MATH 640 Final Project

Jason Michaels (jam521), Niko Paulson (ndp32), Miranda Seitz-McLeese (mgs85)

1 Introduction

This analysis will be done on a data set of a variety of measurements about severe weather in the United States. We chose to use a subset of that data covering the years 1996 through 2016. The data coms from the NOAA website [1]. For this project we focused on the deaths directly attributable to a type of weather event. In the interest of time, we limited our models to two types of severe weather, Tornados and Flash Floods, reponsible for the second and third most direct deaths among weather types. Understanding how and at what rate severe weather events become lethal in the United States has tremendous public health impacts. In this paper we compare four possible models for the deaths: The traditional Poisson and negative binomial distributions, as well as the zero inflated variant of each.

The remainder of this analysis is organized as follows: Section 2 discusses and derives the models. Section 3 describes the results of the analysis. And Section 4 contains the conclusions.

2 Methods

The deaths attributed to a severe weather event is 'count' data. The most common model used for count data is the Poisson distribution. However for some weather events, the negative binomial model is a better fit, because the Poisson distribution assumes that the events being counted occur independently.

Fortunately, the vast majority of severe weather events in the United States involve no deaths, therefore we wanted to also account for the possibility of structural zeros, therefore we also considered zero inflated variants. These distributions are created by returning 0 with probability σ and sampling from the original distribution with probability $(1-\sigma)$. We will derive and fit a model for each of the four distributions and see if there is a difference in our results and evaluate to determine which model best fits the data.

2.1 Poisson

The Poisson model has one parameter, λ represents the expected number of occurances of the event of interest. For a single random variable x, the probability density is:

$$p(x|\lambda) = \frac{\lambda^x e^{-\lambda}}{x!}.$$

Thus the likelihood for a sum of Poisson random variables can be written as follows:

$$\mathcal{L}(X|\lambda) \propto \lambda^{n\bar{X}} e^{-n\lambda}.$$

Since we want the data to speak for itself, we will use a non-informative random variable, namely, Jeffreys' prior. For the Poisson distribtion this is given as follows:

$$\pi(\lambda) \propto \lambda^{1/2-1} e^{-0\cdot\lambda}.$$

We recognize this as the kernel of an improper gamma distribution. Combining the likelihood and the prior distribtion yields the following posterior distribution:

$$p(\lambda|X) = \lambda^{n\bar{X}+1/2-1} e^{-n\lambda}.$$

We recognize this as the kernel of a gamma distribution, namely

$$\lambda|X \sim \mathcal{G}amma(n\bar{X}+1/2, n).$$

2.2 Negative Binomial

The Negative Binomial model has two parameters, r, p represents the expected number of occurances of the event of interest. For a single random variable x, the probability density is:

$$p(X|r,p) = \frac{\Gamma(r+x)}{\Gamma(r)x!} p^x (1-p)^r.$$

Thus the likelihood for a sum of Negative Binomial random variables can be written as follows:

$$\mathcal{L}(X|r,p) = \left[\prod_{i=1}^{n} \frac{\Gamma(r+x_i)}{\Gamma(r)x_i!}\right] p^{n\bar{X}}(1-p)^{nr}.$$

Since we want the data to speak for itself, we will use a non-informative random variable, namely, Jeffreys' prior. For the Poisson distribtion this is given as follows:

$$\pi(r,p) = r^{1/2}p^{-1}(1-2)^{-1/2}.$$

Combining the likelihood and the prior distribtion yields the following posterior distribution:

$$p(r,p|X) = \left\{ \left[\prod_{i=1}^{n} \frac{\Gamma(r+x_i)}{\Gamma(r)x_i!}\right] p^{n\bar{X}}(1-p)^{nr} \right\} \left\{ r^{1/2}p^{-1}(1-2)^{-1/2} \right\}$$

From this posterior we obtain the full conditionals. First consider $p|r,X$:

$$p(p|r,X) \propto p^{n\bar{x}-1}(1-p)^{nr+1/2-1}$$

We recognize this as the kernel of a beta distribution, namely

$$p|r,X \sim \mathcal{B}eta(n\bar{X}, nr+1/2).$$

Next consider $r|p,X$:

$$p(r|p,X) \propto \left[\prod_{i=1}^{n} \Gamma(r+x_i)\right] \Gamma(r)^{-n}(1-p)^{nr}r^{1/2}$$

This is not a recognized distribtion. So if we wish to make inferences on r we must use a Metropolis algorithm to sample from it.

2.3 Zero Inflated Poisson

The Zero Inflated Poisson (ZIP) model has two parameters. The parameter p is the probability of a structural zero, and λ corresponds to the parameter in a typical Poisson model. For a single observation x, the probability density is:

$$p(x|p,\lambda) = pI_{x=0}(x) + (1-p)\frac{e^{-\lambda}\lambda^{x}}{x!}$$

We can write the likelihood as follows:

$$L(p,\lambda|X) = \prod_{x_i=0} \left[p + (1-p)\frac{e^{-\lambda}\lambda^{x_i}}{x_i!}\right] \prod_{x_i\neq 0} \left[(1-p)\frac{e^{-\lambda}\lambda^{x_i}}{x_i!}\right]$$

Bayarri, Berger, and Datta (2008) suggest using the prior distribution $\pi(\lambda,p) \propto \frac{1}{\sqrt{\lambda}}I(0 < p < 1)$. This gives us the following posterior

$$\prod_{x_i=0} \left[p + (1-p)\frac{e^{-\lambda}\lambda^{x_i}}{x_i!}\right] \prod_{x_i\neq 0} \left[(1-p)\frac{e^{-\lambda}\lambda^{x_i-1/2}}{x_i!}\right]$$

Neither of the full conditional distributions is recognizable (see B.1). We can use a Metropolis-Hastings algorithm to sample from both of them. We will use a beta distribution as a proposal for p, and a gamma for λ. We will tune them to obtain a better acceptance rate.

Table 1: Posterior distributions of λ, for both event types. The second column relays the median of the sample for λ, with the 95 percent credible interval in parentheses.

Event Type	λ
Tornado	0.0592 (0.0564, 0.0620)
Flash Flood	0.0179 (0.0170, 0.0189)

Table 2: Posterior distributions of p, for both event types. The second column relays the median of the sample for p, with the 95 percent credible interval in parentheses.

Event Type	p
Tornado	0.0559 (0.0534, 0.0585)
Flash Flood	0.0176 (0.0167, 0.0186)

2.4 Zero Inflated Negative Binomial

The Zero Inflated Negative Binomial (ZINB) model has three parameters σ, the probability of a structural zero, and p, r the usual negative binomial parameters. For a single X the probability density is:

$$p(X|\sigma, p, r) = \sigma I_{X=0}(X) + (1-\sigma)\frac{\Gamma(r+X)}{\Gamma(r)X!}p^X(1-p)^r.$$

We take the uniform priors for σ and p as well as the non-informative gamma for r which is $r^{-1/2}$. For a full derivation, see B.2. My posterior is:

$$p(r, \sigma, p|X) \propto (\sigma + (1-\sigma)(1-p)^r)^Z (1-\sigma)^{N-Z}(1-p)^{(N-Z)r}p^{\sum_{i=1}^N X_i}r^{-1/2}\prod_{i=1}^N\left(\frac{\Gamma(r+X_i)}{\Gamma(r)}\right)$$

This distribution does not factor nicely, so I will use the Metropolis algorithm to sample from it. Because this posterior does not suggest any obvious proposal distributions I will sample each independently from a normal distribution centered at θ^*, and with a variance that is tuned to yield an appropriate acceptance rate.

3 Results

3.1 Poisson

Because the posterior from our Poisson likelihood and noninformative prior had a recognizable kernel, namely $\mathcal{G}amma(n\bar{X} + 1/2, n)$, we were able to sample from this distribution directly. For both Tornados, and Flash Floods, 10,000 samples were taken, to simulate the variable of interest, λ. The results are summarized in Table 1, while density plots can be found in Figure 1.

3.2 Negative Binomial

As in the case of the Poisson distribution, the conditional posterior from our Negative Binomial likelihood and noninformative prior had a recognizable kernel for p, namely $\mathcal{B}eta(n\bar{X}, nr + 1/2)$, thus we were able to sample from this distribution directly. Meanwhile we took r as known. For both Tornados, and Flash Floods, 10,000 samples were taken, to simulate the variable of interest, p. The results are summarized in Table 2, while density plots can be found in Figure 2.

3.3 Zero Inflated Poisson

In tuning the parameters of the two models, slightly different values were chosen for the two different event types. The proposal distribution selected for λ was a gamma(2, 2) for tornados, and a gamma(1, 2) for flash floods. The proposal for p within the tornado model was a beta(1940, 60), whereas it was a beta(2945, 55) in the flash flood model.

For each variable of interest, 20,000 samples were taken. As convergence was not immediately achieved, the first 10,000 samples were discarded as a burn-in. The results are summarized in Table 3. Convergance plots can be found in Figure 3, and density plots can be found in Figure 4.

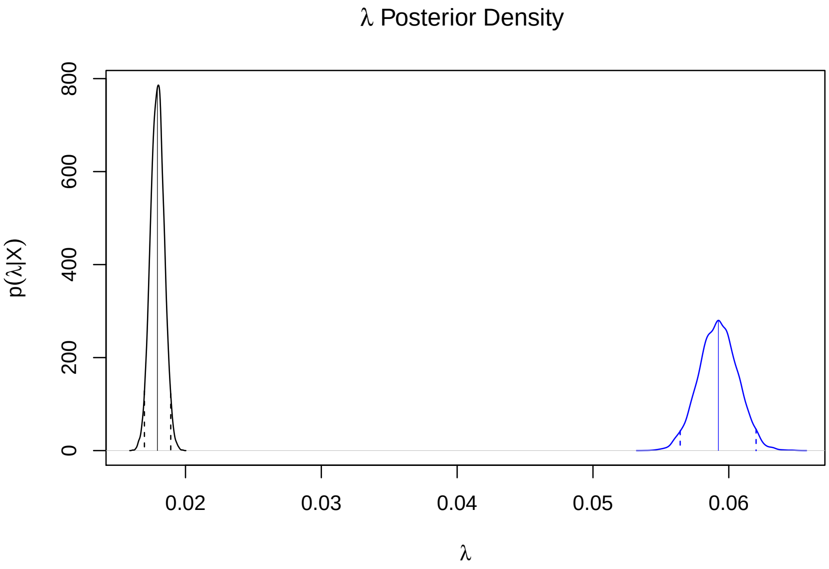

Figure 1: Posterior density plots (Poisson model) for flood parameters (in blue), and tornado parameters (black). The solid line represents the median and the dashed lines indicate the 95% confidence interval.

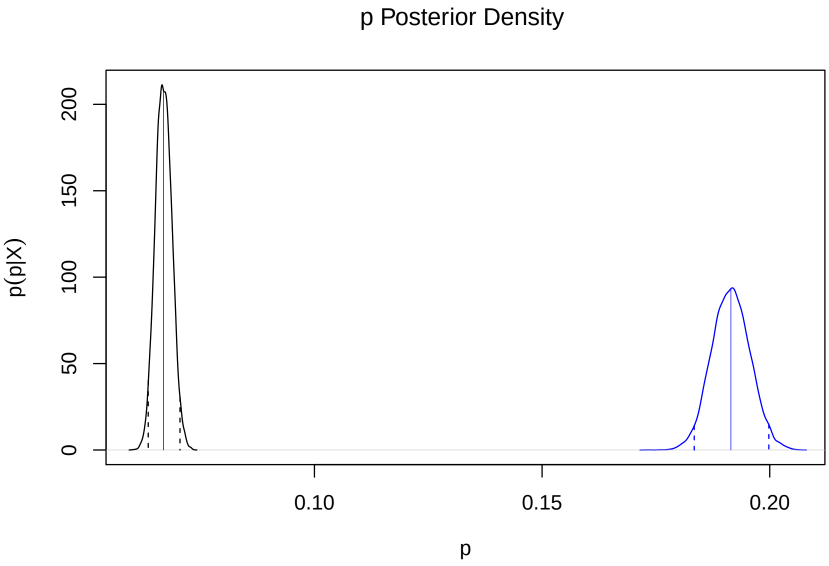

Figure 2: Posterior density plots (Negative Binomial model) for flood parameters (in blue), and tornado parameters (black). The solid line represents the median and the dashed lines indicate the 95% confidence interval.

Table 3: The posterior distributions of λ and p, for both event types. The second column relays the mean of the sample for λ, with the 95 percent credible interval in parentheses. The third does the same for p.

Event Type	λ	p
Tornado	1.872 (1.030, 3.710)	0.971 (0.968, 0.973)
Flash Flood	0.562 (0.293, 0.972)	0.982 (0.981, 0.983)

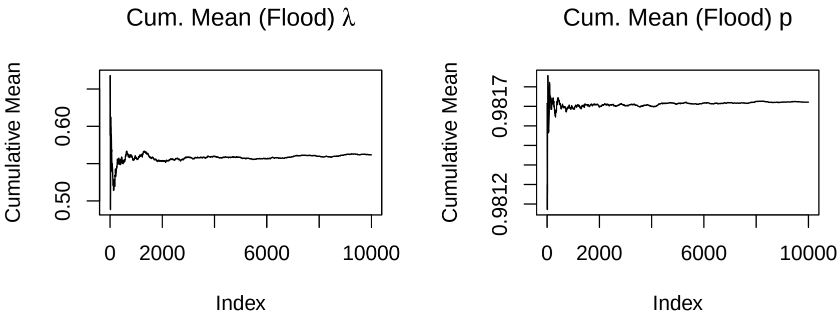

Figure 3: Cumulative mean plots for the posterior samples (ZIP model) of the two parameters from the flood data. Plots for the tornado data are not pictured, but behave similarly.

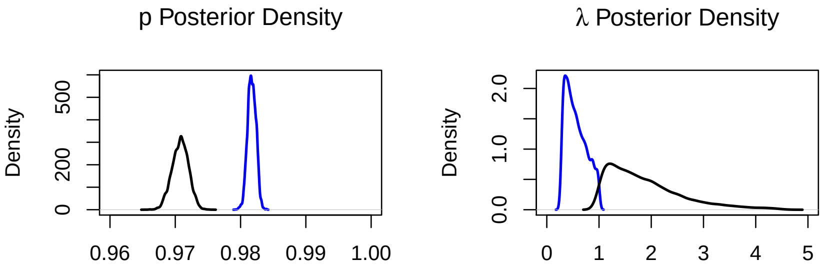

Figure 4: Posterior density plots (ZIP model) for flood parameters (in blue), and tornado parameters (black). The solid line represents the mean and the dashed lines indicate the 95% confidence interval.

Table 4: The following table shows the results of the flash flood ZINB model and the tornado ZINB model. It shows the mean and the median for each parameter as well as the 95% credible interval.

	Flash Flood			Tornado		
Parameter	Mean Value	Median Value	(95% CI)	Mean Value	Median Value	(95% CI)
σ	0.307	0.282	(0.013, 0.716)	0.299	0.276	(0.012, 0.702)
p	0.554	0.555	(0.518, 0.589)	0.862	0.863	(0.841, 0.882)
r	0.024	0.02	(0.014, 0.052)	0.015	0.013	(0.009, 0.033)

Table 5: The following table shows the resulting DIC for each of the four models on both the tornado data and the flash flood data.

Model	DIC Flash Flood	DIC Tornado
Poisson	14237	17606
Negative Binomial	101548	26372
ZIP	10454	8727
ZINB	10804.104	7166.309

3.4 Zero Inflated Negative Binomial

The ZINB model was fit using the Metropolis-Hastings algorithm with a multivariate truncated normal distribution used as the proposal distribution. For the source code of the sampler, please see A.1.

This model suffered severely from the curse of dimensionality. It was very slow to converge, and the very small variance needed for a reasonable acceptance led to high autocorrelation, meaning agressive thinning was necessary. Running mean plots the Flash Flood models can be found in Figure 5.

In the end 1030000 samples were drawn for each model, with a burn in of 30000 and one of every 100 samples was retained, resulting in a final sample size of 10000. The flood model had an acceptance rate of 0.195 and the tornado model had an acceptance rate of 0.195. The resulting densities are pictured in Figure 6, and Table 4, though due to questions of convergence the results should viewed with caution.

The DIC for floods and tornadoes respectively were 10804 and 7166.

4 Discussion

The goal of this analysis was to determine which of the four models was the best fit for the data. In order to make this determination the DIC was calculated for each model on both tornados and flash floods. The results are summarized in Table 5.

In general, models with smaller DIC are preferred to models with larger DIC. From the table we can see that the Zero Inflated Poisson model has the lowest DIC, both for Flash Floods and Tornados. Thus for estimating the rate of leathality of Flash Floods, Tornados, and possibly other severe weather, we should use a Zero Inflated Poisson model to fit the data.

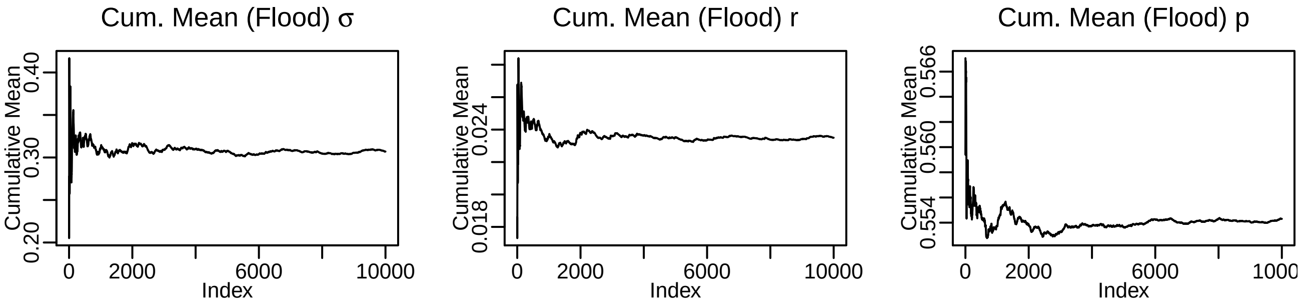

Figure 5: Cumulative mean plots for each of the three parameters in the ZINB model fit on flash flood data. The tornado plots are not pictured, but are similar in character.

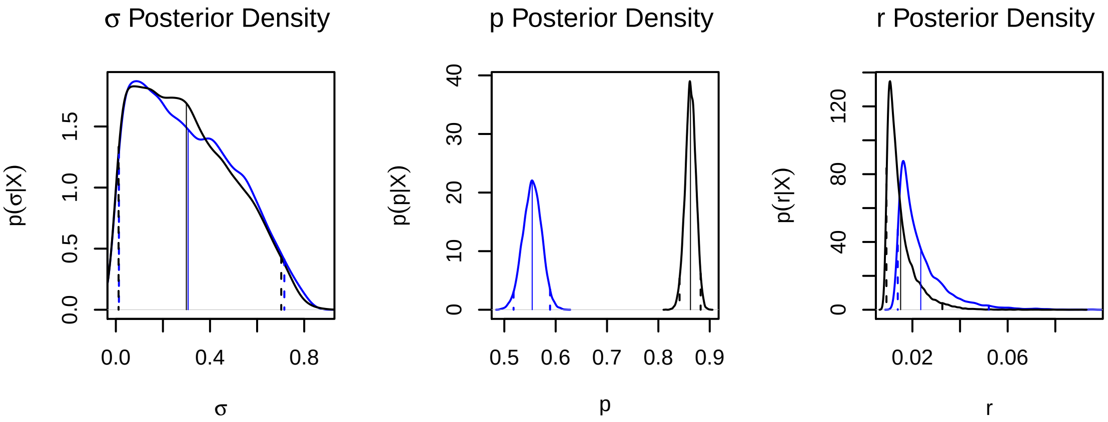

Figure 6: Posterior density curves for each of the three variables. The flood posterior is shown in blue and the tornado posterior is shown in black.

References

[1] NOAA's Severe Weather Data Inventory, https://www1.ncdc.noaa.gov/pub/data/swdi/stormevents/csvfiles/. Accessed April 2017.

[2] Bayarri, M., Berger, J., Datta, G. (2008). Objective testing of Poisson versus inflated Poisson models. IMS Collections, 3, 105-121.

A Code

This appendix includes the code used to implement the models.

A.1 Zero Inflated Negative Binomial code

```
logpost <- function(sigma, p, r, Z, N, S, nonzero){
  lp <- Z*log(1 + (1/sigma-1)*(1-p)^r)
  lp <- lp + Z*log(sigma)
  lp <- lp + (N-Z)*log(1-sigma)
  lp <- lp + (N-Z)*r*log(1-p)
  lp <- lp + S*log(p)
  lp <- lp - log(r)/2
  lp <- lp - (N-Z)*lgamma(r)
  lp <- lp + sum(lgamma(r+nonzero))
  return(lp)
}

loglik <- function(sigma, p, r, Z, N, S, nonzero){
  lp <- Z*log(1 + (1/sigma-1)*(1-p)^r)
  lp <- lp + Z*log(sigma)
  lp <- lp + (N-Z)*log(1-sigma)
  lp <- lp + (N-Z)*r*log(1-p)
  lp <- lp + S*log(p)
  lp <- lp - sum(lgamma(nonzero+1))
  lp <- lp - (N-Z)*lgamma(r)
  lp <- lp + sum(lgamma(r+nonzero))
  return(lp)
}

g <-function(sigma, p, r,
             sigma.mean, p.mean, r.mean,
             sigma.var, p.var, r.var){
    return(dtruncnorm(sigma, 0, 1, sigma.mean, sqrt(sigma.var))*
           dtruncnorm(p, 0, 1, p.mean, sqrt(p.var))*
           dtruncnorm(r, 0, Inf, r.mean, sqrt(r.var)))
}


zinb.sampler <- function(df, event_type, chainlen,
                         r.var=1, p.var=1, sigma.var=1,
                         burnin=0, thinning=1){
  X <- df[df$EVENT_TYPE==event_type,"DEATHS_DIRECT"]
  N <- length(X)
  Z <- sum(X==0)
  S <- sum(X)
  nonzero <- X[X!=0]
  B <- chainlen
  b <- burnin
  r.sd <- sqrt(r.var)
  p.sd <- sqrt(p.var)
  sigma.sd <- sqrt(sigma.var)

  r.array <- rep(0, B)
  sigma.array <- rep(0, B)
  p.array <- rep(0, B)
  ar.array <- rep(0, B)
  posts <- rep(0, B)

  r.array[1] <- .5
  p.array[1] <- min(mean(X), .5)
  sigma.array[1] <- mean(X==0)/2
  #print(c(r.array[1], p.array[1], sigma.array[1]))
  posts[1] <- logpost(sigma.array[1], p.array[1], r.array[1], Z, N, S, nonzero)
  for(i in 2:chainlen){
    sigma.star <- rtruncnorm(1, 0, 1, sigma.array[i-1], sigma.sd)
    p.star <- rtruncnorm(1, 0, 1, p.array[i-1], p.sd)
    r.star <- rtruncnorm(1, 0, Inf, r.array[i-1], r.sd)
    lpost <-logpost(sigma.star, p.star, r.star, Z, N, S, nonzero)
    reject.prob <- exp(lpost-posts[i-1])*
                   dtruncnorm(sigma.array[i-1], 0, 1,
                              sigma.star, sigma.sd)*
                   dtruncnorm(p.array[i-1], 0, 1, p.star, p.sd)*
                   dtruncnorm(r.array[i-1], 0, Inf, r.star, r.sd)/
                  (dtruncnorm(sigma.star, 0, 1,
                              sigma.array[i-1], sigma.sd)*
                     dtruncnorm(p.star, 0, 1, p.array[i-1], p.sd)*
                     dtruncnorm(r.star, 0, Inf, r.array[i-1], r.sd))
    u <- runif(1,0,1)
    if(u < min(reject.prob, 1)){
      r.array[i] <- r.star
      p.array[i] <- p.star
      sigma.array[i] <- sigma.star
      ar.array[i] <- 1
      posts[i] <- lpost
    }
    else{
      r.array[i] <- r.array[i-1]
      p.array[i] <- p.array[i-1]
      sigma.array[i] <- sigma.array[i-1]
      posts[i] <- posts[i-1]
    }
  }
  inds <- seq(b+1, B, by=thinning)
  return(list(sigma=sigma.array[inds],
              p=p.array[inds],
              r=r.array[inds],
              ar=ar.array))
}
```

A.2 Zero Inflated Poisson code

```r
library(dplyr)
tornado <- data %>% filter(EVENT_TYPE == "Tornado") %>% select(DEATHS_DIRECT)
flood <- data %>% filter(EVENT_TYPE == "Flash Flood") %>% select(DEATHS_DIRECT)

# Precompute values to use in posterior function
zeros <- tornado %>% filter(DEATHS_DIRECT == 0) %>% select(DEATHS_DIRECT)
ones <- tornado %>% filter(DEATHS_DIRECT != 0) %>% select(DEATHS_DIRECT)
n <- nrow(ones)
t0_exp <- (zeros)/factorial(zeros)
l_t1_exp <- ones-1/2

# Define Posterior Functions
log_lam <- function(y, p, lambda){
  t0 <- sum(log(p + (1-p)*exp(-1*lambda)*(lambda+2e-10)^t0_exp))
  t1 <- sum(log(exp(-1*lambda)*lambda^l_t1_exp))
  return(t0+t1)
}

log_p <- function(y, p, lambda){
  t0 <- sum(log(p + (1-p)*exp(-1*lambda)*(lambda+2e-10)^t0_exp))
  t1_vec <- rep(1-p, n)
  t1 <- sum(log(t1_vec))
  return(t0+t1)
}

### Sampler for Tornado
set.seed(2011)

l_last <- 1
p_last <- .98

B <- 20000
lambdas <- rep(0, B)
ps <- rep(0, B)

l_accept <- rep(FALSE, B)
p_accept <- rep(FALSE, B)

for(t in 1:B){
  lambda <- rgamma(1, 3, 2)
  p <- rbeta(1, 1940, 60)
  ro_lam <- log_lam(tornado, p_last, lambda) - log(dgamma(lambda, 3, 2)) -
    log_lam(tornado, p_last, l_last) + log(dgamma(l_last, 3, 2))
  ro_p <- log_p(tornado, p, l_last) - log(dbeta(p, 1940, 60)) -
    log_p(tornado, p_last, l_last) + log(dbeta(p_last, 1940, 60))
  U_lambda <- runif(1)
  U_p <- runif(1)
  if(log(U_lambda) < min(0, ro_lam)){
    print(TRUE)
    lambdas[t] <- lambda
    lambda_last <- lambda
    l_accept[t] <- TRUE
  } else{
    print(FALSE)
    lambdas[t] <- lambda_last
  }
  if(log(U_p) < min(0, ro_p)){
    print(TRUE)
    ps[t] <- p
    p_last <- p
    p_accept[t] <- TRUE
  } else{
    print(FALSE)
    ps[t] <- p_last
  }
}

# Do the same for flash floods

# Update values for likelihood function
zeros <- flood %>% filter(DEATHS_DIRECT == 0) %>% select(DEATHS_DIRECT)
ones <- flood %>% filter(DEATHS_DIRECT != 0) %>% select(DEATHS_DIRECT)
n <- nrow(ones)
t0_exp <- (zeros)/factorial(zeros)
l_t1_exp <- ones-1/2

# Sampler:
set.seed(2012)
l_last <- 1
p_last <- .98
B <- 20000
lambdas <- rep(0, B)
ps <- rep(0, B)
l_accept <- rep(FALSE, B)
p_accept <- rep(FALSE, B)
for(t in 1:B){
  print(paste0("t = ", as.character(t)))
  lambda <- rgamma(1, 1, 2)
  print(paste0("lambda = ", as.character(lambda)))
  p <- rbeta(1, 2945, 55)
  print(paste0("p = ", as.character(p)))
  ro_lam <- log_lam(flood, p_last, lambda) - log(dgamma(lambda, 1, 2)) -
    log_lam(flood, p_last, l_last) + log(dgamma(l_last, 1, 2))
  ro_p <- log_p(flood, p, l_last) - log(dbeta(p, 2945, 55)) -
    log_p(flood, p_last, l_last) + log(dbeta(p_last, 2945, 55))
  U_lambda <- runif(1)
  U_p <- runif(1)
  if(log(U_lambda) < min(0, ro_lam)){
    print(TRUE)
    lambdas[t] <- lambda
    lambda_last <- lambda
    l_accept[t] <- TRUE
  } else{
    print(FALSE)
    lambdas[t] <- lambda_last
  }

  if(log(U_p) < min(0, ro_p)){
    print(TRUE)
    ps[t] <- p
    p_last <- p
    p_accept[t] <- TRUE
  } else{
    print(FALSE)
    ps[t] <- p_last
  }
}
```

A.3 Poisson code

```
set.seed(05112017)

# --- model evaluation function ------------------------------------------------------------
DIC <- function(y, theta, loglik) {
  theta_hat = mean(theta) #theta_hat = apply(theta, 2, mean)
  L = loglik(y, theta_hat)
  S = length(theta) #S = nrow(theta) #S = number of iterations
  llSum = 0
  for (s in 1:S) {
    theta_s = theta[s]
    llSum = llSum + loglik(y, theta_s)
  }
  P = 2 * (L - (1 / S * llSum))
  DIC = - 2 * (L - P)
  return(DIC)
}

# --- poisson log-likelihood ----------------------------------------------------------
pois_ll <- function(x, lambda) {
  ll <- sum(dpois(x, lambda, log = TRUE))
  return(ll)
}

# --- poisson model ------------------------------------------------------------------
pois_fit <- function(data, nsim) {
  x <- data
  xbar <- mean(x)
  n <- length(x)
  lambda <- rgamma(nsim, n * xbar + 1 /2, n)
  return(lambda)
}

# --- results - ----------------------------------------------------------------------------
lambda_t <- pois_fit(data = TORNADO$DEATHS, nsim = 10000)
lambda_ff <- pois_fit(data = FLASH_FLOOD$DEATHS, nsim = 10000)

quantile(lambda_t, c(0.025, 0.5, 0.975))
quantile(lambda_ff, c(0.025, 0.5, 0.975))

DIC(TORNADO$DEATHS, lambda_t, pois_ll)
DIC(FLASH_FLOOD$DEATHS, lambda_ff, pois_ll)
```

A.4 Negative Binomial code

```r
# --- negative binomial log-likelihood -----------------------------------------------------------
nb_ll <- function(x, p) {
  ll <- sum(dnbinom(x = x, prob = p, size = 1 / 4, log = TRUE))
  return(ll)
}

# --- negative binomial model -------------------------------------------------------------------
nb_fit <- function(data, nsim) {
  x <- data
  xbar <- mean(x)
  n <- length(x)
  r <- 1 / 4
  p <- rbeta(nsim, n * xbar, n * r + 1 / 2)
  return(p)
}

# --- results - --------------------------------------------------------------------------

p_t <- nb_fit(data = TORNADO$DEATHS, nsim = 10000)
p_ff <- nb_fit(data = FLASH_FLOOD$DEATHS, nsim = 10000)

# --- credible intervals ----------------------------------------------------------------------
quantile(p_t, c(0.025, 0.5, 0.975))
quantile(p_ff, c(0.025, 0.5, 0.975))

# --- dic ------------------------------------------------------------------------
DIC(TORNADO$DEATHS, p_t, nb_ll)
DIC(FLASH_FLOOD$DEATHS, p_ff, nb_ll)
```

B Derivations

This appendix will include details on the calculations required to derive our models.

B.1 Zero Inflated Poisson Derivation

In obtaining our full conditionals, we can simplify this slightly to obtain the following:

$$p(\lambda|X,p) \propto \prod_{x_i=0}\left[p+(1-p)\frac{e^{-\lambda}\lambda^{x_i}}{x_i!}\right]\prod_{x_i\neq 0}\left[e^{-\lambda}\lambda^{x_i-1/2}\right]$$

$$p(p|X,\lambda) \propto \prod_{x_i=0}\left[p+(1-p)\frac{e^{-\lambda}\lambda^{x_i}}{x_i!}\right]\prod_{x_i\neq 0}\left[(1-p)\right]$$

B.2 Zero Inflated Negative Binomial Derivation

The likelihood for the ZINB is

$$\mathcal{L}(X|\sigma,p,r) = \prod_{i=1}^{N}\sigma I_{X=0}(X_i) + (1-\sigma)\frac{\Gamma(r+X_i)}{\Gamma(r)X_i!}$$

For ease of notation let Z be the number of zero values in X, and N be the total number of observations.

$$
\begin{aligned}
&= \prod_{X_i=0}\left(\sigma + (1-\sigma)p^{X_i}(1-p)^r\frac{\Gamma(r+X_i)}{\Gamma(r)X_i!}\right)\prod_{X_i\neq 0}\left((1-\sigma)p^{X_i}(1-p)^r\frac{\Gamma(r+X_i)}{\Gamma(r)X_i!}\right)\\
&= (\sigma + (1-\sigma)(1-p)^r)^Z\prod_{X_i\neq 0}\left((1-\sigma)p^{X_i}(1-p)^r\frac{\Gamma(r+X_i)}{\Gamma(r)X_i!}\right)\\
&\propto (\sigma + (1-\sigma)(1-p)^r)^Z\prod_{X_i\neq 0}\left((1-\sigma)p^{X_i}(1-p)^r\frac{\Gamma(r+X_i)}{\Gamma(r)}\right)\\
&\propto (\sigma + (1-\sigma)(1-p)^r)^Z\,(1-\sigma)^{N-Z}(1-p)^{(N-Z)r}p^{\sum_{i=1}^{N}X_i}\prod_{i=1}^{n}\left(\frac{\Gamma(r+X_i)}{\Gamma(r)}\right)
\end{aligned}
$$

As mentioned in section 2.4 we take the uniform priors for σ and p as well as the non-informative gamma for r which is $r^{-1/2}$. Therefore my joint posterior is:

$$
p(r,\sigma,p|X) \propto (\sigma + (1-\sigma)(1-p)^r)^Z\,(1-\sigma)^{N-Z}(1-p)^{(N-Z)r}p^{\sum_{i=1}^{N}X_i}r^{-1/2}\prod_{i=1}^{N}\left(\frac{\Gamma(r+X_i)}{\Gamma(r)}\right)
$$

I am now going to take the log of the posterior because it helps with computation

$$
\begin{aligned}
\ln\left(p(r,\sigma,p|X)\right) \propto & Z\ln\left(1 + (1/\sigma - 1)(1-p)^r\right) + Z\ln(\sigma) + (N-Z)\ln(1-\sigma) + (N-Z)r\ln(1-p)\\
& + \sum_{i=1}^{N}X_i\ln(p) - \ln(r)/2 - N\ln(\Gamma(r)) + \sum_{i=1}^{N}\ln(\Gamma(r+X_i))
\end{aligned}
$$