Final Paper (DRAFT)

Tara Pozzi

4/14/2020

## Introduction

Risk perception is a fundamental way to characterize a person’s intuitive risk judgement and allows for the identification, characterization, and quantification of risk. Past literature has argued that those with higher risk perception are more likely to take caution and prepare for natural hazards such as floods, fires, earthquakes, etc. (Vinh Hung, Shaw, and Kobayashi 2007). However, several studies have found a lack of correlation between high risk perception’s effect on protective behavioral actions towards natural disasters; which Wachinger et al. identities this unexpected response as the “Risk Perception Paradox.” This research project will dive into this paradox to determine if risk perception is or isn’t reflected in a flood risk manager’s decision to adopt a new technology called Light Detection and Ranging (LiDAR). Lidar is a laser-based survey instrument that captures a high-resolution spatial image of the earth’s surface (e.g. 1-meter resolution).

There are several factors that determine a person’s risk perception. The first area is direct experiences which have been found to have an amplificatory effect on risk perception; this includes an individual’s personal experiences with natural hazard events (the largest effect) and stakehold in the community such as home ownership (Lujala, Lein, and Rød 2015). The second important area of influence is trust both for scientific technology and authorities to ensure that they are being used to protect against natural hazards (Wachinger et al. 2013; Viglione et al. 2014). This has been researched at the layperson level, however I am also curious if it holds true at the authority level. Specifically, I am interested in an individual’s trust in science and the federal government for help in flood risk management. Trust in science will assess whether the person trusts the scientific information that is being provided to them about how to best manage their flood risk. Trust in the federal government will specifically assess if the person trusts that the government has the best interest of the community in mind when providing help with flood risk management. Numerous studies have showed the crucial role trust plays in building innovation, adaptive capacity, and resilience into a system, whether that system in ecological, technological, or social (Chapin et al. 2010; Luo et al. 2010; Viglione et al. 2014).

The third important area that could play an effect in a person’s risk perception is their belief in the potential for increased size and frequency of natural hazard events in their community. There are mixed results on the effect of increased frequency and size of events on an individual’s risk perception (Wachinger et al. 2013). Some studies have found there to be an significant impact from a person’s environmental beliefs and their behavioral intentions (O’Connor, Bard, and Fisher 1999). I am curious if there is a correlation between these environmental beliefs and their risk perception which could consequently then affect their behavioral intentions. Lastly, demographics play a significant role in risk perception and will be taken into account during this analysis (Savage 1993)

The current literature calls for a theoretical framework to quantify flood risk in a way that accurately takes into account the social dimension side risk with the inclusion of risk perceptions (Birkholz et al. 2014; Kellens, Terpstra, and De Maeyer 2013). This work aims to create a theoretical framework that can be replicated to assess risk perceptions for other types of hazards, as well as locations to ultimately help decrease vulnerability and increase adaptive capacity and resilience in communities. The main objectives of this work are:

1. Use a survey instrument to assess four predictor categories of risk perception: direct experience, trust, environmental beliefs, and demographics. In addition, identify if survey respondent uses LiDAR.
2. Create a well-fit model that can accurately describe the relationship between risk perception and LiDAR use.

## Hypotheses

Hypothesis 1 (H1): Flood risk managers with greater perceived risk of damaging floods are more likely to adopt LiDAR.

Null: Perceived risk does not affect adoption of LiDAR in flood risk managers.

Hypothesis 2 (H2): Direct experiences will have a more singificant effect on LiDAR use, than other risk perceptions measures of trust, environmental beliefs, and demographics.

Null: Direct experiences with floods will have the same affect as trust, environmental beliefs, and demographics on the adoption of LiDAR.

## Methods

*Data Collection*

All of the predictors as well as the response variable will be collected by a survey instrument (Appendix A). This is a modified survey instrument that was made specifically for this project, however it is part of a larger survey that will be sent out to flood risk managers across the state of Idaho. However, I have not sent this survey out for response yet, so instead I simulated data with the exact questions that I will eventually have real data for. Within each category of risk perception, there are additional sub-variables that are outlined in Table 1. In addition, the survey asks the respondent whether they currently use lidar or not which gathers the response variable data.

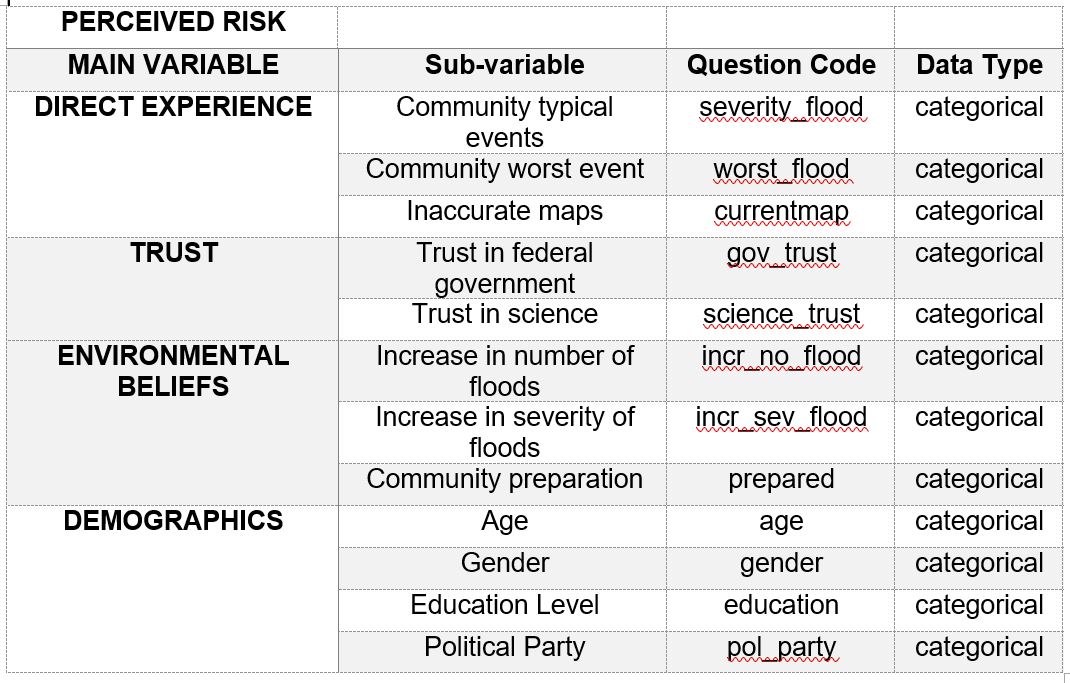


Table 1 shows four main categories of predictors: direct experience, trust, environmental beliefs, and demographics. These four predictor categories make-up a person’s risk perception. Within these four categories are several sub-variables that are potential canidates for this analysis. Direct experience includes typical community flood events, the worst community flood event, the economic cost of floods per year, and if a community has inaccurate flood maps. The trust category is made up of the respondent’s trust in science and the federal government. The environmental beliefs category is made up of the respondent’s climate change beliefs measured by their belief in changing number and severity of flood, as well as their community’s preparedness. The fourth category is the respondent’s demographics which includes age, gender, education level, and political party.

*Data Simulation*

A multivariate logisic regression (with binomial distribution) was chosen to model the risk perception predictors and lidar use. The reason I chose a binomial distribution is because my response variable is yes or no in regards to LiDAR adoption. This analysis has multiple variables so I created a didatic acyclical graph (DAG) to explain how the categories of predictors may influence one another and to help identify any potential confounders.

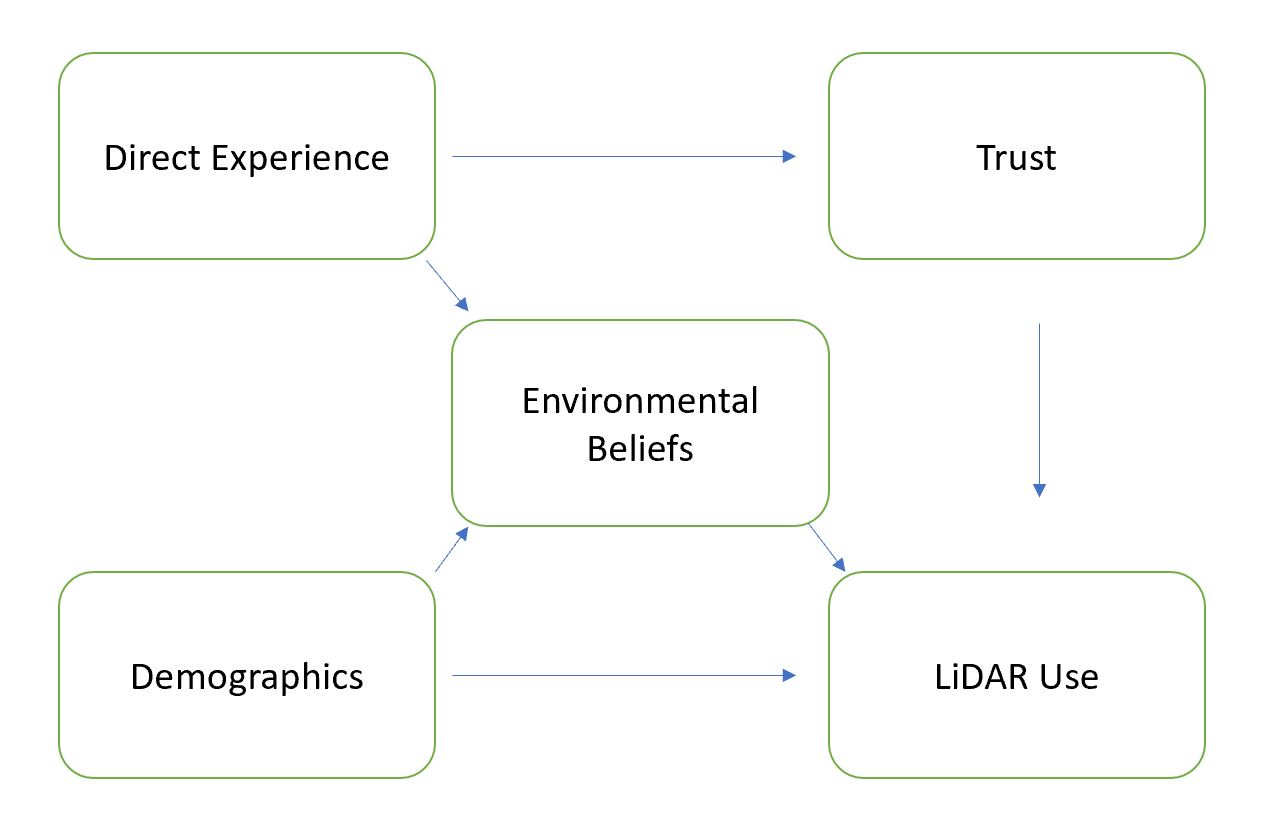


Table 1 shows the DAG diagram for this model where direct experience could potentially be the confounding variable influencing environmental beliefs and trust. In addition, demographics could be influencing environmental beliefs as well as lidar use directly.

Starting with this DAG in mind, I began my data analysis with data simulation. The first step in this process was to specify the number of respondents and the slope estimates for each predictor value. I began with 12 predictor values and decided to narrow down the predictors due to possible confounders. Specifically, I believe there is potential for incr\_no\_flood and incr\_sev\_flood to be correlated, so for the purposes of this example I decided to leave out the incr\_sev\_flood variable. There is also potential for age and pol\_party to be correlated, so for the purposes of this example I left out age. Lastly, I decided to leave out education in order to simplify the number of variables I have in this simulation. After removing these three predictors I was left with nine predictors. Table 2 shows the updated chart of predictors for this analysis.

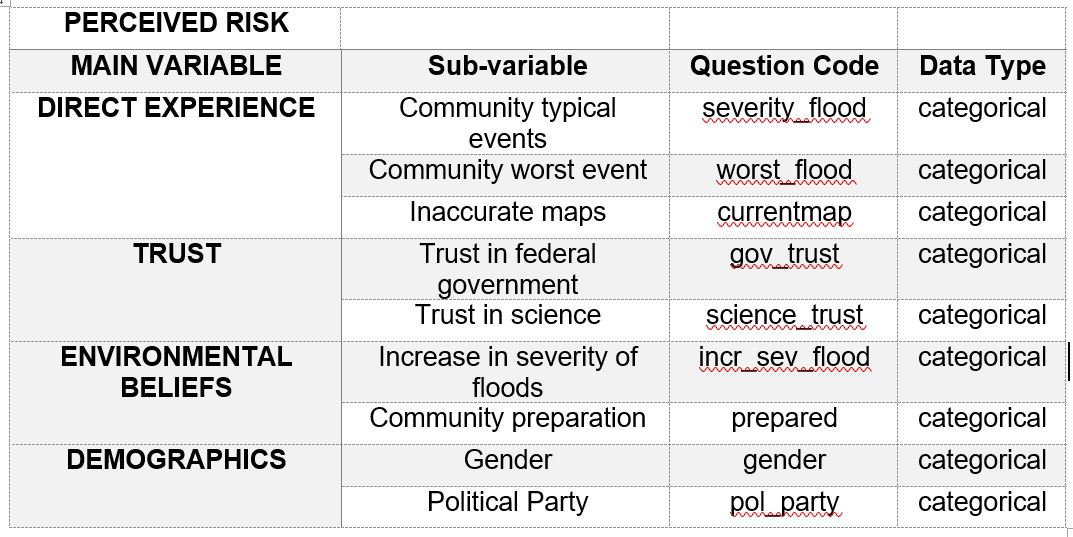


Table 2 shows the updated four main categories of predictors: direct experience, trust, environmental beliefs, and demographics.

*Model Fitting and Comparison*

I took a Bayesian approach for this analysis and ran each model using the stam\_glm function with a binomial specification. The first model I ran included all nine predictor variables. I then plotted each predictor’s affect on lidar use and looked for non-zero parameters. From this, I was able to identify that severity\_flood, worst\_flood, and pol\_party had non-zero effects on lidar use. Although, I was not able to determine exactly what the effect was due to my predictor variables being uncentered and on different scales. Next, I created two additional models to compare to the full model to determine which model had the best fit. The second model looked at severity\_flood, worst\_flood, and pol\_party on lidar use and the third model looked at direct experience on lidar use. The three model’s performances were determined through the comparison of RMSE, MAE, and LOOIC values. Afther this, the model with the lowest LOOIC value and therefore “least bad” fit was selected and examined further to determine the individual effects of each predictor on lidar use.

*Model interpretation and Communication*

Next, I took this model and created a counterfactual plot for each variable of interest. The plot displayed the effect of one predictor on lidar use, while the other two predictors were held at the mininum value. This was done in order to try and isolate the effect of the predictor of interst on lidar use. Next, I created two more counterfactual plots to compare each predictors isolated effect on lidar use.

## Results

Note: I accidentally saved over my simulated data with a new dataset so my numbers are not reproducible with the code as is and are not reflected in the code in Appendix B. However, the numbers are similar.

Model 1 has the best fit in this analysis:

* When the full model was run, the predictors that had a potential significant impact on lidar use were severity\_flood, worst\_flood, and pol\_party. This is because of their minimal overlap with zero.
* Then two additional models were made, Model 1 and Model 2. The results from the model comparison show that model one is the least bad model with a LOOIC of 275.3, with model two after that with 278.2, and the full model with the highest LOOIC of 287.9. This means that the model that includes the three parameters: severity\_flood, worst\_flood, and pol\_party has the best model fit. Model 1 therefore has the least bad predictive capacity because of LOOIC= 275.3.

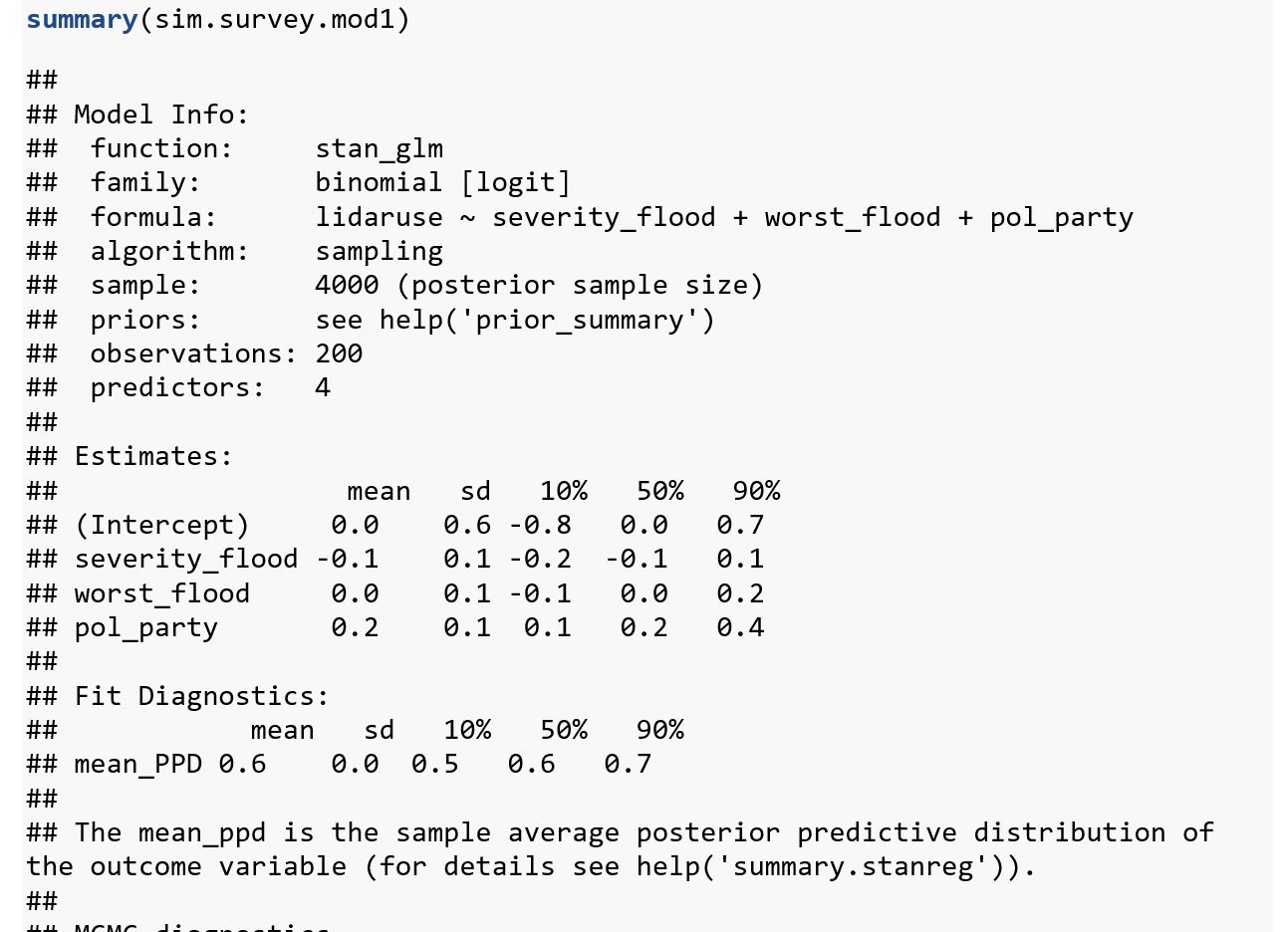


Table 4 displays the results from model one.

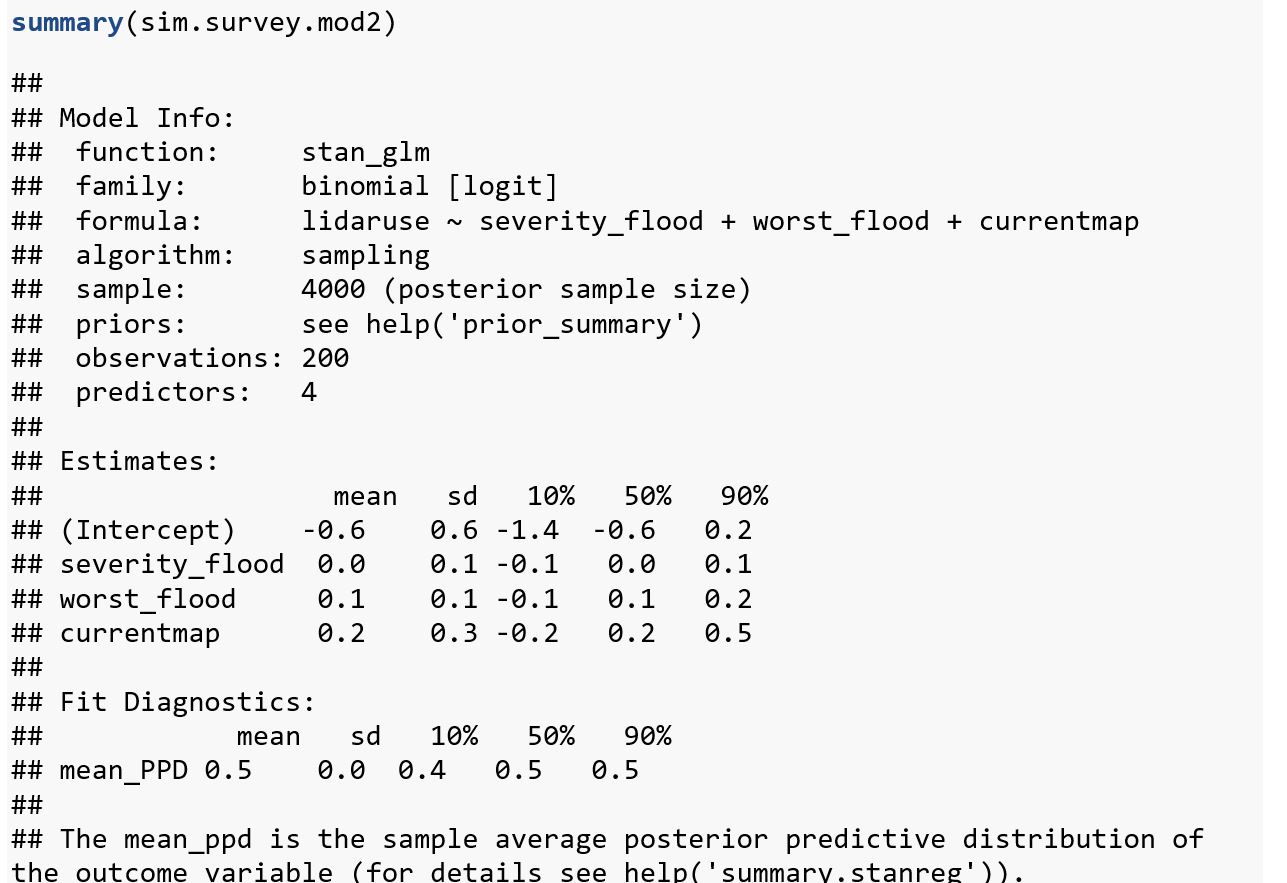


Table 5 displays the results from model two.

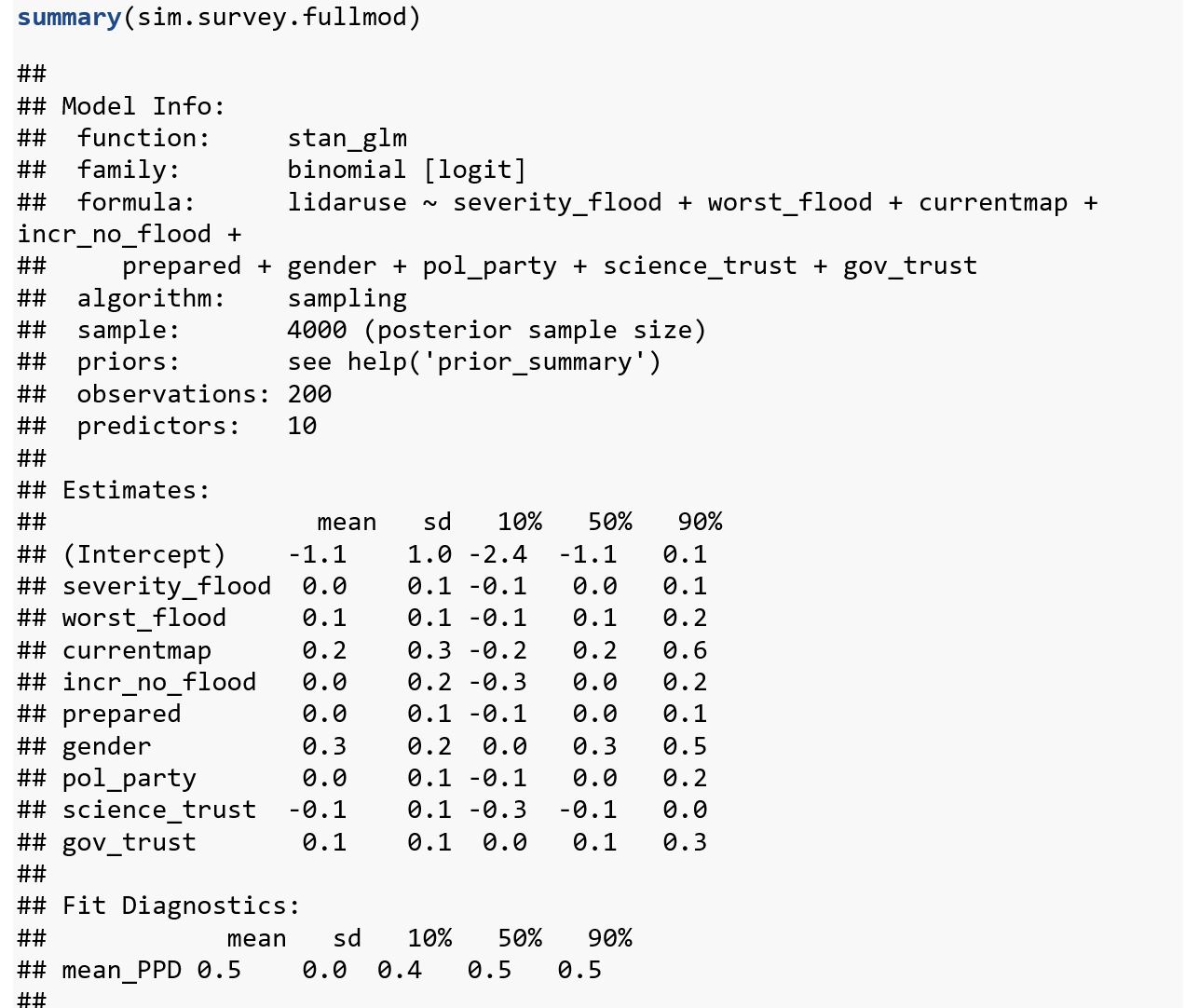


Table 6 displays the results from the full model.

* Model 1 has a RMSE = 0.628 and MAE = 0.39 which are reasonable variations from the mean for lidar use.

Model 1 results:

* The results summary shows that the values for mcse (all equal to 0), Rhat (all equal to 1), and n\_eff (high values) are good.
* Intercept (in this case lidar use) has a mean of 0, which transformed is .5. This intercept value matches with the value I set in the data simulation.
* Parameter estimates from Model 1 are: severity\_flood = -.1, worst\_flood = 0, and pol\_party = 0.2. It is hard to say if this parameter estimates support either of my hypotheses because of several reasons that are explained in the discussion portion of this report.
* Severity\_flood and pol\_party both have an effect on lidar use, however in order to determine their individual effect on lidar use a counterfactual plot needs to be made. Variables are each on a different scale so they can’t be directly compared in a plot all together.
  + The following are the counterfactual plots, however each one didn’t display the expected effect of the parameter on lidar use based on how I set the parameters. Despite this, I made the plots to show what my current analysis is resulting with:

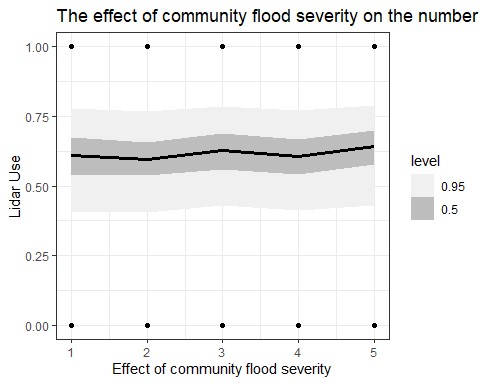


Figure 1 shows a counterfactural plot where worst\_flood and pol\_party are held constant, severity\_flood results in a mean lidar use of 0.54 and mean severity\_flood of 3, meaning moderate flood severity.

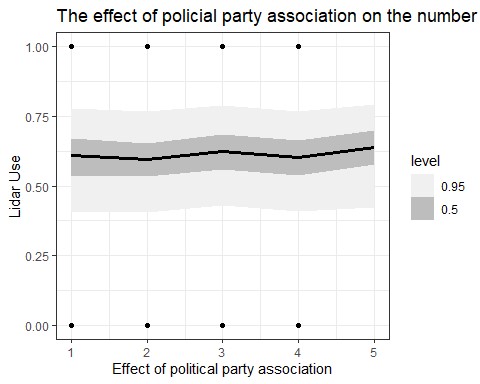


Figure 2 shows a counterfactual plot where worst\_flood and severity\_flood are held constant, pol\_party results in a mean lidar use of 0.54 and the mean political party of 2.5, but since this is a categorical question it doesn’t make sense to look at this parameter value in this way. Based on a histogram for this predictor, category 1 and 4 had the largest frequency of respondents. Category 1 was democrat and category 4 was other.

## Discussion

* I ended up having quite a few complications with my model and therefore can’t draw any conclusions about my hypotheses. Primarily, I was unable to recover the effects I set in the simulation. I think that this is because I have nine variables and each one was set with a small slope and therefore had a small effect. This in turn made it hard for the model to reflect the effects I was expecting.
  + Next time I run this analysis, I want to limit the number of initial variables to five. Then I want to set the effect for two or three variables to larger slopes in hopes that my model will show these effects more clearly.
* The counterfactual plots in my analysis all returned a similar effect between the predictor and lidar use. I think that my analysis returned this because of the interplay between small slopes and categorical nature of my data. For instance, both severity\_flood and pol\_party had a slope of .005 where as worst\_flood was .02 however all three predictors returned a similar effect on lidar use but they have different number of categories. More specifically, the code for sequencing each predictor in the counterfactual plot has 200 random draws between 1 and 5; this is a very limited range and perhaps the pattern of the data is not able to emerge fully due to this limited range.
  + I could try plotting the effect of my predictors on lidar use in a blox plot where the categorical component of the data may be easier to view.
  + I also think if I try larger slopes in the simulation part of the code, then I should be able to recover the slopes more clearly.
* In addition to these concerns, I also want to find a way to account for ordered categorical data in my analysis. I am going to read McElreath’s Chapter 12 about how to incorporate this in my model.
* Overall, I don’t think I was able to recover any meaningful results from this analysis however I was able to establish a methodology for looking at my data and now have a framework in place to run an analysis that will return meaningful results once I change a few key inputs!

## Appendix

# Apendix A

**Survey Instrument**

Q1. Light Detection and Ranging (LiDAR) is a laser-based technology that provides a detailed data map of bare earth, canopy, and other model’s of the earth’s surface. Do you currently use LiDAR?

o Yes o No

Q2. How severe are the consequences of flooding in your community, typically?

o Minor– no disruption of affected area o Minimal– short term minor economic consequences. Relocation and evacuation are not normally necessary. o Moderate– affected areas are disrupted; some areas evacuated or not habitable; dollar losses small but of consequence to those impacted; roads closed for short periods.  
o Significant– affected areas are essentially shut down; homes and/or basements flooded; economic losses are significant; requires temporary relocation of some; roads close for several hours; community infrastructure damaged.  
o Disastrous– equivalent to a major riverine flood; major community disruption; temporary relocation of many in affected areas; severe economic losses in affected areas; break up of social cohesion

Q3. What is the worst consequence of flooding your community has experienced?

o Minor– no disruption of affected area o Minimal– short term minor economic consequences. Relocation and evacuation are not normally necessary. o Moderate– affected areas are disrupted; some areas evacuated or not habitable; dollar losses small but of consequence to those impacted; roads closed for short periods. o Significant– affected areas are essentially shut down; homes and/or basements flooded; economic losses are significant; requires temporary relocation of some; roads close for several hours; community infrastructure damaged. o Disastrous– equivalent to a major riverine flood; major community disruption; temporary relocation of many in affected areas; severe economic losses in affected areas; break up of social cohesion

Q4. Do you think your community’s floodplain maps accurately reflect flood risk?

o Yes o No

Q5. In the future, do you think the number of flood events (of any level) in your community will increase, decrease, or stay the same as the current average?

o Increase o Decrease o Stay the same

Q6. In the future, do you think the severity of flood damage in your community will increase, decrease, or stay the same as the current averages?

o Increase o Decrease o Stay the Same

Q7. If you had to say, is your community prepared for a significant flood event?

o Completely prepared o Somewhat prepared o Neither prepared nor unprepared o Somewhat unprepared o Completely unprepared

Q8. What gender do you identify with?

o Male o Female o Other

Q9. What is your age?

o Less than 20 years o 20-29 years o 30-39 years o 40-49 years o 50+ years

Q10. What is the highest level of education you have completed?

o Some high school o High school diploma o College education, did not graduate  
o College education, Associates degree o College education, Bachelor’s degree o Advanced degree (MA, JD, MBA, PhD)

Q11. If you are registered with a political party, in which one are you registered? o Democrat o Independent o Republican o Other: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Q12. How much do you trust or distrust science as a source of information to improve flood risk management?

o Strongly trust o Somewhat trust o Neither trust nor distrust o Somewhat distrust o Strongly distrust

Q13. How much do you trust or distrust the federal government’s intent with flood risk management (i.e. data collection, accurate mapping, floodplain modeling, flood insurance)?

o Strongly trust o Somewhat trust o Neither trust nor distrust o Somewhat distrust o Strongly distrust

# Appendix B

**Please note that this simulated data is different than what I did my analysis with because I forgot to use set.seed to prevent it from overwriting my data frame. However, the code is the same as what I used.**

**Model Methods & Workflow**

*Create simulated data*

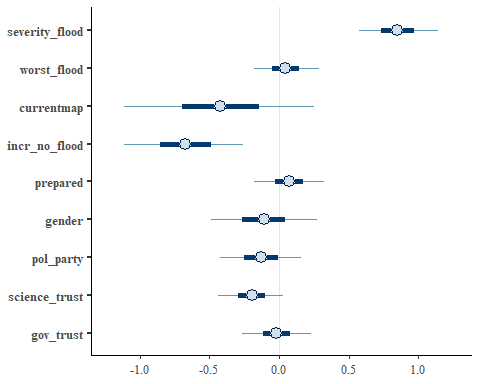
set.seed(200)  
N=200 # number of survey respondents  
  
## 1) Set the intercept ##  
  
intercept=0 ## mean value of lidar use when risk perception equals zero?  
  
## 2) Set the predictor variables ##  
severity\_flood= sample(1:5, N, replace=TRUE) # range of severity of floods, 1 being minor and 5 being disastrous  
worst\_flood= sample(1:5, N, replace=TRUE) # range of severity of floods, 1 being minor and 5 being disastrous  
currentmap= sample(1:2, N, replace=TRUE) # questions asks if respondents thinks their floodmaps accurately reflect their flood risk: 1 means no and 2 means yes  
incr\_no\_flood= sample(1:3, N, replace=TRUE) # this questions asks if respondents think the number of floods in their community is changings: 1 means increase, 2 means decrease, 3 means stay the same  
incr\_sev\_flood= sample(1:3, N, replace=TRUE) # this questions asks if respondents think the severity of floods in their community is changings: 1 means increase, 2 means decrease, 3 means stay the same  
prepared= sample(1:5, N, replace=TRUE) #this questions asks if respondents think their community is preparead for a flood event: 1 means completely prepared and 5 means completely unprepared  
gender= sample(1:3, N, replace=TRUE) # gender for this is: 1 male, 2 female, and 3 other  
age= sample(1:5, N, replace=TRUE) # age is categorical: where 1 is less than 20 years, 2:20-29 years, 3:30-39 years, 4:40-49 years, 5:50+ years  
education= sample(1:6, N, replace=TRUE) # this is the level of education of respondent: 1: some high school, 2:high school diploma, 3: college education, did not garduate, 4: associate's degree, 5: bachelor's degreee, and 6: advanced degree  
pol\_party= sample(1:4, N, replace=TRUE) # there are options for the respondent to choose: 1: democrat, 2: independent, 3: republican, 4: other  
science\_trust= sample(1:5, N, replace=TRUE) # this question asks if respondent trusts science: 1: strongly trust to 5: strongly distrust  
gov\_trust= sample(1:5, N, replace=TRUE) # this question asks if respondent trusts government: 1: strongly trust to 5: strongly distrust  
  
## 3) Potential correlations in data ##  
  
# there is potential for incr\_no\_flood and incr\_sev\_flood to be correlated, so for the purposes of this example I am going to leave out the incr\_sev\_flood variable  
# there is potential for age and pol\_party to be correlated, so for the purposes of this example I am going to leave out age  
# in addition, I am going to leave out education in order to simplify the number of variables I have in this simulation.   
  
## Simulating the response based on these variables ##  
b1=.5  
b2=0  
b3=0  
b4=0  
#b5= 0  
b6=0  
b7=0  
#b8= 0  
#b9= 0  
b10=.005  
b11=.01  
b12=.01  
p <- intercept+(b1\*severity\_flood)+(b2\*worst\_flood)+(b3\*currentmap)+(b4\*incr\_no\_flood)+(b6\*prepared)+(b7\*gender)+(b10\*pol\_party)+(b11\*science\_trust)+(b12\*gov\_trust)  
pr <- plogis(p)  
  
## 4) Set the response variable ##  
  
lidaruse <- rbinom(200,1,pr)  
  
## 5) Combine data into dataframe ##  
  
sim.survey <- data.frame(lidaruse, severity\_flood, worst\_flood, currentmap, incr\_no\_flood, prepared, gender, pol\_party, science\_trust, gov\_trust)  
  
write.csv(sim.survey, "sim\_survey\_results.csv")

*Fit a model to the simulated data*

# bring in the simulated data  
sim.survey <- read.csv("sim\_survey\_results.csv")  
# run the first model  
sim.survey.fullmod <- stan\_glm(lidaruse~ severity\_flood+ worst\_flood+ currentmap+ incr\_no\_flood+ prepared+ gender+ pol\_party+ science\_trust+ gov\_trust, data=sim.survey, family="binomial")

##   
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 1).  
## Chain 1:   
## Chain 1: Gradient evaluation took 0.001 seconds  
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 10 seconds.  
## Chain 1: Adjust your expectations accordingly!  
## Chain 1:   
## Chain 1:   
## Chain 1: Iteration: 1 / 2000 [ 0%] (Warmup)  
## Chain 1: Iteration: 200 / 2000 [ 10%] (Warmup)  
## Chain 1: Iteration: 400 / 2000 [ 20%] (Warmup)  
## Chain 1: Iteration: 600 / 2000 [ 30%] (Warmup)  
## Chain 1: Iteration: 800 / 2000 [ 40%] (Warmup)  
## Chain 1: Iteration: 1000 / 2000 [ 50%] (Warmup)  
## Chain 1: Iteration: 1001 / 2000 [ 50%] (Sampling)  
## Chain 1: Iteration: 1200 / 2000 [ 60%] (Sampling)  
## Chain 1: Iteration: 1400 / 2000 [ 70%] (Sampling)  
## Chain 1: Iteration: 1600 / 2000 [ 80%] (Sampling)  
## Chain 1: Iteration: 1800 / 2000 [ 90%] (Sampling)  
## Chain 1: Iteration: 2000 / 2000 [100%] (Sampling)  
## Chain 1:   
## Chain 1: Elapsed Time: 0.206 seconds (Warm-up)  
## Chain 1: 0.188 seconds (Sampling)  
## Chain 1: 0.394 seconds (Total)  
## Chain 1:   
##   
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 2).  
## Chain 2:   
## Chain 2: Gradient evaluation took 0 seconds  
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.  
## Chain 2: Adjust your expectations accordingly!  
## Chain 2:   
## Chain 2:   
## Chain 2: Iteration: 1 / 2000 [ 0%] (Warmup)  
## Chain 2: Iteration: 200 / 2000 [ 10%] (Warmup)  
## Chain 2: Iteration: 400 / 2000 [ 20%] (Warmup)  
## Chain 2: Iteration: 600 / 2000 [ 30%] (Warmup)  
## Chain 2: Iteration: 800 / 2000 [ 40%] (Warmup)  
## Chain 2: Iteration: 1000 / 2000 [ 50%] (Warmup)  
## Chain 2: Iteration: 1001 / 2000 [ 50%] (Sampling)  
## Chain 2: Iteration: 1200 / 2000 [ 60%] (Sampling)  
## Chain 2: Iteration: 1400 / 2000 [ 70%] (Sampling)  
## Chain 2: Iteration: 1600 / 2000 [ 80%] (Sampling)  
## Chain 2: Iteration: 1800 / 2000 [ 90%] (Sampling)  
## Chain 2: Iteration: 2000 / 2000 [100%] (Sampling)  
## Chain 2:   
## Chain 2: Elapsed Time: 0.216 seconds (Warm-up)  
## Chain 2: 0.216 seconds (Sampling)  
## Chain 2: 0.432 seconds (Total)  
## Chain 2:   
##   
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 3).  
## Chain 3:   
## Chain 3: Gradient evaluation took 0 seconds  
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.  
## Chain 3: Adjust your expectations accordingly!  
## Chain 3:   
## Chain 3:   
## Chain 3: Iteration: 1 / 2000 [ 0%] (Warmup)  
## Chain 3: Iteration: 200 / 2000 [ 10%] (Warmup)  
## Chain 3: Iteration: 400 / 2000 [ 20%] (Warmup)  
## Chain 3: Iteration: 600 / 2000 [ 30%] (Warmup)  
## Chain 3: Iteration: 800 / 2000 [ 40%] (Warmup)  
## Chain 3: Iteration: 1000 / 2000 [ 50%] (Warmup)  
## Chain 3: Iteration: 1001 / 2000 [ 50%] (Sampling)  
## Chain 3: Iteration: 1200 / 2000 [ 60%] (Sampling)  
## Chain 3: Iteration: 1400 / 2000 [ 70%] (Sampling)  
## Chain 3: Iteration: 1600 / 2000 [ 80%] (Sampling)  
## Chain 3: Iteration: 1800 / 2000 [ 90%] (Sampling)  
## Chain 3: Iteration: 2000 / 2000 [100%] (Sampling)  
## Chain 3:   
## Chain 3: Elapsed Time: 0.182 seconds (Warm-up)  
## Chain 3: 0.182 seconds (Sampling)  
## Chain 3: 0.364 seconds (Total)  
## Chain 3:   
##   
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 4).  
## Chain 4:   
## Chain 4: Gradient evaluation took 0 seconds  
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.  
## Chain 4: Adjust your expectations accordingly!  
## Chain 4:   
## Chain 4:   
## Chain 4: Iteration: 1 / 2000 [ 0%] (Warmup)  
## Chain 4: Iteration: 200 / 2000 [ 10%] (Warmup)  
## Chain 4: Iteration: 400 / 2000 [ 20%] (Warmup)  
## Chain 4: Iteration: 600 / 2000 [ 30%] (Warmup)  
## Chain 4: Iteration: 800 / 2000 [ 40%] (Warmup)  
## Chain 4: Iteration: 1000 / 2000 [ 50%] (Warmup)  
## Chain 4: Iteration: 1001 / 2000 [ 50%] (Sampling)  
## Chain 4: Iteration: 1200 / 2000 [ 60%] (Sampling)  
## Chain 4: Iteration: 1400 / 2000 [ 70%] (Sampling)  
## Chain 4: Iteration: 1600 / 2000 [ 80%] (Sampling)  
## Chain 4: Iteration: 1800 / 2000 [ 90%] (Sampling)  
## Chain 4: Iteration: 2000 / 2000 [100%] (Sampling)  
## Chain 4:   
## Chain 4: Elapsed Time: 0.186 seconds (Warm-up)  
## Chain 4: 0.223 seconds (Sampling)  
## Chain 4: 0.409 seconds (Total)  
## Chain 4:

# plot model  
plot(sim.survey.fullmod, pars="beta")



*Create additional models for comparison*

sim.survey.mod1 <- stan\_glm(lidaruse~severity\_flood+worst\_flood+pol\_party, data=sim.survey, family="binomial")

##   
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 1).  
## Chain 1:   
## Chain 1: Gradient evaluation took 0 seconds  
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.  
## Chain 1: Adjust your expectations accordingly!  
## Chain 1:   
## Chain 1:   
## Chain 1: Iteration: 1 / 2000 [ 0%] (Warmup)  
## Chain 1: Iteration: 200 / 2000 [ 10%] (Warmup)  
## Chain 1: Iteration: 400 / 2000 [ 20%] (Warmup)  
## Chain 1: Iteration: 600 / 2000 [ 30%] (Warmup)  
## Chain 1: Iteration: 800 / 2000 [ 40%] (Warmup)  
## Chain 1: Iteration: 1000 / 2000 [ 50%] (Warmup)  
## Chain 1: Iteration: 1001 / 2000 [ 50%] (Sampling)  
## Chain 1: Iteration: 1200 / 2000 [ 60%] (Sampling)  
## Chain 1: Iteration: 1400 / 2000 [ 70%] (Sampling)  
## Chain 1: Iteration: 1600 / 2000 [ 80%] (Sampling)  
## Chain 1: Iteration: 1800 / 2000 [ 90%] (Sampling)  
## Chain 1: Iteration: 2000 / 2000 [100%] (Sampling)  
## Chain 1:   
## Chain 1: Elapsed Time: 0.13 seconds (Warm-up)  
## Chain 1: 0.156 seconds (Sampling)  
## Chain 1: 0.286 seconds (Total)  
## Chain 1:   
##   
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 2).  
## Chain 2:   
## Chain 2: Gradient evaluation took 0 seconds  
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.  
## Chain 2: Adjust your expectations accordingly!  
## Chain 2:   
## Chain 2:   
## Chain 2: Iteration: 1 / 2000 [ 0%] (Warmup)  
## Chain 2: Iteration: 200 / 2000 [ 10%] (Warmup)  
## Chain 2: Iteration: 400 / 2000 [ 20%] (Warmup)  
## Chain 2: Iteration: 600 / 2000 [ 30%] (Warmup)  
## Chain 2: Iteration: 800 / 2000 [ 40%] (Warmup)  
## Chain 2: Iteration: 1000 / 2000 [ 50%] (Warmup)  
## Chain 2: Iteration: 1001 / 2000 [ 50%] (Sampling)  
## Chain 2: Iteration: 1200 / 2000 [ 60%] (Sampling)  
## Chain 2: Iteration: 1400 / 2000 [ 70%] (Sampling)  
## Chain 2: Iteration: 1600 / 2000 [ 80%] (Sampling)  
## Chain 2: Iteration: 1800 / 2000 [ 90%] (Sampling)  
## Chain 2: Iteration: 2000 / 2000 [100%] (Sampling)  
## Chain 2:   
## Chain 2: Elapsed Time: 0.127 seconds (Warm-up)  
## Chain 2: 0.131 seconds (Sampling)  
## Chain 2: 0.258 seconds (Total)  
## Chain 2:   
##   
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 3).  
## Chain 3:   
## Chain 3: Gradient evaluation took 0 seconds  
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.  
## Chain 3: Adjust your expectations accordingly!  
## Chain 3:   
## Chain 3:   
## Chain 3: Iteration: 1 / 2000 [ 0%] (Warmup)  
## Chain 3: Iteration: 200 / 2000 [ 10%] (Warmup)  
## Chain 3: Iteration: 400 / 2000 [ 20%] (Warmup)  
## Chain 3: Iteration: 600 / 2000 [ 30%] (Warmup)  
## Chain 3: Iteration: 800 / 2000 [ 40%] (Warmup)  
## Chain 3: Iteration: 1000 / 2000 [ 50%] (Warmup)  
## Chain 3: Iteration: 1001 / 2000 [ 50%] (Sampling)  
## Chain 3: Iteration: 1200 / 2000 [ 60%] (Sampling)  
## Chain 3: Iteration: 1400 / 2000 [ 70%] (Sampling)  
## Chain 3: Iteration: 1600 / 2000 [ 80%] (Sampling)  
## Chain 3: Iteration: 1800 / 2000 [ 90%] (Sampling)  
## Chain 3: Iteration: 2000 / 2000 [100%] (Sampling)  
## Chain 3:   
## Chain 3: Elapsed Time: 0.136 seconds (Warm-up)  
## Chain 3: 0.162 seconds (Sampling)  
## Chain 3: 0.298 seconds (Total)  
## Chain 3:   
##   
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 4).  
## Chain 4:   
## Chain 4: Gradient evaluation took 0 seconds  
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.  
## Chain 4: Adjust your expectations accordingly!  
## Chain 4:   
## Chain 4:   
## Chain 4: Iteration: 1 / 2000 [ 0%] (Warmup)  
## Chain 4: Iteration: 200 / 2000 [ 10%] (Warmup)  
## Chain 4: Iteration: 400 / 2000 [ 20%] (Warmup)  
## Chain 4: Iteration: 600 / 2000 [ 30%] (Warmup)  
## Chain 4: Iteration: 800 / 2000 [ 40%] (Warmup)  
## Chain 4: Iteration: 1000 / 2000 [ 50%] (Warmup)  
## Chain 4: Iteration: 1001 / 2000 [ 50%] (Sampling)  
## Chain 4: Iteration: 1200 / 2000 [ 60%] (Sampling)  
## Chain 4: Iteration: 1400 / 2000 [ 70%] (Sampling)  
## Chain 4: Iteration: 1600 / 2000 [ 80%] (Sampling)  
## Chain 4: Iteration: 1800 / 2000 [ 90%] (Sampling)  
## Chain 4: Iteration: 2000 / 2000 [100%] (Sampling)  
## Chain 4:   
## Chain 4: Elapsed Time: 0.131 seconds (Warm-up)  
## Chain 4: 0.13 seconds (Sampling)  
## Chain 4: 0.261 seconds (Total)  
## Chain 4:

sim.survey.mod2 <- stan\_glm(lidaruse~severity\_flood+ worst\_flood+ currentmap, data=sim.survey, family="binomial")

##   
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 1).  
## Chain 1:   
## Chain 1: Gradient evaluation took 0 seconds  
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.  
## Chain 1: Adjust your expectations accordingly!  
## Chain 1:   
## Chain 1:   
## Chain 1: Iteration: 1 / 2000 [ 0%] (Warmup)  
## Chain 1: Iteration: 200 / 2000 [ 10%] (Warmup)  
## Chain 1: Iteration: 400 / 2000 [ 20%] (Warmup)  
## Chain 1: Iteration: 600 / 2000 [ 30%] (Warmup)  
## Chain 1: Iteration: 800 / 2000 [ 40%] (Warmup)  
## Chain 1: Iteration: 1000 / 2000 [ 50%] (Warmup)  
## Chain 1: Iteration: 1001 / 2000 [ 50%] (Sampling)  
## Chain 1: Iteration: 1200 / 2000 [ 60%] (Sampling)  
## Chain 1: Iteration: 1400 / 2000 [ 70%] (Sampling)  
## Chain 1: Iteration: 1600 / 2000 [ 80%] (Sampling)  
## Chain 1: Iteration: 1800 / 2000 [ 90%] (Sampling)  
## Chain 1: Iteration: 2000 / 2000 [100%] (Sampling)  
## Chain 1:   
## Chain 1: Elapsed Time: 0.124 seconds (Warm-up)  
## Chain 1: 0.132 seconds (Sampling)  
## Chain 1: 0.256 seconds (Total)  
## Chain 1:   
##   
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 2).  
## Chain 2:   
## Chain 2: Gradient evaluation took 0 seconds  
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.  
## Chain 2: Adjust your expectations accordingly!  
## Chain 2:   
## Chain 2:   
## Chain 2: Iteration: 1 / 2000 [ 0%] (Warmup)  
## Chain 2: Iteration: 200 / 2000 [ 10%] (Warmup)  
## Chain 2: Iteration: 400 / 2000 [ 20%] (Warmup)  
## Chain 2: Iteration: 600 / 2000 [ 30%] (Warmup)  
## Chain 2: Iteration: 800 / 2000 [ 40%] (Warmup)  
## Chain 2: Iteration: 1000 / 2000 [ 50%] (Warmup)  
## Chain 2: Iteration: 1001 / 2000 [ 50%] (Sampling)  
## Chain 2: Iteration: 1200 / 2000 [ 60%] (Sampling)  
## Chain 2: Iteration: 1400 / 2000 [ 70%] (Sampling)  
## Chain 2: Iteration: 1600 / 2000 [ 80%] (Sampling)  
## Chain 2: Iteration: 1800 / 2000 [ 90%] (Sampling)  
## Chain 2: Iteration: 2000 / 2000 [100%] (Sampling)  
## Chain 2:   
## Chain 2: Elapsed Time: 0.146 seconds (Warm-up)  
## Chain 2: 0.141 seconds (Sampling)  
## Chain 2: 0.287 seconds (Total)  
## Chain 2:   
##   
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 3).  
## Chain 3:   
## Chain 3: Gradient evaluation took 0 seconds  
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.  
## Chain 3: Adjust your expectations accordingly!  
## Chain 3:   
## Chain 3:   
## Chain 3: Iteration: 1 / 2000 [ 0%] (Warmup)  
## Chain 3: Iteration: 200 / 2000 [ 10%] (Warmup)  
## Chain 3: Iteration: 400 / 2000 [ 20%] (Warmup)  
## Chain 3: Iteration: 600 / 2000 [ 30%] (Warmup)  
## Chain 3: Iteration: 800 / 2000 [ 40%] (Warmup)  
## Chain 3: Iteration: 1000 / 2000 [ 50%] (Warmup)  
## Chain 3: Iteration: 1001 / 2000 [ 50%] (Sampling)  
## Chain 3: Iteration: 1200 / 2000 [ 60%] (Sampling)  
## Chain 3: Iteration: 1400 / 2000 [ 70%] (Sampling)  
## Chain 3: Iteration: 1600 / 2000 [ 80%] (Sampling)  
## Chain 3: Iteration: 1800 / 2000 [ 90%] (Sampling)  
## Chain 3: Iteration: 2000 / 2000 [100%] (Sampling)  
## Chain 3:   
## Chain 3: Elapsed Time: 0.134 seconds (Warm-up)  
## Chain 3: 0.146 seconds (Sampling)  
## Chain 3: 0.28 seconds (Total)  
## Chain 3:   
##   
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 4).  
## Chain 4:   
## Chain 4: Gradient evaluation took 0 seconds  
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.  
## Chain 4: Adjust your expectations accordingly!  
## Chain 4:   
## Chain 4:   
## Chain 4: Iteration: 1 / 2000 [ 0%] (Warmup)  
## Chain 4: Iteration: 200 / 2000 [ 10%] (Warmup)  
## Chain 4: Iteration: 400 / 2000 [ 20%] (Warmup)  
## Chain 4: Iteration: 600 / 2000 [ 30%] (Warmup)  
## Chain 4: Iteration: 800 / 2000 [ 40%] (Warmup)  
## Chain 4: Iteration: 1000 / 2000 [ 50%] (Warmup)  
## Chain 4: Iteration: 1001 / 2000 [ 50%] (Sampling)  
## Chain 4: Iteration: 1200 / 2000 [ 60%] (Sampling)  
## Chain 4: Iteration: 1400 / 2000 [ 70%] (Sampling)  
## Chain 4: Iteration: 1600 / 2000 [ 80%] (Sampling)  
## Chain 4: Iteration: 1800 / 2000 [ 90%] (Sampling)  
## Chain 4: Iteration: 2000 / 2000 [100%] (Sampling)  
## Chain 4:   
## Chain 4: Elapsed Time: 0.13 seconds (Warm-up)  
## Chain 4: 0.138 seconds (Sampling)  
## Chain 4: 0.268 seconds (Total)  
## Chain 4:

# look at the summary of model outputs  
summary(sim.survey.fullmod)

##   
## Model Info:  
## function: stan\_glm  
## family: binomial [logit]  
## formula: lidaruse ~ severity\_flood + worst\_flood + currentmap + incr\_no\_flood +   
## prepared + gender + pol\_party + science\_trust + gov\_trust  
## algorithm: sampling  
## sample: 4000 (posterior sample size)  
## priors: see help('prior\_summary')  
## observations: 200  
## predictors: 10  
##   
## Estimates:  
## mean sd 10% 50% 90%  
## (Intercept) 2.1 1.5 0.2 2.0 3.9   
## severity\_flood 0.9 0.2 0.6 0.8 1.1   
## worst\_flood 0.0 0.1 -0.1 0.0 0.2   
## currentmap -0.4 0.4 -1.0 -0.4 0.1   
## incr\_no\_flood -0.7 0.3 -1.0 -0.7 -0.3   
## prepared 0.1 0.2 -0.1 0.1 0.3   
## gender -0.1 0.2 -0.4 -0.1 0.2   
## pol\_party -0.1 0.2 -0.4 -0.1 0.1   
## science\_trust -0.2 0.1 -0.4 -0.2 0.0   
## gov\_trust 0.0 0.1 -0.2 0.0 0.2   
##   
## Fit Diagnostics:  
## mean sd 10% 50% 90%  
## mean\_PPD 0.8 0.0 0.7 0.8 0.8   
##   
## The mean\_ppd is the sample average posterior predictive distribution of the outcome variable (for details see help('summary.stanreg')).  
##   
## MCMC diagnostics  
## mcse Rhat n\_eff  
## (Intercept) 0.0 1.0 4652   
## severity\_flood 0.0 1.0 3340   
## worst\_flood 0.0 1.0 4268   
## currentmap 0.0 1.0 3796   
## incr\_no\_flood 0.0 1.0 4004   
## prepared 0.0 1.0 4257   
## gender 0.0 1.0 4303   
## pol\_party 0.0 1.0 4245   
## science\_trust 0.0 1.0 4930   
## gov\_trust 0.0 1.0 3631   
## mean\_PPD 0.0 1.0 4497   
## log-posterior 0.1 1.0 1829   
##   
## For each parameter, mcse is Monte Carlo standard error, n\_eff is a crude measure of effective sample size, and Rhat is the potential scale reduction factor on split chains (at convergence Rhat=1).

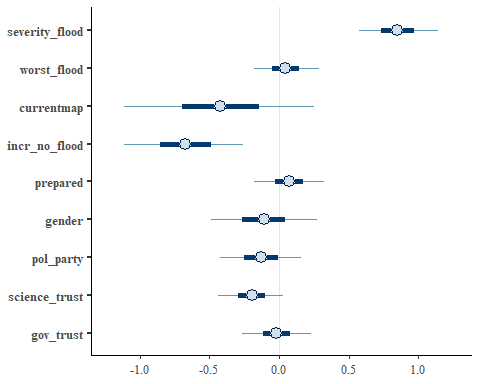
summary(sim.survey.mod1)

##   
## Model Info:  
## function: stan\_glm  
## family: binomial [logit]  
## formula: lidaruse ~ severity\_flood + worst\_flood + pol\_party  
## algorithm: sampling  
## sample: 4000 (posterior sample size)  
## priors: see help('prior\_summary')  
## observations: 200  
## predictors: 4  
##   
## Estimates:  
## mean sd 10% 50% 90%  
## (Intercept) -0.6 0.7 -1.5 -0.6 0.4   
## severity\_flood 0.8 0.2 0.6 0.8 1.0   
## worst\_flood 0.0 0.1 -0.1 0.0 0.2   
## pol\_party -0.1 0.2 -0.3 -0.1 0.1   
##   
## Fit Diagnostics:  
## mean sd 10% 50% 90%  
## mean\_PPD 0.8 0.0 0.7 0.8 0.8   
##   
## The mean\_ppd is the sample average posterior predictive distribution of the outcome variable (for details see help('summary.stanreg')).  
##   
## MCMC diagnostics  
## mcse Rhat n\_eff  
## (Intercept) 0.0 1.0 3802   
## severity\_flood 0.0 1.0 3038   
## worst\_flood 0.0 1.0 4323   
## pol\_party 0.0 1.0 4268   
## mean\_PPD 0.0 1.0 3780   
## log-posterior 0.0 1.0 1922   
##   
## For each parameter, mcse is Monte Carlo standard error, n\_eff is a crude measure of effective sample size, and Rhat is the potential scale reduction factor on split chains (at convergence Rhat=1).

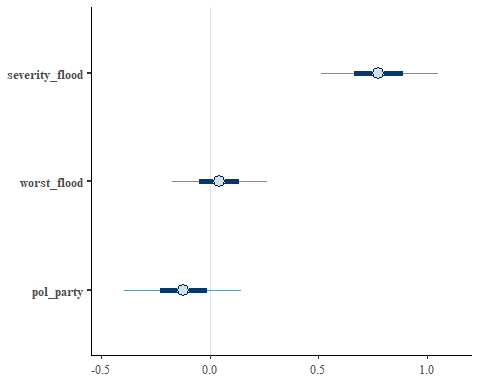
summary(sim.survey.mod2)

##   
## Model Info:  
## function: stan\_glm  
## family: binomial [logit]  
## formula: lidaruse ~ severity\_flood + worst\_flood + currentmap  
## algorithm: sampling  
## sample: 4000 (posterior sample size)  
## priors: see help('prior\_summary')  
## observations: 200  
## predictors: 4  
##   
## Estimates:  
## mean sd 10% 50% 90%  
## (Intercept) -0.4 0.8 -1.5 -0.4 0.6   
## severity\_flood 0.8 0.2 0.6 0.8 1.0   
## worst\_flood 0.0 0.1 -0.1 0.0 0.2   
## currentmap -0.3 0.4 -0.8 -0.3 0.2   
##   
## Fit Diagnostics:  
## mean sd 10% 50% 90%  
## mean\_PPD 0.8 0.0 0.7 0.8 0.8   
##   
## The mean\_ppd is the sample average posterior predictive distribution of the outcome variable (for details see help('summary.stanreg')).  
##   
## MCMC diagnostics  
## mcse Rhat n\_eff  
## (Intercept) 0.0 1.0 4327   
## severity\_flood 0.0 1.0 3525   
## worst\_flood 0.0 1.0 3986   
## currentmap 0.0 1.0 4215   
## mean\_PPD 0.0 1.0 4436   
## log-posterior 0.0 1.0 1582   
##   
## For each parameter, mcse is Monte Carlo standard error, n\_eff is a crude measure of effective sample size, and Rhat is the potential scale reduction factor on split chains (at convergence Rhat=1).

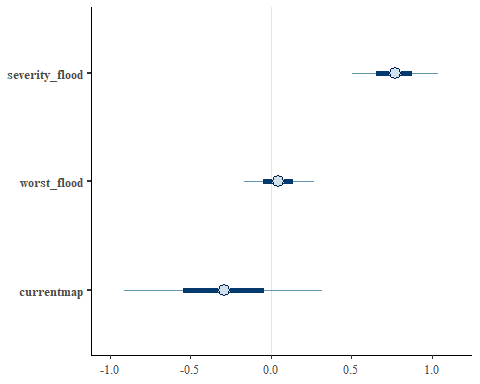
# compare paramter estiamtes  
plot(sim.survey.fullmod, pars="beta")



plot(sim.survey.mod1, pars="beta")



plot(sim.survey.mod2, pars="beta")



# compare the information criteria (loo for bayesian) to see which model loses the least amount of "realness"  
loo\_compare(loo(sim.survey.fullmod), loo(sim.survey.mod1), loo(sim.survey.mod2)) # loo compares all you to easily view the values relative to one another

## elpd\_diff se\_diff  
## sim.survey.mod2 0.0 0.0   
## sim.survey.mod1 -0.1 1.1   
## sim.survey.fullmod -2.3 3.4

# RMSE function:  
rmse <- function(y, ypred) {  
 rmse = sqrt(mean((y - ypred)^2))  
 return(rmse)  
}  
  
#MAE function:  
mae <- function(y, ypred) {  
 mae = (mean(abs(y - ypred)))  
 return(mae)  
}  
  
# Identify model's predicted yhat:  
yhat.full <- posterior\_predict(sim.survey.fullmod)   
yhat.full <- apply(yhat.full, 2, median)   
yhat.1 <- posterior\_predict(sim.survey.mod1)   
yhat.1 <- apply(yhat.1, 2, median)   
yhat.2 <- posterior\_predict(sim.survey.mod2)   
yhat.2 <- apply(yhat.2, 2, median)   
## Find the residual mean squared error and Mean Absolute Error (MAE) for the model's fit (in sample prediction)? I am running for all three models to see how it changes  
rmse(sim.survey$lidaruse, yhat.full)

## [1] 0.4636809

mae(sim.survey$lidaruse, yhat.full)

## [1] 0.215

rmse(sim.survey$lidaruse, yhat.1)

## [1] 0.4582576

mae(sim.survey$lidaruse, yhat.1)

## [1] 0.21

rmse(sim.survey$lidaruse, yhat.2)

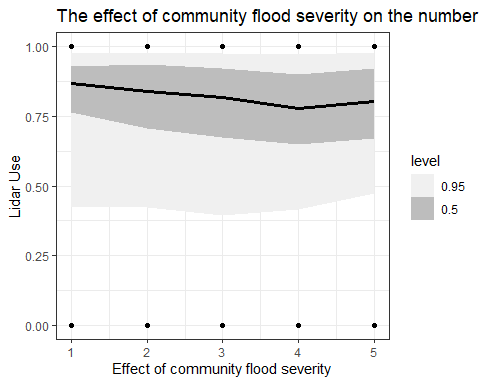
## [1] 0.4527693

mae(sim.survey$lidaruse, yhat.2)

## [1] 0.205

*Make counterfactual plots to determine the effect of each predictor in Model 1*

# we want make a graph that shows the effect of severity\_flood on lidar use while holding worst\_flood and pol\_party at their minimums  
## Make a sequence of flood severity  
sev.flood.gradient <- round(rep(seq(min(sim.survey$severity\_flood),  
 max(sim.survey$severity\_flood),length.out=200),1))   
  
worst.flood.min <- min(sim.survey$worst\_flood) # min flood value  
pol.party.min <- min(sim.survey$pol\_party) # min political affiliation  
  
preds.sev <- add\_fitted\_draws(sim.survey.mod1,   
 newdata=data.frame(sev.flood.g=sev.flood.gradient,  
 worst.flood.m=worst.flood.min, # this is the min value of worst flood   
 pol.party.m=pol.party.min ), # this is the min value of political party affiliation   
 re\_formula=NA,  
 draws = 200, type="response")  
  
#This line loads the original data (actual collected points)  
ggplot(preds.sev, aes(x=sev.flood.g,   
 y=.value)) +   
 stat\_lineribbon(.width = c(0.5, 0.95)) +   
 scale\_fill\_brewer(palette = "Greys") +   
 labs(y="Lidar Use", x = "Effect of community flood severity") +  
 geom\_point(data=sim.survey, aes(x=severity\_flood,y=lidaruse)) +  
 ggtitle("The effect of community flood severity on the number of LiDAR users") +  
 theme\_bw()



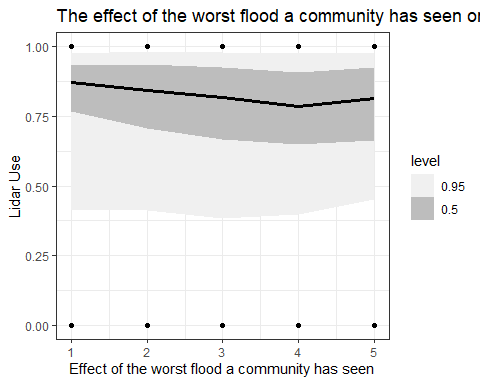
mean(preds.sev$.value)

## [1] 0.7842787

mean(sim.survey$severity\_flood)

## [1] 3.05

# we want make a graph that shows the effect of worst\_flood on lidar use while holding sev\_flood and pol\_party at their minimums  
  
worst.flood.gradient <- round(rep(seq(min(sim.survey$worst\_flood),  
 max(sim.survey$worst\_flood),length.out=200),1))   
  
sev.flood.min <- min(sim.survey$severity\_flood)  
pol.party.min <- min(sim.survey$pol\_party)  
  
preds.worst <- add\_fitted\_draws(sim.survey.mod1,   
 newdata=data.frame(worst.flood.g=worst.flood.gradient,  
 sev.flood.m=sev.flood.min,   
 pol.party.m=pol.party.min ),   
 re\_formula=NA,  
 draws = 200, type="response")  
  
#This line loads the original data (actual collected points)  
ggplot(preds.worst, aes(x=worst.flood.g,   
 y=.value)) +   
 stat\_lineribbon(.width = c(0.5, 0.95)) +   
 scale\_fill\_brewer(palette = "Greys") +   
 labs(y="Lidar Use", x = "Effect of the worst flood a community has seen") +  
 geom\_point(data=sim.survey, aes(x=worst\_flood,y=lidaruse)) +  
 ggtitle("The effect of the worst flood a community has seen on the number of LiDAR users") +  
 theme\_bw()



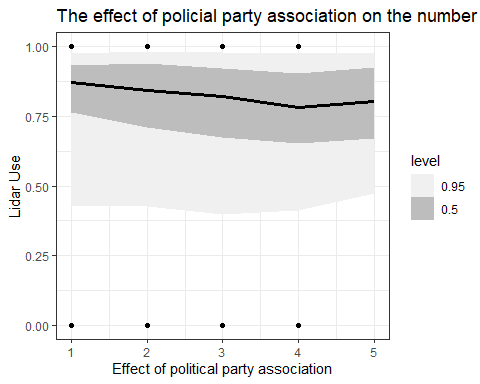
mean(preds.worst$.value)

## [1] 0.7863068

mean(sim.survey$worst\_flood)

## [1] 2.92

# we want make a graph that shows the effect of pol\_party on lidar use while holding worst\_flood and sev\_flood at their minimums  
## Make a sequence of flood severity  
pol.party.gradient <- round(rep(seq(min(sim.survey$pol\_party),  
 max(sim.survey$pol\_party),length.out=200),1))   
  
preds.pol <- add\_fitted\_draws(sim.survey.mod1,   
 newdata=data.frame(pol.party.g=worst.flood.gradient,  
 sev.flood.m=sev.flood.min,   
 worst.flood.m=worst.flood.min ),   
 re\_formula=NA,  
 draws = 200, type="response")  
  
#This line loads the original data (actual collected points)  
ggplot(preds.pol, aes(x=pol.party.g,   
 y=.value)) +   
 stat\_lineribbon(.width = c(0.5, 0.95)) +   
 scale\_fill\_brewer(palette = "Greys") +   
 labs(y="Lidar Use", x = "Effect of political party association") +  
 geom\_point(data=sim.survey, aes(x=pol\_party,y=lidaruse)) +  
 ggtitle("The effect of policial party association on the number of LiDAR users") +  
 theme\_bw()



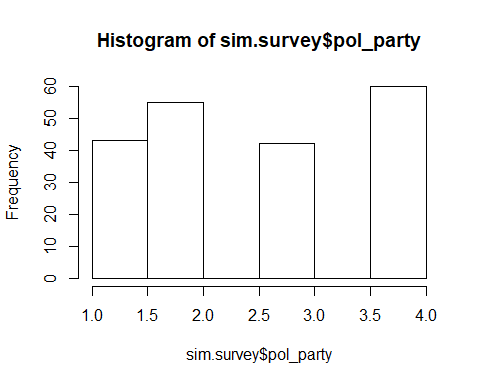
mean(preds.pol$.value)

## [1] 0.7870983

mean(sim.survey$pol\_party)

## [1] 2.595

hist(sim.survey$pol\_party) # to help clarify political party association



# References

Birkholz, S., M. Muro, P. Jeffrey, and H. M. Smith. 2014. “Rethinking the Relationship Between Flood Risk Perception and Flood Management.” *Science of the Total Environment* 478 (April): 12–20. <https://doi.org/10.1016/j.scitotenv.2014.01.061>.

Chapin, F. Stuart, Stephen R. Carpenter, Gary P. Kofinas, Carl Folke, Nick Abel, William C. Clark, Per Olsson, et al. 2010. “Ecosystem Stewardship: Sustainability Strategies for a Rapidly Changing Planet.” *Trends in Ecology & Evolution* 25 (4): 241–49. <https://doi.org/10.1016/j.tree.2009.10.008>.

Kellens, Wim, Teun Terpstra, and Philippe De Maeyer. 2013. “Perception and Communication of Flood Risks: A Systematic Review of Empirical Research: **Perception and Communication of Flood Risks**.” *Risk Analysis* 33 (1): 24–49. <https://doi.org/10.1111/j.1539-6924.2012.01844.x>.

Lujala, Päivi, Haakon Lein, and Jan Ketil Rød. 2015. “Climate Change, Natural Hazards, and Risk Perception: The Role of Proximity and Personal Experience.” *Local Environment* 20 (4): 489–509. <https://doi.org/10.1080/13549839.2014.887666>.

Luo, Xin, Han Li, Jie Zhang, and J. P. Shim. 2010. “Examining Multi-Dimensional Trust and Multi-Faceted Risk in Initial Acceptance of Emerging Technologies: An Empirical Study of Mobile Banking Services.” *Decision Support Systems* 49 (2): 222–34. <https://doi.org/10.1016/j.dss.2010.02.008>.

O’Connor, Robert E., Richard J. Bard, and Ann Fisher. 1999. “Risk Perceptions, General Environmental Beliefs, and Willingness to Address Climate Change.” *Risk Analysis* 19 (3): 461–71. <https://doi.org/10.1111/j.1539-6924.1999.tb00421.x>.

Savage, Ian. 1993. “Demographic Influences on Risk Perceptions.” *Risk Analysis* 13 (4): 413–20. <https://doi.org/10.1111/j.1539-6924.1993.tb00741.x>.

Viglione, Alberto, Giuliano Di Baldassarre, Luigia Brandimarte, Linda Kuil, Gemma Carr, José Luis Salinas, Anna Scolobig, and Günter Blöschl. 2014. “Insights from Socio-Hydrology Modelling on Dealing with Flood Risk – Roles of Collective Memory, Risk-Taking Attitude and Trust.” *Journal of Hydrology*, Creating Partnerships Between Hydrology and Social Science: A Priority for Progress, 518 (October): 71–82. <https://doi.org/10.1016/j.jhydrol.2014.01.018>.

Vinh Hung, Hoang, Rajib Shaw, and Masami Kobayashi. 2007. “Flood Risk Management for the RUA of Hanoi: Importance of Community Perception of Catastrophic Flood Risk in Disaster Risk Planning.” *Disaster Prevention and Management: An International Journal* 16 (2): 245–58. <https://doi.org/10.1108/09653560710739568>.

Wachinger, Gisela, Ortwin Renn, Chloe Begg, and Christian Kuhlicke. 2013. “The Risk Perception Paradox-Implications for Governance and Communication of Natural Hazards: The Risk Perception Paradox.” *Risk Analysis* 33 (6): 1049–65. <https://doi.org/10.1111/j.1539-6924.2012.01942.x>.