PHP2530 Final Project - A Bayesian Analysis of Scott County Injection Network Data

Alyson Singleton

5/1/2020

Introduction

Significant progress has been made in reducing HIV incidence among people who inject drugs (PWID) in the United States (US), but these declines have stalled in the face of frequent rapid HIV transmission events occurring during an escalating drug overdose epidemic [Singh S, et al. Annals of Internal Medicine 2018, Des Jarlais DC, et al. AIDS 2016]. The largest ever of these events among PWID in a non-urban setting in the US occurred in Scott County, Indiana in 2015, when nearly 200 people in a community of 25,000 individuals were newly diagnosed with HIV infection.

Several factors have been associated with rapid HIV transmission in PWID, including a lack of awareness that HIV is a threat in the local setting, high frequency of needle/syringe sharing, inaccessibility of sterile injecting equipment, recent changes in drug usage patterns, and large, interconnected risk networks [Des Jarlais DC, et al. AIDS 2016]. Concerning the structure of those networks, certain micro-structures can further exacerbate HIV transmission and persistence of infection. While instances of rapid HIV transmission are more likely in settings with these characteristics [Des Jarlais DC, et al. AIDS 2016], they are not inevitable, with the potential to intervene on any of these factors, including on the network itself.

The ongoing opioid crisis makes this line of research particularly timely. The 2015 outbreak in Indiana, driven largely by high-frequency injection of the prescription opioid oxymorphone, has raised concerns about the potential for other outbreaks of HIV among PWID. An understanding of the impact of network structure on HIV transmission dynamics will be fundamental to mitigating and preventing future outbreaks.

This final project used Bayesian regularization techniques to investigate which structural network characteristics might be most predictive of the magnitude of epidemic behavior. I characterized the observed risk network from the HIV outbreak in Scott County, Indiana, as a case study. Using an agent-based network model, I generated a series of stochastic networks with comparable numbers of individuals and ties to the observed network to identify how HIV transmission is impacted by varying structural network characteristics. The simulated network data was created as a part of my work as a Graduate Research Assistant with the Marshall Research Group. This final project focuses on variable selection and prediction through a Bayesian framework, an analysis that was previously outside of the scope of the work, but that I have always been interested in.

I will now briefly outline the methods used to create the data set as they are imperative for understanding which structures were forced into the networks and which were emergent and stochastically created. This will serve as the descrription of the implied data collection mechanism. Because I am using simulation data, there is no explicit "missing data" in this problem. However, I want to recognize that the method of data generation, a description of which follows below, creates room for bias and there is likely a limit to the representativeness of the model. I think about this quite often, and while I don't explicitly address the representativeness of the simulation data to the real would in this analysis, it is something that I would like to do in the future.

Background on Observational Data

Our research utilizes previously published information on the observed contact tracing network from the 2015 outbreak in Scott County, Indiana [Campbell EM, et al. Journal of Infectious Diseases 2017, Peters PJ, et al. New England Journal of Medicine 2016]. The investigation of the outbreak was part of an emergency response conducted by the Indiana State Department of Health and the Centres for Disease Control and Prevention. The contact tracing investigation involved disease intervention specialists eliciting the names of past-year sexual and injection contacts of individuals newly diagnosed during the outbreak and offering HIV testing to these contacts.

The observed risk network comprised 420 individuals and 913 named ties, including the main component with 411 individuals. Among these ties, 79.2% were injection-related only, 7.9% were between sex-related only, and 12.9% were multiplex (i.e., were between sexual partners who shared injection equipment). Nearly half of the individuals included in this network (44.5%) were diagnosed with HIV infection during the outbreak investigation.

Background on Simulation Methods

Model Setting

We adapted a previously published version of the TITAN Model, an agent-based model, to explore rapid HIV transmission in Scott County, Indiana [Goedel WC, et al. Clinical Infectious Diseases 2019]. Agent-based modelling is a simulation method that represents micro-level interactions between individual entities called agents to understand the emergence of macro-level trends. Our model was parameterized, where possible, using published information on injection and sexual behaviour in Scott County and supplemented with estimates from the literature where necessary and then calibrated to the observed number of incident HIV infections.

There were very few rural injection networks to serve as appropriate comparators to the observed network. As we aimed to characterize and identify the potentially unique qualities of this network and their contribution to HIV transmission, we instead compared the observed network to agent-built simulated networks.

Each iteration of the simulation models HIV transmission in a network of comparable size (420 agents) in discrete time-steps each representing 1 calendar month for 5 years. Each iteration includes the initialization of the model population, the formation of the contact network, the introduction of HIV into the network through a randomly selected agent, and the monitoring of the progression of HIV throughout the population. A total of 1,000 iterations were simulated.

Population Formation

The model first initialized a virtual population. The size of the initial population varied across simulations from 402 to 441, as we aimed for the number of agents included in the simulated risk network (i.e., the number of agents with at least one tie) to be within ten per cent of the number of nodes in the observed network (n = 420). The attributes of agents in the base population were assigned through stochastic processes to achieve the desired gender distribution and gender-specific prevalence of IDU [Campbell EM, et al. Journal of Infectious Diseases 2017]. HIV prevalence in the network was set at 0% at initialization, reflecting an entirely susceptible population.

Network Formation

Following population formation, the model created ties between agents to build a risk network. The number of ties between agents represented a primary calibration target, where we aimed to generate networks with a total number of ties within ten per cent of the observed network (n = 913). This process ensured that the networks had a comparable density to the observed network. These ties were distributed within the agent population through the following processes.

All agents were assumed to be able to participate in sexual behaviour in the model and were assigned a target number of sexual partners from a negative binomial distribution with a mean of one partner [Goedel WC, et al. Clinical Infectious Diseases 2019]. Given the low number of male-male sexual dyads reported during contact tracing [Campbell EM, et al. Journal of Infectious Diseases 2017], only male-female sexual dyads were assumed to be possible in the simulated networks. Agents who inject drugs were given an additional target number of injection partners based on the observed injection-specific degree distribution [Campbell EM, et al. Journal of Infectious Diseases 2017]. This degree distribution was represented with a step function, with each step representing a range of possible target numbers of injection partners that an agent might be assigned. Each step was assigned a probability of occurrence, calculated from the observed injection-specific degree distribution. An observed 12.9% of the injection ties were also sexual ties and were constructed as such [Campbell EM, et al. Journal of Infectious Diseases 2017].

The network was constructed through an iterative process. The model iterated through the list of agents who had not yet reached their target numbers to match pairs of agents together. It created a list of eligible partners for each agent under the parameters governing partner compatibility (i.e., allowing for only male-female sexual ties to be formed, but ties of any combination of genders for injection ties), only including agents who were also not yet at their target number of partners. Once the model iterated over all agents who needed additional partners, agents were tied to one another based on this pairing algorithm. The pool of agents not yet at their target number was re-built, and the process was repeated until all individuals were paired or there were no more eligible partners for a given agent. This process introduces stochasticity into the structure of the set of simulated networks: the model provided agents target numbers based on the observed distributions while allowing their realized degree to differ.

Introduction of HIV Infection

After the formation of the contact network, a single agent engaging in IDU was chosen to seroconvert spontaneously, thus introducing HIV into the network. Hereafter, we refer to this agent as the initial infection.

HIV Transmission

Briefly, agents were assigned a target number of condomless vaginal intercourse acts per partner per month, drawn from a Poisson distribution with a mean of 13 acts per month [Crosby RA, et al. Annals of Epidemiology 2012]. Sexual acts that included the use of condoms were not explicitly simulated as they were assumed to carry a negligible risk of HIV transmission. Agents who inject drugs were also assigned a target number of injection acts per month from a Poisson distribution with a mean of 150 injection acts [Dasgupta S, et al. AIDS & Behavior 2019]. On average, 34% of these injection acts were estimated to include syringe sharing [Dasgupta S, et al. AIDS & Behavior 2019]. These acts were evaluated as the number of trials (n) in binomial distributions that model HIV transmission, where the probability of success (p) is the probability of transmission associated with particular behaviours.

Simulated Network Analysis

We measured possible differences using network measures that had been previously shown to either facilitate or limit rapid HIV transmission in many past published works. These measures included the number of components in the network (omitting isolated individuals), size of the largest component (referred to as main component), density of the main component, average betweenness centrality of the main component, betweenness centrality of the initial infection, average geodesic distance of the main component, geodesic distance of the initial infection, diameter of the main component, degree of the initial infection, centralization of the main component, the proportion of individuals located in a 2-core in the main component, and transitivity in the main component. A glossary of terms guiding the analysis of structural network characteristics can be found on the following page for your reference.

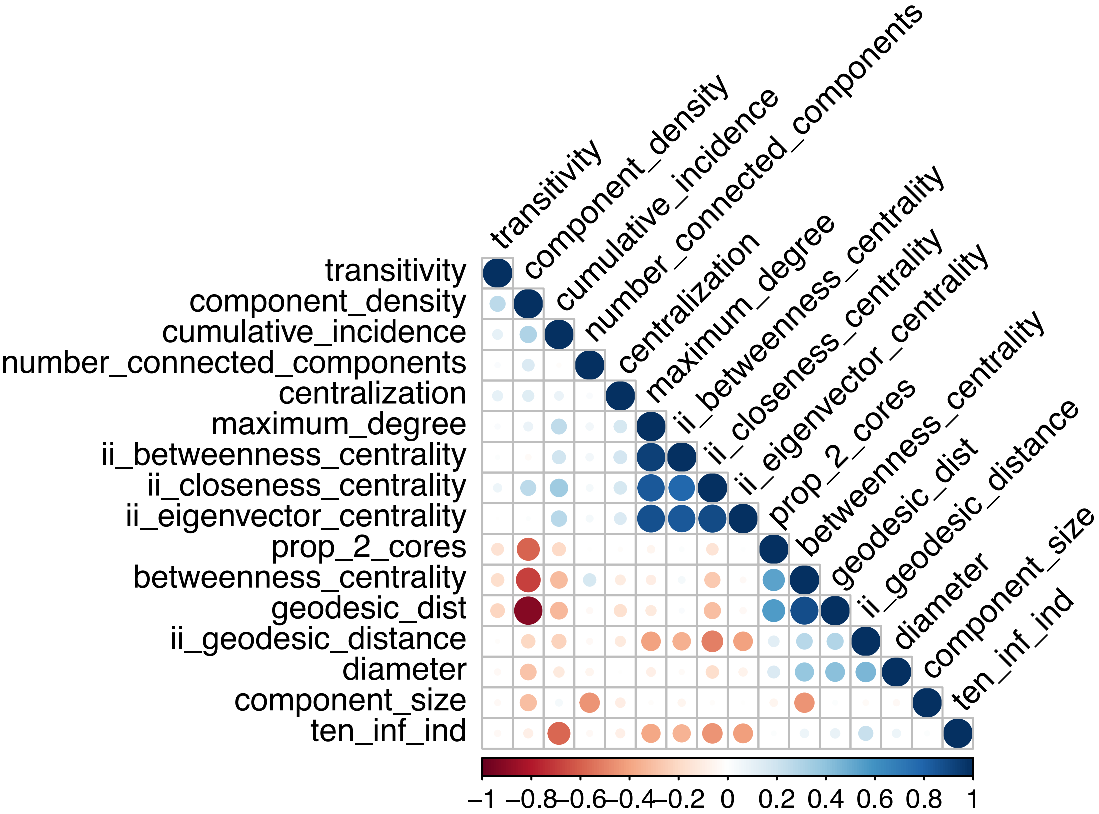
The primary comparison outcome measure across scenarios was the cumulative number of incident HIV infections over the simulation period. We also measured the number of months elapsed until ten incident infections, the number of HIV infections that incited the contact tracing investigation and the subsequent increase in testing activities to investigate how network structure may be related to epidemic velocity. Unfortunately I did not have enough time to investigate this second outcome. I would have very much likely to have created a multivariate model and included the relationship between these two outcomes in my analysis.

Table 1: Glossary of terms guiding the analysis of structural network characteristics.

Metric	Definition	Hypothesized Association with Epidemic Behavior
Number of Components	Number of components in the network, omitting isolated individuals.	The number of components helps represent the dispersion of individuals across a static network, as individuals in separate components cannot, by definition, infect one another [Bearman et al., 2004; Estrada, 2012].
Component Size	Number of individuals within a component.	The size of the largest component in a static network represents an upper-bound for potential disease diffusion in the population [Estrada, 2012].
Density	Number of ties in the network divided by the number of total possible ties in the network.	Network density helps determine outbreak magnitude and represents the "knittedness" of the component of interest [Seidman, 1983; Tarwater and Martin, 2001].
Betweenness centrality	Fraction of shortest paths between all other pairs of individuals that pass through a given index individual.	A component with high average betweenness centrality is dependent on a select number of individuals for disease diffusion, while one with low average betweenness centrality likely has many routes of transmission through many individuals [Christley et al., 2005; Freeman, 1978].
Geodesic Distance	Length of the shortest path between two connected individuals in a network.	Networks with lower average geodesic distance might have multiple paths, or "shortcuts," between individuals, compared to networks that form in long, chain-like structures [Bearman et al., 2004; Estrada, 2012; Wasserman et al., 1994].
Diameter	Maximum distance between a pair of individuals in a component.	A large diameter might indicate the network includes a long, chain-like structure, rather than a more condensed shape [Wasserman et al., 1994].
Degree	An individual's numbers of ties.	An individual's degree represents its potential influence on the network [Bell et al., 1999].
Centralization	Sum of the differences of the largest observed degree and each of the individuals' degrees, divided by the sum of the maximum of these differences.	A high centralization measure could indicate dependence on one or a few individuals in a hierarchical network, and it can either accelerate disease diffusion through these individuals acting as "broadcasters" [Valente, 2010] or reduce disease diffusion through a phenomenon called the "firewall effect" [Khan et al., 2013; Valente, 2010].
K-cores	A connected group in which all individuals are connected to at least k individuals in the group.	Networks with substantial amounts of their individuals located in k-cores represent structures where there are tight-knit sub-groups [Doreian and Woodard, 1994]. 2-core participation specifically has been shown to be a high-risk position within a network [Friedman et al., 1997].
Transitivity	The proportion of all triads that exhibit closure (i.e., a complete triangle).	Individuals with high transitivity are more at risk for acquiring the disease because they can be located in the more dense areas of the network and can be reached through multiple avenues. Diseases spread more readily through highly transitive networks [Shirley and Rushton, 2005].

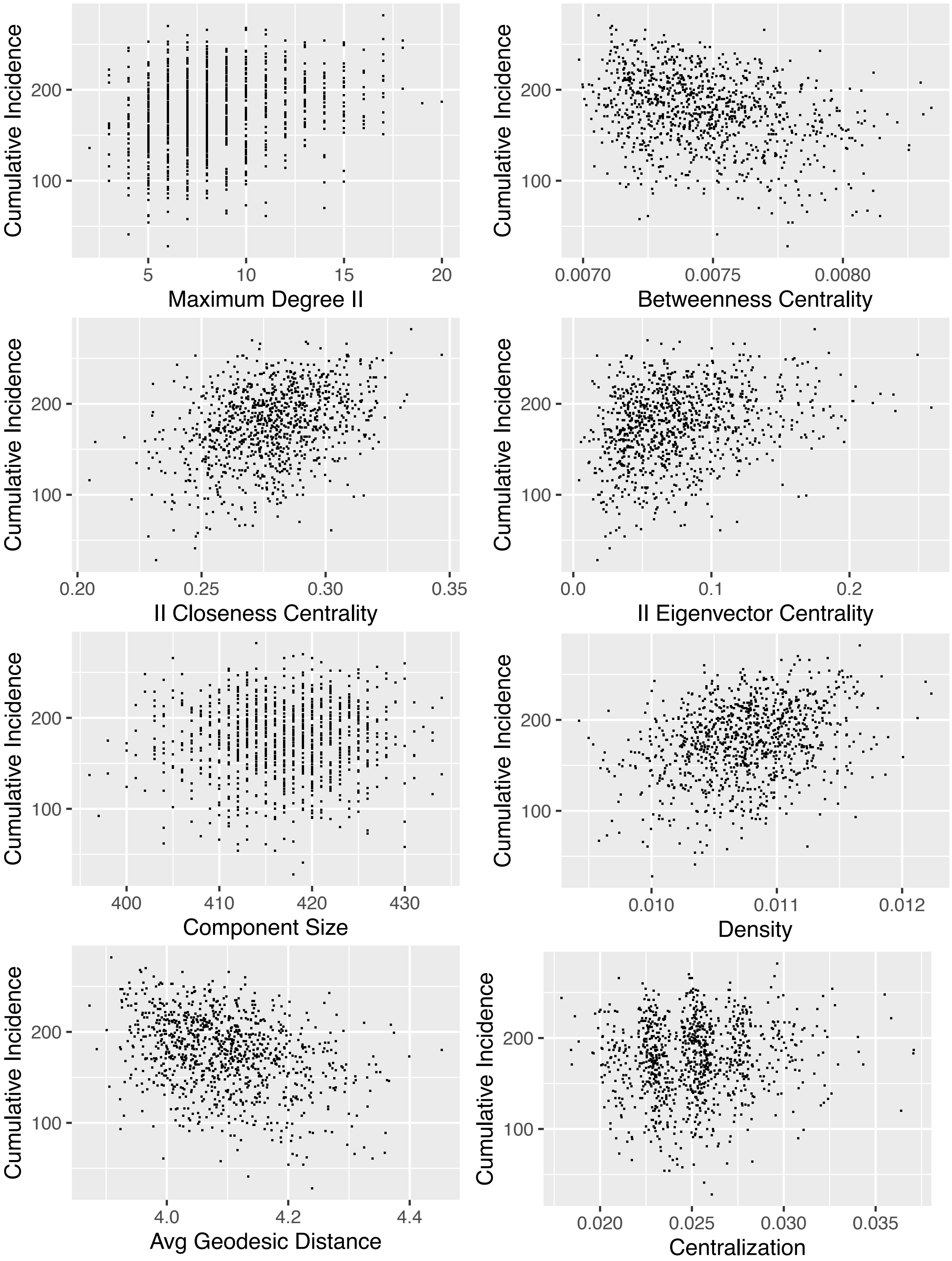
Basic Data Investigation

I began my investigation of the data by investigating the potential collinearity between the structural network measurements. Below you can see a correlation plot. I was unsprised to see groups of variables that were especially correlated. One group is a set of measurements on the placement of the randomly selected initial infection location (notated by "ii" for initial infection). There is a second group that is a set of structural characteristics that are mathematically similar, at least given our impletemented conditions on the network. This group is correlated because they investigate similar structures, although they each have valuable nuance and likely can differ more extemely in other types of networks, with different densities, for example.



This gave me cause to choose Bayesian regularization methods as a focus of my project. Instead of simply removing variables displaying collinearity in the model, I was interest in using a Bayesian framework to analytically choose which predictors are most important in predictions of network vulnerability. I also made sure to standardize by mean-centering and scaling as is common practice.

Next, I looked at the shape of the relationships between each of the network measurements and our outcome. Upon inspection, it appears that most variables have weak relationships with the outcome and are generally fairly linear, with exceptions. I decided to assume linearity for this analysis, but recognize that there are a few characteristics that would have benefitted from non-linear representation. I had hoped to have time to investigate this, but in the end I did not. Also, in this analysis I am particularly concerned in interpretability of the coefficients, given the direct application of the outcomes. A selection of the plots are displayed below that represent the range of shapes.


Cumulative Incidence
100
200
5
10
15
20
Maximum Degree II
Cumulative Incidence
100
200
0.0070
0.0075
0.0080
Betweenness Centrality
Cumulative Incidence
100
200
0.20
0.25
0.30
0.35
II Closeness Centrality
Cumulative Incidence
100
200
0.0
0.1
0.2
II Eigenvector Centrality
Cumulative Incidence
100
200
400
410
420
430
Component Size
Cumulative Incidence
100
200
0.010
0.011
0.012
Density
Cumulative Incidence
100
200
4.0
4.2
4.4
Avg Geodesic Distance
Cumulative Incidence
100
200
0.020
0.025
0.030
0.035
Centralization

Bayesian Model Framework

Lasso Overview

First, we being by reviewing the frequentist defintion of a Lasso regression–a method originally created for shriking the number of predictors in a wide data set. It is usually applied to a linear regresion, which we will employ in this analysis. Recall, where $\mathbf{y}$ is the $n \times 1$ outcomes, μ is the grand mean, $\mathbf{X}$ is the $n \times p$ standardized predictor data, $\boldsymbol{\beta} = (\beta_1, ..., \beta_p)^T$ is the vector of regression coefficients to be determined, and $\boldsymbol{\epsilon}$ is the $n \times 1$ vector of independent and identically distributed normal errors with mean 0 and unknown variance σ^2.

$$\mathbf{y} = \mu\mathbf{1}_n + \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$$

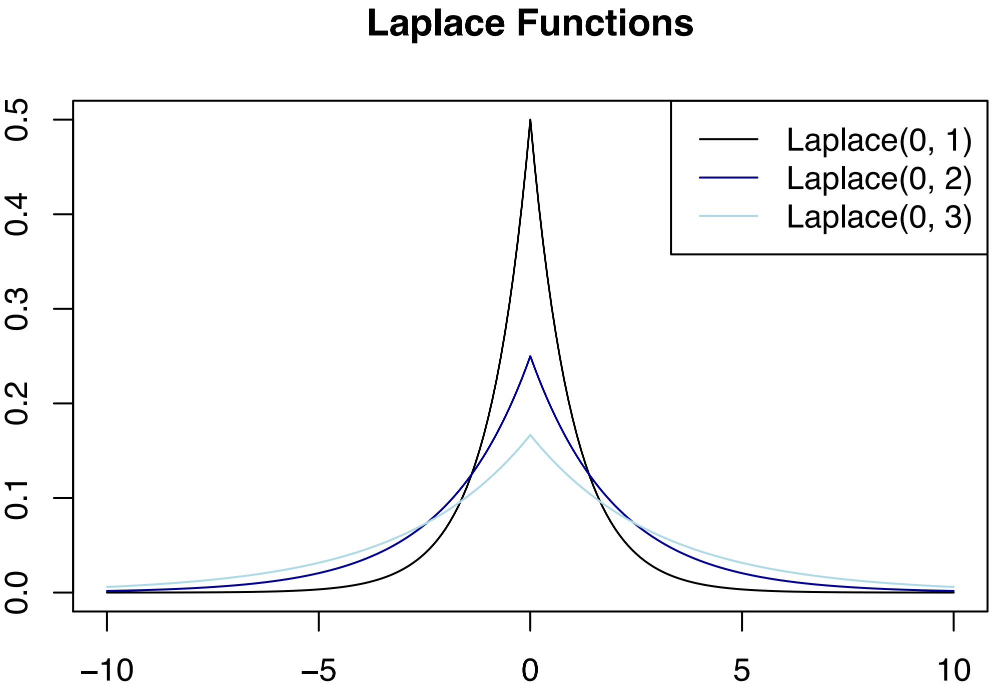
We recall that the Lasso estimate $\hat{\boldsymbol{\beta}}$ minimizes the sum of the square of the residuals while being subject to a bound on its L_1 norm. In other words, the general form of the lasso regression optimization problem is minimizing the following, where $\tilde{\mathbf{y}}$ is the mean-centered outcome vector and $\lambda \geq 0$ related to the aforementioned bound:

$$\min_{\beta}(\tilde{\mathbf{y}} - \mathbf{X}\boldsymbol{\beta})^T(\tilde{\mathbf{y}} - \mathbf{X}\boldsymbol{\beta}) + \lambda \sum_{i=1}^{p} |\beta_j|$$

From here it is not a large leap into the bayesian framework. This expression provides the intuition that we can also interpret the Lasso as a a Bayesian posterior mode estimate when the parameters β_i have independent and identical double exponential (Laplace) priors [Tibshirani, 1996; Hastie et al., 2001]. Indeed, we can write our priors on the β_i in the following manner:

$$p(\boldsymbol{\beta}) = \prod_{j=1}^{p} \frac{\lambda}{2} \exp(-\lambda|\beta_j|)$$

This is quite intuitive when one looks at plots of Laplace functions. See below for three functions, each centered at zero with varying λ values ($\lambda \in \{1, 2, 3\}$). By setting the centers of our priors for our coefficients to zero with a small scaling variable (λ), we can reflect high confidence that most of the coefficients are unimportant. This allows us to identify and select out only the coefficients that strongly supported by the data. We would expect to see the posterior distributions of the strongest predictors to pull away from a center of zero, while those weakest will be encouraged to remain in this shape.



Lastly, if we let σ^2 take on an independent prior when positive, the conditional posterior distribution can indeed be expressed as:

$$p(\boldsymbol{\beta}, \sigma^2|\tilde{\mathbf{y}}) \propto p(\sigma^2)(\sigma^2)^{-(n-1)/2} \exp\left\{-\frac{1}{2\sigma^2}(\tilde{\mathbf{y}} - \mathbf{X}\boldsymbol{\beta})^T(\tilde{\mathbf{y}} - \mathbf{X}\boldsymbol{\beta}) + \lambda \sum_{i=1}^{p} |\beta_j|\right\}$$

For any fixed value of $\sigma^2 > 0$, we maximize $\boldsymbol{\beta}$ as our Lasso esimate, and therefore the posterior mode estimate will be a Lasso estimate as well. Critically, by using this method, we are deciding to summarize the posterior distribution (or, equivalently, the penalized likelihood function) by its mode. Lastly, the estimates will depend on our prior for σ^2 and our choice of hyerparameter λ.

My Models

Now, I will outline the specifics of the models I investigated for this project. Written out clearly, the hierarchical model structure suggested by T. Park and G. Casella [Journal of the American Statistical Association, 2008] is as follows:

$$p(\mathbf{y}|\mu, \mathbf{X}, \boldsymbol{\beta}, \sigma^2) \sim N(\mu \mathbf{1}_n + \mathbf{X}\boldsymbol{\beta}, \sigma^2\mathbf{I})$$
$$p(\boldsymbol{\beta}|\tau_1^2, ..., \tau_p^2, \sigma^2) \sim N(\mathbf{0}_p, \sigma^2\mathbf{D}), \mathbf{D}_\tau = diag(\tau_1^2, ..., \tau_p^2)$$
$$p(\tau_1^2, ..., \tau_p^2) \sim \prod_{j=1}^{p} \frac{\lambda^2}{2} \exp(-\lambda^2\tau_j^2/2)$$
$$p(\sigma^2) \sim \pi(\sigma^2)$$

We are assuming that σ^2 and the $p(\tau_1^2, ..., \tau_p^2)$ are independent. Now, all that we have left to decide is our prior for σ^2 and our choice of hyerparameter λ. We assume an independent, flat prior on μ as recommended.

Prior on Sigma

There are multiple plausible choices of a prior distribution for σ^2 outside of one that is conditionally independent with β. One of these options is the scale invariant prior of $p(\sigma)^2 = \frac{1}{\sigma^2}$, which would result in a conditional prior for β as follows:

$$p(\boldsymbol{\beta}|\sigma^2) = \prod_{j=1}^{p} \frac{\lambda}{2\sqrt{\sigma^2}} \exp(-\lambda|\beta_j|/\sqrt{\sigma^2})$$

Additionally, T. Park and G. Casella suggest using an inverse gamma prior distribution because it is a conjugate prior. Where $\{\sigma^2, a, \gamma\} > 0$,

$$p(\sigma^2) = \frac{\gamma^a}{\Gamma(a)}(\sigma^2)^{-a-1} \exp(\frac{-\gamma}{\sigma^2})$$

As is suggested by the literature and what seems to be most common, I will indeed use the inverse gamma prior. The resulting posterior is proper and takes the following form:

$$p(\boldsymbol{\beta}, \sigma^2|\tilde{\mathbf{y}}) \propto (\sigma^2)^{-(n+p-1)/(2-a-1)} \exp\left\{-\frac{1}{\sigma^2}((\tilde{\mathbf{y}} - \mathbf{X}\boldsymbol{\beta})^T(\tilde{\mathbf{y}} - \mathbf{X}\boldsymbol{\beta})/2 + \gamma) + \frac{\lambda}{\sqrt{\sigma^2}} \sum_{i=1}^{p} |\beta_j|\right\}$$

This is what I assume for the rest of the paper. I would have liked to have explored this derivation and choice further in my analysis had there been more time available.

Hyperparameter Lambda

The literature cites multiple ways to specify a choice of λ. The parameter can chosen using a priori knowledge to select an appropriate hyperprior, using typical cross-validation techniques, or potentially through empirical Bayes via a marginal maximum likelihood estimation. The only option of these available to frequentists is cross-validation–an advantage of using a Bayesian Lasso is that it provides "uniquely Bayesian alternatives" for choices of λ [Park and Casella, Journal of the American Statistical Association, 2008].

With more time I would have loved to write up a Monte Carlo EM algorithm to maximize the likelihood function of the λ. However, due to time constraints, I decided to use a hyperprior for λ instead. I think this is a nice compromise as it is decidely a Bayesian method and still helps to account for the uncertainty in the selection of λ. I will compare the outputs of a model using the chosen hyperprior with one using crossed-validation (the one I presented on). One example of a potential diffuse hyperprior for λ is a class of gamma priors:

$$p(\lambda^2 | r, \delta) \propto (\lambda^2)^{r-1} \exp(-\delta\lambda^2)$$

I will make use of this hyperprior in my analysis due to its conjugacy and the fact that it leads to a proper posterior (the literature discusses how the improper scale-invariant prior $1/\lambda^2$ would lead to an improper posterior) [Park and Casella, Journal of the American Statistical Association, 2008].

Posterior

Lastly, I will write out the full posterior density including our choices of priors:

$$p(\mu, \boldsymbol{\beta}, \sigma^2 | \mathbf{y}) = p(\mathbf{y} | \mu, \boldsymbol{\beta}, \sigma^2)\pi(\sigma^2)\pi(\mu) \prod_{j=1}^{p} \pi(\beta_j | \tau_j^2, \sigma^2)\pi(\tau_j^2) =$$

$$\frac{1}{(2\pi\sigma^2)^{n/2}} \exp\left(-\frac{1}{2\sigma^2} (\mathbf{y} - \mu\mathbf{1}_n - \mathbf{X}\boldsymbol{\beta})^T (\mathbf{y} - \mu\mathbf{1}_n - \mathbf{X}\boldsymbol{\beta}) \right)$$

$$\times \frac{\gamma^a}{\Gamma(a)} (\sigma^2)^{-a-1} \exp(\frac{-\gamma}{\sigma^2}) \prod_{j=1}^{p} \frac{\lambda}{\sqrt{2\pi\sigma^2\tau_j^2}} \exp\left(\frac{1}{2\pi\sigma^2\tau_j^2} \beta_j \right) \frac{\lambda^2}{2} \exp(-\lambda^2\tau_j^2/2)$$

Computational Methods – Evaluation

MY first model of interest is the Frequentist Lasso model. I used the “glmnet” function to calculate the coefficients resulting from a the L1 Norm penalty term defined at the start of the Methods section. I had used this model before, so I was interested to see if there were any differences between the selection made from the typical frequentist method and those under bayesian frameworks, although I did not necessarily expect there to be substantial difference in which variable were selected. Please see the selected β coefficients and their estimates below (Table 2).

Table 2: Frequentist Lasso, Cross Validated Lambda

Intercept	0.000
Maximum Degree	0.000
Betweenness Centrality	0.000
II Betweenness Centrality	0.000
II Closeness Centrality	-0.057
II Eigenvector Centrality	0.209
Component Size	0.114
Number Connected Components	0.122
Component Density	-0.005
Geodesic Distance	0.276
II Geodesic Distance	0.000
Centralization	-0.026
Prop2Cores	0.001
Transitivity	0.006
Diameter	0.015

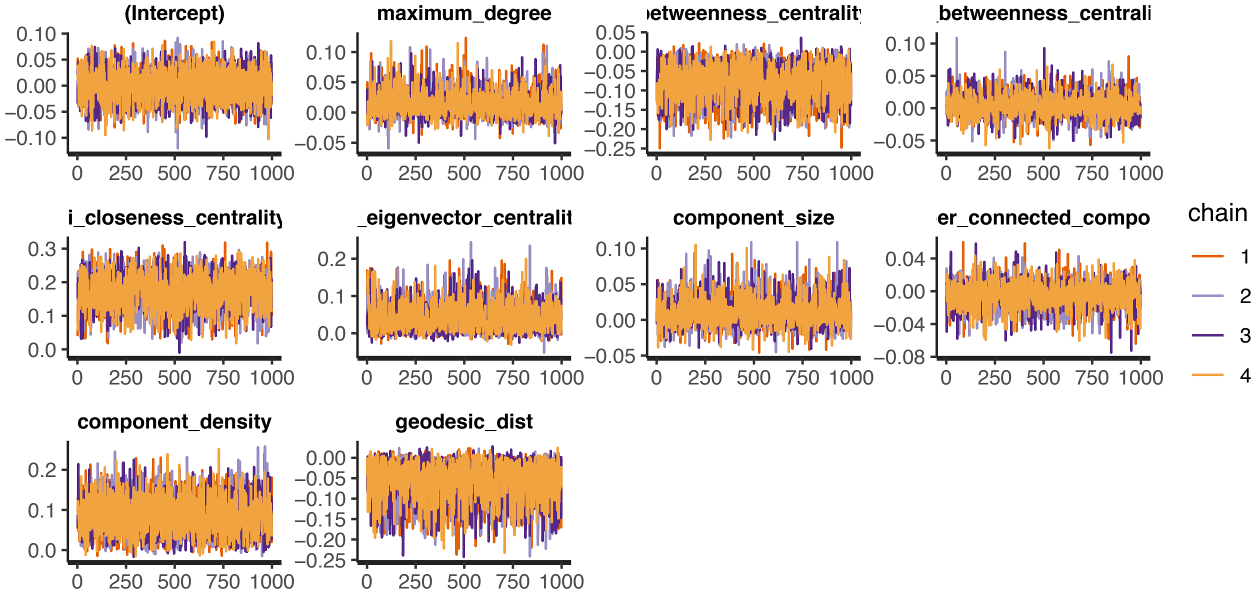
This model elected to keep II Eigenvector Centrality, Component Size, Number of Connected Components, and Geodesic Distance. There were others that were pulled slightly from zero, but none with magnitude larger than +/- 0.1.

Next, I constructed a Bayesian Lasso using STAN. A package called “rstanarm” allowed me to quickly employ Hamiltonian Monte Carlo (HMC) Sampling. I used the default number of chains (4), burn in iterations (1,000), and the number of interations for progression (1,000). For this model I also employed cross validation (cv.EBglmnet from the “EBglmnet” package) to determine the value of λ. The model chooses the lambda that minimizes the square error (for the results displayed below $lambda \approx 0.012$). The β coefficients of the selected variables are displayed below (Table 3).

Table 3: Bayesian Lasso, Cross Validated Lambda

	10%	50%	90%	Std Dev	Rhat
Intercept	-0.04	0.00	0.04	0.03	1
Maximum Degree	-0.01	0.01	0.04	0.02	1
Betweenness Centrality	-0.14	-0.08	-0.02	0.05	1
II Betweenness Centrality	-0.01	0.00	0.02	0.02	1
II Closeness Centrality	0.10	0.17	0.23	0.05	1
II Eigenvector Centrality	0.00	0.03	0.09	0.04	1
Component Size	-0.01	0.01	0.03	0.02	1
Number Connected Components	-0.02	0.00	0.01	0.01	1
Component Density	0.02	0.08	0.15	0.05	1
Geodesic Distance	-0.12	-0.05	0.00	0.05	1
II Geodesic Distance	-0.04	-0.01	0.01	0.02	1
Centralization	-0.01	0.00	0.02	0.01	1
Prop2Cores	-0.03	-0.01	0.01	0.02	1
Transitivity	-0.01	0.01	0.03	0.02	1
Diameter	-0.01	0.00	0.02	0.01	1

This model appears to be converging nicely, all four chains are similar across each predictor, no matter if the predictor’s posterior is pulled away from zero or remains null. The tables above also display the $\hat{R}$ values, which all reach 1. This model selected Betweenness Centrality, II Closeness Centrality, Component Density, and Geodesic Distance as the strongest predictors. This model was much more cautious than the first.



My last model will employ a gamma distribution as a diffuse hyperprior for λ. Although the package above allows for placing a prior on λ, the only option provides is a chi-squared distribution. As I wanted to use a gamma distribution, I will take this as an opportunity to show the tools I have developed in this course and code an explicit Gibbs sampler using the derived conditional probabilities.

All of the code, including this algorithm, is attached at the end of this paper for your reference (Code Appendix). I based my calculations for the heirarchical model functions on their descriptions in the Park and Casella paper and the following citation [https://cs.gmu.edu/~pwang7/gibbsBLasso.html]. Implementing the matrix algebra was a stretch for me on this timeline, so I relied heavily on past implementations of these probability distributions. However, I did personally code the Gibbs sampling algortithm. Although I do not deny their importance, I am more interested in the sampling algorithms than the derivations and probability computations. For this reason I dedication my time towards the sampling rather than the probability functions. With more time I would have loved to have pushed myself to create the heirarchical model functions completely on my own.

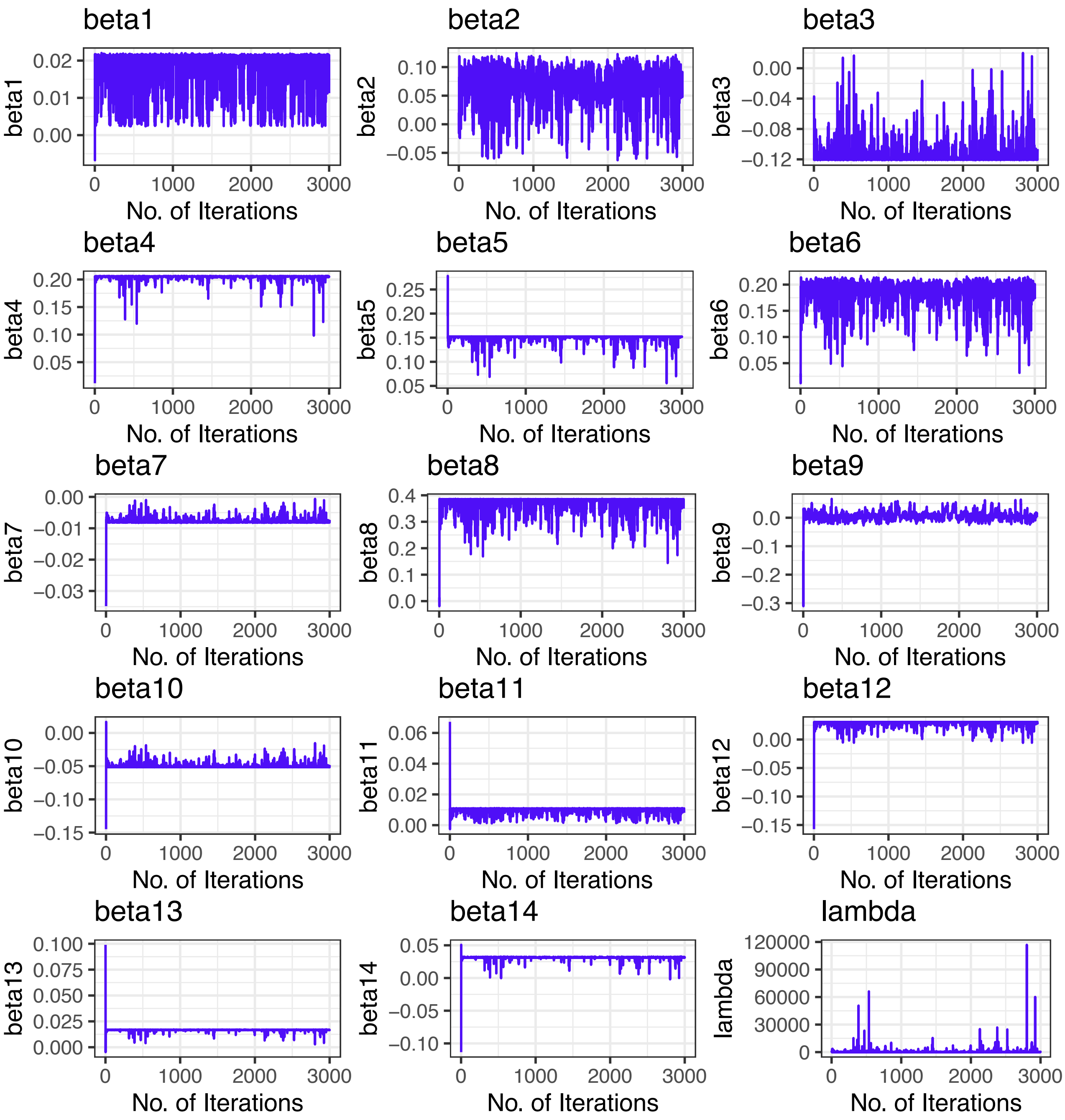
You can see the output (Table 4) and trace plots below. I used 3,000 iterations (including burn in) and constructed 4 chains using four different starting spots.

Table 4: Estimates from My Gibbs Sampler

	2.5%	25%	50%	75%	97.5%
Maximum Degree	0.003	0.017	0.021	0.021	0.022
Betweenness Centrality	-0.018	0.058	0.078	0.090	0.111
II Betweenness Centrality	-0.121	-0.120	-0.120	-0.115	-0.071
II Closeness Centrality	0.196	0.205	0.206	0.206	0.206
II Eigenvector Centrality	0.132	0.152	0.152	0.152	0.153
Component Size	0.118	0.182	0.193	0.201	0.210
Number Connected Components	-0.008	-0.008	-0.008	-0.008	-0.005
Component Density	0.276	0.372	0.384	0.385	0.386
Geodesic Distance	-0.015	0.001	0.005	0.012	0.044
II Geodesic Distance	-0.052	-0.051	-0.051	-0.051	-0.039
Centralization	0.003	0.010	0.011	0.011	0.011
Prop2Cores	0.015	0.029	0.031	0.031	0.031
Transitivity	0.015	0.017	0.017	0.017	0.017
Diameter	0.028	0.031	0.031	0.032	0.032
Sigma	0.002	0.002	0.002	0.002	0.002
Lambda	0.350	1.522	4.308	12.236	56.476

This final model selected II Betweenness Centrality, II Closeness Centrality, II Eigenvector Centrality, Component Size, and Component Density, the strongest predictors. This model was closer to the magnitude of the estimates in first model. Additionally, please note the variety of the lambda values tested out during the sampling procedure. The model was able to select much larger lambda values than was chosen through cross-validation.

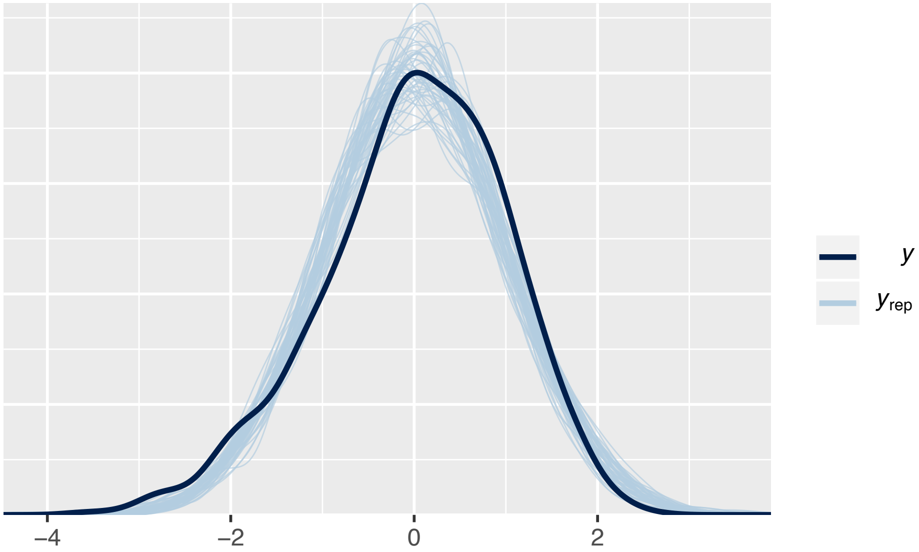
Unfornately, as you can see on the subsequent page, it does not appear that my attemt to use multiple chains is working correctly. They appear to be ovelapping with each other completely, which leads me to believe something is going wrong either with how I was trying to initiate different starting spots or with how I'm progressing through the probability space. I unfortunatley ran out of time to fully dig into this and figure out what was wrong. Aside from the chains, the plots are quite interesting in comparison with the trace plots from the packages sampler. These estimates seem to be more encouraged to remain in one location and seem to take more extreme values than the second model. I wonder if the variable, extreme lambda values played a role in this model's output behavior. Although there seem to be some concerns here, I am encouraged by the fact that some predictors are still pulled to zero as we would expect them to be.


beta1
beta1
0.02
0.01
0.00
0
1000
2000
3000
No. of Iterations
beta2
beta2
0.10
0.05
0.00
−0.05
0
1000
2000
3000
No. of Iterations
beta3
beta3
0.00
−0.04
−0.08
−0.12
0
1000
2000
3000
No. of Iterations
beta4
beta4
0.20
0.15
0.10
0.05
0
1000
2000
3000
No. of Iterations
beta5
beta5
0.25
0.20
0.15
0.10
0.05
0
1000
2000
3000
No. of Iterations
beta6
beta6
0.20
0.15
0.10
0.05
0
1000
2000
3000
No. of Iterations
beta7
beta7
0.00
−0.01
−0.02
−0.03
0
1000
2000
3000
No. of Iterations
beta8
beta8
0.4
0.3
0.2
0.1
0.0
0
1000
2000
3000
No. of Iterations
beta9
beta9
0.0
−0.1
−0.2
−0.3
0
1000
2000
3000
No. of Iterations
beta10
beta10
0.00
−0.05
−0.10
−0.15
0
1000
2000
3000
No. of Iterations
beta11
beta11
0.06
0.04
0.02
0.00
0
1000
2000
3000
No. of Iterations
beta12
beta12
0.00
−0.05
−0.10
−0.15
0
1000
2000
3000
No. of Iterations
beta13
beta13
0.100
0.075
0.050
0.025
0.000
0
1000
2000
3000
No. of Iterations
beta14
beta14
0.05
0.00
−0.05
−0.10
0
1000
2000
3000
No. of Iterations
lambda
lambda
120000
90000
60000
30000
0
0
1000
2000
3000
No. of Iterations

Posterior Predictive Checks

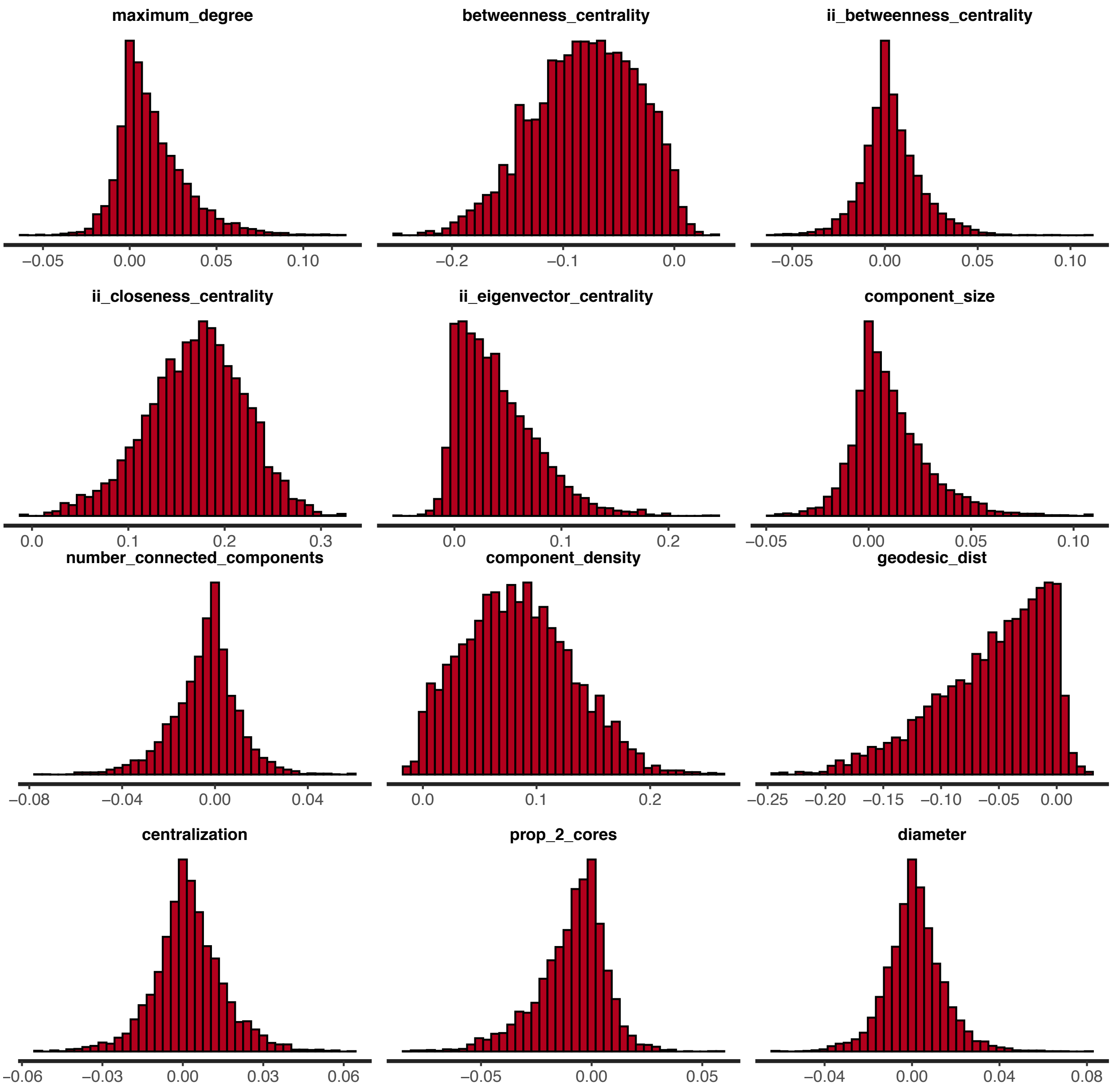
I will complete my posterior predictive checks primarily on the cross-validated Bayesian model. Had I had time, I would have liked to have tested out the model built from my Gibbs sampling work, but given my concerns with its output and my limited time, I will have to leave that for potential future work.

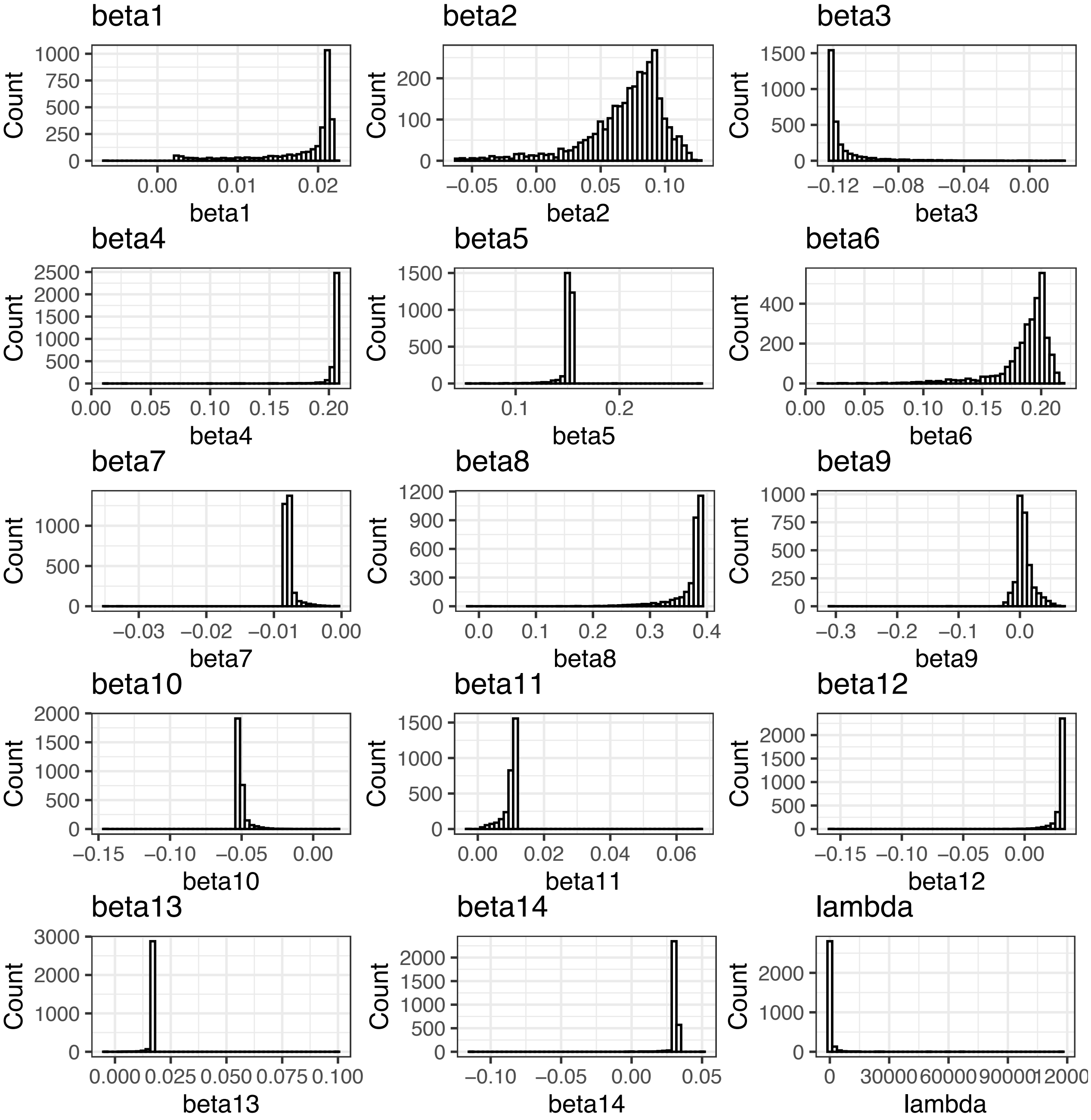
Our first primary predictive check is of the distribution of the outcome. The observed distribution is diplayed as the dark blue line, and the distribution of the predicted outcomes from the repeated simulations are in the lighter blue behind it. You can see that while it is capturing the height and spread of the distribution well, there are shapes to the distribution that it seems to be missing. Two examples are (1) the second "peak" that we see just below 1 and (2) the y_{rep} lines consistently overestimate the observed while increasing from -2 to -0.5, and consistently underestimate the observed while decreasing from 0.75 to 2.



On the next page we display the posterior distribution of our predictors. I find these plots very helpful to intuite what my model has done to create its predictions and selections. Some of the posterior distributions remain very similar in shape to the Laplace prior assigned to it. For example, the Maximum Degree and Geodesic Distance plots are very similar in shape to the Laplace functions I displayed earlier in the paper. On the other hand, this is also an effective way to see which predictors were supported enough by the data to vary away from their prior. Examples of these include Betweenness Centrality, Initial Infection Closeness Centrality, and Component Density.

For reference, I have also included the plots of the posterior distributions of the predictors from the Gibbs algorithm. They look as we would expect from their trace plots. Please note that their outliers are skewing some of the visualizations (I apologize I did not have time to remove them).


maximum_degree
−0.05
0.00
0.05
0.10
betweenness_centrality
−0.2
−0.1
0.0
ii_betweenness_centrality
−0.05
0.00
0.05
0.10
ii_closeness_centrality
0.0
0.1
0.2
0.3
ii_eigenvector_centrality
0.0
0.1
0.2
component_size
−0.05
0.00
0.05
0.10
number_connected_components
−0.08
−0.04
0.00
0.04
component_density
0.0
0.1
0.2
geodesic_dist
−0.25
−0.20
−0.15
−0.10
−0.05
0.00
centralization
−0.06
−0.03
0.00
0.03
0.06
prop_2_cores
−0.05
0.00
0.05
diameter
−0.04
0.00
0.04
0.08


beta1
Count
1000
750
500
250
0
0.00
0.01
0.02
beta1
beta2
Count
200
100
0
–0.05
0.00
0.05
0.10
beta2
beta3
Count
1500
1000
500
0
–0.12
–0.08
–0.04
0.00
beta3
beta4
Count
2500
2000
1500
1000
500
0
0.00
0.05
0.10
0.15
0.20
beta4
beta5
Count
1500
1000
500
0
0.1
0.2
beta5
beta6
Count
400
200
0
0.00
0.05
0.10
0.15
0.20
beta6
beta7
Count
1000
500
0
–0.03
–0.02
–0.01
0.00
beta7
beta8
Count
1200
900
600
300
0
0.0
0.1
0.2
0.3
0.4
beta8
beta9
Count
1000
750
500
250
0
–0.3
–0.2
–0.1
0.0
beta9
beta10
Count
2000
1500
1000
500
0
–0.15
–0.10
–0.05
0.00
beta10
beta11
Count
1500
1000
500
0
0.00
0.02
0.04
0.06
beta11
beta12
Count
2000
1500
1000
500
0
–0.15
–0.10
–0.05
0.00
beta12
beta13
Count
3000
2000
1000
0
0.000
0.025
0.050
0.075
0.100
beta13
beta14
Count
2000
1500
1000
500
0
–0.10
–0.05
0.00
0.05
beta14
lambda
Count
2000
1000
0
0
30000
60000
90000
12000
lambda

Conclusion

I can conclude that predictors involving the position of the initial infection (II Betweenness Centrality, II Closeness Centrality, II Eigenvector Centrality), those that represent the size of the vulnerable population (Component Size), and those that display the degree of knittedness in the network (Component Density, Avg Geodesic Distance) have the most influence on potential epidemic behavior. All trends observed are quite intuitive, although the models' specific choices of measures offer new information. If taken into consideration, these results could be used to help optimize network based interventions among PWID.

There is much that I could do to further extend this project. I made note of the limitations of this work throughout the paper, and hope that admist them I have been able to demonstrate my familiarity with Bayesian modes of thinking. Given more time, I would have wanted to further develop my personal algorithm, as it allowed for the most flexibility and was the easiest to personalize. Using the packages was much much easier, but with the chance that you might be assuming or defining something you did not mean to (something I am even a bit concerned about in this analysis). Additionally, I would have like to more rigorously choose my priors and/or investigate how different priors might have affected my final estimates. Thirdly, I very much would have wanted to include my second outcome in this analysis and potentially attempted a multivariate model.

Lastly, I believe that in conditioning the network to have comparable density, size and degree distribution to the observed network, I could be too severely limiting the amount of variability in the set of networks for this type of analysis. I conditioned the networks in this way so that they would be representative to Scott County. As described at the start of this document, there is still a substantial amount of noise and randomness introduced into the set of networks. However, I believe that the structural conditions restrict the set of networks from providing the richness we might have seen from 1,000 observed networks. I would be interested to see what would happen if I released some of the restrictions on the networks. I would guess that some of the restrictions would be strong predictors themselves (a reason we chose to define them in the first place). To conclude, these models might be fitting the data as best they can, and what they are missing is simply random noise. It might be most prudent to build another dataset first, and then begin investigating other models types (such as a gaussian mixture model).

To close, I now much prefer the Bayesian Lasso to its Frequentist counterpart. The intuition that underlines this model was a pleasure to delve into over the course of this project. I appreciate its clearly defined and algorithmic choices, and can see why it has become such a popular tool in the behavioral sciences.

I greatly enjoyed this project and this class. Thank you very much for your instruction and guidance this semester. I will look forward to any and all further thoughts you have on this work.

Code Appendix

```
############################################
# SetUp
knitr::opts_chunk$set(echo = F, eval=F, results='asis', warning=F,
                      message=F, cache=T, fig.align="center",
                      set.seed(100), fig.height=4.5, fig.width=6)
# load libraries
pacman::p_load(reshape2, naniar, actuar, dplyr, tidyr, kableExtra, knitr, LearnBayes,
               boot, lattice, gtools, ggplot2, grDevices, GGally, MASS, leaps,
               LaplacesDemon, corrplot, rstan, rstanarm, bayesplot, glmnet, neuralnet,
               devtools, ggbiplot, caret, standardize, EBglmnet, car, mnormt, VGAM)
############################################
# Glossary of Network Measures
glossary.df <- t(data.frame(c("Number of Components", "Number of components in the network, omitting is
                            c("Component Size", "Number of individuals within a component.", "The size
                            c("Density", "Number of ties in the network divided by the number of total
                            c("Betweenness centrality", "Fraction of shortest paths between all other p
                            c("Geodesic Distance", "Length of the shortest path between two connected i
                            c("Diameter", "Maximum distance between a pair of individuals in a componen
                            c("Degree", "An individual's numbers of ties.", "An individual's degree rep
                            c("Centralization", "Sum of the differences of the largest observed degree
                            c("K-cores", "A connected group in which all individuals are connected to a
                            c("Transitivity", "The proportion of all triads that exhibit closure (i.e.,
colnames(glossary.df) <- c("Metric", "Definition", "Hypothesized Association with Epidemic Behavior")
rownames(glossary.df) <- c(1:10)

kable(glossary.df, "latex",
      caption = "Glossary of terms guiding the analysis of structural network characteristics.",
      booktabs = T) %>%
  kable_styling(latex_options = c("striped", "hold_position"), font_size = 8) %>%
  column_spec(2:3, width = "6cm")
############################################
# Load Data
sc.set.original <- read.csv("~/Desktop/research/SC_inj_net_dynamics/scripts/A3_conditions_ten_percent.c

#specify only structural exposure vars and cuminc outcome
sc.set.original[,1:6] <- NULL
sc.set <- sc.set.original[c(1:19,30,43)]
sc.set.cum.inc <- sc.set.original[c(1:19,30)]
sc.set.ten.inc.inf <-  sc.set.original[c(1:19,43)]
############################################
# Correlation Investigation
# correlation plot of all
# corrplot(cor(sc.set), method = "circle", type = "lower",
  #tl.col = "black", tl.srt = 45, order = "hclust")
#remove some with high correlation that are uninteresting
  #(avg degree, totalnodecount, totaledgecount, CCnodescount, CCedgescount)
# make a seperate set with a few correlated variabls removed
sc.set.limit <- sc.set[-c(1,10:13)]
corrplot(cor(sc.set.limit), method = "circle", type = "lower",
         tl.col = "black", tl.srt = 45, order = "hclust") #FPC
sc.set.cum.inc <- sc.set.cum.inc[-c(1,10:13)]
sc.set.ten.inc.inf <-  sc.set.ten.inc.inf[-c(1,10:13)]
#corrplot.mixed(cor(sc.set.limit))

#remove even more for simplicity for this project
#sc.set.limit$geodesic_dist <- NULL
#sc.set.limit$maximum_degree <- NULL
#sc.set.limit$ii_closeness_centrality <- NULL
#sc.set.limit$ii_eigenvector_centrality <- NULL
#sc.set.limit$betweenness_centrality <- NULL
#sc.set.limit$prop_2_cores <- NULL
############################################
# Correlation bw Outcomes
#relationship between the outcomes? yes!
ggplot(data=sc.set, aes(x=ten_inf_ind, y=cumulative_incidence)) +
  geom_point() +
  labs(y="Number of Cumulative Infections",
       x="Time Until 10 Incident Infections") +
       #title = "Relationship between Outcomes") +
  theme(legend.position = "none",
        legend.title = element_text(hjust = 0.5, face = "bold", size = 14),
        legend.text = element_text(size = 10),
        legend.direction = "vertical",
        legend.box.just = "center",
        legend.spacing.x = unit(0.5, 'cm'),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        panel.background = element_blank(),
        axis.ticks = element_blank(),
        axis.line = element_line(color = "black"),
        axis.title = element_text(size = 16, colour = "black", face = "bold"),
        axis.text = element_text(size = 16, colour = "black"),
        plot.title = element_text(size = 18, colour = "black", face = "bold",
                                  hjust=0.5, vjust = 0.8))
############################################
# Pairs Plots Wrt CumInc

#ggpairs(sc.set.limit,
#        diag=list(continuous="density"),
        #columns=c(sc.set.limit$cumulative_incidence),
#        axisLabels="show",
#        mapping = ggplot2::aes(shape = ".", alpha = 0.1),
#        progress=TRUE)
# not linear wrt cuminc: max degree,ii_betweenness_centrality,ii_eigenvector_centrality

ggplot() +
  geom_point(aes(x=sc.set.limit$maximum_degree,
                 y=sc.set.original$cumulative_incidence), shape = ".") +
  labs(x="Maximum Degree II", y="Cumulative Incidence", size=6)

ggplot(data=sc.set.limit) +
  geom_point(aes(x=sc.set.limit$betweenness_centrality,
                 y=sc.set.limit$cumulative_incidence), shape = ".") +
```

```r
  labs(x="Betweenness Centrality", y="Cumulative Incidence", size=6)

ggplot() +
  geom_point(aes(x=sc.set.limit$ii_closeness_centrality,
                 y=sc.set.original$cumulative_incidence), shape = ".") +
  labs(x="II Closeness Centrality", y="Cumulative Incidence", size=6)

ggplot(data=sc.set.limit) +
  geom_point(aes(x=sc.set.limit$ii_eigenvector_centrality,
                 y=sc.set.limit$cumulative_incidence), shape = ".") +
  labs(x="II Eigenvector Centrality", y="Cumulative Incidence", size=6)

ggplot() +
  geom_point(aes(x=sc.set.limit$component_size,
                 y=sc.set.original$cumulative_incidence), shape = ".") +
  labs(x="Component Size", y="Cumulative Incidence", size=6)

ggplot(data=sc.set.limit) +
  geom_point(aes(x=sc.set.limit$component_density,
                 y=sc.set.limit$cumulative_incidence), shape = ".") +
  labs(x="Density", y="Cumulative Incidence", size=6)

ggplot() +
  geom_point(aes(x=sc.set.limit$geodesic_dist,
                 y=sc.set.original$cumulative_incidence), shape = ".") +
  labs(x="Avg Geodesic Distance", y="Cumulative Incidence", size=6)

ggplot(data=sc.set.limit) +
  geom_point(aes(x=sc.set.limit$centralization,
                 y=sc.set.limit$cumulative_incidence), shape = ".") +
  labs(x="Centralization", y="Cumulative Incidence", size=6)

############################################
# Laplace Visualizations
x<-seq(-10,10,0.1)
plot(x=seq(-10,10,0.1), y=LaplacesDemon::dlaplace(x,0,1),
     col="black", type = "l",
     xlim=c(-10,10), ylim=c(0,0.5),
     xlab="", ylab="",
     main="Laplace Functions")
lines(x=seq(-10,10,0.1), y=LaplacesDemon::dlaplace(x,0,2), col="darkblue")
lines(x=seq(-10,10,0.1), y=LaplacesDemon::dlaplace(x,0,3), col="lightblue")
legend("topright",col = c("black","darkblue", "lightblue"),
       legend=c("Laplace(0, 1)","Laplace(0, 2)", "Laplace(0, 3)"),
       lty = c(1,1,1))
############################################
# standardize/center/scale
sc.set.limit <- data.frame(scale(sc.set.limit))
sc.set.cum.inc <- data.frame(scale(sc.set.cum.inc))
sc.set.ten.inc.inf <- data.frame(scale(sc.set.ten.inc.inf))
############################################
# CI Frequentist Lasso
set.seed(100)
```

```r
#initializations
grid=10^seq(10,-2,length=100)
x <- model.matrix(cumulative_incidence~.,data=sc.set.cum.inc)[,-15]
y <- sc.set.cum.inc$cumulative_incidence
# find best fit lambda
cv.out.lasso=cv.glmnet(x,y,alpha=1,nfolds = 10, type.measure="mse")
#plot(cv.out.lasso)
bestlam.lasso=cv.out.lasso$lambda.min
# use bestlam and lasso entire data set
out.lasso=glmnet((model.matrix(cumulative_incidence~.,data=sc.set.cum.inc)[,-15]),
                 sc.set.cum.inc$cumulative_incidence,alpha=1,lambda=bestlam.lasso)
lasso.coef=predict(out.lasso,type="coefficients",s=bestlam.lasso)[0:15,]
lasso.coef.non.zero <- lasso.coef#[lasso.coef!=0]
no.coef.lasso <- length(lasso.coef.non.zero)-1
#### betweenness_centrality, ii_closeness_centrality, ii_eigenvector_centrality, component_density
# predictions and performance

lasso.coef.non.zero.df <- data.frame(lasso.coef.non.zero)
rownames(lasso.coef.non.zero.df) <- c("Intercept", "Maximum Degree",
                                      "Betweenness Centrality",
                                      "II Betweenness Centrality",
                                      "II Closeness Centrality",
                                      "II Eigenvector Centrality",
                                      "Component Size", "Number Connected Components",
                                      "Component Density", "Geodesic Distance", "II Geodesic Distance",
                                      "Centralization", "Prop2Cores", "Transitivity","Diameter")
colnames(lasso.coef.non.zero.df) <- c("")
lasso.coef.non.zero.df <- round(lasso.coef.non.zero.df, 3)
kable(lasso.coef.non.zero.df, "latex", caption = "Frequentist Lasso,
      Cross Validated Lambda", booktabs = T) %>%
  kable_styling(latex_options = c("striped", "hold_position"))
############################################
# CI Bayesian Lasso, Cross Validated Lambda
# Use cv.EBglmnet to get lambda value
cv.EBglmnet(as.matrix(sc.set.cum.inc[,1:14]), as.matrix(sc.set.cum.inc[,"cumulative_incidence"]), famil
            prior= "lassoNEG", nfolds=5, Epis = FALSE, group = FALSE, verbose = 0)
lasso.cv <- cv.EBglmnet(as.matrix(sc.set.cum.inc[,1:14]), as.matrix(sc.set.cum.inc[,"cumulative_inciden
            prior= "lasso", nfolds=5, Epis = FALSE, group = FALSE, verbose = 0)
#lasso.cv
output <- lasso.cv$fit

# Use stan_glm to employ found lambda value
glm_post_cvlam <- stan_glm(cumulative_incidence ~ .,
                           data = sc.set.cum.inc,
                           prior = laplace(0,0.01201505),
                           prior_intercept = NULL)
                           #prior_aux = )
                           #autoscale = TRUE)
#summary(glm_post_cvlam)
############################################
# Build and Print Table
out.glm_post_cvlam <- summary(glm_post_cvlam)[1:15,c(3:6,8)]
out.glm_post_cvlam <- out.glm_post_cvlam[, c(2, 3, 4, 1, 5)]
out.glm_post_cvlam <- data.frame(round(out.glm_post_cvlam, 2))
colnames(out.glm_post_cvlam) <- c("10%", "50%", "90%", "Std Dev", "Rhat")
rownames(out.glm_post_cvlam) <- c("Intercept", "Maximum Degree",
                                  "Betweenness Centrality",
                                  "II Betweenness Centrality",
                                  "II Closeness Centrality",
                                  "II Eigenvector Centrality",
                                  "Component Size", "Number Connected Components",
                                  "Component Density", "Geodesic Distance", "II Geodesic Distance",
                                  "Centralization", "Prop2Cores", "Transitivity","Diameter")

kable(out.glm_post_cvlam, "latex",
      caption = "Bayesian Lasso, Cross Validated Lambda", booktabs = T) %>%
  kable_styling(latex_options = c("striped", "hold_position"))
############################################
# Look at trace plots
stan_trace(glm_post_cvlam)
# _____________________________________________________________________________________________________
############################################
#heirarchical model functions

beta.post.sample <- function(x,y,invTau2,sigma2,lambda) {
  XtX <- t(x) %*% x
  xy <- t(x) %*% y

  invD <- diag(invTau2)
  invA <- solve(XtX + invD)
  mean <- invA %*% xy
  varcov <- sigma2 * invA
  beta <- drop(rmnorm(1, mean, varcov))
  return(beta)
}

sigma.post.sample <- function(x,y,beta,invTau2) {
  n <- 1000 #num rows
  invD <- diag(invTau2)
  shape <- (n+14-1)/2
  residuals <- drop(y - x %*% beta)
  scale <- (t(residuals) %*% residuals + t(beta) %*% invD %*% beta)/2
  sigma2 <- 1/rgamma(1, shape, 1/scale)
  return(sigma2)
}

tau.post.sample <- function(x,y,beta,sigma2,lambda) {
  muPrime <- sqrt(lambda^2 * sigma2 / beta^2)
  lambdaPrime <- lambda^2
  invTau2 <- rep(0, 14)
  for (i in seq(14)) {
    invTau2[i] <- rinv.gaussian(1, muPrime[i], lambdaPrime)
  }
  return(invTau2)
}
```

```
lambda.post.sample <- function(x,y,invTau2) {
  r <- 1
  shape = r + 1/2
    delta <- 0.1
  scale = delta + sum(1/invTau2)/2
  lambda <- rgamma(1, shape, 1/scale)
  return(lambda)
}
############################################
#gibbs sampler function
sims <- 3000
x <- as.matrix(sc.set.cum.inc[,1:14])
y <- as.matrix(sc.set.cum.inc[,15])

gibbs.sampler <- function(x,y) {
  #initializations
  no.params <- 30
  param.storage <- matrix(NA, sims, no.params)
  colnames(param.storage) <- c("beta1", "beta2", "beta3", "beta4", "beta5", "beta6", "beta7",
                               "beta8", "beta9", "beta10", "beta11", "beta12", "beta13", "beta14",
                               "tau1", "tau2", "tau3", "tau4", "tau5", "tau6", "tau7",
                               "tau8", "tau9", "tau10", "tau11", "tau12", "tau13", "tau14",
                               "sigma2", "lambda")
  #to save some time
  XtX <- t(x) %*% x
    xy <- t(x) %*% y
  #starting values for betas, taus, sigma2 and lambda
  param.storage[1,1:14] <- drop(backsolve(XtX + diag(nrow=14), xy)) #starting betas seq(0.1,100,length.
  residuals <- drop(y - x %*% param.storage[1,1:14])
  param.storage[1,15:28] <- 1 / (param.storage[1,1:14] * param.storage[1,1:14]) #starting taus seq(0.1,
  param.storage[1,29] <- drop((t(residuals) %*% residuals) / 1000) #starting sigma 100#
  param.storage[1,30] <- 14*sqrt(param.storage[1,29]) / sum(abs(param.storage[1,1:14])) #starting lambd

  #interations
  for (i in 2:sims) {
    if (i %% 1000 == 0) {
      cat('Iteration:', i, "\r")
    }
    param.storage[i,1:14] <- beta.post.sample(x, y, param.storage[i-1,15:28],
                                              param.storage[i-1,29], param.storage[i-1,30])
    param.storage[i,29] <- sigma.post.sample(x, y, param.storage[i-1,1:14],
                                             param.storage[i-1,15:28])
    param.storage[i,30] <- lambda.post.sample(x, y, param.storage[i-1,15:28])
    param.storage[i,15:28] <- tau.post.sample(x, y, param.storage[i,1:14],
                                              param.storage[i,29],param.storage[i,30])
  }
  return(param.storage)
}

#run sampler!
param.storage <- gibbs.sampler(x, y)
param.storage.chain.2 <- gibbs.sampler(x, y)
param.storage.chain.3 <- gibbs.sampler(x, y)
param.storage.chain.4 <- gibbs.sampler(x, y)

#snag output (second halves as directed in text)
gibbs.output <- param.storage[seq(sims/2+1, sims, 1),]
gibbs.output[,15:29] <- sqrt(gibbs.output[,15:29])
gibbs.output[,30] <- sqrt(gibbs.output[,30])
#summary table
df.output <- data.frame(t(apply(gibbs.output, 2, function(x) quantile(x, c(.025,.25,.5,.75,.975)))))
df.output <- round(df.output, 3)
df.output <- df.output[c(1:14,29,30),]
colnames(df.output) <- c("2.5%", "25%", "50%", "75%", "97.5%")
rownames(df.output) <- c("Maximum Degree",
                         "Betweenness Centrality",
                         "II Betweenness Centrality",
                         "II Closeness Centrality",
                         "II Eigenvector Centrality",
                         "Component Size", "Number Connected Components",
                         "Component Density", "Geodesic Distance", "II Geodesic Distance",
                         "Centralization", "Prop2Cores", "Transitivity","Diameter",
                         "Sigma", "Lambda")
kable(df.output, "latex", caption = "Estimates from My Gibbs Sampler", booktabs = T) %>%
  kable_styling(latex_options = c("striped", "hold_position"))
############################################
#convergence?
param.storage.df<-data.frame(param.storage)
param.storage.chain.2.df<-data.frame(param.storage.chain.2)
param.storage.chain.3.df<-data.frame(param.storage.chain.3)
param.storage.chain.4.df<-data.frame(param.storage.chain.4)

vars <- param.storage.df[,c(1:14,30)]
for (i in 1:ncol(vars)) {
  plot <- ggplot() +
          geom_line(data=param.storage.df, aes(x=seq(1:sims), y=vars[,i]), colour = alpha("purple")) +
          geom_line(data=param.storage.chain.2.df, aes(x=seq(1:sims), y=vars[,i]), colour = alpha("blue
          #geom_line(data=param.storage.chain.3.df, aes(x=seq(1:sims), y=vars[,i]), colour = alpha("yel
          #geom_line(data=param.storage.chain.4.df, aes(x=seq(1:sims), y=vars[,i]), colour = alpha("red
          labs(x="No. of Iterations", y=colnames(vars[i])) +
          ggtitle(colnames(vars[i])) +
          theme_bw()
  print(plot)
}
############################################s
# Main Posterior Predictive Check
pp_check(glm_post_cvlam)
############################################
#  Posterior Predictive Checks for Bayesian Lasso, Diffuse Hyperprior Lambda

ppc_intervals(y = sc.set.limit$cumulative_incidence,
              yrep = posterior_predict(glm_post_hypprior.lam),
              x = sc.set.limit$maximum_degree)
ppc_intervals(y = sc.set.limit$cumulative_incidence,
              yrep = posterior_predict(glm_post_hypprior.lam),
              x = sc.set.limit$betweenness_centrality)
```

```
ppc_intervals(y = sc.set.cum.inc$cumulative_incidence,
              yrep = posterior_predict(glm_post_hypprior.lam),
              x = sc.set.cum.inc$ii_betweenness_centrality)
ppc_intervals(y = sc.set.cum.inc$cumulative_incidence,
              yrep = posterior_predict(glm_post_hypprior.lam),
              x = sc.set.cum.inc$ii_closeness_centrality)
ppc_intervals(y = sc.set.cum.inc$cumulative_incidence,
              yrep = posterior_predict(glm_post_hypprior.lam),
              x = sc.set.cum.inc$ii_eigenvector_centrality)
ppc_intervals(y = sc.set.cum.inc$cumulative_incidence,
              yrep = posterior_predict(glm_post_hypprior.lam),
              x = sc.set.cum.inc$component_size)
ppc_intervals(y = sc.set.limit$cumulative_incidence,
              yrep = posterior_predict(glm_post_hypprior.lam),
              x = sc.set.limit$number_connected_components)
ppc_intervals(y = sc.set.limit$cumulative_incidence,
              yrep = posterior_predict(glm_post_hypprior.lam),
              x = sc.set.limit$component_density)
ppc_intervals(y = sc.set.cum.inc$cumulative_incidence,
              yrep = posterior_predict(glm_post_hypprior.lam),
              x = sc.set.cum.inc$geodesic_dist)
ppc_intervals(y = sc.set.cum.inc$cumulative_incidence,
              yrep = posterior_predict(glm_post_hypprior.lam),
              x = sc.set.cum.inc$ii_geodesic_distance)
ppc_intervals(y = sc.set.cum.inc$cumulative_incidence,
              yrep = posterior_predict(glm_post_hypprior.lam),
              x = sc.set.cum.inc$centralization)
ppc_intervals(y = sc.set.cum.inc$cumulative_incidence,
              yrep = posterior_predict(glm_post_hypprior.lam),
              x = sc.set.cum.inc$prop_2_cores)
ppc_intervals(y = sc.set.cum.inc$cumulative_incidence,
              yrep = posterior_predict(glm_post_hypprior.lam),
              x = sc.set.cum.inc$transitivity)
ppc_intervals(y = sc.set.cum.inc$cumulative_incidence,
              yrep = posterior_predict(glm_post_hypprior.lam),
              x = sc.set.cum.inc$diameter)
#############################################
#  Posterior Predictive Checks for Bayesian Lasso, Diffuse Hyperprior Lambda
stan_hist(glm_post_cvlam, pars=c("maximum_degree","betweenness_centrality",
                                        "ii_betweenness_centrality",
                                        "ii_closeness_centrality",
                                        "ii_eigenvector_centrality", "component_size"),
          bins=40, fill_color="blue")

stan_hist(glm_post_cvlam, pars=c("number_connected_components",
                                        "component_density",
                                        "geodesic_dist","centralization",
                                        "prop_2_cores", "diameter"),
          bins=40, fill_color="blue")

#post_samps_speed <- as.data.frame(glm_post_cvlam, pars=c("geodesic_dist"))[,"geodesic_dist"]
#mn_speed <- mean(post_samps_speed) # posterior mean
#ci_speed <- quantile(post_samps_speed, probs=c(0.05, 0.95))
```

```r
############################################
#remove some outliers
for (ii in 1:20) {
  for(i in 1:ncol(vars)) {
    param.storage.df[,i][which.max(param.storage.df[,i])] <- median(param.storage.df[,i])
    param.storage.df[,i][which.min(param.storage.df[,i])] <- median(param.storage.df[,i])
  }
}

#plot posterior densities
for (i in 1:ncol(vars)) {
  plot <- ggplot(param.storage.df, aes(x=vars[,i])) +
          geom_histogram(color="black", fill="white",bins=50) +
          labs(x=colnames(vars[i]), y="Count") +
          ggtitle(colnames(vars[i])) +
          theme_bw()
  print(plot)
}
```