Methodology:

Introducing Stan and the Bayesian Framework:

To approach the research question at hand, this paper employs a Bayesian framework. As Bayesian analysis is still in the minority to frequentist statistics, this paper will spend a portion of the methodology section explaining the framework, and why it is an improvement on other approaches.

To implement the Bayesian approach we will take advantage of the Stan programing language. Stan is a probabilistic high level programming language that is used to specify statistical models. A Stan program defines a log probability function over parameters conditioned on specified data and constraints. Stan provides the user with the ability to conduct full Bayesian inference through Markov chain Monte Carlo methods such as No-U-Turn sampling. Stan is set up such that densities, gradients, and Hessians, along with intermediate quantities are easily accessible. Prior to Stan it was difficult to accurately compute Bayes rule and apply it to research settings.

Stan utilizes markov-chain monte carlo simulation to get draws from the posterior predicitive distribution. A Markov process is a sequence of random variables with a particular dependence structure where the future is conditionally independent of the past given the present, but noting is marginally independent of anything else. We can construct a Markov process such that the marginal distribution of a random variable is a given target distribution as the number of simulations moves to infinity. As a result, we can get a random draw, or a set of dependent draws, from the target distribution by letting the Markov process to run for many interations.

Stan specifically uses a form of Markov chain Monte Carlo simulation called No-U-Turn Sampling. First, a quick review of ancient Markov chain Monte Carlo samplers. Metropolis-Hasitings only requires user to specify the numerator of Bayes Rule. However, only 22 percent of proposals ideally get accepted to get relatively big jumps in the sampling process. The effective sample size, as a result, can be essentially zero. Gibbs sampling, in general, forces a user to work out all full-conditional distributions. Jumps are always accepted in the sampling process, but they might not be very big. Effective sample size is low if the parameters are highly correlated. Stan, like Metropolis-Hastings, only requires the user to specify the numerator of Bayes Rule. Like Metrolopis-Hastings, but unlike Gibs sampling, proposals are joint. Unlike Metropolis-Hastings, but like Gibbs sampling, proposals are always accepted and tuning of proposals is (often) not required. Unlike both Metropolis-Hastings and Gibbs sampling, the effective sample size is typically 25 percent to 125% of the nominal number of draws from the posterior distribution. As mention, Stan utilizes No U-Turn Sampling (NUTS) as its sampler for the Markov Chain Monte Carlo Simulation. The location of Θ moving according to Hamiltonian physics at any instant would be a valid draw from the posterior distribution. The challenge became then determing when to stop as Θ moves indefinitely (in the absence of friction). Hoffman and Gelman (2014) proposed stopping when there is a U-Turn. In the sense that the footprints found in the monte carlo simulation turn around and start to head in the direction they just came from. After the U-Turn, one footprint is selected with probability proportional to the posterior kernel to be the realization of Θ on interation *s* and the process repeats itself S times. NUTS discretizes a continuous-time Hamiltonian process in order to solve a system of Ordinary Differential Equations (ODEs). These ODEs require a step size that is also tuned during the warmup phase of the Monte Carlo simulation.

Model Methodology:

To explore the data generating process for the variable of interest, general subsidy levels, I will use the Bayesian regression framework. Bayesian regression differs from frequentist analysis in that it does not rely on a single point estimate for a parameter. Instead it looks at the parameter across the entire sample space and is able to generate a posterior distribution, or a range of values, within which the parameter might probabalisiticaly fall. This is far more helpful in understanding exactly the values that the parameter could take when faced with new data. In this way we can gain a more complete understanding of the data generating process. We will build 5 models in our process, compare them using leave one out cross validation, and use projection prediction to see if our more complicated models can be expressed with fewer variables.

Our five models will grow in complexity by adding more potential variables that potentially aid to the data generating process for general subsidy levels. In model one we will only include the intercept and investment returns, which are defined as the change in market value of the endowment portfolio between time equal to t-1 and time equal to t. In model 2 we will also include non-instructional expenses, which is defined as total expenses less instructional E&G&K expenses. In model 3 we will include the categorical variable year to see if any differences in general subsidy are generated based on the year the data was collected. In model 4 we will include a more granular break out of non-tuition revenue sources including: total gift revenue, total grants and appropriations revenue, net auxiliary revenue, net hospital revenue, and net other revenue.

To compare the models presented in the paper we will use leave one out cross validation. Penalty functions that are used in supervised learning make for poor priors because they are intended to shift the mode rather than reflect the authors original prior beliefs. Supervised learning does not attempt to quantify the uncertainty in parameters or model choice. This differs sharply from the Bayesian approach which seeks to track, as closely as possible, the types of uncertainty that are being introduced during the model process. While supervised learning attempts to optimize for a single point estimate, there is a more complete and improved way to approach the model validation problem. For Bayesians, we estimate over an entire function, in this case the log of the predictive density function is most appropriate. We will seek to choose a model that maximizes the expectation of the log predicitive density function over future data.

Leave one out cross validation can be described with the following procedure. Obtain draws from the posterior PDF and condition of those posterior draws to draw from the posterior predicitive distribution for the outcome variable of interest. Omitt the ith observation and draw from the posterior PDF evaluating the log-likelihood for the ith observation. Repeat this process across the posterior draws of Θ to compute the expected log predictive density. Though in absolute terms this does not mean much, it does allow us to compare between models. We are looking to see how concentrated the prediction density function is over the distribution of the real data. Leave one out cross validation utilizes Pareto smoothed importance sampling. Pareto smoothed importance sampling provides a move accurate and reliable estimate by fitting a Pareto distribution to the upper tail of the importance weights distribution. Pareto smoothed importance sampling allows us to compute a leave one out cross validation using importance weights that would otherwise be unreliable and noisy.