project 1.1. (Cohort Study Smoking)

```
# Bayesian Inference Project A

# Question 1
library(HDInterval)


# Define variables
n_smoker <- 3435 # The total number of smokers
x_smoker <- 171  # The number of smokers who had a stroke
n_non_smoker <- 4437 # The total number of non-smokers
x_non_smoker <- 117 # The number of non-smokers who had a stroke

# Define prior parameters with unit prior, Beta(1,1)
alpha_prior <- 1
beta_prior <- 1

# Posterior parameters, for ϑ+ and ϑ-
alpha_post_smoker <- alpha_prior + x_smoker
beta_post_smoker <- beta_prior + n_smoker - x_smoker
alpha_post_non_smoker <- alpha_prior + x_non_smoker
beta_post_non_smoker <- beta_prior + n_non_smoker - x_non_smoker

# Define density functions
prior_density <- function(x) dbeta(x, alpha_prior, beta_prior)
post_smoker_density <- function(x) dbeta(x, alpha_post_smoker, beta_post_smoker)
post_non_smoker_density <- function(x) dbeta(x, alpha_post_non_smoker, beta_post_non_smoker)

# Define summary measures for the posterior distribution
# for smoker, ϑ+
post_smoker_mean <- alpha_post_smoker / (alpha_post_smoker + beta_post_smoker)
post_smoker_median <- qbeta(0.5, alpha_post_smoker, beta_post_smoker)
post_smoker_mode <- (alpha_post_smoker - 1) / (alpha_post_smoker + beta_post_smoker - 2)
post_smoker_sd <- sqrt((alpha_post_smoker * beta_post_smoker) / ((alpha_post_smoker + beta_post
_smoker)^2 * (alpha_post_smoker + beta_post_smoker + 1)))
post_smoker_ci <- qbeta(c(0.025, 0.975), alpha_post_smoker, beta_post_smoker)
post_smoker_hpd <- hdi(rbeta(1000, alpha_post_smoker, beta_post_smoker))

# for non-smoker, θ-
post_non_smoker_mean <- alpha_post_non_smoker / (alpha_post_non_smoker + beta_post_non_smoker)
post_non_smoker_median <- qbeta(0.5, alpha_post_non_smoker, beta_post_non_smoker)
post_non_smoker_mode <- (alpha_post_non_smoker - 1) / (alpha_post_non_smoker + beta_post_non_sm
oker - 2)
post_non_smoker_sd <- sqrt((alpha_post_non_smoker * beta_post_non_smoker) / ((alpha_post_non_sm
oker + beta_post_non_smoker)^2 * (alpha_post_non_smoker + beta_post_non_smoker + 1)))
post_non_smoker_ci <- qbeta(c(0.025, 0.975), alpha_post_non_smoker, beta_post_non_smoker)
post_non_smoker_hpd <- hdi(rbeta(1000, alpha_post_non_smoker, beta_post_non_smoker))


# Print summary measures
# for Smoker, θ+
cat("Smoker:\n")
cat("Mean:", post_smoker_mean, "\n")
cat("Median:", post_smoker_median, "\n")
cat("Mode:", post_smoker_mode, "\n")
cat("Standard Deviation:", post_smoker_sd, "\n")
cat("95% Credible Interval:", '[',post_smoker_ci[1], ',', post_smoker_ci[2],']', "\n")
cat("95% HPD:", '[',post_smoker_hpd[1], ',', post_smoker_hpd[2],']\n')
# for non-smoker, θ-
cat("Non-Smoker:\n")
cat("Mean:", post_non_smoker_mean, "\n")
cat("Median:", post_non_smoker_median, "\n")
cat("Mode:", post_non_smoker_mode, "\n")
cat("Standard Deviation:", post_non_smoker_sd, "\n")
cat("95% Credible Interval:", '[',post_non_smoker_ci[1],',',post_non_smoker_ci[2],']', "\n")
cat("95% HPD:", '[',post_non_smoker_hpd[1], ',', post_non_smoker_hpd[2],']\n')

# Set up the plot layout to have 1 row and 2 columns
par(mfrow = c(1, 2))
# Plot for smoker, posterior distribution of θ+
plot(0, 0, type = "n", xlim = c(0, 0.1), ylim = c(0, 180),
     xlab = "Probability of Disease", ylab = "Density",
     main = "Posterior Distribution of θ+")
curve(post_smoker_density, add = TRUE, col = "blue", lwd = 2)
abline(v = post_smoker_mean, col = "red", lwd = 2, lty = 2)

# Plot for non-smoker, posterior distribution of  θ-
plot(0, 0, type = "n", xlim = c(0, 0.1), ylim = c(0, 180),
     xlab = "Probability of Disease ", ylab = "Density",
     main = "Posterior Distribution of θ-")
curve(post_non_smoker_density, add = TRUE, col = "blue", lwd = 2)
abline(v = post_non_smoker_mean, col = "red", lwd = 2, lty = 2)


# Question 3
# Draw samples from posterior distributions
n_samples <- 1000
smoker_samples <- rbeta(n_samples, alpha_post_smoker, beta_post_smoker)
non_smoker_samples <- rbeta(n_samples, alpha_post_non_smoker, beta_post_non_smoker)

# Calculate relative risk
rel_risk_samples <- smoker_samples / non_smoker_samples

# Plot posterior distribution of relative risk
par(mfrow = c(1, 1))
hist(rel_risk_samples, breaks=100, main = "", xlab="Relative Risk",
     col = "lightblue", border = "black")
abline(v=mean(rel_risk_samples), col="red", lwd = 2, lty = 2)  # Line at RR=1

# Calculate and print summary measures
# Mode function to calculate the mode
Mode <- function(x) {
  ux <- unique(x)
  ux[which.max(tabulate(match(x, ux)))]
}
cat("Mean Relative Risk:", mean(rel_risk_samples), "\n")
cat("Median Relative Risk:", median(rel_risk_samples), "\n")
cat("Mode Relative Risk:", Mode(rel_risk_samples), "\n")  # Mode function to calculate the mode
cat("Standard Deviation of Relative Risk:", sd(rel_risk_samples), "\n")  # Calculate standard d
eviation
cat("95% Credible Interval for Relative Risk:", '[',quantile(rel_risk_samples, c(0.025, 0.975))
,']\n')
cat("95% HPD for Relative Risk:", '[',hdi(rel_risk_samples),']\n')
```

```r
# Question 4, 5, 6
library("rjags")

# Data
model.data = list(
  x1 = 171,
  n1 = 3435,
  x2 = 117,
  n2 = 4437
)

model.inits <- list(theta1 = 0.5, theta2 = 0.5)

# Model
modelString = "
model {
  # Priors
  theta1 ~ dbeta(1, 1)
  theta2 ~ dbeta(1, 1)

  # Likelihood
  x1 ~ dbin(theta1, n1)
  x2 ~ dbin(theta2, n2)

  # Relative risk
  theta_RR <- theta1 / theta2
}
"

# specify model, data, number of parallel chains
jags <- jags.model(textConnection(modelString),
                   data = model.data,
                   inits = model.inits,
                   n.chains = 2)

# Generate MCMC samples and save output for specified variables
out <- coda.samples(jags,
                    c("theta1", "theta2", "theta_RR"),
                    n.iter=10000, thin=1)

# Posterior summary statistics
burnin <- 2000
summary(window(out,start=burnin))
# Iterations = 2000:11000
# Thinning interval = 1
# Number of chains = 2
# Sample size per chain = 9001
#
# 1. Empirical mean and standard deviation for each variable,
# plus standard error of the mean:
#
#   Mean       SD  Naive SE Time-series SE
# theta1   0.05008 0.003714 2.768e-05      3.481e-05
# theta2   0.02658 0.002436 1.816e-05      2.364e-05
# theta_RR 1.90032 0.225936 1.684e-03      2.131e-03
#
# 2. Quantiles for each variable:
```

```
#
#   2.5%     25%     50%     75%   97.5%
# theta1   0.04309 0.04753 0.04996 0.05249 0.05768
# theta2   0.02198 0.02491 0.02652 0.02817 0.03163
# theta_RR 1.50402 1.74229 1.88404 2.04298 2.38514

# HPD
out2.combined <- combine.mcmc(out2)
HPDinterval(out2.combined)
# lower      upper
# theta1   0.04306756 0.05772057
# theta2   0.02189134 0.03148368
# theta_RR 1.47313610 2.34421193

# Produce general summary of obtained MCMC sampling,
# History plot & posterior distributions & autocorrelation plot
print(out,digits=3)
plot(out, trace=TRUE, density = TRUE)
plot(window(out,start=burnin), trace=TRUE, density = TRUE)

# Convergence tests
out.mcmc <- as.mcmc.list(out)
gelman.diag(out.mcmc)
gelman.plot(out.mcmc,ask=FALSE)

geweke.diag(out.mcmc)
geweke.plot(out.mcmc,ask=FALSE)

# Question 7
# Model
modelString2 = "
model {
  # Priors
  theta1 ~ dbeta(1, 1)
  theta2 ~ dbeta(1, 1)

  # Likelihood
  x1 ~ dbin(theta1, n1)
  x2 ~ dbin(theta2, n2)

  # Relative risk
  theta_AR <- 1 - theta2 / theta1
}
"

# specify model, data, number of parallel chains
jags2 <- jags.model(textConnection(modelString2),
                    data = model.data,
                    inits = model.inits,
                    n.chains = length(model.inits))

# Generate MCMC samples and save output for specified variables
out2 <- coda.samples(jags2,
                     c("theta1", "theta2", "theta_AR"),
                     n.iter=10000, thin=1)

# Posterior summary statistics
burnin2 <- 2000
```

```
summary(window(out2,start=burnin2))
# Iterations = 2000:11000
# Thinning interval = 1
# Number of chains = 2
# Sample size per chain = 9001
#
# 1. Empirical mean and standard deviation for each variable,
# plus standard error of the mean:
#
#   Mean       SD  Naive SE Time-series SE
# theta1   0.04997 0.003667 2.733e-05      3.316e-05
# theta2   0.02662 0.002400 1.789e-05      2.288e-05
# theta_AR 0.46438 0.062436 4.653e-04      5.819e-04
#
# 2. Quantiles for each variable:
#
#   2.5%     25%     50%     75%   97.5%
# theta1   0.04294 0.04749 0.04989 0.05244 0.0573
# theta2   0.02219 0.02495 0.02653 0.02819 0.0315
# theta_AR 0.33122 0.42544 0.46740 0.50772 0.5763

# History plot & posterior distributions & autocorrelation plot
plot(out2, trace=TRUE, density = TRUE)
plot(window(out2,start=burnin2), trace=TRUE, density = TRUE)

# HPD
out2.combined <- combine.mcmc(out2)
HPDinterval(out2.combined)
# lower      upper
# theta1   0.04292285 0.05727797
# theta2   0.02200795 0.03132428
# theta_AR 0.34114014 0.58434425

# Convergence tests
out2.mcmc <- as.mcmc.list(out2)
gelman.diag(out2.mcmc)
gelman.plot(out2.mcmc,ask=FALSE)

geweke.diag(out2.mcmc)
geweke.plot(out2.mcmc,ask=FALSE)
```

Code For Project 2.2. (Cumulative Incidence)

Contents

The packages that we use

```
library("nimble")
library("coda")
library("readr")
library(LearnBayes)
```

Question 1

```
N <- c(5946, 980, 2074, 6101)
z <- c(1319, 134, 241, 193)
model.dataZ <- list('z' = z)
model.dataN <- list('N' = N)

# Set 2 different initials for 2 chains
init1 <- list(p = 0) # 'extreme' choice for prevalence
init2 <- list(p = 1) # 'extreme' choice for prevalence
model.initials <- list(init1, init2)
#other ways:
# model.initials2<- list(list(p = a),
#                        list(p = b)
#                        )

# Model specification
ModelPrevalence <- nimbleCode(
  {
    # likelihood specification : binomial
    for (i in 1:length(N)) {
      z[i] ~ dbinom(p[i], N[i])
    }

    # prior information : vague one
    for (i in 1:length(N)) {
      p[i] ~  dunif(0, 1)
    }

    # derived quantity : nothing
    # .... can put something here if we have it;
    # like a link function or etc

  })

# Run 2 chains using 2 different sets of initial; 5000 iterationns (no burn-in yet)
OutputMCMC_q1 <- nimbleMCMC(
  code=ModelPrevalence,
  data=model.dataZ,
  constants=model.dataN,
  inits=model.initials,
  monitors=c("p"),
  niter=5000,
  nchains=2,
  #nburnin=0,
  setSeed=2023,
  summary=TRUE
  )

#Getting the chains
chain1_q1 <- OutputMCMC_q1$samples$chain1
chain2_q1 <- OutputMCMC_q1$samples$chain2

# History plots: p1 (each chain)
traceplot(as.mcmc(chain1_q1[,'p[1]']),
          col = "turquoise",
          main = 'Trace Plot of Prevalence for New York City',
          ylab = 'Value')
traceplot(as.mcmc(chain2_q1[,'p[1]']),
          add = TRUE,
          col = "pink")
```

```
#to check if chain1 and chain2 produce the same result: sum(chain1_q1 == chain2_q1)

# History plots: p2 (each chain)
traceplot(as.mcmc(chain1_q1[,'p[2]']),
          col = "turquoise",
          main = 'Trace Plot of Prevalence for Westchester/Rockland Counties',
          ylab = 'Value')
traceplot(as.mcmc(chain2_q1[,'p[2]']),
          add = TRUE,
          col = "pink")

# History plots: p3 (each chain)
traceplot(as.mcmc(chain1_q1[,'p[3]']),
          col = "turquoise",
          main = 'Trace Plot of Prevalence for Long Island',
          ylab = 'Value')
traceplot(as.mcmc(chain2_q1[,'p[3]']),
          add = TRUE,
          col = "pink")

# History plots: p4 (each chain)
traceplot(as.mcmc(chain1_q1[,'p[4]']),
          col = "turquoise",
          main = 'Trace Plot of Prevalence for Rest of NYS',
          ylab = 'Value')
traceplot(as.mcmc(chain2_q1[,'p[4]']),
          add = TRUE,
          col = "pink")

# Autocorrelation plots: p1 (each chain)
autocorr.plot(as.mcmc(
  c(chain1_q1[,'p[1]'], chain2_q1[,'p[1]'])),
  main = 'Autocorrelation of Prevalence for New York City')

# Autocorrelation plots: p2 (each chain)
autocorr.plot(as.mcmc(
  c(chain1_q1[,'p[2]'], chain2_q1[,'p[2]'])),
  main = 'Autocorrelation of Prevalence for for Westchester/Rockland Counties')

# Autocorrelation plots: p3 (each chain)
autocorr.plot(as.mcmc(
  c(chain1_q1[,'p[3]'], chain2_q1[,'p[3]'])),
  main = 'Autocorrelation of Prevalence for Long Island')

# Autocorrelation plots: p4 (each chain)
autocorr.plot(as.mcmc(
  c(chain1_q1[,'p[4]'], chain2_q1[,'p[4]'])),
  main = 'Autocorrelation of Prevalence for Rest of NYS')

#Gelman and Rubin's convergence diagnostics updated model
chain.combined_q1 <- mcmc.list(
  as.mcmc(chain1_q1[, c(1,2,3,4)]),
  as.mcmc(chain2_q1[, c(1,2,3,4)])
  )
gelman.diag(chain.combined_q1)
gelman.plot(chain.combined_q1)
```

Question 2

```
# Run MCMC and set the burn-in and thinning, in the end, save 5000 iterations
OutputMCMC_q2 <- nimbleMCMC(
  code=ModelPrevalence,
  data=model.dataZ,
  constants=model.dataN,
  inits=model.initials,
  monitors=c("p"),
  nchains=2,
  nburnin=1000,
  niter=16000,
  thin = 3,
  setSeed=2023,
  summary=TRUE
  )

#Getting the chains
chain1_q2 <- OutputMCMC_q2$samples$chain1
chain2_q2 <- OutputMCMC_q2$samples$chain2

# History plots: p1 (each chain)
traceplot(as.mcmc(chain1_q2[,'p[1]']),
          col = "turquoise",
          main = 'Trace Plot of Prevalence for New York City',
          ylab = 'Value')
traceplot(as.mcmc(chain2_q2[,'p[1]']),
          add = TRUE,
          col = "pink")
#to check if chain1 and chain2 produce the same result: sum(chain1_q2 == chain2_q2)

# History plots: p2 (each chain)
traceplot(as.mcmc(chain1_q2[,'p[2]']),
          col = "turquoise",
          main = 'Trace Plot of Prevalence for Westchester/Rockland Counties',
          ylab = 'Value')
traceplot(as.mcmc(chain2_q2[,'p[2]']),
          add = TRUE,
          col = "pink")

# History plots: p3 (each chain)
traceplot(as.mcmc(chain1_q2[,'p[3]']),
          col = "turquoise",
          main = 'Trace Plot of Prevalence for Long Island',
          ylab = 'Value')
traceplot(as.mcmc(chain2_q2[,'p[3]']),
          add = TRUE,
          col = "pink")

# History plots: p4 (each chain)
traceplot(as.mcmc(chain1_q2[,'p[4]']),
          col = "turquoise",
          main = 'Trace Plot of Prevalence for Rest of NYS',
          ylab = 'Value')
traceplot(as.mcmc(chain2_q2[,'p[4]']),
          add = TRUE,
          col = "pink")

# Autocorrelation plots: p1 (each chain)
autocorr.plot(as.mcmc(
  c(chain1_q2[,'p[1]'], chain2_q2[,'p[1]'])),
  main = 'Autocorrelation of Prevalence for New York City')

# Autocorrelation plots: p2 (each chain)
```

```
autocorr.plot(as.mcmc(
  c(chain1_q2[,'p[2]'], chain2_q2[,'p[2]'])),
  main = 'Autocorrelation of Prevalence for for Westchester/Rockland Counties')

# Autocorrelation plots: p3 (each chain)
autocorr.plot(as.mcmc(
  c(chain1_q2[,'p[3]'], chain2_q2[,'p[3]'])),
  main = 'Autocorrelation of Prevalence for Long Island')

# Autocorrelation plots: p4 (each chain)
autocorr.plot(as.mcmc(
  c(chain1_q2[,'p[4]'], chain2_q2[,'p[4]'])),
  main = 'Autocorrelation of Prevalence for Rest of NYS')

#Gelman and Rubin's convergence diagnostics updated model
chain.combined_q2 <- mcmc.list(
  as.mcmc(chain1_q2[, c(1,2,3,4)]),
  as.mcmc(chain2_q2[, c(1,2,3,4)])
  )
gelman.diag(chain.combined_q2)
gelman.plot(chain.combined_q2)
```

Question 3

```
#TESTING DIFFERENCE BETWEEN REGIONS
# Get summary statistics for p
(p1.mean <- OutputMCMC_q2$summary$all.chains['p[1]','Mean'])
(p2.mean <- OutputMCMC_q2$summary$all.chains['p[2]','Mean'])
(p3.mean <- OutputMCMC_q2$summary$all.chains['p[3]','Mean'])
(p4.mean <- OutputMCMC_q2$summary$all.chains['p[4]','Mean'])

# Preparation: Get MCMC samples for each p
p1.sample <- c(chain1_q2[,'p[1]'],
               chain2_q2[,'p[1]'])
p2.sample <- c(chain1_q2[,'p[2]'],
               chain2_q2[,'p[2]'])
p3.sample <- c(chain1_q2[,'p[3]'],
               chain2_q2[,'p[3]'])
p4.sample <- c(chain1_q2[,'p[4]'],
               chain2_q2[,'p[4]'])

# Test whether the difference: New York VS Westchester
cat("\n Test whether the difference: New York VS Westchester \n")
d12.sample <- p1.sample - p2.sample
quantile(d12.sample,
         probs=c(2.5, 97.5)/100) # Equal tail interval
HPDinterval(mcmc(data=d12.sample),
            alpha = 0.05) # HPD interval

# Test whether the difference: New York VS Long Island
cat("\n Test whether the difference: New York VS Long Island \n")
d13.sample <- p1.sample - p3.sample
quantile(d13.sample,
         probs=c(2.5, 97.5)/100) # Equal tail interval
HPDinterval(mcmc(data=d13.sample),
            alpha = 0.05) # HPD interval

# Test whether the difference: New York VS Rest of NYS
cat("\n Test whether the difference:  New York VS Rest of NYS \n")
d14.sample <- p1.sample - p4.sample
quantile(d14.sample,
         probs=c(2.5, 97.5)/100) # Equal tail interval
HPDinterval(mcmc(data=d14.sample),
            alpha = 0.05) # HPD interval

# Test whether the difference: Westchester VS Long Island
cat("\n Test whether the difference:  Westchester VS Long Island \n")
d23.sample <- p2.sample - p3.sample
quantile(d23.sample,
         probs=c(2.5, 97.5)/100) # Equal tail interval
HPDinterval(mcmc(data=d23.sample),
            alpha = 0.05) # HPD interval

# Test whether the difference: Westchester VS Rest of NYS
cat("\n Test whether the difference:  Westchester VS Rest of NYS \n")
d24.sample <- p2.sample - p4.sample
quantile(d24.sample,
         probs=c(2.5, 97.5)/100) # Equal tail interval
HPDinterval(mcmc(data=d24.sample),
            alpha = 0.05) # HPD interval

# Test whether the difference: Long Island VS Rest of NYS
cat("\n Test whether the difference:  Long Island VS Rest of NYS \n")
d34.sample <- p3.sample - p4.sample
quantile(d34.sample,
         probs=c(2.5, 97.5)/100) # Equal tail interval
HPDinterval(mcmc(data=d34.sample),
            alpha = 0.05) # HPD interval
```

Question 4

```
# Calculate the MODE (peak) for the Beta distribution
beta.mode<- function(alpha, beta) {
  return((alpha-1) / (alpha+beta-2))}

#sensitivity
beta.mode(205,29) #result:0.8793103
#specificity =
beta.mode(288,2) #result: 0.9965278


# Create plots of Beta distribution
# Define range
p = seq(0, 1, length=1000)

# Plot for sensitivity:
par(mfrow=c(1,1))
plot(p, dbeta(p, 205 , 29), type='l', ylab="Density", xlab="Sensitivity",
     main="Prior distribution for Sensitivity ~ Beta(205, 29)",
     xlim=c(0.80,1), ylim=c(-0.5,30))
points(0.879, 0.5, col="red", pch=19) #the mode or point estimate
abline(v=0.879, col="purple", lty =2) #the mode or point estimate
text(0.879, 3, "0.879", col="purple")

# Plot for specificity:
par(mfrow=c(1,1))
plot(p, dbeta(p, 288, 2), type='l', ylab="Density", xlab="Specificity",
     main="Prior distribution for Specificity ~ Beta(288, 2)",
     xlim=c(0.80,1), ylim=c(-0.5,120))
points(0.996, 0.5, col="red", pch=19) #the mode or point estimate
abline(v=0.996, col="purple", lty =2) #the mode or point estimate
text(0.996, 6, "0.996", col="purple")
```

Question 5

```
# Set 2 different initials for 2 chains including for se and sp
init1_q5 <- list(p = 0, se=0, sp=0)
init2_q5 <- list(p = 1, se=1, sp=1) # 'extreme' choice for prevalence
model.initials2 <- list(init1_q5, init2_q5)

# Model specification
ModelCumIncidence <- nimbleCode(
  {
    # likelihood specification : binomial
    for (i in 1:length(N)) {
      z[i] ~ dbinom(p[i], N[i])
    }

    # derived quantity : pi
    for (i in 1:length(N)) {
      pi[i] <- ( p[i] + sp - 1 ) / (se + sp -1)
    }

    # prior information : vague one for the prevalence
    for (i in 1:length(N)) {
      p[i] ~  dunif(0, 1)
    }
    se ~ dbeta(205 , 29)
    sp ~ dbeta(288 , 2)

  })

# Run 2 chains using 2 different sets of initial; 5000 iterations in the end
OutputMCMC_q5 <- nimbleMCMC(
  code=ModelCumIncidence,
  data=model.dataZ,
  constants=model.dataN,
  inits=model.initials2,
  monitors=c("p","pi"),
  nchains=2,
  nburnin=1000,
  niter=16000,
  thin = 3,
  setSeed=2023,
  summary=TRUE
  )

#Getting the chains
chain1_q5 <- OutputMCMC_q5$samples$chain1
chain2_q5 <- OutputMCMC_q5$samples$chain2

# History plots: pi1 (each chain)
traceplot(as.mcmc(chain1_q5[,'pi[1]']),
          col = "turquoise",
          main = 'Trace Plot of Cummulative Incidence for New York City',
          ylab = 'Value')
traceplot(as.mcmc(chain2_q5[,'pi[1]']),
          add = TRUE,
          col = "pink")

# History plots: pi2 (each chain)
traceplot(as.mcmc(chain1_q5[,'pi[2]']),
          col = "turquoise",
          main = 'Trace Plot of Cummulative Incidence for Westchester/Rockland Counties',
          ylab = 'Value')
traceplot(as.mcmc(chain2_q5[,'pi[2]']),
          add = TRUE,
          col = "pink")

# History plots: pi3 (each chain)
traceplot(as.mcmc(chain1_q5[,'p[3]']),
          col = "turquoise",
          main = 'Trace Plot of Cummulative Incidence for Long Island',
          ylab = 'Value')
traceplot(as.mcmc(chain2_q5[,'p[3]']),
          add = TRUE,
          col = "pink")

# History plots: pi4 (each chain)
traceplot(as.mcmc(chain1_q5[,'p[4]']),
          col = "turquoise",
          main = 'Trace Plot of Cummulative Incidence for Rest of NYS',
          ylab = 'Value')
traceplot(as.mcmc(chain2_q5[,'p[4]']),
          add = TRUE,
          col = "pink")

# Autocorrelation plots: pi1 (each chain)
autocorr.plot(as.mcmc(
  c(chain1_q5[,'pi[1]'], chain2_q5[,'pi[1]'])),
  main = 'Autocorrelation of Cummulative Incidence for New York City')

# Autocorrelation plots: pi2 (each chain)
autocorr.plot(as.mcmc(
  c(chain1_q5[,'pi[2]'], chain2_q5[,'pi[2]'])),
  main = 'Autocorrelation of Cummulative Incidence for Westchester/Rockland Counties')

# Autocorrelation plots: pi3 (each chain)
autocorr.plot(as.mcmc(
  c(chain1_q5[,'pi[3]'], chain2_q5[,'pi[3]'])),
  main = 'Autocorrelation of Prevalence for Long Island')

# Autocorrelation plots: pi4 (each chain)
autocorr.plot(as.mcmc(
  c(chain1_q5[,'pi[4]'], chain2_q5[,'pi[4]'])),
  main = 'Autocorrelation of Cummulative Incidence for Rest of NYS')


#Gelman and Rubin's convergence diagnostics updated model
chain.combined_q5 <- mcmc.list(
  as.mcmc(
    chain1_q5[,c("pi[1]","pi[2]","pi[3]","pi[4]")]),
  as.mcmc(
    chain2_q5[,c("pi[1]","pi[2]","pi[3]","pi[4]")])
  )
gelman.diag(chain.combined_q5)
gelman.plot(chain.combined_q5)

# Here comes the conclusion: region-based pi
(pi1.mean <- OutputMCMC_q5$summary$all.chains['pi[1]','Mean'])
(pi2.mean <- OutputMCMC_q5$summary$all.chains['pi[2]','Mean'])
(pi3.mean <- OutputMCMC_q5$summary$all.chains['pi[3]','Mean'])
(pi4.mean <- OutputMCMC_q5$summary$all.chains['pi[4]','Mean'])
```